



Universidad del Azuay
Facultad de Ciencia y Tecnología
Escuela de Ingeniería Electrónica

**“Implementación de un sistema de adquisición de datos para
monitoreo de motos de velocidad”**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autores:

PAOLA TATIANA QUEZADA TENESACA

VÍCTOR MANUEL RUBIO CEDILLO

Director:

LEOPOLDO CARLOS VÁZQUEZ RODRÍGUEZ

CUENCA - ECUADOR

2015

DEDICATORIA

Este trabajo les dedico a mis padres, Blanca y Luis, quienes son mi ejemplo y me han apoyado incondicionalmente en mis estudios y proyectos, permitiéndome así cumplir mis metas y sueños.

Le dedico a mi hermano Pablo quien ha sido mi compañero y amigo toda la vida.

Paola.

DEDICATORIA

El presente trabajo va dedicado a mis padres Luis y María quienes han sido y son mi apoyo, mi guía, mi ejemplo y la más grande motivación en mi vida.

Víctor.

AGRADECIMIENTO

Le doy gracias a Dios por haberme regalado salud, felicidad y permitir que cumpla una de mis metas; les doy gracias a mis padres quienes me han apoyado y aconsejado en cada decisión que he tomado, a mis abuelos y hermanos, en especial a mi hermano Pablo por acompañarme en todo momento. A Víctor por apoyarme, ayudarme y por haber realizado conmigo este trabajo; a todos ellos les agradezco por haber estado a mi lado siempre.

A mis amigos con quienes compartí muchos momentos de felicidad; y a mis profesores, en especial al Lcdo. Leopoldo Vázquez y al Ing. Daniel Iturralde, quienes nos apoyaron en este proyecto para que se haga realidad.

Paola.

AGRADECIMIENTO

Agradezco principalmente a Dios, a mis padres, hermanas, cuñados y sobrinos por su apoyo incondicional, así como también a Paola Quezada, con quien día a día hemos trabajado con dedicación y esfuerzo para la realización de este proyecto; quiero agradecer de la misma manera al Lcdo. Leopoldo Vázquez y al Ing. Daniel Iturralde por su ayuda y colaboración en este trabajo.

Víctor.

IMPLEMENTATION OF A DATA ACQUISITION SYSTEM FOR SPEED MOTORCYCLES MONITORING

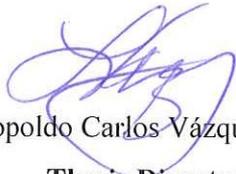
ABSTRACT

A data acquisition system was developed in order to enable the pilot to control different parameters in real time. This system includes speed, rpm, engine temperature, and fuel and tire pressure of a speed motorcycle.

The data are taken from different types of sensors implemented on the motorcycle; these signals are processed and transmitted by Bluetooth Android device where different real time parameters are shown to the pilot.

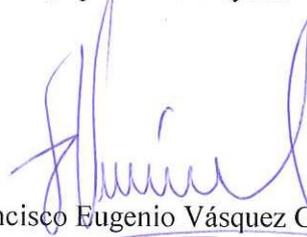
The parameters, which are freely accessible to all users, are also sent to a database and displayed on a website.

Keywords: Data Acquisition, Microcontroller, Web Server, Speed Motorcycle.



Leopoldo Carlos Vázquez Rodríguez

Thesis Director

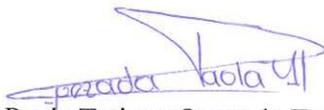


Francisco Eugenio Vázquez Calero

School Director

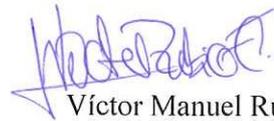


UNIVERSIDAD DEL
AZUAY
ESCUELA
Ingeniería Electrónica



Paola Tatiana Quezada Tenesaca

Author



Víctor Manuel Rubio Cedillo

Author



Translated by,
Lic. Lourdes Crespo

IMPLEMENTACION DE UN SISTEMA DE ADQUISICIÓN DE DATOS PARA MONITOREO DE MOTOS DE VELOCIDAD

RESUMEN

Se desarrolló un sistema de adquisición de datos, los cuales son velocidad, rpm, temperatura del motor, gasolina y presión de los neumáticos de una moto de velocidad, con la finalidad de que el piloto pueda controlar los diferentes parámetros en tiempo real.

Los datos son tomados de los diferentes tipos de sensores que se implementaron en la moto, estas señales son procesadas y transmitidas por Bluetooth a un dispositivo Android, donde se muestran los diferentes parámetros en tiempo real al piloto.

Los parámetros también son enviados a una base de datos y son mostrados en una página web, estos son de libre acceso para todos los usuarios.

Palabras clave: Adquisición de datos, Microcontrolador, Servidor Web, Moto de velocidad.



Leopoldo Carlos Vázquez Rodríguez

Director de Tesis

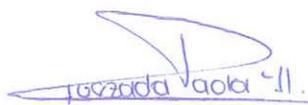


Francisco Eugenio Vázquez Calero

Director de Escuela



UNIVERSIDAD DEL
AZUAY
Escuela de Ingeniería Electrónica



Paola Tatiana Quezada Tenesaca

Autor



Víctor Manuel Rubio Cedillo

Autor

ÍNDICE DE CONTENIDOS

DEDICATORIA	II
AGRADECIMIENTO	IV
RESUMEN	VI
ABSTRACT	VII
ÍNDICE DE FIGURAS	xii
ÍNDICE DE TABLAS	xv
ÍNDICE DE ANEXOS.....	xvi
INTRODUCCIÓN	1
CAPÍTULO 1: GENERALIDADES	3
1.1. Objetivo general.....	3
1.2. Objetivos específicos	3
1.4. Importancia	4
1.5. Ventajas.....	5
1.6. Metodología de la investigación	5
CAPÍTULO 2: MARCO TEÓRICO	9
2.1. Circuitos:.....	9
2.1.1. Divisor de tensión:	9
2.1.2. Conversor de frecuencia a voltaje:.....	10
2.1.2.1. Conversión de frecuencia a voltaje con C.I. 555	10
2.1.2.2. Conversión de frecuencia a voltaje con LM2907	12
2.1.3. Regulador de voltaje	12
2.1.4. Arduino	13
2.1.4.1. Software Arduino:.....	14
2.1.4.2. Hardware Arduino:.....	14
2.1.4.3. Arduino Mega 2560:	15
2.2. Comunicación	17

2.2.1.	Bluetooth.....	17
2.3.	Sistemas operativos móviles	18
2.3.1.	Android	19
2.3.2.	IOS	20
2.3.3.	Windows Phone	20
2.4.	Aplicaciones móviles	21
2.5.	GPS.....	22
2.6.	Sistemas mecánicos y de control:	23
2.6.1.	ECU	23
2.6.2.	CKP.....	24
2.6.3.	Sensores de temperatura	24
2.6.4.	Sensor de presión de los neumáticos	27
2.7.	Servidor y página web.....	28
2.7.1.	Servidor Web	28
2.7.2.	Base de Datos.....	28
2.7.3.	Página Web	29

CAPÍTULO 3: ANALISIS DE TIPOS DE SENSORES Y ACOPLAMIENTO EN LA MOTO

3.1.	Tipos de sensores	30
3.1.1.	Sensor de velocidad:	30
3.1.2.	Sensor de temperatura del motor	35
3.1.3.	Sensor de ángulo de giro.....	36
3.1.4.	Sensor de ángulo de inclinación	38
3.1.5.	Sensor de gasolina	38
3.1.6.	Sensor de presión de los neumáticos	40
3.2.	Ubicación de los sensores en la moto	42
3.2.1.	Diseño de la ubicación de los sensores	43

CAPÍTULO 4: DISEÑO ELECTRÓNICO E IMPLEMENTACIÓN DEL HARDWARE

4.1.	Circuito electrónico.....	46
------	---------------------------	----

4.1.1.	Convertor de frecuencia a voltaje para cálculo de las revoluciones por minuto RPM	46
4.1.2.	Regulador de tensión	49
4.1.3.	Divisor de tensión	50
4.2.	Adquisición de la señal de los sensores de presión de los neumáticos	52
	Módulo TPMS M202L	52
4.2.1.	Características claves del módulo TPMS.....	52
4.2.1.1.	Información específica del TPMS, pruebas y procedimiento general .	56
4.2.2.	Adquisición de la señal	62
4.2.3.	Construcción de un módulo para obtener la presión de los neumáticos..	63
4.3.	Microcontrolador, implementación del <i>hardware</i>	64
4.3.1.	Arduino Mega 2560	64
4.4.	Módulo Bluetooth, implementación del <i>hardware</i>	65
4.4.1.	Módulo Bluetooth HC-06	66
4.4.2.	Conexión del módulo HC-06 con el Arduino	67
4.5.	Dispositivo Android.....	68
4.6.	Interconexión del Sistema de Adquisición de Datos.....	69
CAPÍTULO 5: IMPLEMENTACIÓN DEL SOFTWARE.....		72
5.1.	Arduino, implementación del <i>software</i>	72
5.1.1.	IDE de Arduino.....	72
5.1.2.	Librerías Arduino.....	74
5.1.3.	Esquema del programa de Arduino	75
5.2.	Android, implementación del <i>software</i>	77
5.2.1.	Eclipse.....	77
5.2.1.1.	Eclipse ADT.....	78
5.2.1.2.	Software eclipse	79
5.2.2.	Creación del programa de Android.....	80
5.2.2.1.	Codificación del programa: Sensor de inclinación y del ángulo de giro.....	81
5.2.2.2.	Código del programa: Adquisición de la velocidad.....	82
5.2.2.3.	Código del programa: Comunicación inalámbrica	83
5.2.2.4.	Código del programa: Conexión Web.....	85

5.2.3.	Diagrama de flujo del programa final.....	86
CAPÍTULO 6: CONSULTA DE LOS DATOS GUARDADOS DE LA MOTOCICLETA.....		90
6.1.	Base de datos.....	91
6.2.	Archivo DAS.....	93
6.3.	Registra la base de datos	94
6.4.	Página web	94
6.4.1.	Consulta general	95
6.4.2.	Consulta de la velocidad máxima según cada vehículo.....	96
6.4.3.	Consulta de la velocidad media de cada vehículo	97
6.4.4.	Consulta de la máxima RPM de cada vehículo	97
6.4.5.	Consulta de la media de las RPM de cada vehículo.	98
6.4.6.	Consulta de los registros según la fecha	98
6.4.7.	Consulta de los registros según el nombre del vehículo	99
CAPÍTULO 7: PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS.....		101
7.1.	Diseño y construcción de la motocicleta.....	101
7.2.	Simulación y construcción del circuito electrónico	103
7.2.1.	Pruebas del TPMS M202L	103
7.3.	Pruebas con la tarjeta Arduino Mega 2560.....	106
7.4.	Pruebas de la Aplicación para el dispositivo Android	106
7.5.	Prueba de la consulta de la base de datos.....	107
7.6.	Prototipo final	108
CONCLUSIONES Y RECOMENDACIONES.....		109
BIBLIOGRAFÍA.....		111
ANEXOS		114

ÍNDICE DE FIGURAS

Figura 1: Esquema eléctrico del motor	6
Figura 2: Esquema de un divisor de tensión resistivo.....	9
Figura 3: Circuito Integrado.....	11
Figura 4: Diagramas C.I. LM555.....	11
Figura 5: Diferentes presentaciones de los reguladores de voltaje.	13
Figura 6: Arduino Mega 2560.....	15
Figura 7: Esquema de tipos de sensores de temperatura.....	25
Figura 8: Emisión de la señal del GPS.....	32
Figura 9: Posición del usuario, del satélite y del pseudorange.	33
Figura 10: Sensor de temperatura del motor.....	35
Figura 11: Proyección de los tres ejes de coordenadas de un acelerómetro.	37
Figura 12: Prototipo de un acelerómetro de un dispositivo Android.....	38
Figura 13: Esquema de un sensor de gasolina.	39
Figura 14: Sensor TPMS M202L para moto.....	40
Figura 15: Esquema del sensor de presión de los neumáticos.	41
Figura 16: Prototipo de la moto.	43
Figura 17: Esquema del sistema de control y adquisición de datos.....	45
Figura 18: Esquema del conversor para el cálculo de las RPM.....	49
Figura 19: Regulador de voltaje.....	50
Figura 20: Divisores de tensión.....	51
Figura 21: Módulo del sensor de presión que va ubicado en los neumáticos.....	52
Figura 22: Módulo del sensor de presión de los neumáticos que contiene el display.	54
Figura 23: Esquema del circuito electrónico de los módulos del sensor de presión de los neumáticos.....	56
Figura 24: Tiempo de activación del TPMS 16.432ms.	60
Figura 25: Tiempo de activación del TPMS 30.061s.....	60
Figura 26: Potencia de salida, con antena en polarización horizontal.	61
Figura 27: Potencia de salida, con antena en polarización vertical.....	61
Figura 28: Ancho de banda de la señal del TPMS	62
Figura 29: Módulo Bluetooth HC-6.....	66
Figura 30: Conexión del módulo bluetooth con Arduino.	68

Figura 31: Dispositivo Android.	69
Figura 32: Diseño del circuito electrónico de adquisición de datos de una moto de velocidad.	71
Figura 33: Circuito impreso del sistema de adquisición de datos de una moto de velocidad.	71
Figura 34: Ventana del IDE de Arduino.	73
Figura 35: Librerías de Arduino.	75
Figura 36: Diagrama de flujo para el código de programación del sistema de adquisición de datos usando Arduino.	76
Figura 37: Pantalla del software Eclipse.	79
Figura 38: Pantallas de trabajo de eclipse, a la izquierda se tiene la pantalla gráfica de trabajo y a la derecha se tiene la pantalla de texto.	80
Figura 39: Permisos para el acceso al GPS.	83
Figura 40: Permisos para la comunicación con el módulo bluetooth.	84
Figura 41: Permisos para acceso a la web y a la red.	85
Figura 42: Diagrama de flujo para el código de programación de Android.	88
Figura 43: Interfaz gráfica de la aplicación.	89
Figura 44: Diagrama de flujo para la consulta de datos guardados de la motocicleta.	91
Figura 45: Pantalla principal para crear la base de datos.	92
Figura 46: Creación de la tabla de la base de datos.	92
Figura 47: Tabla DAS.	93
Figura 48: Pantalla página web.	95
Figura 49: Consultas.	95
Figura 50: Datos de la consulta de máxima velocidad.	97
Figura 51: Datos de la consulta de la velocidad media.	97
Figura 52: Datos de la consulta de máxima RPM.	98
Figura 53: Datos de la consulta de la media de las RPM.	98
Figura 54: Consulta de los registros según la fecha.	99
Figura 55: Consulta de los registros según el nombre del vehículo.	100
Figura 56: Diseño del prototipo.	102
Figura 57: Construcción del prototipo.	103
Figura 58: Módulo abierto del sensor M202L.	104
Figura 59: Señal recibida del USRP del módulo M202L.	104

Figura 60: Señal recibida del USRP del módulo M202L.	105
Figura 61: Señal recibida del USRP del módulo M202L.	105
Figura 62: Prototipo final.	108

ÍNDICE DE TABLAS

Tabla 1: Información Técnica del motor Sherco.....	5
Tabla 2: Pines del ECU	7
Tabla 3: Valores calculados de Frecuencia para diferentes RPM.....	47
Tabla 4: Ancho de banda M202L.....	58
Tabla 5: Datos generales del TPMS M202L.....	59
Tabla 6: Tabla de consultas de datos guardados de la moto.	96
Tabla 7: Tabla de los datos de la consulta según la fecha.....	99
Tabla 8: Tabla de los datos de la consulta según el nombre del vehículo.	100

ÍNDICE DE ANEXOS

Anexo 1: Datasheet del Microcontrolador Atmega 2560	114
Anexo 2: Análisis de las señales del motor.....	118

Quezada Tenesaca Paola Tatiana

Rubio Cedillo Víctor Manuel

Trabajo de Graduación

Lcdo. Leopoldo Carlos Vázquez Rodríguez

Marzo 2015.

IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS PARA MONITOREO DE MOTOS DE VELOCIDAD

INTRODUCCIÓN

Hasta el 21 de Septiembre del 2014 no se ha construido ninguna motocicleta 100% ecuatoriana, en el año 2012 cerca de 100,000 motocicletas fueron ensambladas en Ecuador. Para 17 de las ensambladoras, el CKD¹ para el ensamblaje es de procedencia china y no más del 3% son piezas nacionales, estas motocicletas no contienen un sistema de monitoreo y control electrónico para la seguridad del piloto. (Revista Líderes, 2012)

El motociclismo es una actividad de gran importancia a nivel mundial, tanto como medio de transporte o como actividad recreativa y deportiva. Es absolutamente real la necesidad del desarrollo de conocimientos para lograr el avance de la industria ecuatoriana de motocicletas.

El Proyecto Internacional MOTOSTUDENT promueve un desafío entre estudiantes universitarios de todo el mundo; con la construcción, control y monitoreo de una moto de velocidad tanto para carreras o como para uso general; la cual será primeramente puesta a prueba para luego entrar a la competencia. (Motostudent, 2014)

¹ CKD: Kit para ensamblaje, contiene todas las piezas necesarias para armar un objeto.

Al haber sido parte de una competencia de elite en la cual como primer objetivo fue demostrar conocimientos y destrezas para que el equipo destaque sobre los demás competidores, como parte de la prueba de innovación se realizó la implementación de un sistema de adquisición de datos para monitorear diferentes partes de la moto, ya que al tener pleno conocimiento de todas las posibles fallas durante el transcurso de la carrera, se pudo informar al piloto de los daños para que ingrese a Pits y se repare de una forma más rápida ganando valioso tiempo durante este tipo de carreras.

Hoy en día los equipos participantes en las competencias no tienen un monitoreo completo del funcionamiento de la moto en el momento de la carrera, o bien existe un control básico de la moto, es decir se logra monitorear la velocidad, el tiempo y las RPM.

Este proyecto va enfocado a eso, al control del funcionamiento de la moto durante la competencia, adquiriendo los datos del monitoreo en tiempo real y guardándolos en una base de datos la cual será enviada a un servidor donde podrán ser revisados; esto implicaría una ventaja donde el equipo podrá ganar tiempo en la competencia ya que cada segundo es crucial en este tipo de carreras.

Se conocerían todos los errores y problemas en la moto durante la carrera, y al momento que el piloto ingrese a Pits solo se realizarían las modificaciones y arreglos de la misma, ya no se tendría que evaluar todos los parámetros de la moto sino solo ejecutar las acciones requeridas. Por lo tanto satisface las necesidades tanto del piloto como del equipo, como también se disminuirían los accidentes en el momento de la competencia.

Hasta el 2014 en nuestro país se han registrado alrededor de 1.000 muertes por año, en la cual el 15% son por accidente de motocicletas. (Ecuavisa, 2014)

La mayoría de motos no tienen un sistema de control a tiempo real, para la seguridad del piloto. Por lo que la implementación del sistema de adquisición de datos es de gran ayuda para prevenir accidentes ya que el piloto tendrá alarmas en la motocicleta cuando algún parámetro está fallando y se podrá detener a tiempo, o bien podrá revisar su base de datos antes de hacer uso de la moto.

CAPÍTULO 1

GENERALIDADES

En el siguiente capítulo se tratarán los objetivos, la problemática y la metodología de la investigación aplicados a este proyecto.

Donde se da una idea general de los temas a tratar, el propósito y la importancia del mismo en el área a la cual se enfoca la realización de este sistema, ayudando de diversas formas a las competencias de motos y a tener un mayor control del vehículo en el cual fuese implementado.

1.1. Objetivo general

- Implementar un sistema de adquisición de datos para el monitoreo de una moto de velocidad.

1.2. Objetivos específicos:

- Analizar los tipos de sensores que se usaran para el monitoreo de una moto de velocidad.
- Analizar hardware y software utilizado para la comunicación entre Android y Arduino.
- Crear un servidor con el cual podamos acceder a los datos monitoreados de la moto mediante una página Web.
- Implementar un sistema de adquisición de los datos monitoreados de una moto de velocidad.
- Montar y probar el sistema de adquisición de datos en una moto.

1.3. Problemática

Hoy en día los equipos participantes en las competencias no tienen un monitoreo completo del funcionamiento de la moto en el momento de la carrera, o bien existe un control básico de la moto.

La mayoría de accidentes en las carreras de motos se producen por la falta de conocimiento del funcionamiento de la moto por parte del equipo y del usuario; por lo tanto al implementar este sistema, los accidentes en el momento de la competencia pueden disminuir; el piloto tendrá alarmas en la motocicleta cuando algún parámetro está fallando y se podrá detener a tiempo, o bien podrá revisar su base de datos antes de hacer uso de la moto.

El sistema puede ser aplicado a una motocicleta de uso general, por lo que ayudaría a evitar accidentes en las vías públicas, evitando así que las muertes por accidentes de motocicletas disminuyan.

1.4. Importancia

Al implementar este proyecto en los vehículos de competencia se puede tener mayor control de los mismos, ya que se podrían conocer durante el trayecto los datos más relevantes de los mismos, en este caso de la moto, tomando las medidas adecuadas y preventivas si fueran necesarias.

Éste sistema no solo sería de gran ayuda en competencias, sino también en la vida diaria, ya que la moto también es de uso general o bien se podría implementar en otras motocicletas. Así los usuarios que de este tipo de vehículos, podrán tener un diagnóstico rápido de acuerdo a la variación de los diferentes parámetros.

1.5. Ventajas

Que el sistema sea adaptable a cualquier tipo de motocicleta es una ventaja, debido que hasta hoy no existe a la venta solo el sistema de adquisición de datos, sino las motocicletas de marcas distinguidas vienen incorporadas con un sistema similar pero

no completo, únicamente muestran los parámetros generales y en una pantalla no móvil que viene incluida en el vehículo.

Otra ventaja es la seguridad que el sistema brinda al usuario, ya que al tener conocimiento de los parámetros de la moto antes de usarla y al estar andando, puede prevenir cualquier problema.

El usuario tiene la ventaja de controlar los parámetros tomados de la motocicleta, desde su dispositivo Android, o bien desde la página web puede hacer consultas de todos los datos guardados.

1.6. Metodología de la investigación

Investigación de los sensores a utilizar

Método lógico deductivo, Experimental, Empírico.

A los participantes de la competencia MotoStudent se les entrego un motor *Sherco 250 i-R* el cual posee las siguientes características:

Tabla 1: Información Técnica del motor Sherco

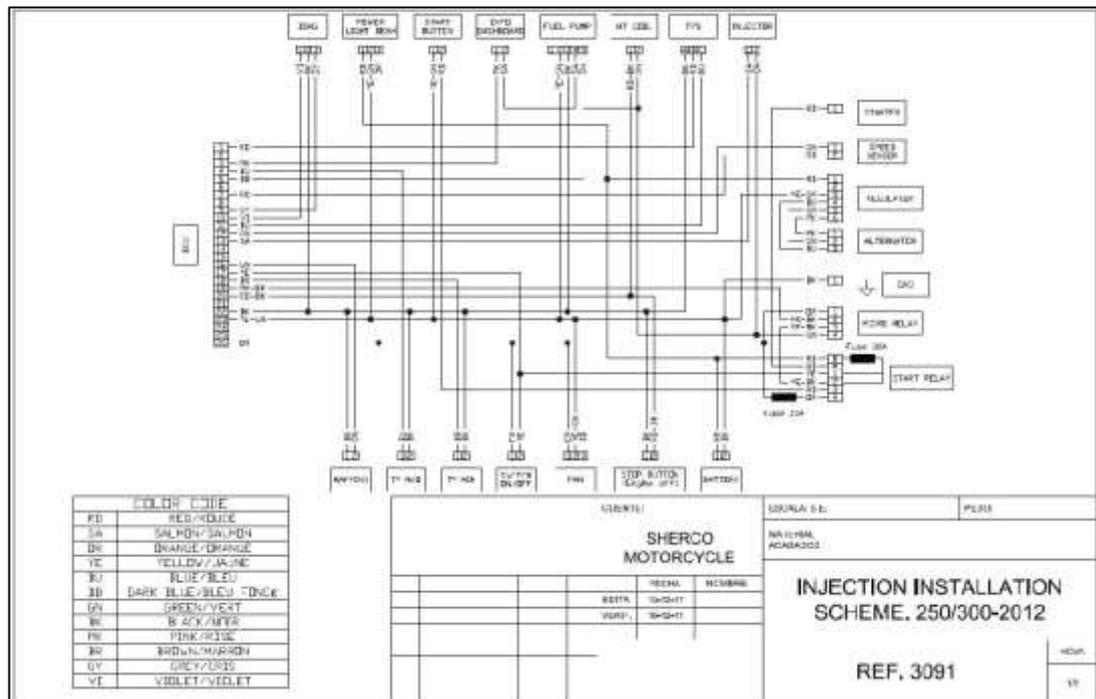
Información Técnica / Technical Information	
Type	strokes 1 cylinder with liquid coolant
Displacement	249.4 CC
Bore/ Stroke	78/52mm
Compression ratio	12.6 :1
Fuel	Super, unleaded, minimum octane index 95
Valve timing	4 valves, DOHC, driven by tooth type chain
Intake valve diameter	29mm
Exhaust valve diameter	25mm
Thickness intake valve	0.15-0.2mm
Thickness exhaust valve	0.2-0.25mm
Crankshat bearing	2 roller bearing
Piston	Aluminium forge

Lubrication pumps	Lubrication under pressure with 2 trochoidal
Motor oil	1 L SAE 10W60
Primary reduction ratio	(21:70)
Gear box:	6 speed
1	(14:33)
2	(17:30)
3	(19:28)
4	(21:26)
5	(23:24)
6	(25:22)
Final pinion	213
Ignition system / Batterie	Electric start/12V 6.5 Ah
Electronic injection	Magneti Marelli

Fuente: Datasheet Sherco

El esquema eléctrico del motor es el siguiente:

Figura 1: Esquema eléctrico del motor



Fuente: Datasheet Sherco.

Tabla 2: Pines del ECU²

PIN	Función
1	5 volts
2	Non conected
3	RPM information
4	Water Termperature signal
5	Ventilator command
6	Non conected
7	RPM sensor
8	Non conected
9	Diagnosis conector
10	Diagnosis conector
11	TPS sensor
12	RPM sensor
13	Injector
14	Non conected
15	Non conected
16	GND for double ignition mapping
17	(+BATT 12v)
18	Water temperature signal
19	Fuel pump relay
20	Ignition spark plug
21	Non conected
22	GND Sensor
23	GND Power
24	Non conected
25	Non conected
26	(+ After Key)

Fuente: Datasheet Sherco.

² ECU: Unidad electrónica de control. Cerebro de un vehículo o motocicleta.

En el esquema eléctrico se muestran las distintas entradas al ECU, de estas las que se utilizarán serán las del sensor CKP³ el cual nos ayudara a conocer las revoluciones por minuto del motor, y la de la temperatura con la cual obtendremos la misma.

Para la obtención de los valores de RMP del CKP se utilizó el método lógico deductivo al tener conocimiento que se debería realizar una conversión de frecuencia a voltaje y que para esto es necesaria la implementación de un circuito electrónico, el cual está establecido en el *datasheet* del LM2907 el cual es un conversor de frecuencia a voltaje.

Para la obtención de la temperatura del motor se utilizó el método lógico deductivo, ya que para medir la temperatura se utiliza una PT100 sobre la cual se tienen conocimientos previos, al estar alimentado con 12v (voltaje de la batería de la motocicleta), este valor será el máximo a entregar por el transductor, si se sabe que la señal máxima que acepta el microcontrolador es de 5v, se concluye que se debe implementar un partidor de tensión para que dicha señal pueda ser leída por el microcontrolador si dañar la integridad del mismo.

Al integrar los diferentes circuitos en uno solo se utiliza el método experimental ya que con conocimientos, pruebas y en base a simulaciones se los pudo unificar, teniendo como resultado un solo circuito.

Para la manipulación de las distintas señales se utiliza el método experimental, al tener conocimientos sobre la programación de microcontroladores y mediante la prueba y error hasta obtener resultados apropiados.

El mismo método es utilizado para la programación de la aplicación Android, del servidor y de la página web, dándonos como resultado el sistema de adquisición de datos completo. (EquadMotosport, 2014)

³ CKP: Sensor magnético o de efecto hall, que envía la información a la computadora de la posición del cigüeñal y las RPM del motor.

CAPÍTULO 2

MARCO TEÓRICO

En éste capítulo se describirá los elementos, *hardware* y *software* que fueron analizados para este proyecto, de aquí se pudo optar por el que tiene mejor características.

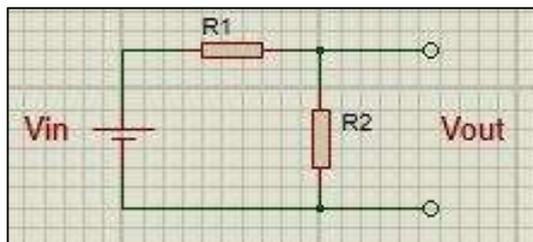
2.1. Circuitos:

2.1.1. Divisor de tensión:

El divisor de tensión es la configuración de un circuito electrónico en el cual un arreglo de resistencias reparte el voltaje según sea requerido; se pueden calcular estas resistencias, conociendo el voltaje de entrada y dando el voltaje de salida requerido.

La fórmula del partidor de tensión se basa en las leyes de Kirchhoff. (Pesantez & Sarmiento , 2012)

Figura 2: Esquema de un divisor de tensión resistivo.



Elaborado por: Los autores.

$$\frac{V_{out}}{R_2} = \frac{V_{in}}{R_1 + R_2} \quad (\text{Ec. 2.1})$$

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (\text{Ec. 2.2})$$

Donde:

- V_{out} : voltaje de salida requerido.
- V_{in} : voltaje de alimentación del sistema.
- R1 y R2: las resistencias requeridas para el divisor de tensión.

2.1.2. Conversor de frecuencia a voltaje:

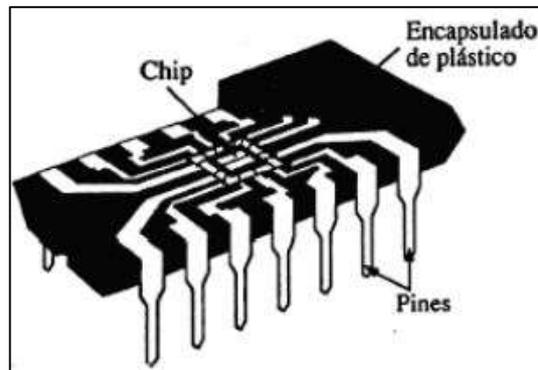
La conversión de frecuencia a voltaje se utiliza para cambiar un tren de pulsos a un valor analógico el cual puede ser interpretado de distintas maneras según el uso que se le dé al mismo; esta conversión es usada por lo general para: Control de velocidad de motores, medición de flujo, demodulación de FM, transmisión de datos, aislamiento de sistemas, enlaces ópticos, interface de transductores con sistemas digitales, multiplexación análoga, control crucero, control de hornos, interruptores por sonido o *touch*, etc.

Se puede realizar la conversión de frecuencia a voltaje por diferentes métodos los más utilizados son mediante un circuito oscilador usando un integrado 555 o un circuito integrado entre esos pueden estar: LM331, LM2907, LM2917, AD650, VFC32, etc. (Cárdenas Espinosa, 2009)

2.1.2.1. Conversión de frecuencia a voltaje con C.I. 555

Circuito integrado: son elementos electrónicos que poseen en su interior circuitos electrónicos muy pequeños, la ventaja de estos chips son la reducción de los costes y la miniaturización de los circuitos electrónicos finales, generalmente están constituidos de silicio, estos pequeños circuitos se encuentran encapsulados en cerámica o plástico con conectores externos que permiten conectarlos con otros componentes o montarlos en otros circuitos. En la Figura 3 se muestra el integrado LM555. (Naranjo, 2008)

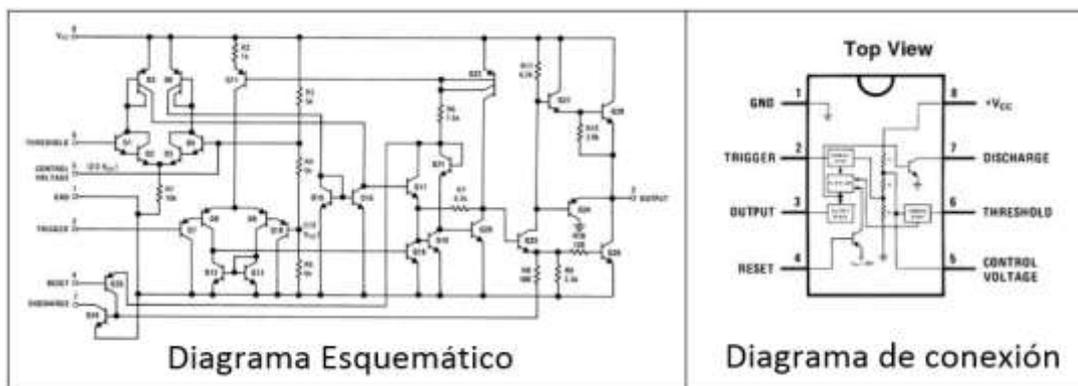
Figura 3: Circuito Integrado.



Fuente: LM555 Datasheet.

C.I. LM555: es un circuito integrado muy estable como temporizador, generando retardos o contabilizando tiempo, la precisión de este temporizador u oscilador depende del cálculo correcto de las resistencias y condensadores externas las cuales le darán el tiempo de retraso o el periodo de oscilación; entre sus aplicaciones podemos encontrar: Temporizador de precisión, pulsos de posición o amplitud modulada, generación de rampa lineal, generación de pulsos entre otros. (Texas Instruments, 2014)

Figura 4: Diagramas C.I. LM555.



Fuente: LM555 Datasheet.

Circuito integrado 555 como conversor de frecuencia a voltaje

Al ser este un oscilador lo podemos configurar como monoestable, la señal de entrada a este circuito va a ser a frecuencia la cual deseamos convertir a voltaje, a la salida tendremos una señal PWM con frecuencia variable, se añade un filtro RC, el cual debe tener por lo menos una frecuencia diez veces mayor a la mínima admisible por la frecuencia; la señal PWM con ayuda de un condensador podremos convertir a una señal diente de sierra, siendo esta una desventaja para este método. En la Figura 4 se muestra el circuito del C.I. LM555.

2.1.2.2. Conversión de frecuencia a voltaje con LM2907

Es un circuito integrado diseñado para la conversión de frecuencia a voltaje entre una de sus varias aplicaciones, entre las ventajas de este integrado están su intercambio de su salida hacia tierra para entradas de frecuencia cero, a más de esto posee un regulador zener en su interior el cual da una conversión frecuencia a voltaje más estable, de la misma manera con las conversiones a corriente, solo con ayuda de una red RC se provee la doble frecuencia.

Para el cálculo del voltaje de salida se emplea la siguiente fórmula:

$$V_{out} = f_{in} \times R1 \times C1 \times V_{cc} \quad (\text{Ec. 2.3})$$

De esta fórmula podemos calcular las resistencias requeridas para un voltaje de salida máximo, con una frecuencia de entrada dada. (National Semiconductor, 1995)

2.1.3. Regulador de voltaje

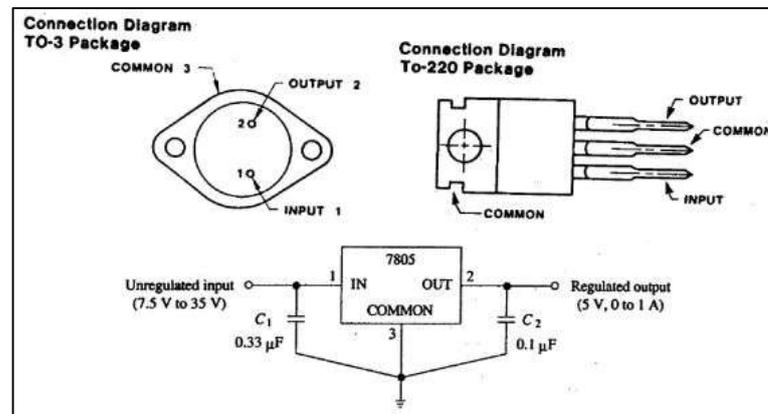
Los reguladores de voltaje nos permiten obtener una salida continua sin fluctuaciones con respecto a la carga o la señal de entrada, la salida es entregada como una tensión continua.

Existen diferentes tipos de reguladores como se muestra en la Figura 5; en la actualidad los más utilizados son los circuitos integrados, siendo preferidos por sus bajos costos, menor espacio en el circuito y su confiabilidad, estos chips ajustan un voltaje de entrada a un voltaje de salida para que pueda ser usado por los distintos

circuitos protegiendo los mismos de sobrecargas de tensión, en el mercado encontramos diferentes reguladores de tensión los cuales se ajustan a voltajes tanto positivos como negativos, los mismo pueden tener diferentes valores de tensión.

Existen también reguladores de tensión ajustables los cuales, nos permiten tener una tensión de salida específica, la cual es calculada mediante un arreglo de resistencias. (Drake Moyano, 2005)

Figura 5: Diferentes presentaciones de los reguladores de voltaje.



Fuente: 7805 Datasheet.

2.1.4. Arduino

Arduino fue concebido como un proyecto en el 2005, el objetivo de este era diseñar una plataforma para la programación de microcontroladores económica, de fácil programación y con tecnología *Plug and Play*, haciéndola independiente de otros dispositivos al momento de grabar los microcontroladores, haciéndola de esta manera competitiva en el mercado, fácil uso y eficaz.

Las tarjetas arduino son circuitos impresos que por lo general utilizan microcontroladores AVR, estos circuitos impresos ofrecen las entradas y salidas respectivas de cada microcontrolador en ellas, a más de los diferentes puertos de comunicación, ya sea con la PC o con otros dispositivos.

2.1.4.1. Software Arduino:

Este software se puede utilizar en los diferentes sistemas operativos como: Windows, Linux e Mac OS, este se puede descargar libremente para su uso, el lenguaje de programación que utiliza es un programa propio de Arduino basado principalmente en C y C++, sin embargo se pueden utilizar diversos software y lenguajes para la programación de estos microcontroladores. (Margolis, 2011)

2.1.4.2. Hardware Arduino:

La placa arduino tiene un microcontrolador Atmel AVR de 8 bits el cual es pregrabado con un programa de gestor de arranque, que le permite que sea reconocido como arduino.

Las placas arduino poseen un conector de alimentación o bien se pueden alimentar únicamente conectando el puerto USB. Las placas se alimentan con 5v de corriente continua pero también pueden alimentarse con 3.3v de corriente continua.

Para la comunicación y programación poseen otro microcontrolador programado como un convertor de USB a Serial, en algunas placas el microcontrolador principal realiza esta función por USB.

Existe una variada gama tanto de tarjetas como de *shields*, estas varían en los microcontroladores que poseen, en los puertos de comunicación, tamaño, uso, etc. Entre esta variedad de tarjetas encontramos la tarjeta Arduino Mega la cual fue utilizada para este desarrollo, pero su número de puertos, velocidad de procesamiento, costo, tamaño, etc.

Existen varios tipos de módulos para esta tarjeta, las cuales brindan diferentes aplicaciones a las mismas como: comunicaciones inalámbricas (wi-fi, RF), conexión USB, pantallas LCD, entre otras; con estos dispositivos adaptables a las tarjetas Arduino su uso y aplicaciones son innumerables y fáciles de implementar tanto para profesionales como para estudiantes y aficionados al área. Para la comunicación se utilizara el módulo bluetooth, ya que permite la comunicación entre el dispositivo Android y el Arduino. (Massimo Banzi, 2011)

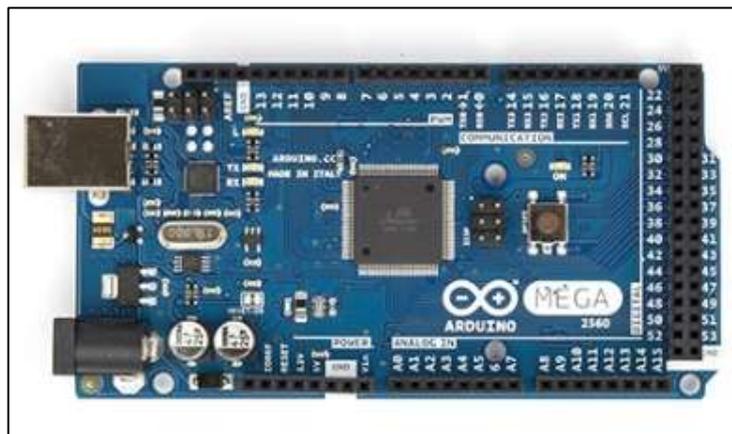
2.1.4.3. Arduino Mega 2560:

Esta placa se basa en un microcontrolador Atmega2560, cuenta con 16 entradas analógicas, 54 pines digitales de entrada y salida de los cuales 15 se pueden utilizar para PWM, 4 puertos seriales, el oscilador del cristal es de 16MHz, tiene 256Kb de memoria, los cuales son compartidos entre la memoria para el *bootloader*, la SRAM, la EEPROM y para el almacenamiento del código, cuenta con interrupciones tanto internas como externas. Posee un microcontrolador convertidor de USB a Serial para la comunicación con el computador.

La placa como se muestra en la Figura 6, tiene un Jack para su alimentación, se puede alimentar con un rango de voltajes entre siete y doce voltios ya que su voltaje de operación es de cinco voltios. Para la comunicación con la PC en su *hardware* viene integrado un puerto USB. (Bell, 2013)

En los anexos se podrá observar la asignación de los pines para el Atmega2560.

Figura 6: Arduino Mega 2560.



Fuente: Arduino.

Sus principales características son:

- Canales de PWM (*Pulse Width Modulation*)
Quince canales, de los pines 2 al 13 y del 44 al 46.
- Interrupciones externas
Pines: 2 – 3 – 18 – 19 – 20 y 21.
- Sistema de comunicación serial USART (*Universal Asynchronous Receiver – Transmitter*)

	Recepción	Transmisión
	Rx	Tx
Serial 0	0	1
Serial 1	19	18
Serial 2	17	16
Serial 3	15	14

- Sistema de comunicación SPI (*Serial Peripheral Interface*)

Pines	Sistema
50	MISO
51	MOSI
52	SCK
53	SS

- Sistema de comunicación I2C (*Inter-Integrated Circuit*)

Pines	Sistema
20	SDA
21	SCL

2.2. Comunicación

Las telecomunicaciones son los distintos métodos o técnicas por las cuales se transmite información entre dos puntos o de un punto a varios puntos, entre estas técnicas de comunicación están las guiadas y no guiadas.

Comunicaciones guiadas: este tipo de comunicaciones se envían mediante líneas de comunicación como bus de datos, par de cobre, fibra óptica, etc.

Comunicaciones no guiadas: estas comunicaciones son transmitidas utilizando como canal el aire, son enviadas por medio de radiofrecuencia, ondas sonoras, ondas de luz, etc.

Para este proyecto se han considerado tres tipos de comunicación los cuales son: radiofrecuencia, comunicación USB y vía bluetooth.

2.2.1. Bluetooth

Es una especificación industrial que facilita el envío y recepción de datos y voz inalámbricamente entre dos dispositivos cercanos, opera en la frecuencia de 2,4 GHz, la misma frecuencia utilizada por las comunicaciones WI-FI, la diferencia entre estas dos reside en la potencia con la que estas se transmite, siendo la potencia del WI-FI mucho más grande permitiendo así comunicaciones a grandes distancias; debido a la baja potencia de emisión del bluetooth permite comunicaciones entre dispositivos cercanos relativamente, una de las facilidades del bluetooth es que pueda encontrar y comunicarse con otros dispositivos que cuenten con esta tecnología, ya que no se requiere de configuración previa de protocolos u otras condiciones para la transmisión de datos.

Existen diferentes categorías de esta tecnología, se numeran de la uno a la cuatro, las diferencias entre las mismas está en su alcance y en su capacidad de transmisión. Pudiendo llegar a una distancia máxima de 100m o a una velocidad de transmisión de hasta 3 Mbits/s nominales, haciendo del Bluetooth una tecnología eficaz, confiable y segura. (Torrente Artero, 2013)

2.2.2. Comunicación por radiofrecuencia

Al enviar información por este tipo de telecomunicación se transmiten los datos por medio de ondas radio eléctricas, la este tipo de comunicación opera entre los 3 Hz y los 3 GHz, para que los datos puedan ser transmitidos es necesario modular la señal, existen varios tipos de modulación siendo la modulación en frecuencia y amplitud los más utilizados. (Canga Feito, 2011)

2.2.3. Comunicación USB

Comunicación USB que proviene las siglas Universal Serial Bus, trabaja básicamente como la comunicación serial al enviar los datos por una línea con intervalos de tiempo para la diferenciación de los mismos, pero opera bajo sus propios estándares, este tipo de comunicación utiliza cuatro hilos, de los cuales dos son de alimentación y los otros dos son un par de cobre trenzado por el cual fluye la información. La comunicación de datos por este medio es de alta fiabilidad a más de poder transmitir a altas velocidades. (Bojorges & Rodríguez, 2004)

2.3. Sistemas operativos móviles

Estos sistemas operativos fueron creados con el fin de poder controlar los dispositivos móviles, el primer sistema operativo móvil llamado Windows CE fue creado para controlar una Palm en el año de 1996, posteriormente aparece Symbian en el año 2000, el cual se consideró como el primer sistema operativo para teléfonos inteligentes, este sistema operativo controlaba los teléfonos móviles Ericsson, a lo largo del tiempo nuevos sistemas operativos se fueron desarrollando, algunos de los cuales aún perduran, de la misma manera otros fueron desapareciendo como Symbian, con el desarrollo de nuevos y mejorados sistemas operativos fueron apareciendo cada vez mayores atractivos en estos dispositivos móviles como por ejemplo en el año 2007 Apple muestra su sistema operativo para móviles denominado IOS, el cual poseía capacidades de reproducción multimedia, navegador web y también era un teléfono móvil.

Algunos sistemas operativos fueron creados con el único objetivo de controlar dispositivos móviles, entre estos Windows Phone y Symbian, mientras que se desarrollaron también sistemas operativos multiplataforma los cuales pueden comandar hasta computadoras como IOS y Android.

En la actualidad son tres los sistemas operativos móviles más utilizados y reconocidos estos son: Android, Windows Phone e IOS. (Déniz, Salido, & Bueno, 2013)

2.3.1. Android

Android es un sistema operativo desarrollado por la empresa *Open Handset Alliance*, este es un grupo empresarial dedicado al desarrollo e impulso de los estándares abiertos para dispositivos móviles, el desarrollo de este sistema operativo fue desde el comienzo el principal objetivo de este grupo empresarial el cual fue comprado en el año 2005 por Google, este último después de adquirirlo hizo de Android un código libre, su lanzamiento oficial fue a finales del año 2008.

Este sistema operativo fue implementado en lenguaje de programación C, C++ y Java, la última versión estable en la que podemos encontrarlo es en 5.0.2 (*Lollipop*), una característica de este sistema operativo es que sus diferentes versiones poseen diferentes nombres de postres y estos van en orden alfabético, (*Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat* y el último mencionado anteriormente *Lollipop*), al ser un sistema operativo de código abierto cada plataforma en la que se instala este sistema operativo tiene alguna variación del mismo cambiando ficheros, imágenes o cualquier detalle para caracterizar su marca.

Las aplicaciones para Android se las puede encontrar en su tienda virtual a la cual se puede acceder desde cada dispositivo o la web simplemente con una cuenta Google, aquí podremos encontrar una infinidad de aplicaciones tanto pagadas como gratuitas, a más de esto se pueden desarrollar en Java aplicaciones personales para cada dispositivo si se cuenta con el conocimiento necesario.

Inicialmente fue pensado para controlar teléfonos inteligentes y tablets principalmente pero con el tiempo se lo ha adaptado para muchos artículos de la vida cotidiana como televisores, relojes, refrigeradoras y más.

Los dispositivos Android actualmente son los más vendidos en el mercado después de estos vienen los dispositivos con IOS y en tercer puesto los dispositivos con Windows Phone. (Déniz, Salido, & Bueno, 2013)

2.3.2. IOS

IOS es un sistema operativo desarrollado por Apple, principalmente fue dirigido para controlar sus dispositivos móviles, los cuales inicialmente eran el Iphone y los reproductores Ipod, pero con el tiempo y la innovación también se cargó este sistema operativo en tablets (Ipads) y reproductores de *TV streaming* (Apple TV). Hace su aparición en el año 2007 con el lanzamiento del Iphone OS en el mes de junio, este sistema operativo es derivado del Mac OS X, lo cual indica que parte de los lenguajes en los cuales fue escrito son: C, C++, Objective-C y Swift.

A diferencia de Android y Windows Phone el sistema IOS solo utilizan los productos de Apple este no permite la instalación de su software en otras plataformas; la última versión estable en la que la podemos encontrar es la 8.1.2 y estando la versión 9.0 en pruebas, puede ser operado en treinta y cuatro diferentes idiomas.

Las aplicaciones para este sistema operativo pueden ser descargadas desde su tienda virtual (APP Store), y pueden ser desarrolladas y cargadas mediante Itunes, el cual es el gestor de contenidos de este sistema, mediante Itunes se puede cargar todo tipo de archivo (compatible) al dispositivo móvil. (Apple, 2014)

2.3.3. Windows Phone

Este sistema operativo móvil fue desarrollado por Microsoft, fue concebida con el fin de ocupar el lugar de su predecesor Windows Mobile ya que este último estaba

orientado a personas más ejecutivas, el nuevo sistema operativo debía ser orientado a desarrollo y al mercado que estaba en crecimiento.

Windows Phone es uno de los más recientes sistemas operativos móviles, fue lanzado en el año 2010, la última versión estable que encontramos de este sistema es la 8.1 y la actualización de la misma está en pruebas.

Este nuevo sistema operativo está codificado en *Expression Blend*, *Visual Studio*, *Silverlight* y el *framework XNA*; al igual que los demás sistemas operativos este posee su propia tienda virtual llamada *Windows Phone Marketplace*, pero los desarrolladores y aficionados pueden crear sus propias aplicaciones. (Microsoft, 2014)

2.4. Aplicaciones móviles

Una aplicación móvil es un *software* desarrollado con cierto propósito, utilizado en cualquier dispositivo móvil compatible con la misma, existen diferentes tipos de aplicaciones móviles, entre estas están:

- **Aplicaciones móviles nativas:** son las aplicaciones que vienen instaladas de fábrica y ayudan al sistema operativo con funciones específicas, un ejemplo de este tipo de aplicaciones podría ser la aplicación que ayuda con la actualización del *software* o la que nos facilita el trabajo de conectarnos a una red WI-FI.
- **Aplicaciones móviles utilitarias:** estas nos ayudan en tareas cotidianas como recordatorios, o cálculos un claro ejemplo de estas son los calendarios, la calculadora, las notas, etc.
- **Aplicaciones Web:** este tipo de aplicaciones se conectan con servidores externos o con la web, ya sea para consultas, registro de datos, acceso a cierta información, etc.

- Aplicaciones de entretenimiento: como su nombre lo indica estas están orientadas al entretenimiento del usuario, ya sea como reproductor de audio o video o juegos.

La plataforma de desarrollo de las aplicaciones móviles depende del sistema operativo en el cual se la vaya a instalar, de esto también depende el lenguaje de programación, cada plataforma distribuye una infinidad de aplicaciones en sus tiendas virtuales. (Federal Trade Commission, 2011)

2.5. GPS

Proviene de las siglas *Global Position System* que traducido al español quiere decir Sistema de Posicionamiento Global, fue iniciado en el año 1973, este sistema fue creado con fines militares en un principio, desarrollado por el departamento de defensa de los Estados Unidos de América.

Este sistema de geo-posicionamiento está constituido por 29 satélites, de los cuales 5 están de respaldo, los satélites están en 6 distintas orbitas alrededor del planeta, se lanzó el primer satélite en 1978.

Como su nombre lo indica el propósito específico de este sistema es la geo localización mediante coordenadas, para la determinación de dichas coordenadas el sistema se basa en la distancia del punto del cual se requiere la ubicación hasta cuatro diferentes satélites los cuales tienen coordenadas conocidas, las señales de los satélites es emitida en el rango de los 1500 MHz.

Este servicio es gratuito y se puede utilizar las 24 horas del día sin problema alguno, si bien en su principio fue concebido con fines militares en la actualidad existe un sin número de aplicaciones a este servicio como son:

- Posición y trayectoria de un vehículo en una cartografía digital.
- Localización de cualquier dispositivo que posea las características necesarias y este configurado para la petición y recepción de la ubicación.

- Orientación y navegación a un sitio dado desde un punto de referencia o un vehículo en movimiento en tiempo real.
- Control de deformación terrestre.
- Estudio de evolución de cuencas fluviales.
- Levantamientos batimétricos
- Control e inventarió de distribuciones viales.

En la actualidad muchos de los dispositivos denominados Smart tienen integrado un módulo GPS, haciendo del GPS una herramienta fundamental y de gran ayuda para el día a día. (Huerta, Mangiaterra, & Noguera, 2005)

2.6. Sistemas mecánicos y de control:

2.6.1. ECU

ECU son las siglas de Unidad Electrónica de Control, esta unidad se la relaciona con el cerebro de un vehículo ya que esta recibe, se procesa y envía toda la información sobre el estado del motor, esta unidad es básicamente un circuito con varios componentes ensamblados y programados con una configuración tal para el control de todos los procesos que ocurren en el vehículo en el cual se lo ha instalado, al ECU pueden llegar tanto pulsos digitales en frecuencias variadas como señales analógicas toda esta información es tratada, procesada, y entendida; de acuerdo a ciertos valores o cambios que se realice en el vehículo el ECU envía distintas señales a diversos sensores o actuadores para el correcto funcionamiento del vehículo.

Entre las diferentes tareas del ECU encontramos las siguientes: Control de la inyección de combustible, control del tiempo de inyección, Control de la bomba de combustible, control del Relenti, Control del regulador de presión, Control del régimen de marcha en vacío, etc. (Laica Cunalata, 2012)

2.6.2. CKP

Proviene de las siglas *Crankshaft Position Sensor*, que en español quiere decir Sensor de posición del cigüeñal, está constituido por un imán permanente, un rotor dentado de hierro, un núcleo de hierro el cual dirige el flujo magnético y una bobina enrollada por la cual circula la tensión inducida, la ubicación de este sensor generalmente se encuentra en la parte baja del motor, incrustado en el bloque de cilindros o a un lado de la polea principal del cigüeñal.

Se puede utilizar el CKP para diferentes propósitos como puede ser:

- Determinar la posición del cigüeñal, de esta manera se podría controlar la ignición y la inyección del combustible.
- Para obtener la velocidad de las ruedas.
- Para el controlar la tracción. (Laica Cunalata, 2012)

2.6.3. Sensores de temperatura

Como su nombre lo indican estos tipos de sensores ayudan a determinar la temperatura de un medio o de un objeto, para esto hoy en día existen innumerables tipos de sensores de temperatura, hay diferentes tipos de los mismos de acuerdo al medio, su uso o su funcionamiento, existen cuatro clases de dispositivos para la medición de temperatura, estos pueden ser: Eléctricos, Mecánicos, por Radiación Térmica, etc. (Booster, 2009)

Figura 7: Esquema de tipos de sensores de temperatura.

Eléctricos	Mecánicos	Radiación Térmica	Varios
<ul style="list-style-type: none"> • Termocuplas • Termorresistencias • Termistores • Diodos • Sensores de silicio con efecto resistivo 	<ul style="list-style-type: none"> • Sistemas de dilatación • Termómetros de vidrio con líquidos • Termómetros bimetalógicos 	<ul style="list-style-type: none"> • Pirómetros de radiación • Termómetros infrarrojos 	<ul style="list-style-type: none"> • Indicadores de color • Sondas neumáticas • Sensores ultrasónicos • Indicadores paramétricos • Termómetros acústicos • Cristales líquidos • Sensores fluidicos • Indicadores de luminiscencia • (Termografía)

Fuente: Los autores.

En la Figura 7 se puede observar un esquema de los diferentes tipos de sensores de temperatura, a continuación se explican los tres sensores eléctricos más utilizados:

- **Termocuplas**

Estos sensores están hechos básicamente de dos alambres unidos generalmente soldados, estos dos alambres deben ser de distinto material, esta diferencia permite que se genere una pequeña diferencia de potencial al existir una diferencia térmica entre sus extremos, hay dos puntos para la medición, la unión fría o punto de referencia y la unión caliente o punto de medida.

Este tipo de sensores son más utilizados a nivel industrial para determinar las temperaturas en los hornos de fundición de metales, existen varios tipos de Termocuplas la diferencia entre estos son los materiales de los que está compuesto y de esto también depende la capacidad de soportar calor.

- **Termoresistencias**

Este tipo de transductores térmicos se basan principalmente en el cambio de la resistencia proporcionalmente a la variación de la temperatura del medio en la que se encuentra, son generalmente un metal encapsulado, un buen metal para este

transductor es el Platino ya que una termoresistencia de este material puede ser usado para mediciones que van desde los -260°C hasta los 630°C , siendo este material uno de los más exactos, a las termo resistencias de este metal se las llama PT100, a cero grados centígrados tienen un valor óhmico igual a cien con un intervalo de 38,5 ohms.

La fórmula para encontrar el valor de la resistencia a una temperatura medida es la siguiente:

$$R(t) = R_0(1 + at) \quad (\text{Ec. 2.4})$$

Donde:

- **$R(t)$** : es el valor de la resistencia a un valor de temperatura medido.
- **R_0** : es el valor de la resistencia a 0°C
- **a** : es el valor del coeficiente de variación de la resistencia en función de la temperatura.
- **t** : es el valor de la temperatura medida.

• Termistores

Estos transductores de igual manera que los anteriores varían su resistencia de acuerdo al cambio de temperatura la diferencia entre estos dos transductores reside en que la variación de resistencia de los termistores no es lineal.

Hay dos tipos de termistores los NTC (Coeficiente de temperatura negativo) y los PTC (Coeficiente de temperatura positivo).

- NTC: al subir la temperatura del medio en el que se encuentra la resistencia de este termistor baja, de ahí el nombre de Negativo.
- PTC: Si la temperatura que está siendo medida por este termistor sube, esto causara que la resistencia del mismo también se eleve.

Uno de los inconvenientes de este transductor es su no linealidad; haciendo esto un poco menos efectivos que los métodos anteriormente mencionados. (Bausa Aragonés, y otros, 2004)

2.6.4. Sensor de presión de los neumáticos

La tecnología avanza y se crean dispositivos que ayudan a prevenir cualquier accidente a las personas. Los sensores de presión de los neumáticos son uno de ellos, técnicamente se les conoce como TPMS (sistema de monitorización de la presión de los neumáticos), su función es avisar al piloto una falla en la presión de los mismos, evitando así un consumo mayor del carburante, que el vehículo pierda adherencia con el suelo, que pierda estabilidad, que el neumático reviente, entre otras cosas.

Estos sensores envían la señal a un receptor el cual muestra datos o tiene una alarma o señales luminosas para que el piloto esté al tanto de lo que sucede.

Existen dos tipos de sistemas de monitorización de la presión de los neumáticos, que son el TPMS directo y el TPMS indirecto o ITPMS.

En el TPMS directo se coloca un sensor en cada neumático, el cual contiene una batería interna y envía la señal a un receptor para que este se encargue de mostrar los datos o alarmas al conductor.

En el TPMS indirecto no se coloca ningún sensor físico en los neumáticos, mide la presión a partir de la velocidad de giro de los neumáticos, está ubicado en la centralilla del ABS, ofrece valores relativos. (Chen, Chen, Gan, & Liu, 2005)

Los sensores más usados hoy en día son los TPMS. Existen varios tipos de sensores TPMS en el mercado, hay sensores que va entre el aro y la llanta y sensores tipo tuerca.

Los sensores que van entre la llanta y el aro son más confiables debido que miden la presión de los neumáticos internamente dando un dato real sin error, otra ventaja es que miden la temperatura del neumático, lo que ayuda a tener un mejor control y prevenir un accidente.

Los sensores tipo tuerca, únicamente miden la presión de los neumáticos, estos se colocan como tapas de las válvulas, por lo que la vibración puede hacer que nos dé un margen de error pequeño.

2.7. Servidor y página web

2.7.1. Servidor Web

Un servidor web aloja páginas web o aplicaciones web las cuales son consultadas o solicitadas por los usuarios a los cuales se les llama clientes, este servidor no es más que una aplicación que se mantiene corriendo todo el tiempo en espera de una solicitud, requerimiento o consulta; para la transferencia de páginas HTML los servidores utilizan el protocolo HTTP (protocolo de transferencia de hipertexto).

Existen servidores internos o externos, a la hora de alojar una página web es necesario ver lo más conveniente si este servicio debe estar dentro o fuera de la organización.

Existe una gran variedad de servidores entre estos encontramos: Apache, Microsoft, Sun, nginx, Google, etc; siendo Apache el más utilizado a nivel mundial. (Ambomo, 2012)

2.7.2. Base de Datos

Una base de datos web es la recopilación y almacenamiento de ciertos datos con alguna relación entre sí, a los cuales se pueden acceder mediante solicitudes o búsquedas.

Estos datos pueden ser manipulados, y editados a conveniencia de un usuario con los permisos requeridos, toda base de datos está constituida por una o más tablas las cuales tienen columnas y filas con un número acorde a su información. (Competencias en TIC, 2011)

Hay distintos tipos de bases de datos, estas pueden ser *MySQL*, *Oracle*, *GNU*, *FireFox*, *Access*, *Microsoft SQL Sybase*, etc; el criterio para escoger una de estas bases de datos es el costo ya sea el inicial o el de mantenimiento, y sobre todo su desempeño, su capacidad de almacenamiento y su robustez. (Donoso, 2004)

2.7.3. Página Web

Las páginas web por lo general están codificadas en HTML o XML, estas poseen cierta información la cual es cargada por el desarrollador o por quien la maneje, estas pueden tener enlaces, videos, imágenes, música ya sea para entretenimiento, información, actualización, aprendizaje, etc; todo depende del uso o usuario al que vaya dirigido la página.

Estas están alojadas en servidores ya sean locales o externos, mediante estas páginas web podemos hacer consultas o manejar las bases de datos web.

Por lo tanto podemos concluir que las páginas web son documentos electrónicos accesibles por todo el mundo al estar adaptada a la *World Wide Web* (WWW). (Gallardo Benitez, 2008)

CAPÍTULO 3

ANÁLISIS DE TIPOS DE SENSORES Y ACOPLAMIENTO EN LA MOTO

En este capítulo se detallará los diferentes tipos de sensores que existen en el mercado y el respectivo análisis que se realizó para elegir los que mejores características tenían para el proyecto. Se describirá el procedimiento del acoplamiento de cada sensor que se colocó en la moto.

3.1. Tipos de sensores

3.1.1. Sensor de velocidad:

Como se explicó en el primer capítulo existen varios tipos de sensores para velocidad, se eligió el sensor de velocidad del dispositivo Android por medio del Sistema de Posicionamiento Global conocido como GPS.

Se optó por esta opción debido que el dispositivo está ubicado de manera permanente en la moto por lo que podrá tomar los datos a tiempo real, como también es ergonómico para el diseño de la moto y permite una mejor presentación de la misma.

Velocidad por medio del Sistema de posicionamiento Global, GPS:

El GPS es un sistema de radionavegación que se encarga de recibir señales desde una constelación de varios satélites, a partir de éstas calcula la posición de una persona u objeto, el tiempo y la velocidad, los cuales requieren de un equipo receptor especial.

Con esos datos que se obtiene de las señales, el sistema calcula la velocidad mediante operaciones matemáticas o algoritmos de tipo recursivo, es decir llama al mismo algoritmo varias veces hasta resolver el problema.

La velocidad del GPS se presume que es exacta pero se debe tomar en cuenta que tiene un retraso de un segundo que es el tiempo que tarda el sistema en medir la velocidad, lo que conlleva un error del 4%, aproximadamente. (Alvaro, 2014)

Calculo de la velocidad

Si el movimiento es constante, una de las formas de cálculo de la velocidad es a partir de las posiciones calculadas en dos instantes de tiempo diferentes.

$$\vec{v} = \frac{d\vec{r}}{dt} \approx \frac{\vec{r}(t_2) - \vec{r}(t_1)}{t_2 - t_1} \quad (\text{Ec. 3.1})$$

Para el cálculo de la velocidad instantánea nos basamos en el Desplazamiento Doppler.

El Efecto Doppler es el cambio de la frecuencia de una onda, debido al movimiento del emisor respecto al receptor. (Torres, 2011)

Por la tanto si tenemos al observador acercándose o alejándose a la fuente se define:

Velocidad de las ondas con respecto al observador

$$v' = v + v_0 \quad (\text{Ec. 3.2})$$

La longitud de onda será la misma si la velocidad no varía

$$f = \frac{v}{\lambda} \quad (\text{Ec. 3.3})$$

El observador al acercarse a la fuente escuchará un sonido más agudo, por lo tanto la frecuencia es mayor; y si se aleja, la frecuencia es menor

$$f' = \frac{v'}{\lambda}$$

$$f' = \frac{v}{\lambda} \pm \frac{v_0}{\lambda}$$

$$f' = f + \frac{v_0}{\lambda}$$

$$f' = f * \left(1 \pm \frac{v_0}{f * \lambda}\right)$$

$$f' = f * \left(1 \pm \frac{v_0}{v}\right) \quad (\text{Ec. 3.4})$$

Partiendo de éste efecto tenemos un conjunto de ecuaciones y matrices de las que se deduce la velocidad \vec{u} del receptor. Para esto se debe basar en el cálculo de la

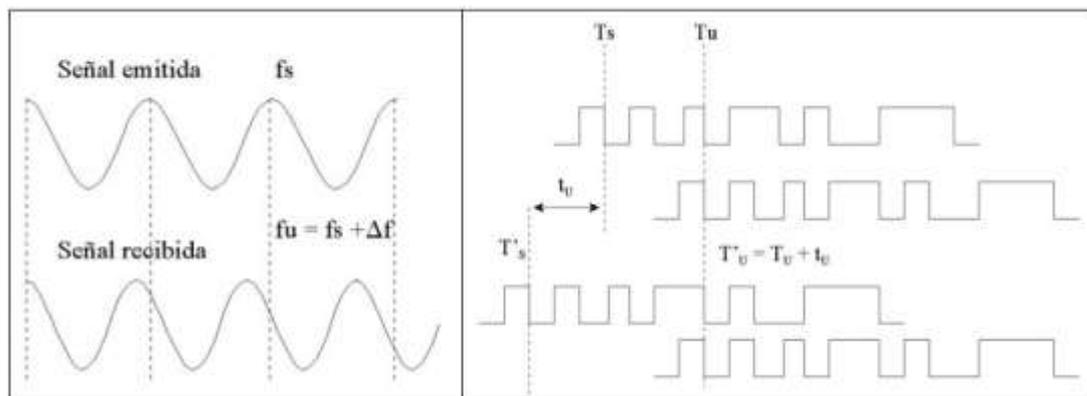
posición, donde el receptor mide la diferencia de tiempo que hay entre el momento que se recibe la señal y el instante que se considera que se envió la señal, así se calcula el tiempo que paso para que las secuencias pseudoaleatorias se alineen.

Es decir, si se tiene el GPS por un lado y el satélite por otro, ambos generan una señal auditiva al mismo tiempo, si se pudiera oír (no se puede oír el sonido que se propaga en el vacío) primero se oyera la que genera el receptor del GPS y luego de un tiempo la que viene del satélite, ya que esta señal tiene que recorrer la distancia a la que se encuentra el satélite; por lo tanto las dos señales no están sincronizadas. (China C. , 2004)

Por lo tanto tendríamos la emisión de la señal del GPS como se muestra en la Figura 8:

- **T_s** Origen de tiempos de la emisión (reloj del sistema)
- **T_u** Instante en que la señal alcanza al receptor (reloj del sistema)
- **tu** Desviación del reloj del receptor
- **$T's$** Origen de tiempos de la emisión (reloj del receptor)
- **$T'u$** instante en que la señal alcanza al receptor (reloj del receptor)

Figura 8: Emisión de la señal del GPS.



Fuente: GPS, Global Positioning System.

Entonces se debe retardar la emisión de la señal del GPS para saber cuánto se demora la señal del satélite. Así se tendría el tiempo de retardo para sincronizar los dos señales.

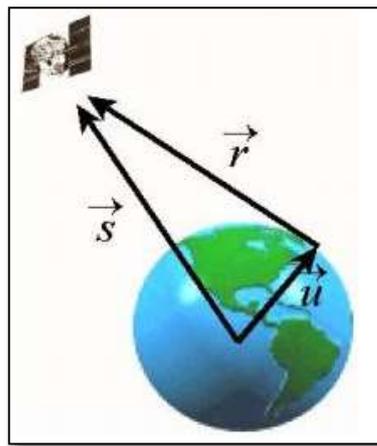
$$c * \Delta t = c * (T_u + t_u - T_s) \quad (\text{Ec. 3.5})$$

La distancia medida por el receptor se conoce como pseudoaleatorio. Las secuencias pseudoaleatorias es una sucesión de pulsos altos y bajos, es un código muy complejo como un ruido eléctrico generado al azar.

Este código ayuda a que el receptor del GPS no se sintonice con otra señal, ya que ninguna otra señal puede tener esa secuencia tan compleja. Como también ayuda a que el receptor no se confunda de satélite.

Permite amplificar las señales de GPS. Las señales débiles emitidas por los satélites pueden ser captadas por los receptores de GPS sin el uso de grandes antenas.

Figura 9: Posición del usuario, del satélite y del pseudorange.



Fuente: Curso Básico GPS.

Por lo tanto:

$$p_i = \|\vec{s}_i - \vec{u}\| + ct_u \quad (\text{Ec. 3.6})$$

Se tiene las mismas medidas en cualquier satélite

$$p_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \quad (\text{Ec. 3.7})$$

$$p_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \quad (\text{Ec. 3.8})$$

$$p_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2 + ct_u} \quad (\text{Ec. 3.9})$$

$$p_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2 + ct_u} \quad (\text{Ec.3.10})$$

Se necesita una aproximación de primer orden la cual se obtiene realizando el método de serie de Taylor.

Se debe calcular los coeficientes de la matriz H:

$$a_{xi} = \frac{x_i - \hat{x}_u}{\hat{r}_i} \quad (\text{Ec. 3.11})$$

$$a_{yi} = \frac{y_i - \hat{y}_u}{\hat{r}_i} \quad (\text{Ec. 3.12})$$

$$a_{zi} = \frac{z_i - \hat{z}_u}{\hat{r}_i} \quad (\text{Ec. 3.13})$$

Calcular las desviaciones según la posición y sincronismo:

$$\begin{aligned} \Delta \vec{r}_u &= \hat{H}^{-1} * \Delta \hat{p}_u \\ \Delta \hat{p}_u &= \hat{H} * \Delta \hat{r}_u \end{aligned} \quad (\text{Ec. 3.14})$$

A partir de esto podemos obtener la ecuación de la velocidad

$$\vec{u}(t) = \vec{u}(t_0) + \left. \frac{d\vec{u}(t)}{dt} \right|_{t_0} (t - t_0) + \frac{1}{2} \left. \frac{d^2\vec{u}(t)}{d^2t} \right|_{t_0} (t - t_0)^2 + \frac{1}{3!} \left. \frac{d^3\vec{u}(t)}{d^3t} \right|_{t_0} (t - t_0)^3 + \dots$$

$$\vec{u}(t) \approx \vec{u}(t_0) + \vec{v}(t_0)|_{t_0} (t - t_0) \quad (\text{Ec. 3.15})$$

Conociendo el funcionamiento del GPS en cuanto a la velocidad, se realizó la respectiva programación para el dispositivo Android. (Leick, 2004)

3.1.2. Sensor de temperatura del motor

El sensor que se utilizó para medir la temperatura del motor es un termistor de tipo NTC (coeficiente de temperatura negativo), este sensor varía el valor de su resistencia al cambiar la temperatura, es decir, si la temperatura del anticongelante aumenta la resistencia del sensor disminuye, y viceversa.

Este sensor obtiene la temperatura del motor a partir de la temperatura del líquido refrigerante del mismo, informando a la unidad de control para que regule la mezcla y el momento de encendido del combustible. Se encuentra situado por la conexión del tubo del agua del radiador.

Construcción y funcionamiento:

Figura 10: Sensor de temperatura del motor.



Fuente: Los autores.

El funcionamiento del termistor se basa en la teoría del electrón agujero, mientras aumenta la temperatura aumentara el número de electrones y agujeros, debido a la agitación térmica, esto hace que disminuya la resistencia. (Harper, 2000)

Para obtener los datos de éste sensor, se realizó una conexión extra desde el conector del sensor al circuito de adquisición de datos, donde la señal es recibida y procesada para mostrar los datos en el dispositivo Android, la construcción de un sensor de temperatura se muestra en la Figura 10.

3.1.3. Sensor de ángulo de giro

Como se indicó en el capítulo anterior lo ideal para obtener el ángulo de giro es mediante un acelerómetro, para esto se utilizó directamente el sensor del dispositivo Android. El cual a continuación se indicará su funcionamiento.

ACELERÓMETRO DEL DISPOSITIVO ANDROID

El acelerómetro es un dispositivo electromecánico que se encarga de medir las fuerzas de aceleración que pueden ser fuerzas estáticas o fuerzas mecánicas.

Dentro de las fuerzas estáticas tenemos la fuerza de la gravedad que es la fuerza que nos atrae hacia el centro de la tierra, y el acelerómetro funciona midiendo el ángulo de inclinación del dispositivo respecto a la tierra.

En las fuerzas dinámicas tenemos el movimiento o la vibración, el acelerómetro lo que hace es medir todo el momento si el dispositivo se mueve, puede ser en dos o tres dimensiones, así tenemos la aceleración y con este dato podemos calcular el movimiento.

Por lo tanto, el tipo de acelerómetro que se utilizó es un acelerómetro que mide fuerzas dinámicas.

Éste sensor se encarga de dar la orientación del dispositivo, basándose en tres coordenadas principales; el eje X, el eje Y y el eje Z.

Dependiendo cual sea el movimiento del acelerómetro trabajara en los dos planos respectivos. (Yáñez, 2009)

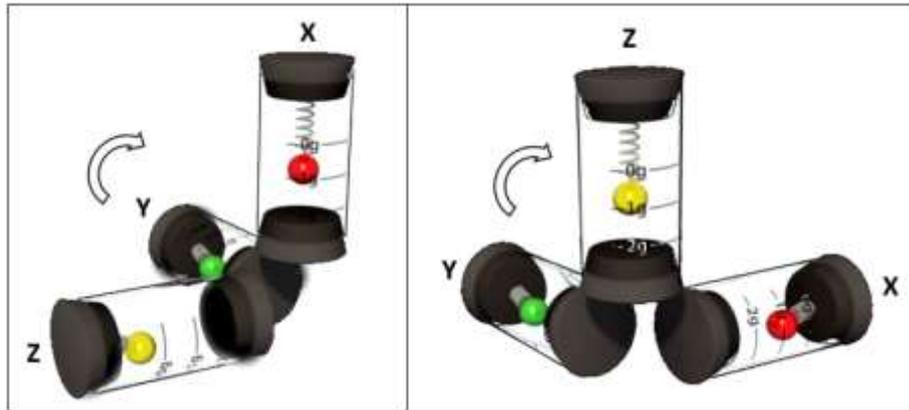
Funcionamiento

Supongamos que para el acelerómetro tenemos dos partes importantes, un tubo y una masa.

Dentro del tubo sujetamos la masa con un resorte para que este en una posición fija y si hay algún movimiento se mueva en el sentido correcto, como se muestra en la Figura 11.

El acelerómetro se le puede proyectar como un conjunto de tres tubos como el que se mostró en la figura anterior. Con esto simulamos los tres ejes de coordenadas tridimensionales, por lo tanto al girar estos ejes se sabrá la posición del dispositivo ya que la masa dentro de ellos se moverá e indicará la posición correcta.

Figura 11: Proyección de los tres ejes de coordenadas de un acelerómetro.



Fuente: Eight Amazing Engineering Stories.

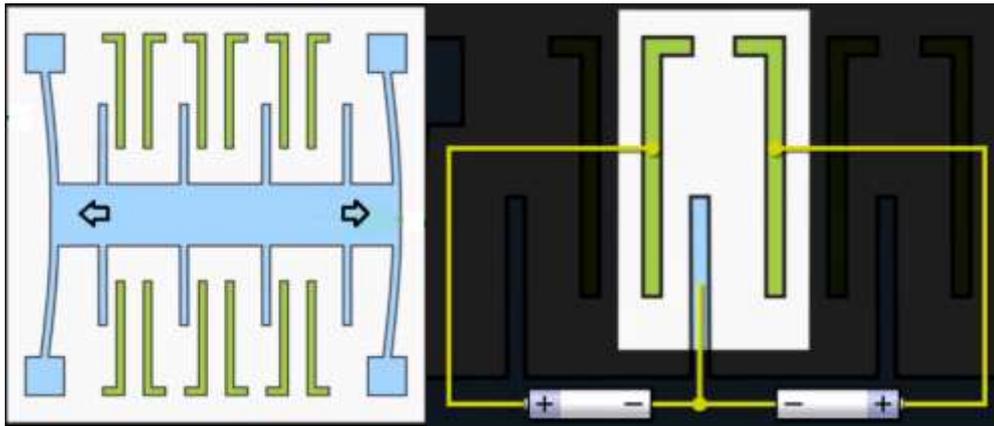
La Figura 11 muestra la proyección del movimiento con los tres ejes de coordenadas, por lo tanto se observa cómo se ubica la masa dentro de los tubos, la misma que nos da la posición del dispositivo, para esto es necesario solamente dos ejes de coordenadas.

Ésta sería una breve explicación de cómo funciona internamente el acelerómetro de un dispositivo Android.

Construcción y funcionamiento

Estos tipos de sensores están contruidos de una carcasa de silicio que contienen un tipo peine en el centro el cual se puede mover y equivale a la masa sísmica que se indicó anteriormente, como se muestra en la Figura 12.

Figura 12: Prototipo de un acelerómetro de un dispositivo Android.



Fuente: Eight Amazing Engineering Stories.

Por lo tanto este sistema funciona como un condensador diferencial⁴, si el peine del centro está en movimiento, la corriente fluirá, se hace una relación de la misma y se obtiene la aceleración.

De esta manera es como el dispositivo Android nos da los datos de ángulo de giro por medio de un acelerómetro, realizando la respectiva programación para Android se mostró los datos en el teléfono. (Hammack, Ryan, & Ziech, 2012)

3.1.4. Sensor de ángulo de inclinación

Se utilizó el mismo sensor que el punto anterior, es decir, para medir el ángulo de inclinación se usó el acelerómetro del dispositivo Android.

Su funcionamiento es el mismo, lo que se diferencia del ángulo de giro es que utiliza diferentes coordenadas, lo cual se modifica con la programación.

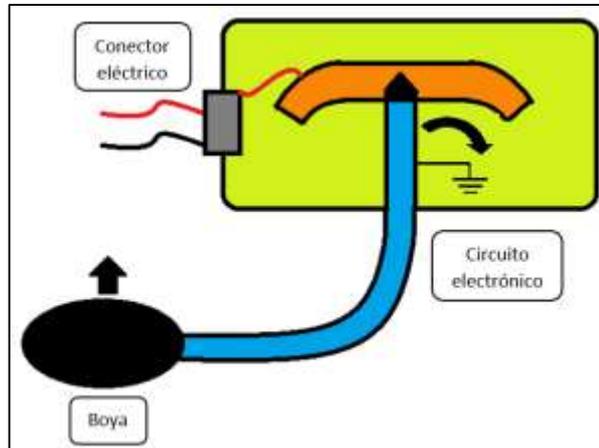
3.1.5. Sensor de gasolina

El sensor que se utilizó para medir el nivel de gasolina de la moto está ubicado dentro del tanque de gasolina, este sensor está construido por una boya que flota en

⁴ Condensador diferencial: formado por dos condensadores variables dispuestos físicamente que experimentan el mismo cambio pero en sentido opuesto. La salida es lineal.

el momento que el tanque esta con gasolina, esta boya tiene una varilla en su extremo, la cual va sujeta a un circuito electrónico.

Figura 13: Esquema de un sensor de gasolina.



Fuente: Los autores.

La boya flota hacia arriba o hacia abajo dependiendo de la cantidad de la gasolina que este en el tanque, es decir, si hay un nivel alto de gasolina la boya se encontrara en la parte superior o viceversa. Como la boya contiene una varilla en su extremo, el momento de flotar hace que esta se mueva de derecha a izquierda dependiendo el sentido en el que flote, como se muestra en la Figura 13.

Esta varilla está sujeta a un circuito electrónico que simplemente es una resistencia variable o un potenciómetro. Según el movimiento de la varilla esta marca la resistencia.

Cuando el nivel de gasolina sube la boya se eleva, moviendo la varilla a la que va unida y esto hace que la resistencia aumente ya que el indicador se movería a su derecha. Si el nivel de gasolina disminuye, la varilla se moverá en sentido contrario y la resistencia será menor.

Esta es la señal que envía este sensor, por lo tanto se hace una conexión del conector al circuito electrónico de adquisición de datos, donde se recibe esa señal, se procesa y es enviada al dispositivo Android para que los datos sean mostrados.

3.1.6. Sensor de presión de los neumáticos

El sensor que se utilizó es un sensor TPMS M202L directo que va ubicado entre la llanta y el aro de la moto, como se muestra en la Figura 14; como se explicó en el capítulo anterior, este sensor es más confiable, nos da un dato en tiempo real sin error; también se eligió este sensor debido que tiene un tamaño más grande que el de tipo tuerca, así fue más fácil su manipulación para el proyecto.

Los sensores para motocicleta son muy escasos, muchos países no tienen a la venta este tipo de sensores, mientras que varios concesionarios ya disponen de sensores para vehículos, pero éstos sensores que van entre el aro y la llanta de los vehículos no pueden ser usados en motocicletas, debido que la válvula de una moto está ubicada en el centro del aro, perpendicular al mismo, por lo que sensores para motos vienen completamente rectos. Mientras que la válvula de un vehículo está ubicado al lado del aro por lo tanto el sensor es diagonal.

Figura 14: Sensor TPMS M202L para moto.



Fuente: Orange Electronics.

El sensor se debe colocar correctamente para no tener errores en los datos recibidos, se coloca dónde va la válvula de aire y se debe fijar bien con la tuerca que viene incluida.

Este sensor incluye una batería interna que dura aproximadamente de 3 a 5 años dependiendo del uso del mismo. Se comunica inalámbricamente por antenas RF (radio frecuencia), con un módulo que contiene un display para indicar los datos.

En el módulo va conectado a la batería de la moto, y se puede configurar los rangos de los valores de presión que estén correctos, así cuando los valores estén fuera del rango ya sean menores o mayores se alerta al piloto mostrando una pantalla roja.

El sensor no permanece prendido todo el tiempo, éste se activa el momento que el modulo se enciende. (Nabipoor & Maijlis, 2006)

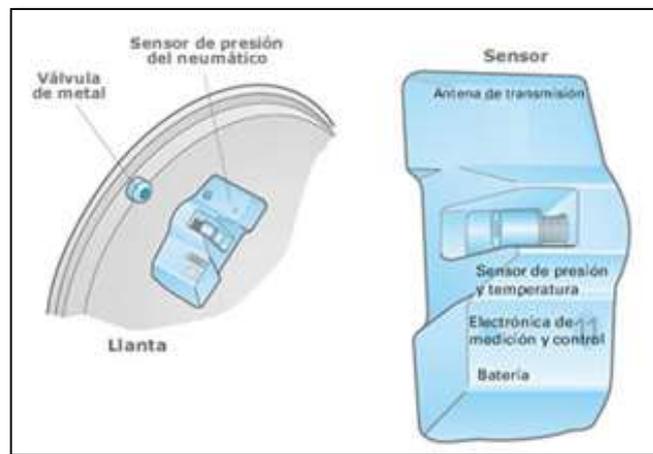
Construcción:

Los módulos que van ubicados en los neumáticos tienen incluido en su interior:

- Antena transmisora por radio frecuencia, 314.85MHz.
- Batería de 3v.
- Microcontrolador.
- Sensor de presión, 50psi a 350kpsi.
- Circuito de electrónica y control.
- Válvula de 11.3mm.

Como se muestra en la siguiente figura:

Figura 15: Esquema del sensor de presión de los neumáticos.



Fuente: Control automático de presión de neumáticos.

El sensor tiene internamente un circuito electrónico de medición y control, el cual contiene un microcontrolador, una antena de radio frecuencia RF y una batería.

La señal es tomada por el sensor, y enviada al microcontrolador, el cual se encarga de procesarla y obtener los datos de la presión. Estos datos son transmitidos al módulo mediante una antena de radio frecuencia la misma que trabaja a una frecuencia de 315Mhz. (Hammack, Ryan, & Ziech, 2012)

El modulo está construido por un circuito electrónico que contiene un microcontrolador y un *display*. El microcontrolador se encarga de recibir los datos y mostrarlos en la pantalla. Como también permite realizar los cambios que el usuario necesita, como ingresar los rangos de presión, cambiar las unidades de medida, etc.

3.2. Ubicación de los sensores en la moto

Los sensores descritos anteriormente son los que se analizaron y se usaron en el proyecto.

Como se indicó la motocicleta fue construida desde cero, para la construcción del bastidor de la moto se tomó en cuenta las medidas de cada pieza que iba a ir en el así como las medidas de cada sensor para que queden acoplados correctamente en la moto de manera sencilla y segura.

Todos los sensores tienen sus especificaciones y características, por lo que esto fue lo primero que se tomó en cuenta para buscar la mejor ubicación. Se hizo un estudio y análisis completo de cada uno, viendo sus medidas, su peso, en el caso del sensor de temperatura del motor se analizó los rangos de temperatura que aguanta; en el sensor de gasolina se analizó el nivel al que alcanza y los sensores de presión se analizó todas las medidas, para ver si el sensor entraba por el hueco de la válvula y si no rozaba con la llanta, se tomó el peso para ver si no afectaba en un futuro en el momento que la moto rodaba.

3.2.1. Diseño de la ubicación de los sensores

El diseño se realizó en el *software* SolidWorks, en el cual se fue construyendo cada pieza de la motocicleta con sus medidas reales, dejando los espacios apropiados para los sensores, con medidas exactas.

Se realizaron diferentes pruebas y los sensores fueron acoplados y probados en cada una de ellas hasta que se obtuvo la posición correcta donde se colocaron los mismos.

Figura 16: Prototipo de la moto.



Fuente: Los autores.

Se realizó pruebas con diferentes tipos de sensores para la medición de cada parámetro; los sensores no debían afligir en la moto, no debían afectar su peso, su estabilidad, el espacio, y la comodidad para el piloto. Por lo que se tomó en cuenta estas medidas para la elección de cada uno de ellos.

La figura 16 muestra donde va ubicado cada sensor en la moto.

Donde:

1. Sensor de presión del neumático trasero.
2. Sensor de presión del neumático delantero.
3. Sensor de temperatura del motor.
4. Sensor de gasolina.
5. Dispositivo Android
 - a. Sensor de ángulo de giro
 - b. Sensor de inclinación
 - c. Velocidad
 - d. RPM

CAPÍTULO 4

DISEÑO ELECTRÓNICO E IMPLEMENTACIÓN DEL *HARDWARE*.

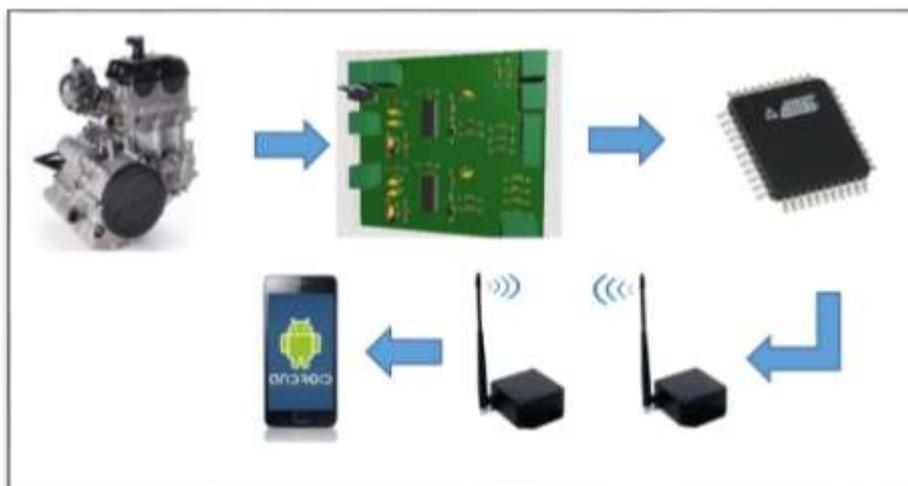
El siguiente capítulo abarcará los temas de la parte electrónica que se implementó en la moto. El diseño electrónico es una de las partes principales del sistema, este contiene chips y microcontroladores programables, adquisición de datos de los sensores y circuitos de comunicación.

El sistema de control y adquisición de datos consiste en adquirir las diferentes señales obtenidas de nuestro prototipo y presentarlas de manera gráfica en un dispositivo móvil como se muestra en la Figura 17.

Hay dos tipos de datos que se mostraran en dicho dispositivo, las que se toman directamente de la moto y las programadas por *software*.

Entre las señales adquiridas de la moto tenemos las revoluciones por minuto del motor, temperatura del motor y el nivel de gasolina, estas son procesadas por un circuito específicamente diseñado para la manipulación de las mismas, después de esto, los datos son adquiridos por medio de un microcontrolador, el cual recibe la información, la procesa y la envía lista para ser presentada por el dispositivo móvil; la comunicación del dispositivo móvil con el microcontrolador es inalámbricamente.

Figura 17: Esquema del sistema de control y adquisición de datos.



Fuente: Los autores.

A continuación se explicara el funcionamiento y construcción de cada uno de los componentes que forman este sistema de adquisición de datos para una moto de velocidad.

4.1. Circuito electrónico

4.1.1. Conversor de frecuencia a voltaje para cálculo de las revoluciones por minuto RPM

Este circuito se encarga de convertir frecuencia a voltaje, es decir se tiene una serie de pulsos y se convierte en una señal analógica; es muy utilizado para la medición de la velocidad de motores.

Se construyó con el fin de obtener la velocidad del motor y las revoluciones por minuto y transmitirla al microcontrolador; para esto el conversor nos da una salida análoga de 5 voltios que es el valor que permite un microcontrolador.

Para el circuito del conversor solo se necesitó una configuración con resistencias y capacitores (configuración RC), para que se genere el voltaje que se necesita.

Para el cálculo de las resistencias se basó en la siguiente ecuación:

$$v_{out} = v_{cc} * f_{in} * R1 * C1 * k \quad (\text{Ec. 4.1})$$

Donde:

- v_{out} : Voltaje de salida
- v_{cc} : Tensión de alimentación
- f_{in} : Frecuencia de la señal que ingresa al conversor
- $R1$: Resistencia
- $C1$: Capacitor
- k : Ganancia

Para este cálculo se considera un valor de capacitancia comercial, el valor de C1 será de 22nf que es una capacitancia disponible en el mercado, puesto que en la hoja de datos indica que debe ser mayor que 500pf. También se debe tomar en cuenta que el valor de la resistencia R1 debe ser alto para que la corriente sea mínima.

La velocidad de giro del motor se mide en revoluciones por minuto; para conocer la frecuencia de la señal pulsante se debe calcular con la siguiente formula:

$$f = \frac{rpm}{60} * 2 \quad (\text{Ec. 4.2})$$

Se divide el valor de las rpm para 60 para saber a cuantas revoluciones gira el motor en un segundo y se multiplica por dos ya que la bobina de encendido en la moto se activa cada dos veces por cada revolución del motor. Así podemos tener la siguiente tabla:

Tabla 3: Valores calculados de Frecuencia para diferentes RPM.

RPM	Frecuencia
0	0
300	10
600	20
900	30
1200	40
1500	50
1800	60
2100	70
2400	80
2700	90
3000	100
3300	110
3600	120
3900	130
4200	140
4500	150
4800	160
5100	170
5400	180
5700	190

6000	200
6300	210
6600	220
6900	230
7200	240
7500	250

Fuente: Los autores.

Realizamos esta operación para un rango de rpm de 0 a 7500, llegamos a este valor ya que la salida máxima es de 5v a 7500 rpm, por lo que la frecuencia y la R1 varia.

Por lo tanto los cálculos son los siguientes, conociendo que a una velocidad de 180km/h el convertor entrega un valor de voltaje máximo de salida, se tomó un valor de 4v para no estar cercar del límite superior y tener problemas de saturación:

$$v_{out} = v_{cc} * f_{in} * R1 * C1 * k \quad (\text{Ec. 4.3})$$

$$R1 = \frac{V_{out}}{V_{cc} * f_{in} * C1 * k}$$

$$R1 = \frac{4}{5 * 250 * 22nf * 1}$$

$$R1 = 145.45 \text{ k}\Omega$$

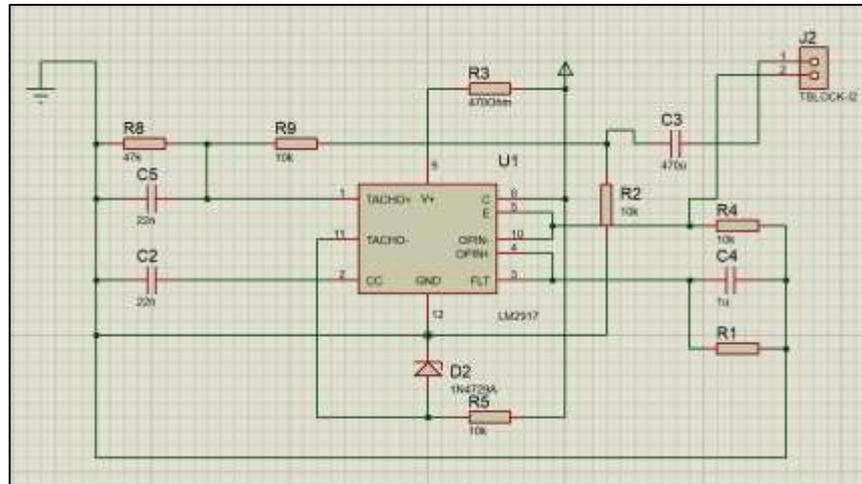
$$R1 \text{ comercial} = 150 \text{ K}\Omega$$

También se debe considerar que el convertor debe estar regulado por un diodo zener, cuando se necesita que la salida de este no dependa de la tensión de alimentación; por lo tanto para elegir la resistencia de descarga se debe considerar que el amplificador operacional únicamente necesita 3mA que es la corriente que se obtendrá por el diodo zener. Se ha optado por una resistencia de 470 Ω para reducir al mínimo la variación de voltaje en el diodo zener.

En este circuito se utilizó el componente electrónico LM2917 como se muestra en la Figura 18, que es el convertor de frecuencia a voltaje, su circuito interno está construido con un comparador de tensión en la entrada, una bomba de carga como

convertidor frecuencia a voltaje y un amplificador operacional con un transistor de salida.

Figura 18: Esquema del conversor para el cálculo de las RPM.



Fuente: Los autores.

Conociendo los materiales necesarios con sus respectivos valores se realizó la construcción del circuito, fue diseñado en el *software* Isis Proteus y simulado en Ares Proteus; teniendo una respuesta correcta con la simulación se realizó su montaje.

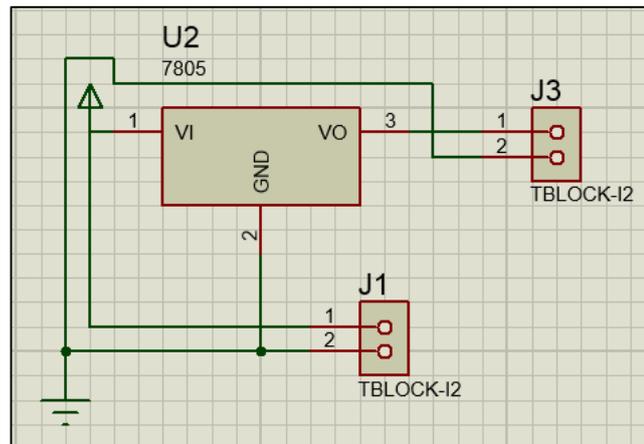
4.1.2. Regulador de tensión

Es un componente electrónico que está diseñado para tener un valor de voltaje constante, tiene tres pines que son para el voltaje de entrada, tierra y voltaje de salida.

Este dispositivo se utilizó para la alimentación del circuito de conversión de frecuencia a voltaje para el cálculo de las revoluciones por minuto, y para alimentar al Arduino, ya que los dos se alimentan con 5v.

El regulador utilizado es un 7805 que es un regulador positivo, como se muestra en la Figura 19; nos entrega 5v de corriente continua a la salida, y la tensión de alimentación debe ser menor a 35v y mayor 2v más que la de salida.

Figura 19: Regulador de voltaje.



Fuente: Los autores.

Se tiene la conexión de alimentación al circuito que son los 12v de la batería de la moto, pasa por el regulador de tensión y se tiene a la salida 5v de corriente continua, los cuales son óptimos para el conversor LM2917.

4.1.3. Divisor de tensión

Se utilizó divisores de tensión resistivos, los cuales se encarga de repartir la tensión por medio de un arreglo de resistencias.

Se utilizaron estos circuitos para la conexión de los sensores de gasolina, temperatura del motor con el microcontrolador, y para la conexión del módulo Bluetooth.

Los sensores de temperatura del motor y de gasolina se alimentan a 12v que es el voltaje de la batería de la moto, y se necesita 5v para la conexión con el microcontrolador, por lo tanto los circuitos dividen la tensión de entrada y da la tensión de salida requerida. Para el cálculo de las resistencias se manejó la siguiente fórmula, se utilizó una resistencia comercial de 1k para el cálculo de la segunda:

$$V_{out} = \frac{R2}{R1 + R2} * V_{in} \quad (\text{Ec. 4.4})$$

$$R1 = \left(\frac{V_{in}}{V_{out}} * R2 \right) - R2$$

$$R1 = \left(\frac{12}{5} * 1K \right) - 1K$$

$$R1 = 1.5k$$

Estas resistencias se utilizaron en los dos circuitos de divisores de tensión, para el sensor de gasolina y para el sensor de temperatura del motor.

El modulo Bluetooth se alimenta con 3.3v, por la tanto se usó otro divisor de tensión para bajar el voltaje de 5v a 3.3v ya que se alimentó desde el microcontrolador. Igual que el cálculo anterior se escogió una resistencia comercial de 1k para el cálculo de la segunda resistencia.

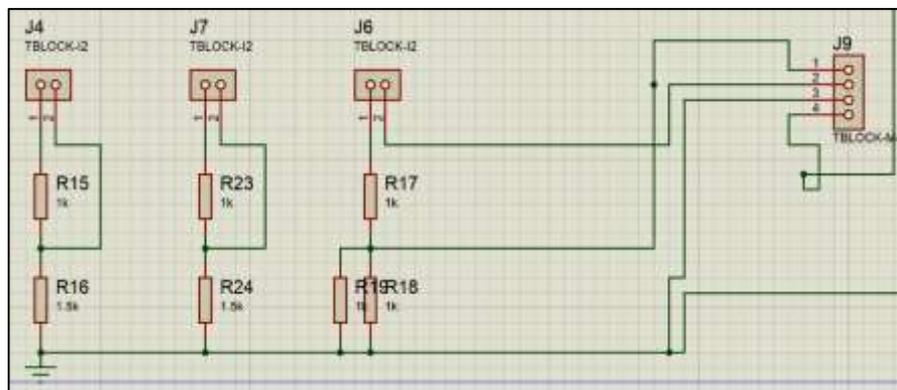
$$R1 = \left(\frac{V_{in}}{V_{out}} * R2 \right) - R2 \quad (\text{Ec. 4.5})$$

$$R1 = \left(\frac{5}{3.3} * 1K \right) - 1K$$

$$R1 = 515\Omega$$

$$R1 \text{ comercial} = 500\Omega$$

Figura 20: Divisores de tensión.



Fuente: Los autores.

En la Figura 20 se ilustra los tres divisores de tensión que se utilizó para el circuito final.

4.2. Adquisición de la señal de los sensores de presión de los neumáticos

Módulo TPMS M202L

Como se explicó en el capítulo anterior, el sistema de los sensores de presión de los neumáticos viene con tres módulos, dos módulos donde se encuentran los sensores de presión del neumático delantero y del neumático trasero que van ubicados en las llantas de la moto y un módulo donde se muestran los datos en un *display*.

Este sistema se comunica de manera inalámbrica, por radio frecuencia.

El kit TPMS consta de dos módulos: un receptor MPXY8300 Freescale ESTAR y un transmisor LF 125KHz. También contiene su software asociado.

Los módulos que van ubicados en los neumáticos tienen incluido en su interior:

- Antena transmisora por radio frecuencia, 314.85MHz.
- Batería de 3v.
- Microcontrolador.
- Sensor de presión, 50psi a 350kpsi.
- Circuito de electrónica y control.

4.2.1. Características claves del módulo TPMS

Figura 21: Módulo del sensor de presión que va ubicado en los neumáticos.



Fuente: MPXY8300 Design Reference Manual.

Las características claves de éste módulo son:

- Microcontrolador de 8-bit
- S08 Core con SIM
- Memoria RAM 512
- 8K FLASH (además incluye *firmware* de fábrica y ajuste de datos)
- 32-byte, de baja potencia y registro de parámetros
- 4 pines GPIO⁵ con opción de *pull-ups/pull-downs* y con interrupción
- Tiempo real de interrupción por LFO⁶ con intervalos de interrupción de 8, 16, 32, 64, 128, 256, 512 o 1024 ms
- Temporizador y reposición periódica impulsado por LFO
- Tiempo de espera con Fuentes de reloj y tiempos seleccionables.
- 2 canales de propósito general, temporizador/PWM
- Oscilador interno y configuración MPXY8300RM, *Freescale Semiconductor*
7
- Reloj del bus MCU de 0.5, 1, 2 y 4 MHz (1, 2, 4 y 8 MHz HFO)
- Reloj de carga de 10MHZ
- Baja frecuencia (LFO) con periodo de 1ms
- Frecuencia media, decodificador LFR y sensor del reloj (MFO) con periodo de 8s
- Bajo voltaje de detección
- Reinicio de temperatura normal en el hardware (sobre la temperatura detectada por el software)
- Entrada diferencial LF (*low frequency*), detector/decodificador
- Sensor de temperatura con interfaz de señal para ADC10
- Sensor de presión con interfaz de señal para ADC10
- Acelerómetro con coordenadas Z y X con interfaz de señal para ADC10
- Voltaje de referencia medido por ADC10
- Transmisión interna RF de 315/434MHz
 - Oscilador externo del cristal

⁵ GPIO: *general purpose Input/Output*, Entrada/Salida de propósito general. Chip cuyo comportamiento se puede controlar por el usuario en tiempo de ejecución.

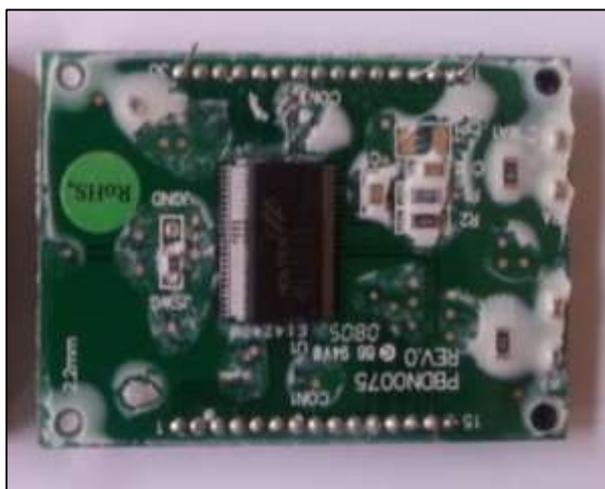
⁶ LFO: *Low Frequency Oscillation*, baja frecuencia de oscilación.

- Capacidad de modulación ASK y FSK
- Generador de velocidad de datos programables
- Codificación de datos Manchester o bi-fase
- Buffer de datos RF de 128 bits con RTS/CTS
- Acceso directo para transmisión RF desde el microcontrolador a únicos formatos
- Tensión de alimentación de carga para compensar las baterías frías con tiempos de carga seleccionables de 30, 60, 120 y 240ms
 - Capacidad de programación del MCU de 8 bit a través del conector de 4 pines y multilink BDM
 - Circuito de red de adaptación flexible que permite utilizar la antena externa
 - Alcance inalámbrico de 15m en el aire con la antena externa
 - Batería de 3v

Mientras que el módulo donde se muestran los datos tiene incluido dos placas de circuitos electrónicos, una que se encarga de recibir los datos que envían los sensores y otra que se encarga de mostrar los datos en un *display*.

Como se muestra en la siguiente figura:

Figura 22: Módulo del sensor de presión de los neumáticos que contiene el display.



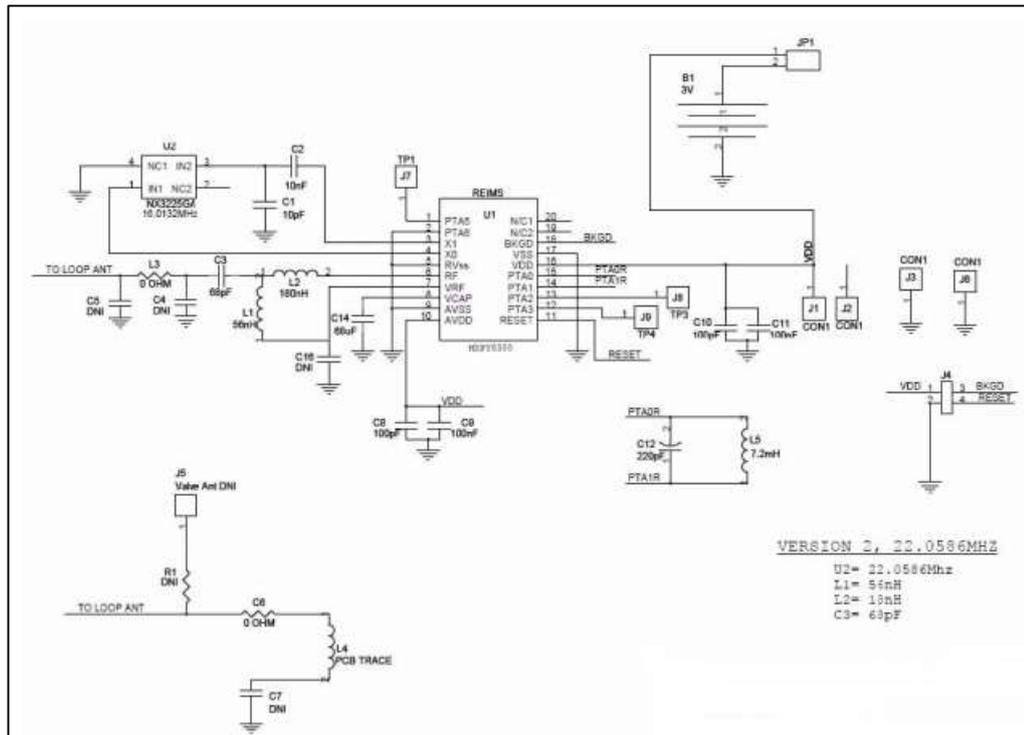
Fuente: Los autores.

Las características claves de éste módulo son:

- Comunicación *wireless* de 315 a 434 MHz
- Alcance *wireless* de 20m
- Velocidad de datos de 9.6kb/s
- Un pulsante para la programación del microcontrolador ya que este contiene un gestor de arranque.
- Capacidad de programación del MCU de 8 bit a través del conector de 4 pines y multilink BDM
- Flexibilidad del formato de datos obtenido por la programación del MCU
- Alcance de operación de hasta 0.5m
- Señal de modulación LF a través de MC33690 *Freescale's* usando un MCU estándar de TPM2CH0

El microcontrolador tiene incluido las antenas de radio frecuencia, este es de mucha ayuda para que el módulo tenga el tamaño ideal para ser colocado en los neumáticos, como podemos observar a continuación, los circuitos de los módulos no son pequeños, pero los componentes que conforman el mismo si lo son, por lo que no tienen ningún inconveniente al ser ubicados en la llanta de la moto. (Lenzen, 2012)

Figura 23: Esquema del circuito electrónico de los módulos del sensor de presión de los neumáticos.



Fuente: MPXY8300 Design Reference Manual.

4.2.1.1. Información específica del TPMS, pruebas y procedimiento general

Interferencia por la línea eléctrica

Este procedimiento fue analizado a través de Estándar ANSI, se encontraron dos líneas, utilizando 50uH. El ancho de banda del espectro analizado fue de 10KHz con una velocidad de barrido apropiada.

Radiación de interferencia

Este procedimiento fue analizado a través de Estándar ANSI, al igual que el anterior el ancho de banda del espectro analizado fue de 10KHz con una velocidad de barrido apropiada, el analizador ha sido calibrado en dB por encima de un micro voltio en la salida de la antena, el ancho de banda de resolución es de 100KHz.

Fórmula de los factores de conversión

Se estableció la intensidad de campo a 3m, añadiendo la medida del analizador de espectro (mide hasta en unidades de dBmV) más el factor de corrección de la antena dado por el fabricante de la antena, esto se expresa en dB, con una frecuencia de 33MHz:

$$FS = \text{Lectura del analizador} + \text{factor dado por el fabricante} + \text{perdida en el cable}$$

(Ec. 4.6)

$$FS = 20dBuV + 10.36 \frac{dB}{m} + 6db$$

$$FS = 36.36 \frac{dBuV}{m} \text{ a } 3m$$

Transmisor RF:

Consiste en un microcontrolador con una salida para una antena RF y un buffer de datos de *hardware* que sirve para un control directo desde el microcontrolador. Cuenta con una bomba de carga para una tensión de alimentación mayor para la transmisión RF, cuando el voltaje de la batería es demasiado bajo.

Tiene una memoria con 32 registros, los mismos que contienen bits de control y de estado, que sirven para los datos de buffer, variables, y registro de prueba; para el acceso a estos bits es a través de una interfaz interna SPI.

Límite de tiempo de transmisión:

Los dispositivos podrán operar bajo disposiciones del fabricante, su operación está limitada para que sea automática, de manera que la duración de cada transmisión no sea superior a un segundo y el periodo de silencio entre transmisiones sea al menos de 30 veces la duración de transmisión, pero no menos de 10 segundos, todo este proceso es controlado por software.

El resultado de la medición real de la duración de cada transmisión es 16,432 ms, y el resultado para el periodo de silencio entre las transmisiones es 30.060 segundos.

Ancho de banda

El ancho de banda de la emisión no será más que el 0.25% que el ancho de la frecuencia central para los dispositivos operando por encima de 70 MHz y por debajo de 900 MHz. Para los dispositivos que funcionan por encima de 900 MHz, la emisión no será más que el 0.5% de la frecuencia central. El ancho de banda se determina en los 20 dB de la portadora modulada.

Tabla 4: Ancho de banda M202L.

Frecuencia	Ancho de banda	Limite
315 MHz	208.416 kHz	0.7875 MHz

Fuente: Los autores.

Receptor RF:

Receptor LF, éste bloque permite activar el microcontrolador MPXY8300 con un estado de baja potencia, stop, cuando el datagrama *carrier* o Manchester es detectado. Si este tiene una codificación Manchester, puede decodificarse por la máquina de estados LF, pero asumiendo que el módulo Lf está configurado correctamente.

Si no se tiene una codificación Manchester, el microcontrolador necesita primero ser activado por un *carrier (mode detection)*, y luego cuando el microcontrolador este en modo directo ya podrá decodificar el telegrama.

Frecuencia de muestreo LF

Para reducir la energía de consumo, hay un periodo programable para la activación del mecanismo por modo Manchester a una frecuencia del oscilador de un 1KHz.

La frecuencia baja del oscilador permite que el módulo este en un estado bajo y se active periódicamente cada 32.768s.

Para el adecuado muestro LF, se debe tomar en cuenta la exactitud del tiempo de muestreo relacionado directamente con la frecuencia de 1KHz LFO. (Orange Electronic Co. Ltd., 2013)

Datos analizados y pruebas del fabricante

Tabla 5: Datos generales del TPMS M202L.

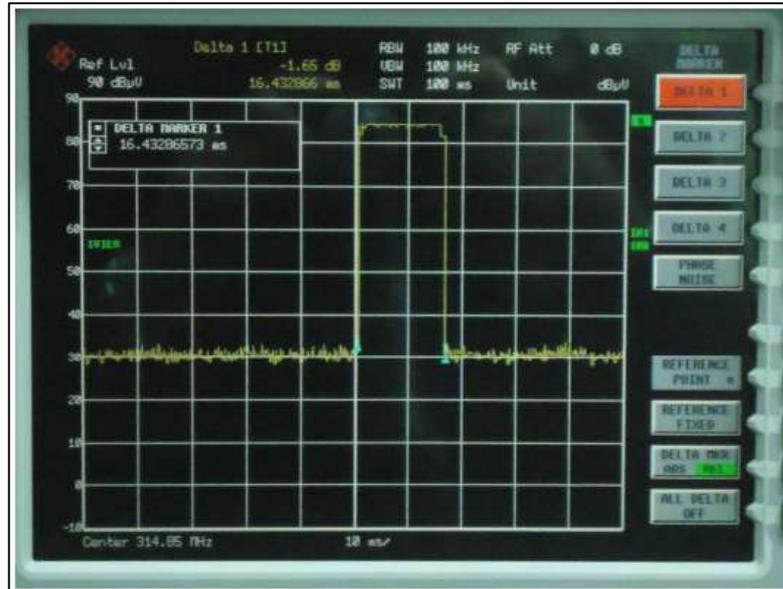
Model:	M202	Date:	2012/3/5					Engineer:	Kevin	
Mode:		Temperature:	24°C							
Polarization	Horizontal	Humidity:	60%							
Frequency (MHz)	Reading (dBuV) Peak	Factor (dB)		Result @3m (dBuV/m)		Limit @3m (dBuV/m)		Margin (dB)	Table Degree (Deg.)	Ant. High (cm)
		Corr.	Duty	Peak	Ave.	Peak	Ave.			
314.9003	54.24	15.96	-15.69	70.20	54.51	87.66	67.66	-12.85	250	100
Polarization: Vertical										
Frequency (MHz)	Reading (dBuV) Peak	Factor (dB)		Result @3m (dBuV/m)		Limit @3m (dBuV/m)		Margin (dB)	Table Degree (Deg.)	Ant. High (cm)
		Corr.	Duty	Peak	Ave.	Peak	Ave.			
314.7800	42.06	15.95	-15.69	58.01	42.32	87.66	67.66	-25.34	130	100

Fuente: M202L Datasheet.

Diagramas de medidas:

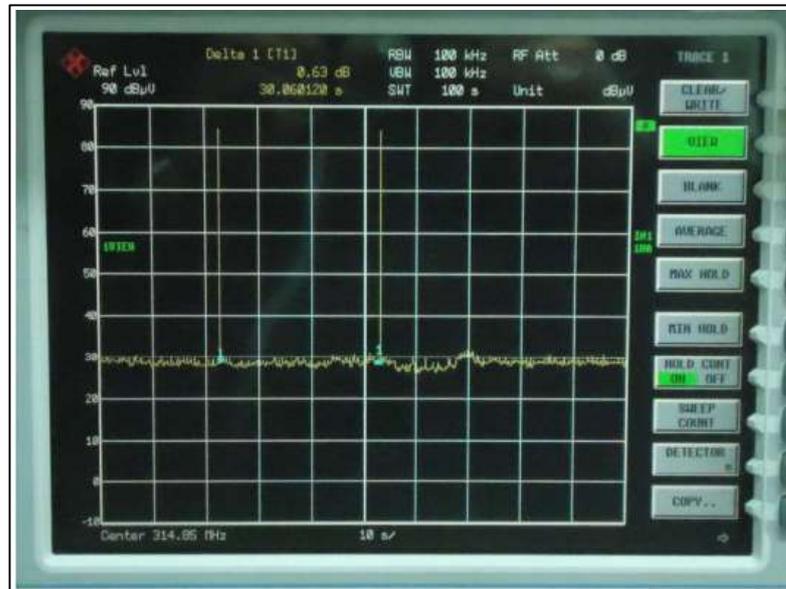
- a) Tiempo de activación
- b) Potencia de salida
- c) Ancho de banda

Figura 24: Tiempo de activación del TPMS 16.432ms.



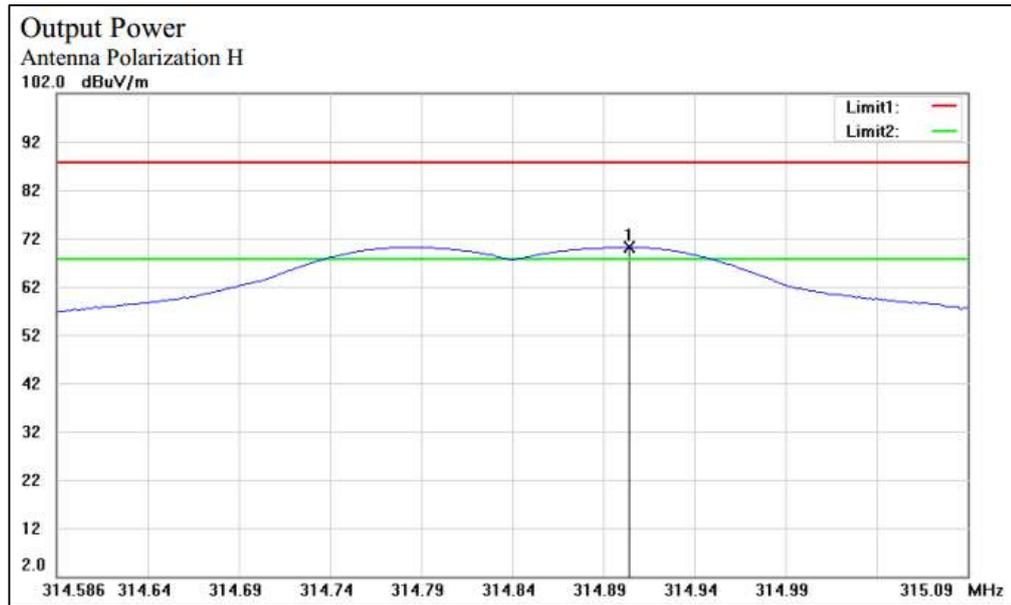
Fuente: FCC PART 15 SUBPART C TEST REPORT for Tire Pressure Monitoring System.

Figura 25: Tiempo de activación del TPMS 30.061s.



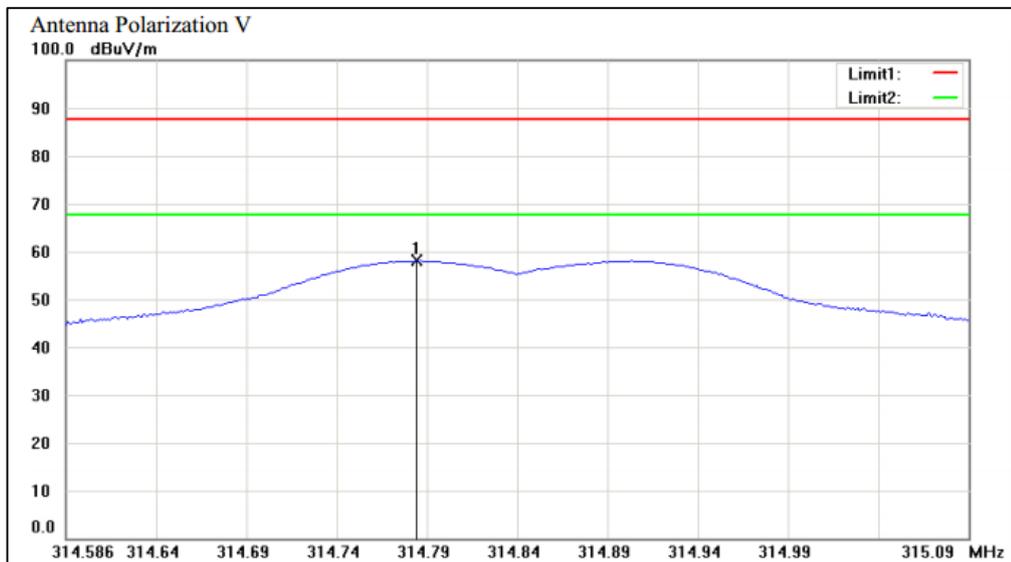
Fuente: FCC PART 15 SUBPART C TEST REPORT for Tire Pressure Monitoring System.

Figura 26: Potencia de salida, con antena en polarización horizontal.



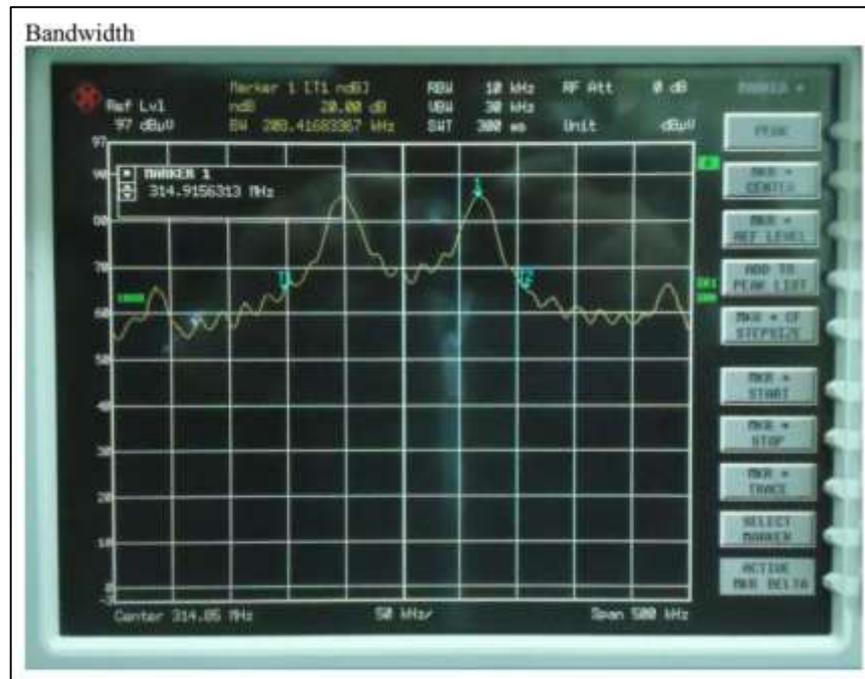
Fuente: FCC PART 15 SUBPART C TEST REPORT for Tire Pressure Monitoring System.

Figura 27: Potencia de salida, con antena en polarización vertical.



Fuente: FCC PART 15 SUBPART C TEST REPORT for Tire Pressure Monitoring System.

Figura 28: Ancho de banda de la señal del TPMS



Fuente: FCC PART 15 SUBPART C TEST REPORT for Tire Pressure Monitoring System.

4.2.2. Adquisición de la señal

La señal de los sensores vienen encriptados, por esta razón se optó por abrir el modulo que contiene el *display*.

Una de las placas de este módulo tiene un microcontrolador que es el que se encargaba de recibir la señal y transmitirla a la siguiente placa para que puedan ser mostrados los datos en el *display*.

Los módulos de los sensores no funcionan si el modulo que contiene la pantalla no está alimentado.

Se hizo la prueba con un solo sensor y el módulo del *display* prendido; con un USRP⁷ (*Universal Software Radio Pheripetral*) se obtuvo la señal que enviaba el sensor al módulo, se pudo detectar que la señal se enviaba cada 45 segundos.

⁷ USRP: es un sistema periférico universal de radio. Se encarga de procesar integralmente toda forma de onda, así se tendrá la modulación y demodulación de la misma.

Era complicado analizar la señal directo en la computadora lo que daba el USRP ya que la señal se observaba como un solo pulso. Por esa razón se optó utilizar el *Software* LabView para detectar la señal y analizarla.

En Labview pudimos observar que la señal era demasiado pequeña, alrededor de los micro voltios, por lo que fue difícil obtener los datos de la señal.

Mientras que en la placa del módulo del *display* era complejo obtener los datos, ya que la señal llegaba encriptado, por lo que para tener los datos se necesitaba realizar un procedimiento completo que era:

- Aparejarse con el módulo que envía la señal.
- Receptar la señal.
- Decodificar la señal.
- Desencriptar la señal para recibir los datos.

Este procedimiento era demasiado largo y complejo por lo que se optó por construir un módulo con un sensor para obtener la presión de los neumáticos.

4.2.3. Construcción de un módulo para obtener la presión de los neumáticos.

Se hizo un análisis para la construcción de un módulo que vaya ubicado en los neumáticos, este módulo se comunicaba inalámbricamente con el dispositivo Android.

Para la construcción se hizo un análisis de los componentes necesarios que se utilizarían para el mismo, que son:

- Sensor de presión
- Módulo Bluetooth / Antenas RF
- Batería
- Microcontrolador
- Circuito electrónico

Es necesario que el módulo tenga dimensiones pequeñas y que no pese más de 30 gramos, para que no desestabilice la llanta y la moto quede estable y no tenga

problemas en el momento de andar y más que todo por la seguridad del piloto. Por lo tanto los componentes que van dentro del módulo deben ser muy pequeños.

Se hizo un análisis de todos los componentes necesarios para la construcción y se buscó en el mercado tratando de ver los elementos más pequeños y livianos.

Para el circuito electrónico se encontró componentes pequeños, un microcontrolador que tiene las características suficientes para la construcción del prototipo y una batería pequeña. El módulo bluetooth o las antenas RF que hay en el mercado no son pequeñas para ser colocadas en el módulo, lo cual fue un problema para la adaptación del módulo en la llanta.

El problema más grande fue encontrar un sensor pequeño y con las características necesarias para obtener la señal de la presión del neumático, en el mercado no existen estos tipos de sensores pequeños que midan rangos de 50psi a 350kpsi, existen sensores que necesitan conexión eléctrica o sensores inalámbricos industriales, que son demasiado grandes para colocar en este módulo.

De los vehículos y motocicletas que ya vienen con estos sensores cada casa automotriz o cada fábrica que construye estos sensores, tienen su propio microcontrolador y circuito electrónico, de manera que las medidas y peso sean justos para ser acoplado en sus vehículos. Por lo tanto no se pudo construir este dispositivo.

4.3. Microcontrolador, implementación del *hardware*

4.3.1. Arduino Mega 2560

Pines utilizados del Arduino Mega 2560 y funcionamiento:

Las señales de revoluciones por minuto del motor, del sensor de gasolina y del sensor de temperatura del motor, son señales analógicas, para tomar éstas señales, se hizo uso de los pines de entradas analógicas del arduino, los pines utilizados fueron:

- Pin A0: Señal de RPM
- Pin A3: Señal del sensor de gasolina
- Pin A5: Señal del sensor de temperatura del motor

Estos pines son activados por medio de la programación para que se reciba esta señal, sea procesada y transformada en los datos que se quiere mostrar.

Luego que la señal de revoluciones por minuto del motor fue procesada se hizo una conexión de leds con pines de salidas digitales del arduino, estos leds indican los niveles de RPM, cuando esta máximo, medio y bajo; los pines son:

- Pin 8: Nivel alto de RPM
- Pin 6: Nivel medio de RPM
- Pin 4: Nivel bajo de RPM

Estas salidas digitales fueron un complemento para guiar al piloto en el momento que está conduciendo, ya que no puede estar pendiente todo el tiempo del dispositivo Android.

También se utilizaron los pines del sistema de comunicación USART, esta conexión se utilizó para la comunicación del módulo Bluetooth con el Arduino, se usó el Serial 1 que incluye los pines:

- Pin 18: Transmisión
- Pin 19: Recepción

El arduino se alimenta con 5v de corriente continua desde el regulador de voltaje que se explicó anteriormente. (Margolis, 2012)

4.4. Módulo Bluetooth, implementación del *hardware*

Hoy en día se utiliza la comunicación inalámbrica “*wireless*” que es una forma sencilla y eficaz de conectarse con otros dispositivos, fácil de encontrar en el mercado y económica; la comunicación Bluetooth es parte de esto, donde el enlace de datos es sin ningún vínculo físico, pero es confiable y seguro.

Existen varios tipos de módulos bluetooth, parecidos todos, lo que difiere son las conexiones que cambian de un modelo a otro. El modulo bluetooth que se utilizó para la comunicación con el dispositivo Android es el HC-06.

4.4.1. Módulo Bluetooth HC-06

Éste módulo se alimenta con 3.3v y tiene un bajo consumo de 8mA en transmisión y recepción activa. Se puede escoger la velocidad de transmisión desde los 1200bps hasta 1,3Mbps.

Consta de un pin de transmisión, un pin de recepción, un pin de Vcc y un pin de GND. Aparte de estos cuatro pines que son los principales, también tiene un pin “key” y “state” (pin 26), que se encargan de entrar al modo de comandos AT, se activan mediante un estado alto en el pin 36; como se indica en la Figura 29.

Los comandos AT permiten configurar al módulo, permite realizar cambios como el nombre del dispositivo, generar o quitar una contraseña, ponerle en modo maestro – esclavo, etc.

Es un módulo maestro – esclavo, es decir, no solo puede estar conectado a un dispositivo móvil, sino también puede generar conexiones con otros dispositivos bluetooth.

Una vez que se ha realizado el primer enlace con otro dispositivo, éste dispone de una memoria y puede recordar la clave, así no necesita preguntar cada momento para realizar la conexión; pero se si activa el pin 26 (key) la información se elimina y el modulo solicitara de nuevo la contraseña.

Figura 29: Módulo Bluetooth HC-6.



Fuente: Comunicación Bluetooth.

Las características principales son:

- Compatible con el protocolo Bluetooth V2.0.
- Voltaje de alimentación: 3.3VDC – 6VDC.
- Voltaje de operación: 3.3VDC.
- Tamaño: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
- Corriente de operación: < 40mA
- Corriente modo *sleep*: < 1mA

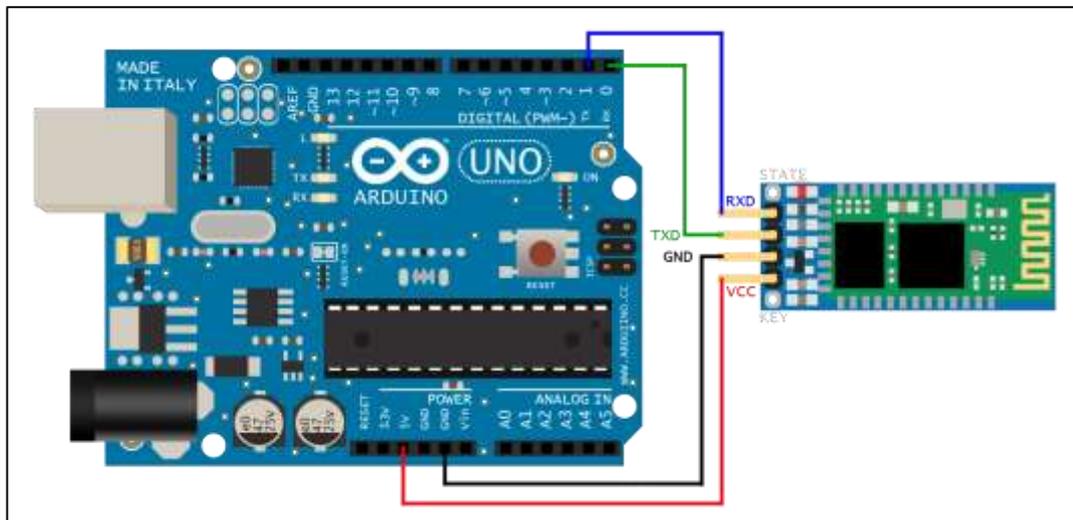
4.4.2. Conexión del módulo HC-06 con el Arduino

La conexión con el Arduino Mega 2560 es bastante sencillo ya que este dispone solo de cuatro pines que son:

Pines		
1	Recepción	Rx
2	Transmisión	Tx
3	Alimentación	Vcc
4	Tierra	GND

El pin de alimentación y de tierra se debe conectar respectivamente en el Arduino, pero los pines de comunicación se debe conectar cruzado, es decir, el Rx del módulo bluetooth va en el pin de Tx del arduino, y el Tx del módulo bluetooth va con el Rx de la placa del arduino, como se indica a continuación en la Figura 30.

Figura 30: Conexión del módulo bluetooth con Arduino.



Fuente: Comunicación Bluetooth.

Luego de que la conexión esté realizada, se debe configurar el módulo, para esto es necesario que este no se esté usando.

4.5. Dispositivo Android

Después que las señales fueron adquiridas del prototipo son presentadas de manera gráfica en un dispositivo móvil.

Hay dos tipos de datos que se muestran en dicho dispositivo, los que se toman directamente de la moto y las programadas por *software*.

El dispositivo cuenta con Android como plataforma para la aplicación. Tiene conexión Bluetooth y Acelerómetro que fue lo que se utilizó para el prototipo.

El dispositivo puede ser una Tablet o un teléfono móvil, la aplicación Android está desarrollada para cualquiera de las opciones.

Figura 31: Dispositivo Android.



Fuente: Caffelli P.

4.6. Interconexión del Sistema de Adquisición de Datos

Para la interconexión del sistema de adquisición de datos se realizó un solo conjunto con todos los elementos anteriormente descritos. Para esto fue necesario realizar conexiones físicas y eléctricas de todos los elementos.

Para la conexión se tomó en cuenta lo siguiente:

- Para el diseño de la moto se tomó en cuenta la ubicación y medidas reales de cada uno de los sensores y del dispositivo Android.
- Ubicar en un lugar seguro los circuitos y placas electrónicas, donde esté libre de aceite y gasolina.
- Debido a la cantidad de conexiones entre dispositivos, fue necesario adaptar conectores los cuales permitieron que las conexiones sean fáciles, confiables y seguras, como también se aislaron correctamente para evitar contacto con medio externo.
- Los sensores ubicados en la moto son alimentados con la batería de la misma, y deben estar conectados correctamente a los circuitos electrónicos para regular su voltaje y que la señal pueda ser recibida por el microcontrolador.

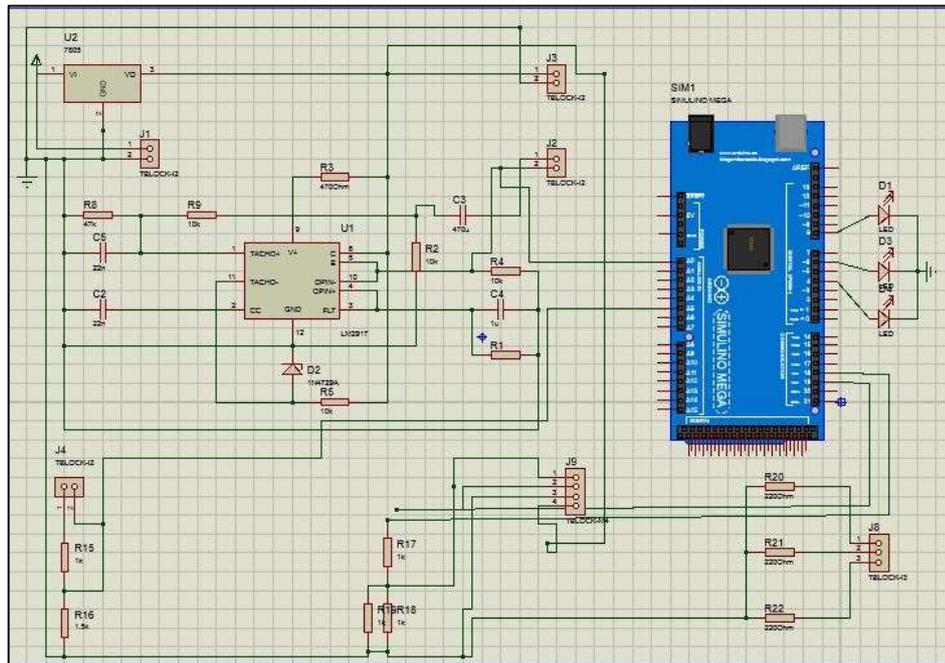
- La comunicación que se realizó con el microcontrolador fue por Bluetooth y su velocidad fue determinada en la programación del mismo.
- El dispositivo donde se mostraran los datos, será un dispositivo basado en la plataforma Android para su comunicación con Arduino.
- El momento que el dispositivo esté conectado al módulo Bluetooth, ningún otro dispositivo podrá comunicarse con el mismo.

Las placas que se adaptaron son circuitos impresos que se realizaron en el *software* ISIS PROTEUS, (Software dedicado para la creación y simulación de circuitos electrónicos), mientras que los circuitos PCB se realizaron en el *software* ARES PROTEUS (Software dedicado al diseño de circuitos impresos).

Se realizó pruebas de cada circuito por separado para verificar su correcto funcionamiento.

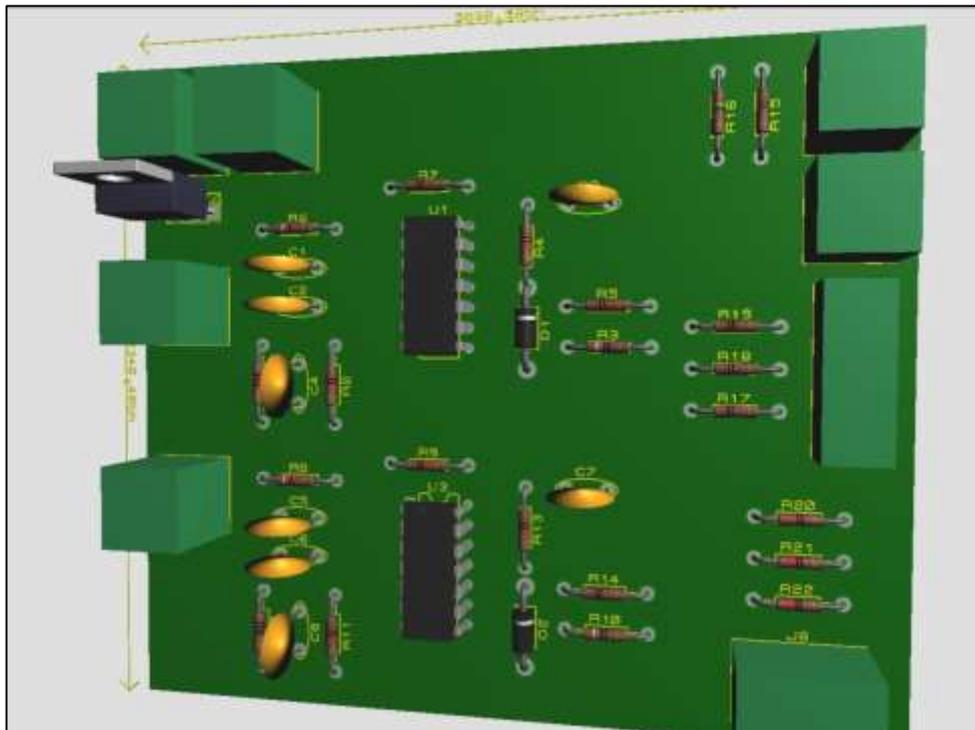
Se unió todos los circuitos en una sola placa, como también se realizó la conexión con el Arduino Mega 2560 y con el módulo Bluetooth HC-06, para que sea más fácil su ubicación y conexión en la moto, como se muestran en la Figura 32 y en la Figura 33.

Figura 32: Diseño del circuito electrónico de adquisición de datos de una moto de velocidad.



Fuente: Los autores.

Figura 33: Circuito impreso del sistema de adquisición de datos de una moto de velocidad.



Fuente: Los autores.

CAPÍTULO 5

IMPLEMENTACIÓN DEL SOFTWARE

En este capítulo se explica el desarrollo de los programas tanto para la adquisición de datos como para la interfaz en el dispositivo Android.

El objetivo es tener un software para la adquisición de datos de los sensores de la moto a través del software Arduino; como también se construyó un programa utilizando los sensores del dispositivo a través de la plataforma Android.

También se implementó un sistema de comunicación inalámbrica por medio de un módulo bluetooth que permita el intercambio de datos de manera confiable, ordenada y segura.

5.1. Arduino, implementación del *software*

5.1.1. IDE de Arduino

IDE (*Integrated Development Environment*), entorno de desarrollo integrado, es una aplicación que permite a los programadores la facilidad de crear *softwares* nuevos.

El IDE permite minimizar el tiempo para la construcción de un programa, reducir las configuraciones de los mismos siendo programas legibles.

Un IDE está compuesto por un editor de código de fuente, herramientas de programación, librerías, un depurador y un compilador.

Arduino es una plataforma de hardware y software libre, tiene un entorno de programación basado en el software *Wiring/Processing*.

Wiring es una aplicación escrita en Java, es el lenguaje de programación del microcontrolador que contiene el Arduino, pero su IDE fue desarrollado en *Processing* que es para facilitar su escritura, es un lenguaje de programación de código abierto basado en Java.

El entorno de desarrollo de Arduino se comunica con el *Hardware* Arduino, éste entorno contiene un editor de texto para que pueda escribirse el código de

programación, una barra de herramientas con botones y menús para funciones básicas, un área de mensajes y una consola que muestra los errores de compilación.

Arduino utiliza los *Sketch* para escribir el código de programación, como se muestra en la Figura 34:

Figura 34: Ventana del IDE de Arduino.



Fuente: Arduino.

La barra de herramientas y menús de un Sketch contienen lo siguiente:

Barra de herramientas:

- Verificar  : Compila el código, cheque y busca errores.
- Cargar  : Compila y graba el código programado en la placa Arduino.
- Nuevo  : Crear un nuevo código de programación.
- Abrir  : Presenta un menú de todos los programas de su librería.

- Guardar  : Guardar el código de programación.
- Monitor Serial  : Inicia la monitorización por el puerto serie.

Menú del Sketch:

- Archivo
- Editar
- Sketch
- Herramientas
- Ayuda

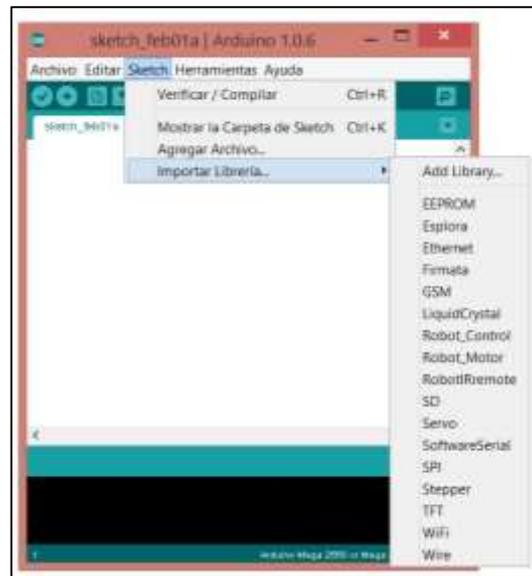
Los cuales permiten importar ejemplos, importar librerías, crear programas, se tiene un menú de ayuda, etc.

5.1.2. Librerías Arduino

Es muy importante conocer que las librerías son varios códigos de programación independientes que pueden ser parte de un programa que está en desarrollo, esto hace que sea más fácil conectarse a un sensor, a un módulo, a una memoria, entre otras.

El *software* de Arduino tiene incorporado bibliotecas pero también se puede añadir bibliotecas adicionales y se las tiene que instalar, en la Figura 35 se muestra la pantalla de las librerías incluidas en el Sketch.

Figura 35: Librerías de Arduino.



Fuente: Arduino.

5.1.3. Esquema del programa de Arduino

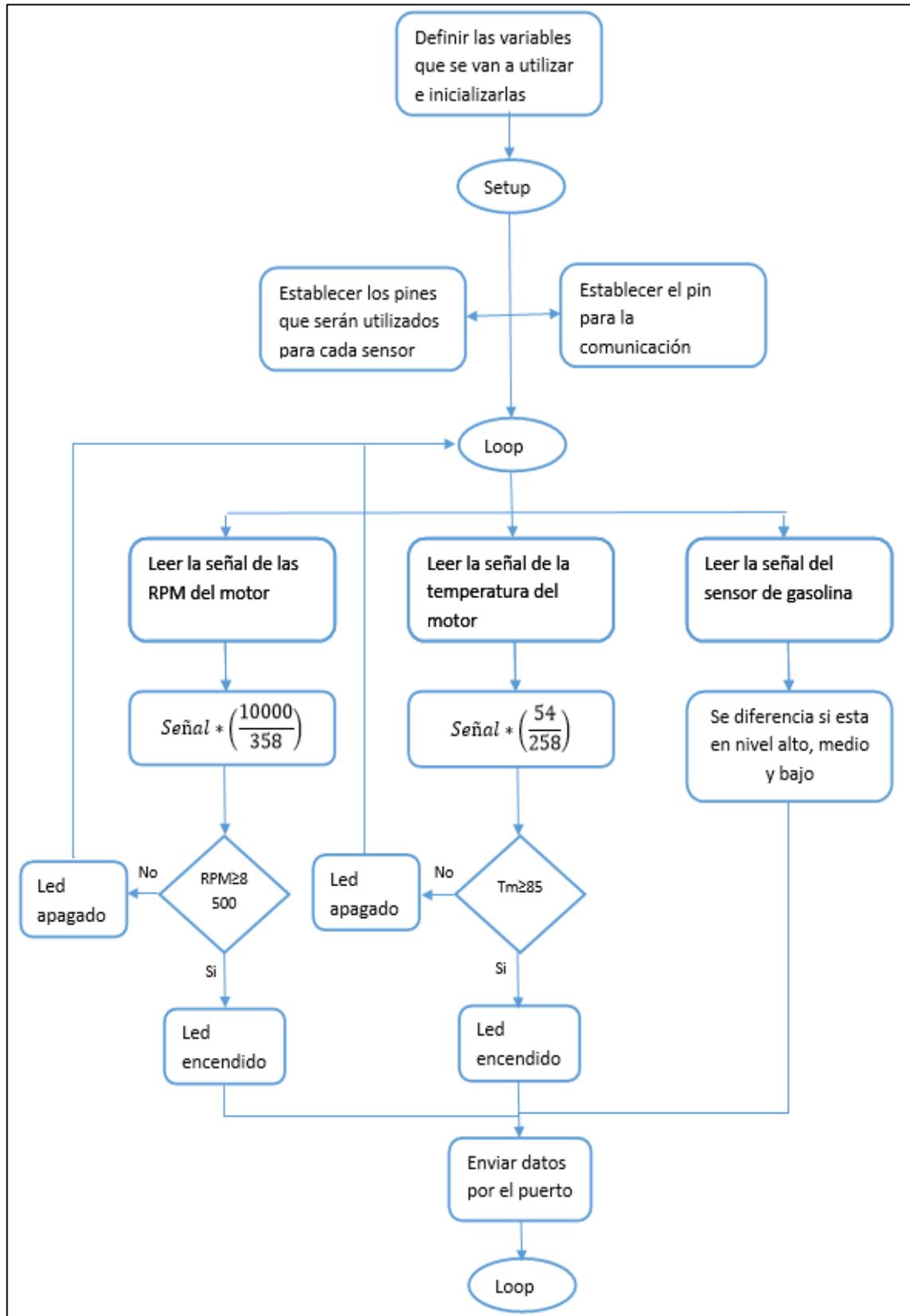
Para una programación con Arduino se debe tomar en cuenta dos funciones básicas para realizar el código; la función *setup* y la función *loop*.

La función *setup* es la cual se encarga de la configuración básica del programa como es definir variables, llamar a librerías, definir si los pines son de entrada o salida, etc. Esta función se ejecuta una vez al iniciar el programa, o cuando se reinicia el microcontrolador.

La función *loop* contiene toda la programación que va a ser ejecutada, esta función tiene un bucle infinito hasta ser detenida o bien se sale de este bucle con una interrupción y luego continúa con el mismo.

Se describe a continuación la programación de Arduino para la adquisición de los datos de los sensores ubicados en la motocicleta y la transmisión de los mismos para el módulo bluetooth. Se analizó con un diagrama de flujo y luego se realizó el programa.

Figura 36: Diagrama de flujo para el código de programación del sistema de adquisición de datos usando Arduino.



Fuente: Los autores.

Como se indica en el diagrama de flujo de la Figura 36, primero se nombra las variables a utilizar, se inicializan en cero y se las puso como un número flotante, debido que las señales que se van a recibir de los sensores que están ubicados en la moto son señales analógicas.

En seguida se define los puertos de entrada analógica A0, A3 y A5, y los puertos de salida que serán digitales 4, 6 y 8; como también se define el puerto serial que tiene en arduino. Para la comunicación se configura el puerto y se lo define por programación el Serial 1.

La siguiente función que es el *loop*, llama a las rutinas creadas para la adquisición de datos de los sensores, se realizan los respectivos cálculos para los datos de las señales esta información se construye en una trama de comunicación y se envía por el Serial 1 al módulo Bluetooth.

También se tiene rangos de los valores de cada sensor, para enviar una alarma el momento que los datos están fuera de los rangos indicados.

En anexos se muestra el código de programación de Arduino para la adquisición de datos de los sensores de la moto, y la comunicación.

5.2. Android, implementación del *software*

El dispositivo en el que se mostró los datos tiene plataforma Android, es un sistema operativo de desarrollo con código libre, el cual puede ser programado en diferentes software. El software utilizado fue Eclipse.

5.2.1. Eclipse

Eclipse es un programa informático de código abierto basado en Java; es un conjunto de servicios para construir entornos de desarrollos integrados IDE a partir de componentes conectados (plug-in), proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones, como herramientas de desarrollo de Java JDT y PDE que vienen incluidos en el IDE (entorno de desarrollo integrado) de eclipse y son herramientas de plug-in; las tres forman el SDK de Eclipse, un entorno de desarrollo completo.

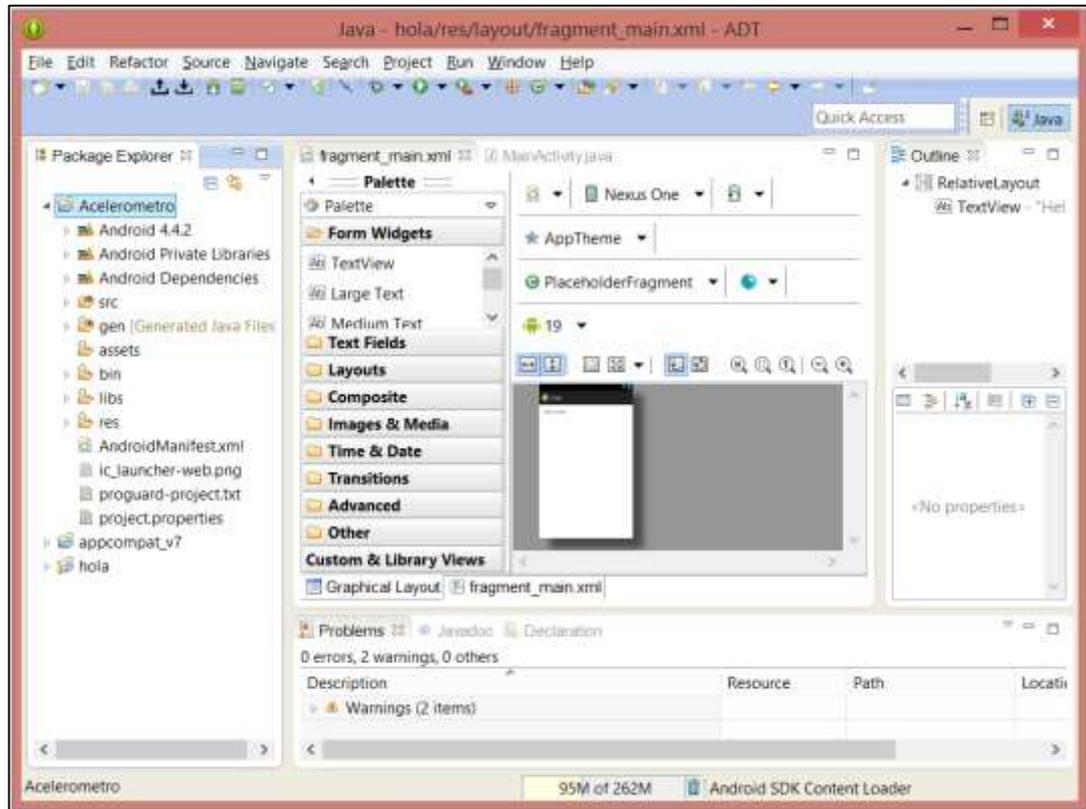
- El IDE como se explicó anteriormente permite la facilidad de crear programas nuevos, minimizando el tiempo de construcción. Se compone de herramientas, un editor, librerías, etc.
- **JDT (*Java Development Tools*):** Herramientas de desarrollo Java. Proporciona las herramientas de plug-in que implementan un IDE de Java para apoyar el desarrollo de cualquier aplicación Java; añade una perspectiva de Java para la plataforma Eclipse, así como una serie de puntos de vista, editores, asistentes, constructores y código de fusión y herramientas de refactorización.
- **PDE (*Plug-in development environment*):** Plug-in entorno de desarrollo. Proporciona una serie de vistas y editores que hacen es más fácil construir plug-ins de Eclipse. Facilita la integración de los plug-ins fácil y divertido.
- **SDK (*Software development kit*):** Kit de desarrollo de software. Es un conjunto de herramientas que sirven para el desarrollo de software que le permite al usuario crear aplicaciones para un sistema en general. (Eclipse Foundation, 2004)

5.2.1.1. Eclipse ADT

ADT (*Android Development Tools* – herramientas de desarrollo Android). Es un plug-in para el IDE Eclipse que está diseñado para la construcción de aplicaciones de Android. Amplía las capacidades de Eclipse para permitir crear a más de nuevos proyectos, crear una interfaz de usuario de la aplicación, agregar paquetes, depurar sus aplicaciones utilizando las herramientas del SDK de Android, y la exportación de archivos .apk.

5.2.1.2. Software eclipse

Figura 37: Pantalla del software Eclipse.



Fuente: Los autores.

Consta de una barra de herramientas que contiene las funciones principales:

-  Nuevo, guardar, imprimir.
-  gestor de Android SDK.
-  compilar el programa.
-  nuevo paquete de Java.
-  nueva clase de Java.

Como se muestra en la Figura 37, la pantalla contiene una paleta al lado izquierdo donde están los paquetes de los programas que se han creado, cada paquete contiene

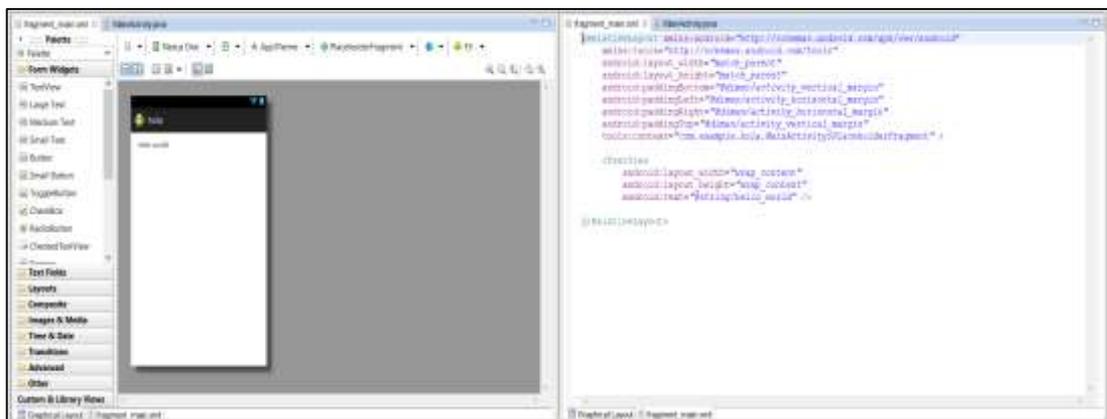
carpetas donde se encuentran los archivos (.java, .xml, .jar, .apk, etc). Desde este panel podemos escoger los archivos a programar o los que queremos modificar.

Al lado derecho tenemos una paleta con componentes que ayudan a la programación, contiene las tablas, los botones, los menús, iconos, cuadro de textos, entre otros. Estos componentes se pueden arrastrar hasta el editor de texto para una programación más sencilla.

Contiene un editor de texto, que es el que permite introducir el código de programación, eclipse permite realizar la programación de .java que es la actividad principal y la programación de .xml, estas dos son las fundamentales.

Eclipse permite trabajar tanto en una vista de diseño gráfico, como en una vista de texto, como se muestra en la Figura 38:

Figura 38: Pantallas de trabajo de eclipse, a la izquierda se tiene la pantalla gráfica de trabajo y a la derecha se tiene la pantalla de texto.



Fuente: Los autores.

5.2.2. Creación del programa de Android

El código de programación se fue construyendo por partes, realizando sus respectivas pruebas de cada código que se construyó, luego se fue uniendo la programación.

Primero se empezó programando la parte del sensor del dispositivo Android, el acelerómetro, seguido por la programación de la velocidad, la comunicación con el módulo Bluetooth y por último la conexión web.

La aplicación para el dispositivo consta de una interfaz gráfica, la cual es mostrada en la pantalla principal, esta contiene *TextView* donde se muestran los parámetros principales de los datos adquiridos, como son la velocidad, las rpm y para la temperatura del motor, muestra alarmas gráficas del nivel de la gasolina y del ángulo de inclinación, a más de esto contiene un *EditText* en el cual ingresaremos el nombre del vehículo, para poder diferenciar los datos guardados de cada usuario.

5.2.2.1. Codificación del programa: Sensor de inclinación y del ángulo de giro

Para obtener los datos de la inclinación de la moto se realizó la programación del acelerómetro del dispositivo.

En esta aplicación se visualiza en la pantalla del dispositivo los valores a los que gira el dispositivo en los tres ejes de coordenadas que se los nombró como X, Y, Z.

Al crear la aplicación “Acelerómetro”, se crea en primera instancia el manifiesto, “*AndroidManifest.xml*” el cual contiene las configuraciones principales de la aplicación, como son su versión y la configuración de la actividad principal.

Seguido se realiza la programación de “*activity_main.xml*”, que es la que se encarga de las configuraciones y los componentes de la interfaz gráfica del programa, en este caso se nombra tres cuadros de textos para mostrar las variables que representan los ejes de coordenadas.

Estos tres cuadros de textos solo se utilizó de prueba para esta primera parte del programa, en el código final se tomó en cuenta únicamente rangos a los que el piloto puede girar; se colocó un cuadro de texto que envíe una alarma al piloto cuando este fuera de estos rangos.

Finalmente se realiza la programación principal:

- Para empezar se importó las librerías necesarias para el código.
- La clase *MainActivity* es en la cual se programa el sensor con “*SensorEventListener*”.
- Se crearon las variables para los valores de los datos de cada eje como también variables para el *Timestamp* (da la fecha y hora de la última actualización o ultima vez que se detectó el movimiento).

- Se trabajó con “*onAccuracyChanged*” para evitar problemas de concurrencia al estar trabajando con el sensor.
- Se obtiene el *timestamp* de la hora y fecha actual.
- Se obtiene los valores de los tres ejes del acelerómetro.
- Se realizó una comparación de los valores de los ejes si son iguales a cero, si esto ocurre se toma los últimos valores de y se los guarda en las variables del *timestamp*.
- Se obtuvo la diferencia entre la última actualización y el *timestamp* actual, esto sirvió para calcular el movimiento. Para ello, se tomó la posición actual y se restó de la posición anterior.
- Finalmente se muestra la alarma en el dispositivo móvil.

En los anexos podemos apreciar esta parte del código de programación.

5.2.2.2. Código del programa: Adquisición de la velocidad

Para obtener los datos de la velocidad de la moto se realizó la programación a través del GPS del dispositivo.

En esta aplicación se visualiza en la pantalla del dispositivo un cuadro de texto que es en el que se muestra el valor de la velocidad.

Al crear la aplicación se establece en primera instancia el manifiesto, “*AndroidManifest.xml*”, en esta parte del código de programación a más de realizar las configuraciones se debe tomar en cuenta programar los permisos para el acceso al GPS del dispositivo móvil, como se muestra en la Figura 39 a continuación:

Figura 39: Permisos para el acceso al GPS.

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

```

Fuente: Los autores.

Luego de configurar el manifiesto, se modifica el código de “*activity_main.xml*”, en la cual únicamente se coloca un cuadro de texto donde se mostrará el valor de la velocidad.

Se realiza la programación principal:

- Para empezar se importó las librerías necesarias para el código.
- La clase *MainActivity* es en la cual se configura el sensor con “*LocationManager*”, que es el que permite utilizar el GPS del dispositivo.
- Se crearon las variables para los valores de los datos.
- Se activa el GPS.
- Se guarda el valor en la variable, se realizan cálculos y se muestra en el cuadro de texto.

5.2.2.3. Código del programa: Comunicación inalámbrica

Al crear la aplicación “*Bluetooth*” se modifica el código del manifiesto para la comunicación con el Módulo *Bluetooth*, en esta parte del código se da un permiso a la aplicación para que esta pueda hacer uso del *Bluetooth*, como se muestra en la Figura 40 a continuación el código que otorga dichos permisos:

Figura 40: Permisos para la comunicación con el módulo bluetooth.

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

Fuente: Los autores.

Con esto ya podemos realizar el código de la programación de la comunicación *Bluetooth*. Como se sabe, tenemos que empezar por la interfaz gráfica, programando el código de “*activity_main.xml*”; para la parte del *bluetooth* solo es necesario un *TextView* que nos indicara el texto recibido; cabe recalcar que este cuadro de texto solo se colocó para pruebas, en el código del programa principal no es necesario, ya que la comunicación es automática y no es necesario mostrar si el dispositivo está conectado o no, basta con un mensaje de error si el dispositivo no se pudo conectar.

Luego de tener la interfaz gráfica se procede a la programación del código principal (.java):

- Se importan las librerías que se utilizan.
- La clase *MainActivity* se programa la función de la aplicación.
- Declaramos como atributo en el código al adaptador.
- Declaramos una variable para el *TextView*.
- El método encargado de configurar inicialmente el adaptador tendrá que comprobar, en primer lugar, si el dispositivo está disponible.
- Se añade la dirección MAC del dispositivo con el cual va a comunicarse.
- Se detecta el estado en el que se encuentra el *bluetooth*. Una vez referenciado el adaptador *bluetooth*, devolverá la información dependiendo de si está activo o no.
- Se crea una clase *Handler* la cual se encarga de leer los datos recibidos desde el Arduino.
- Se recibe el mensaje, y se crea una cadena de caracteres con el mismo.
- Extrae la cadena, limpia y muestra en el *TextView*.
- También se tiene una clase que verifica la conexión, nos da un error de no encontrarla y se encarga de buscar la dirección y conectar nuevamente.

Al tener la comunicación con el módulo *bluetooth*, se recibirá ya los datos de los sensores que están ubicados en la moto, los datos de las revoluciones por minuto del motor, del sensor de gasolina, y del sensor de temperatura del motor, estos datos llegan todos como una sola cadena de caracteres delimitados entre sí por una constante para poder ser identificados, separados y mostrados en los *TextView* respectivos.

5.2.2.4. Código del programa: Conexión Web

Ésta parte del código de programación fue creado para poder enviar todos los parámetros del sistema de adquisición para ser registrados en una base de datos. Estos datos son enviados mediante un URL (*Uniform Resource Locator*).

El localizador de recursos uniforme URL es una secuencia de caracteres, tiene un formato estándar que se encarga de nombrar o llamar una dirección única a cada uno de los recursos disponibles en internet.

Un URL tiene un formato estándar, que es:

esquema://máquina/directorio/archivo

Al crear la aplicación de ésta conexión web, lo primero que se debe realizar es nombrar los permisos para la utilización de los accesos a la web y a la red. Estos permisos se nombran en el manifiesto "*AndroidManifest.xml*", como se muestra a continuación los permisos utilizados en la Figura 41.

Figura 41: Permisos para acceso a la web y a la red.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Fuente: Los autores.

Si ya se tiene los permisos para la conexión web se realiza la programación de la interfaz gráfica, la cual contiene un *TextView* para mostrar si los datos se han recibido o no en la base de datos.

Esto solo se realizó como prueba para probar la conexión, en el programa final no se necesita este *TextView*.

En el *Main Activity* configuramos la comunicación de la aplicación con la página web:

- Se importan las librerías necesarias para el código, en este caso las librerías de acceso a internet.
- Se declaran las variables a utilizar.
- Esta configuración necesita de una API distinta para lo cual se implementa un código el cual resuelve este problema.
- Se coloca un *timer* para llamar al ejecutar cada cierto tiempo, nuestro caso cada 3 segundos.
- Se genera un *String* el cual se utilizará como URL.
- Se implementa el protocolo de comunicación.
- Se comunica con la página web la cual tomará los valores enviados y los almacenará en una base de datos.
- Por último se crea un código para detección de errores el cual informará de los mismos al momento de la comunicación.

5.2.3. Diagrama de flujo del programa final

Toda la programación anteriormente descrita se unió luego de sus respectivas pruebas de funcionamiento.

Primero se unió todas las librerías que se importan para cada parte, luego se declararon las variables que se utilizaron para cada programa.

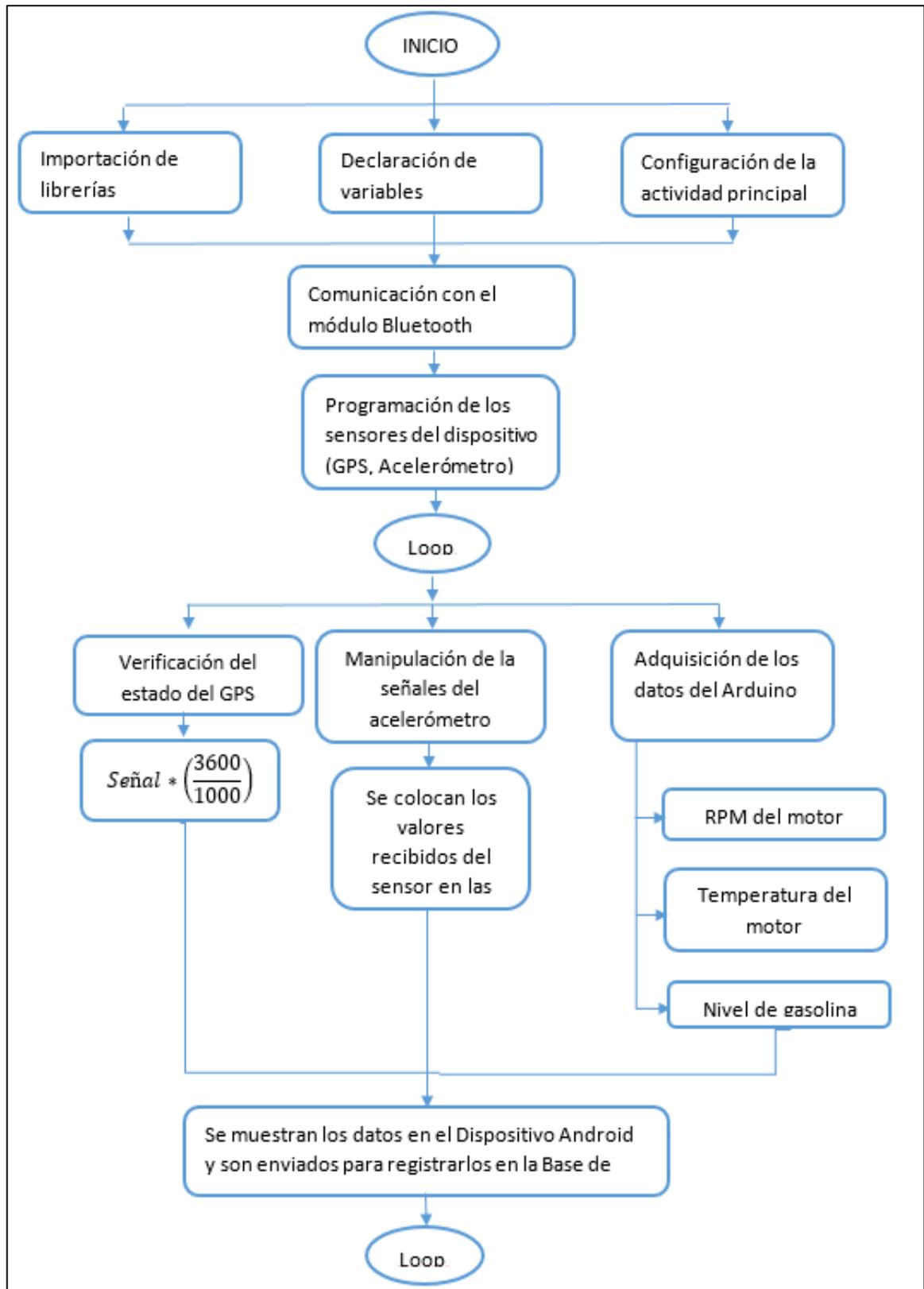
Dentro de la clase principal también se inicializó la comunicación inalámbrica de Arduino con Android a través del módulo Bluetooth.

Con la conexión activa se empezó a obtener los datos de todos los sensores de la moto, a los cuales se realizaron sus respectivos cálculos para ser mostrados en los cuadros de textos añadidos en la interfaz gráfica del programa.

Se implementó un código el cual nos da errores si hay problemas en la conexión o recepción de los datos. Cabe recalcar que el *bluetooth* y GPS deben estar activados antes del inicio del programa.

A continuación se muestra un diagrama de flujo del programa en la Figura 42:

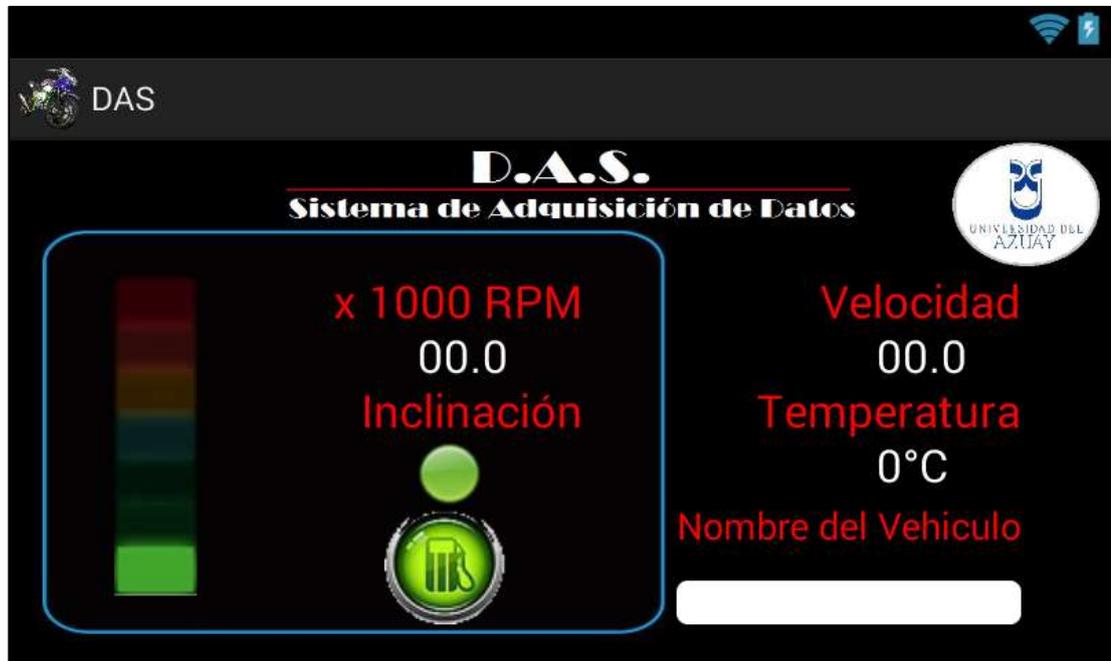
Figura 42: Diagrama de flujo para el código de programación de Android.



Fuente: Los autores.

Al realizar la programación se hizo las respectivas pruebas y por último se modificó la interfaz gráfica como se muestra a continuación en la Figura 43:

Figura 43: Interfaz gráfica de la aplicación.



Fuente: Los autores.

CAPÍTULO 6

CONSULTA DE LOS DATOS GUARDADOS DE LA MOTOCICLETA

Como se indicó en el resumen, a más del sistema de adquisición de datos se implementó un servidor web, donde todos los parámetros de la moto son guardados y el usuario tiene acceso a esos datos guardados a través de una página web, en la cual puede hacer consultas por fecha, por vehículos, etc.

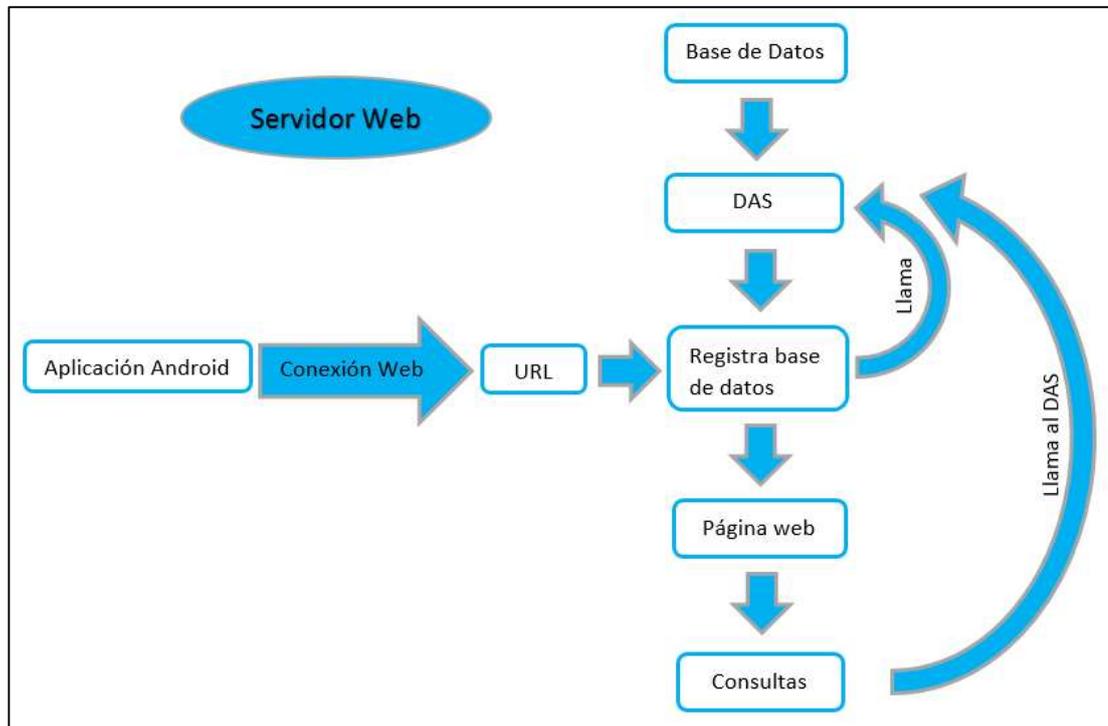
Este sistema está compuesto por diferentes parámetros:

- Una base de datos
- Un archivo DAS que contiene la información principal
- Un registra base de datos
- Una página web
- Consultas

Android tiene una conexión web con la base de datos mediante el URL (www.pinfre.com/das.html); para empezar se debe crear la base de datos, seguido por una programación que registre los datos de la aplicación y la guarde en la base, esta información será mostrada en una página web en la cual se podrá realizar consultas de diferentes parámetros.

A continuación en la Figura 44 se muestra un diagrama de flujo del funcionamiento de este sistema:

Figura 44: Diagrama de flujo para la consulta de datos guardados de la motocicleta.



Fuente: Los autores.

6.1. Base de datos

Para la creación de la base de datos en MySQL se utiliza el software “phpMyAdmin”, el cual utiliza un servidor local conocido como *localhost*.

La creación de la base de datos es muy sencilla, al ingresar en phpMyAdmin nos muestra la pantalla inicial donde nos indica los datos del servidor de base de datos, servido web, configuraciones generales del software, configuraciones de apariencia y una pantalla pequeña de MySQL *localhost* que es la que nos permite crear la base de datos.

Debajo de crear la base de datos se introduce el nombre de nuestra base y damos clic en crear, como se muestra en la Figura 45.

Figura 45: Pantalla principal para crear la base de datos.



Fuente: phpMyAdmin.

En seguida se tiene la opción de crear una tabla, se coloca un nombre a la tabla y se denomina el número de campos que se van a utilizar.

Figura 46: Creación de la tabla de la base de datos.



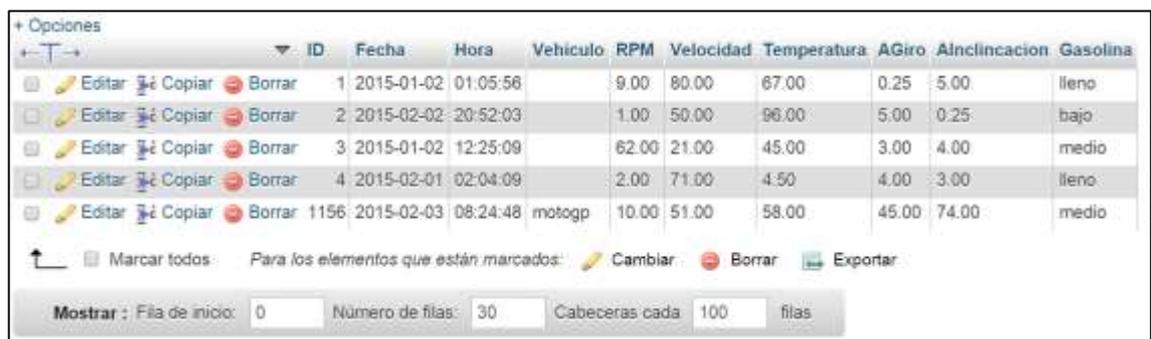
Fuente: phpMyAdmin.

Como podemos observar en la Figura 46, se puede introducir los nombre, el tipo de variable que tendríamos (entero, flotante, carácter, etc) la longitud que vamos a tener de cada campo, entre otros.

De esta manera se llamó a la tabla con el nombre de “DAS_Tabla”, que se muestra en la Figura 47, contiene diez campos para guardar los datos que se recibirán del dispositivo Android, que son:

- Id
- Fecha
- Hora
- Vehículo
- Revoluciones por minuto del motor
- Velocidad
- Temperatura del motor
- Ángulo de giro
- Ángulo de inclinación
- Nivel de gasolina

Figura 47: Tabla DAS.



ID	Fecha	Hora	Vehículo	RPM	Velocidad	Temperatura	AGiro	AInclinacion	Gasolina
1	2015-01-02	01:05:56		9.00	80.00	67.00	0.25	5.00	lleno
2	2015-02-02	20:52:03		1.00	50.00	96.00	5.00	0.25	bajo
3	2015-01-02	12:25:09		62.00	21.00	45.00	3.00	4.00	medio
4	2015-02-01	02:04:09		2.00	71.00	4.50	4.00	3.00	lleno
1156	2015-02-03	08:24:48	motogp	10.00	51.00	58.00	45.00	74.00	medio

Fuente: Los autores.

6.2. Archivo DAS

Es importante conocer que se tiene un archivo llamado “DAS.php”, éste archivo contiene todos los datos necesarios para la comunicación con la base de datos, este archivo va a ser llamado siempre que se requiera la comunicación con la base de

datos ya se para consultas o para el registro de datos, los datos que contiene este archivo son:

- Host de la base de datos
- Usuario de la base de datos
- Contraseña
- Nombre de la tabla de la base de datos

6.3. Registra la base de datos

El registra es un código PHP de programación que se creó para obtener los datos del dispositivo Android y guardarlos en la base de datos.

“Registra.php” es la página a la cual se dirige la aplicación Android, esta aplicación manda por medio del URL todos los parámetros que son recibidos o generados por la aplicación, “Registra.php” toma cada uno de estos parámetros, y los guarda en la base de datos junto con la hora y la fecha en los que fueron recibidos.

En los anexos se tiene el código de programación en PHP del registra la basa de datos.

6.4. Página web

La página web está diseñada básicamente para realizar las diferentes consultas que pueden existir, como se muestra en la Figura 48; el usuario escoge el tipo de consulta que quiere realizar de entre las siguientes opciones:

- Consulta general de los datos
- Por fecha.
- Por el nombre del vehículo.
- Máxima velocidad de los distintos vehículos registrados
- Máxima RPM de los distintos vehículos registrados.
- Velocidad media de los distintos vehículos registrados.
- RPM media de los distintos vehículos registrados

Figura 48: Pantalla página web.



Fuente: Los autores.

Para cada consulta que se realice, se va a abrir en una página diferente, con la consulta solicitada, esta consulta se mostrara a manera de tablas, las cuales tendrán uno o varios campos dependiendo del tipo de consulta.

La vista donde se pueden realizar las consultas es la siguiente:

Figura 49: Consultas.



Fuente: Los autores.

6.4.1. Consulta general

Al presionar el botón de consulta general este redirige a una página .php la cual hace la consulta de toda la base de datos, y la muestra en una tabla, para esto accede a

“Das.php” que es el documento que contiene los parámetros para la comunicación con la base de datos.

El comando utilizado para la consulta de la base de datos es “*SELECT * FROM DASTabla*”, donde DASTabla es la tabla que contiene todos parámetros del vehículo.

Esto nos muestra la consulta de la siguiente manera:

Tabla 6: Tabla de consultas de datos guardados de la moto.

D.A.S. (Data Acquisition System)									
A continuación se muestran todos los datos adquiridos									
ID	Fecha	Hora	Vehículo	RPM	Velocidad (Km/h)	Temperatura (°C)	AGiro (°)	Alucinacion (°)	Gasolina
1	2015-02-02	01:05:56		9.00	30.00	67.00	0.25	5.00	limo
2	2015-02-02	20:52:03		4.00	50.00	56.00	5.00	0.25	limo
3	2015-02-02	13:25:09		62.00	21.00	45.00	3.00	4.00	limo
4	2015-02-01	01:04:09		2.00	71.00	4.50	4.00	5.00	limo
1156	2015-02-03	08:24:43	castro	10.00	51.00	58.00	45.00	74.00	limo
1157	2015-02-03	11:44:16	castro	11.00	5.10	56.00	2.00	7.00	limo
1158	2015-02-03	11:44:24	limo	11.00	5.10	56.00	2.00	7.00	limo
1159	2015-02-03	11:45:38	limo	1.10	25.10	5.50	12.00	73.00	limo
1160	2015-02-04	09:14:59		0.00	0.00	0.00	14.65	19.25	



Fuente: Los autores.

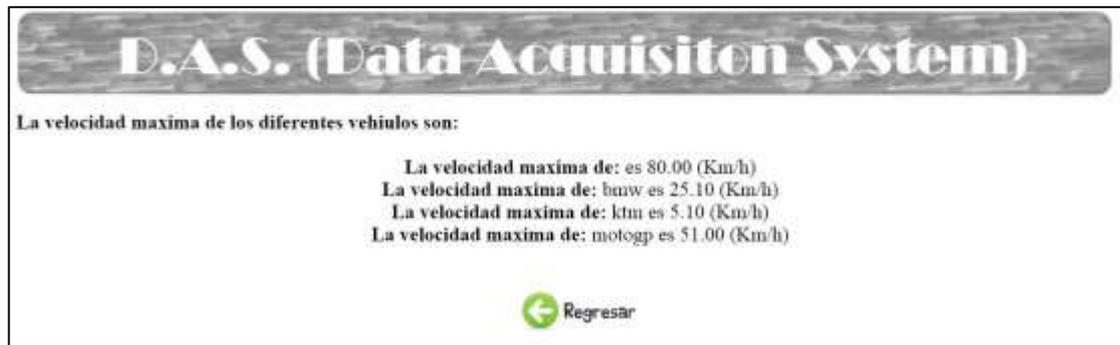
6.4.2. Consulta de la velocidad máxima según cada vehículo

Una vez que presionamos el botón para la consulta de la máxima velocidad este nos redirige a la página “maxvel.php” la cual muestra las velocidades máximas de cada vehículo registrado.

El comando que nos muestra la máxima velocidad de cada vehículo es “*SELECT Vehiculo, max(Velocidad) FROM DASTabla GROUP BY Vehiculo*”, donde Vehículo es una de las columnas de las tablas donde se registran los nombres de los vehículos que utilizan el sistema.

Esta consulta nos da como respuesta de la siguiente manera:

Figura 50: Datos de la consulta de máxima velocidad.



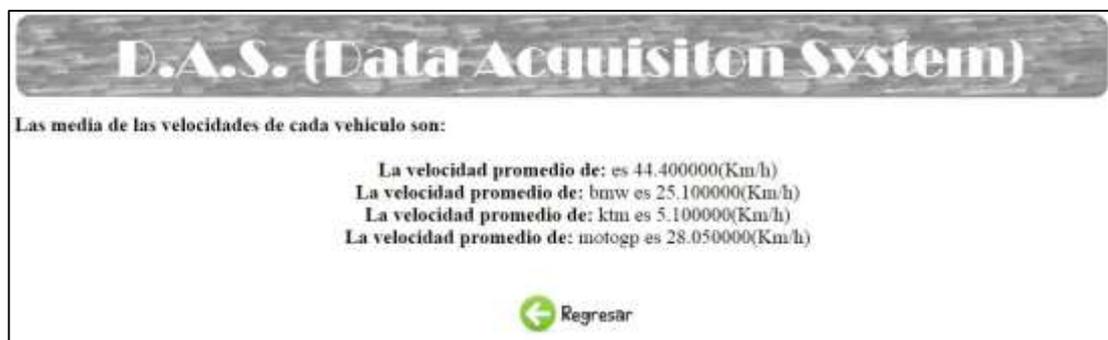
Fuente: Los autores.

6.4.3. Consulta de la velocidad media de cada vehículo

Esta consulta ocurre en la página "mevel.php" la cual imprime la media de las velocidades de cada vehículo, el comando para encontrar este valor es *"SELECT Vehiculo, AVG(Velocidad) FROM DASTabla GROUP BY Vehiculo"*.

Esta consulta genera el siguiente resultado:

Figura 51: Datos de la consulta de la velocidad media.



Fuente: Los autores.

6.4.4. Consulta de la máxima RPM de cada vehículo

Esta consulta se realiza en la página "rpmmax.php", la cual muestra los valores máximos registrados de RPM de cada vehículo, para generar esta consulta utilizamos el comando *"SELECT Vehiculo, max(RPM) FROM DASTabla GROUP BY Vehiculo"*.

El cual nos da como resultado la siguiente información:

Figura 52: Datos de la consulta de máxima RPM.



Fuente: Los autores.

6.4.5. Consulta de la media de las RPM de cada vehículo.

Esta consulta se genera en la página “merpm.php” la cual muestra la media de todas la RPM de cada vehículo, para generar esta consulta se requiere del comando *"SELECT Vehiculo, AVG(RPM) FROM DASTabla GROUP BY Vehiculo"*, el cual toma la media de las ROM de la tabla DASTabla de la columna RPM.

La consulta generada se visualiza de la siguiente manera:

Figura 53: Datos de la consulta de la media de las RPM.



Fuente: Los autores.

6.4.6. Consulta de los registros según la fecha

Esta consulta se genera en “Fecha.php” y nos muestra todos los registros específicos de la fecha incluida, esta fecha es ingresada en la página web la cual por medio del método *POST* envía la fecha a la página que genera la consulta, la cual por medio del

método *REQUEST* recibe este dato, el cual es usado para generar la consulta con el siguiente comando "*SELECT * FROM DASTabla WHERE Fecha= '\$fecha'*".

Para consultar se verá de la siguiente manera en la página web:

Figura 54: Consulta de los registros según la fecha.

Si desea consultar una fecha ingresela en el siguiente formato YY-MM-DD

Fuente: Los autores.

La consulta generada se muestra a manera de tablas de la siguiente forma:

Tabla 7: Tabla de los datos de la consulta según la fecha.

D.A.S. (Data Acquisition System)

SE MUESTRAN TODOS LOS DATOS DE LA FECHA: 2015-02-03

ID	Fecha	Hora	Vehículo	RPM	Velocidad (Km/h)	Temperatura (°C)	AGiro (°)	Aceleracion (°)	Gasolina
1156	2015-02-03	08:24:48	motop	10.60	11.00	38.00	45.00	74.00	gasolina
1157	2015-02-03	11:44:16	motop	11.00	15.10	50.00	2.00	7.00	alco
1158	2015-02-03	11:44:24	km	11.00	15.10	50.00	2.00	7.00	alco
1159	2015-02-03	11:45:30	km	1.10	25.10	5.50	82.00	73.00	gasolina

Fuente: Los autores.

6.4.7. Consulta de los registros según el nombre del vehículo

Esta consulta se genera en la página “vehiculo.php”, esta página muestra todos los registros de un vehículo especificado, el nombre de este es enviado de la página web para la consulta por medio del método *POST*, y es recibido por la “vehiculo.php” por medio del método *REQUEST*.

En la página web se verá así:

Figura 55: Consulta de los registros según el nombre del vehículo.

Si desea ver los datos de un vehiculo en especifico ingrese su nombre:

ktm

Consultar

Fuente: Los autores.

El comando para la consulta es *"SELECT * FROM DASTabla WHERE Vehiculo = '\$vehiculo'"*.

Y se muestra en la página web de la siguiente manera:

Tabla 8: Tabla de los datos de la consulta según el nombre del vehículo.

D.A.S. (Data Acquisiton System)

SE MUESTRAN TODOS LOS DATOS DEL VEHICULO: ktm

ID	Fecha	Hora	Vehiculo	RPM	Velocidad (Km/h)	Temperatura (°C)	AGiro (°)	Inclinacion (°)	Gasolina
1158	2015-02-03	11:44:24	ktm	11.00	10	56.00	2.00	2.00	aho

[← Regresar](#)

Fuente: Los autores.

CAPÍTULO 7

PRUEBAS DE FUNCIONAMIENTO Y RESULTADOS

Con el objetivo de comprobar el funcionamiento del sistema de adquisición de datos de una moto de velocidad, se realizaron pruebas para determinar la eficiencia del sistema. Se realizó simulaciones para comprobar el funcionamiento del programa de adquisición de datos de los sensores, se realizó mediciones del alcance de la señal y de la comunicación.

7.1. Diseño y construcción de la motocicleta

Como se comentó anteriormente la moto fue construida desde cero, por lo que antes de su construcción se hizo su diseño respectivo, con medidas reales de cada pieza desde la más pequeña hasta su carenado.

Dentro de este diseño se vio la ubicación de los sensores, de la placa electrónica y de la tarjeta de adquisición de datos, se tomó en cuenta por donde iba el cableado eléctrico y electrónico y donde iba a ir el dispositivo móvil.

El diseño se realizó por partes y luego se unieron todas como se muestra en la Figura 56, fue diseñado en el Software SolidWorks.

Figura 56: Diseño del prototipo.

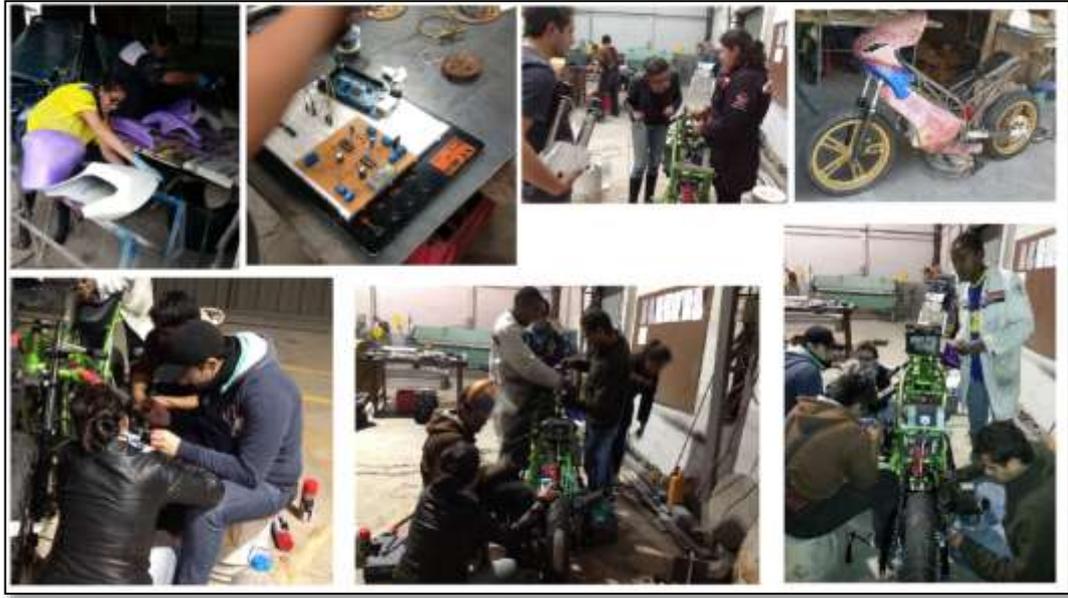


Fuente: Los autores.

De la misma manera la construcción se realizó por partes, probando cada una de las piezas y sensores, comprobando sus medidas y si su ubicación está correcta; en las pruebas se tomó en cuenta que el circuito electrónico y la placa deben estar en un lugar donde esté libre de aceite y gasolina.

Se fue realizando pruebas con cada sensor por separado, y recibiendo los datos en el dispositivo de la misma manera, como se muestra en la Figura 57.

Figura 57: Construcción del prototipo.



Fuente: Los autores.

Al estar el funcionamiento correcto se hizo la conexión correcta y el aislamiento correcto, se aseguró la caja donde estaba ubicado el circuito electrónico y la tarjeta y se colocó el dispositivo Android.

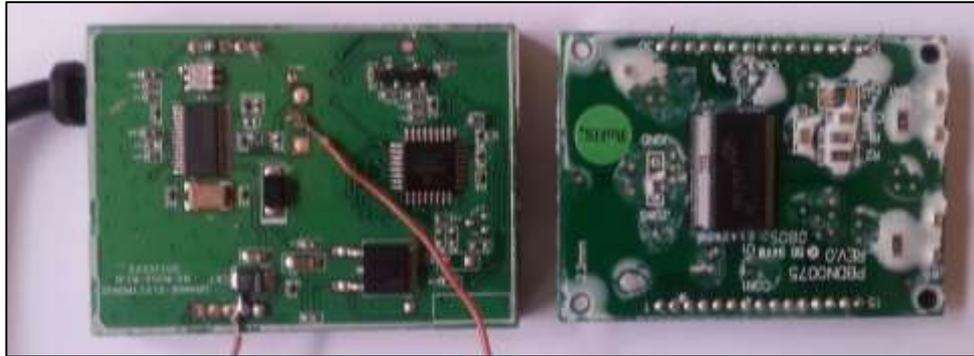
7.2. Simulación y construcción del circuito electrónico

La simulación del circuito electrónico del sistema de adquisición de datos se realizó mediante el software ISIS Proteus, se realizó con los componentes disponibles en el mercado y los que se adquirieron para realizar el circuito, también con los cálculos necesarios.

7.2.1. Pruebas del TPMS M202L

Como se explicó anteriormente se optó por abrir el módulo del sensor de presión TPMS M202L, que contiene el display, a continuación en la Figura 58 se tiene la fotografía del sensor abierto.

Figura 58: Módulo abierto del sensor M202L.



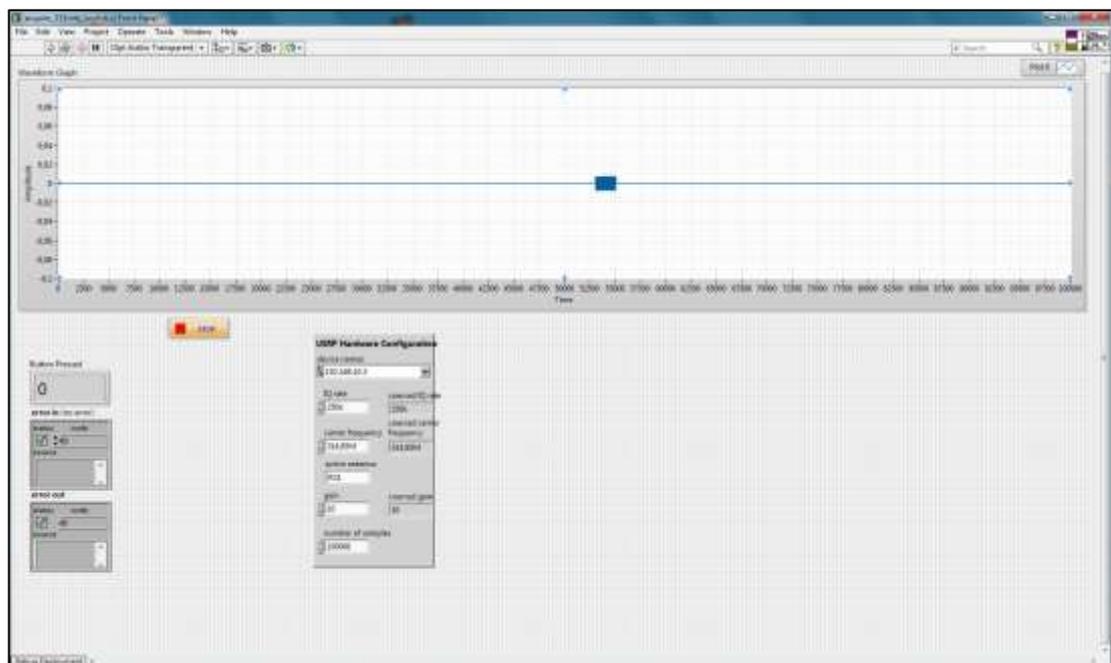
Fuente: Los autores.

Se obtuvo la señal que enviaba el sensor al módulo del display, utilizando un USRP, el dato que se pudo obtener de esta señal es que se enviaba cada 45 segundos.

La señal se observaba como un solo pulso por lo que se optó utilizar el *Software* LabView donde pudimos observar que la señal era demasiado pequeña, alrededor de los micro voltios, por lo que fue difícil obtener los datos de la señal.

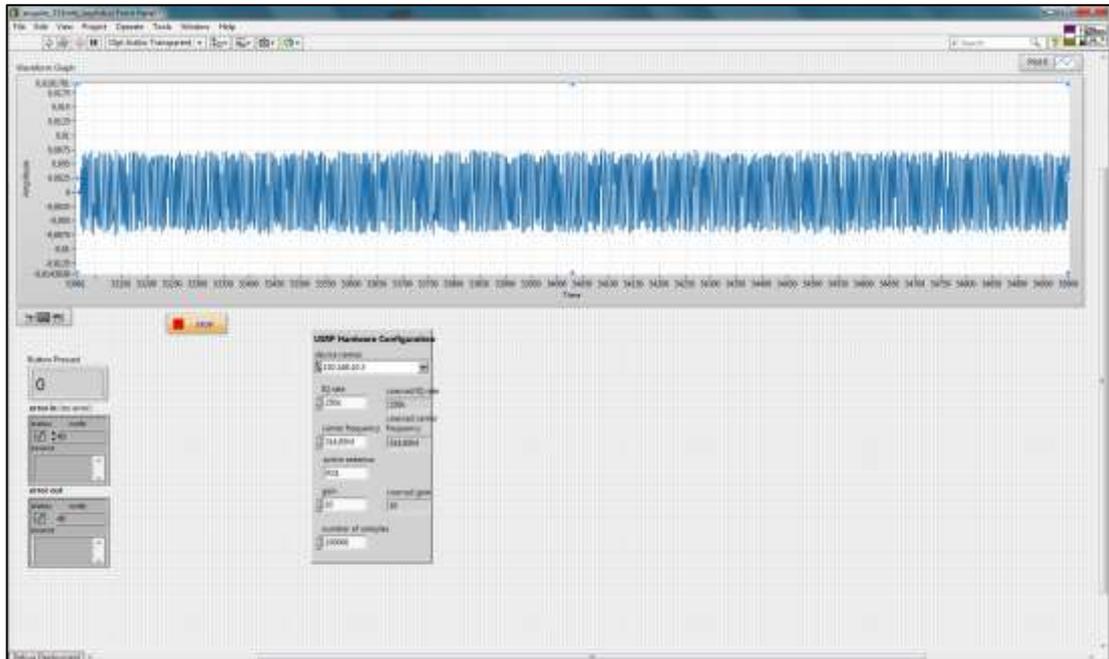
Mientras que en la placa del módulo del *display* era complejo obtener los datos, ya que la señal llegaba encriptado.

Figura 59: Señal recibida del USRP del módulo M202L.



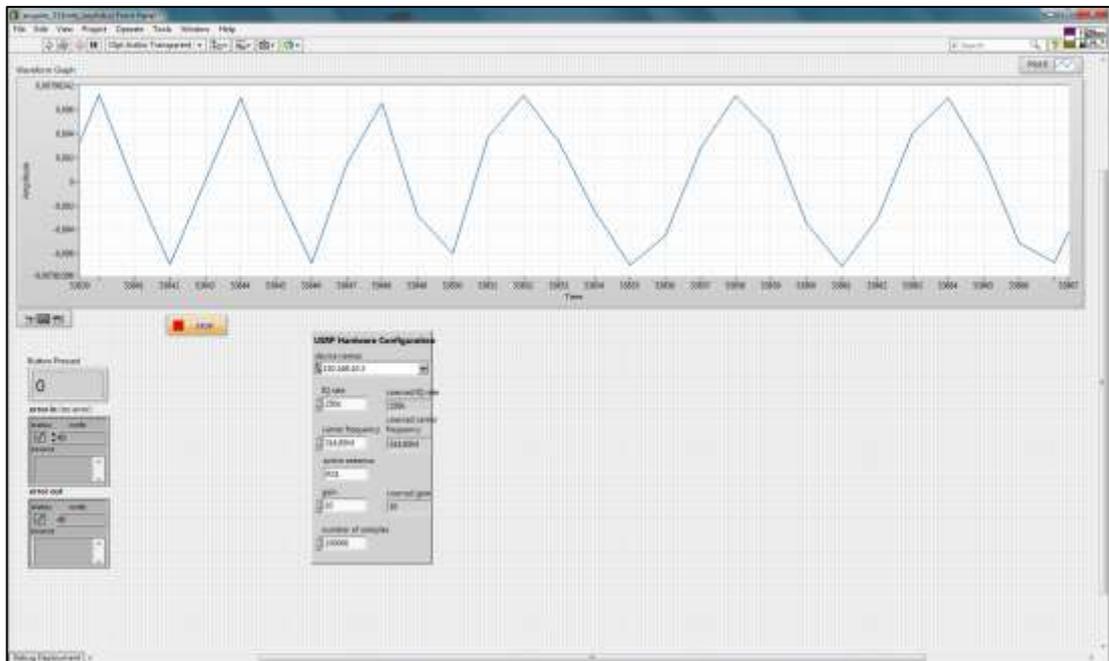
Fuente: Los autores.

Figura 60: Señal recibida del USRP del módulo M202L.



Fuente: Los autores.

Figura 61: Señal recibida del USRP del módulo M202L.



Fuente: Los autores.

En la Figura 59 se muestra la señal toma con el USRP del sensor de presión, como se puede observar la señal es demasiado pequeña está en el orden de los micro-voltios en las Figuras 60 y 61 se muestra la señal ampliada, pudiendo observar que esta modulada en frecuencia.

7.3. Pruebas con la tarjeta Arduino Mega 2560

En primer lugar se realizó la conexión de la tarjeta Arduino con el módulo bluetooth, se realizó las configuraciones del módulo y no hubo problema con la conexión por el puerto Serial 1.

Al conectar el circuito se recibieron los sin problema, lo único que se realizó son los respectivos cálculos para poder obtener los valores en las unidades de medidas correctas.

7.4. Pruebas de la Aplicación para el dispositivo Android

Para la aplicación android se hizo pruebas por códigos separados:

- Acelerómetro: para esta parte del código se tomó en cuenta que trabaja con los tres ejes de coordenadas X, Y, Z, pero para tomar el ángulo de inclinación de la moto se utilizan solo dos de ellos, y de igual forma para el ángulo de giro.

Por lo tanto para obtener estos dos ángulos fue necesario coger dos ejes y ponerle al tercero en cero, para que éste pueda ser utilizado en el otro ángulo y no se tenga problemas debido que se están utilizando los mismos ejes de coordenadas.

- GPS: también se realizó pruebas de la velocidad por GPS, al tener el dispositivo Android con 3G se dio problemas de retardo con la conexión, otro problema que se obtuvo que el dato obtenido de la velocidad estaba dado en metros/segundos, por lo que hubo que realizar una conversión a metros por kilómetros.

- **Conexión Inalámbrica:** al realizar la comunicación con el módulo Bluetooth, se realizó la configuración para la conexión automática, colocando el código MAC del módulo Bluetooth. este código se pide solo al iniciar la aplicación, obviamente el dispositivo móvil tiene que tener encendido el Bluetooth.
- **Conexión Web:** se realizaron varias pruebas de la conexión a internet, esta comunicación se realiza mediante URL con un servidor Web.

7.5. Prueba de la consulta de la base de datos

Se creó la base de datos en MySQL en un servidor local, con el archivo DAS que es el que permite el acceso a la base de datos, los otros códigos de programación llaman al mismo para acceder a la base, para esto se realizó las configuraciones respectivas del usuario, contraseña, host y la tabla de la base de datos.

Se creó la página web para información y para consultar la base de datos por diferentes parámetros.

7.6. Prototipo final

Figura 62: Prototipo final.



Fuente: Los autores.

CONCLUSIONES Y RECOMENDACIONES

- El sensor de temperatura y el sensor de revoluciones por minuto vinieron incluidos en el motor, por lo que fue una ventaja para el diseño del prototipo. El sensor de las RPM tiene un margen de error pequeño debido al cálculo y a la conversión de frecuencia a voltaje y de este a revoluciones por minuto.
- El módulo del sensor de presión que se adquirió, viene altamente protegido, esto no permite un acceso fácil a él, ya que contiene una resina que protege a las placas electrónicas y es complicado retirarla sin causar daño las placas; el momento que se abrió el módulo el Display que venía incluido se trizo y algunos pines de él se rompieron.

Los microcontroladores que vienen incluidos en los módulos tienen un código encriptado; se obtuvo la señal de transmisión en la cual va el dato de la presión y temperatura del sensor, ésta señal se pudo obtener por medio del USRP en la banda de los 315MHz cada 45 segundos, al obtener la señal no se pudo decodificar, ya que viene el código encriptado y sería un procedimiento largo.

No se pudo construir el módulo con el sensor debido que en el mercado solo hay sensores industriales con un precio elevado y de tamaño demasiado grandes como para ser incorporados en el neumático de la moto; solo concesionarios o fábricas tienen sus propios circuitos electrónicos de tamaño pequeño para esta aplicación.

- El acelerómetro funciona por separado para el cálculo del ángulo de inclinación y del ángulo de giro de la moto, esto es debido a que trabaja con tres ejes de coordenadas, en los cuales dos son utilizados para cada cálculo; es decir, se utilizan dos para el ángulo de inclinación y dos para el ángulo de giro. Por ésta razón el un eje de coordenadas queda en cero en la utilización de cada uno.

- En el cálculo de la velocidad por GPS tiene un ligero error de retardo de un segundo debido a la conexión con el satélite. También se da este error cuando la velocidad es pequeña debido a la conexión y al error de precisión del GPS. Cuando la velocidad aumenta se pierde este error.
- Al tener demasiada información la base de datos al cargar se vuelve lenta por lo que es recomendable hacer una limpieza de registros antiguos o no necesarios.
- Para una futura tesis se recomienda que se analice y construya un módulo o aplicación para sensar la presión de los neumáticos, ya que hoy en día no existen los componentes necesarios para la construcción del mismo.

BIBLIOGRAFÍA:

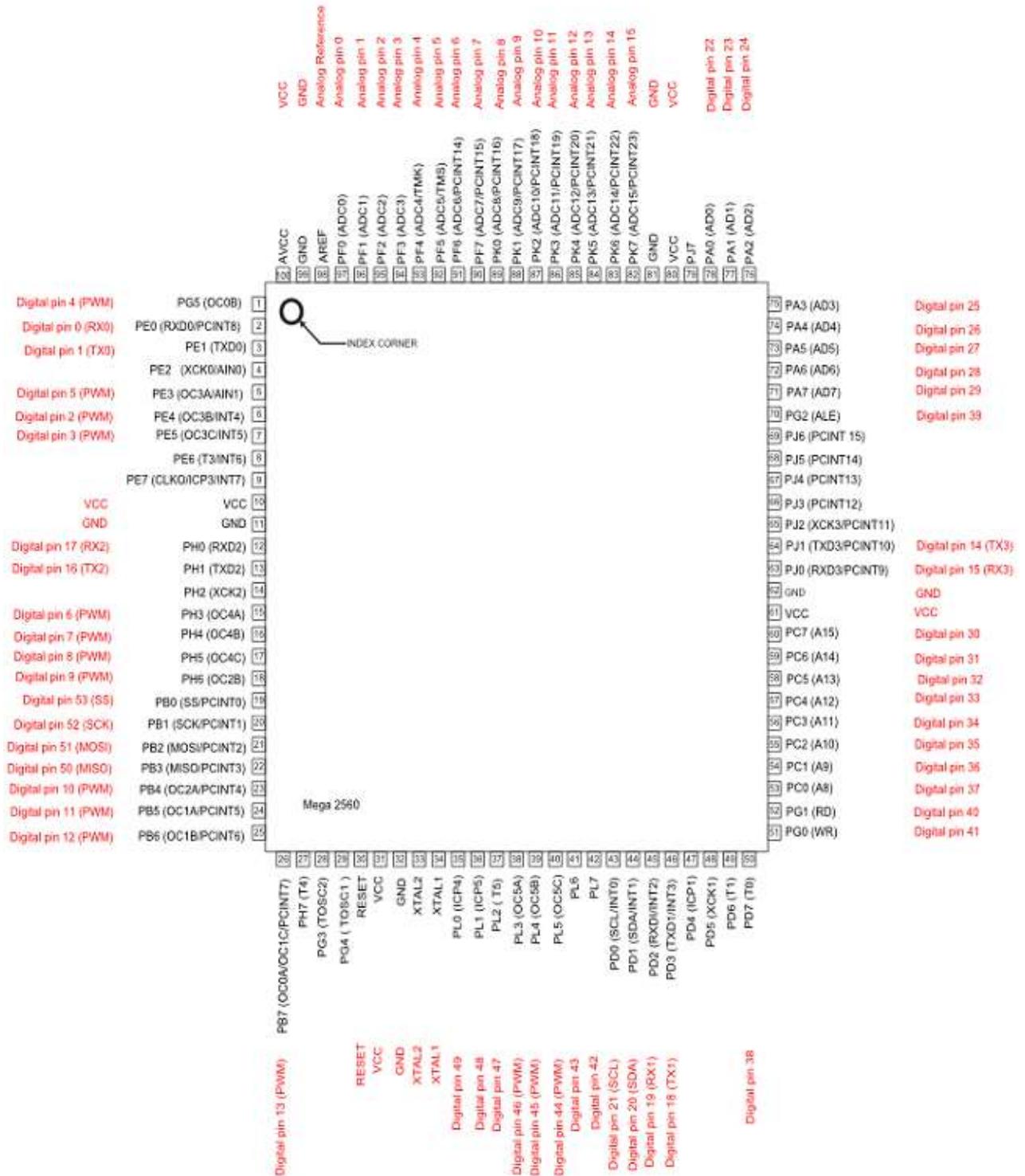
- ALVARO. (26 de Noviembre de 2014). *Velocidad del Velocímetro y Velocidad del GPS*. Obtenido de Bookmark: <http://portalvasco.com/blog/2007/09/velocidad-del-velocimetro-y-velocidad-del-gps/>
- AMBOMO, J. (2012). *Servidores Web Teoría*. España.
- APPLE. (2014). *iOS Security*. U.S.A.: Apple Inc.
- BAUSA, J., GARCÍA, C., ZARAGOZÍ, B., GIL, A., MORENO, D., & GALIANA, A. (2004). *Sensores de temperatura*. Mexico D.F.
- BELL, C. (2013). *Beginning Sensor Networks with Arduino and Raspberry Pi*. United States: Apress.
- BOJORGES, R., & RODRÍGUEZ, C. (2004). *Analizador de estados lógicos conectados a través de USB*. Mexico: Limusa S.A. de C.V.
- BOOSTER, B. (2009). *Sensores de Temperatura*.
- CANGA, R. (2011). *Curso de iniciación a las telecomunicaciones*. España: CPR de Avilés.
- CÁRDENAS, R. (2009). *Diseño Electrónico Análogo*. Colombia.
- CHEN, J., CHEN, B., GAN, Z., & LIU, S. (2005). *Solicitud de patente US 10 / 906.444*. Taiwan: Electronics Corporation.
- CHINEA, C. (2004). *Introducción al efecto doppler-Fizeau, un arma para la astrofísica*. España.
- COMPETENCIAS EN TIC. (2011). *Arquitectura de sitios web*. Argentina.
- DÉNIZ, O., SALIDO, J., & BUENO, G. (2013). *Programación de apps de visión artificial*. España: Bubok Publishing S.L.
- DONOSO, M. (2004). *Bases de datos para la Web*. Quito.
- DRAKE, J. M. (2005). *Instrumentación electrónica de comunicaciones*. España: Santander.
- ECLIPSE FOUNDATION. (2004). *Eclipse*. Obtenido de <http://www.eclipse.org/>
- ECUAVISA. (Julio de 2014). Más de 1.000 personas fallecieron en accidentes de tránsito en 5 meses. Ecuador.
- EQUADMOTOSPORT. (Marzo de 2014). *Proyecto EquadMotosport*. Obtenido de <http://www.equadmotosport.com>
- FEDERAL TRADE COMMISSION. (2011). *Cómo funcionan las aplicaciones móviles: Preguntas y respuestas*. United States.

- GALLARDO, N. (2008). *¿Qué es una página Web?* España.
- HAMMACK, B., RYAN, P., & ZIECH, N. (2012). *Eight Amazing Engineering Stories*. Illinois: Bill Hammack - 1st Edition.
- HARPER, G. (2000). *El ABC de la instrumentación en el control de procesos industriales*. Mexico: Editorial Limusa.
- HUERTA, E., MANGIATERRA, A., & NOGUERA, G. (2005). *GPS: posicionamiento satelital*. Argentina: UNR Editora.
- LAICA, W. M. (2012). *Implementación de un Banco de Pruebas para la Unidad de Control Electrónico de Vehículos con Sistemas de Inyección Electrónica*. Quito.
- LEICK, A. (2004). *GPS Satellite Surveying*. New York: John Wiley And Sons, 3th Edition.
- LENZEN, R. (2012). *MPXY8300 Design Reference Manual*. Toulouse: Freescale Toulouse Systems Laboratories.
- MARGOLIS, M. (2011). *Arduino Cookbook*. New York: O'REILLY.
- MARGOLIS, M. (2012). *Arduino Cookbook*. New York: O'Really Inc. Second Edition.
- MASSIMO. (2011). *Getting Started with Arduino*. United States: O'REILLY.
- MICROSOFT. (2014). *Windows Phone 8.1 Mobile Device Management overview*. U.S.A.: Microsoft Corporation.
- MOTOSTUDENT. (Agosto de 2014). *III Competencia Internacional Motostudent*. Obtenido de Motostudent: <http://www.motostudent.com>
- NABIPOOR, M., & MAIJLIS, B. (2006). A new passive telemetry LC pressure and temperature sensor optimized for TPMS. *Journal of Physics* (pág. 770). IOP Publishing.
- NARANJO, W. (2008). *Circuitos Integrados Digitales*. Ecuador.
- NATIONAL SEMICONDUCTER. (1995). *LM2907/LM2917 Frequency to Voltage Converter*. United States.
- ORANGE ELECTRONIC CO. LTD. (2013). *Fcc part 15 subpart c test report for Tire Pressure Monitoring System*. Taiwan: Worldwide Testing Services.
- PESANTEZ, X., & SARMIENTO, J. (2012). *Diseño e implementación de un sistema de adquisición de parámetros para la caracterización de la red de cobre de la Empresa ETAPA EP*. Cuenca.
- REVISTA LÍDERES. (2012). *La moto china se ensambla en plantas ecuatorianas*. Quito: El Comercio.
- TEXAS INSTRUMENTS. (2014). *LM555 Timer*. Estados Unidos: Texas Instruments Incorporated.

- TORRENTE, O. (2013). *Arduino Curso Práctico de formación*. Mexico: Alfaomega S.A.
- TORRES, J. P. (2011). *Efecto Doppler. En División de Ciencias Naturales y Exactas*. Mexico.
- YÁNEZ, A. (2009). *Diseño y construcción de un prototipo usando un acelerómetro tridimensional*. Quito.

Anexos:

Anexo 1: Datasheet del Microcontrolador Atmega 2560



Pin Number	Pin Name	Mapped Pin
1	PG5 (OC0B)	Digital pin 4
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5
6	PE4 (OC3B/INT4)	Digital pin 2
7	PE5 (OC3C/INT5)	Digital pin 3
8	PE6 (T3/INT6)	
9	PE7 (CLK0/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17
13	PH1 (TXD2)	Digital pin 16
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6
16	PH4 (OC4B)	Digital pin 7
17	PH5 (OC4C)	Digital pin 8
18	PH6 (OC2B)	Digital pin 9
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52
21	PB2 (MOSI/PCINT2)	Digital pin 51
22	PB3 (MISO/PCINT3)	Digital pin 50
23	PB4 (OC2A/PCINT4)	Digital pin 10
24	PB5 (OC1A/PCINT5)	Digital pin 11
25	PB6 (OC1B/PCINT6)	Digital pin 12
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46
39	PL4 (OC5B)	Digital pin 45
40	PL5 (OC5C)	Digital pin 44
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21
44	PD1 (SDA/INT1)	Digital pin 20
45	PD2 (RXDI/INT2)	Digital pin 19
46	PD3 (TXD1/INT3)	Digital pin 18
47	PD4 (ICP1)	
48	PD5 (XCK1)	

49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15
64	PJ1 (TXD3/PCINT10)	Digital pin 14
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0

98	AREF	Analog
99	GND	GND
100	AVCC	VCC

Tabla: Asignación de pines Arduino MEGA tomada de

Anexo 2: Análisis de las señales del motor.

