



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES
Y REDES

“IMPLEMENTACIÓN DE UN PROTOTIPO ELECTRÓNICO
PARA REGISTROS TELEMÁTICOS Y DETECCIÓN DE FALLOS
EN MOTORES DE AUTOMÓVILES MEDIANTE SISTEMA
OBD II.”

Trabajo de titulación

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES

AUTORA: MONTESDEOCA VIVANCO ANDREA GABRIELA

DIRECTOR: Ing. ALTAMIRANO SANTILLÁN EDWIN VINICIO

Riobamba – Ecuador

2021

©2020, Andrea Gabriela Montesdeoca Vivanco.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Andrea Gabriela Montesdeoca Vivanco, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autora asumo la responsabilidad legal y académica de los contenidos de este trabajo de Titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.
Riobamba, 26 de febrero de 2021.



Andrea Gabriela Montesdeoca Vivanco
2300271380

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y
REDES

El Tribunal del trabajo de titulación certifica que el trabajo de investigación, tipo: Dispositivo Tecnológico, **IMPLEMENTACION DE UN PROTOTIPO ELECTRÓNICO PARA REGISTROS TELEMÁTICOS Y DETECCIÓN DE FALLOS EN MOTORES DE AUTOMÓVILES MEDIANTE SISTEMAS OBD II**, realizado por la señorita **ANDREA GABRIELA MONTESDEOCA VIVANCO**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA (2021/02/26)
Dr. Mario Humberto Paguy Cuvi PRESIDENTE DEL TRIBUNAL	 MARIO HUMBERTO PAGUAY CUVI <small>Firmado digitalmente por MARIO HUMBERTO PAGUAY CUVI Fecha: 2021.03.11 06:31:21 -05'00'</small>	(2021/02/26)
Ing. Edwin Vinicio Altamirano Santillán. DIRECTOR DEL PROYECTO DE TITULACIÓN	 EDWIN VINICIO ALTAMIRAN O SANTILLAN <small>Firmado digitalmente por EDWIN VINICIO ALTAMIRANO SANTILLAN Fecha: 2021.03.11 09:03:21 -05'00'</small>	(2021/02/26)
Dr. Geovanny Estuardo Vallejo Vallejo MIEMBRO DEL TRIBUNAL	 GEOVANNY ESTUARDO VALLEJO VALLEJO <small>Firmado digitalmente por GEOVANNY ESTUARDO VALLEJO VALLEJO Fecha: 2021.03.10 21:20:27 -05'00'</small>	(2021/02/26)

DEDICATORIA

Esta tesis está dedicada:

Principalmente a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados y agradezco a mi tutor Ingeniero Edwin Altamirano, quien con sus conocimientos y apoyo me guio a través de cada una de las etapas de mi tesis para alcanzar los resultados que buscaba.

Quiero dedicar y agradecer a toda mi familia, por apoyarme aun cuando mis ánimos decaían. En especial, quiero mencionar a mis padres, todo este esfuerzo está dedicado a mi Madrecita querida porque sé que ella me ayudó en las buenas y en las malas y lo sigue haciendo, además de haberme dado la vida siempre confió en mí y nunca me abandonó, Te Amo mamita mi querida Carmita y a mi Papito amado mi Lito que a pesar de tantas enfermedades ha luchado por darme una sonrisa y mucho amor a su chiquita, 'su ultimita'. Mis amados padres siempre estuvieron ahí para darme palabras de apoyo y un abrazo reconfortante para renovar energías, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy he sido muy orgullosa y he tenido el privilegio de ser su hija, son los mejores padres.

Finalmente, a mi novio Alexis que en el camino te encontré e iluminaste mi vida, con su apoyo alcance de mejor manera mis metas, a través de sus consejos, de su amor, y paciencia me ayudo a concluir esta meta, por apoyarme cuando más lo necesite, por extender su mano en momentos difíciles y por el amor brindado cada día, de verdad muchas gracias, siempre lo llevo en mi corazón te amo mucho.

Muchas gracias a todos.

Andrea M.

AGRADECIMIENTO

A agradezco a Dios por ser mi guía y acompañarme en el transcurso de mi vida, brindándome paciencia y sabiduría para culminar con éxito mis metas propuestas.

Mi profundo agradecimiento a todas las autoridades y personal que hacen la Escuela Superior Politécnica de Chimborazo, por confiar en mí, abrirme las puertas y darme todos los conocimientos necesarios para concluir mi carrera quienes dentro de esta institución con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

A mis padres por haberme dado la oportunidad de formarme en esta prestigiosa universidad y haber sido mi apoyo durante todo este tiempo, ser mi pilar fundamental y haberme apoyado incondicionalmente, pese a las adversidades e inconvenientes que se presentaron.

Por el esfuerzo, dedicación, paciencia, por su confianza y por todo lo que me ha dado a lo largo de mi carrera y de mi vida, este proyecto de titulación va dedicado a mi Madre se lo agradezco por todo lo que no se rindió por mi te amo Sra. Carmita.

Finalmente quiero expresar mi más grande y sincero agradecimiento al Ingeniero Edwin Altamirano, principal colaborador durante todo este proceso, quien, con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo de titulación y no solo por haberme guiado en la elaboración de este trabajo de titulación, sino a lo largo de mi carrera universitaria y haberme brindado el apoyo para desarrollarme profesionalmente y seguir cultivando mis valores.

Andrea M.

TABLA DE CONTENIDO

INDICE DE TABLAS	xiv
INDICE DE FIGURAS	xv
INDICE DE GRÁFICOS	xviii
INDICE DE ANEXOS	xix
RESUMEN	xx
ABSTRACT	xxi
INTRODUCCIÓN	1
CAPITULO 1	
1 MARCO TEORICO REFERENCIAL	6
1.1 La electrónica automotriz	6
1.2 Sensores Automotrices	6
1.3 Tipos de sensores Automotrices	7
1.3.1 Sensor ESP	7
1.3.2 <i>Sensor de velocidad</i>	8
1.3.3 <i>Sensor de impulsos</i>	9
1.3.4 <i>Sensor de oxigeno</i>	9
1.3.5 <i>Sensor CMP</i>	10
1.3.6 <i>Sensor CKP</i>	10
1.3.7 <i>Sensor TPS</i>	11
1.3.8 <i>Sensor IAT</i>	11
1.3.9 <i>Sensor MAP</i>	12
1.3.10 <i>Sensor MAF</i>	12
1.4 Almacenamiento de datos de los sensores	13
1.4.1 <i>Datalogger</i>	13
1.4.2 <i>Thingspeak</i>	14
1.5 Etapa de Control	14
1.5.1 <i>Tarjeta de desarrollo</i>	15
1.5.1.1 <i>Arduino</i>	15

1.5.1.2	<i>Tipos de tarjetas de desarrollo Arduino</i>	16
1.5.1.3	<i>Raspberry pi</i>	17
1.6	Pantalla de Cristal Líquido (LCD)	17
1.6.1	<i>Memorias del LCD</i>	18
1.6.2	<i>Memoria DDRAM</i>	18
1.6.3	<i>Memoria CGROM y CGRAM</i>	19
1.7	Comunicación	19
1.7.1	<i>Comunicación alámbrica</i>	19
1.7.1.1	<i>Ventajas de la comunicación alámbrica</i>	20
1.7.1.2	<i>Desventajas de la comunicación alámbrica</i>	20
1.7.2	<i>Comunicación inalámbrica</i>	20
1.7.2.1	<i>Ventajas del protocolo de comunicación Wiffi.</i>	21
1.7.2.2	<i>Desventajas del protocolo de comunicación Wiffi.</i>	21
1.7.3	<i>Módulos de comunicación</i>	21
1.7.3.1	<i>Módulo GSM 1800L</i>	22
1.7.3.2	<i>Especificaciones técnicas del Módulo GSM 1800L</i>	22
1.8	Módulo lector de memoria Micro SD	23
1.8.1	<i>Especificaciones técnicas del módulo lector de memoria Micro SD</i>	23
1.9	Sistema OBD - II	24
1.9.1	<i>Compatibilidad</i>	24
1.9.2	<i>Conector y comunicación</i>	25
1.9.2.1	<i>Características</i>	26
1.10	Software de diseño CAD	26
1.10.1	<i>Solidworks</i>	26
CAPITULO II		
2	PROPUESTA Y DISEÑO DE PROTOTIPO	28
2.1	Estructura de la tarjeta Electrónica del prototipo	29
2.1.1	<i>Conexión Arduino – modulo micro SD</i>	29
2.1.1.1	<i>Diseño esquemático del circuito Arduino –micro SD</i>	30

2.1.2	Conexión Arduino – módulo SIM 800L	32
2.1.2.1	<i>Diseño esquemático del circuito Arduino - módulo SIM 800L</i>	33
2.1.3	Conexión Arduino – pantalla LCD	34
2.1.3.1	<i>Interfaz de conexión I2C</i>	34
2.1.3.2	<i>Diseño esquemático del circuito Arduino - LCD</i>	35
2.1.4	Conexión Arduino - sistema OBDII	36
2.1.4.1	<i>Diseño esquemático del circuito Arduino - sistema OBD II</i>	36
2.1.5	Circuito esquemático general de la tarjeta electrónica	37
2.1.6	Diseño PCB del circuito electrónico	38
2.1.6.1	<i>Diseño de pistas</i>	38
2.1.7	Visualización 3D	40
2.2	Implementación del prototipo	41
2.2.1	Construcción de la placa PCB	41
2.2.2	Colocación de elementos y módulos electrónicos	42
2.3	Construcción de la caja de protección	43
2.3.1	Construcción final de la caja de protección	43
2.3.2	Armado total del prototipo	44
2.3.3	Prototipo Final	45
2.4	Diseño del algoritmo de programación.	45
2.5	Funcionamiento del prototipo	47
2.5.1	Pantalla principal	47
2.5.2	Leds indicadores	47
2.5.3	Puerto USB de programación Arduino	48
2.5.4	Puerto de alimentación	49
2.5.5	Ranura para micro SD	49
2.5.6	Puerto de conexión para sistema OBD II	50
CAPITULO III		
3	VALIDACIÓN DEL PROTOTIPO	51
3.1	Introducción	51

3.2	Análisis de funcionamiento	51
3.2.1	<i>Funcionamiento del prototipo</i>	52
3.3	Visualización de resultados en Datalogger	53
3.3.1	<i>Visualización de datos en Excel</i>	53
3.4	Visualización de resultados en Thingspeak	56
3.5	Adquisición de datos del sensor RPM	57
3.5.1	<i>Datos del sensor RPM almacenados en el Datalogger</i>	57
3.5.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor RPM</i>	58
3.5.2.1	<i>Promedio de los datos del sensor RPM</i>	58
3.5.2.2	<i>Error absoluto</i>	59
3.5.2.3	<i>Cálculo del error relativo</i>	60
3.5.3	<i>Datos del sensor RPM visualizados en la plataforma Thingspeak</i>	60
3.6	Adquisición de datos del sensor temperatura de refrigerante	60
3.6.1	<i>Datos del sensor Temperatura de refrigerante almacenados en el Datalogger</i>	61
3.6.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de refrigerante</i>	62
3.6.2.1	<i>Promedio de los datos del sensor Temperatura de refrigerante</i>	62
3.6.2.2	<i>Error absoluto</i>	63
3.6.2.3	<i>Cálculo del error relativo</i>	64
3.6.3	<i>Datos del sensor temperatura de refrigerante visualizados en la plataforma Thingspeak.</i>	64
3.7	Adquisición de datos de sensor temperatura de aceite	64
3.7.1	<i>Datos del sensor temperatura de Aceite almacenados en el Datalogger</i>	65
3.7.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de aceite</i>	66
3.7.2.1	<i>Promedio de los datos del sensor Temperatura de aceite</i>	66
3.7.2.2	<i>Error absoluto</i>	67
3.7.2.3	<i>Cálculo del error relativo</i>	68
3.7.3	<i>Datos del sensor temperatura de aceite visualizados en la plataforma Thingspeak</i> .68	
3.8	Adquisición de datos de sensor temperatura de admisión	68

3.8.1	<i>Datos del sensor temperatura de admisión almacenados en el Datalogger</i>	69
3.8.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de admisión</i>	70
3.8.2.1	<i>Promedio de los datos del sensor Temperatura de admisión</i>	70
3.8.2.2	<i>Error absoluto</i>	70
3.8.2.3	<i>Cálculo del error relativo</i>	71
3.8.3	<i>Datos del sensor temperatura de admisión visualizados en la plataforma Thingspeak</i>	72
3.9	Adquisición de datos de sensor MAF	72
3.9.1	<i>Datos del sensor MAF almacenados en el Datalogger</i>	73
3.9.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor MAF</i>	74
3.9.2.1	<i>Promedio de los datos del sensor MAF</i>	74
3.9.2.2	<i>Error absoluto</i>	74
3.9.2.3	<i>Cálculo del error relativo</i>	75
3.9.3	<i>Datos del sensor MAF visualizados en la plataforma Thingspeak</i>	76
3.10	Adquisición de datos de sensor MAP	76
3.10.1	<i>Datos del sensor MAP almacenados en el Datalogger</i>	77
3.10.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor MAP</i>	78
3.10.2.1	<i>Promedio de los datos del sensor MAP</i>	78
3.10.2.2	<i>Error absoluto</i>	78
3.10.2.3	<i>Cálculo del error relativo</i>	79
3.10.3	<i>Datos del sensor MAP visualizados en la plataforma Thingspeak</i>	80
3.11	Adquisición de datos de sensor de velocidad	80
3.11.1	<i>Datos del sensor de velocidad almacenados en el Datalogger</i>	81
3.11.2	<i>Cálculo del promedio, error absoluto y relativo de los datos de velocidad</i>	82
3.11.2.1	<i>Promedio de los datos del sensor de velocidad</i>	82
3.11.2.2	<i>Error absoluto</i>	82
3.11.2.3	<i>Cálculo del error relativo</i>	83
3.11.3	<i>Datos del sensor de velocidad visualizados en la plataforma Thingspeak</i>	84
3.12	Adquisición de datos de la distancia recorrida	84

3.12.1	<i>Datos del sensor de distancia almacenados en el Datalogger</i>	85
3.12.2	<i>Cálculo del promedio, error absoluto y relativo de los datos de distancia recorrida</i> 86	
3.12.2.1	<i>Promedio de los datos de la distancia recorrida</i>	86
3.12.2.2	<i>Error absoluto</i>	86
3.12.2.3	<i>Cálculo del error relativo</i>	87
3.12.3	<i>Datos del sensor de distancia visualizados en la plataforma Thingspeak</i>	88
3.13	Adquisición de datos del sensor posición acelerador	88
3.13.1	<i>Datos del sensor posición acelerador almacenados en el Datalogger</i>	89
3.13.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor de posición del acelerador</i>	90
3.13.2.1	<i>Promedio de los datos del sensor de posición del acelerador</i>	90
3.13.2.2	<i>Error absoluto</i>	90
3.13.2.3	<i>Cálculo del error relativo</i>	91
3.13.3	<i>Datos del sensor posición acelerador visualizados en la plataforma Thingspeak.</i> ..	92
3.14	Adquisición de datos del sensor temperatura ambiente	92
3.14.1	<i>Datos del sensor temperatura ambiente almacenados en el Datalogger</i>	93
3.14.2	<i>Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura ambiente</i>	94
3.14.2.1	<i>Promedio de los datos del sensor Temperatura ambiente</i>	94
3.14.2.2	<i>Error absoluto</i>	95
3.14.2.3	<i>Cálculo del error relativo</i>	96
3.14.3	<i>Datos del sensor Temp ambiente visualizados en la plataforma Thingspeak</i>	96
3.15	Análisis de consumo del Prototipo	97
3.16	Análisis de costos del prototipo	98
3.17	Análisis comparativo	99
3.17.1	<i>Comparación de datos del sensor RPM</i>	100
3.17.2	<i>Comparación de datos del sensor MAP</i>	101
3.17.3	<i>Comparación de datos del sensor Temperatura de refrigerante</i>	103
3.18	Análisis comparativo de equipos	104
	CONCLUSIONES	107

RECOMENDACIONES..... 108

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

INDICE DE TABLAS

Tabla 1-1:	Características generales del Arduino Mega	16
Tabla 1-3:	Datos procesados en Excel del sensor RPM.....	57
Tabla 2-3:	Error absoluto de cada dato obtenido por el sensor RPM	59
Tabla 3-3:	Datos obtenidos en Excel del sensor temperatura de refrigerante	61
Tabla 4-3:	Error Absoluto de cada dato obtenido por el sensor Temp de refrigerante.	63
Tabla 5-3:	Datos procesados en Excel del sensor de temperatura de aceite	65
Tabla 6-3:	Error Absoluto de cada dato obtenido	67
Tabla 7-3:	Datos procesados en Excel del sensor	69
Tabla 8-3:	Error Absoluto de cada dato obtenido por	71
Tabla 9-3:	Datos procesados en Excel del sensor MAF	73
Tabla 10-3:	Error Absoluto de cada dato obtenido por el sensor MAF	75
Tabla 11-3:	Datos procesados en Excel del sensor MAP	77
Tabla 12-3:	Error Absoluto de cada dato obtenido por el sensor MAP	79
Tabla 13-3:	Datos procesados en Excel del sensor de velocidad.....	81
Tabla 14-3:	Error Absoluto de cada dato obtenido por el sensor de velocidad	83
Tabla 15-3:	Datos procesados en Excel de la distancia recorrida por el vehículo.	85
Tabla 16-3:	Error Absoluto de los datos obtenido de la distancia recorrida	87
Tabla 17-3:	Datos procesados en Excel del sensor posición aceleradora	89
Tabla 18-3:	Error Absoluto de cada dato obtenido por el sensor de posición de aceleración. 91	
Tabla 19-3:	Datos procesados en Excel del sensor temperatura ambiente.....	93
Tabla 20-3:	Error Absoluto de cada dato obtenido por el sensor Temperatura ambiente.....	95
Tabla 21-3:	Elementos electrónicos utilizados	97
Tabla 22-3:	Costos de fabricación del prototipo	98
Tabla 23-3:	Tabla de costos para realización de porcentajes	99
Tabla 24-3:	Scanner más utilizados	123
Tabla 25-3:	Porcentaje de costos en los scanner automotriz.....	125

INDICE DE FIGURAS

Figura 1-1:	Sistemas electrónicos integrados a un vehículo.....	6
Figura 2-1:	Sensores comandados por una ECU.	7
Figura 3-1:	Componente físico de un sensor ESP	8
Figura 4-1:	Componente Físico de un sensor de velocidad	8
Figura 5-1:	Componente físico de un sensor de impulsos	9
Figura 6-1:	Componente físico de un sensor de oxígeno.....	9
Figura 7-1:	Componente físico de un sensor CMP.....	10
Figura 8-1:	Componente físico de un sensor CKP.....	10
Figura 9-1:	Componente físico de un sensor TPS	11
Figura 10-1:	Componente físico de un sensor IAT.....	11
Figura 11-1:	Componente físico de un sensor MAP.....	12
Figura 12-1:	Componente físico de un sensor MAF.....	13
Figura 13-1:	Entorno de visualización de un Datalogger	14
Figura 14-1:	Entorno de visualización de Thingspeak	14
Figura 15-1:	Arduino Mega	15
Figura 16-1:	Tarjeta de desarrollo Raspberry Pi.....	17
Figura 17-1:	Pantalla de Cristal Líquido de 2x16.....	18
Figura 18-1:	Memoria DDRAM del LCD	18
Figura 19-1:	Memoria CGROM de LCD	19
Figura 20-1:	Entorno de conexión alámbrica.....	20
Figura 21-1:	Protocolo de comunicación Wiffi.	21
Figura 22-1:	Módulo GSM 1800L.....	22
Figura 23-1:	Módulo lector de memoria Micro SD.....	23
Figura 24-1:	Conector OBD - II	24
Figura 25-1:	Certificación OBDII de cada vehículo.....	25
Figura 26-1:	Diseños realizados por el software Solidworks	27
Figura 1-2:	Conexión Arduino – micro SD	30
Figura 2-2:	Circuito esquemático Arduino - micro SD	31
Figura 3-2:	Conexión de pines para modulo micro SD	32
Figura 4-2:	Conexión Arduino – módulo SIM 800L	32
Figura 5-2:	Circuito esquemático del módulo GSM.....	33
Figura 6-2:	Conexión Arduino – pantalla LCD.	34
Figura 7-2:	Conexión de la interfaz I2c	34
Figura 8-2:	Circuito esquemático Arduino - pantalla LCD	35

Figura 9-2:	Conexión Arduino - sistema OBDII	36
Figura 10-2:	Circuito esquemático Arduino - sistema OBDII.....	37
Figura 11-2:	Circuito esquemático de la tarjeta electrónica	38
Figura 12-2:	Diseño PCB del circuito electrónico	40
Figura 13-2:	Visualización 3D del circuito electrónico	41
Figura 14-2:	Proceso de corte de la placa PCB.....	41
Figura 15-2:	Diseño final de la placa PCB	42
Figura 16-2:	Implementación de elementos electrónicos	42
Figura 17-2:	Arduino en la placa PCB.....	43
Figura 18-2:	Proceso de impresión de la caja de protección.	43
Figura 19-2:	Construcción final de la caja de protección	44
Figura 20-2:	Ubicación de los módulos.....	44
Figura 21-2:	Prototipo Final	45
Figura 22-2:	Pantalla principal del prototipo	47
Figura 23-2:	Leds indicadores	48
Figura 24-2:	Puerto USB de programación Arduino	48
Figura 25-2:	Puerto de alimentación en el prototipo	49
Figura 26-2:	Ranura para micro SD.....	49
Figura 27-2:	Puerto OBDII integrado en el prototipo.....	50
Figura 1-3:	Posición de conexión a la unidad de control electrónica de motor	51
Figura 2-3:	Funcionamiento del prototipo.....	52
Figura 3-3:	Datos almacenados en datalogger	53
Figura 4-3:	Apertura de Blog de notas en Excel.....	54
Figura 5-3:	Delimitación de caracteres.....	54
Figura 6-3:	Selección de tabulación y coma	55
Figura 7-3:	Formato de los datos de columnas en general.....	55
Figura 8-3:	Datalogger final	56
Figura 9-3:	Proceso de los datos por la red.....	56
Figura 10-3:	Visualización de datos del sensor RPM	57
Figura 11-3:	Visualización de datos del sensor temperatura de refrigerante	61
Figura 12-3:	Visualización de datos del sensor temperatura de aceite	65
Figura 13-3:	Visualización de datos del sensor MAF.....	73
Figura 14-3:	Visualización de datos del sensor MAP.....	77
Figura 15-3:	Visualización de datos del sensor de velocidad	80
Figura 16-3:	Visualización de la distancia recorrida	84
Figura 17-3:	Visualización del dato en el sensor posición acelerado	88
Figura 18-3:	Visualización de datos del sensor temperatura ambiente.....	93

Figura 19-3:	Pruebas de consumo del prototipo	98
Figura 20-3:	Datos emitidos por la ECU	100
Figura 21-3:	Datos del sensor RPM obtenidos por el scanner	100
Figura 22-3:	Datos del sensor RPM Obtenidos mediante OBDII.....	101
Figura 23-3:	Datos del sensor MAP obtenidos por el scanner.....	102
Figura 24-3:	Datos del sensor MAP Obtenidos mediante OBDII	102
Figura 25-3:	Datos del sensor temperatura de refrigerantes obtenidos por el scanner	103
Figura 26-3:	Datos del sensor temperatura de refrigerante obtenidos mediante OBDII.....	103

INDICE DE GRÁFICOS

Gráfico 1-2:	Etapas del diseño del prototipo.....	28
Gráfico 2-2:	Estructura de la tarjeta electrónica del prototipo	29
Gráfico 3-2:	Diagrama de flujo referente al algoritmo de programación del prototipo	45
Gráfico 1-3:	Etiqueta en eje Y rangos de 600 a 1800 RPM.....	58
Gráfico 2-3:	Sensor RPM en Thingspeak	60
Gráfico 3-3:	Etiqueta en eje Y con temperatura del refrigerante entre 85°C a 105°C.....	62
Gráfico 4-3:	Sensor Temperatura de refrigerante en Thingspeak	64
Gráfico 5-3:	Etiqueta en eje Y con temperatura del aceite entre 90°C a 94°C	66
Gráfico 6-3:	Sensor Temperatura de Aceite en Thingspeak	68
Gráfico 7-3:	Visualización de datos del sensor temperatura de admisión	69
Gráfico 8-3:	Etiqueta en eje Y con temperatura de admisión entre 70°C a 80°C	70
Gráfico 9-3:	Sensor temperatura de admisión en Thingspeak	72
Gráfico 10-3:	Etiqueta en eje Y con datos obtenidos por el MAF de 0g a 15g	74
Gráfico 11-3:	Sensor MAF en Thingspeak.....	76
Gráfico 12-3:	Etiqueta en eje Y con datos obtenidos por el MAP de 75kpa	78
Gráfico 13-3:	Sensor MAP en Thingspeak.....	80
Gráfico 14-3:	Etiqueta en eje Y con datos obtenidos de velocidad de 10km a 140km.....	82
Gráfico 15-3:	Sensor de velocidad en Thingspeak	84
Gráfico 16-3:	Etiqueta en eje Y con datos obtenidos de distancia de 449km.....	86
Gráfico 17-3:	Sensor de distancia en Thingspeak.....	88
Gráfico 18-3:	Etiqueta en eje Y con datos obtenidos del sensor posición acelerador.....	90
Gráfico 19-3:	Sensor de posición acelerador en Thingspeak.....	92
Gráfico 20-3:	Etiqueta en eje Y con datos obtenidos de la temperatura de 27°C a 28°C	94
Gráfico 21-3:	Sensor de temperatura ambiente en Thingspeak	96

INDICE DE ANEXOS

Anexo A: Planos de la caja protectora

Anexo B: Código de programación Arduino para el funcionamiento del prototipo

RESUMEN

Se implementó un prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistema OBD II. Los dispositivos que componen su construcción fueron adquiridos mediante un análisis, estudio y requerimientos que permitan realizar las diferentes funciones propuestas por dicho prototipo, para ello se consideró el costo de cada uno de ellos, facilidad de adquisición y compatibilidad de software. Se utilizó un sistema denominado OBDII encargado del diagnóstico del vehículo, permitiendo un monitoreo constante y almacenamiento de datos para detección de fallas que puedan presentar y afectar el funcionamiento de los sensores del motor; esto mediante un módulo micro SD. Se fabricó una placa de control a la cual se le integró una tarjeta de desarrollo Arduino Mega encargada de controlar las funciones mediante algoritmos de programación y librerías para cada módulo y de manera directa con el sistema OBDII. El prototipo permite visualizar los datos obtenidos mediante dos plataformas virtuales denominadas Datalogger y Thingspeak. El datalogger almacena los datos obtenidos y procesados por la unidad de control electrónica de motor (ECU) de cada sensor y la muestra mediante un blog de notas. La plataforma Thingspeak propia de Matlab muestra las señales en un entorno analítico del internet de las cosas (IoT) que permite visualizar, agregar y analizar las señales de los sensores del motor directo en una nube de internet debido al módulo GSM 1800L. Las pruebas de funcionamiento fueron realizadas en un vehículo Híbrido Toyota Camry modelo 2017, se lograron extraer datos de los principales sensores del vehículo tales como: revoluciones por minuto (RPM), refrigerante, aceite, velocidad, distancia, entre otros; se logró evidenciar mediante análisis de error absoluto y relativo de los datos obtenidos un resultado entre 1% a 2% de error considerando que mediante este análisis los datos extraídos son aceptables.

Palabras claves: <TELECOMUNICACIONES>, <DISPOSITIVO TECNOLÓGICO>, <DIAGNÓSTICO DE VEHÍCULO>, <SISTEMA OBDII>, <SENSORES AUTOMOTRICES>, <UNIDAD ELECTRÓNICA DE CONTROL>, <DATALOGGER (SOFTWARE)>, <THINGSPEAK (SOFTWARE)>.



FIRMA ELECTRÓNICA por:
ELIZABETH
FERNANDA AREVALO
MEDINA



0570-DBRAI-UPT-2021

ABSTRACT

An electronic prototype was implemented for telematic registers and engine failure detection through the OBD II system. The devices that make up its construction were acquired through an analysis, study, and requirements that allow the different functions proposed by this prototype. Each device's cost, ease of acquisition, and software compatibility were considered to carry out this prototype. The OBDII system was in charge of diagnosing the vehicle, allowing constant monitoring and data storage to detect faults that may appear and affect the engine operation sensors; this monitoring was developed using a micro SD module. A control board was manufactured where an Arduino Mega development board was integrated, which was in charge of controlling the functions through programming algorithms and libraries for each module and directly with the OBDII system. The prototype allows viewing the data obtained through two virtual platforms called Datalogger and Thingspeak. The Datalogger stores the data obtained and processed by the electronic engine control unit (ECU) of each sensor and displays it through a blog of notes. Matlab's Thingspeak platform shows the signals in an analytical environment of the Internet of Things (IoT), which allows to visualize, add and analyze the engine sensors' signals directly in an internet cloud due to the GSM 1800L module. The test runs were carried out on a 2017 Toyota Camry Hybrid vehicle. It was possible to extract data from the vehicle's primary sensors, such as revolutions per minute (RPM), coolant, oil, speed, distance, among others. Besides, it was possible to show using absolute and relative error analysis of the data obtained a result between 1% to 2% of error, considering that the extracted data are acceptable through this analysis.

Keywords: <TELECOMMUNICATIONS>, <TECHNOLOGICAL DEVICE>, <VEHICLE DIAGNOSTIC>, <OBD II SYSTEM>, <AUTOMOTIVE SENSORS>, <ELECTRONIC CONTROL UNIT (ECU)>, <DATALOGGER (SOFTWARE) >, <THINGSPEAK (SOFTWARE) >.

INTRODUCCIÓN

ANTECEDENTES

Con el desarrollo tecnológico a nivel mundial, específicamente en el campo de la electrónica como la base fundamental en el mejoramiento y aporte a los demás campos de la ingeniería entre las más comunes: industrial, automotriz y mecánica. Donde se han dado soluciones a un sinnúmero de problemas presentes diariamente a nivel mundial; con respecto al campo automotriz y la evolución de sus sistemas manuales a la emigración de sistemas electrónicos por medio del uso e implementación de módulos y sensores, han ofrecido facilidades al momento de análisis de fallas y errores en un vehículo automotriz.

Ambrosio & Sánchez (2017, p. 1), afirma que el desarrollo automotriz se ha centrado en la implementación de materiales ligeros, miniaturización, inteligencia, movilidad y energía, para ello su función con sistemas electrónicos mediante módulos y algoritmos de programación, es por ello que la electrónica dentro del área automotriz toma una gran importancia en la creación de tecnologías en los vehículos; esta evolución tecnológica cada vez sustituyen los sistemas mecánicos a electrónicos, un claro ejemplo es la interacción de una memoria o ECU (unidad de control electrónica de motor) la misma que establece un control completo a lo sensores instalados en el vehículo.

Según una investigación de la revista Motor (2017, p. 5), no se puede negar que la electrónica está suponiendo un avance tecnológico sin precedentes, posiblemente en los futuros libros de historia se le dé la misma importancia, si no más que, por ejemplo a la máquina de vapor y su revolución industrial, dada la intensa influencia que está teniendo en nuestro modo de vida, los sistemas electrónicos sustituyen complejos sistemas mecánicos por elementos más precisos, más pequeños y generalmente más económicos. Incluso gracias a la electrónica es posible crear sistemas capaces de realizar trabajos de tal complejidad que ni siquiera podrían existir en versión “analógica”.

En la actualidad existe un simulador de señal para sensores automotrices MST- 9000, la cual es una herramienta muy útil para el técnico profesional durante la reparación de la ECU del automóvil y la programación. Esta herramienta de simulación de señal de sensor de automóvil es una herramienta de reparación de ECU de automóvil indispensable, con una actualización reciente y una función más estable. Después de muchos años de I + D y tiempos de actualizaciones, ahora nuestro experto en simulación de sensores ha sido el equipo que más ha solucionado los problemas

con la tecnología avanzada para muchos talleres y escuelas de capacitación automovilística (Orozco, 2015).

El dispositivo ECU 22, es un instrumento que permite probar el funcionamiento y fallas en una computadora automotriz fuera del vehículo, es decir, es necesario que la computadora sea extraída del vehículo para su correcto análisis y conexión en el dispositivo, este equipo no contra de sistema OBD II para una conexión directa vehículo y dispositivo analizador (Orozco, 2015).

FORMULACIÓN DEL PROBLEMA

¿Existen dispositivos capaces de efectuar un análisis y diagnóstico de los sensores del motor de un automóvil para la detección de fallos mediante sistema OBD II?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cuáles son los fundamentos básicos de los sensores del motor que pueden ser analizados para detectar fallos?

¿Qué requerimientos debe cumplir el prototipo para la detección de fallos en el motor mediante sistemas OBD II?

¿Cuál es el software, hardware y diseño electrónico adecuado que permite implementar el diseño propuesto del prototipo?

¿Cómo evaluar si el prototipo electrónico implementado cumple con los requerimientos planteados?

JUSTIFICACIÓN TEORICA

Mediante este proyecto de grado se pretende desarrollar un prototipo electrónico conectado mediante un sistema OBD II para analizar y diagnosticar la unidad electrónica de control del motor, es decir, el funcionamiento y fallo de los sensores cuyos datos censados serán mostrados en un datalogger.

El estudiante Cando (2017, p. 15), de la Escuela Superior Politécnica desarrollo un simulador para el diagnóstico de la unidad de control electrónico de motor (ECU), mediante las diferentes tecnologías existentes en el país, dicho dispositivo consta de 4 etapas las cuales mediante un analizador examinan las fallas existentes en los sensores del motor, las muestras y señales de cada sensor se interpretan y visualizan mediante una pantalla HMI y un osciloscopio integrado al sistema.

El estudiante Tapia (2013, p. 135), de la Escuela Superior Politécnica implementó un sistema de transferencia de datos por telemetría empleado en un vehículo de competencia mediante comunicación bluetooth, la cual utilizando una aplicación denominada Mega Tune 2.25P3 se visualiza parámetros técnicos electrónicos y mecánicos en tiempo real, con el fin de llevar un monitoreo constante sobre el rendimiento del motor en diferentes fases de prueba tales como arranque y condiciones extremas; de los resultados emitidos el sistema puede modificar parámetros y variaciones mediante su tecnología bluetooth.

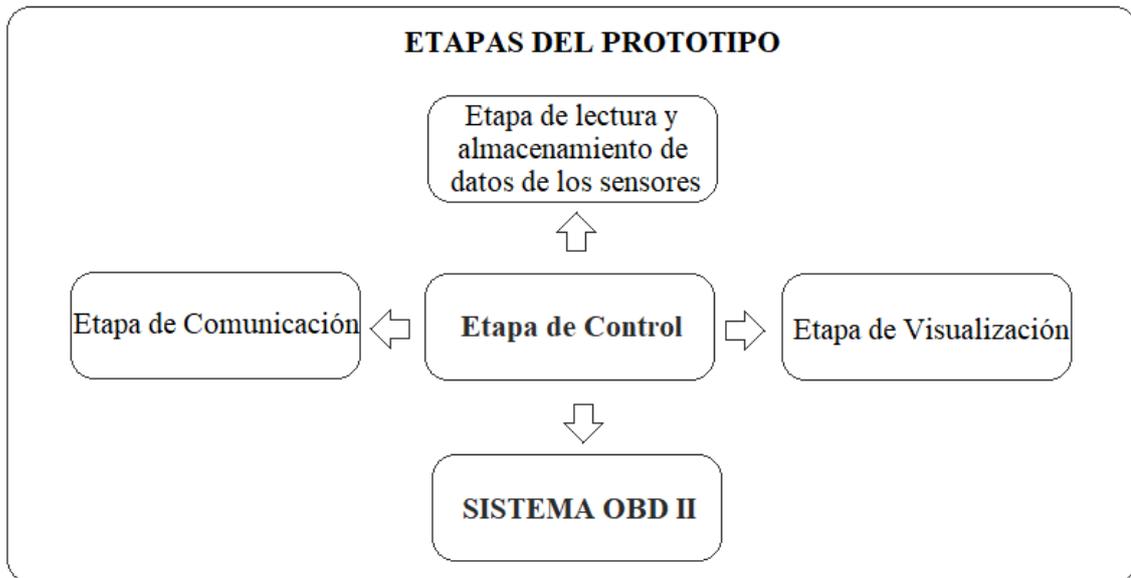
Según un estudio realizado con respecto a los fallos presentes en los sistemas electrónicos de los vehículos automotrices en general, se consideró que los establecimientos, concesionarios y centro especializados en su mayoría no constan con un sistema capaz de detectar fallos mediante registro telemático en tiempo real.

JUSTIFICACIÓN APLICATIVA

En la actualidad los analizadores de sistemas electrónicos en un vehículo empleados al diagnóstico de los diferentes sensores integrados al motor y cuyos datos enviados a la unidad de control electrónica, no constan de un análisis con registro telemático y en muchos de los casos la adquisición de datos no es reflejada y conservada en entornos de visualización tales como base de datos y datalogger.

Es por ello, que se pretende desarrollar e instalar un prototipo que pueda realizar un análisis constante del funcionamiento de los sensores automotrices y lleve un registro telemático para el usuario del vehículo o de los centros especializados mediante sistema OBD II, en caso de presentar alguna falla en las mediciones muestre una alerta para realizar la respectiva revisión. Los sensores a ser evaluados en este trabajo de investigación son: SP, VELOCIDAD, IMPULSOS, OXÍGENO, CMP, CKP, TPS, IAT, MAP, MAF.

Para el desarrollo del prototipo se establecen diferentes etapas que cumplen con los requerimientos necesarios para la implementación del proyecto, tal como se muestra en la siguiente figura 0-1:



La implementación del prototipo consta de 5 etapas de desarrollo las cuales son:

- **Etapa de control:** Es la encargada de comandar el funcionamiento del prototipo mediante un microcontrolador y algoritmos de programación establecidos para su funcionamiento
- **Etapa de visualización:** Se encarga de mostrar los datos obtenidos de los sensores instalados en el motor y fallas.
- **Etapa de comunicación:** Mediante un módulo GSM permite comunicar los resultados del sistema a una plataforma de sitio web.
- **Etapa de Lectura y Almacenamiento de datos de los sensores:** Se encarga de almacenar los datos receptados por los sensores.
- **Sistema OBD II:** Es el sistema principal dentro del funcionamiento del prototipo, su función es realizar una conexión y comunicación con los sensores internos del motor.

OBJETIVOS

OBJETIVO GENERAL

Implementar un prototipo electrónico para registros telemáticos y detección de fallos en motores de automóviles mediante sistema OBD II.

OBJETIVO ESPECIFICO

- Conocer los fundamentos básicos de funcionamiento de los sensores del motor a ser analizados para detectar fallos en el vehículo
- Especificar los requerimientos que debe cumplir el prototipo para la adquisición telemétrica de datos mediante el sistema OBD II
- Determinar el software, hardware y diseño electrónico adecuados para el diseño propuesto del prototipo
- Realizar las evaluaciones del prototipo electrónico implementado para verificar el cumplimiento de los requerimientos planteados

CAPITULO 1

1 MARCO TEORICO REFERENCIAL

1.1 La electrónica automotriz

Ambrosio & Sánchez (2017, p. 5), en su publicación afirma que el crecimiento tecnológico de información y la electrónica de consumo ha tenido una gran demanda en la industria automotriz con la ventaja de aprovechar su tecnología la cual involucra el 30% del costo de producción de un vehículo relacionado con la electrónica del mismo, por ejemplo una unidad de control electrónica la cual ejerce la función del corazón del vehículo debido a sus múltiples funciones tales como: el control del combustible, sincronismo, sensores para cada parte del motor.

Hoy en día un vehículo típicamente está compuesto de 25 a 35 microcontroladores y en casos excepcionales un vehículo de lujo tiene un aproximado de 60 a 100 microcontroladores encargados de establecer un correcto sincronismo entre la parte mecánica y electrónica del auto por medio de algoritmos de programación (Ambrosio, y otros, 2017), la figura 1-1 a continuación muestra la composición de los sistemas electrónicos integrados en un vehículo.

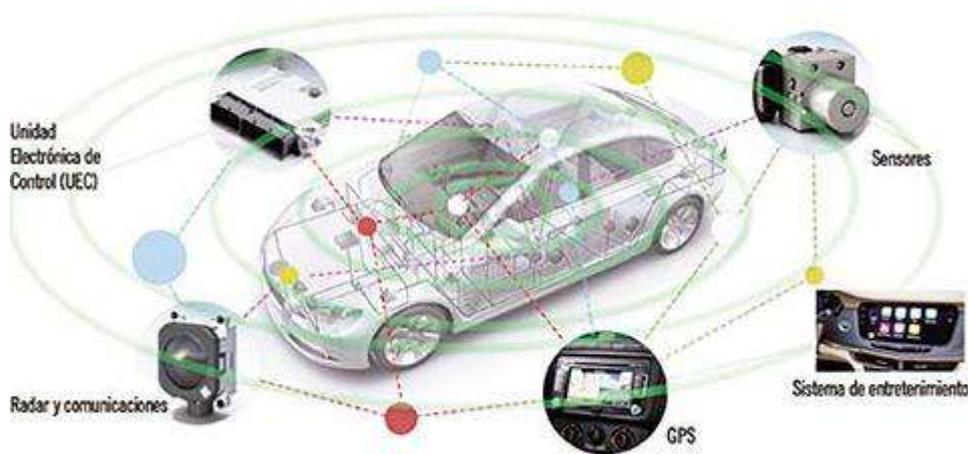


Figura 1-1: Sistemas electrónicos integrados a un vehículo
Fuente: Ambrosio & Sánchez, 2017)

1.2 Sensores Automotrices

Medina (2018, pp. 1 -5), concluye que el desarrollo tecnológico y la inclusión de la ingeniería electrónica en el campo automotriz permitieron que una importante cantidad de sensores ejerzan

una perfecta gestión entre la unidad de control y el motor del vehículo; el sensor también conocido como sonda de transmisión convierte una magnitud física, temperatura, revoluciones del motor, gases de escape, calidad de aire que generalmente no son señales eléctricas.

La señal eléctrica que establece como salida el sensor instalado en el vehículo no es considerada solamente como una señal de corriente o tensión, estos sensores son considerados como elementales, la parte que trata las señales generadas por el sensor se encarga en general de dar a las señales de los sensores la forma normalizada necesaria para ser interpretada por la unidad de control ECU (Medina, 2018); en la siguiente figura 2-1 se visualiza los tipos de sensores que comanda la unidad de control electrónica (ECU).

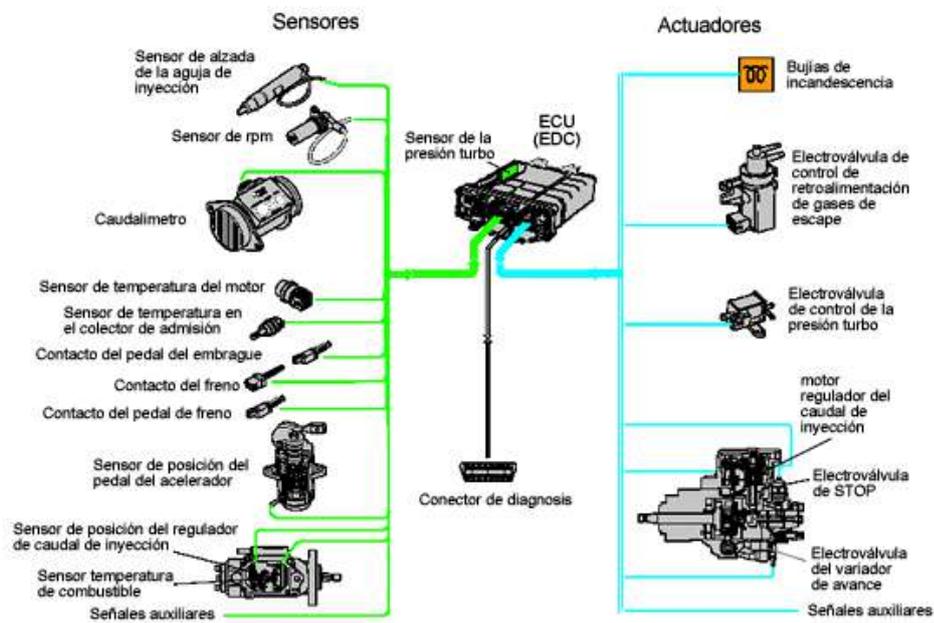


Figura 2-1: Sensores comandados por una ECU.

Fuente: (Medina, 2018)

1.3 Tipos de sensores Automotrices

El sistema electrónico del automóvil comandado por la unidad de control electrónica consta de sensores encargados de ejercer funciones en cada parte del sistema mecánico del vehículo, dentro de las cuales los sensores más comunes son:

1.3.1 Sensor ESP

Denominados como ESP, ESC, DSC, VDC, VSA y VSC son las siglas comerciales del control electrónico de estabilidad, uno de los avances tecnológicos en materia de seguridad activa más importantes de los últimos tiempos. Su función es conseguir que el vehículo se mantenga en la

trayectoria marcada por el conductor con el volante, reduciendo en buena medida los siniestros viales derivados de un derrape, la siguiente figura 3-1 muestra el componente físico de un sensor ESP. (Medina, 2018; P. 5).



Figura 3-1: Componente físico de un sensor ESP
Fuente: (Medina, 2018)

1.3.2 Sensor de velocidad

Los sensores ABS o también conocidos como sensores de revolución por minuto, este elemento están ubicados en el buje de las ruedas y detectan la velocidad de giro de cada una de ellas, transfiriendo esta información a la unidad de control del sistema ABS, si la unidad de control del sistema ABS detecta que hay un descenso importante de velocidad en una o varias ruedas que podría provocar que se bloquearan, interviene rápidamente modulando la presión de frenado de cada rueda individualmente; de este modo se evita que las ruedas se bloqueen y garantizan un frenado seguro permitiendo al conductor mantener el control del vehículo, la siguiente figura 4-1 muestra el componente físico de un sensor de velocidad. (Medina, 2018; P. 8).



Figura 4-1: Componente Físico de un sensor de velocidad
Fuente: (Medina, 2018)

1.3.3 Sensor de impulsos

Los Sensores de posición de cigüeñal o de árbol de Levas generalmente están ubicados sobre una rueda dentada y discos con ventanas sectorizadas, anillos de material ferromagnético, dichos sensores detectan las variaciones de flujo magnético que producen los dientes y valles de la rueda fónica o los relieves de la leva, convirtiendo esa lectura en una señal eléctrica que va directamente a la unidad de control.

En la figura 5-1 se muestra el componente físico de un sensor de impulsos. (Medina, 2018; P. 12).



Figura 5-1.: Componente físico de un sensor de impulsos
Fuente: (Medina, 2018)

1.3.4 Sensor de oxígeno

El sensor convierte la cantidad de oxígeno de los gases que produce el automóvil en señal eléctrica, la unidad de control electrónica toma esa señal y así se da cuenta si la mezcla de gasolina y aire está en un punto óptimo, si este resultado resulta opuesto el sensor toma medidas para hacerlo, en la figura 6-1 se muestra el componente físico de un sensor de oxígeno. (Medina, 2018; P. 13).



Figura 5-1: Componente físico de un sensor de oxígeno
Fuente: (Medina, 2018)

1.3.5 *Sensor CMP*

Se le denomina como sensor de posición de árbol de levas a través de una señal pulsada enviada por el sensor, la unidad de comando calcula la velocidad de rotación del eje donde está instalado; este tipo de sensor es efecto hall compuesto por una pequeña pastilla de material semiconductor circularada por una corriente continua. La resistencia eléctrica de la pastilla es sensible a la presencia de campo magnético, en la figura 7-1 se muestra el componente físico de un sensor CMP. (Thomson, 2017).



Figura 6-1: Componente físico de un sensor CMP
Fuente: (Thomson, 2017)

1.3.6 *Sensor CKP*

El Sensor CKP determina la posición exacta del cigüeñal, en el caso del sensor de posición del árbol de levas que controla la velocidad de rotación del árbol de levas, indicando la posición de la válvula en el ciclo de combustión, el sensor CKP determina la posición del cigüeñal para controlar la velocidad de rotación del cigüeñal y así saber la posición del pistón en punto muerto superior, la siguiente figura 8-1 muestra el componente físico de un sensor CKP. (Auto, 2019; p. 8).



Figura 7-1: Componente físico de un sensor CKP
Fuente: (Auto, 2019)

1.3.7 *Sensor TPS*

Conocido como sensor de posición del acelerador, está montado en el cuerpo de aceleración y convierte el ángulo del papalote del cuerpo de aceleración en una señal eléctrica, A medida que el papalote se abre, el voltaje de la señal se incrementa, en la figura 9-1 se muestra el componente físico de un sensor TPS. (Strauss, 2017; p. 2).



Figura 8-1: Componente físico de un sensor TPS
Fuente: (Strauss, 2017)

1.3.8 *Sensor IAT*

El sensor IAT posee un resistencia que varía según la temperatura de entrada al motor de acuerdo a esto el sensor de temperatura de aire de entrada puede ayudar a realizar los ajuste necesarios en la mezcla por medio de señales que le envía a la computadora, dicho componente es un dispositivo termistor su resistencia cuando el aire tiene una temperatura de 10 °C es equivalente a 100ohmios, cuando el aire está a 130 °C entonces la resistencia es de más o menos 70 Ohmios; esto indica que el aumento de la temperatura implica una disminución de la resistencia, la figura 10-1 muestra el componente físico de un sensor IAT. (Mundo, 2017).



Figura 9-1: Componente físico de un sensor IAT
Fuente: (Mundo, 2017)

1.3.9 Sensor MAP

El sensor de presión absoluta múltiple se utiliza para medir la presión en el colector de admisión cuando se envía este valor de salida de vuelta al sistema de gestión del motor, puede determinar el abastecimiento o la cantidad de avance de la sincronización de encendido del vacío; en la mayoría de los casos el sensor de MAP medirá una presión negativa, pero también puede utilizarse para medir una presión positiva en un vehículo equipado con un turbo compresor, la figura 11-1 muestra el componente físico de un sensor MAP. (Pico, 2015).



Figura 10-1: Componente físico de un sensor MAP
Fuente: (Pico, 2015)

1.3.10 Sensor MAF

El sensor MAF está diseñado para medir el flujo de aire que ingresa al motor, este dato viaja hasta la unidad electrónica de control por medio de un cable el cual envía una señal de voltaje que cambia de acuerdo al flujo; este sensor internamente funciona mediante un hilo muy fino metálico el cual se encuentra a muy alta temperatura, en el momento que comienza a entrar aire el aire enfría este hilo y las cargas cambiantes de aire causan un efecto diferente sobre la temperatura del hilo, entonces todo el circuito que maneja el tema del calentamiento del hilo generara una señal de voltaje de acuerdo a que tanto es enfriado, la figura 12-1 muestra el componente físico de un sensor MAF. (Cise, 2017).



Figura 11-1: Componente físico de un sensor MAF
Fuente: (Cise, 2017)

1.4 Almacenamiento de datos de los sensores

Para un almacenamiento de datos suministrado por los sensores automotrices se utiliza herramientas digitales que permitan una interacción digital con el usuario, al momento de interpretar los datos adquiridos.

1.4.1 Datalogger

La estudiante Quito, (2015) de la Escuela Politécnica Nacional, en su trabajo de titulación afirma que un datalogger es un sistema diseñado para registrar datos utilizado en su mayoría por empresas o industrias, este tipo de sistemas permite tener datos históricos importantes para realizar proyecciones y predicciones en base a todos los datos adquiridos y en busca de una solución; un monitoreo realizado de manera diaria, semanal o mensual de diferentes datos o variables mejora el análisis de máquinas o dispositivos y previene fallos comunes, la figura 13-1 muestra un entorno de visualización de un datalogger. (Quito, 2015).



Figura 12-1: Entorno de visualización de un Datalogger
Fuente: (Quito, 2015)

1.4.2 Thingspeak

La plataforma Thingspeak por datos de internet es un entorno analítico de I o T que permite visualizar, agregar y analizar transmisiones de datos directo en una nube, es posible extraer los datos un canal designado por Thingspeak para mostrarse en un entorno Matlab, la figura 14-1 muestra un entorno de visualización de Thingspeak. (MathWorks, 2018).

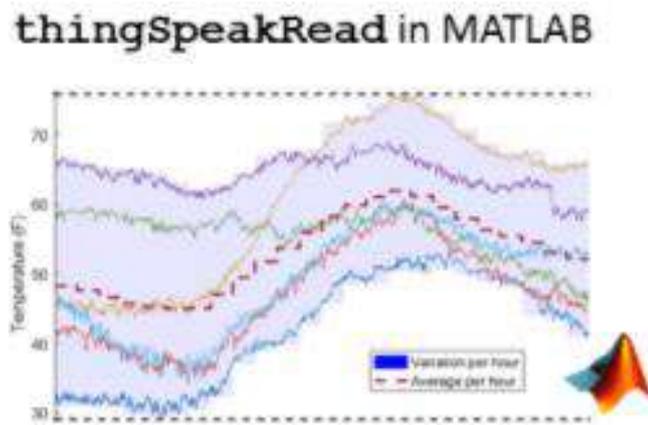


Figura 13-1: Entorno de visualización de Thingspeak
Fuente: (MathWorks, 2018)

1.5 Etapa de Control

La etapa de control es la parte que comanda el diseño electrónico en diferentes proyectos innovadores, quienes cumplen aquella función son las tarjetas de desarrollo, las mismas que

aportan gran facilidad para el desarrollo de diferentes proyectos técnicos y tecnológicos por su fácil accesibilidad y software libre constituidos por sistemas digitales y analógicos; dentro de las más utilizadas tenemos:

- Arduino
- Raspberry Pi

1.5.1 Tarjeta de desarrollo

Las tarjetas de desarrollo electrónica fueron creadas con el fin de facilitar la creación y desarrollo de diferentes tipos de prototipos y proyectos, que necesitan obtener mayor rapidez al momento de ejecutar las actividades designadas. Los componentes principales de estas tarjetas son los microcontroladores y micro procesadores en una sola placa.

Al momento de elegir una tarjeta desarrollo electrónica, se debe tomar en cuenta el procesador, simplicidad, precio, velocidad de entrega y facilidad al momento de su programación. La principal característica que debe tener la tarjeta a utilizar en la comunicación GSM y el procesamiento de datos digitales. Las tarjetas de desarrollo electrónica que cumplen con estas especificaciones son:

1.5.1.1 Arduino

La tarjeta Arduino posee un procesador Atmega16U2, posee periféricos, protocolos de comunicación, comparación e interrupciones. Actualmente posee una gran variedad de dispositivos compatibles con estas tarjetas y su costo de bajo en comparación con otros tipos de tarjetas de desarrollo. En la Figura 15-1 se muestra un Arduino Mega.

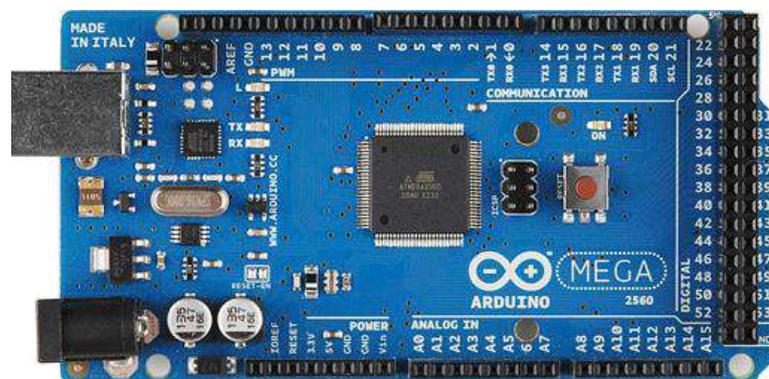


Figura 14-1: Arduino Mega
Fuente: (Arduino, 2017)

1.5.1.2 Tipos de tarjetas de desarrollo Arduino

En la actualidad existen varios tipos de tarjetas de desarrollo Arduino, las cuales se diferencian por el número de pines con E/S, la memoria utilizada y la funcionalidad que este va a tener. Las más conocidas son:

- Arduino Mega 2560
- Arduino Ethernet
- Arduino Fio
- Arduino Nano
- Arduino Pro Mini
- Arduino Uno
- Arduino Leonardo

Ventajas del uso de la tarjeta de desarrollo Arduino

Las principales ventajas del uso de Arduino son:

- Permite simplificar el proceso del trabajo por medio de microcontroladores
- Las placas de desarrollo Arduino son más económicos en comparación a otras tarjetas
- El software de Arduino es compatible con diferentes sistemas operativos
- Al ser un software libre existen muchos códigos abiertos
- La programación es sencillo y fácil, ya que está basado en el entorno de programación procesing

En la Tabla 1-1, se muestran las principales características de Arduino Mega.

Tabla 1-1: Características generales del Arduino Mega

Microcontrolador	ATmega2560
Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB se usan por el bootloader
SRAM	8 KB

EEPROM	4 KB
Velocidad del reloj	16 MHz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Fuente: (Andrade, 2018)

Realizado por: Montesdeoca, Andrea, 2020

1.5.1.3 Raspberry pi

La tarjeta de desarrollo electrónica Raspberry Pi posee un CPU ARM® C rtex™ A53 de cuatro n cleos de 1,2 GHz y 64 bits, incluye una tarjeta LAN inal mbrica 802.11n integrada y capacidad Bluetooth 4.1, posee 4 puertos USB los cuales son para conectar teclado y mouse, conector HMI. Este tipo de tarjetas necesitan un almacenamiento extra, es decir una tarjeta micro SD. Su costo es m s alto en comparaci n a la tarjeta Arduino. (Redrob n, 2018)

En la Figura 16-1 se muestra una tarjeta de desarrollo Raspberry Pi.



Figura 15-1: Tarjeta de desarrollo Raspberry Pi

Fuente: (Michael, 2016)

1.6 Pantalla de Cristal L quido (LCD)

Las pantallas de cristal l quido (LCD) son dispositivos electr nicos de visualizaci n, en los cuales se puede mostrar informaci n, caracteres o figuras de manera clara dependiendo la aplicaci n que se lo va a dar, este dispositivo permite la comunicaci n m quina humano, ya que este puede mostrar mensajes de funcionamiento de las mismas, la LCD se muestra en la figura 17-1.

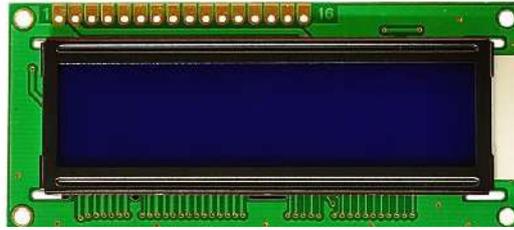


Figura 16-1: Pantalla de Cristal Líquido de 2x16
Fuente: (Xukyo, 2020)

Para el desarrollo de este prototipo se elige un lcd ya que en ellos se puede visualizar mayor número de caracteres con claridad y se optimiza el número de pines de conexión. Para la programación existe una librería específica para el uso de lcd.

1.6.1 Memorias del LCD

Las pantallas de cristal líquido cuentan con dos memorias internas que son: la memoria DDRAM y la memoria CGROM.

1.6.2 Memoria DDRAM

La memoria DDRAM (Data Display Ram) es la encargada de almacenar los datos que se van a visualizar en la pantalla de cristal líquido siendo estos caracteres, símbolos o figuras con sus respectivas posiciones.

En la figura 18 -1se muestra los caracteres visualizados en la pantalla usando memoria DDRAM.



Figura 17-1: Memoria DDRAM del LCD
Fuente: (Xukyo, 2020)

1.6.3 Memoria CGROM y CGRAM

La memoria CGROM es la memoria donde se almacena una tabla con los posibles caracteres que se pueden visualizar en la pantalla de cristal líquido, y la memoria CGRAM (Character generator Ram) es en la que puede almacenar datos propios, como se muestra en la figura 19-1. (SALAS, 2013)

		4 higher bits of address															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4 lower bits of address	XXXX 0000	CG RAM	(1)	0	0	P	`	P		-	3	0	p				
	XXXX 0001	CG RAM	(2)	!	1	A	Q	a	a		.	7	4	á	q		
	XXXX 0010	CG RAM	(3)	"	2	B	R	b	r		"	í	ú	z	p	ø	
	XXXX 0011	CG RAM	(4)	#	3	C	S	c	s		」	ウ	て	ε	ε	ω	
	XXXX 0100	CG RAM	(5)	\$	4	D	T	d	t		\	И	т	μ	α		
	XXXX 0101	CG RAM	(6)	%	5	E	U	e	u		.	オ	ナ	い	σ	ü	
	XXXX 0110	CG RAM	(7)	&	6	F	V	f	v		ヲ	カ	ニ	ヨ	ρ	Σ	
	XXXX 0111	CG RAM	(8)	'	7	G	W	g	w		ア	キ	ア	ウ	g	π	
	XXXX 1000	CG RAM	(9)	(8	H	X	h	x		イ	ウ	キ	ウ	ア	Σ	
	XXXX 1001	CG RAM	(10))	9	I	Y	i	y		ウ	ク	ル	ウ	γ		
	XXXX 1010	CG RAM	(11)	*	:	J	Z	j	z		エ	コ	ル	レ	キ		
	XXXX 1011	CG RAM	(12)	+	:	K	C	k	c		オ	サ	ヒ	ロ	ア	キ	
	XXXX 1100	CG RAM	(13)	<	<	L	#	l	#		ト	シ	フ	ワ	キ	π	
	XXXX 1101	CG RAM	(14)	=	=	M]	m)		ユ	ス	ン	ト	キ		
	XXXX 1110	CG RAM	(15)	.	.	N	^	n	^		ヨ	セ	ホ	ア	キ		
	XXXX 1111	CG RAM	(16)	/	/	0	_	o	_		ウ	ツ	マ	ア	キ		

Figura 18-1: Memoria CGROM de LCD
Fuente: (Beltran, 2015)

1.7 Comunicación

Existen diversos tipos de comunicación empleados en un sinnúmero de proyectos, cuya facilidad de interacción permite efectuar funciones a cortas y largas distancia entre emisor receptor, para ello se detallan dos tipos:

- Comunicación alámbrica
- Comunicación inalámbrica

1.7.1 Comunicación alámbrica

Este tipo de comunicación es conocida por la utilización de cables comúnmente de cobre, el cual une al emisor y receptor y la transmisión de transmite mediante impulsos eléctricos, este tipo de comunicación tiene sus ventajas y desventajas, en la siguiente figura 20-1 se muestra un entorno de conexión alámbrica.

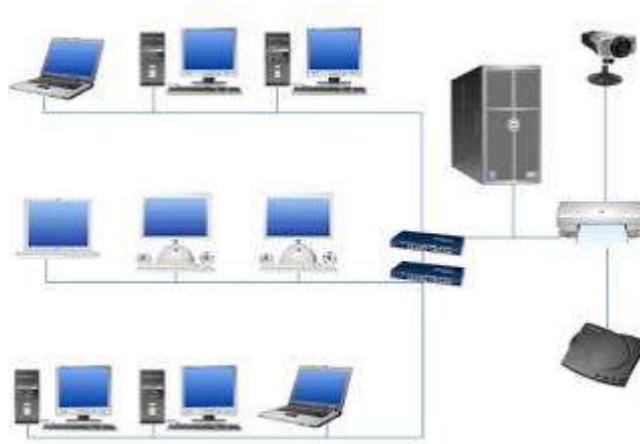


Figura 19-1: Entorno de conexión alámbrica

Fuente: (Alámbrica, 2015)

1.7.1.1 Ventajas de la comunicación alámbrica

- Costo más elevado.
- Máximo rendimiento posible.
- Excelente capacidad.
- Mayor velocidad.
- Cable de Ethernet estándar de 100Mbps.

1.7.1.2 Desventajas de la comunicación alámbrica

- Ocupa muchos accesorios de conexión.
- El acceso físico es un problema común.
- Menos estético.

1.7.2 Comunicación inalámbrica

Este tipo de comunicación es reconocida por tener su propio espacio, se entiende por red inalámbrica al conjunto interconectado de ordenadores autónomos capaces de intercambiar archivos, compartir información y periféricos entre sí mediante soluciones que no requieren el uso de cables de interconexión, la figura 21-1 muestra un ejemplo de comunicación alámbrica (Alámbrica, 2015).

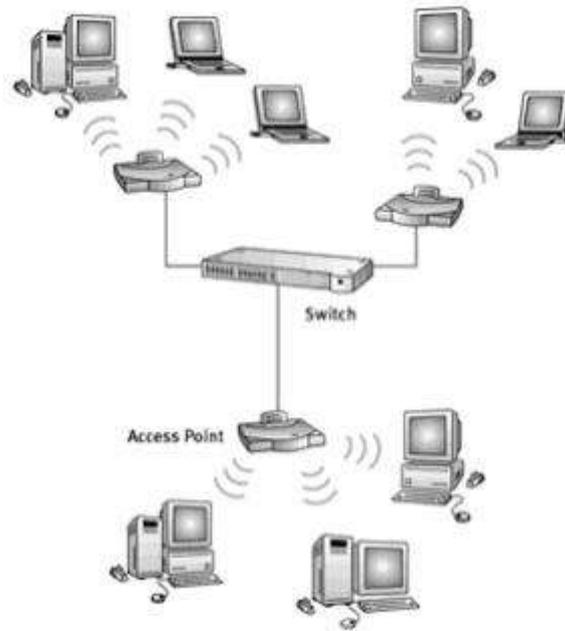


Figura 20-1: Protocolo de comunicación Wifi.
Fuente: (Alámbrica, 2015)

1.7.2.1 Ventajas del protocolo de comunicación Wifi.

- Cobertura dentro de la zona instalada
- Poca planificación
- Evita el cableado
- Robustez

1.7.2.2 Desventajas del protocolo de comunicación Wifi.

- Calidad
- Límite de distancia de transmisión
- Costo
- Incertidumbre tecnológica

1.7.3 Módulos de comunicación

Existen módulos que permiten realizar una interacción entre emisor y receptor de manera inalámbrica, dentro de los más utilizados está el siguiente modulo:

1.7.3.1 Módulo GSM 1800L

El módulo de telefonía móvil que trabaja a 850, 900, 1800 y 1900 MHz, también permite añadir voz, SMS, y dato en los proyectos desarrollados con Arduino, ya que se puede conectar un teléfono móvil a al proyecto. En la Figura 22-1 se muestra un módulo GSM 1800L.



Figura 21-1: Módulo GSM 1800L
Fuente: (AMAZON, 2018)

1.7.3.2 Especificaciones técnicas del Módulo GSM 1800L

Las principales especificaciones técnicas del módulo GSM1800L son:

- Voltaje de Operación: 3.4V - 4.4V DC
- Nivel Lógico de 3V a 5V
- Consumo de corriente (máx): 500 mA
- Consumo de corriente (modo de reposo): 0.7 mA
- Interfaz: Serial UART
- Quad-band 850/900/1800/1900MHz – se conectan a cualquier red mundial
- GSM con cualquier SIM 2G
- Trabaja solo con tecnología 2G
- Hacer y recibir llamadas de voz usando un auricular o un altavoz de 8Ω externo y un micrófono electret.
- Enviar y recibir mensajes SMS
- Enviar y recibir datos GPRS (TCP/IP, HTTP, etc.)
- Escanear y recibir emisiones de radio FM
- Controlado por Comandos AT
- Interfaz de comandos AT con detección “automática” de velocidad de transmisión
- Soporta A-GPS

- Velocidad máxima de transmisión 85.6 Kbps
- Protocolo TCP/IP en chip
- Codificación: CS-1, CS-2, CS-3 y CS-4
- Soporta USSD
- Soporta reloj en tiempo real (RTC)
- Velocidades de transmisión serial desde 1200bps hasta 115 200 bps
- Tamaño de la SIM: Micro SIM (ELECTROPRO, 2019)

1.8 Módulo lector de memoria Micro SD

El módulo lector de memoria Micro SD es una tarjeta de interfaz que fue diseñada con el fin de poder acceder a una memoria micro SD en modo SPI, donde se etiqueta las señales de control en el bus antes misionado. Internamente posee un circuito de conversión de voltaje que permite la comunicación a 3.3V o 5V. En la Figura 23-1 se muestra el módulo lector de memoria Micro SD. (Eneka, 2018)

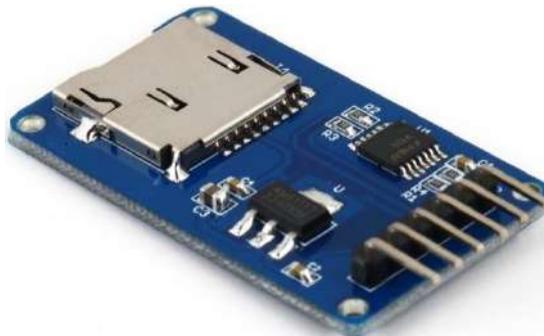


Figura 22-1: Módulo lector de memoria Micro SD
Fuente: (Eneka, 2018)

1.8.1 Especificaciones técnicas del módulo lector de memoria Micro SD

Las principales especificaciones técnicas de estos módulos son:

- Voltaje de Operación: 3.3V-5V
- Interfaz: SPI
- Cuenta con todos los pines SPI de la tarjeta SD: MOSI, MISO, SCK, CS
- Permite almacenar grandes cantidades de datos en memorias SD utilizando Arduino. (Eneka, 2018)

1.9 Sistema OBD - II

Este sistema fue requerido por la industria automotriz para ayudar a los técnicos en el diagnóstico de vehículos ya que los sistemas que existían hasta ese entonces no facilitaban dicho proceso, su función consiste en un sistema de diagnóstico a bordo del vehículo y el cual permite monitorear el sistema y almacenar las diferentes fallas que se puedan presentar en el funcionamiento del motor, los componentes dentro del sistema son verificados en su funcionalidad; aunque la falla es detectada los efectos que puede ocasionar en el funcionamiento global del sistema de operación no son tomados en cuenta (Globaltech, 2019). En la figura 24-1 se muestra el conector ODB-II



Figura 23-1: Conector OBD - II
Fuente: (Globaltech, 2019)

Los sistemas OBD II de diagnóstico a bordo, permite detectar averías en el vehículo, este sistema consta de una toma exterior en algún lugar predeterminado del vehículo generalmente en la zona de fusibles, este sistema se conecta mediante un cable el cual extrae la información necesaria suministrada por la unidad de control electrónica de motor; este sistema realiza funciones tales como:

- Recogida de datos
- Transmisión de datos
- Análisis de datos (Motor, 2017).

1.9.1 Compatibilidad

Para el uso de este módulo es importante que se verifique la certificación expuesta en el vehículo, este puerto viene integrado generalmente debajo de la columna de dirección o ligeramente a la

izquierda de la misma, la certificación OBDII, la figura 25-1 muestra la certificación OBDII de cada vehículo.



Figura 24-1: Certificación OBDII de cada vehículo.
Fuente: (Freematics, 2018)

1.9.2 Conector y comunicación

El tipo de comunicación interno esta denominado CAN compatible con la mayoría de los vehículos que en su interior sean comandados por la unidad de control electrónica de motor dentro de las más comunes tenemos:

- CAN 500Kbps/29bit.
- CAN 500Kbps/11bit.
- CAN 250Kbps/29bit.
- CAN 250Kbps/11bit.
- KWP2000 Fast.
- KWP2000 5Kbps.
- ISO9141-2 (Partial).

El tipo de conector es un cable desenchufable de 2 y 4 terminales incluida la alimentación VCC/GND y líneas de transmisión RC/TX, su compatibilidad con Arduino mediante pines de ruptura permite facilidad de conexión acorde a cada diseño por parte del usuario. (Freematics, 2018)

1.9.2.1 Características

Conector de alimentación (2-pin 2.54 Dupont conector)

- Rojo: VCC (Conectado a VCC de Arduino)
- Negro: GND (Conectado a GND de Arduino)

Conector de datos serie UART (Conector Dupont de 2 pines y 2,54)

- White: Rx (Conectado a transmisión serial de Arduino)
- Green: Tx (Conectado al RX serial de Arduino). (Freematics, 2018)

1.10 Software de diseño CAD

Existen un sinnúmero de programas que permiten realizar diferentes diseños de prototipos o productos terminados en un entorno 3D, en la cual se puede verificar sus dimensiones y formas; dentro de los más completos y conocidos está el software de diseño Solidworks.

1.10.1 Solidworks

Denominado como un software de diseño CAD 3D asistido por computadora, encargado de diseñar, modelar, ensamblar piezas y planos en 2D y 3D brindando soluciones en el proceso de desarrollo del producto; este programa está constituido por un conjunto de que permite la creación e innovación de procesos.

“En 1993, el fundador de SOLIDWORKS, Jon Hirschtick, contrató a un equipo de ingenieros con el objetivo explícito de hacer que la tecnología CAD 3D fuera más accesible para todos, y así lo hicieron desarrollando la primera tecnología de CAD en 3D que se ejecutaba en la plataforma de Windows. En el año 1995 lanzaron su primera versión del CAD 3D al mercado y en dos meses la herramienta ganó reconocimientos por su facilidad de uso”. (BI, 2017, p; 1)

En la siguiente figura 26-1 se aprecia el tipo de diseños realizados por el software Solidworks.

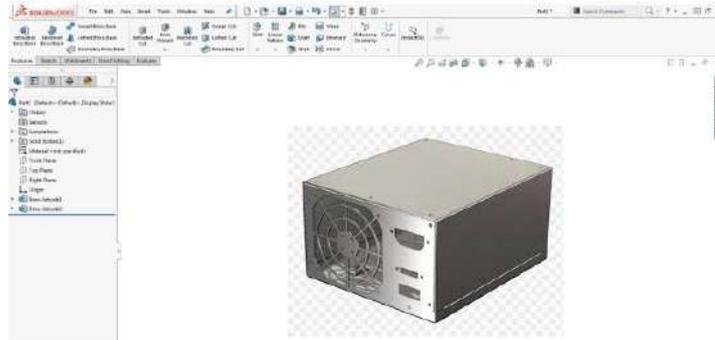


Figura 25-1: Diseños realizados por el software Solidworks
Fuente: (BI, 2017)

CAPITULO II

2 PROPUESTA Y DISEÑO DE PROTOTIPO

En el presente capítulo se presenta los diferentes diseños, procesos, construcción e implementación de un prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistemas OBD II, estos diseños se desarrollaron en base a módulos especializados que cumplan los requerimientos de funcionamiento del prototipo.

Como base principal para la construcción del prototipo se realizó una investigación bibliográfica, basada en trabajos previos con características similares, es por ello que para la realización se emplea una investigación experimental la cual se basa en la interpretación de resultados obtenidos durante la adquisición de datos, detección de fallos en el funcionamiento de los sensores en el motor.

Basado en los objetivos planteados al inicio del proyecto, en la Grafica 1-2 se muestra un diagrama en el cual se detalla las etapas del diseño del prototipo.

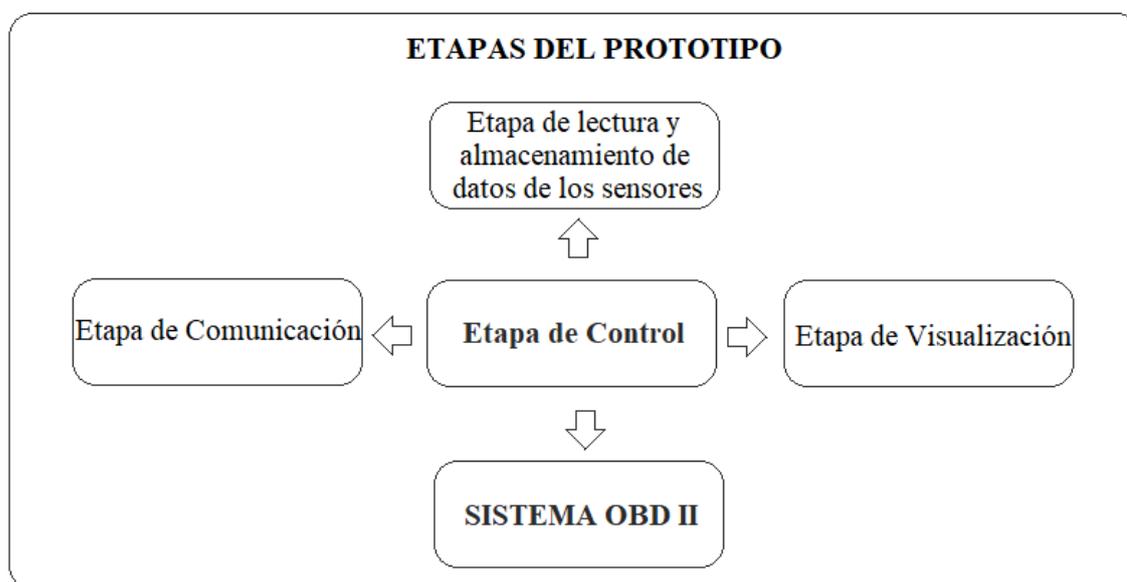


Gráfico 1-2: Etapas del diseño del prototipo

Realizado por: Montesdeoca, Andrea, 2020

La grafica anterior explica mediante etapas la constitución del diseño de un prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistemas OBD II, continuación se detallarán de manera específica cada una de ella.

2.1 Estructura de la tarjeta Electrónica del prototipo

Para el diseño de la placa o tarjeta electrónica que permitirá la puesta en marcha del prototipo se realiza una estructura, en la cual se muestra todos los elementos y módulos electrónicos que permitan su diseño, el Grafico 2-2 muestra a continuación la estructura de la tarjeta electrónica del prototipo.

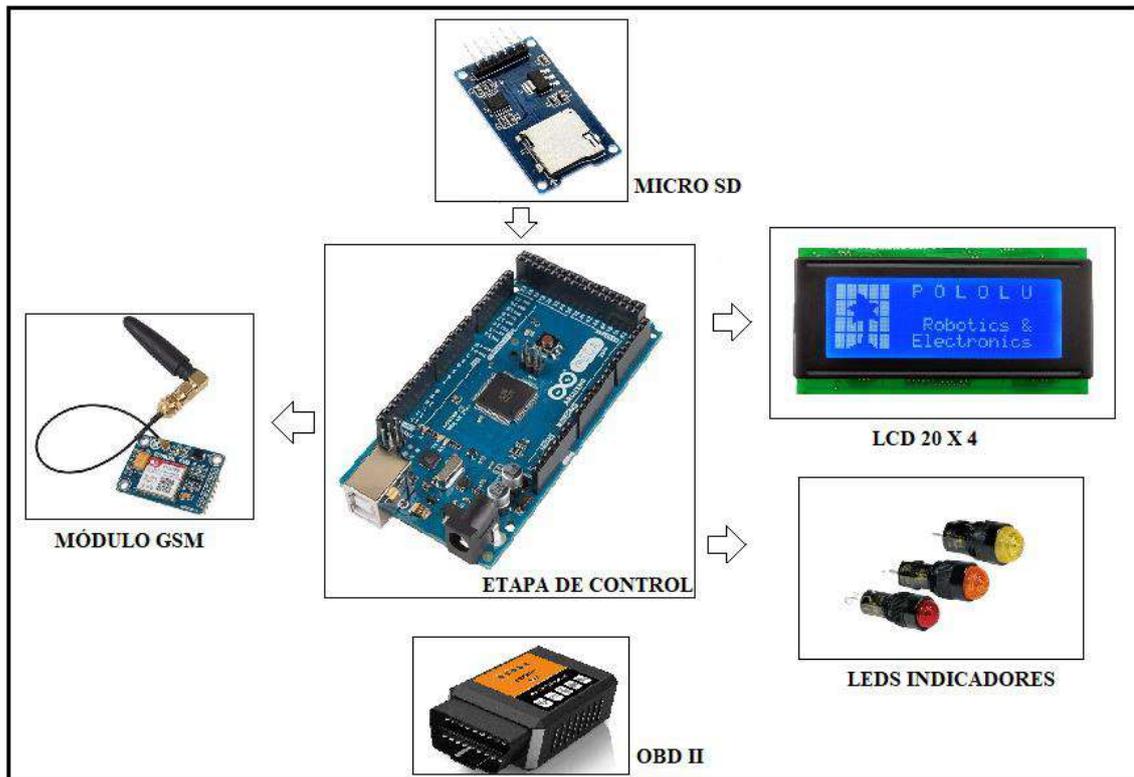


Gráfico 2-2: Estructura de la tarjeta electrónica del prototipo
Realizado por: Montesdeoca, Andrea, 2020

Como se muestra en el gráfico la estructura del diseño de la placa electrónica muestra los módulos más importantes para su implementación, entonces se detalla de la siguiente manera.

2.1.1 Conexión Arduino – modulo micro SD

Para la etapa de almacenamiento se realiza un análisis de funcionamiento y compatibilidad con un módulo micro SD compatible con Arduino, cuya función es almacenar los datos emitidos por la unidad de control electrónica de motor (ECU) del vehículo, la siguiente figura 1-2 muestra el tipo de conexión Arduino –micro SD.

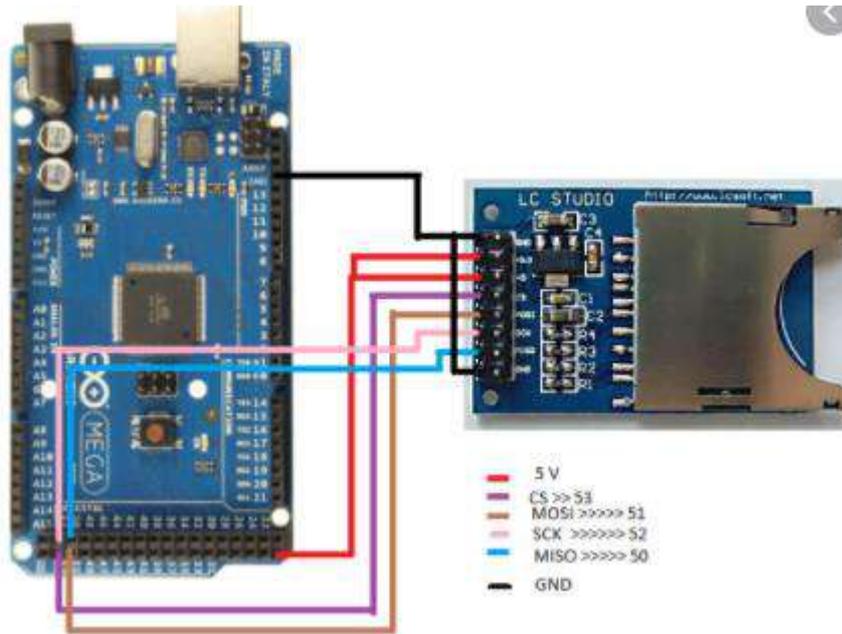


Figura 1-2: Conexión Arduino – micro SD

Realizado por: Montesdeoca, Andrea, 2020

2.1.1.1 *Diseño esquemático del circuito Arduino –micro SD*

Se realiza el diseño de un circuito electrónico que permita realizar el almacenamiento de datos mediante la utilización de un módulo SD, el cual permitirá almacenar datos emitidos como respuesta de la unidad electrónica de motor (ECU) en el vehículo; la siguiente figura 2-2 muestra el circuito esquemático Arduino - micro SD.

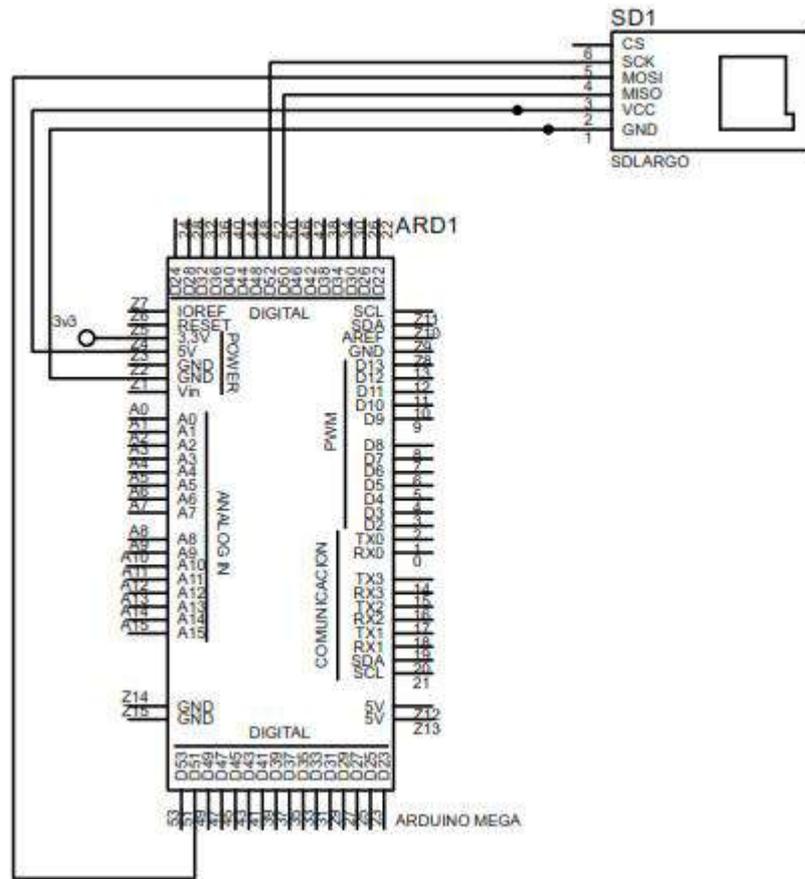


Figura 2-2: Circuito esquemático Arduino - micro SD
 Realizado por: Montesdeoca, Andrea, 2020

La tensión de alimentación del módulo es de 3.3VDC, su compatibilidad con la tarjeta de desarrollo Arduino mega incluye un regulador de voltaje una alimentación directa de 5V, dependiendo el diseño y la finalidad de cada usuario.

Se ha implementado una tarjeta micro SD con la finalidad y ventaja de proporcionar una memoria casi ilimitada al momento de realizar el almacenamiento de datos que se obtendrá de los sensores automotrices; se realizó una conexión muy sencilla principalmente con la alimentación directa mediante los terminales de 5V y GND, se realizó la conexión de los terminales con el módulo micro SD tal como se muestra en la siguiente figura 3-2.

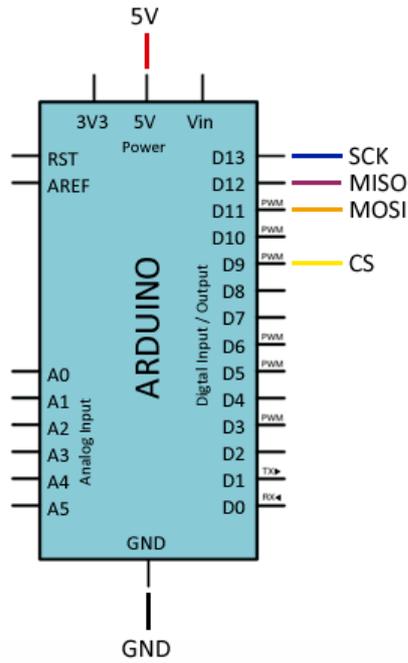


Figura 3-2: Conexión de pines para módulo micro SD
 Realizado por: Montesdeoca, Andrea, 2020

2.1.2 Conexión Arduino – módulo SIM 800L

Para la etapa de comunicación se implementa un módulo de comunicación denominado SIM 800L compatible con los terminales de la tarjeta de desarrollo Arduino mega, la figura 4-2 muestra el tipo de conexión Arduino – módulo SIM 800L recomendado para la comunicación.

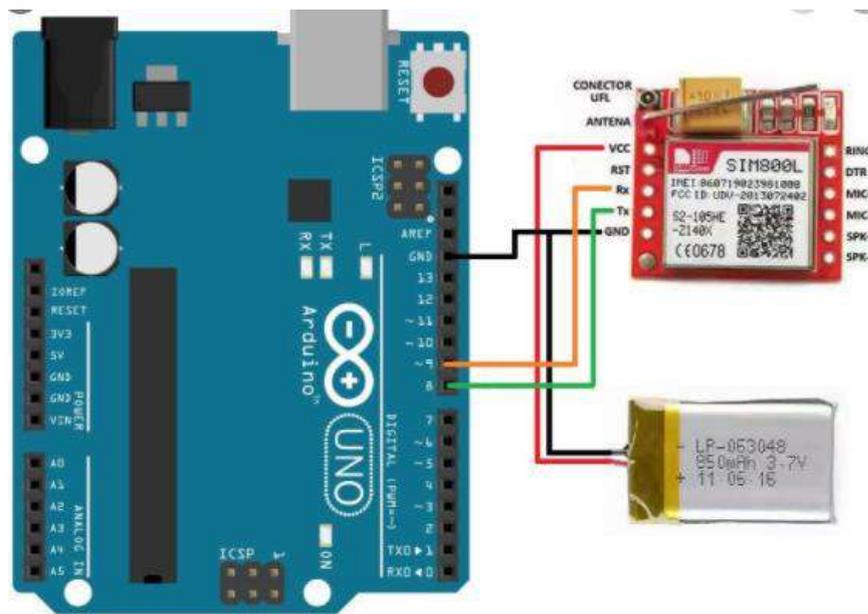


Figura 4-2: Conexión Arduino – módulo SIM 800L
 Realizado por: Montesdeoca, Andrea, 2020

2.1.2.1 Diseño esquemático del circuito Arduino - módulo SIM 800L

Para este diseño esquemático se realizó la conexión mediante el SIM800L la cual está configurada y controlada por vía UART usando comandos AT de manera externa, tomaremos en cuenta las siguientes especificaciones:

- Vin – Vin (Se encuentra físicamente en el terminal 5).
- GND – GND (Se encuentra físicamente en el terminal 1).
- TX1 – El que corresponde al TX, o pin 2.
- RX1 – El que corresponde al RX, o pin 3.

Según las especificaciones anteriores, la figura 5-2 muestra el circuito esquemático del módulo GSM.

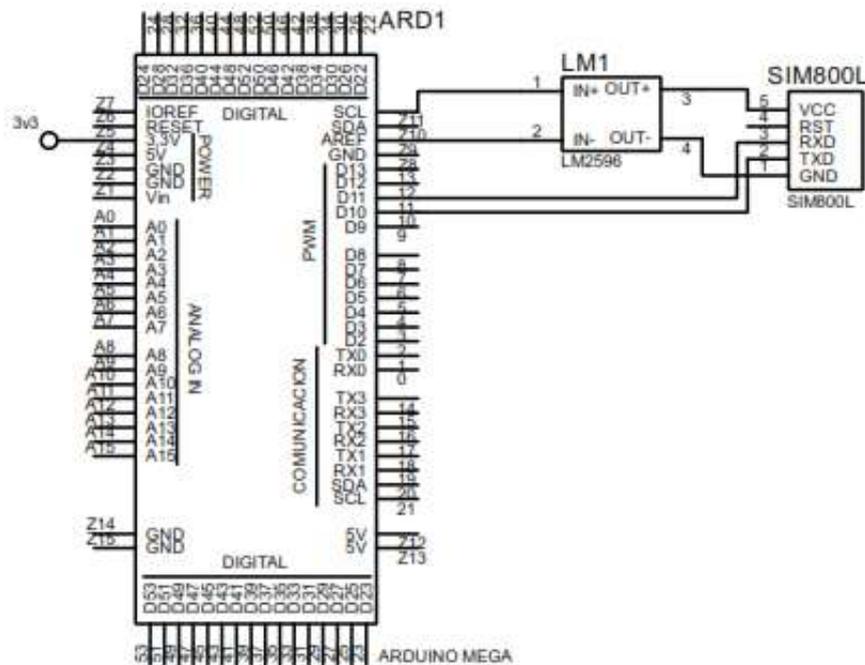


Figura 2-5: Circuito esquemático del módulo GSM

Realizado por: Montesdeoca, Andrea, 2020

Se observa en la figura 5-2 que dicho modulo contiene un terminal de alimentación de 5VDC y de terminales adicionales de comunicación Rx y Tx, por este motivo el método de diseño para la comunicación es simple, pero se debe tener en cuenta que si vamos a subir instrucciones o datos vía serial se debe definir los terminales de comunicación vía software, usando librerías de Arduino Software Serial.

Para el tipo de comunicación con la tarjeta de desarrollo Arduino se conecta el terminal de comunicación Rx al terminal 11 del Arduino y el terminal de transmisión Tx al 10 tal como se muestra en la figura 5-2.

2.1.3 Conexión Arduino – pantalla LCD

Para el diseño de la etapa de visualización del circuito electrónico se implementa un módulo LCD 20 X 4, el cual permitirá reflejar los datos obtenidos por los sensores del motor, la figura 6-2 muestra la conexión Arduino – pantalla LCD.

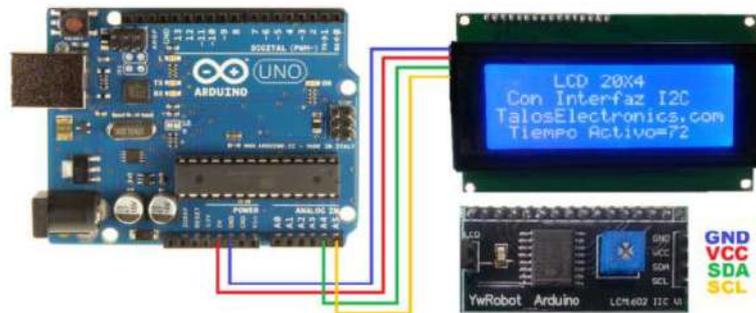


Figura 6-2: Conexión Arduino – pantalla LCD.
Realizado por: Montesdeoca, Andrea, 2020

2.1.3.1 Interfaz de conexión I2C

El módulo I2C para LCD permite la comunicación entre un Arduino o microcontrolador y un LCD 20x4 por medio de I2C, lo cual facilita las conexiones a la hora de llevar a cabo cualquier tipo de aplicación, este adaptador permite administrar de una manera mucho más óptima la distribución de puertos ya que se pueden llevar a cabo la presentación de datos a través de solamente dos cables, la siguiente figura 7-2 muestra el método de conexión de la interfaz I2C.

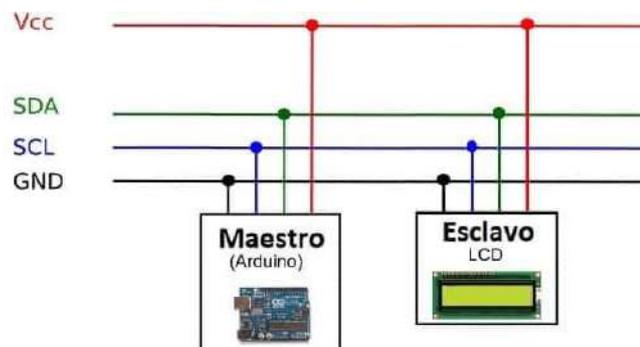


Figura 7-2: Conexión de la interfaz I2c
Realizado por: Montesdeoca, Andrea, 2020

2.1.3.2 Diseño esquemático del circuito Arduino - LCD

Se seleccionó un Display LCD, el cual mediante la etapa de desarrollo recepta los datos emitidos por la unidad de control electrónica de motor (ECU) del vehículo, procesa y se muestra en la pantalla LCD, a continuación, la figura 8-2 muestra el circuito esquemático Arduino - pantalla LCD.

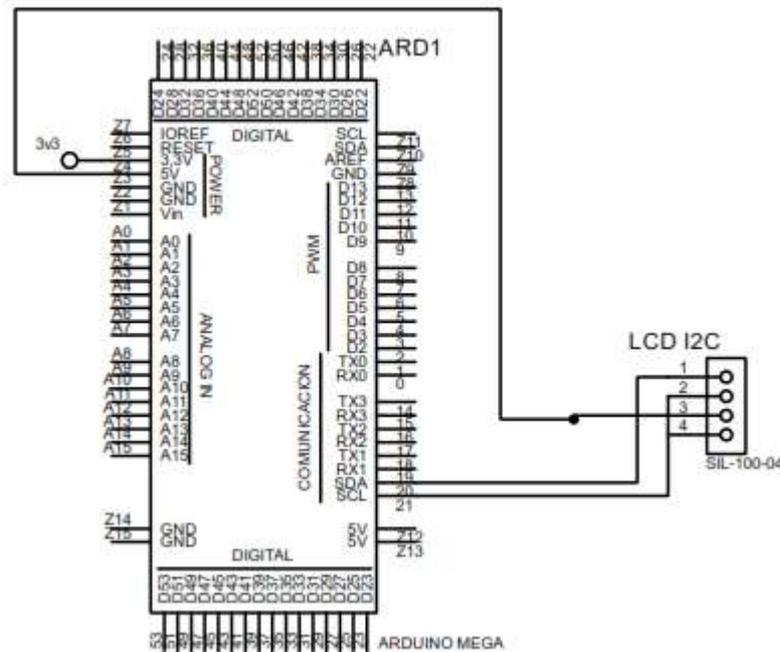


Figura 8-2: Circuito esquemático Arduino - pantalla LCD
Realizado por: Montesdeoca, Andrea, 2020

El módulo utilizado trabaja mediante comunicación I2C, el mismo que por una interfaz de salidas y entradas digitales se comunica de manera directa y con la utilización de 2 terminales con la tarjeta de desarrollo Arduino.

La dirección I2C por defecto del módulo puede ser 0x3F o en otros casos 0x27, es muy importante identificar correctamente la dirección I2C de nuestro módulo, pues de otra forma nuestro programa no funcionará correctamente. Para identificar la dirección específica de nuestro módulo podemos utilizar un pequeño sketch de prueba llamado: I2C Scanner, el cual nos permite identificar la dirección I2C del dispositivo conectado al Arduino; si en caso existiera la necesidad de trabajar con más de un LCD podemos modificar la dirección I2C del módulo adaptador para esto es necesario conectar los terminales SDA/ACL del Arduino a los pines 1 y 2 de la interfaz I2C como se muestra anteriormente en la figura 8-2.

2.1.4 Conexión Arduino - sistema OBDII

Al diseño general del prototipo se añade un módulo de comunicación, este producto funciona como un puente de datos entre el puerto OBD de un automóvil y Arduino o hardware similar con una biblioteca dedicada de código abierto, proporciona acceso a datos OBD-II de alta velocidad e integra un sensor de movimiento 9-DOF con un algoritmo de fusión de sensor incorporado; el adaptador se alimenta directamente desde el puerto OBD y emite un voltaje regulado de 5 V / 2A para alimentar los dispositivos conectados, la siguiente figura 9-2 muestra el tipo de conexión Arduino – sistema OBDII

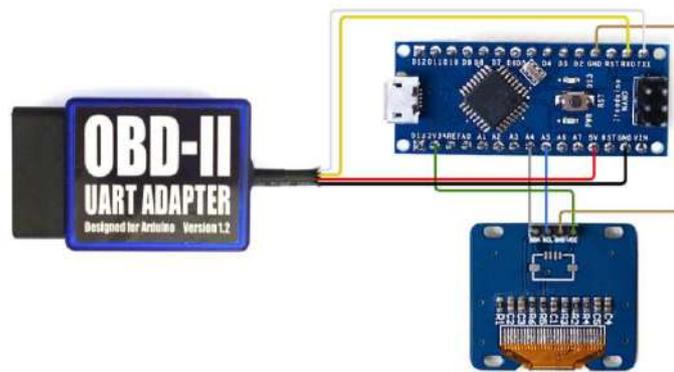


Figura 9-2: Conexión Arduino - sistema OBDII
Realizado por: Montesdeoca, Andrea, 2020

2.1.4.1 Diseño esquemático del circuito Arduino - sistema OBD II

El adaptador permanece enchufado en el puerto OBD generalmente ubicado debajo de la columna de dirección o ligeramente a la izquierda de la misma, un cable sale del adaptador y se divide en un conector de 4 terminales, incluidas las líneas eléctricas (VCC / GND) y las líneas de datos (Rx / Tx). Se pueden conectar a Arduino con terminales de ruptura integrados o escudo de ruptura, el módulo se verá ordenado en el automóvil con solo un cable, la figura 10-2, muestra el circuito esquemático Arduino – sistema OBDII.

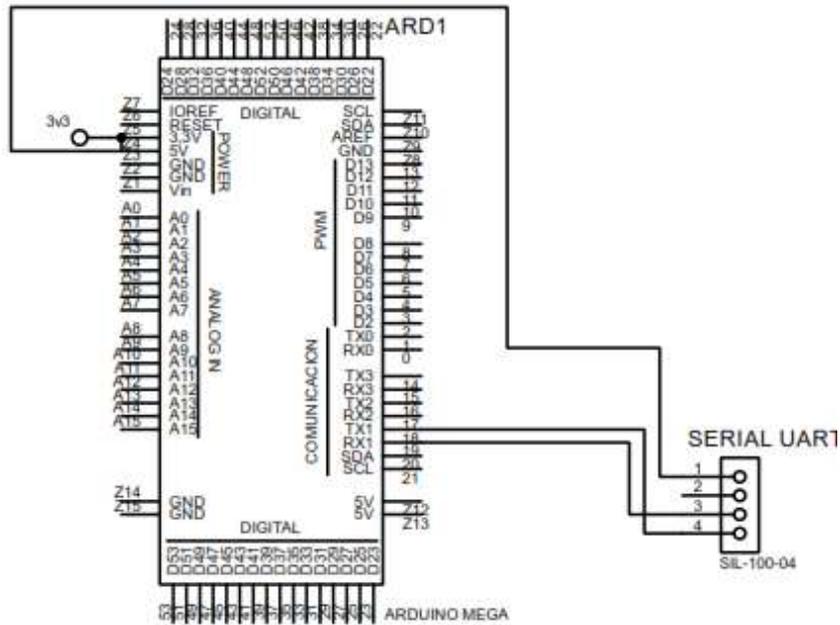


Figura 10-2: Circuito esquemático Arduino - sistema OBDII
 Realizado por: Montesdeoca, Andrea, 2020

2.1.5 Circuito esquemático general de la tarjeta electrónica

Culminada todos los diseños de las diferentes etapas mediante software proteus se unifican todos los diseños para mostrar un resultado general del circuito esquemático, dentro de ellos se establece los siguientes:

- Arduino – Modulo micro SD.
- Arduino – Modulo GSM.
- Arduino – Pantalla LCD.
- Arduino – Sistema OBDII.

El entorno ISIS de Proteus muestra un diseño general que será utilizado como muestra para un entorno ARES que determina el diseño PCB, la siguiente figura 11-2 muestra el circuito esquemático general de la tarjeta electrónica.

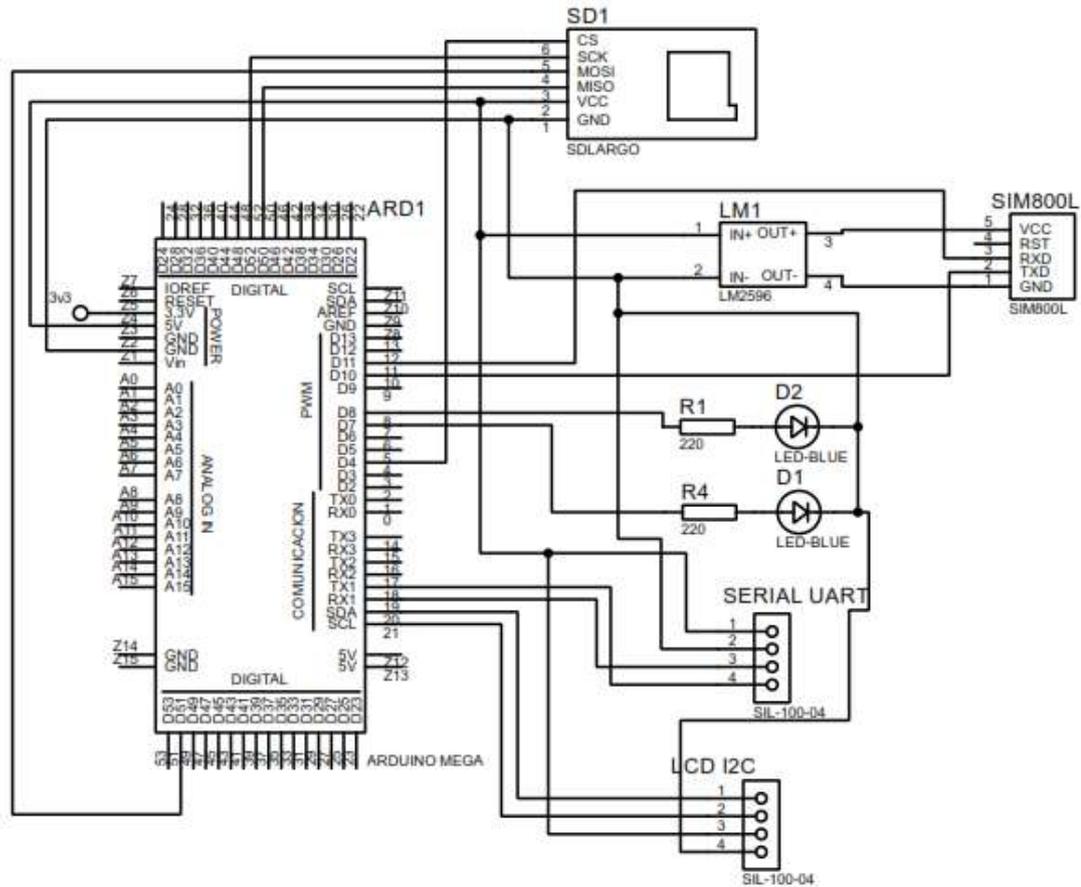


Figura 11-2: Circuito esquemático de la tarjeta electrónica
 Realizado por: Montesdeoca, Andrea, 2020

2.1.6 Diseño PCB del circuito electrónico

Culminado el proceso de diseño esquemático, se realiza el PCB propio del software Proteus, a la cual se denomina ARES cuya funcionalidad es crear un PCB del diseño esquemático realizado en ISIS, luego el sistema permite implementar un circuito o placa electrónica.

2.1.6.1 Diseño de pistas

Se realiza el cálculo de pistas tomando en cuenta lo siguiente:

Ecuación 1: Ecuación para el cálculo del grosor de las pistas.

$$lonza / pie^2 = 35 \text{ micras de espesor}$$

Para el cálculo de la pistas de tomo en cuenta la siguiente fórmula establecida a continuación:

En donde:

I= Corriente Máxima

ΔT = Diferencia de temperatura

K1, K2, K3= Constantes para el calculo

Cabe recalcar que los valores establecidos las K1, K2, K3 vienen establecidos de manera que existen diseños de pistas PCB a dos caras en la baquelita, de esta manera se realizan los siguientes cálculos tomando en cuenta que 1A es la corriente máxima que debe soportar las pistas.

Datos:

$$I= 1\text{AMP}$$

$$\Delta T= 25^{\circ}\text{C}$$

$$\text{GROSOR}= 1\text{oz /ft}^2$$

$$K1= 0.0647$$

$$K2= 0.4281$$

$$K3= 0.6732$$

Ecuación 2: Ecuación del área y ancho de la pista

$$AREA = \left(\frac{I}{K1 * \Delta T^{K2}} \right)^{\frac{1}{K3}}$$

$$AREA = \left(\frac{1}{0.0647 * 25^{0.4281}} \right)^{\frac{1}{0.6732}}$$

$$AREA = 7.539$$

$$ANCHO = \frac{AREA}{GROSOR * 1.378}$$

$$ANCHO = \frac{7.539}{1 * 1.378}$$

$$ANCHO = 5.471\text{th}$$

Una vez calculado el ancho de las pistas, se toma en cuenta que (th) es la simbología establecida por el software Proteus y que en este caso es equivalente al ancho; la siguiente figura 12-2 muestra el diseño PCB del circuito electrónico.

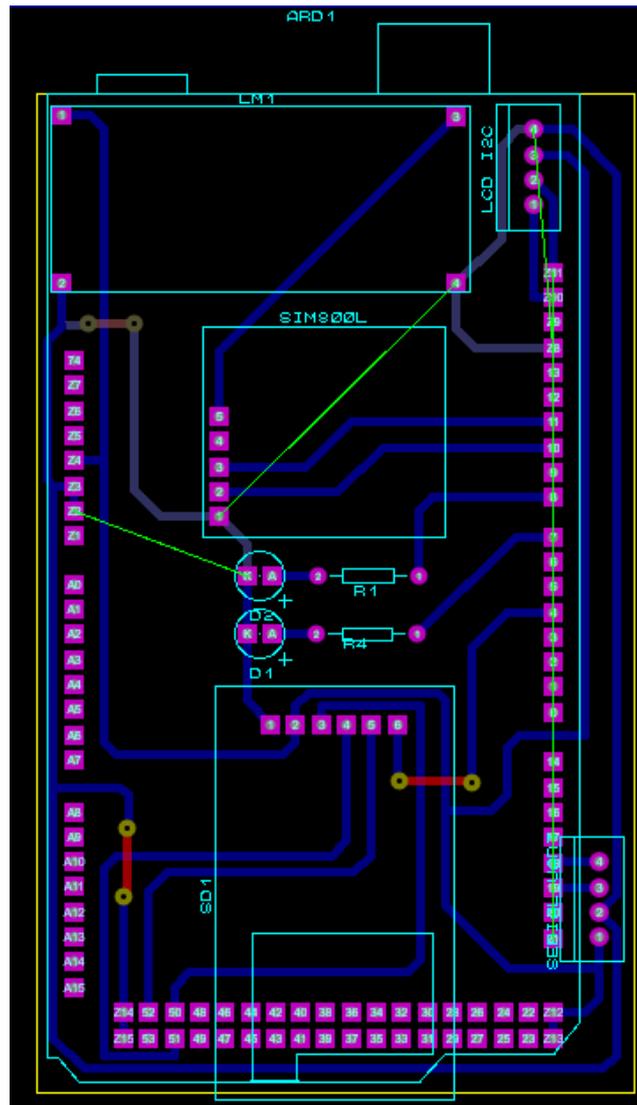


Figura 12-2: Diseño PCB del circuito electrónico
 Realizado por: Montesdeoca, Andrea, 2020

2.1.7 Visualización 3D

Se realiza una visualización 3D de la placa del circuito esquemático y PCB para apreciar la correcta instalación de los elementos y módulos electrónicos. A continuación, en la Figura 13-2 se visualiza en 3D del circuito electrónico.

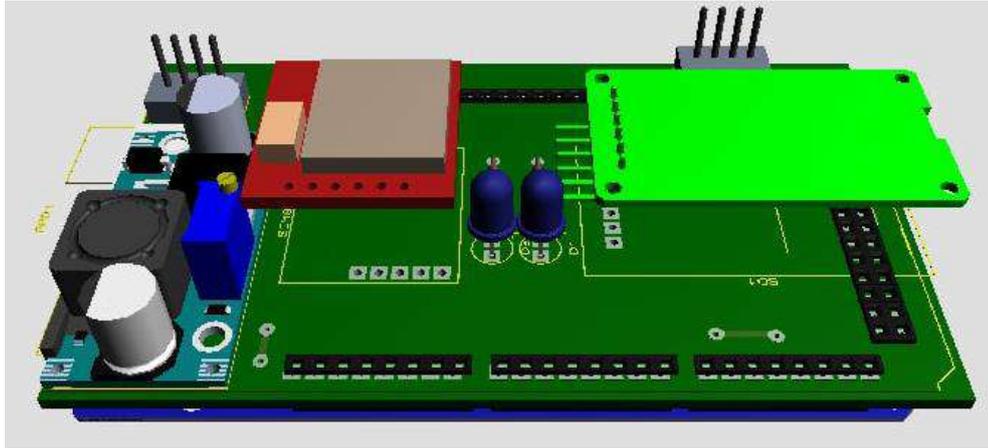


Figura 13-2: Visualización 3D del circuito electrónico
Realizado por: Montesdeoca, Andrea, 2020

2.2 Implementación del prototipo

En esta etapa se muestra el proceso de construcción física del prototipo, los diseños mediante software Proteus sirven como base principal para la construcción mediante la utilización de módulos y dispositivos electrónicos.

2.2.1 Construcción de la placa PCB

Para la construcción de la placa PCB se utiliza una máquina CNC router adecuada para la fabricación de circuitos electrónicos, este sistema de construcción es el más utilizado y actualizado, la figura 14-2 muestra el proceso de corte de la placa PCB.



Figura 14-2: Proceso de corte de la placa PCB.
Realizado por: Montesdeoca, Andrea, 2020

El resultado final del corte muestra todo el diseño realizado para la colocación de cada módulo y elementos electrónicos, la siguiente figura 15-2 muestra el diseño final de la placa PCB:

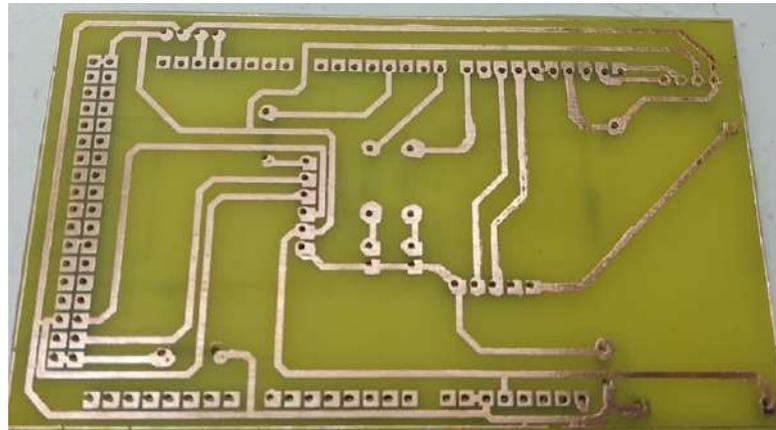


Figura 15-2: Diseño final de la placa PCB
Realizado por: Montesdeoca, Andrea, 2020

2.2.2 Colocación de elementos y módulos electrónicos

Al resultado final de la placa PCB se añade los diferentes módulos y elementos electrónicos adquiridos tales como:

- Arduino Mega.
- Módulo GSM.
- Modulo SD.
- LCD 20 X 7.
- Leds indicadores.
- Bornes para puerto OBDII.

Las siguientes Figuras 16-2 y 17-2 muestran la implementación de los elementos electrónicos en la placa PCB.

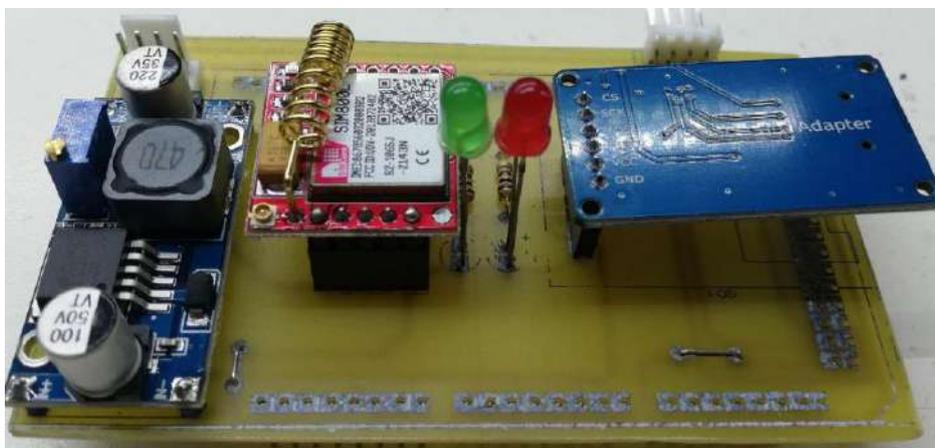


Figura 16-2: Implementación de elementos electrónicos
Realizado por: Montesdeoca, Andrea, 2020

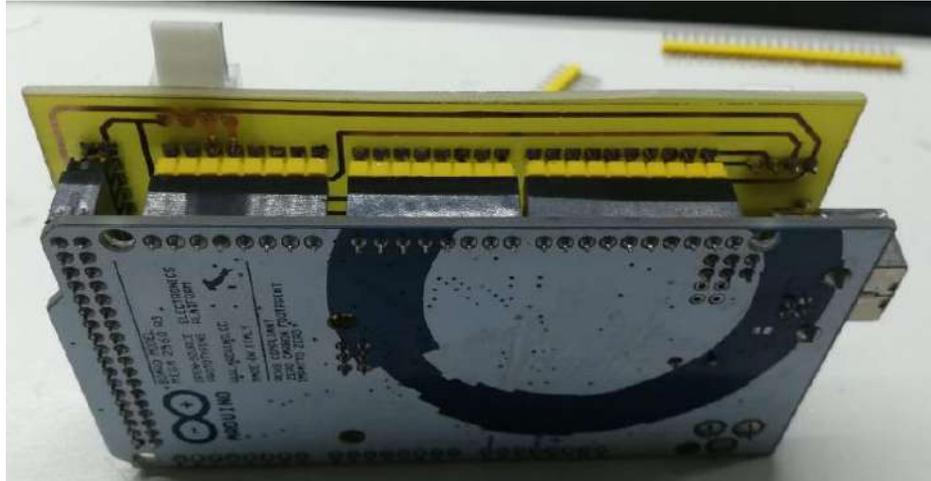


Figura 17-2: Arduino en la placa PCB
Realizado por: Montesdeoca, Andrea, 2020

2.3 Construcción de la caja de protección

Se diseñó una caja de protección, en la cual se inserta la placa electrónica y los diferentes módulos que constituyen el prototipo, para ello se realizó un diseño mediante software Solidworks e impresora 3D, la figura 18-2 a continuación muestra el proceso de impresión de la caja de protección.

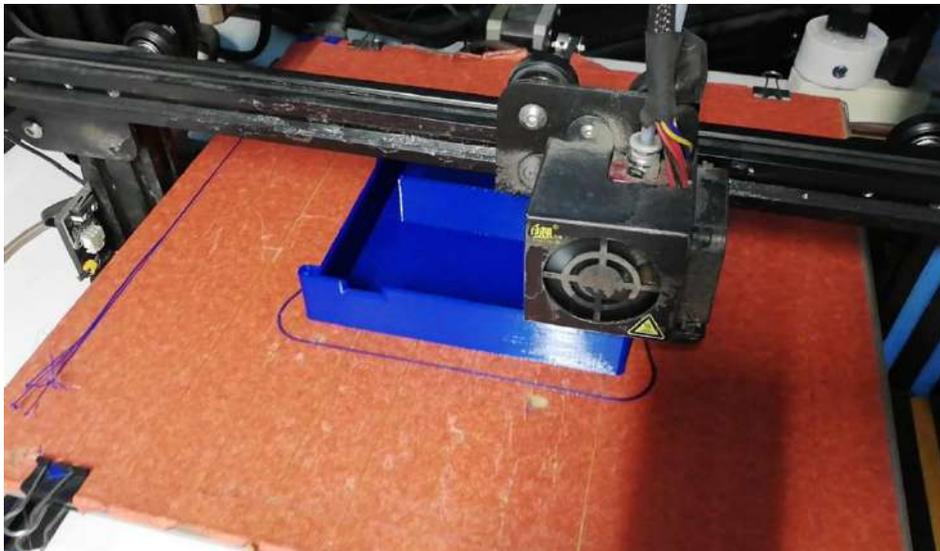


Figura 18-2: Proceso de impresión de la caja de protección.
Realizado por: Montesdeoca, Andrea, 2020

2.3.1 Construcción final de la caja de protección

Culminado el proceso de impresión 3D que dura alrededor de 10 horas, la siguiente figura 19-2 muestra la construcción final de la caja de protección.



Figura 19-2: Construcción final de la caja de protección
Realizado por: Montesdeoca, Andrea, 2020

2.3.2 Armado total del prototipo

Se ubica la tarjeta de control electrónica y el módulo de visualización LCD en la caja de protección contra factores externos, con este proceso se da más estética al resultado final y los módulos que constituyen al dispositivo están totalmente protegidos, la figura 20-2 muestra la ubicación de los módulos.



Figura 20-2: Ubicación de los módulos
Realizado por: Montesdeoca, Andrea, 2020

2.3.3 Prototipo Final

Luego de diferentes procesos y etapas, se muestra en la figura 21-2 el resultado del prototipo final electrónico para registros telemáticos y detección de fallos del motor mediante sistemas OBDII.



Figura 21-2: Prototipo Final
Realizado por: Montesdeoca, Andrea, 2020

2.4 Diseño del algoritmo de programación.

Para el funcionamiento del prototipo se diseña un algoritmo de programación que permita ejecutar las diferentes funciones establecidas por el dispositivo, para ello se muestra en el Gráfico 3-2 el diagrama de flujo referente al algoritmo de programación del prototipo.

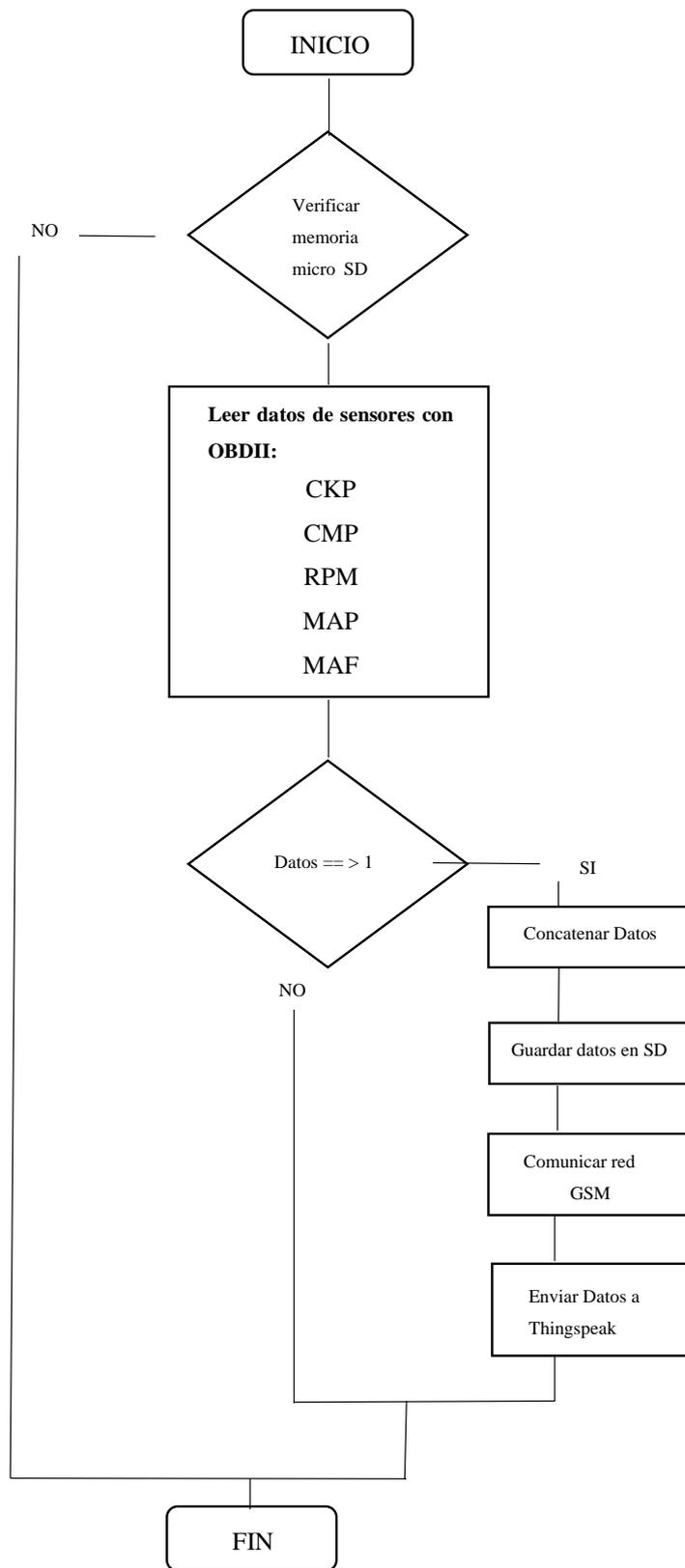


Gráfico 3-2: Diagrama de flujo referente al algoritmo de programación del prototipo

Realizado por: Montesdeoca, Andrea, 2020

2.5 Funcionamiento del prototipo

Para empezar con las pruebas de funcionamiento vamos a entender la constitución externa del prototipo de la siguiente manera:

2.5.1 Pantalla principal

La pantalla principal muestra los datos requeridos y emitidos por cada sensor del vehículo, se puede apreciar los datos de:

- Tipo de sensor
- Valor del sensor

La figura 22-2 a continuación muestra la pantalla principal del prototipo.



Figura 22-2: Pantalla principal del prototipo
Realizado por: Montesdeoca, Andrea, 2020

2.5.2 Leds indicadores

En el dispositivo se visualiza 2 tipos de led indicadores, los cuales cumplen con las siguientes funciones:

Led verde: Refleja el encendido del prototipo, comprobando su correcto funcionamiento.

Led azul: Indica la correcta lectura de datos.

La figura 23-2 se muestra los diferentes leds indicadores.



Figura 23-2: Leds indicadores
Realizado por: Montesdeoca, Andrea, 2020

2.5.3 Puerto USB de programación Arduino

El prototipo está integrado por un puerto de comunicación USB, que sirve para establecer comunicación entre ordenador y placa electrónica para su programación, en la cual se puede realizar varios cambios o ingresar un nuevo algoritmo de programación según requerimientos del usuario, la figura 24-2 muestra el puerto USB de programación Arduino.



Figura 24-2: Puerto USB de programación Arduino
Realizado por: Montesdeoca, Andrea, 2020

2.5.4 Puerto de alimentación

El prototipo consta de un puerto de alimentación común, su alimentación de entrada para el funcionamiento es de 12 VDC, este puerto es utilizado cuando se manipula el prototipo de manera externa, la figura 25-2 muestra el puerto de alimentación en el prototipo.



Figura 25-2: Puerto de alimentación en el prototipo
Realizado por: Montesdeoca, Andrea, 2020

2.5.5 Ranura para micro SD

Se integró una ranura para micro SD, la cual es la encargada de realizar el almacenamiento de los sensores en el vehículo, la tarjeta micro SD es de fácil adquisición y puede ser colocada en un adaptador para que su información sea extraída a un ordenador, la figura 26-2 muestra la ranura para micro SD integrada al prototipo.



Figura 26-2: Ranura para micro SD
Realizado por: Montesdeoca, Andrea, 2020

2.5.6 Puerto de conexión para sistema OBD II

Dependiendo del vehículo la ubicación de la conexión del puerto puede cambiar de lugar, el puerto OBD II suele estar ubicado en la zona de los pies del conductor, ya sea debajo del volante o en la caja de fusibles, otros fabricantes sitúan la conexión OBD II en la parte del cenicero o incluso en el asiento del copiloto; mediante la tarjeta electrónica y el Arduino se adecuo un puerto de salida conectado hacia el sistema OBD II.

La figura muestra 27-2 muestra el puerto OBDII integrado en el prototipo.



Figura 27-2: Puerto OBDII integrado en el prototipo
Realizado por: Montesdeoca, Andrea, 2020

CAPITULO III

3 VALIDACIÓN DEL PROTOTIPO

3.1 Introducción

Finalizado la implementación de un prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistema OBD II, se procede a realizar las diferentes pruebas de obtención de datos del prototipo.

3.2 Análisis de funcionamiento

Se realizó un análisis para determinar el correcto funcionamiento y efectividad del prototipo; el dispositivo se instaló en un vehículo Híbrido Toyota modelo Camry modelo 2017, el módulo de conexión vehículo y sistema OBDII son de fácil de conexión. Se encuentra situado en la parte baja del volante del conductor, la figura 1-3 muestra la posición de conexión al puerto OBDII del vehículo.



Figura 1-3: Posición de conexión a la unidad de control electrónica de motor
Realizado por: Montesdeoca, Andrea, 2020

El vehículo Toyota Camry en el cual se efectuaron la prueba, arrojó como resultado la captura de 10 datos importantes los cuales son:

- Engine RPM (rpm).
- Engine coolant temperature (°C).
- Engine oil temperature (°C).
- Intake temperature (°C).
- Flow pressure (grams/s).
- Barometric pressure (kPa).
- Vehicle speed (km/h).
- Vehicle running distance (km).
- Throttle position (%).
- Ambient temperature (°C).

3.2.1 *Funcionamiento del prototipo*

Para realizar la adquisición de datos el vehículo debe estar en modo encendido con el motor en ON, cada uno de los principales sensores funcionan cuando el vehículo se encuentra en movimiento, la toma de datos se la realiza en tiempo real.

En la figura 2-3 se muestra el funcionamiento del prototipo.

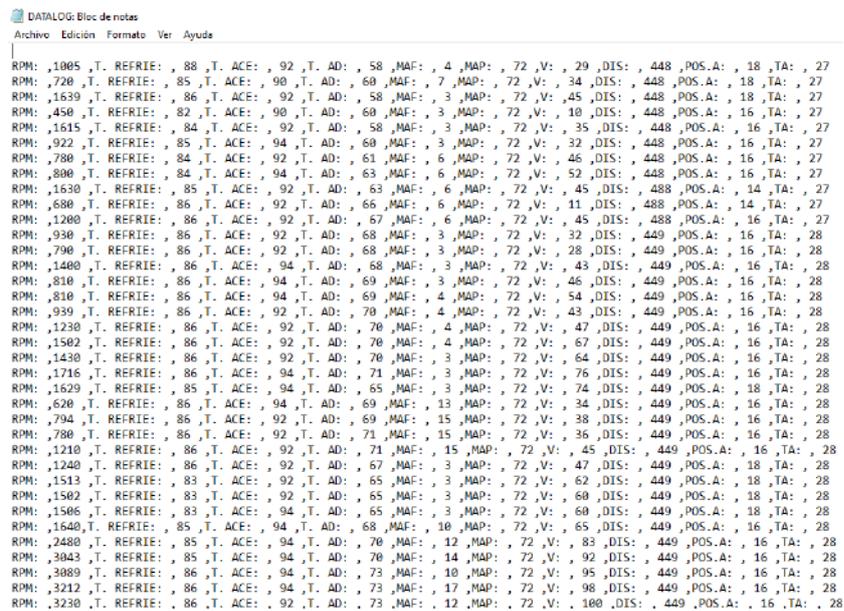


Figura 2-3: Funcionamiento del prototipo.
 Realizado por: Montesdeoca, Andrea, 2020

3.3 Visualización de resultados en Datalogger

Los datos enviados por la unidad de control electrónica de motor (ECU) de los sensores automotrices son almacenados en una tarjeta de memoria micro SD, la ventaja de la tarjeta es su extracción y conexión en un computador; los datos adquiridos se encuentran almacenados en texto plano con sus variables y nombres separados por comas, para visualizar los datos en un computador se lo hace mediante blog de notas o Excel de Windows.

La figura 3-3 muestra los datos almacenados en datalogger.



DATALOG: Bloc de notas
Archivo Edición Formato Ver Ayuda

```
RPM: ,1005 ,T. REFRIE: , 88 ,T. ACE: , 92 ,T. AD: , 58 ,MAF: , 4 ,MAP: , 72 ,V: , 29 ,DIS: , 448 ,POS.A: , 18 ,TA: , 27  
RPM: ,720 ,T. REFRIE: , 85 ,T. ACE: , 90 ,T. AD: , 60 ,MAF: , 7 ,MAP: , 72 ,V: , 34 ,DIS: , 448 ,POS.A: , 18 ,TA: , 27  
RPM: ,1639 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 58 ,MAF: , 3 ,MAP: , 72 ,V: , 45 ,DIS: , 448 ,POS.A: , 18 ,TA: , 27  
RPM: ,450 ,T. REFRIE: , 82 ,T. ACE: , 90 ,T. AD: , 60 ,MAF: , 3 ,MAP: , 72 ,V: , 10 ,DIS: , 448 ,POS.A: , 16 ,TA: , 27  
RPM: ,1615 ,T. REFRIE: , 84 ,T. ACE: , 92 ,T. AD: , 58 ,MAF: , 3 ,MAP: , 72 ,V: , 35 ,DIS: , 448 ,POS.A: , 16 ,TA: , 27  
RPM: ,922 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 60 ,MAF: , 3 ,MAP: , 72 ,V: , 32 ,DIS: , 448 ,POS.A: , 16 ,TA: , 27  
RPM: ,780 ,T. REFRIE: , 84 ,T. ACE: , 92 ,T. AD: , 61 ,MAF: , 6 ,MAP: , 72 ,V: , 46 ,DIS: , 448 ,POS.A: , 16 ,TA: , 27  
RPM: ,800 ,T. REFRIE: , 84 ,T. ACE: , 94 ,T. AD: , 63 ,MAF: , 6 ,MAP: , 72 ,V: , 52 ,DIS: , 448 ,POS.A: , 16 ,TA: , 27  
RPM: ,1630 ,T. REFRIE: , 85 ,T. ACE: , 92 ,T. AD: , 63 ,MAF: , 6 ,MAP: , 72 ,V: , 45 ,DIS: , 488 ,POS.A: , 14 ,TA: , 27  
RPM: ,680 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 66 ,MAF: , 6 ,MAP: , 72 ,V: , 11 ,DIS: , 488 ,POS.A: , 14 ,TA: , 27  
RPM: ,1200 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 67 ,MAF: , 6 ,MAP: , 72 ,V: , 45 ,DIS: , 488 ,POS.A: , 16 ,TA: , 27  
RPM: ,930 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 68 ,MAF: , 3 ,MAP: , 72 ,V: , 32 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,790 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 68 ,MAF: , 3 ,MAP: , 72 ,V: , 28 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1400 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 68 ,MAF: , 3 ,MAP: , 72 ,V: , 43 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,810 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 69 ,MAF: , 3 ,MAP: , 72 ,V: , 46 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,810 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 69 ,MAF: , 4 ,MAP: , 72 ,V: , 54 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,939 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 70 ,MAF: , 4 ,MAP: , 72 ,V: , 43 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1230 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 70 ,MAF: , 4 ,MAP: , 72 ,V: , 47 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1502 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 70 ,MAF: , 4 ,MAP: , 72 ,V: , 67 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1430 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 70 ,MAF: , 3 ,MAP: , 72 ,V: , 64 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1716 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 71 ,MAF: , 3 ,MAP: , 72 ,V: , 76 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1629 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 65 ,MAF: , 3 ,MAP: , 72 ,V: , 74 ,DIS: , 449 ,POS.A: , 18 ,TA: , 28  
RPM: ,620 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 69 ,MAF: , 13 ,MAP: , 72 ,V: , 34 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,794 ,T. REFRIE: , 85 ,T. ACE: , 92 ,T. AD: , 69 ,MAF: , 15 ,MAP: , 72 ,V: , 38 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,780 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 71 ,MAF: , 15 ,MAP: , 72 ,V: , 36 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1210 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 71 ,MAF: , 15 ,MAP: , 72 ,V: , 45 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,1240 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 67 ,MAF: , 3 ,MAP: , 72 ,V: , 47 ,DIS: , 449 ,POS.A: , 18 ,TA: , 28  
RPM: ,1513 ,T. REFRIE: , 83 ,T. ACE: , 92 ,T. AD: , 65 ,MAF: , 3 ,MAP: , 72 ,V: , 62 ,DIS: , 449 ,POS.A: , 18 ,TA: , 28  
RPM: ,1502 ,T. REFRIE: , 83 ,T. ACE: , 92 ,T. AD: , 65 ,MAF: , 3 ,MAP: , 72 ,V: , 60 ,DIS: , 449 ,POS.A: , 18 ,TA: , 28  
RPM: ,1506 ,T. REFRIE: , 83 ,T. ACE: , 94 ,T. AD: , 65 ,MAF: , 3 ,MAP: , 72 ,V: , 60 ,DIS: , 449 ,POS.A: , 18 ,TA: , 28  
RPM: ,1640 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 68 ,MAF: , 10 ,MAP: , 72 ,V: , 65 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,2480 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 70 ,MAF: , 12 ,MAP: , 72 ,V: , 83 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,3043 ,T. REFRIE: , 85 ,T. ACE: , 94 ,T. AD: , 70 ,MAF: , 14 ,MAP: , 72 ,V: , 92 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,3089 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 73 ,MAF: , 10 ,MAP: , 72 ,V: , 95 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,3212 ,T. REFRIE: , 86 ,T. ACE: , 94 ,T. AD: , 73 ,MAF: , 17 ,MAP: , 72 ,V: , 98 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28  
RPM: ,3230 ,T. REFRIE: , 86 ,T. ACE: , 92 ,T. AD: , 73 ,MAF: , 12 ,MAP: , 72 ,V: , 100 ,DIS: , 449 ,POS.A: , 16 ,TA: , 28
```

Figura 3-3: Datos almacenados en datalogger
Realizado por: Montesdeoca, Andrea, 2020

3.3.1 Visualización de datos en Excel

Los datos de los sensores almacenados en texto plano se los procesa en el software Excel con la finalidad de tomar gráficas de funcionamiento de cada sensor.

Las siguientes figuras 4-3 y 5-3 muestran el proceso de texto plano a Excel.

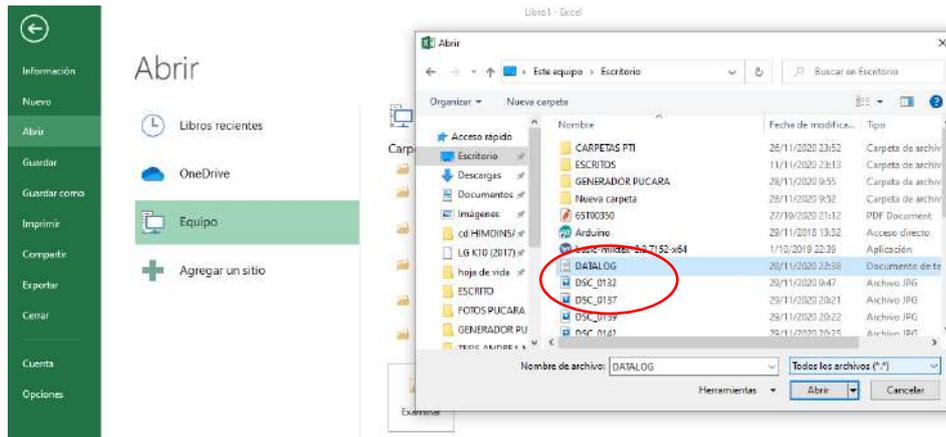


Figura 4-3: Apertura de Blog de notas en Excel
Realizado por: Montesdeoca, Andrea, 2020

Al cargar el archivo a Excel, en la pantalla se debe delimitar los caracteres a separar. La figura 5-3, muestra el siguiente paso de delimitación de caracteres:

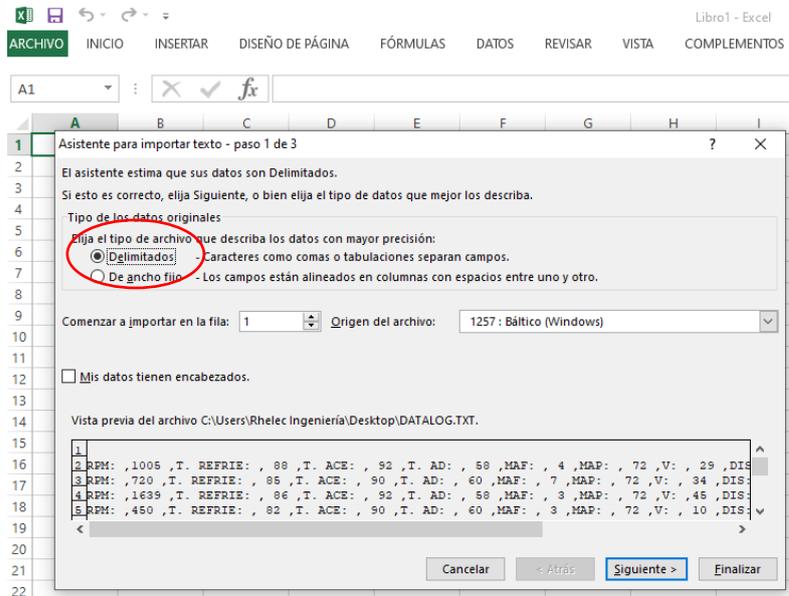


Figura 5-3: Delimitación de caracteres
Realizado por: Montesdeoca, Andrea, 2020

Delimitado los caracteres en Excel, la siguiente pantalla muestra un entorno en el cual los separadores deben incluir tabulaciones y coma, la figura 6-3, muestra la selección de tabulación y coma en el archivo.

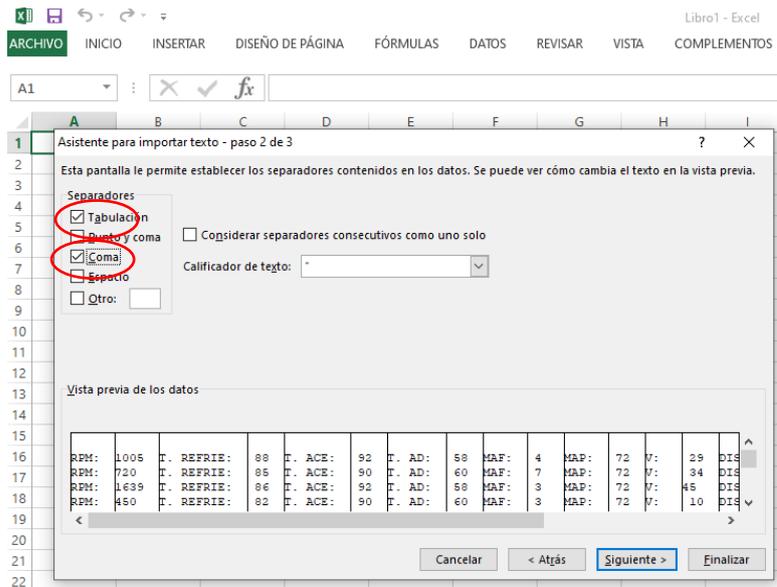


Figura 6-3: Selección de tabulación y coma
Realizado por: Montesdeoca, Andrea, 2020

Para finalizar la configuración en Excel se realiza la siguiente acción, la figura 7-3 muestra el formato de los datos de columnas en general.

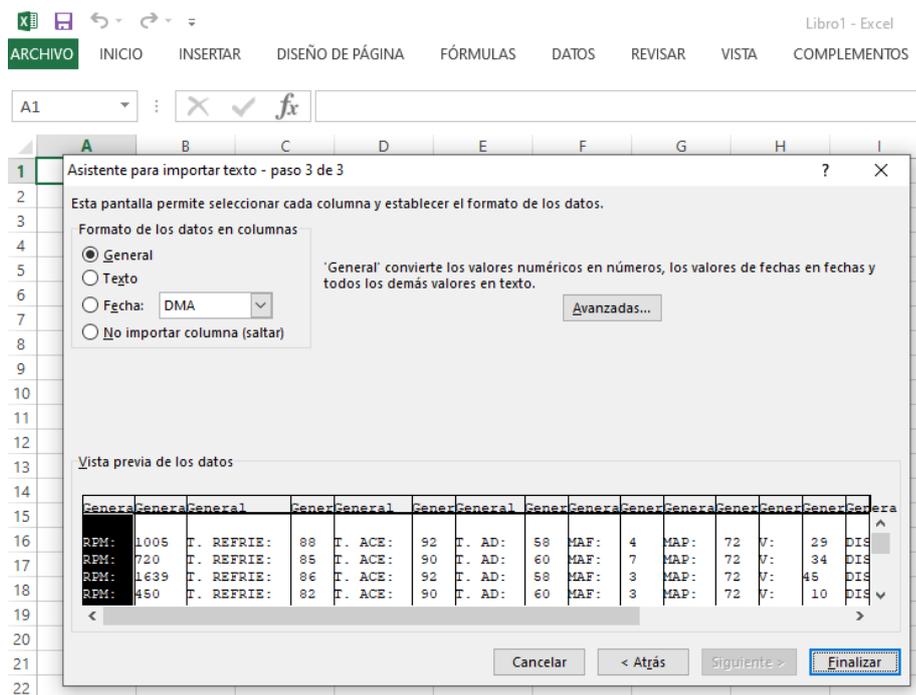


Figura 7-3: Formato de los datos de columnas en general
Realizado por: Montesdeoca, Andrea, 2020

El resultado de los procesos anteriores muestra el datalogger con los nombres y las variables organizadas en columnas, la siguiente figura 8-3 muestra el datalogger final.

DATALOG.TXT - Excel

ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FÓRMULAS DATOS REVISAR VISTA COMPLEMENTOS

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1																					
2	RPM:	1000 T.FREPRE	88 T.ADE	92 T.ADI	58 MAP:		4 MAP:	72 %		29 DIS:	488 POS:A:	B TA:								27	
3	RPM:	700 T.FREPRE	86 T.ADE	90 T.ADI	56 MAP:		7 MAP:	72 %		31 DIS:	488 POS:A:	B TA:									27
4	RPM:	1039 T.FREPRE	84 T.ADE	92 T.ADI	58 MAP:		3 MAP:	72 %		45 DIS:	488 POS:A:	B TA:									27
5	RPM:	100 T.FREPRE	82 T.ADE	90 T.ADI	56 MAP:		3 MAP:	72 %		11 DIS:	488 POS:A:	B TA:									27
6	RPM:	800 T.FREPRE	84 T.ADE	92 T.ADI	58 MAP:		3 MAP:	72 %		35 DIS:	488 POS:A:	B TA:									27
7	RPM:	525 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		3 MAP:	72 %		33 DIS:	488 POS:A:	B TA:									27
8	RPM:	780 T.FREPRE	84 T.ADE	92 T.ADI	61 MAP:		6 MAP:	72 %		48 DIS:	488 POS:A:	B TA:									27
9	RPM:	800 T.FREPRE	84 T.ADE	94 T.ADI	55 MAP:		6 MAP:	72 %		50 DIS:	488 POS:A:	B TA:									27
10	RPM:	1050 T.FREPRE	86 T.ADE	92 T.ADI	53 MAP:		5 MAP:	72 %		49 DIS:	488 POS:A:	B TA:									27
11	RPM:	880 T.FREPRE	88 T.ADE	92 T.ADI	56 MAP:		5 MAP:	72 %		7 DIS:	488 POS:A:	B TA:									27
12	RPM:	1200 T.FREPRE	88 T.ADE	92 T.ADI	67 MAP:		6 MAP:	72 %		49 DIS:	488 POS:A:	B TA:									27
13	RPM:	800 T.FREPRE	86 T.ADE	92 T.ADI	58 MAP:		3 MAP:	72 %		32 DIS:	488 POS:A:	B TA:									28
14	RPM:	780 T.FREPRE	86 T.ADE	92 T.ADI	58 MAP:		3 MAP:	72 %		28 DIS:	488 POS:A:	B TA:									28
15	RPM:	1000 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		3 MAP:	72 %		43 DIS:	488 POS:A:	B TA:									28
16	RPM:	800 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		3 MAP:	72 %		46 DIS:	488 POS:A:	B TA:									28
17	RPM:	800 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		4 MAP:	72 %		54 DIS:	488 POS:A:	B TA:									28
18	RPM:	838 T.FREPRE	86 T.ADE	92 T.ADI	70 MAP:		4 MAP:	72 %		43 DIS:	488 POS:A:	B TA:									28
19	RPM:	1030 T.FREPRE	86 T.ADE	92 T.ADI	70 MAP:		4 MAP:	72 %		47 DIS:	488 POS:A:	B TA:									28
20	RPM:	800 T.FREPRE	86 T.ADE	92 T.ADI	70 MAP:		4 MAP:	72 %		67 DIS:	488 POS:A:	B TA:									28
21	RPM:	1000 T.FREPRE	86 T.ADE	92 T.ADI	70 MAP:		3 MAP:	72 %		69 DIS:	488 POS:A:	B TA:									28
22	RPM:	106 T.FREPRE	86 T.ADE	94 T.ADI	71 MAP:		3 MAP:	72 %		75 DIS:	488 POS:A:	B TA:									28
23	RPM:	1029 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		3 MAP:	72 %		74 DIS:	488 POS:A:	B TA:									28
24	RPM:	320 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		10 MAP:	72 %		38 DIS:	488 POS:A:	B TA:									28
25	RPM:	794 T.FREPRE	86 T.ADE	92 T.ADI	58 MAP:		6 MAP:	72 %		38 DIS:	488 POS:A:	B TA:									28
26	RPM:	780 T.FREPRE	86 T.ADE	92 T.ADI	71 MAP:		6 MAP:	72 %		38 DIS:	488 POS:A:	B TA:									28
27	RPM:	700 T.FREPRE	86 T.ADE	92 T.ADI	71 MAP:		6 MAP:	72 %		45 DIS:	488 POS:A:	B TA:									28
28	RPM:	1040 T.FREPRE	86 T.ADE	92 T.ADI	67 MAP:		3 MAP:	72 %		43 DIS:	488 POS:A:	B TA:									28
29	RPM:	500 T.FREPRE	86 T.ADE	92 T.ADI	65 MAP:		3 MAP:	72 %		62 DIS:	488 POS:A:	B TA:									28
30	RPM:	1006 T.FREPRE	86 T.ADE	92 T.ADI	58 MAP:		3 MAP:	72 %		60 DIS:	488 POS:A:	B TA:									28
31	RPM:	1006 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		3 MAP:	72 %		60 DIS:	488 POS:A:	B TA:									28
32	RPM:	1040 T.FREPRE	86 T.ADE	94 T.ADI	58 MAP:		10 MAP:	72 %		65 DIS:	488 POS:A:	B TA:									28
33	RPM:	2480 T.FREPRE	86 T.ADE	94 T.ADI	70 MAP:		10 MAP:	72 %		80 DIS:	488 POS:A:	B TA:									28
34	RPM:	3043 T.FREPRE	86 T.ADE	94 T.ADI	70 MAP:		11 MAP:	72 %		90 DIS:	488 POS:A:	B TA:									28

Figura 8-3: Datalogger final
 Realizado por: Montesdeoca, Andrea, 2020

3.4 Visualización de resultados en Thingspeak

La plataforma Thingspeak almacena los datos emitidos por los sensores y se las puede visualizar mediante la red internet en cualquier lugar del mundo, para usar la plataforma es necesario iniciar sesión en MatWorks.

La figura 9-3 muestra el proceso de los datos por la red hasta la plataforma Thingspeak.

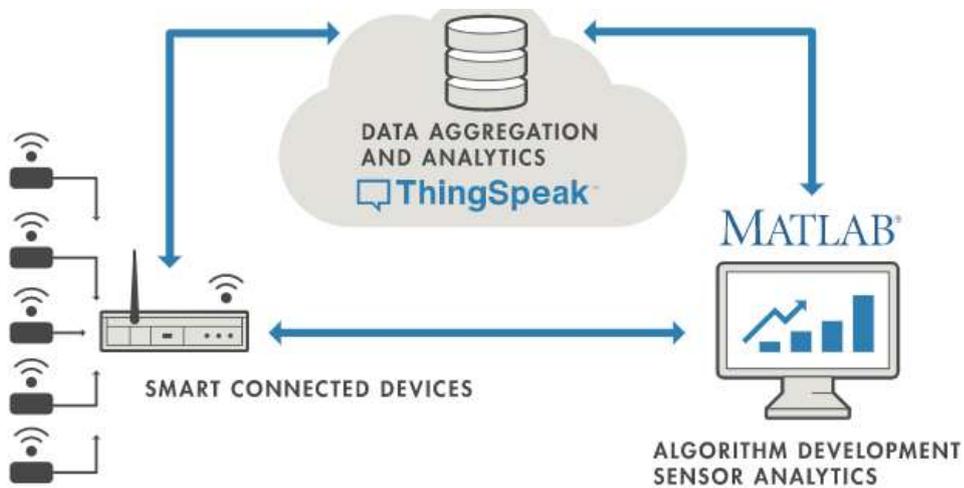


Figura 9-3: Proceso de los datos por la red
 Realizado por: Montesdeoca, Andrea, 2020

3.5 Adquisición de datos del sensor RPM

La figura 10-3 visualiza los datos extraídos con respecto al sensor RPM, el dato nos permite comprobar si las revoluciones actuales a las que giran el árbol de levas y el cigüeñal muestran el rendimiento de funcionalidad del motor.

- **RPM:** 114



Figura 10-3: Visualización de datos del sensor RPM
Realizado por: Montesdeoca, Andrea, 2020

3.5.1 Datos del sensor RPM almacenados en el Datalogger

La tarjeta micro SD mediante el módulo y el Arduino mega, almacenan los datos emitidos por el sensor, la siguiente tabla 1-3 muestra los datos procesados en Excel del sensor RPM.

Tabla 1-3: Datos procesados en Excel del sensor RPM

SENSOR	VALOR
RPM:	1005
RPM:	720
RPM:	1639
RPM:	450
RPM:	1615
RPM:	922
RPM:	780
RPM:	800
RPM:	1630
RPM:	680
RPM:	1200
RPM:	930
RPM:	790
RPM:	1400
RPM:	810
RPM:	810
RPM:	939

RPM:	1230
RPM:	1502

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lo normal leído por un scanner es de 600 a 1800 RPM, afirmando su correcta funcionalidad, la gráfica 1-3 muestra el comportamiento del sensor RPM según sus datos obtenidos.

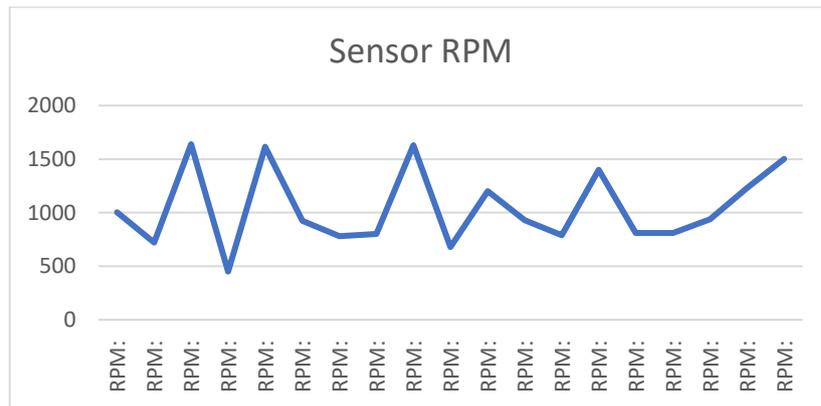


Gráfico 1-3: Etiqueta en eje Y rangos de 600 a 1800 RPM

Realizado por: Montesdeoca, Andrea, 2020

3.5.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor RPM

3.5.2.1 Promedio de los datos del sensor RPM

Para el cálculo del promedio de los datos obtenidos por el sensor RPM que se muestran en la Tabla 1-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{1005 + 720 + 1639 + 450 + 1615 + 922 + 780 + 800 + 1630 + 680 + 1200 + 930 + 790 + 1400 + 810 + 810 + 939 + 1230 + 1502}{19}$$

$$\bar{X} = \frac{19852}{19} = 1044,84$$

El promedio del número de RPM censados es de 1044,84

3.5.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor que se muestran en la Tabla 2-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 2-3: Error absoluto de cada dato obtenido por el sensor RPM

Dato Censado RPM	$X_i - \bar{X}$
1005	39,84
720	324,84
1639	594,16
450	594,84
1615	570,16
922	122,84
780	264,84
800	244,84
1630	585,16
680	364,84
1200	155,16
930	114,84
790	254,84
1400	355,16
810	234,84
810	234,84
939	105,84
1230	185,16
1502	457,16
Error Absoluto	305,484

Fuente: Montesdeoca, Andrea, 2020

El margen de error es de 305,484; por lo que promedio de las RPM censadas es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 1044,84 \pm 305,484$$

3.5.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{305,484}{1044,84} * 100\%$$

$$Error_{relativo} = 0,2923 * 100\%$$

$$Error_{relativo} = 29,23 \%$$

3.5.3 Datos del sensor RPM visualizados en la plataforma Thingspeak.

La plataforma permite una visualización de datos interpretados en gráficas, mediante comunicación GSM los datos son enviados a una nube denominada Thingspeak propia de Matlab. En el Gráfico 2-3 se muestra los datos de RPM en la plataforma Thingspeak.



Gráfico 2-3: Sensor RPM en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.6 Adquisición de datos del sensor temperatura de refrigerante

La pantalla visualiza los datos extraídos con referencia al sensor temperatura de refrigerante, cuyos son capturados desde la unidad de control electrónica de motor (ECU) sobre la temperatura de servicio; se considera que una temperatura adecuada menor a los 95°C considerando como normal.

La siguiente figura 11-3 muestra los datos extraídos del sensor temperatura de refrigerante.



Figura 11-3: Visualización de datos del sensor temperatura de refrigerante
Realizado por: Montesdeoca, Andrea, 2020

3.6.1 Datos del sensor Temperatura de refrigerante almacenados en el Datalogger.

Los datos obtenidos en Excel muestran los valores de sensor de temperatura de refrigerante en la tabla 3-3.

Tabla 3-3: Datos obtenidos en Excel del sensor temperatura de refrigerante

SENSOR	VALOR
T. REFRIE:	88
T. REFRIE:	85
T. REFRIE:	86
T. REFRIE:	82
T. REFRIE:	84
T. REFRIE:	85
T. REFRIE:	84
T. REFRIE:	84
T. REFRIE:	85
T. REFRIE:	86

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lo normal leído por un scanner es de 85°C a 105°C afirmando su correcta funcionalidad.

El siguiente gráfico 3-3 muestra el comportamiento del sensor temperatura de refrigerante con respecto a los datos extraídos.

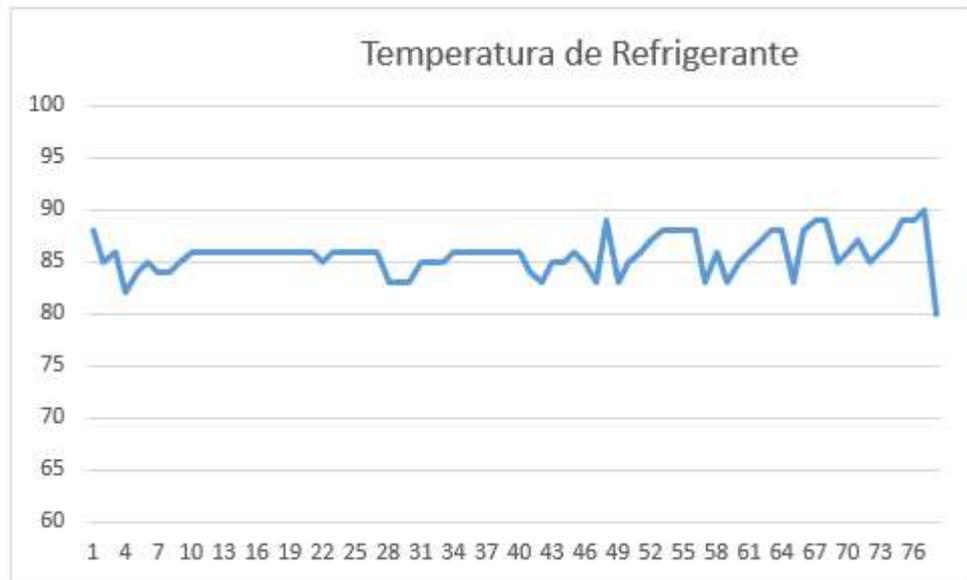


Gráfico 3-3: Etiqueta en eje Y con temperatura del refrigerante entre 85°C a 105°C
 Realizado por: Montesdeoca, Andrea, 2020

3.6.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de refrigerante

3.6.2.1 Promedio de los datos del sensor Temperatura de refrigerante

Para el cálculo del promedio de los datos obtenidos por el sensor Temperatura de refrigerante que se muestran en la Tabla 3-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{88 + 85 + 86 + 82 + 84 + 85 + 84 + 84 + 85 + 86 + 86 + 86 + 86 + 86 + 86 + 86 + 86 + 86 + 86}{19}$$

$$\bar{X} = \frac{1623}{19} = 85,421$$

El promedio de la temperatura del refrigerante es de 85,421 °C.

3.6.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor Temperatura de refrigerante que se muestran en la Tabla 3-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 4-3: Error Absoluto de cada dato obtenido por el sensor Temperatura de refrigerante.

Dato Censado °C	$ X_i - \bar{X} $
88	2,579
85	0,421
86	0,579
82	3,421
84	1,421
85	0,421
84	1,421
84	1,421
85	0,421
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
86	0,579
Error Absoluto	0,9418

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 0,9418, por lo que promedio de la temperatura censada es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 85,421 \pm 0,9418$$

3.6.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{0,9418}{85,421} * 100\%$$

$$Error_{relativo} = 0,011 * 100\%$$

$$Error_{relativo} = 1,1 \%$$

3.6.3 Datos del sensor temperatura de refrigerante visualizados en la plataforma Thingspeak.

El sensor de temperatura de refrigerante al momento de las pruebas es de 85°C, estos datos se muestran en la plataforma Thingspeak de la gráfica 4-3.

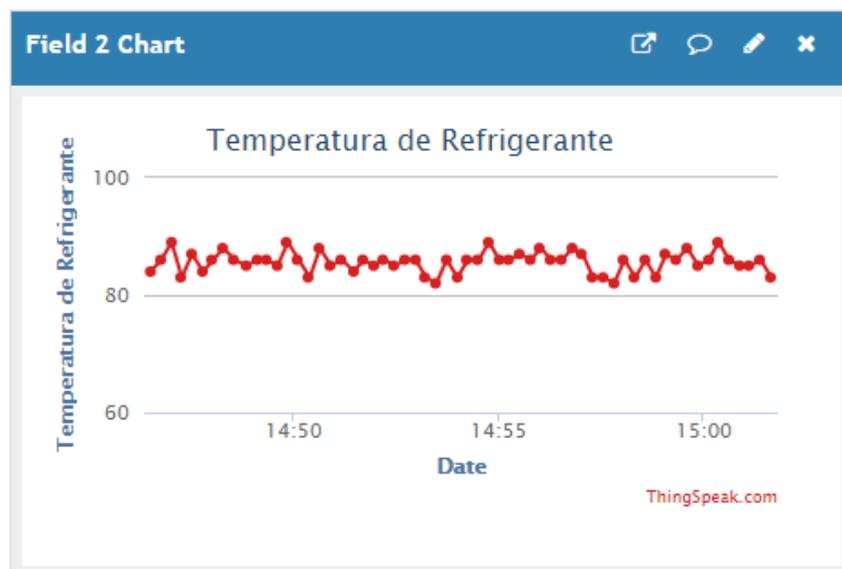


Gráfico 4-3: Sensor Temperatura de refrigerante en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.7 Adquisición de datos de sensor temperatura de aceite

La información extraída del sensor de temperatura de aceite permitirá detectar alteraciones en la puesta en marcha del vehículo Toyota Camry, según el sistema OBDII el dato extraído se muestra en la figura 12-3, cuya temperatura se encuentra en un rango de 92°C apropiado para el funcionamiento del motor.



Figura 12-3: Visualización de datos del sensor temperatura de aceite
 Realizado por: Montesdeoca, Andrea, 2020

3.7.1 Datos del sensor temperatura de Aceite almacenados en el Datalogger

La siguiente tabla 5-3 muestra los datos extraídos por el sensor de temperatura de aceite y procesados en Excel.

Tabla 5-3: Datos procesados en Excel del sensor de temperatura de aceite

SENSOR	VALOR
T. ACE:	92
T. ACE:	90
T. ACE:	92
T. ACE:	90
T. ACE:	92
T. ACE:	94
T. ACE:	92
T. ACE:	94
T. ACE:	92
T. ACE:	94
T. ACE:	94
T. ACE:	94
T. ACE:	92

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) los datos receptados por el mismo no deben sobrepasar los 100°, en esta prueba el sensor recepto datos entre 90°C a 94°C, comprobando su correcta funcionalidad.

Según los datos obtenidos, en la siguiente grafica 5-3 se muestra el comportamiento del sensor de temperatura de aceite

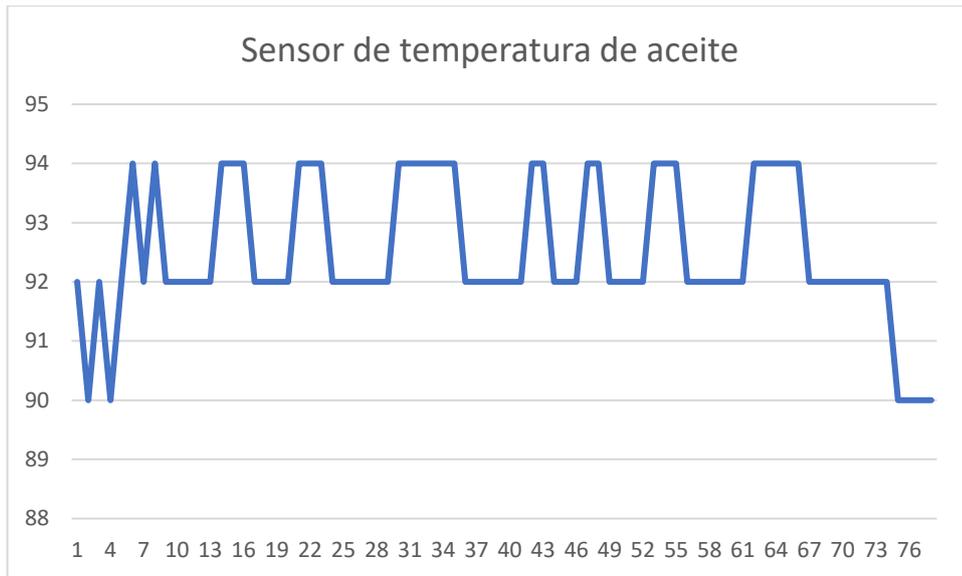


Gráfico 5-3: Etiqueta en eje Y con temperatura del aceite entre 90°C a 94°C
 Realizado por: Montesdeoca, Andrea, 2020

3.7.2 *Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de aceite*

3.7.2.1 *Promedio de los datos del sensor Temperatura de aceite*

Para el cálculo del promedio de los datos obtenidos por el sensor Temperatura de aceite que se muestran en la Tabla 5-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{92 + 90 + 92 + 90 + 92 + 94 + 92 + 94 + 92 + 92 + 92 + 92 + 92 + 94 + 94 + 94 + 92}{17}$$

$$\bar{X} = \frac{1570}{17} = 92,351$$

El promedio de la temperatura del aceite es de 92,351 °C.

3.7.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor Temperatura de aceite que se muestran en la Tabla 5-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 6-3: Error Absoluto de cada dato obtenido por el sensor Temperatura de aceite

Dato Censado °C	$ X_i - \bar{X} $
92	0,35
90	2,35
92	0,35
90	2,35
92	0,35
94	1,65
92	0,35
94	1,65
92	0,35
92	0,35
92	0,35
92	0,35
92	0,35
92	0,35
94	1,65
94	1,65
94	1,65
92	0,35
Error Absoluto	0,9676

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 0,9676, por lo que promedio de la temperatura censada es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 92,351 \pm 0,9676$$

3.7.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{0,9676}{92,351} * 100\%$$

$$Error_{relativo} = 0,0104 * 100\%$$

$$Error_{relativo} = 1,04 \%$$

3.7.3 Datos del sensor temperatura de aceite visualizados en la plataforma Thingspeak.

En la gráfica 6-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak, en la plataforma basada en Matlab el rango de 92°C está dentro del rango normal para el funcionamiento de un motor.

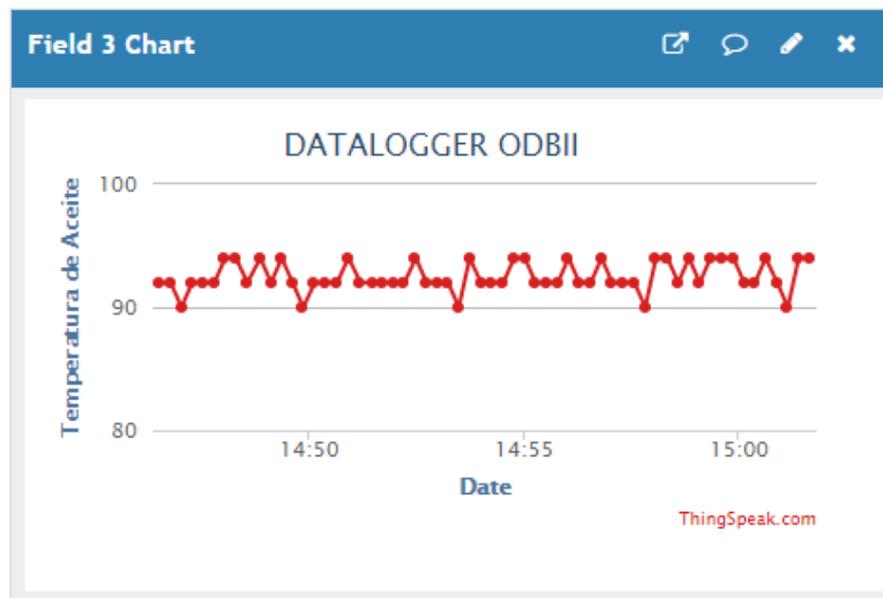


Gráfico 6-3: Sensor Temperatura de Aceite en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.8 Adquisición de datos de sensor temperatura de admisión

La figura 7-3 muestra los datos extraídos del sensor temperatura de admisión, el cual está encargado de monitorear la cantidad y temperatura de aire que ingresa al motor del vehículo, según las pruebas realizadas de muestra un valor de 65°C, el cual según los datos estándar el valor cumple con el rol de funcionamiento.



Gráfico 7-3: Visualización de datos del sensor temperatura de admisión
 Realizado por: Montesdeoca, Andrea, 2020

3.8.1 Datos del sensor temperatura de admisión almacenados en el Datalogger

La tabla 7-3 muestra los datos extraídos por el sensor de temperatura de aceite y procesados en Excel.

Tabla 7-3: Datos procesados en Excel del sensor temperatura de admisión

SENSOR	VALOR
T. AD:	58
T. AD:	60
T. AD:	58
T. AD:	60
T. AD:	58
T. AD:	60
T. AD:	61
T. AD:	63
T. AD:	63
T. AD:	66
T. AD:	67
T. AD:	68
T. AD:	68
T. AD:	68
T. AD:	69
T. AD:	69
T. AD:	70

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lo normal leído por un scanner es de 10°C a 80°C afirmando su correcta funcionalidad; según los datos obtenidos en la gráfica 8-3 se muestra el comportamiento del sensor de temperatura de admisión.

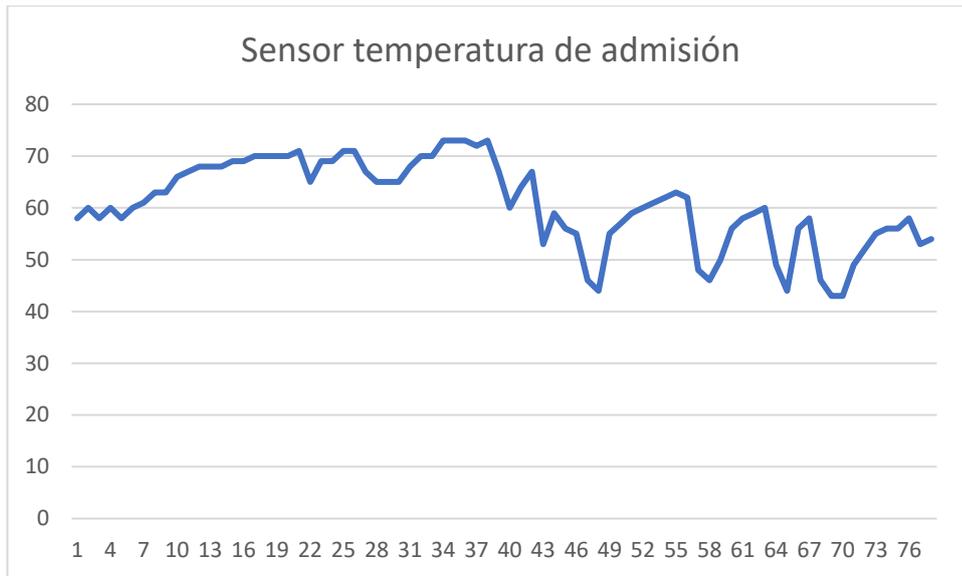


Gráfico 8-3: Etiqueta en eje Y con temperatura de admisión entre 10°C a 80°C
 Realizado por: Montesdeoca, Andrea, 2020

3.8.2 *Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura de admisión*

3.8.2.1 *Promedio de los datos del sensor Temperatura de admisión*

Para el cálculo del promedio de los datos obtenidos por el sensor Temperatura de admisión que se muestran en la Tabla 3-7, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{58 + 60 + 58 + 60 + 58 + 60 + 61 + 63 + 63 + 66 + 67 + 68 + 68 + 68 + 69 + 69 + 70}{17}$$

$$\bar{X} = \frac{1086}{17} = 63,882$$

El promedio de la temperatura de admisión es de 63,882 °C.

3.8.2.2 *Error absoluto*

Para el cálculo del error absoluto de los datos obtenidos por el sensor Temperatura de admisión que se muestran en la Tabla 3-8, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 8-3: Error Absoluto de cada dato obtenido por el sensor temperatura de admisión

Dato Censado °C	$X_i - \bar{X}$
58	5,882
60	3,882
58	5,882
60	3,882
58	5,882
60	3,882
61	2,882
63	0,882
63	0,882
66	2,118
67	3,118
68	4,118
68	4,118
68	4,118
69	5,118
69	5,118
70	6,118
Error Absoluto	3,993

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 3,993, por lo que promedio de la temperatura censada es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 66,883 \pm 3,993$$

3.8.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{3,993}{66,883} * 100\%$$

$$Error_{relativo} = 0,0625 * 100\%$$

$$Error_{relativo} = 6,25 \%$$

3.8.3 Datos del sensor temperatura de admisión visualizados en la plataforma Thingspeak.

En la gráfica 9-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak, en la plataforma basada en Matlab la temperatura esta entre su rango normal.

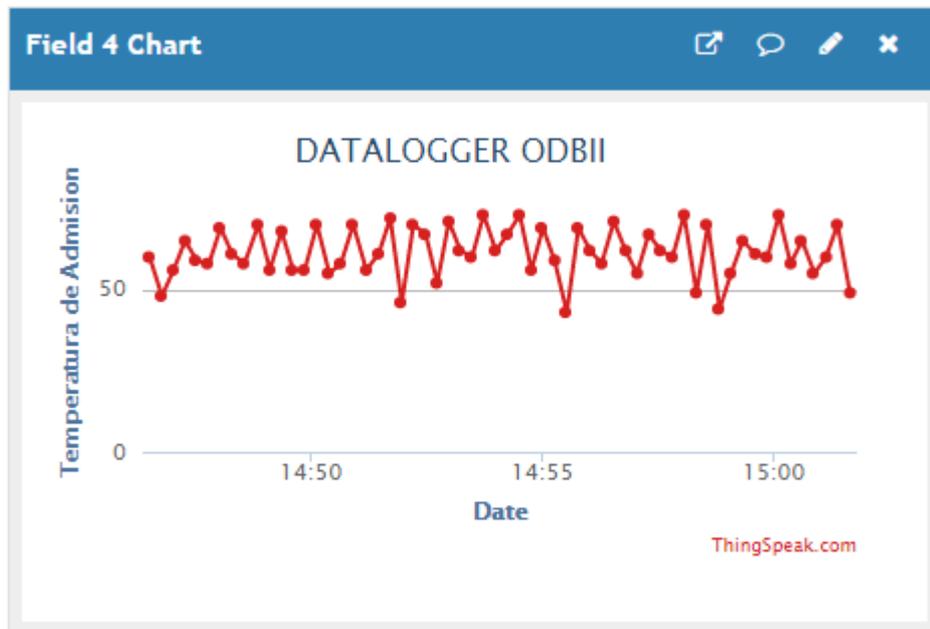


Gráfico 9-3: Sensor temperatura de admisión en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.9 Adquisición de datos de sensor MAF

Los datos adquiridos por el sensor MAF permiten controlar la medición del flujo de aire que ingresa al motor, estos datos adquiridos son enviados por la unidad de control electrónica de motor (ECU) controlando el flujo de aire.

La siguiente figura 13-3 muestra la visualización de dato del sensor MAF.



Figura 13-3: Visualización de datos del sensor MAF
Realizado por: Montesdeoca, Andrea, 2020

3.9.1 Datos del sensor MAF almacenados en el Datalogger

La siguiente tabla 9-3 muestra los datos extraídos por el sensor de MAF procesados en Excel.

Tabla 9-3: Datos procesados en Excel del sensor MAF

SENSOR	VALOR
MAF:	4
MAF:	7
MAF:	3
MAF:	6
MAF:	3
MAF:	4

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lo normal leído por un scanner es de 0 a 35g, afirmando su correcta funcionalidad; Según los datos obtenidos, en la siguiente grafica 10-3 se muestra el comportamiento del sensor MAF.

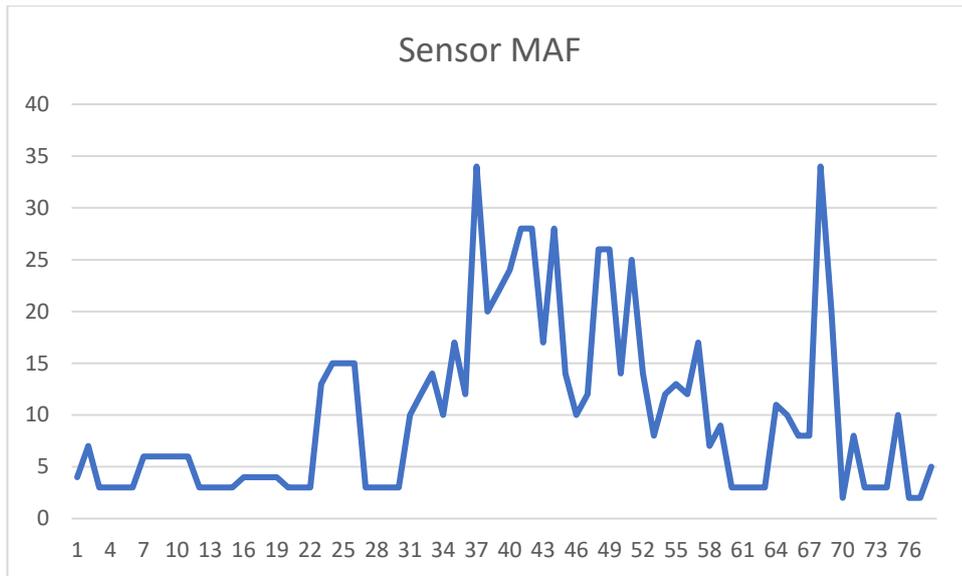


Gráfico 10-3: Etiqueta en eje Y con datos obtenidos por el MAF de 0g a 35g
 Realizado por: Montesdeoca, Andrea, 2020

3.9.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor MAF

3.9.2.1 Promedio de los datos del sensor MAF

Para el cálculo del promedio de los datos obtenidos por el sensor MAF que se muestran en la Tabla 8-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{4 + 7 + 3 + 3 + 3 + 3 + 6 + 6 + 6 + 6 + 6 + 6 + 3 + 3 + 3 + 3 + 4}{16}$$

$$\bar{X} = \frac{69}{16} = 4,312$$

El promedio de los datos obtenidos por el sensor MAF es de 4,31 g.

3.9.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor MAF que se muestran en la Tabla 8-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 10-3: Error Absoluto de cada dato obtenido por el sensor MAF

Dato Censado (g)	$X_i - \bar{X}$
4	0,31
7	2,69
3	1,31
3	1,31
3	1,31
3	1,31
6	1,69
6	1,69
6	1,69
6	1,69
6	1,69
3	1,31
3	1,31
3	1,31
3	1,31
4	0,31
Error Absoluto	1,39

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 1,39, por lo que promedio de los datos censados es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 4,312 \pm 1,39$$

3.9.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{1,39}{3,312} * 100\%$$

$$Error_{relativo} = 0,3223 * 100\%$$

$$Error_{relativo} = 32,23 \%$$

3.9.3 Datos del sensor MAF visualizados en la plataforma Thingspeak.

En la gráfica 11-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak, en la plataforma basada en Matlab muestra el rango del sensor.

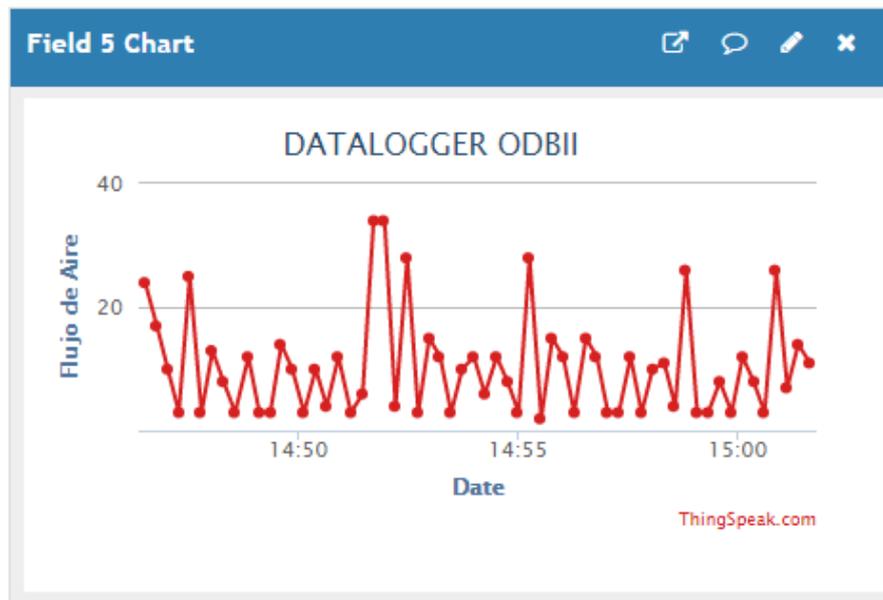


Gráfico 11-3: Sensor MAF en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.10 Adquisición de datos de sensor MAP

Los datos adquiridos por el sensor MAP permiten inspeccionar de manera constante el vacío del múltiple de admisión.

La siguiente figura 14-3 muestra la visualización del valor en el sensor MAP.



Figura 14-3: Visualización de datos del sensor MAP

Realizado por: Montesdeoca, Andrea, 2020

3.10.1 Datos del sensor MAP almacenados en el Datalogger

La siguiente tabla 11-3 muestra los datos extraídos por el sensor de MAP procesados en Excel.

Tabla 11-3: Datos procesados en Excel del sensor MAP

SENSOR	VALOR
MAP:	72

Fuente: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lo normal leído por un scanner es de un máximo de 750mbar equivalentes a 75kpa afirmando su correcta funcionalidad; según los datos obtenidos, en la siguiente grafica 12-3 se muestra el comportamiento del sensor de MAP.

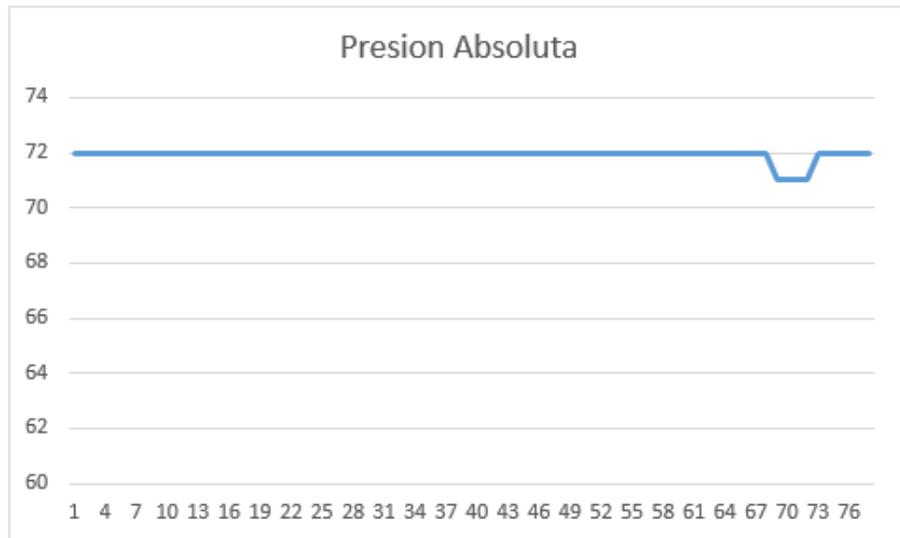


Gráfico 12-3: Etiqueta en eje Y con datos obtenidos por el MAP de 72kpa
Realizado por: Montesdeoca, Andrea, 2020

3.10.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor MAP

3.10.2.1 Promedio de los datos del sensor MAP

Para el cálculo del promedio de los datos obtenidos por el sensor MAP que se muestran en la Tabla 11-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72 + 72}{16}$$

$$\bar{X} = \frac{1158}{16} = 72$$

El promedio de los datos obtenidos por el sensor MAF es de 72 kpa.

3.10.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor MAP que se muestran en la Tabla 3-11, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 12-3: Error Absoluto de cada dato obtenido por el sensor MAP

Dato Censado Kpa	$X_i - \bar{X}$
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
72	0
Error Absoluto	0

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 0, por lo que promedio de los datos censados es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 72 \pm 0$$

3.10.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{0}{72} * 100\%$$

$$Error_{relativo} = 0 * 100\%$$

$$Error_{relativo} = 0 \%$$

3.10.3 Datos del sensor MAP visualizados en la plataforma Thingspeak.

En la gráfica 13-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak en la plataforma basada en Matlab.

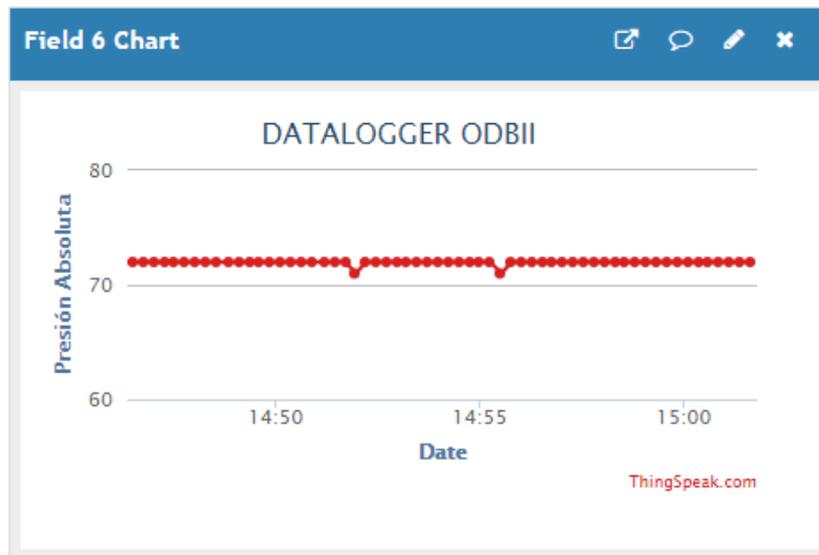


Gráfico 13-3: Sensor MAP en Thingspeak

Realizado por: Montesdeoca, Andrea, 2020

3.11 Adquisición de datos de sensor de velocidad

El dato adquirido por el sensor muestra la velocidad de desplazamiento del vehículo, esta señal es capturada desde la unidad de control electrónica de motor (ECU); la siguiente figura 15-3 muestra la visualización de dato del sensor de velocidad.



Figura 15-3: Visualización de datos del sensor de velocidad

Realizado por: Montesdeoca, Andrea, 2020

3.11.1 Datos del sensor de velocidad almacenados en el Datalogger

La siguiente tabla 13-3 muestra los datos extraídos por el sensor de velocidad y procesados en Excel.

Tabla 13-3: Datos procesados en Excel del sensor de velocidad

SENSOR	VALOR
V:	29
V:	34
V:	45
V:	10
V:	35
V:	32
V:	46
V:	52
V:	45
V:	11
V:	45
V:	32
V:	28
V:	43
V:	46
V:	54
V:	43

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) su medida varía de acuerdo a la aceleración suministrada por el piloto del vehículo está dentro de un rango normal y permitido entre 10 a 140 km; según los datos obtenidos en la siguiente grafica 14-3 se muestran el comportamiento del sensor de velocidad.

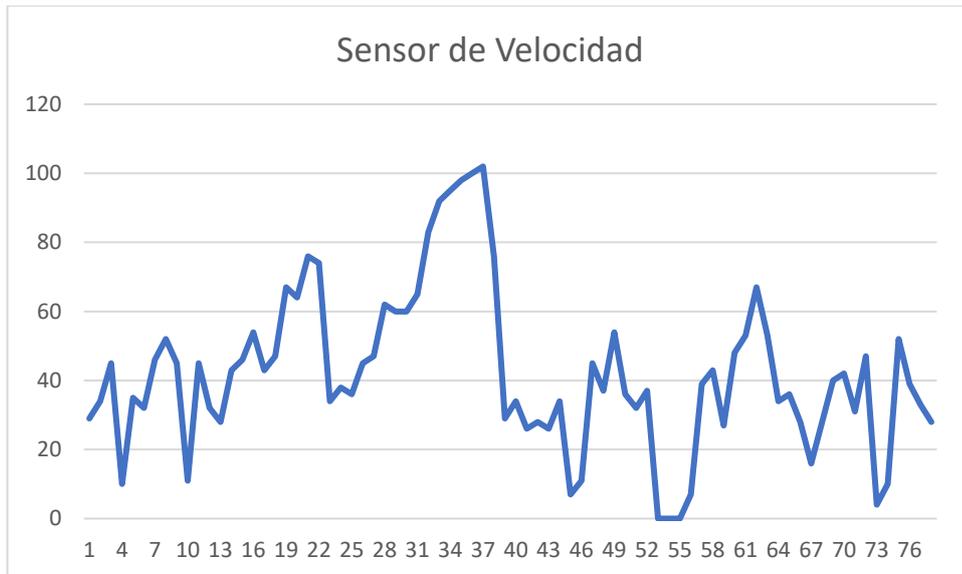


Gráfico 14-3: Etiqueta en eje Y con datos obtenidos de velocidad de 10km a 140km
Realizado por: Montesdeoca, Andrea, 2020

3.11.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor de velocidad

3.11.2.1 Promedio de los datos del sensor de velocidad

Para el cálculo del promedio de los datos obtenidos por el sensor de velocidad que se muestran en la Tabla 13-4, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{29 + 34 + 45 + 10 + 35 + 32 + 46 + 52 + 45 + 11 + 45 + 32 + 28 + 43 + 46 + 54 + 43}{17}$$

$$\bar{X} = \frac{630}{17} = 37,058$$

El promedio de los datos obtenidos por el sensor de velocidad es de 37,058 km/h.

3.11.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor de velocidad que se muestran en la Tabla 14-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 14-3: Error Absoluto de cada dato obtenido por el sensor de velocidad

Dato Censado Km/h	$ X_i - \bar{X} $
29	43
34	38
45	27
10	62
35	37
32	40
46	26
52	20
45	27
11	61
45	27
32	40
28	44
43	29
46	26
54	18
43	29
Error Absoluto	10,055

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 10,055, por lo que promedio de los datos censados es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 37,058 \pm 10,055$$

3.11.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{10,055}{37,058} * 100\%$$

$$Error_{relativo} = 0,2713 * 100\%$$

$$Error_{relativo} = 27,13 \%$$

3.11.3 Datos del sensor de velocidad visualizados en la plataforma Thingspeak.

En la gráfica 15-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak de la plataforma basada en Matlab.

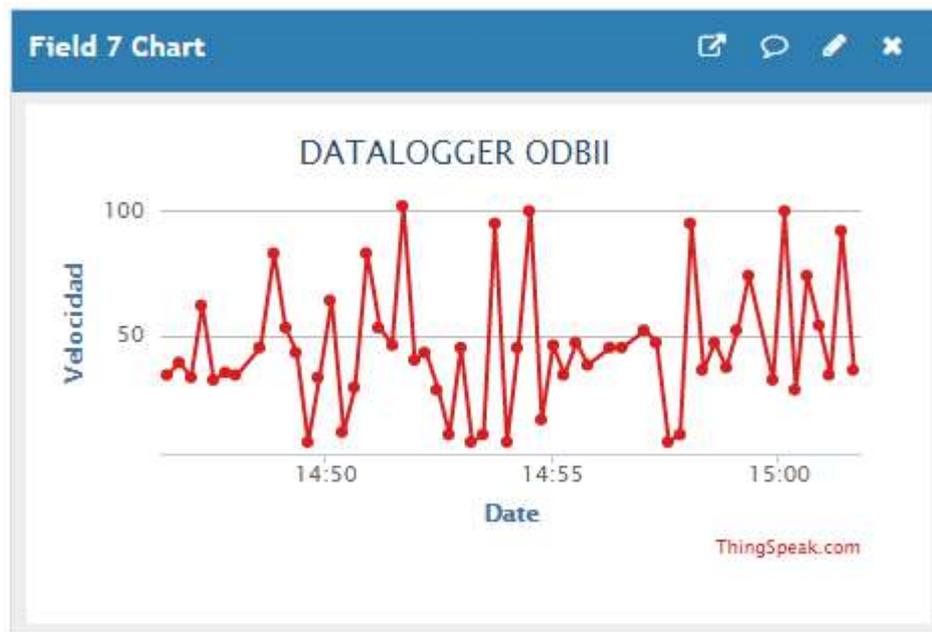


Gráfico 15-3: Sensor de velocidad en Thingspeak

Realizado por: Montesdeoca, Andrea, 2020

3.12 Adquisición de datos de la distancia recorrida

Los datos adquiridos, muestran la distancia recorrida del vehículo en Km, la siguiente figura 3-16 muestra la visualización de la distancia recorrida por el vehículo.

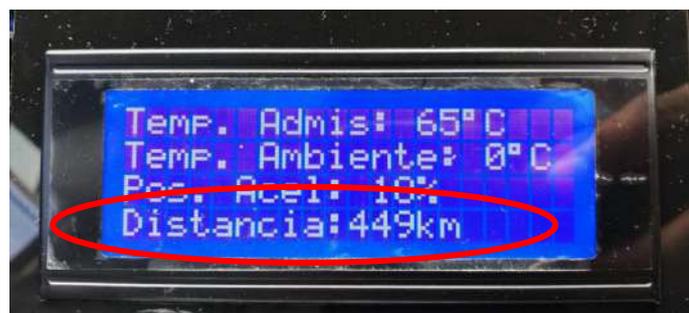


Figura 16-3: Visualización de la distancia recorrida

Realizado por: Montesdeoca, Andrea, 2020

3.12.1 Datos del sensor de distancia almacenados en el Datalogger

La siguiente tabla 15-3 muestra los datos extraídos de la distancia recorrida por el vehículo y procesados en Excel.

Tabla 15-3: Datos procesados en Excel de la distancia recorrida por el vehículo.

SENSOR	VALOR
DIS:	448
DIS:	488
DIS:	488
DIS:	488
DIS:	449

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) lee y guarda los datos de según los kilometro recorridos por el vehículo, en este caso un recorrido de 449km; este varía según la distancia propuesta por el usuario, es por ello que se afirma su correcta funcionalidad; según los datos obtenidos, en la siguiente grafica 16-3 se muestra el comportamiento de la distancia recorrida por el vehículo

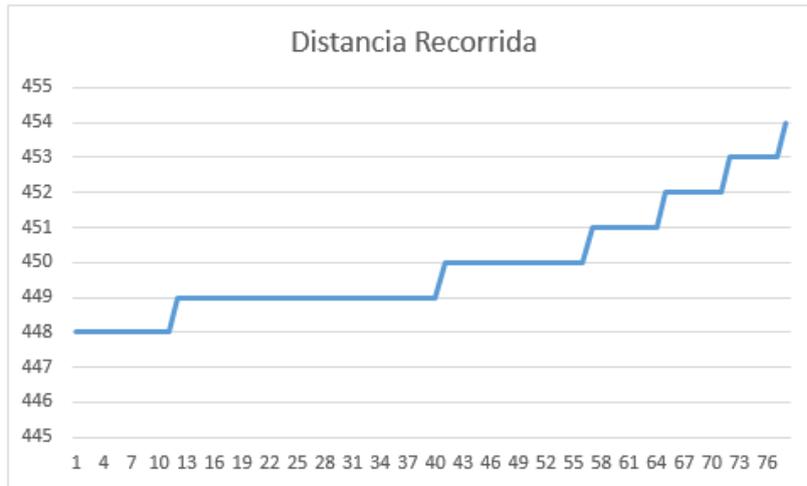


Gráfico 16-3: Etiqueta en eje Y con datos obtenidos de distancia de 449km
Realizado por: Montesdeoca, Andrea, 2020

3.12.2 Cálculo del promedio, error absoluto y relativo de los datos de la distancia recorrida

3.12.2.1 Promedio de los datos de la distancia recorrida

Para el cálculo del promedio de los datos obtenidos de la distancia recorrida que se muestran en la Tabla 15-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{448 + 448 + 448 + 448 + 448 + 448 + 448 + 448 + 448 + 488 + 488 + 488 + 449 + 449 + 449 + 449 + 449 + 449}{17}$$

$$\bar{X} = \frac{7742}{17} = 455,411$$

El promedio de los datos obtenidos por el sensor de la distancia recorrida es de 455,411 km.

3.12.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor de la distancia recorrida que se muestran en la Tabla 16-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 16-3: Error Absoluto de los datos obtenido de la distancia recorrida

Dato Censado Km	$X_i - \bar{X}$
448	7,411
448	7,411
448	7,411
448	7,411
448	7,411
448	7,411
448	7,411
448	7,411
448	7,411
488	32,589
488	32,589
488	32,589
449	6,411
449	6,411
449	6,411
449	6,411
449	6,411
449	6,411
Error Absoluto	11,501

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 11,501, por lo que promedio de los datos censados es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 455,411 \pm 11,501$$

3.12.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{11,501}{455,411} * 100\%$$

$$Error_{relativo} = 0,0252 * 100\%$$

$$Error_{relativo} = 2,52 \%$$

3.12.3 Datos del sensor de distancia visualizados en la plataforma Thingspeak.

En la gráfica 17-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak de la plataforma basada en Matlab.

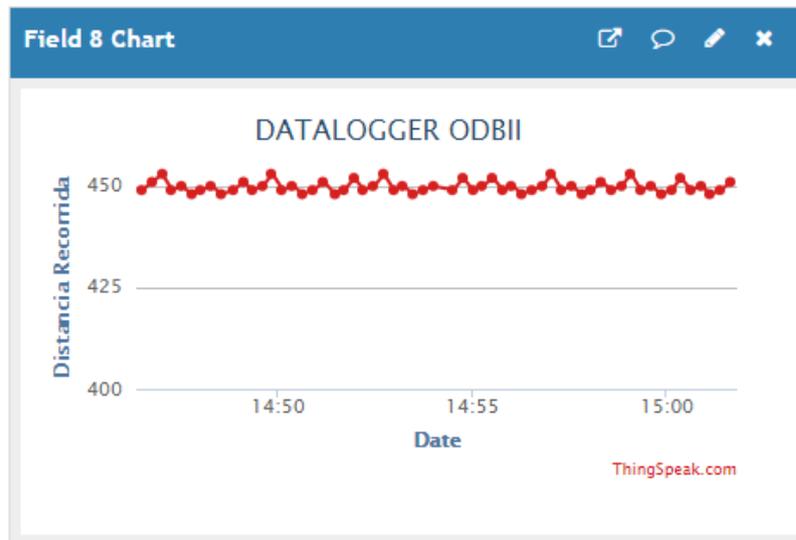


Gráfico 17-3: Sensor de distancia en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.13 Adquisición de datos del sensor posición acelerador

Los datos adquiridos por el sensor son los encargados de regular el ingreso del flujo de aire en las cantidades necesarias, la siguiente figura 17-3 muestra la visualización del dato en el sensor de distancia.



Figura 17-3: Visualización del dato en el sensor posición acelerado
Realizado por: Montesdeoca, Andrea, 2020

3.13.1 Datos del sensor posición acelerador almacenados en el Datalogger

La siguiente tabla 17-3 muestra los datos extraídos por el sensor posición acelerado, procesados en Excel.

Tabla 17-3: Datos procesados en Excel del sensor posición aceleradora

SENSOR	VALOR
POS.A:	18
POS.A:	18
POS.A:	18
POS.A:	16
POS.A:	14
POS.A:	14
POS.A:	16

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un rango de funcionamiento óptimo. Según (Werner, 2015) interpreta la entrada del flujo de aire requerido por el motor en porcentaje, en esta prueba el motor recibe un 16% necesarios; en la siguiente gráfica 18-3 se muestra el comportamiento del sensor posición acelerador.

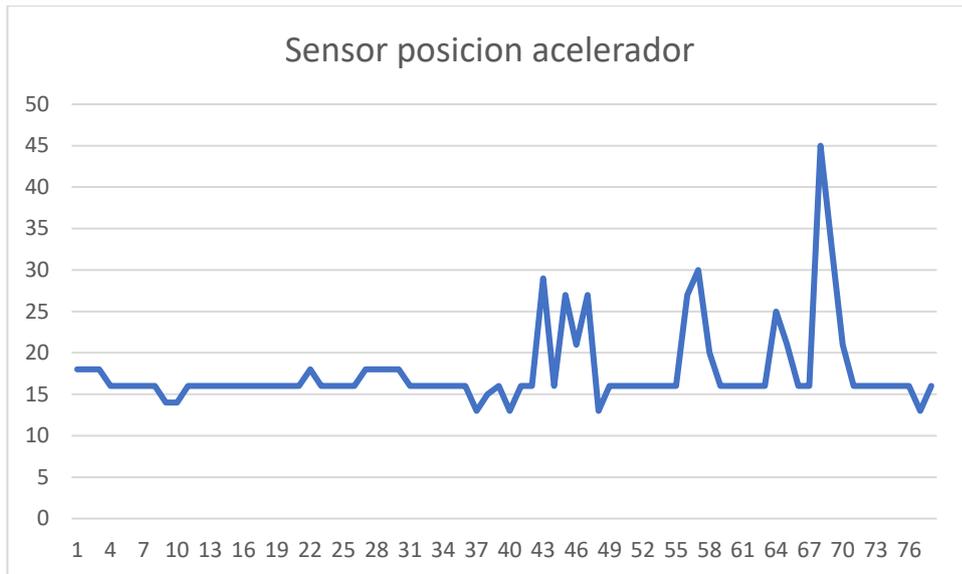


Gráfico 18-3: Etiqueta en eje Y con datos obtenidos del sensor posición acelerador
Fuente: Montesdeoca, Andrea, 2020

3.13.2 *Cálculo del promedio, error absoluto y relativo de los datos del sensor de posición del acelerador*

3.13.2.1 *Promedio de los datos del sensor de posición del acelerador*

Para el cálculo del promedio de los datos obtenidos por el sensor posición acelerador que se muestran en la Tabla 3-17 se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{18 + 18 + 18 + 16 + 16 + 16 + 16 + 16 + 14 + 14 + 16 + 16 + 16 + 16 + 16 + 16}{16}$$

$$\bar{X} = \frac{258}{16} = 16,125$$

El promedio de los datos obtenidos por el sensor de la posición de aceleración es de 16,125%.

3.13.2.2 *Error absoluto*

Para el cálculo del error absoluto de los datos obtenidos por el sensor de posición de aceleración que se muestran en la Tabla 18-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 18-3: Error Absoluto de cada dato obtenido por el sensor de posición de aceleración

Dato Censado %	$X_i - \bar{X}$
18	1,875
18	1,875
18	1,875
16	0,125
16	0,125
16	0,125
16	0,125
16	0,125
14	2,125
14	2,125
16	0,125
16	0,125
16	0,125
16	0,125
16	0,125
16	0,125
16	0,125
Error Absoluto	0,703

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 0,703, por lo que promedio de los datos censados es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 16,125 \pm 0,703$$

3.13.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{0,703}{16,125} * 100\%$$

$$Error_{relativo} = 0,0436 * 100\%$$

$$Error_{relativo} = 4,36 \%$$

3.13.3 Datos del sensor de posición acelerador visualizados en la plataforma Thingspeak.

En la gráfica 19-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak de la plataforma basada en Matlab.

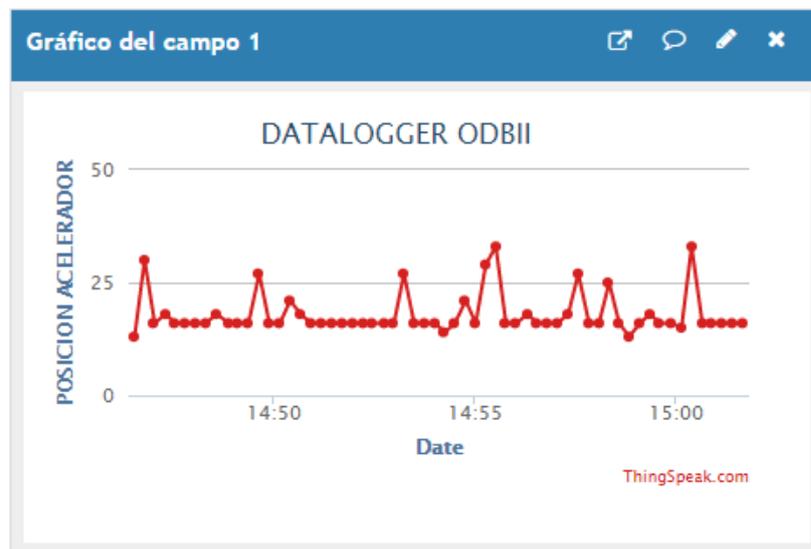


Gráfico 19-3: Sensor de posición acelerador en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.14 Adquisición de datos del sensor temperatura ambiente

Los datos mostrados en la pantalla determinan la temperatura que hay en el exterior del vehículo con el fin de tomar datos informativos y mostrárselo al conductor.

La figura 18-3 a continuación muestra la visualización de datos del sensor temperatura ambiente.



Figura 18-3: Visualización de datos del sensor temperatura ambiente
 Realizado por: Montesdeoca, Andrea, 2020

3.14.1 Datos del sensor temperatura ambiente almacenados en el Datalogger

La siguiente tabla 19-3 muestra los datos extraídos por el sensor de temperatura ambiente y procesada en Excel.

Tabla 19-3: Datos procesados en Excel del sensor temperatura ambiente

SENSOR	VALOR
TA:	27
TA:	28

Realizado por: Montesdeoca, Andrea, 2020

El sensor muestra un funcionamiento óptimo. Según (Werner, 2015) el rango de medición es de -20 a 80°C, en las pruebas realizadas el rango varía entre los 27° y 28°C, comprobando la correcta

funcionalidad del mismo; según los datos obtenidos, en la siguiente grafica 20-3 se muestra el comportamiento del sensor temperatura ambiente.

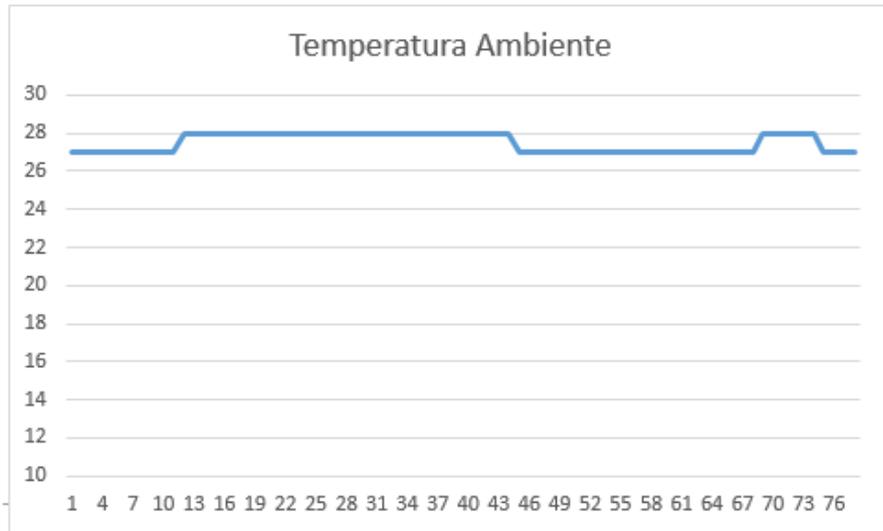


Gráfico 20-3: Etiqueta en eje Y con datos obtenidos de la temperatura de 27°C a 28°C
Realizado por: Montesdeoca, Andrea, 2020

3.14.2 Cálculo del promedio, error absoluto y relativo de los datos del sensor de Temperatura ambiente

3.14.2.1 Promedio de los datos del sensor Temperatura ambiente

Para el cálculo del promedio de los datos obtenidos por el sensor Temperatura ambiente que se muestran en la Tabla 19-3, se aplica la siguiente fórmula.

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 28 + 28 + 28 + 28 + 28 + 28 + 28}{18}$$

$$\bar{X} = \frac{493}{18} = 27,588$$

El promedio de la temperatura ambiente es de 27,588 °C.

3.14.2.2 Error absoluto

Para el cálculo del error absoluto de los datos obtenidos por el sensor Temperatura ambiente que se muestran en la Tabla 20-3, se aplica la siguiente fórmula.

$$Error_{Abs} = |X_i - \bar{X}|$$

Donde:

X_i : Dato sensado por el sensor

\bar{X} : Promedio calculado

Tabla 20-3: Error Absoluto de cada dato obtenido por el sensor Temperatura ambiente

Dato Censado °C	$X_i - \bar{X}$
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
27	0,388
28	0,612
28	0,612
28	0,612
28	0,612
28	0,612
28	0,612
Error Absoluto	0,475

Realizado por: Montesdeoca, Andrea, 2020

El error absoluto es de 0,475, por lo que promedio de la temperatura censada es de:

$$\text{Margen de error} = \bar{X} \pm \text{Error Absoluto}$$

$$\text{Margen de error} = 27,388 \pm 0,475$$

3.14.2.3 Cálculo del error relativo

El error relativo está representado en porcentaje y este dado por la siguiente ecuación.

$$Error_{relativo} = \frac{Error_{Abs}}{\bar{X}} * 100\%$$

$$Error_{relativo} = \frac{0,475}{27,388} * 100\%$$

$$Error_{relativo} = 0,017 * 100\%$$

$$Error_{relativo} = 1,7 \%$$

Para determinar el error relativo del prototipo se calculó el promedio de los errores absolutos de cada sensor analizado, obteniendo un error absoluto de 10,55%.

3.14.3 Datos del sensor Temperatura ambiente visualizados en la plataforma Thingspeak.

En la gráfica 21-3 se observa el comportamiento del sensor almacenado en la nube Thingspeak de la plataforma basada en Matlab.

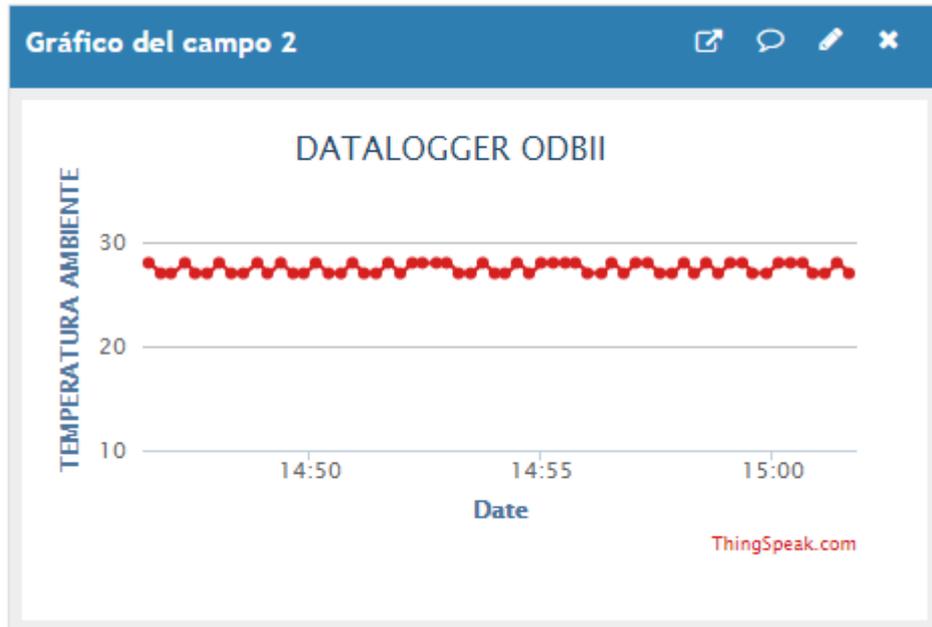


Gráfico 21-3: Sensor de temperatura ambiente en Thingspeak
Realizado por: Montesdeoca, Andrea, 2020

3.15 Análisis de consumo del Prototipo

Se realiza un cálculo de consumo general de corriente del prototipo en base a la composición de sus módulos integrados, para ello se realizó la siguiente tabla 21-3 la cual muestra los elementos electrónicos utilizados.

Tabla 21-3: Elementos electrónicos utilizados

No.	Elemento/módulo	Potencia	Corriente	Voltaje
1	Arduino Mega	0.2 W	20 mA	5 V
2	Módulo GSM	0.0155 W	15 mA	3.4 V
3	Micro SD	0,01W	10 mA	5 V
4	Pantalla LCD	0.025W	25 mA	5 V
5	Sistema OBD II	0.1W	100 mA	12 V
TOTAL		0.170W	170 mA	-

Realizado por: Montesdeoca, Andrea, 2020

La tabla 21-3, muestra los módulos integrados al prototipo y su consumo total de 170 mA, al valor se realiza un sobredimensionamiento del 10% de corriente, para ello se analiza la siguiente ecuación:

Ecuación 3: Ecuación del consumo total

$$I_{Total} = (Corriente consumida)X (10\% de sobredimensionamiento)$$

$$I_{Total} = 170mA * 1.1$$

$$I_{Total} = 0.17A * 1.1$$

$$I_{Total} = 0.187A$$

El consumo total del prototipo es de 0.187A amperios, según este análisis se considera que no afecta a la batería del vehículo al momento de su conexión; para corroborar el cálculo realizado se realizó la medición del consumo con una fuente de corriente y voltaje como se muestra en la figura 19-3 los datos arrojados son:

- Consumo: 0.19
- Voltaje de alimentación: 12.4



Figura 19-3: Pruebas de consumo del prototipo
 Realizado por: Montesdeoca, Andrea, 2020

3.16 Análisis de costos del prototipo

Para la implementación de prototipo se utilizaron elementos y módulos electrónicos cuya función cumplen con cada uno de los requerimientos establecidos para el funcionamiento, la siguiente tabla 22-3 muestra los costos de fabricación del prototipo.

Tabla 22-3: Costos de fabricación del prototipo

CANTIDAD	MATERIAL	VALOR USD
1	Arduino Mega	20.00 USD
1	Fuente LM2596	15.00 USD
1	Módulo GSM	15.00 USD
1	Micro SD	5.00 USD
1	Pantalla LCD	16.00 USD
1	Sistema OBD II	400.00 USD
1	Impresión 3D	40.00 USD
1	Fabricación PCB	50.00 USD
1	Elementos varios	50.00 USD
1	Mano de obra	150.00 USD
Total		761.00 USD

Realizado por: Montesdeoca, Andrea, 2020

La tabla 23-3 muestra los costos empleados en la construcción del prototipo, se realiza un análisis en porcentajes para verificar cual es el valor más significativo en la construcción.

Tabla 23-3: Tabla de costos para realización de porcentajes

CANTIDAD	MATERIAL	VALOR	PORCENTAJE
1	Arduino Mega	20	3%
1	Fuente	15	2%
1	Módulo GSM	15	2%
1	Micro SD	5	1%
1	Pantalla LCD	16	2%
1	Sistema OBDII	400	53%
1	Impresión 3D	40	5%
1	Fabricación PCB	50	7%
1	Elementos varios	50	7%
1	Mano de Obra	150	20%
TOTAL		761	100%

Realizado por: Montesdeoca, Andrea, 2020

Para visualizar el porcentaje en los gastos de fabricación del prototipo, se toma como referencia el 100% equivalente al valor total de los costos; es así como se aprecia que el valor más significativo dentro la construcción del prototipo electrónico para registros telemáticos y detección de fallos del motor es el sistema OBDII equivalente al 53% del valor total del costo de todos los materiales.

3.17 Análisis comparativo

Se realiza un análisis comparativo entre los datos obtenidos por el prototipo y un scanner automotriz profesional de marca launch x431 pro mini, para ello se toma como ejemplo 3 datos de sensores

- RPM.
- MAP.
- SENSOR DE TEMPERATURA DE REFRIGERANTE.

La conexión del escáner automotriz se la efectúa con el vehículo encendido, de la misma manera que el conector OBDII del Arduino; caso contrario la unidad electrónica de motor no emite ningún dato de información.

La figura 20-3 a continuación muestra los diferentes datos emitidos por la unidad de control electrónica de motor (ECU) mediante el scanner launch x431 pro mini.

Nombre	Valor	Valor de estándar	Unidad
Posición de motor de control de aire de ralenti	36	0 - 100	%
Posición del Acelerador	18	0 - 100	%
Posición del Acelerador	3.10	0.00 - 5.00	V
Sensor MAP (Presión absoluta colector)	36	0 - 255	Kpa
Sensor MAP (Presión absoluta colector)	8.10	0.00 - 5.00	V
Velocidad ralenti deseada	800	0 - 8000	rpm

Figura 20-3: Datos emitidos por la ECU
 Realizado por: Montesdeoca, Andrea, 2020

3.17.1 Comparación de datos del sensor RPM

Se realiza una comparación con el scanner launch x431 pro mini y el prototipo datalogger del sensor RPM, la siguiente figura 21-3 muestra los datos obtenidos por el scanner.

Nombre	Valor	Unidad
Accionador del control del régimen de ralenti	39.50	%
Carga del Motor	13.10	%
Par real	9.08	%
Posición del acelerador	0	%
Sensor de Presión Absoluta del colector	8.05	V
Sensor de temperatura del aire de admisión	32.50	degree C
Sensor de temperatura del refrigerante del motor	43.50	degree C
Velocidad objetivo de ralenti	700	rpm

Figura 21-3: Datos del sensor RPM obtenidos por el scanner
 Realizado por: Montesdeoca, Andrea, 2020

Se observa que el dato obtenido por el sensor RPM, que en el caso del scanner launch x431 pro mini, lo reconoce como la velocidad objetivo de relanti, marca un valor de 720 rpm.

Se realiza el mismo proceso de toma de datos con el prototipo OBDII, el mismo que en la figura 22-3 muestra los datos obtenidos.



Figura 22-3: Datos del sensor RPM Obtenidos mediante OBDII

Realizado por: Montesdeoca, Andrea, 2020

Se observa que mediante el análisis efectuado con un scanner automotriz launch x431 pro mini y el prototipo datalogger OBDII, emiten los mismos datos, demostrando así que los datos obtenidos de la unidad de control electrónica de motor son iguales, es decir, el prototipo cumple con su función de enviar información y recibir datos al igual que un scanner profesional.

La ECU es un ordenador automotriz que mediante sus protocolos de comunicación puede enviar y recibir datos para interacción entre los sensores y el usuario; la ventaja de la unidad electrónica de motor es que dentro de sus funciones es procesar y acondicionar las señales proporcionadas por los sensores integrados al motor, por lo que se obtiene valores iguales al momento del análisis con los equipos.

3.17.2 Comparación de datos del sensor MAP

Se realizó una comparación con el scanner y el prototipo datalogger del sensor MAP, la siguiente figura 23-3 muestra los datos obtenidos por el scanner launch x431 pro mini.

Nombre	Valor	Valor de estándar	Unidad
Posición de motor de control de aire de ralenti	36	0 - 100	%
Posición del Acelerador	18	0 - 100	%
Posición del Acelerador	3.30	0.00 - 5.00	V
Sensor MAP (Presión absoluta colector)	30	0 - 255	Kpa
Sensor MAP (Presión absoluta colector)	3.40	0.00 - 5.00	V
Velocidad ralenti deseada	800	0 - 8000	rpm

Figura 23-3: Datos del sensor MAP obtenidos
Realizado por: Montesdeoca, Andrea, 2020

Se observa en la figura anterior que el dato obtenido por el sensor MAP con el scanner launch x431 pro mini marca un valor de 30kpa.

Se realiza el mismo proceso de toma de datos con el prototipo OBDII, el mismo que en la figura 24-3 muestra los datos obtenidos.

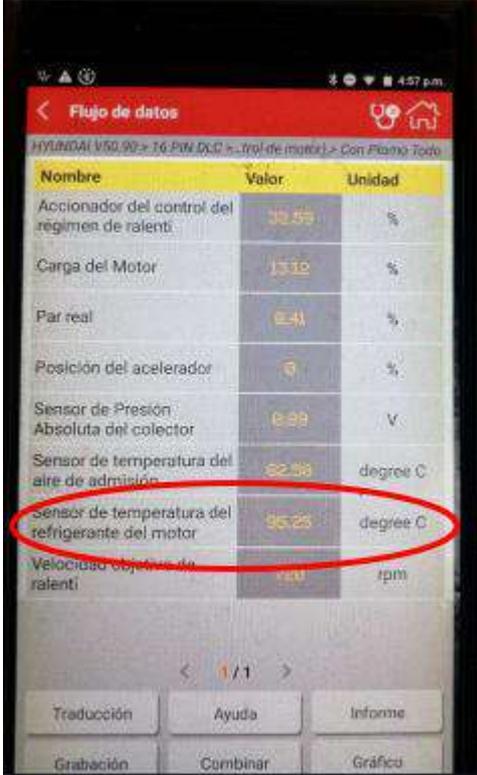


Figura 24-3: Datos del sensor MAP Obtenidos mediante OBDII
Realizado por: Montesdeoca, Andrea, 2020

Se observa que mediante el análisis efectuado con un scanner automotriz y el prototipo Datalogger OBDII, emiten los mismos datos demostrando así que los datos obtenidos de la unidad de control electrónica de motor son iguales, es decir, el prototipo cumple con su función de enviar información y recibir datos al igual que un scanner profesional.

3.17.3 Comparación de datos del sensor Temperatura de refrigerante

Se realiza una comparación con el scanner y el prototipo datalogger del sensor Temperatura de refrigerante, la siguiente figura 25-3 muestra los datos obtenidos por el scanner launch x431 pro mini.



Nombre	Valor	Unidad
Accionador del control del regimen de ralenti	30.55	%
Carga del Motor	17.10	%
Par real	6.41	%
Posición del acelerador	0	%
Sensor de Presión Absoluta del colector	0.99	V
Sensor de temperatura del aire de admisión	92.50	degree C
Sensor de temperatura del refrigerante del motor	95.25	degree C
Velocidad objetivo de ralenti	800	rpm

Figura 25-3: Datos del sensor temperatura de refrigerantes
Realizado por: Montesdeoca, Andrea, 2020

Se observa en la figura anterior que el dato obtenido por el sensor temperatura de refrigerante con el scanner launch x431 pro mini marca un valor de 95.5°C.

Se realiza el mismo proceso de toma de datos con el prototipo OBDII, el mismo que en la figura 26-3 muestra los datos obtenidos.



RPM: 720	MAF: 44g
MAR: 30kPa	V: 34km/h
Temp. Refrig: 96°C	
Temp. Aceite: 95°C	

Figura 26-3: Datos del sensor temperatura de refrigerante
Realizado por: Montesdeoca, Andrea, 2020

3.18 Análisis comparativo de equipos

Se realiza un análisis comparativo entre equipos profesionales tales como scanner automotrices con el prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistema OBD II, dichos scanner automotrices en su mayoría son de difícil adquisición, debido a su importación y altos impuestos que conlleva.

La siguiente tabla muestra los scanner más utilizados y de mayor relevancia en el campo automotriz al momento del análisis de sensores integrados al motor del vehículo.

Tabla 24-3: Scanner más utilizados.

ID	MODELO	TIPO	COSTO
1	<p>LAUNCH x431</p> 	SCANNER	1,199 USD
2	<p>AUTEL MAXICOM MK906BT</p> 	SCANNER	1,269 USD
3	<p>AUTEL AUTOLINK AL301</p> 	SCANNER	3,141 USD
4	<p>AUTEL MAXI IM508</p>	SCANNER	1,369 USD

			
5	<p>AUTEL IM608</p> 	SCANNER	3,299 USD
6	<p>X431 V + 4,0</p> 	SCANNER	1,199 USD
7	<p>PROT. OBDII</p> 	SCANNER	761 USD

Realizado por: Montesdeoca, Andrea, 2020

Fuente: (Amazon, 2021)

La tabla expuesta anteriormente muestra los tipos de scanner más utilizados y de mayor relevancia en el campo automotriz, a continuación se realizan un análisis comparativo de costos para observar los porcentajes de diferencia en cada equipo, en la siguiente tabla se muestra los porcentajes de costos en los equipos scanner.

Tabla 25-3: Porcentajes de costos en los equipos scanner.

ID	MODELO	TIPO	COSTO	VALOR EN %
1	LAUNCH x431	SCANNER	1,199	36%
2	AUTEL MAXICOM MK906BT	SCANNER	1,269	38%
3	AUTEL AUTOLINK AL301	SCANNER	3,141	95%
4	AUTEL MAXI IM508	SCANNER	1,369	41%
6	X431 V + 4,0	SCANNER	1,199	36%
7	PROT. OBDII	SCANNER	761	23%
5	AUTEL IM608	SCANNER	3,299	100%

Realizado por: Montesdeoca, Andrea, 2020

Se toma como referencia al scanner AUTEL IM608 de mayor costo como el 100%, entonces observamos que los escáneres más costosos son: AUTEL IM608, AUTEL AUTOLINK AL301, AUTEL MAXICOM MK906BT, AUTEL MAXI IM508 los cuales toman un porcentaje entre el 38, 41 y 95%; a diferencia del prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistema OBD II, cuyo valor en porcentaje equivale a un 23%, siendo el más barato con respecto a costos y considerando que fue fabricado en el país.

CONCLUSIONES

- Se implementó un prototipo para registros telemáticos y detección de fallos del motor mediante sistema OBD II para vehículos de marca Toyota Camry modelo 2017 para detectar fallos en el motor, mediante el uso del hardware y software adecuado.
- Mediante el análisis e investigación se conoció el funcionamiento de los sensores automotrices incorporados en el motor de un vehículo, el control electrónico (ECU) que permiten medir las variables de estudio de esta investigación y determinar posibles fallas.
- Para el almacenamiento de los datos obtenidos como respuesta del procesamiento de la ECU al conectarse el sistema OBDII con la tarjeta Arduino, permite que estos se almacenen en la nube (web) para tener un registro del funcionamiento del motor del vehículo cuando la cobertura de la red le permite.
- De las pruebas realizadas del funcionamiento del prototipo al momento de adquirir los datos de los sensores se determinó que el sensor RPM operan en un rango normal de 600 a 1800 RPM (mediante el datalogger), del sensor de temperatura del refrigerante, aceite, admisión y ambiente, muestran un error relativo del 1% y 2% que se debe a la temperatura ambiente y la producida por el funcionamiento del motor.
- Los resultados obtenidos del sensor de distancia y velocidad, tiene un error relativo del 10,55% y 27.13% que se encuentra dentro de los rangos permitidos, tomando en cuenta que estos sensores toman datos de manera variable. El consumo de corriente del prototipo es de 0.187A (amperios) que no afecta a la carga de la batería del vehículo.
- Del análisis de costo en la implementación del prototipo para registros telemáticos y detección de fallos del motor, el sistema OBDII, es el que tiene mayor costo dentro de la implementación, por lo que puede ser adquirido por los talleres automotriz.
- Según un análisis de costo se observa que los scanner profesionales AUTEL IM608, AUTEL AUTOLINK AL301, AUTEL MAXICOM MK906BT, AUTEL MAXI IM508, toman un porcentaje entre el 38, 41 y 95%; a diferencia del prototipo electrónico para registros telemáticos y detección de fallos del motor mediante sistema OBD II, cuyo valor en porcentaje equivale a un 23%, siendo el más barato con respecto a costos y considerando que fue fabricado en el país.

RECOMENDACIONES

- La recolección de datos solo es posible para vehículos que tengan certificación OBDII, es decir, autos que en su interior integran una unidad electrónica de motor (ECU) o más conocida como memoria.
- Para la adquisición de los datos es necesario que el vehículo se encuentre con el motor encendido (ON) para que la ECU pueda proporcionar información de los sensores a analizarse y cuente con el certificado del protocolo de comunicación OBDII para la correcta comunicación con el prototipo.
- Las pruebas del prototipo deben realizarse en un lugar donde exista cobertura de la operadora celular del dispositivo móvil para poder subir la información de las variables de estudio a la base de datos en la web.
- Los datos obtenidos mediante el prototipo deben ser analizados por personal capacitado en ECUS y diagnóstico automotriz avanzado para una mejor toma de decisiones cuando al caso lo amerite para un mejor funcionamiento del vehículo.

GLOSARIO

Aceleración: Variación de la velocidad con el paso del tiempo t . (Bragado, 2003)

Algoritmo: Es una secuencia de instrucciones secuenciales, utilizadas para ejecutar ciertos procesos que permiten dar solución a necesidades. (Raffino, 2020)

CAD: Diseño asistido por ordenador es un sistema de hardware y software, para diseñar y documentar objetos del mundo real. (ArcMap, 2019)

Datalogger: Dispositivo electrónico que registra datos en el tiempo o en relación a la ubicación por medio de sensores que se encuentran conectados interna o externamente. (Carrera, 2018)

Datos: Datos es la representación de una variable que puede ser cuantitativa o cualitativa que indica un valor y se representa por medio de símbolos, números o letras. (Suarez, 2019)

Distancia: Longitud que un objeto se mueve a lo largo de una trayectoria. (Salas, 2012)

ECU: Unidad de control del motor es el componente principal que regula, lee, procesa y comanda todos los componentes electrónicos que vienen integrados en el motor del vehículo. (AutoAvance, 2018)

Error absoluto: Diferencia entre el valor real y el valor aproximado en valor absoluto. (Ekuatio, 2019)

Error relativo: Determina la precisión de la medición, indica la proporción del error con respecto al valor exacto de la medición. La medición se considera buena cuando supera el 5%. (Ekuatio, 2019)

Esquemático: Representación de tipo gráfico o bosquejo de algún proceso. (REA, 2016)

Interfaz: Conjunto de elementos que se visualizan en una pantalla que permite al usuario realizar acciones sobre una aplicación o sitio web. (Beta, 2018)

OBD II: Los sistemas OBD II permite detectar averías en el vehículo, este sistema consta de una toma exterior en algún lugar predeterminado del vehículo generalmente en la zona de fusibles, este sistema se conecta mediante un cable el cual extrae la información necesaria suministrada por la unidad de control electrónica de motor. (Motor, 2017)

Thingspeak: La plataforma Thingspeak por datos de internet es un entorno analítico de I o T que permite visualizar, agregar y analizar transmisiones de datos directo en una nube, es posible extraer los datos un canal designado por Thingspeak para mostrarse en un entorno Matlab. (MathWorks, 2018).

Velocidad: Variación de la posición con el paso del tiempo t . (Bragado, 2003)

BIBLIOGRAFÍA

ALAMBRICA COMUNICACIÓN. *Comunicación alámbrica e inalámbrica.* [blog] 2015. [Consulta: 19 de Noviembre de 2020.] Disponible en: <https://sites.google.com/site/comunicacionalainala/evaluacion>.

AMAZON. *Sim800l GPRS TTL Puerto del Módulo GSM Micro Tarjeta de Base Sim Cuatribanda para Arduino.* [blog] 2018. [Consulta: 09 de Noviembre de 2020.] Disponible en: <https://www.amazon.es/Sim800l-Puerto-Tarjeta-Cuatribanda-Arduino/dp/B01FK3OPQ6>.

AMBROSIO, Roberto; & SANCHÉZ, Luis. *La importancia de la electrónica en el desarrollo mundial.* [En línea] 2017. [Consulta: 25 de 09 de 2020.] Disponible en: <https://saberesciencias.com.mx/2017/06/04/la-importancia-de-la-electronica-en-el-desarrollo-del-automovil/>.

ANDRADE, Silvia. *Arduino Mega 2560.* [blog] 2018. [Consulta: 09 de Noviembre de 2020.] Disponible en: <http://manueldelgadocrespo.blogspot.com/p/arduino-mega-2560.html>.

ARCMAP. *Qué son los datos CAD.* [blog] 2019. [Consulta: 08 de Enero de 2021.] Disponible en: <https://desktop.arcgis.com/es/arcmap/10.3/manage-data/cad/what-is-cad-data.htm>.

ARDUINO. *Arduino Mega 2560.* [En línea] 2017. [Consulta: 08 de Noviembre de 2020.] Disponible en: <http://manueldelgadocrespo.blogspot.com/p/arduino-mega-2560.html>.

AUTOAVANCE. *Sensor de posición del cigüeñal.* [En línea] 2019. [Consulta: 27 de Octubre de 2020.] Disponible en: <https://www.autoavance.co/blog-tecnico-automotriz/135-sensor-de-posicion-del-ciguenal-ckp/>.

AUTOAVANCE. *Ecu Automotriz Funcionamiento.* [blog] 2018. [Consulta: 08 de Enero de 2021.] Disponible en: <https://www.autoavance.co/blog-tecnico-automotriz/ecu-automotriz-funcionamiento/>.

BELTRAN, Manuel. *Memoria CGROM.* [En línea] 2015. [Consulta: 08 de Noviembre de 2020.] Disponible: <https://es.scribd.com/document/232839448/Memoria-CGROM>.

BETA. *¿Qué es una Interfaz?* [En línea] 2018. [Consulta: 08 de Enero de 2021.] Disponible en: <https://www.guiadigital.gob.cl/articulo/que-es-una-interfaz.html>.

BI SOLID. *Solidworks, que es y para que sirve.* [bog] 2017. [Consulta: 5 de Diciembre de 2020.] Disponible en: <https://solid-bi.es/solidworks/>.

BRAGADO, Ignacio. *Física general*. [En línea] Bogota - Colombia, 2003. [Consulta: 08 de Enero de 2021.] Disponible en: <https://fisicas.ucm.es/data/cont/media/www/pag-39686/fisica-general-libro-completo.pdf>.

CANDO, Alex. Diseño e implementación de un simulador para el diagnóstico de la unidad de control electrónico de motor. [En línea] (Trabajo de titulación). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. pp. 27 - 40. 2017. [Consulta: 28 de 9 de 2020.] Disponible en: <http://dspace.espech.edu.ec/bitstream/123456789/8958/1/108T0222.pdf>.

CARRERA, Luis. *¿Que es un Datalogger?* [blog] 2018. [Consulta: 08 de Enero de 2021.] Disponible en: <https://www.finaltest.com.mx/product-p/art-4.htm>.

CISE, Electronica. *El sensor MAF*. [En línea] 2017. [Consulta: 27 de Octubre de 2020.] Disponible en: <http://www.cise.com/portal/notas-tecnicas/item/142-el-sensor-maf.html>.

EKUATIO. *Error absoluto y error relativo: Qué son y cómo se calculan*. [En línea] 2019. [Consulta: 08 de Enero de 2021.] Disponible en: https://ekuat.io/error-absolutos-y-error-relativos-que-son-y-como-se-calculan/#Error_absoluto.

ELECTROPRO. *Módulo GSM SIM800L*. [En línea] 2019. [Consulta: 09 de Noviembre de 2020.] Disponible en: <http://electropro.pe/image/data/imgProductos/140.%20M%C3%B3dulo%20GSM%20SIM800/SIM800L.pdf>.

ENKA. *Módulos Almacenamiento*. [En línea] 2018. [Consulta: 09 de Noviembre de 2020.] Disponible en: <http://www.enka.com.uy/robotica/modulos-almacenamiento/m%C3%B3dulo-para-tarjeta-micro-sd-detail.html>.

FREEMATICS. *Freematics OBD-II UART*. [En línea] 14 de Noviembre de 2018. [Consulta: 21 de Noviembre de 2020.] Disponible en: https://freematics.com/store/index.php?route=product/product&product_id=83.

GLOBALTECH. *Sistema OBD II*. [En línea] 2019. [Consulta: 27 de 09 de 2020.] Disponible en: https://www.globaltechla.com/eBusiness/fichas_tecnicas/TEORIA-CONECTIVIDAD-OBD.pdf.

MATHWORDS. *Matlab*. [En línea] 2018. [Consulta: 25 de Octubre de 2020.] Disponible en: <https://es.mathworks.com/help/matlab/iot-data.html>.

MEDINA, Santiago. *Sensores Automotrices*. [En línea] 2018. [Consulta: 27 de Octubre de 2020.] Disponible en: https://www.researchgate.net/publication/328808726_Sensores_Automotrices.

MICHAEL, Henry. *Raspberry Pi*. [En línea] 2016. [Consulta: 09 de Noviembre de 2020.] Disponible en: https://upload.wikimedia.org/wikipedia/commons/f/f1/Raspberry_Pi_4_Model_B_-_Side.jpg.

MOTOR. *Cómo funciona el sistema OBD de un coche*. [En línea] 2017. [Consulta: 4 de Octubre de 2020.] Disponible en: <https://www.motor.mapfre.es/consejos-practicos/consejos-de-mantenimiento/como-funciona-el-sistema-obd/>.

MOTOR. *Hasta que punto la electronica es la nueva mecanica del motor*. [En línea] 2017. [Consulta: 9 de 26 de 2020.] Disponible en: <https://www.motorpasion.com/n/hasta-que-punto-la-electronica-es-la-nueva-mecanica-del-motor>.

MUNDO CARROS. *Sensores* [En línea] 2017. [Consulta: 27 de Octubre de 2020.] Disponible en: <https://mundocarros.info/sensor-iat/>.

OROZCO, José. *Simulador de sensores y actuadores*. [En línea] 2015. [Consulta: 4 de Noviembre de 2020.] Disponible en: <https://www.electronicayservicio.com/Modules/melbastudioCommerce/Main/Src/Catalog/Items/648.pdf>.

PICO, Auto. *Sensor MAP digital*. [En línea] 2015. [Consulta: 27 de Octubre de 2020.] Disponible en: <https://www.picoa.com/es/library/automotive-guided-tests/sensor-map-digital/>.

QUITO, Mónica. *Diseño y construcción de un Data Logger Universal con comunicación RS-232 y USB hacia una PC*. [En línea] (Trabajo de titulación) Escuela Politécnica Nacional, Quito Ecuador. 2015. pp. 47 - 60 [Consulta: 25 de Octubre de 2020.] Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/9817/1/2005AJIEE-16.pdf>

RAFFINO, María. *Algoritmo en informática*. [En línea] 2020. [Consulta: 08 de Enero de 2021.] Disponible en: <https://concepto.de/algoritmo-en-informatica/>.

REA. *Qué es, concepto o significado*. [En línea] 2016. [Consulta: 08 de Enero de 2021.] Disponible en: <https://definiciona.com/esquematico/>.

REDROBÁN, Laura. *Diseño e implementación de un prototipo para el control y tarifación del servicio de taxi a través de sensores inerciales*. [En línea] (Trabajo de titulación) Escuela Superior Politécnica de Chimborazo. Riobamba, Ecuador. 2018. [Consulta: 09 de Noviembre de 2020.] Disponible en: <http://dSPACE.esPOCH.edu.ec/bitstream/123456789/9207/1/108T0251.pdf>.

SALAS, Jorge. *Física*. [En línea] 2012. [Consulta: 08 de Enero de 2021.] Disponible en: <https://educacion.gob.ec/wp-content/uploads/downloads/2013/03/Fisica.pdf>.

STRAUSS, Gonzalo. *Sensores TPS.* [En línea] 2017. [Consulta: 27 de 9 de 2020.] Disponible en: https://www.academia.edu/25304404/CURSO_DE_SENSORES_TPS.

SUAREZ, Jazmin. *¿Qué es un dato?* [En línea] 2019. [Consulta: 08 de Enero de 2021.] Disponible en: <https://concepto.de/dato/>.

TAPIA, Victor. Diseño e implementación de un sistema de transferencia de datos por telemetría para un vehiculo de competencia con el uso de tecnología inalámbrica bluetooth. [En línea] (Trabajo de titulación) Escuela Superior Politécnica de Chimborazo. Riobamba, Ecuador. 2013. [Consulta: 4 de Octubre de 2020.] Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/2918>.

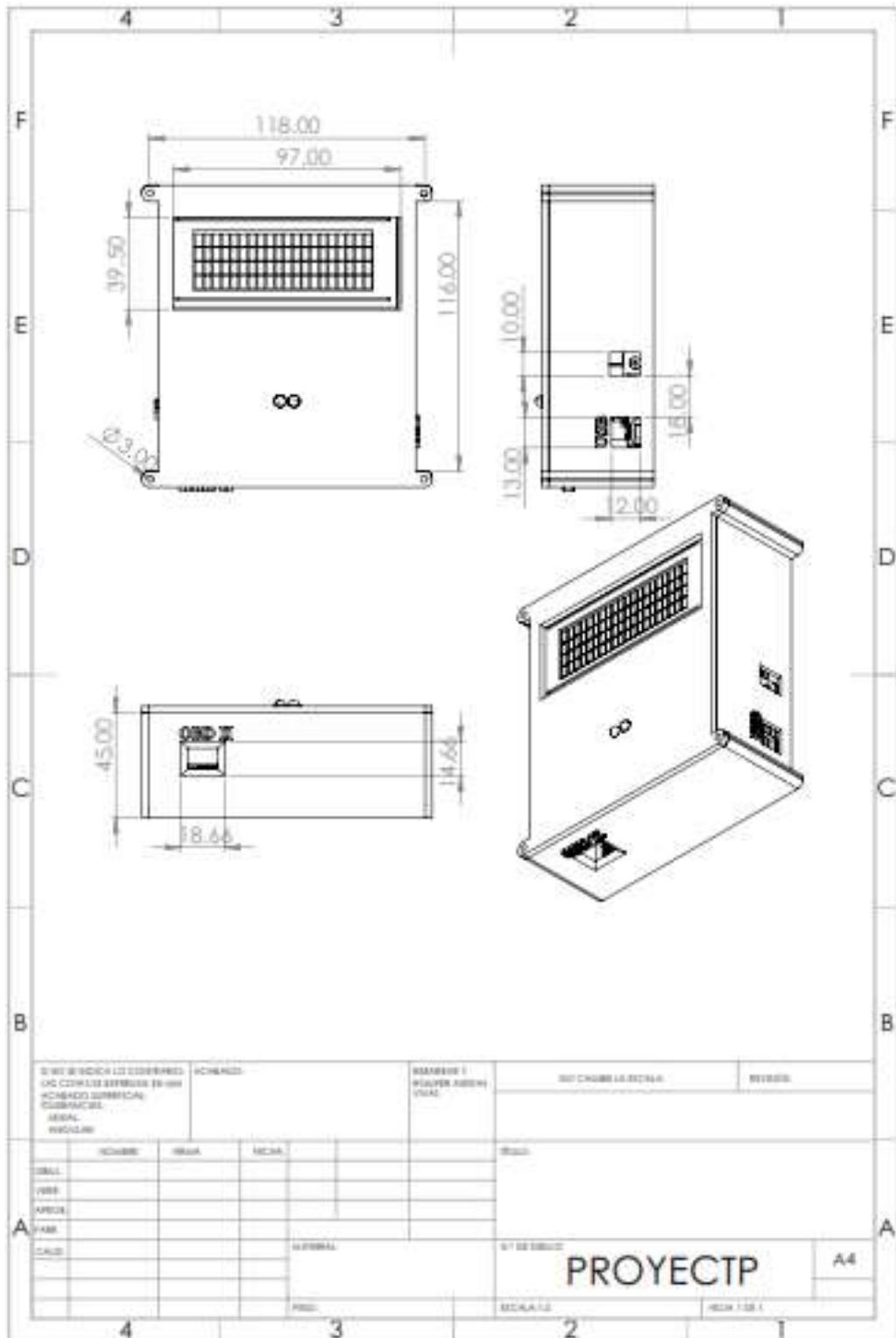
THOMSON. *Sensor posición* [En línea] 2017. [Consulta: 27 de 9 de 2020.] Disponible en: <https://mte-thomson.com/es/productosmte/sensor-posicion-arbol-de-levas-cmp/>.

WERNER, Rolando. *Parametros de lectura en el escaner.* [En línea] 2015. [Consulta: 16 de Diciembre de 2020.] Disponible en: <https://es.scribd.com/doc/165828407/PARAMETROS-DE-LECTURA-EN-EL-ESCANER-doc>.

XUKYO. *Uso de un pantalla 16x2LCD con Arduino.* [En línea] 2020. [Consulta: 06 de Noviembre de 2020.] Disponible en: <https://www.aranacorp.com/es/uso-de-un-pantalla-16x2lcd-con-arduino/>.

ANEXOS

Anexo A: Planos de la caja protectora



Anexo B: Código de programación Arduino para el funcionamiento del prototipo

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>
#include <Arduino.h>
#include <Wire.h>
#include <OBD.h>
#include <LiquidCrystal_I2C.h>
SoftwareSerial gprsSerial(10,11);
#define I2C_ADDR 0x27
const int chipSelect = 4;
LiquidCrystal_I2C lcd(I2C_ADDR,2, 1, 0, 4, 5, 6, 7);
COBD obd;
int RPM,COTE,OILTE,INTE,MAF,MAP,VEL,DIST,THRO,TEAM;
int leda=8,ledv=7,contador=1;
String CADENASD="";
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  Serial.begin(115200);
  gprsSerial.begin(9600);
  digitalWrite(ledv,HIGH);

  lcd.begin (20,4);
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home ();

  lcd.setCursor(4, 0);
  lcd.print(" DATALOGGER");
  lcd.setCursor(8, 1);
  lcd.print("OBDII");
  digitalWrite(ledv,LOW);
  if (!SD.begin(chipSelect)) {
    lcd.clear();
    digitalWrite(leda,HIGH);
    lcd.setCursor(0, 0);
    lcd.print(" NO HAY MEMORIA SD");
    delay(100);
    digitalWrite(leda,LOW);
    delay(100);
    // don't do anything more:
    while (1);
  }

  obd.begin();
}
void loop()
{
  lectura();
  pantalla();
}
```

```

digitalWrite(ledv, HIGH);
delay(1000);
digitalWrite(ledv, LOW);
delay(1000);
    contador=contador+1;
if(contador==10)
{
    GSMDATA1();
    GSMDATA2();
    contador=1;
}
}
void lectura()
{
obd.readPID(PID_RPM,RPM);
obd.readPID(PID_COOLANT_TEMP,COTE);
obd.readPID(PID_ENGINE_OIL_TEMP,OILTE);
obd.readPID(PID_INTAKE_TEMP,INTE);
obd.readPID(PID_MAF_FLOW,MAF);
obd.readPID(PID_BAROMETRIC,MAP);
obd.readPID(PID_SPEED,VEL);
obd.readPID(PID_THROTTLE,THRO);
obd.readPID(PID_DISTANCE,DIST);
obd.readPID(PID_AMBIENT_TEMP,TEAM);
CADENASD="RPM: ";
CADENASD+=RPM;
CADENASD+=" ,T. REFRIE: , ";
CADENASD+=COTE;
CADENASD+=" ,T. ACE: , ";
CADENASD+=OILTE;
CADENASD+=" ,T. AD: , ";
CADENASD+=INTE;
CADENASD+=" ,MAF: , ";
CADENASD+=MAF;
CADENASD+=" ,MAP: , ";
CADENASD+=MAP;
CADENASD+=" ,V: , ";
CADENASD+=VEL;
CADENASD+=" ,DIS: , ";
CADENASD+=DIST;
CADENASD+=" ,POS.A: , ";
CADENASD+=THRO;
CADENASD+=" ,TA: , ";
CADENASD+=TEAM;
digitalWrite(leda,HIGH);
sd();
}
void pantalla()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("RPM:");
    lcd.print(RPM);

    // MAF

```

```
lcd.setCursor(11, 0);
lcd.print("MAF:");
lcd.print(MAF);
lcd.print("g");

// MAP
lcd.setCursor(0, 1);
lcd.print("MAP:");
lcd.print(MAP);
lcd.print("kPa");

    lcd.setCursor(11, 1);
    lcd.print("V:");
    lcd.print(VEL);
    lcd.print("km/h");

//Temperatura de Refrigerante
lcd.setCursor(0, 2);
lcd.print("Temp. Refrig: ");
lcd.print(COTE);
lcd.print((char)223);
lcd.print("C");

//Temperatura Aceite
lcd.setCursor(0, 4);
lcd.print("Temp. Aceite: ");
lcd.print(OILTE);
lcd.print((char)223);
lcd.print("C");

delay(2000);

//Temperatura Admision
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Temp. Admis: ");
lcd.print(INTE);
lcd.print((char)223);
lcd.print("C");

    lcd.setCursor(0, 1);
    lcd.print("Temp. Ambiente: ");
    lcd.print(TEAM);
    lcd.print((char)223);
    lcd.print("C");

    lcd.setCursor(0, 2);
    lcd.print("Pos. Acel: ");
    lcd.print(THRO);
    lcd.print("%");

    lcd.setCursor(0, 3);
    lcd.print("Distancia:");
    lcd.print(DIST);
    lcd.print("km");
```

```

    delay(500);

}

void sd()
{
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
  digitalWrite(leda, HIGH);

  if (dataFile) {
    dataFile.println(CADENASD);
    dataFile.close();
    // print to the serial port too:
    //Serial.println(CADENASD);
    digitalWrite(leda, HIGH);
    delay(100);
    digitalWrite(leda, LOW);
    delay(100);
    digitalWrite(leda, HIGH);
    delay(100);
    digitalWrite(leda, LOW);
    delay(100);
  }
  // if the file isn't open, pop up an error:
  else {
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("ERROR ABRIENDO");
    lcd.setCursor(2, 2);
    lcd.print(" DATALOGGER.TXT");
    delay(1000);
  }
}

void GSMDATA1()
{
  gprsSerial.println("AT");
  delay(1000);
  gprsSerial.println("AT+CPIN?");
  delay(1000);

  gprsSerial.println("AT+CREG?");
  delay(1000);

  gprsSerial.println("AT+CGATT?");
  delay(1000);

  gprsSerial.println("AT+CIPSHUT");
  delay(1000);

  gprsSerial.println("AT+CIPSTATUS");
}

```

```

delay(2000);

gprsSerial.println("AT+CIPMUX=0");
delay(2000);

gprsSerial.println("AT+CSTT=\"internet.tuenti.ec\"); //start task and setting the APN,
delay(1000);

gprsSerial.println("AT+CIICR"); //bring up wireless connection
delay(3000);

gprsSerial.println("AT+CIFSR"); //get local IP adress
delay(2000);

gprsSerial.println("AT+CIPSPRT=0");
delay(3000);

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //start up the
connection
delay(4000);
gprsSerial.println("AT+CIPSEND"); //begin send data to remote server
delay(2000);
String str="GET https://api.thingspeak.com/update?api_key=CFT7OQ3H57FA4JJK&field1="
+ String(RPM)
+"&field2="+String(COTE)+"&field3="+String(OILTE)+"&field4="+String(INTE)+"&field5="
+"&field6="+String(MAF)+"&field7="+String(MAP)+"&field8="+String(DIST
);
Serial.println(str);
gprsSerial.println(str); //begin send data to remote server
delay(2000);
gprsSerial.println((char)26); //sending
delay(2000); //waitting for reply, important! the time is base on the condition of internet
gprsSerial.println();
gprsSerial.println("AT+CIPSHUT"); //close the connection
delay(100);
}

void GSMDATA2()
{
gprsSerial.println("AT");
delay(1000);
gprsSerial.println("AT+CPIN?");
delay(1000);

gprsSerial.println("AT+CREG?");
delay(1000);

gprsSerial.println("AT+CGATT?");
delay(1000);

gprsSerial.println("AT+CIPSHUT");
delay(1000);

gprsSerial.println("AT+CIPSTATUS");
delay(2000);

```

```

gprsSerial.println("AT+CIPMUX=0");
delay(2000);

gprsSerial.println("AT+CSTT=\"internet.tuenti.ec\");//start task and setting the APN,
delay(1000);

gprsSerial.println("AT+CIICR");//bring up wireless connection
delay(3000);

gprsSerial.println("AT+CIFSR");//get local IP adress
delay(2000);

gprsSerial.println("AT+CIPSPRT=0");
delay(3000);

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\");//start up the
connection
delay(4000);
gprsSerial.println("AT+CIPSEND");//begin send data to remote server
delay(2000);
String str="GET https://api.thingspeak.com/update?api_key=I1LX1H43SH8Y1IW7&field1="
+ String(THRO) + "&field2="+String(Team);
Serial.println(str);
gprsSerial.println(str);//begin send data to remote server
delay(2000);
gprsSerial.println((char)26);//sending
delay(2000);//waitting for reply, important! the time is base on the condition of internet
gprsSerial.println();
gprsSerial.println("AT+CIPSHUT");//close the connection
delay(100);
}

```



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO



DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN

UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 01 / 02 / 2021

INFORMACIÓN DE LA AUTORA	
Nombres – Apellidos: ANDREA GABRIELA MONTESDEOCA VIVANCO	
INFORMACIÓN INSTITUCIONAL	
Facultad: INFORMÁTICA Y ELECTRÓNICA	
Carrera: INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES	
Título a optar: INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES	
f. Analista de Biblioteca responsable:	 Firmado electrónicamente por: ELIZABETH FERNANDA AREVALO MEDINA

