



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA CON EL  
MÓDULO DE COMUNICACIÓN RF PARA MONITOREAR EL PROTOTIPO  
SHELL-ECO MARATHON DE LA UPS**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero Electrónico

AUTORES: Brandon Fabián Guato Pozo

Mario Wladimir Yánez Vela

TUTOR: Carlos Alberto Carranco Quiñonez

Quito - Ecuador

2022

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DE TRABAJO DE TITULACIÓN**

Nosotros, Brandon Fabián Guato Pozo con documento de identificación N°1720757259 y Mario Wladimir Yáñez Vela con documento de identificación N°1724328040; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que, sin fines de lucro, la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 25 de julio del año 2022

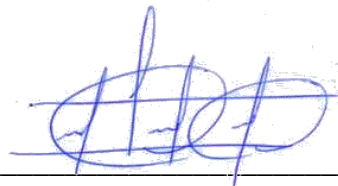
Atentamente,



---

Brandon Fabián Guato Pozo

1720757259



---

Mario Wladimir Yáñez Vela

1724328040

## **CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN**

Nosotros, Brandon Fabián Guato Pozo con documento de identificación N°1720757259 y Mario Wladimir Yánez Vela con documento de identificación N°1724328040, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico :”Diseño e implementación de un sistema de telemetría con el módulo de comunicación RF para monitorear el prototipo Shell-Eco Marathon de la ups”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

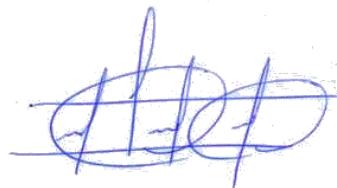
Quito, 25 de julio del año 2022

Atentamente,



Brandon Fabián Guato Pozo

1720757259



Mario Wladimir Yánez Vela

1724328040

## **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.**

Yo, Carlos Alberto Carranco Quiñonez con documento de identificación N.º 1713629564, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA CON EL MÓDULO DE COMUNICACIÓN RF PARA MONITOREAR EL PROTOTIPO SHELL-ECO MARATHON DE LA UPS, realizado por Brandon Fabián Guato Pozo con documento de identificación N.º 1720757259 y Mario Wladimir Yáñez Vela con documento de identificación N.º 1724328040, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 25 de julio del año 2022

Atentamente,



---

Ing. Carlos Alberto Carranco Quiñonez MSc.

1713629564

## **DEDICATORIA Y AGRADECIMIENTO**

Este trabajo de titulación lo dedico a mis padres Rosa Vela y Mario Yáñez ya que su amor, paciencia, apoyo y sacrificio han sido fundamentales para poder culminar mis estudios y así alcanzar una meta más en mi vida, gracias por confiar en mis capacidades y virtudes las cuales me han permitido ser una gran persona.

Agradezco a Dios por permitirme culminar mis metas propuestas a lo largo de mi vida por brindarme fortaleza, salud y capacidad.

A mi madre quien con su ayuda y dedicación en todo momento me ha impulsado a seguir adelante para no rendirme y ser un profesional.

A mi padre quien en vida fue un gran hombre, con su ejemplo y grandes valores inculcados en mi permitieron trazar mi camino y culminar esta meta.

***Mario Wladimir Yáñez Vela***

El presente trabajo de titulación va dedico primeramente a Dios quien me ha guiado en todos los caminos de mi vida a mis padres Fabian Guato y Margoth Pozo que han sido el pilar fundamental para poder culminar mis estudios con su sacrificio, paciencia y amor nunca me dejaron decaer incluso en los momentos más arduos, gracias por apoyarme y así poder cumplir una meta más en mi vida, nunca dudaron de mis capacidades incluso cuando yo llegue a dudar de mí en serio muchas gracias.

Agradezco a Dios por ser el guía en todos los proyectos a lo largo de mi vida y así poder culminar una meta más en mi vida.

A mi padre quien fue el sustento de mis estudios, le agradezco sus buenos consejos los cuales me han formado como una gran persona.

A mi madre quien ha sido mi apoyo incondicional en toda mi vida, sus palabras de aliento fueron el motor para continuar con mis estudios y así no rendirme cuando estuve a punto de hacerlo.

***Brandon Fabián Guato Pozo***

## ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DE TRABAJO DE TITULACIÓN .....	II
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN .....	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN. ....	IV
DEDICATORIA Y AGRADECIMIENTO .....	V
ÍNDICE GENERAL .....	VI
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS .....	XI
RESUMEN .....	XII
ABSTRACT .....	XIII
INTRODUCCIÓN .....	1
1 CAPÍTULO I .....	2
ANTECEDENTES .....	2
1.1 Planteamiento del problema .....	2
1.1.1 Delimitación del problema .....	3
1.2 Justificación .....	3
1.3 Objetivos .....	4
1.3.1 Objetivo general .....	4
1.3.2 Objetivos específicos .....	4
1.4 Metodología .....	4
2 CAPÍTULO II .....	5
FUNDAMENTACIÓN TEÓRICA .....	5
2.1 Telemetría .....	5
2.1.1 Historia de la telemetría .....	5
2.1.2 Aplicaciones de la telemetría .....	6
2.1.3 Telemetría en automóviles .....	7
2.2 Tipos de Transmisión .....	9
2.3 Tecnología LoRa .....	10
2.3.1 Tipos de datos permitidos en LoRa .....	10
2.3.2 Alcance .....	11
2.3.3 Arquitectura LoRa .....	11
3 CAPÍTULO III .....	12

DISEÑO E IMPLEMENTACIÓN .....	12
3.1 Módulos .....	12
3.2 Elementos electrónicos .....	14
3.3 Selección de Sensores .....	16
3.3.1 Definición de sensor.....	16
3.3.2 Sensor TPS (Throttle Position Sensor).....	16
3.3.3 Sensor ECT (Engine Coolant Temperature) .....	17
3.3.4 Sensor MAP (Manifold Absolute Pressure).....	17
3.3.5 Sensor CKP (Crankshaft Position Sensor) .....	18
3.3.6 Sensor WSS (Wheel Speed Sensor).....	19
3.3.7 Sensor IAT (Intake Air Temperature).....	20
3.4 Fase de desarrollo .....	20
3.5 Librerías .....	20
3.6 Fase Preliminar .....	21
3.7 Desarrollo De La Programación .....	24
3.7.1 Sensores.....	24
3.7.2 Elementos electrónicos.....	29
3.8 Diseño Del Emisor.....	31
3.9 Diseño Del Receptor.....	32
3.10 Diseño de Placas en PCB (Placa de Circuito Impreso) .....	33
3.10.1 Diseño de placa PCB Emisor .....	33
3.10.2 Diseño de placa PCB Receptor .....	35
3.11 Diseño e Implementación Interfaz Gráfica en LabView .....	37
3.11.1 Desarrollo de la programación en LabVIEW .....	37
4 CAPÍTULO IV.....	39
ANÁLISIS DE RESULTADOS .....	39
4.1 Pruebas de Comunicación.....	39
4.2 Pruebas de distancia de recorrido .....	43
4.3 Pruebas De Funcionamiento en el prototipo Shell.....	50
4.3.1 Energización de los módulos.....	50
4.3.2 Funcionamiento de los diferentes sensores .....	52
5 CAPÍTULO V .....	53
GESTIÓN DEL PROYECTO .....	53
5.1 Recursos Y Materiales .....	53
5.1.1 Costo De Producción.....	53

5.1.2	Recursos Materiales .....	53
6	CAPÍTULO VI.....	54
	CONCLUSIONES Y RECOMENDACIONES .....	54
6.1.1	CONCLUSIONES .....	54
6.1.2	RECOMENDACIONES .....	55
	BIBLIOGRAFÍA .....	56
	ANEXOS .....	58



## ÍNDICE DE FIGURAS

Figura 2.1. Sistema automotriz de telemetría.....	5
Figura 2.2. Aplicaciones de la telemetría.....	6
Figura 2.3. Telemetría en la Fórmula 1.....	7
Figura 2.4. Sistema de Telemetría ATLAS.....	9
Figura 2.5. Tipos de Transmisión.....	9
Figura 2.6. Aplicaciones de LoRaWan.....	10
Figura 2.7. Arquitectura LoRa.....	11
Figura 3.1. Modulo Arduino Mega2560.....	12
Figura 3.2. Modulo Arduino Uno.....	13
Figura 3.3. LoRa Ra-01 SX1278.....	13
Figura 3.4. Regulador DC-DC LM2596.....	14
Figura 3.5. Sensor LM35.....	14
Figura 3.6. Pantalla LCD 16x2.....	14
Figura 3.7. Divisor de tensión.....	15
Figura 3.8. Pantalla LCD 20x4.....	15
Figura 3.9. Módulo adaptador serial I2C para pantalla LCD.....	16
Figura 3.10. Sensor TPS.....	16
Figura 3.11. Señal del sensor TPS.....	17
Figura 3.12. Sensor ECT.....	17
Figura 3.13. Sensor MAP (Haltech).....	18
Figura 3.14. Tipos de sensores CKP.....	18
Figura 3.15. Señal del sensor CKP (Hall).....	19
Figura 3.16. Sensor WSS.....	19
Figura 3.17. Señal del sensor WSS efecto Hall.....	19
Figura 3.18. Señal del sensor WSS efecto Hall.....	20
Figura 3.19. Diseño preliminar del receptor.....	22
Figura 3.20. Diseño preliminar del Emisor.....	23
Figura 3.21. Divisor de voltaje.....	30
Figura 3.22. Proceso del diseño para el emisor.....	31
Figura 3.23. Proceso del diseño para el receptor.....	32
Figura 3.24. Esquemmatización del diagrama del Emisor.....	33
Figura 3.25. Diseño de la placa PCB del emisor.....	34
Figura 3.26. Montaje de la placa PCB.....	34
Figura 3.27. Esquemmatización del diagrama del Receptor.....	35
Figura 3.28. Diseño de la placa PCB.....	36
Figura 3.29. Montaje de la placa PCB.....	36
Figura 3.30. Interfaz gráfica LabVIEW.....	37
Figura 3.31. Programación en LabVIEW.....	38
Figura 4.1. Ejemplo LoRaSender.....	39
Figura 4.2. Código LoRaSender.....	40
Figura 4.3. Monitor Serial LoRaSender.....	41
Figura 4.4. Ejemplo LoRaReceiver.....	41
Figura 4.5. Código LoRaReceiver.....	42
Figura 4.6. Monitor Serial LoRaSender.....	43

Figura 4.7. Distancia de 200.61 metros Parque ecológico Santa Ana.....	44
Figura 4.8. RSSI a 200 metros.....	45
Figura 4.9. Distancia de 400.55 metros Parque ecológico Santa Ana.....	45
Figura 4.10. RSSI a 400 metros.....	46
Figura 4.11. Distancia de 600.24 metros Parque ecológico Santa Ana.....	46
Figura 4.12. RSSI a 600 metros.....	47
Figura 4.13. Distancia de 800.28 metros Parque ecológico Santa Ana.....	47
Figura 4.14. RSSI a 800 metros.....	48
Figura 4.15. Distancia de 1000 metros Parque ecológico Santa Ana.....	48
Figura 4.16. RSSI a 1000 metros.....	49
Figura 4.17. Comparación de datos RSSI. ....	49
Figura 4.18. Módulo Emisor energizado .....	50
Figura 4.19. Módulo Receptor energizado .....	51
Figura 4.20. Sistema de telemetría funcionando. ....	51

## ÍNDICE DE TABLAS

Tabla 3.1 Librerías.....	21
Tabla 3.2 Conexiones de pines del módulo LoRa a los pines del Arduino Uno. ....	22
Tabla 3.3 Conexiones de pines del módulo LoRa a los pines del Arduino Mega. ....	23
Tabla 3.4. Cálculo de la resistencia. ....	30
Tabla 4.1. Parámetros LoRa. ....	43
Tabla 4.2. Resultados obtenidos .....	52
Tabla 5.1. Costo de producción. ....	53
Tabla 5.2. Recursos Materiales.....	53

## RESUMEN

El presente trabajo tecnológico muestra el proceso de implementación y diseño de un sistema de telemetría, para la transmisión y control de datos en tiempo real del prototipo Shell Eco Marathon de la Universidad Politécnica Salesiana y de esta manera verificar los sensores de funcionamiento, como las RPM del motor, la temperatura del motor y cabina, la velocidad del prototipo, la presión del aire de admisión, la temperatura del aire de admisión, el nivel de la batería, la posición de la aleta de aceleración, para tener un mejor seguimiento de la conducción del piloto y poder asistirlo de manera eficiente, el sistema está basado en la lectura, procesamiento y visualización de la información en tiempo real, obtenidos del prototipo en funcionamiento. La implementación tecnológica está basada en la utilización de Arduino, el cual es un hardware libre que será el que recopila todas las señales de los sensores. Para la transmisión, recepción de datos se lo realizo mediante el uso de módulos LoRa SX12-78, el cual permite la comunicación inalámbrica entre uno o varios dispositivos. La interfaz gráfica se la desarrollo en LabVIEW, la cual nos permite visualizar mediante gráficas el funcionamiento de cada una de las señales de los diferentes sensores instalados en el prototipo en tiempo real.

**PALABRAS CLAVE:** Telemetría, Prototipo, Comunicación Inalámbrica, Interfaz gráfica, LabVIEW (Software), Arduino (Hardware).

## **ABSTRACT**

The present technological work shows the procedure for the implementation of a telemetry system, for the monitoring and transmission of data in real time of the Eco Shell Marathon prototype of the Salesian Polytechnic University and to be able to verify the operating parameters such as the engine RPM, the engine temperature, prototype speed, intake air pressure, intake air temperature, battery level, throttle position, to have a better follow-up of the pilot's driving and power of In the same way, assist you efficiently, the system is based on the acquisition, transmission and visualization of data in real time obtained from the prototype in operation. The electronic system is based on the use of Arduino, which is free hardware that will collect all the signals from the sensors. For the data transmission part, it was done through the use of LoRa SX12-78 modules, which is a wireless communication module that allows interconnection between one or more devices. The graphical interface was developed in LabVIEW, which allows us to graphically visualize the operation of each of the signals of the sensors installed in the prototype in real time.

**KEY WORDS:** Telemetry, Prototype, Wireless Communication, Graphical Interface, LabVIEW (Software), Arduino (Hardware).

## INTRODUCCIÓN

En el capítulo 1 se presenta una descripción de la competencia Shell Eco Marathon y el aporte tecnológico que se desarrollara como proyecto técnico, ya que la Universidad Politécnica Salesiana a través de la carrera de Ingeniería Automotriz, son participantes en la competencia Shell Eco Marathon, se ha tenido varias participaciones en esta competencia, viendo así la necesidad de adaptar varias mejoras para futuras participaciones mediante una previa investigación se plantea desarrollar el diseño e implementación de un sistema de telemetría. Con este proyecto se busca mejorar la eficiencia del prototipo a gasolina.

Toda la información recolectada se analizará en el capítulo 2 basados en estudios preliminares como datos, parámetros y criterios de varios investigadores esto permitirá adaptar el desarrollo del proyecto técnico de una manera más eficiente para el diseño final del sistema de telemetría aplicada al prototipo Shell Eco Marathon de la Universidad Politécnica Salesiana.

La selección de módulos de comunicación, diseño a nivel de hardware y software, los pasos y requerimientos para un correcto funcionamiento se presentan en el capítulo 3.

El capítulo 4, expone el análisis de resultados obtenidos como pruebas de comunicación entre módulos y alcancé, además se presentan las respectivas pruebas de funcionamiento.

Se dará a conocer la factibilidad técnico-económico del proyecto, detallando el costo de producción, talento humano, recursos utilizados, se presentan en el capítulo 5.

Por último, el capítulo 6 se plantean las diferentes conclusiones y recomendación que se tuvieron, a medida que se fue desarrollando el proyecto tecnológico.

# CAPÍTULO I

## ANTECEDENTES

### 1.1 Planteamiento del problema

La competencia Shell Eco Marathon se da de manera anual patrocinada por la empresa Shell, los participantes construyen prototipos con el objetivo de generar la mayor eficiencia energética posible, existen categorías como: motores a gasolina, eléctricos, en base a etanol, entre otros.

En 2018, alrededor de 5.000 estudiantes de más de 700 universidades provenientes de 52 países participaron en esta competencia. El alcance digital de la Shell Eco Marathon es de aproximadamente varios millones de espectadores a nivel mundial.

Shell Eco Marathon se lleva a cabo cada año en diferentes partes del mundo, en el caso de América Latina la cita es en Brasil, en la ciudad de Río de Janeiro. Esta iniciativa fue establecida en 1939, en sus inicios, esta competencia se desarrolló entre sus empleados con el fin de determinar quién de ellos lograría recorrer una mayor distancia con la misma cantidad de combustible. (SHELL, 2022)

El prototipo desarrollado por la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana; después de participar en la competencia Shell, se evidenció problemas relacionados con la falta de comunicación y monitoreo de sensores tales como: CKP (RPM), ECT (°C), IAT(°C), MAP (kPa), TPS (%), WSS (Km/h), Termistor (°C), Batería(V), los cuales son parte del prototipo de la categoría: Motores a Gasolina.

El problema principal es que el operador del centro de monitoreo no podía visualizar los diferentes datos de los sensores antes mencionados, en consecuencia, las diferentes decisiones sobre las acciones de conducción permiten al piloto tener una mayor eficiencia en el ahorro de combustible del prototipo Shell Eco Marathon.

### **1.1.1 Delimitación del problema**

- Objeto de investigación: Telemetría
- Campo de acción: Electrónica, Automotriz
- Límite espacial: La investigación fue realizada en la Carrera de Ingeniería Automotriz, de la UPS, en la provincia de Pichincha.
- Delimitación temporal: La investigación se llevo a cabo en el año 2022.

### **1.2 Justificación**

Para las futuras participaciones en la competencia Shell Eco Marathon, y después de haber determinado que el principal problema es la falta de comunicación y control de los sensores. Como estudiantes de la carrera de Ingeniería Electrónica hemos visto la necesidad de contribuir con este prototipo, al diseñar e implementar un sistema de telemetría aplicable en los prototipos de categoría a gasolina, participantes en la competencia.

Por tal motivo se desarrollará e implementará un dispositivo con tecnología capaz de analizar el desempeño del prototipo, acudiendo en la toma de decisiones y corrección de errores de esta manera se crean formulación de estrategias.

Con este proyecto se busca mejorar la eficiencia del prototipo a gasolina para la competencia, por medio de un aporte tecnológico desarrollado en Arduino a través de módulos LoRa; mismos que funcionan mediante Radiofrecuencia; para determinar una comunicación entre un emisor y un receptor, logrando que durante la competencia el piloto y el centro de monitoreo puedan intercambiar información y datos relevantes en tiempo real.



## 1.3 Objetivos

### 1.3.1 Objetivo general

Diseñar un sistema de telemetría con el módulo de comunicación RF que monitoree el prototipo Shell Eco Marathon de la Universidad Politécnica Salesiana.

### 1.3.2 Objetivos específicos

- Configurar y adaptar los módulos de comunicación LoRa simplex, semi - dúplex, dúplex, según los parámetros deseados en el prototipo dentro de la configuración electrónica para la adquisición de los datos.
- Crear la programación en Arduino que nos permita el envío y la obtención de datos por medio de módulos de radiofrecuencia LoRa.
- Desarrollar un HMI (human-machine interface) en LabVIEW que nos permita visualizar los diferentes parámetros en tiempo real.
- Monitorear los sensores del motor en tiempo real para formular estrategias de conducción.
- Realizar las pruebas de resultados, mediante el uso de los dispositivos como el módulo LoRa Ra-01 SX1278 en base a sus especificaciones técnicas que forman parte del registro de datos, con el piloto y el operador del centro de monitoreo en pista.

## 1.4 Metodología

**Metodología analítica:** Gracias a esta metodología se podrá examinar las variables físicas de prototipo Shell Eco Marathon de la UPS. Para encontrar los transductores apropiados para la implementación tecnológica.

**Metodología deductiva:** Con la información obtenida se diseñará e implementará los dispositivos de comunicación por radiofrecuencia.

**Metodología experimental:** Con las pruebas realizadas en campo comprobamos el funcionamiento de los dispositivos, la intensidad de la señal y la distancia de alcance.

## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

#### 2.1 Telemetría

Inicio con la necesidad de tomar magnitudes en lugares de difícil acceso, como la temperatura dentro de un motor, y se ha convertido en una tecnología compleja que se puede realizar dentro de un misil autoguiado o en cualquier ubicación remota. (Morocho Arroyo, 2020)

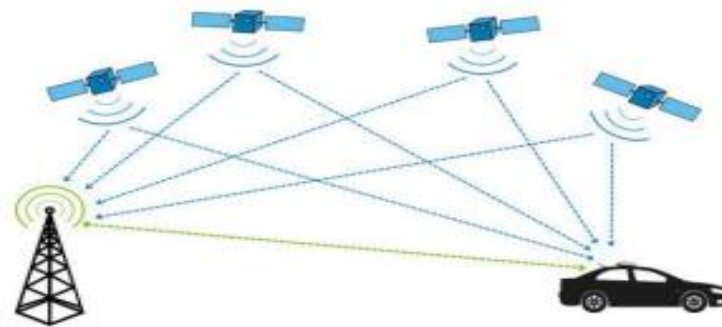
Esta técnica permite realizar mediciones de cantidades físicas en lugares de difícil acceso y puede transmitir los resultados de las mediciones a un observador remoto.

En la primera guerra mundial se utilizó por primera vez esa tecnología para poder medir la distancia a la que se encontraban objetivos de artillería. (Tapia, 2013)

Múltiples sistemas utilizan varios transductores, este método recopila datos en un solo enlace de radiofrecuencia, para procesar datos muy complejos antes de mostrarlos. (Morocho Arroyo, 2020)

En la Figura 2.1. podemos observar como la telemetría es aplicada en el campo automotriz ya que posee los medios adecuados como antenas, emisores, receptores, para el correcto funcionamiento tecnológico.

Figura 2.1. Sistema automotriz de telemetría.



Fuente: (Tapia, 2013)

##### 2.1.1 Historia de la telemetría

La telemetría se puede describir como un proceso de comunicación que implica recoger datos en puntos de difícil acceso, para enviarlos a recepción de equipos con fines de seguimiento y control.

Esta tecnología se remonta al siglo XIX. En 1845 se desarrolló los primeros circuitos de transmisión de datos. En 1874, ingenieros franceses diseñaron sensores meteorológicos y de profundidad para enviar información en tiempo real a París (Aranda, 2014)

En 1901, el inventor estadounidense C. Michelle creó SELSYN, dispositivo que permitía el envío de información a distancia, gracias a esta tecnología se mejoró los tiempos de recepción de la información.

En 1906 se construyeron varias estaciones sísmicas de telemetría en el observatorio Pulkovo en Rusia. En 1912 para monitorear las cargas eléctricas. En 1914, se utilizaron sistemas de telemetría para monitorear los niveles de agua de esta manera que completó el Canal de Panamá.

Los primeros años de la década de 1930 vieron el desarrollo simultáneo de receptores de radio, con la primera aparición de la radiosonda. (Díaz, 2017)

### 2.1.2 Aplicaciones de la telemetría

Las aplicaciones son varias ya que múltiples dispositivos requieren un monitoreo continuo, entre las principales aplicaciones que se presentan en la actualidad se tiene:

- Industria espacial
- Industria automotriz
- Industrias químicas
- Puntos de suministros

El monitoreo continuo permite al operador tomar decisiones eficientes para realizar acciones oportunas, de los sistemas a distancia.

En la siguiente Figura 2.2. se muestra claramente donde es aplicable la telemetría.

Figura 2.2. Aplicaciones de la telemetría



Fuente: (SITRACK, 1999)

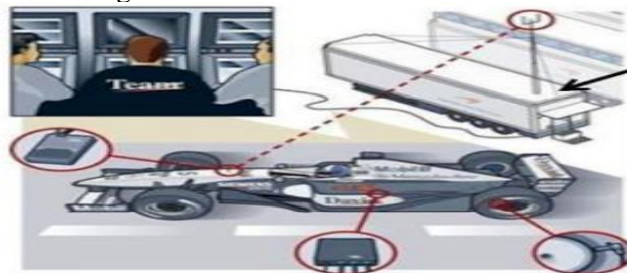
Por otro lado, la telemetría con mayor precisión es usada en carreras como la Fórmula 1, MotoGP. En este campo, es importante porque el dispositivo consta de múltiples sensores que recopilan datos en tiempo real, lo que permite al centro de monitoreo trabajar con esta información de manera eficiente. (Rhetassi, 2019)

### 2.1.3 Telemetría en automóviles

La Fórmula Uno y la telemetría llevan de la mano más de 30 años con el paso del tiempo se ha convertido en una herramienta vital para recolectar y estudiar datos. Un vehículo de Fórmula Uno se mantiene bajo estudio constante gracias a la telemetría. (Carpentiers, 2016)

Esta tecnología se introdujo a fines de la década 1980, y ha evolucionado a lo largo de los años, la FIA (Federación Internacional del Automóvil) a principios del año 2000 permitió la telemetría bidireccional. “En la Fórmula Uno se usa la telemetría de una manera tan eficiente dijo el jefe de electrónica de Ferrari, Andrea Beneventi. Por ejemplo, en GP2 existe un 35 a 40 % menos de parámetros monitoreados. Existen competencias que se llevan a cabo las 24 horas, Le Mans tienen un sistema de telemetría muy sofisticado”. La telemetría es un elemento clave de la F1 ya que permite al conductor y a los ingenieros recopilar grandes cantidades de datos mientras compiten. Los datos pueden ser interpretados y usados para obtener máximo rendimiento.

Figura 2.3. Telemetría en la Fórmula 1



Fuente: (Díaz, 2017)

Principales aplicaciones de la telemetría en la Fórmula Uno:

- Monitoreo de la paleta de aceleración
- Lecturas de sensores
- Velocidad de las ruedas
- Suspensión
- Presión
- Posicionamiento en pista

Las acciones del conductor también son monitoreadas y llevan un registro el cual permite al equipo evaluar el rendimiento general.

- Acción del pedal de freno
- Acción del pedal del acelerador
- Ángulo de dirección
- Posición de marcha.

TAG Electronics desarrollo la telemetría bidireccional. Esta tecnología busco la forma de enviar un mensaje al conductor, permitiendo actualizar los parámetros del automóvil en tiempo real, por ejemplo, cambiando el mapeo del motor.

En el 2003 se prohibió la telemetría bidireccional en la Fórmula Uno, por el ente regulador (FIA). Los participantes de F1 llevan una gran cantidad de equipos informáticos con el objetivo de brindar la ayuda necesaria al piloto y a los ingenieros, para dar solución a cualquier inconveniente que se presente con el vehículo. (Denton, 2011)

Existen dos tipos de telemetría en la F1:

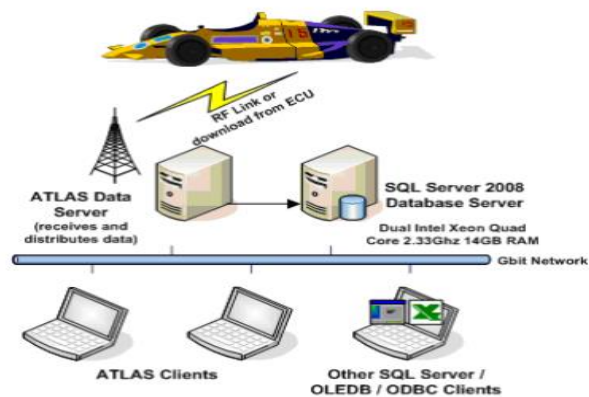
- Envío de información en pequeños paquetes en tiempo real.
- A través de un estallido de microondas cuando el vehículo pasa por los pits.

Los bancos de computadoras se utilizan para procesar la información enviada cuando el automóvil está en la carretera. Estas computadoras usan software de compañías como Hewlett Packard, Compaq y TAG Electronics.

El software presenta la información en una interfaz gráfica de esta manera el centro de monitoreo pueda interpretar datos, revisa cuidadosamente indicadores como la temperatura del motor y la presión del aceite para asegurarte de que no haya fallas importantes. Si la lectura es más alta o más baja de lo normal, el centro de monitoreo puede comunicarse de forma inalámbrica con el conductor y pedirle que reduzca la velocidad del motor o frene más rápido para evitar fallas.

McLaren ha desarrollado un sistema único llamado Advanced Telemetry Linked Acquisition System (ATLAS). El sistema proporciona una representación gráfica en tiempo real del automóvil en todos los tramos de la pista. Existen varios equipos que poseen un sistema similar, pero McLaren actualmente se considera uno de los mejores equipos.

Figura 2.4. Sistema de Telemetría ATLAS



Fuente: (Carpentiers, 2016)

## 2.2 Tipos de Transmisión

- **Simplex**

Los canales de comunicación simplex envían información en una sola dirección. Por ejemplo, una estación de radio generalmente envía una señal al público, pero nunca recibe una señal del público, por lo que una radio posee una conexión simplex.

- **Semidúplex**

Un canal semidúplex, la información se puede transmitirse en ambas direcciones, excepto que no al mismo tiempo. El walkie-talkie posee una comunicación semidúplex.

- **Dúplex**

Los canales de comunicación dúplex pueden transmitir datos en ambas direcciones al mismo tiempo. Está construido como un par de enlaces simplex que permite la transmisión dual, el teléfono posee una comunicación dúplex.

Figura 2.5. Tipos de Transmisión



Fuente: (Orenda, 2016)

## 2.3 Tecnología LoRa

LoRa se deriva de las siglas Long Range, que es una tecnología inalámbrica similar (WiFi, Bluetooth, LTE, SigFox, Zigbee), con la gran diferencia que posee un bajo consumo de energía y largo alcance de funcionamiento de kilómetros.

LoRa es la capa física del protocolo LoRaWAN, que permite comunicaciones inalámbricas de baja potencia y larga distancia ideales para proyectos de Internet de las cosas. (Bartoletti, Paiotti, & Leca, 2019)

Figura 2.6. Aplicaciones de LoRaWAN



Fuente: [www.catsensors.com](http://www.catsensors.com)

### 2.3.1 Tipos de datos permitidos en LoRa

LoRa se la puede usar en forma de red, formando así lo que se conoce como una (LPWAN). En la topología de red, el LoRa funciona en la topología de estrella (una central/concentrador y N nos/ end devices) o punto a punto, dependiendo de la cantidad de radios LoRa en el enlace en cuestión. (Bartoletti, Paiotti, & Leca, 2019)

- **Topología de estrella:** es una configuración para una red en la que se puede tener N nodos (esclavos) conectados a un punto central (maestro), El nodo central (maestro) debe de ser un procesador rápido y potente puesto que por el pasara todo el tráfico de datos de los demás nodos.
- **Topología punto a punto:** como su nombre lo dice es una red que se puede comunicar solo de un punto A hacia otro punto B, es decir solo se puede conectar dos dispositivos (A y B) sin importar cuál sea el emisor y el receptor, en este tipo de red los dispositivos actúan como socios iguales es decir no existe ni maestro ni esclavo. Cualquiera de los dos dispositivos pueden trabajar como emisor y receptor en un lapso diferente de tiempo.

### 2.3.2 Alcance

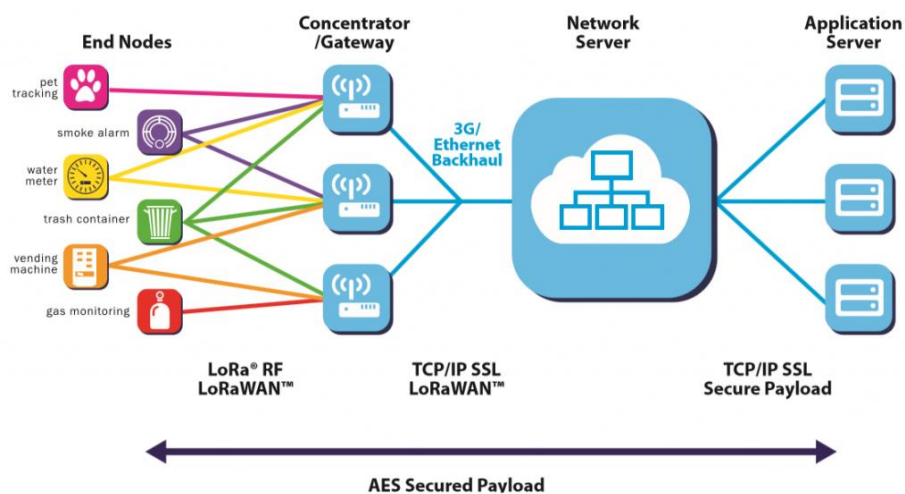
La radio frecuencia de LoRa tiene un alcance muy grande en comparación con otros módulos de radio frecuencia, si se utilizan antenas tiene un alcance en el orden de los kilómetros, muy útil para implementar proyectos de telemetría en zonas que son de difícil acceso.

### 2.3.3 Arquitectura LoRa

La arquitectura de la red LoRa consta de una topología en estrella con una o más puertas de enlace que enrutan los mensajes desde los nodos del cliente al servidor de la red central. (Candia & Varela, 2017). A continuación, se describirá los elementos de la arquitectura de la red LoRa.

- **Nodos:** Son los componentes que permiten la conexión de los actuadores o sensores finales y que van a estar en contacto con directo con el objeto. Pueden transmitir o recibir información del Gateway.
- **Puerta de enlace (Gateway):** El Gateway en una red LoRa es el encargado de recibir información de los nodos y transmitir hacia los servidores y viceversa.
- **Servidores de red:** Los equipos servidores de red son los receptores de la información enviada por el Gateway para poder procesar dicha información.
- **Servidor de aplicaciones:** Son dispositivos de software que permiten servicios de aplicación a los nodos clientes, en su mayoría gestionan la mayor parte de las funciones lógicas.

Figura 2.7. Arquitectura LoRa



Fuente: (Alliance, 2015)



## CAPÍTULO III

### DISEÑO E IMPLEMENTACIÓN

Este capítulo, analiza los requerimientos técnicos para el diseño e implementación del sistema de telemetría aplicables al prototipo de competencia Shell Eco Marathon de la Universidad Politécnica Salesiana, se mencionará una descripción del software y hardware utilizado, para la construcción del sistema emisor y receptor, se muestra los módulos usados, programación Arduino, interfaz en LabVIEW, envío de datos, recepción de datos, diseño e implementación.

#### 3.1 Módulos

- **Arduino Mega 2560**

Es una placa de desarrollo compuesta de un microcontrolador Atmel ATmega2560, de ahí su nombre. Además, existen 54 pines de entrada y salida digital, además 15 de los cuales se usan como salidas PWM (modulación por ancho de pulso). Además, se puede utilizar 16 entradas analógicas, 4 UART como hardware de puerto serie, un oscilador de cristal de 16 MHz, conexión USB., Conector de alimentación, cabezal ICSP, botón de reinicio. Especificaciones técnicas Especificaciones técnicas (ANEXO I).

Figura 3.1. Modulo Arduino Mega2560



Fuente: (Arroyo, 2020)

- **Arduino Uno**

El Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328, que ofrece numerosas funciones y herramientas, como 14 entradas/salidas digitales, Seis de ellas se utilizan como salida PWM (modulación de ancho de pulso) mientras que otras 6 como entradas analógicas un resonador cerámico de 16 MHz, un conector para USB tipo hembra, un Jack para fuente de Poder, un conector ICSP y un botón reset. Especificaciones técnicas, (ANEXO I).

Figura 3.2. Modulo Arduino Uno



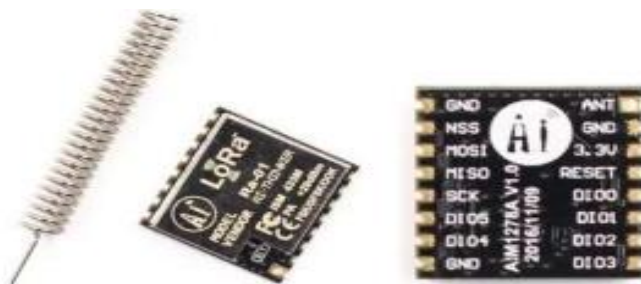
Fuente: (arduino.cc, 2022)

- **Módulos LoRa Ra-01 SX1278**

El módulo SX1278 Ra-01 posee características ideales para el prototipo Shell ya que posee un largo alcance y un bajo consumo energético. Trabaja bajo el protocolo LoRaWAN, nos permite configurar el dispositivo de manera simplex, semidúplex, dúplex según los parámetros deseados para el proyecto. Además, este módulo consta de una antena primavera. Especificaciones técnicas, (ANEXO II).

Estos módulos son capaces de cumplir protocolos y estándares como IEEE 802.15.4g y WMBus. Utilizados en una variedad de aplicaciones de banda médica, científica e industrial (ISM), los transpondedores de RF tienen lectores de medidores de largo y corto alcance para una variedad de elementos naturales. (Satían&Cantuña, 2019)

Figura 3.3. LoRa Ra-01 SX1278



Fuente: (Amazon, 2021)

- **Regulador DC-DC LM2596**

Este dispositivo es un módulo que cumple con la función de regular un voltaje a partir de una fuente de alimentación, esto permite trabajar con microcontroladores como Arduino, PIC's. Entre sus principales características posee un circuito integrado monolítico y una precisa regulación de voltaje, por medio de este módulo se logra simplificar el diseño en placa. Especificaciones técnicas, (ANEXO III). (Morocho Arroyo, 2020)

Figura 3.4. Regulador DC-DC LM2596.



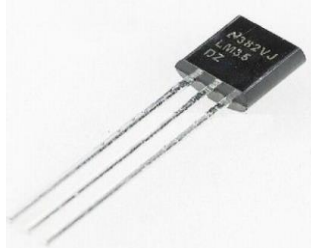
Fuente: (Arroyo, 2020)

### 3.2 Elementos electrónicos

- **Sensor de temperatura LM35**

Este componente electrónico puede medir temperaturas de  $-55^{\circ}\text{C}$  a  $150^{\circ}\text{C}$ , su salida es analógica es muy usado en la electrónica ya que no es necesario tener un microprocesador o microcontrolador para medir la temperatura, el LM35 proporciona 10 mV por cada grado centígrado. Especificaciones técnicas, (ANEXO IV).

Figura 3.5. Sensor LM35.

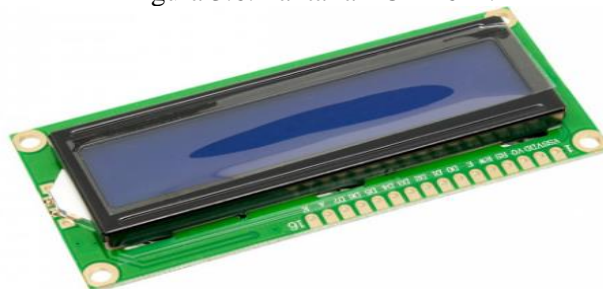


Fuente: Guato Brandon, Yáñez Mario

- **Pantalla LCD 16x2**

La LCD de 16x2 es un módulo de visualización electrónica de 16 caracteres y 2 líneas se muestra con iluminación de fondo azul o verde con letras blancas, se puede utilizar fácilmente con varios microcontroladores. Especificaciones técnicas, (ANEXO V).

Figura 3.6. Pantalla LCD 16x2.



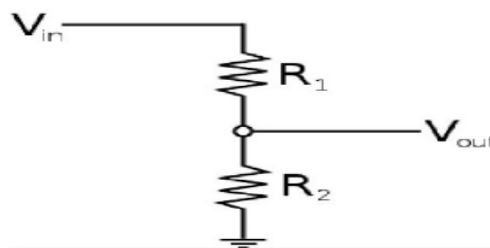
Fuente: Guato Brandon, Yáñez Mario

- **Divisor de tensión**

Su principal función es oponer cierta dificultad al paso de la corriente eléctrica, además cumplen varias funciones como polarización, atenuación, divisor de tensión, limitador de corriente, carga, filtrado. Es de vital importancia elegir bien cada resistencia para un funcionamiento correcto en el dispositivo electrónico.

El divisor de tensión nos permite configurar un circuito electrónico, gracias a que divide la tensión de la fuente entre dos o más resistencias conectadas en serie.

Figura 3.7. Divisor de tensión.

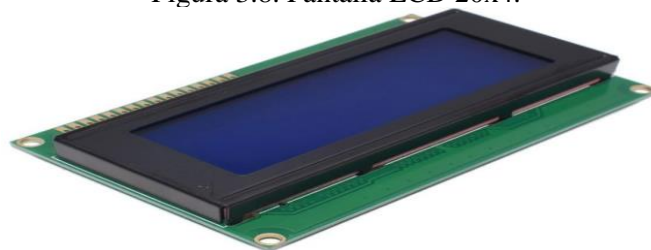


Fuente: Guato Brandon, Yáñez Mario

- **Pantalla LCD 20x4**

La LCD 20x4 es una pantalla de 20 caracteres por línea, en 4 líneas con iluminación en fondo azul o verde. Esta pantalla visual electrónica posee propiedades de modulación por luz es ideal para diversos proyectos electrónicos. Especificaciones técnicas, (ANEXO V).

Figura 3.8. Pantalla LCD 20x4.

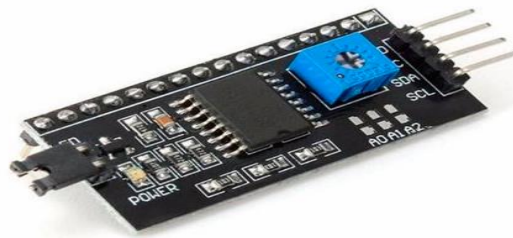


Fuente: Guato Brandon, Yáñez Mario

- **Módulo adaptador serial I2C para pantalla LCD**

Este módulo permite conectar una pantalla I2C a un Arduino usando únicamente dos pines digitales (SDA y SCL). Este adaptador se puede usar para convertir cualquier pantalla estándar de 16x2 o 20x4 de manera serial, además puede compartir con otros dispositivos compatibles con el estándar I2C como: reloj en tiempo real DS1307, memoria EEPROM, controladores de servos. Especificaciones técnicas, (ANEXO VI).

Figura 3.9. Módulo adaptador serial I2C para pantalla LCD



Fuente: Guato Brandon, Mario Yáñez

### 3.3 Selección de Sensores

#### 3.3.1 Definición de sensor

Un sensor es un dispositivo capaz de captar magnitudes físicas y convertirlas en variables eléctricas.

#### 3.3.2 Sensor TPS (Throttle Position Sensor)

Como su nombre en inglés, es un sensor de posición del acelerador. Está conectado directamente a la válvula de aceleración. Este sensor funciona como un potenciómetro o divisor de tensión. El sensor TPS Convierte la posición del ángulo del acelerador en una señal eléctrica de entrada, puede aumentar o disminuir la cantidad de gasolina que ingresa al motor. Además, cuenta con tres pines.

Con conector de 3 pines.

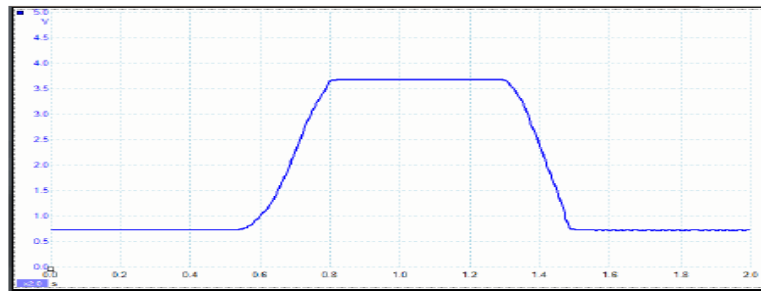
- Alimentación: +5V
- Tierra: Un pin GND
- Salida (Señal): Un pin de salida de la señal de forma analógica.

Figura 3.10. Sensor TPS



Fuente: (Arroyo, 2020)

Figura 3.11. Señal del sensor TPS



Fuente: Guato Brandon, Yáñez Mario

### 3.3.3 Sensor ECT (Engine Coolant Temperature)

Este sensor es el que mide la temperatura del refrigerante del motor. La ubicación de este sensor es por lo general enroscado en el interior del bloque del motor. Funciona como un NTC esto significa, que al aumentar la temperatura la resistencia del sensor cae. La señal que registra este sensor ingresa a la ECU, la cual controla la activación o desactivación del electroventilador para el enfriamiento del motor. Especificaciones técnicas (ANEXO VII).

Figura 3.12. Sensor ECT



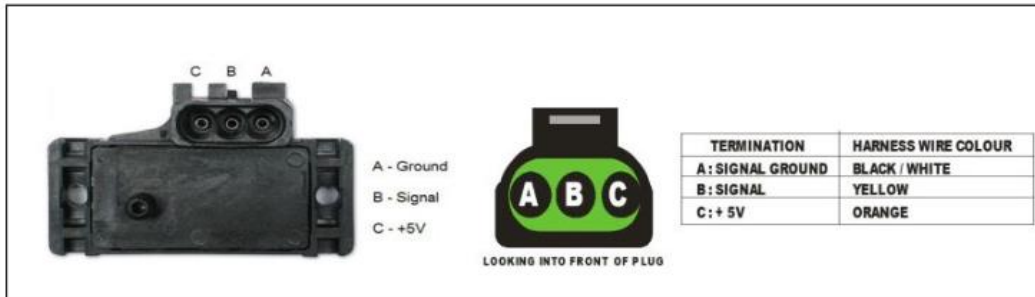
Fuente: (Arroyo, 2020)

### 3.3.4 Sensor MAP (Manifold Absolute Pressure)

MAP (presión absoluta del colector) se encarga de proporcionar el flujo de aire que ingresa al motor, es necesario saber dicho flujo de aire para calcular la cantidad de combustible que inyectara la ECU al motor para su correcto funcionamiento. Este sensor monitorea constantemente la presión del colector, además envía señales de voltaje a la ECU, la cual aumentará o disminuirá el ancho del pulso de inyección. Por lo general tienen 3 pines de conexión. Especificaciones técnicas, (ANEXO VIII).

- Alimentación: +5V
- Tierra: Un pin GND
- Salida (Señal): Un pin de salida de la señal de forma analógica.

Figura 3.13. Sensor MAP (Haltech)



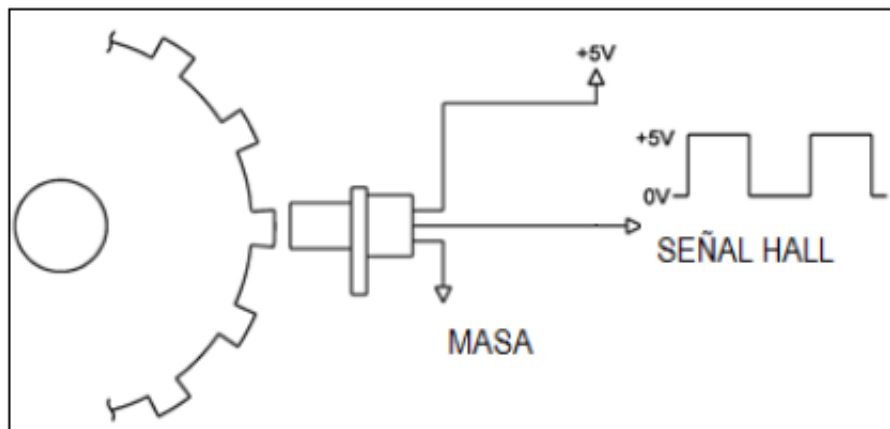
Fuente: (Reinoso , Salazar , & Enríquez , 2012)

### 3.3.5 Sensor CKP (Crankshaft Position Sensor)

Sensor de posición del cigüeñal (CKP), indica a la ECU las revoluciones por minuto del motor y así determinar el momento preciso para para la inyección de gasolina y la chispa de la bobina. La ubicación de este sensor debe de ser lo más cercano a la rueda fónica del motor, dicha rueda tiene cierta cantidad de dientes y un faltante el cual nos indica el punto muerto superior.

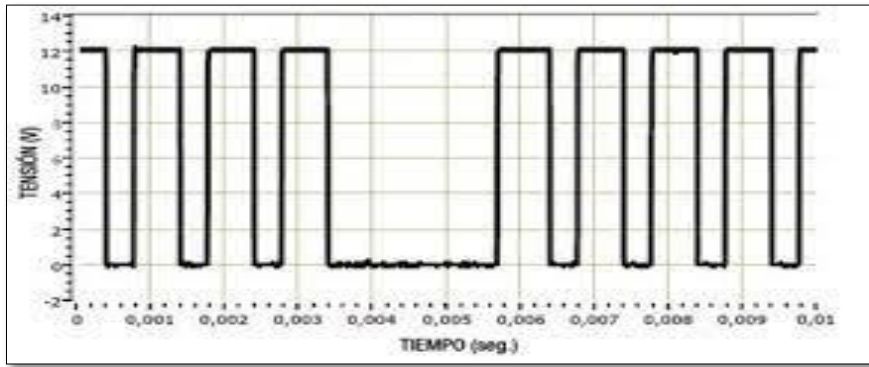
Existen dos tipos de sensores CKP un analógico (inductivo) y un digital (efecto hall), la diferencia entre estos dos sensores es que el analógico genera su propio voltaje inductivamente y el digital es de efecto hall el cual debe de siempre estar suministrado por una fuente de alimentación.

Figura 3.14. Tipos de sensores CKP



Fuente: (Cornejo & Tintin, 2010).

Figura 3.15. Señal del sensor CKP (Hall)

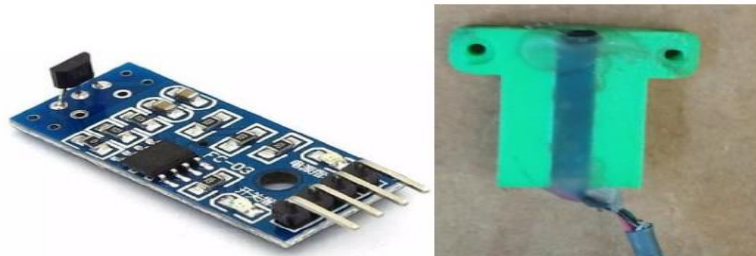


Fuente: Guato Brandon, Yáñez Mario

### 3.3.6 Sensor WSS (Wheel Speed Sensor)

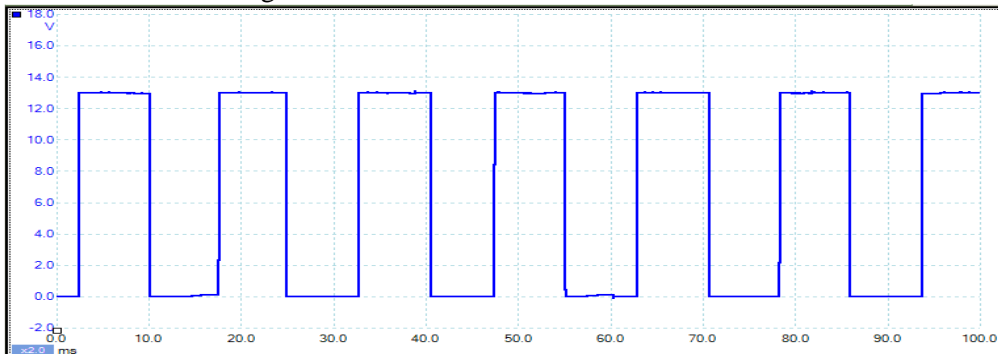
WSS (Wheel Speed Sensor) conocido como sensor de velocidad de rueda. Este sensor es de tipo efecto Hall. Es decir, cuando se aplica un campo magnético en la dirección perpendicular al sensor, produce una señal digital. Especificaciones técnicas, (ANEXO IX).

Figura 3.16. Sensor WSS



Fuente: (Arroyo, 2020)

Figura 3.17. Señal del sensor WSS efecto Hall



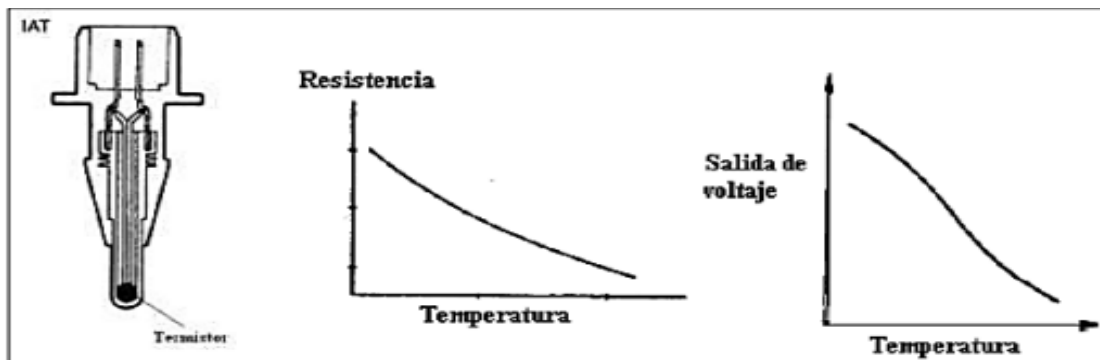
Fuente: Guato Brandon, Yáñez Mario



### 3.3.7 Sensor IAT (Intake Air Temperature)

IAT (Intake Air Temperature) en español temperatura en la toma de aire, es un termistor de tipo NTC, es decir a medida que su resistencia baja el nivel de temperatura aumenta, este sensor nos indica el valor de la temperatura que está ingresando al motor enviando la señal hacia la ECU y así esta poder determinar el nivel de gasolina que inyectara al motor.

Figura 3.18. Señal del sensor WSS efecto Hall



Fuente: (Bonnick, 2001)

### 3.4 Fase de desarrollo

En este apartado analizaremos el desarrollo, la implementación y diseño del sistema de telemetría aplicada al prototipo de competencia Shell Eco Marathon de la Universidad Politécnica Salesiana, a través del desarrollo de una fase de programación y diseño preliminar, pruebas, desarrollo del sistema definitivo y obtención de resultados en tiempo real.

### 3.5 Librerías

Con el fin de realizar una comunicación, enlace y correcto funcionamiento entre los dispositivos de comunicación como LoRa, I2C, IDE ARDUINO entre otras, es preciso incluir algunas librerías que permiten realizar estas tareas en la siguiente tabla se muestran las librerías que se usaron y la fuente donde se adquirieron.

Tabla 3.1 Librerías

Módulos	Librería original	Librería externa	Url	Definición
IDE ARDUINO	<SPI>	-	-	Habilita la comunicación en serial
	<wire.h>	-	-	Habilita la comunicación I2C / TWI
LoRa	-	<LoRa.h>	“GitHub - sandeepmistry/arduino-LoRa: An Arduino library for sending and receiving data using LoRa radios.”	Permite la transmisión y recepción de datos LoRa.
LCD I2C	-	<LiquidCrystal_I2C.h>	“https://github.com/marcoschwarz/LiquidCrystal_I2C”	Habilita la comunicación LCD con I2C

Fuente: Guato Brandon, Mario Yáñez

### 3.6 Fase Preliminar

En esta etapa se diseñó un prototipo del emisor y receptor lo cual será implementado en Protoboard para las pruebas pertinentes de envío y recepción de datos, para el diseño del emisor encontramos sensores como MAP, TPS para la cual adaptaremos un potenciómetro de un  $1k\Omega$  y un sensor de temperatura LM35 ya que dichos componentes emiten señales analógicas similares a los sensores antes mencionados y para la simulación de señales digitales usaremos un pulsador en cual nos permite conocer una señal electrónica Pull-Up o Pull-Down.

Además, se implementará dos pantallas LCD de 16x2 y de 20x4 con los módulos I2C las cuales nos permitirán visualizar las variables en tiempo real, a la vez se conectará el módulo LoRa Ra-01 SX1278 el cual nos permite establecer una comunicación SPI con el Arduino Mega y el Arduino Uno.

El módulo LoRa Ra-01 SX1278 utiliza una antena para una comunicación de radiofrecuencia adecuada, misma que posee un alcance de hasta 10 km según las especificaciones técnicas del fabricante, tiene un total de 16 pines y trabaja con frecuencias de 420 a 450 MHz.

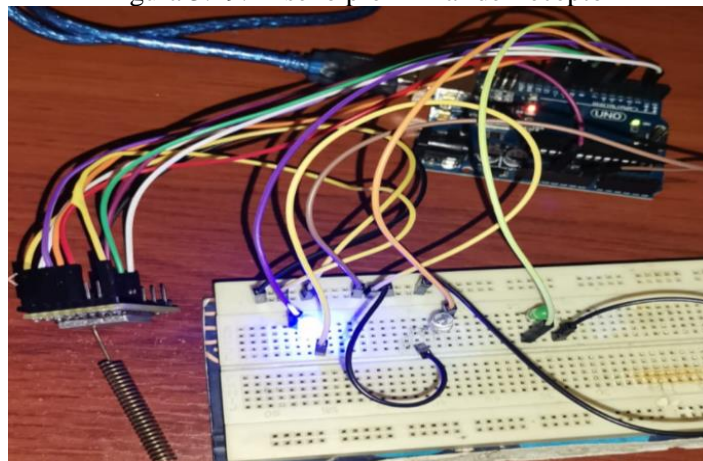
Para el control se usó un microcontrolador Arduino Mega, Arduino Uno el cual cumple los parámetros deseados para la implementación.

Para energizar los componentes se utilizó una fuente de 12VDC y un regulador LM2596 DC-DC el cual permite regular el voltaje para los diferentes componentes del sistema.

Para el prototipo del receptor que se implementó en un Protoboard se usó un microcontrolador Arduino Uno el cual permite recibir las diferentes señales analógicas y digitales; a la vez se conectó el módulo LoRa Ra-01 SX1278 para recibir mediante radiofrecuencia las diferentes señales enviadas por el emisor.

Para el diseño preliminar se agregó 2 LEDS y 2 resistencias de 220  $\Omega$  cada una, como indicadores de una comunicación correcta en azul y en rojo si no existe algún fallo en la comunicación tanto para el diseño del emisor como la del receptor.

Figura 3.19. Diseño preliminar del receptor



Fuente: Guato Brandon, Yáñez Mario

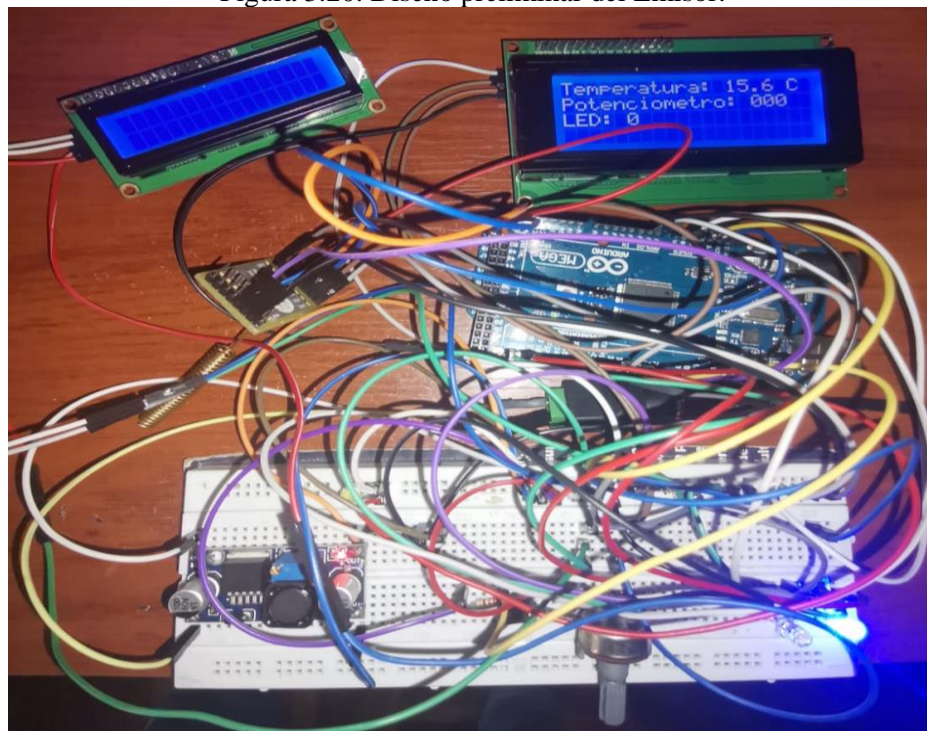
En la Figura 3.19 se puede apreciar las diferentes conexiones del módulo LoRa al Arduino Uno a continuación se indica el diagrama de conexión.

Tabla 3.2 Conexiones de pines del módulo LoRa a los pines del Arduino Uno.

LoRa Ra-01 SX1278	Arduino Uno
NSS	10
MOSI	11
MISO	12
SCK	13
3.3V	3.3V
RESET	9
DIO0	2
GND	GND

Fuente: Guato Brandon, Yáñez Mario

Figura 3.20. Diseño preliminar del Emisor.



Fuente: Guato Brandon, Yáñez Mario

En la Figura 3.20. se puede apreciar las diferentes conexiones del módulo LoRa al Arduino Mega a continuación se indica el diagrama de conexión.

Tabla 3.3 Conexiones de pines del módulo LoRa a los pines del Arduino Mega.

LoRa Ra-01 SX1278	Arduino Mega
NSS	53
MOSI	51
MISO	50
SCK	52
3.3V	3.3V
RESET	9
DIO0	2
GND	GND

Fuente: Guato Brandon, Yáñez Mario

## 3.7 Desarrollo De La Programación

### 3.7.1 Sensores

- **TPS**

El sensor determina la posición de la aleta de aceleración. Para poder obtener la señal de este sensor utilizamos el canal Analógico A0 y la función map de Arduino para poder transformar las unidades crudas (0 a 1023) en un porcentaje (0 a 100 %), el cual tendrá la relación con la aleta del cuerpo de aceleración.

Programación TPS:

```
float TPS=A0; // Variable flotante que lee la entrada analógica A0
float lectura1; // Variable flotante que lee las unidades crudas de la
entrada analógica
float TPSFIN; // Valor final de la lectura del sensor TPS
void setup() {
  Serial.begin(9600);
  pinMode(TPS,INPUT); // pin entrada
}

void loop() {
lectura1= analogRead(TPS); // lee la entrada analógica en unidades
crudas 0 a 1023
TPSFIN = map(analogRead(TPS),24,868,1,100); // Realiza la conversion a
porcentaje
Serial.print("lectura: ");
Serial.print(lectura1); // Imprime las unidades crudas para conocer el
mínimo y máximo valor del analógico A0

Serial.print(" SEÑAL DE TPS: ");
Serial.print(TPSFIN); // Imprime el valor en porcentaje del TPS
Serial.print("%");

Serial.println();
}
```

- **MAP**

El sensor MAP nos indica la presión absoluta en el colector, esta presión dependerá del motor. Utilizamos la entrada analógica A3 y la función map de Arduino para poder transformar las unidades crudas (0 a 1023) en unidades de presión entre 25 y 75 KPa que es lo que leerá el sensor MAP.

Programación MAP:

```
int MAP=A3; // Variable entera que lee la entrada analogica A0
int lectura2; // Variable flotante que lee las unidades crudas de la
entrada analogica
int Pabs; // Variable que almacena el valor de la presion absoluta
void setup() {
  Serial.begin(9600);
```

```

pinMode(MAP, INPUT);
}

void loop() {
  lectura2= analogRead(MAP); // lee la entrada analogica en unidades
  crudas 0 a 1023
  Pabs = map(lectura, 249, 620, 27, 70); // Realiza la conversion a KPa
  Serial.print("Unidades Crudas: ");
  Serial.print(lectura2); // Imprime las unidades crudas para conocer
  el minimo y maximo valor del analogico A0

  Serial.print(" PRESION ABSOLUTA: ");
  Serial.print(Pabs); // Imprime el valor en KPa del sensor MAP
  Serial.print("Kpa");

  Serial.println();
}

```

- **CKP**

Crankshaft Position Sensor indica a la ECU el momento en el cual debe poner la chispa para la mezcla aire combustible en el cilindro, nos permite conocer las revoluciones del motor. En este caso este sensor es de efecto Hall por lo cual, utilizaremos una entrada digital en el Pin 2 del sistema emisor.

Programación CKP:

```

float revoluciones=0; // Variable flotante inicializada en 0 para
  contar las interrupciones
int rpm1=0; // Variable entera inicializada en 0 para las revoluciones
long start=0;
long elapsed;

void setup()
{
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP); // establece el pin en la entrada
}

void loop() {
  revoluciones=0; rpm1=0;
  start=millis();
  attachInterrupt(digitalPinToInterrupt(2), interrupt, RISING); //
  configuracion del pin para la interrupcion
  delay(1000);
  detachInterrupt(2);

  //Ahora veamos cuantos conteos hemos tenido del sensor de efecto Hall
  y calculemos las RPM
  elapsed=millis()-start; // encuentra el tiempo, debe estar muy cerca
  de 1 segundo

  if(revoluciones>0)
  {
    rpm1=(max(1, revoluciones) * 2100) / elapsed; //calcula rpm //calcula
    el maximo de dos numero y devuelve el mayor
  }
}

```

```

}
String outMsg = String("RPM :") + rpm1; // Muestra en el monitor
serial las revoluciones
Serial.println(outMsg);
}

void interrupt() //interrumpir rutina de servicio
{
  revoluciones++;
}

```

- **IAT**

Es un termistor con sus siglas(Intake Air Temperature) de tipo NTC, es decir que mediante la temperatura sube la resistencia del sensor baja, este sensor mide la temperatura del aire que ingresa al motor. Utilizamos la entrada analógica A5, la señal de este sensor es enviada a la ECU y así poder realizar una buena combustión en el motor.

#### Programación IAT:

```

#define resistencia 10000
#define sensortemp A5 //define el pin analógico del voltaje de entrada
del divisor de voltaje
#define intera 5 //define el número de muestras que se tomarán para un
promedio uniforme
const float steinconstA = 0.00132774106461327; //constante A de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia
const float steinconstB = 0.000254470874104285; //constante B de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia
const float steinconstC = 0.000000101216538378909; //constante C de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia

int muestra[intera]; //variable para almacenar el número de muestras a
tomar

void setup() {
  Serial.begin(4800); //iniciar monitor serie
}

void loop() {
  uint8_t i; //entero para bucle
  float promedio; //para el promedio variable decimal

  for (i=0; i<intera; i++) {
    muestra[i] = analogRead(sensortemp); //toma muestras en el número
definido con un breve retraso entre muestras
    delay(10);
  }

  promedio = 0;
  for (i=0; i< intera; i++) {
    promedio += muestra[i]; // suma todo el número de muestras juntas
  }
  promedio /= intera; //divide por número de muestras para generar el
promedio
}

```

```

promedio = (resistencia*promedio)/(1023-promedio); //ecuación de
conversión para leer la resistencia del divisor de voltaje
float hart; //ecuación de Steinhart para estimar el valor de la
temperatura en cualquier resistencia de la curva del sensor del
termistor
hart = log(promedio);
hart = pow(hart,3);
hart *= steinconstC;
hart += (steinconstB*(log(promedio)));
hart += steinconstA;
hart = 1.0/hart;
hart -= 273.15;
Serial.print("Temperatura = ");
Serial.print(hart); //imprime la temperatura final en celcius
Serial.println(" *C");

delay(1000); //retraso entre lecturas
}

```

- **WSS**

Wheel Speed Sensor o sensor de velocidad en español nos permite conocer la velocidad del prototipo, este sensor es de tipo efecto hall entonces para la programación utilizamos un pin digital (digital 2) con interrupciones para poder conocer la velocidad exacta del prototipo, dicha interrupción se efectuará siempre que se detecte un flanco de subida en el pin.

Programación WSS:

```

float revoluciones=0; // Variable flotante inicializada en 0 para
contar las interrupciones
int rpm1=0; // Variable entera inicializada en 0 para las revoluciones
long start=0;
long elapsed;

void setup()
{
Serial.begin(9600);
pinMode(2, INPUT_PULLUP); // establece el pin en la entrada
}

void loop() {
revoluciones=0; rpm1=0;
start=millis();
attachInterrupt(digitalPinToInterrupt(2),interrupt,RISING); //
configuracion del pin para la interrupcion
delay(1000);
detachInterrupt(2);

//Ahora veamos cuantos conteos hemos tenido del sensor de efecto Hall
y calculemos las RPM
elapsed=millis()-start; // encuentra el tiempo, debe estar muy cerca
de 1 segundo

if(revoluciones>0)

```



```

{
  rpm1=(max(1, revoluciones) * 2100) / elapsed; //calcula rpm //calcula
el maximo de dos numero y devuelve el mayor
}
String outMsg = String("RPM :") + rpm1; // Muestra en el monitor
serial las revoluciones
Serial.println(outMsg);
}

void interrupt() //interrumpir rutina de servicio
{
  revoluciones++;
}

```

- **ECT**

Engine Coolant Temperature o temperatura refrigerante del motor en español, es un termistor del tipo NTC el cual nos ayuda a verificar la temperatura a la cual el motor está funcionando y así poder evitar un sobrecalentamiento en el motor y que éste deje de funcionar, para este sensor utilizamos la entrada analógica A2.

Programación ECT:

```

#define resistencia 1250
#define sensortemp A2 //define el pin analógico del voltaje de entrada
del divisor de voltaje
#define intera 5 //define el número de muestras que se tomarán para un
promedio uniforme
const float steinconstA = 0.00132774106461327; //constante A de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia
const float steinconstB = 0.000254470874104285; //constante B de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia
const float steinconstC = 0.000000101216538378909; //constante C de la
ecuación de steinhart, determinada a partir de las ecuaciones de
wikipedia

int muestra[intera]; //variable para almacenar el número de muestras a
tomar

void setup() {
  Serial.begin(4800); //iniciar monitor serie
}

void loop() {
  uint8_t i; //entero para bucle
  float promedio; //para el promedio variable decimal

  for (i=0; i<intera; i++) {
    muestra[i] = analogRead(sensortemp); //toma muestras en el número
definido con un breve retraso entre muestras
    delay(10);
  }

  promedio = 0;
}

```

```

for (i=0; i< intera; i++) {
promedio += muestra[i]; // suma todo el número de muestras juntas
}
promedio /= intera; //divide por número de muestras para generar el
promedio
promedio = (resistencia*promedio)/(1023-promedio); //ecuación de
conversión para leer la resistencia del divisor de voltaje

float hart; //ecuación de Steinhart para estimar el valor de la
temperatura en cualquier resistencia de la curva del sensor del
termistor
hart = log(promedio);
hart = pow(hart,3);
hart *= steinconstC;
hart += (steinconstB*(log(promedio)));
hart += steinconstA;
hart = 1.0/hart;
hart -= 273.15;
Serial.print("Temperatura = ");
Serial.print(hart); //imprime la temperatura final en celcius
Serial.println(" *C");

delay(1000); //retraso entre lecturas
}

```

### 3.7.2 Elementos electrónicos

- **Sensor de temperatura LM35**

Para el sensor de temperatura LM35 que será colocado en la cabina del piloto, desarrollaremos una programación la cual, mediante fórmulas matemáticas, expresaremos la temperatura en grados centígrados y se podrá visualizar en el LCD de 16x2, hemos optado por este sensor ya que es muy usado para diversos proyectos en el campo de la electrónica, ya que permite leer temperaturas de -55°C a 150°C.

Programación LM35:

```

int SENSOR2; // variable entrada analógica A0
float TEMPERATURA2; // valor temperatura
float SUMA2; // promedio 5 lecturas

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

void loop(){
  SUMA2 = 0; // valor inicial en 0
  for (int i=0; i < 5; i++){ // bucle 5 veces
    SENSOR2 = analogRead(A0); // entrada analógica A0
    TEMPERATURA2 = ((SENSOR2 * 5000.0) / 1023) / 10; // fórmula matemática
    SUMA2 = TEMPERATURA2 + SUMA2;
    delay(500); // retardo
  }
  Serial.println(SUMA2/5.0, 1); // imprime en monitor serial valor
  //promedio de las 5 lecturas
}

```

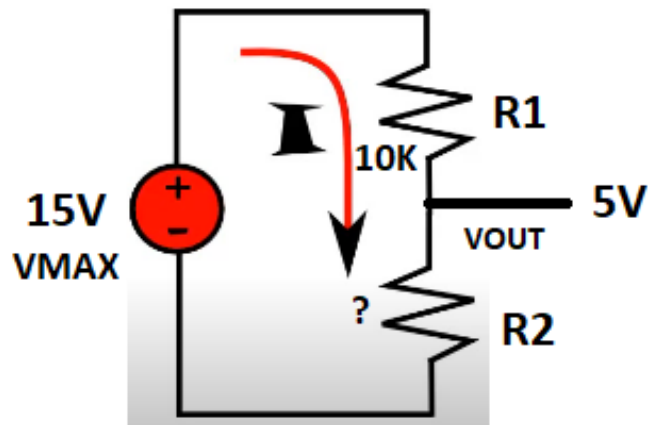
- **Batería**

Para poder visualizar en el LCD de 16x2 el voltaje que nos entrega la batería del prototipo Shell Eco Marathon hemos optado por desarrollar una programación la cual nos permita conocer este voltaje para la cual mediante un divisor de voltaje podremos obtener esta información.

El divisor de voltaje tiene la capacidad de dividir el voltaje de la fuente entre uno o más impedancia. Con solo dos resistencias en serie y un voltaje de entrada se obtiene el voltaje de salida deseado. Además, se puede cambiar el voltaje fijo a un voltaje variable. Para usar un divisor de voltaje es necesario tener dos o mas resistencias conectadas en serie.

### Cálculo de la resistencia R2

Figura 3.21. Divisor de voltaje



Fuente: Guato Brandon, Yáñez Mario  
Tabla 3.4. Cálculo de la resistencia.

$$V1 = VMAX - VOUT \quad \text{Ec. (3.1)}$$

$$V1 = 15V - 5V = 10V$$

$$V1 = R1 \times I \quad \text{Ec. (3.2)}$$

$$\frac{V1}{R1} = I \quad \text{Ec. (3.3)}$$

$$\frac{10V}{10K} = I$$

$$VOUT = R2 \times I \quad \text{Ec. (3.4)}$$

$$VOUT \times I = R2 \quad \text{Ec. (3.5)}$$

$$5V \times I = R2$$

$$5k = R2$$

Fuente: Guato Brandon, Yáñez Mario

## Programación Batería:

```
float voltInput;//Declaramos coma flotante por sus decimales

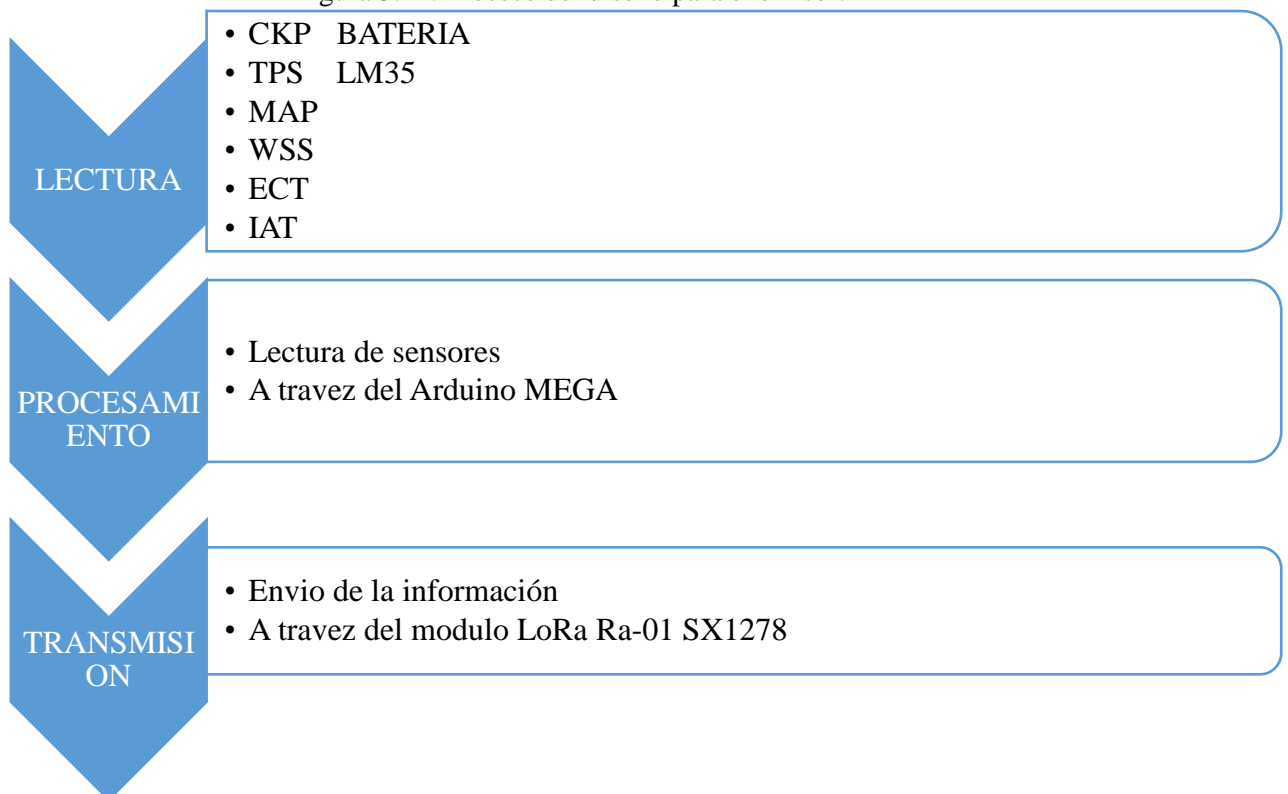
void setup() {
  Serial.begin(9600);
}

void loop() {
  voltInput = ((analogRead(A5))*4.86/1023)*3;//leemos el dato
//analógico y aplicamos la siguiente formula matemática
  Serial.println(voltInput);//imprimimos en el monitor serial
  delay(500);// retardo
}
```

### 3.8 Diseño Del Emisor

Para el dispositivo emisor debemos de tener claro el funcionamiento, esto consta básicamente en enviar los datos de los sensores conectados en la ECU del prototipo de competencia Shell, estos datos ingresan al Arduino Mega el cual los decodificará y mediante el módulo LoRa que también está conectado al Arduino transmitirá la información al módulo receptor y de esta manera se visualizará la información deseada en la interfaz gráfica de LabVIEW.

Figura 3.22. Proceso del diseño para el emisor.

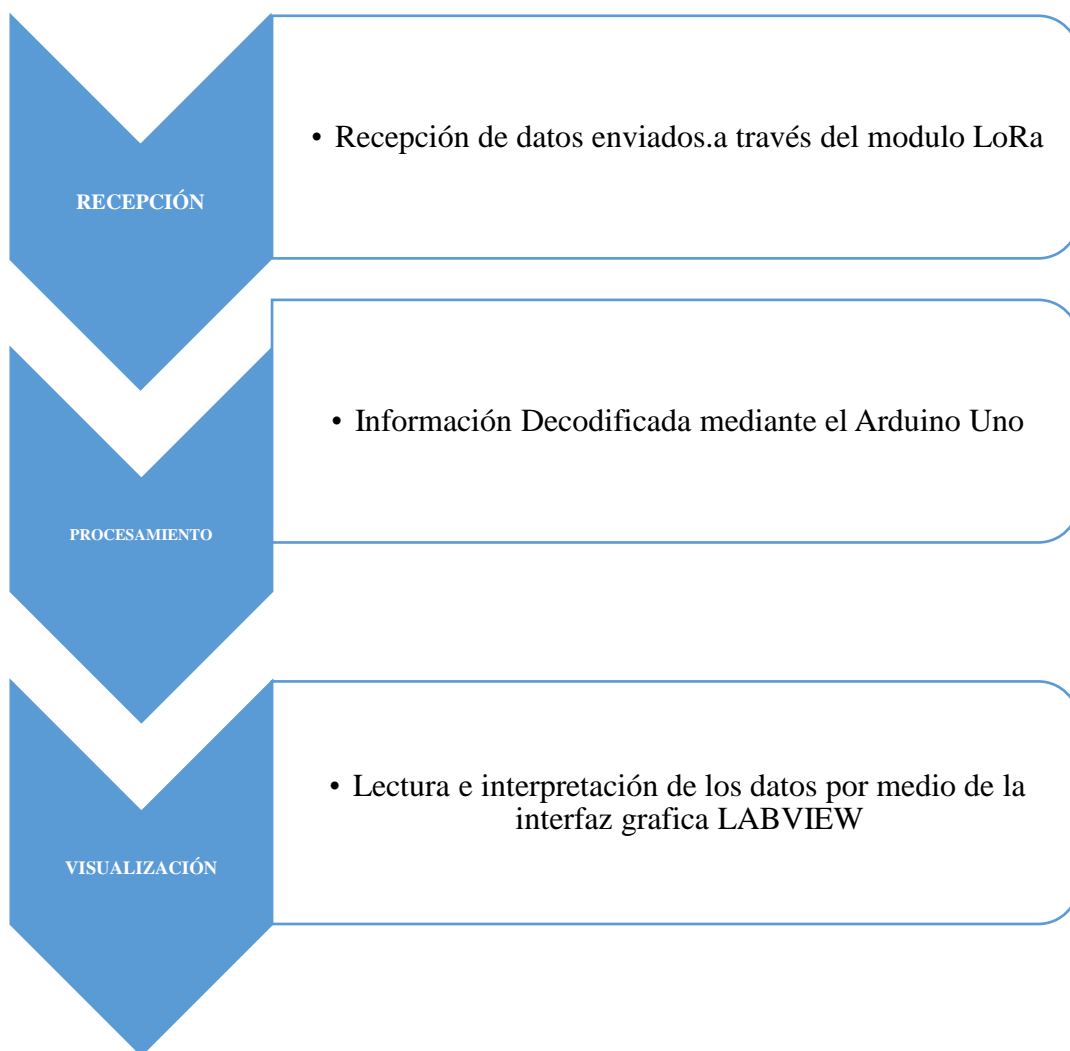


Fuente: Guato Brandon, Mario Yáñez

### 3.9 Diseño Del Receptor

Para el diseño del receptor se definirá su funcionamiento, el cual permite la recepción de datos enviados de los diferentes sensores del prototipo de competencia Shell Eco - Marathon mediante el módulo LoRa, mismo que facilita la comunicación por radiofrecuencia y esta información al ser decodificada por un Arduino Uno a través del código de programación; el mismo que usando la comunicación serial, nos permite visualizar la información procesada e interpretarla en tiempo real en una interfaz gráfica que en este caso es LABVIEW.

Figura 3.23. Proceso del diseño para el receptor.



Fuente: Guato Brandon, Mario Yáñez

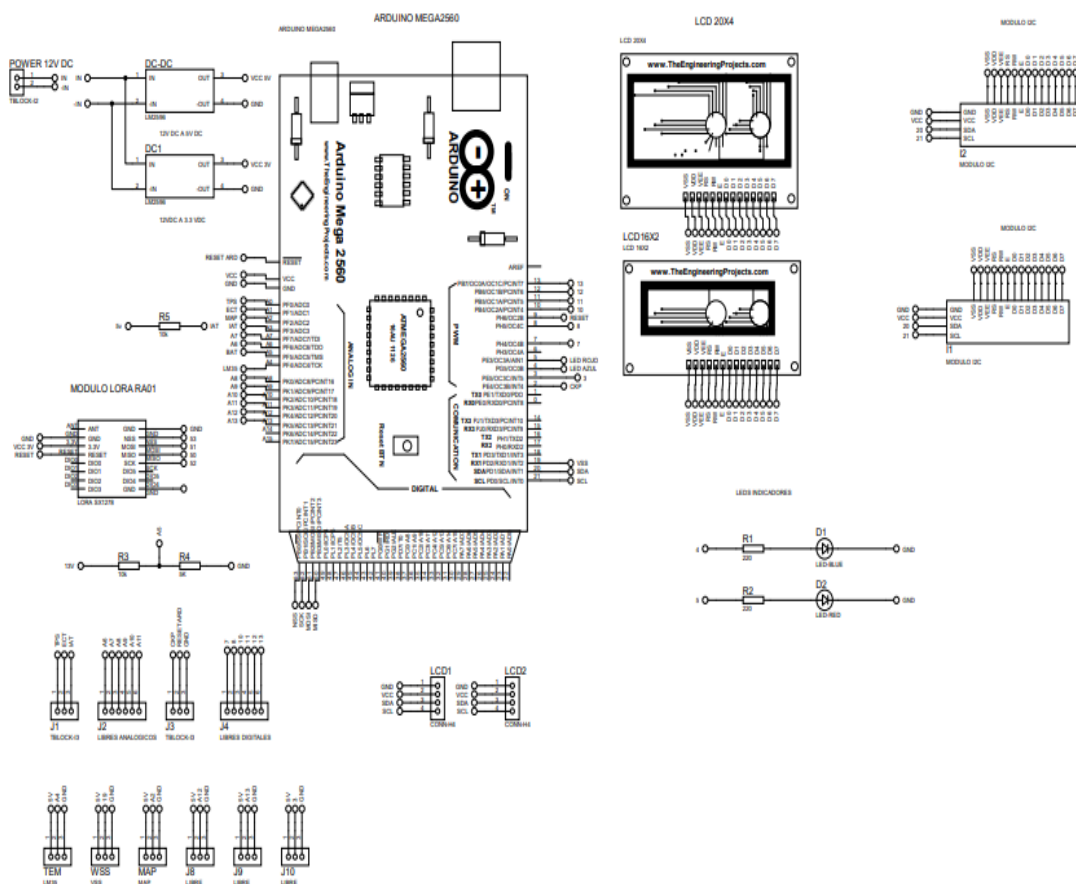
### 3.10 Diseño de Placas en PCB (Placa de Circuito Impreso)

Una placa es un circuito eléctrico que contiene diversos componentes; entre sus principales características que posee encontramos una alta conductividad ya que incluyen pistas de cobre. La estructura consiste en un aislamiento multicapa entre capas de materiales conductores. La estructura general está recubierta y cubierta con una máscara de soldadura no conductora y serigrafiada para colocar componentes electrónicos.

#### 3.10.1 Diseño de placa PCB Emisor

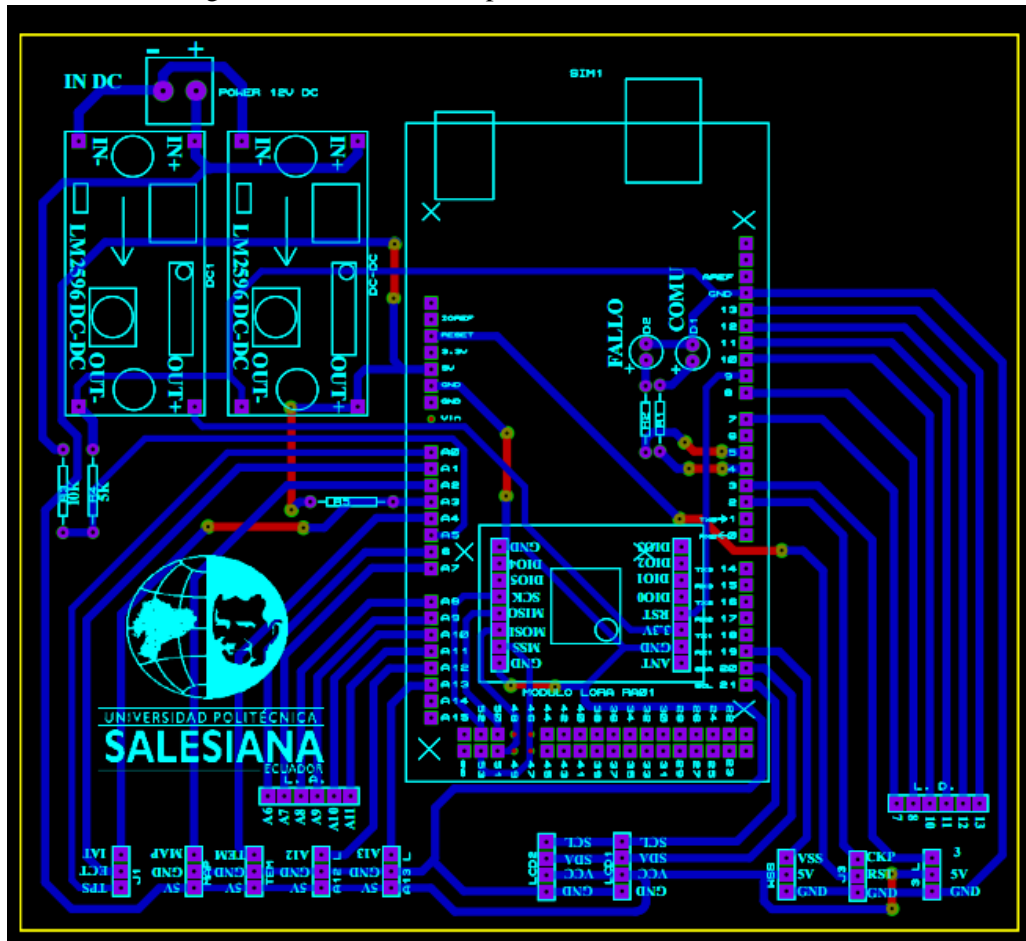
Luego de realizar las pruebas de funcionales correspondientes del emisor se procede a realizar el diseño de la placa PCB en el software de Proteus. Esta herramienta es muy útil para implementar proyectos electrónicos, eléctricos en todas sus fases, como el diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, depuración de errores, documentación y construcción. Cómo se presenta a continuación. Diseño de la placa PCB del emisor en 3D (ANEXO XI).

Figura 3.24. Esquematación del diagrama del Emisor.



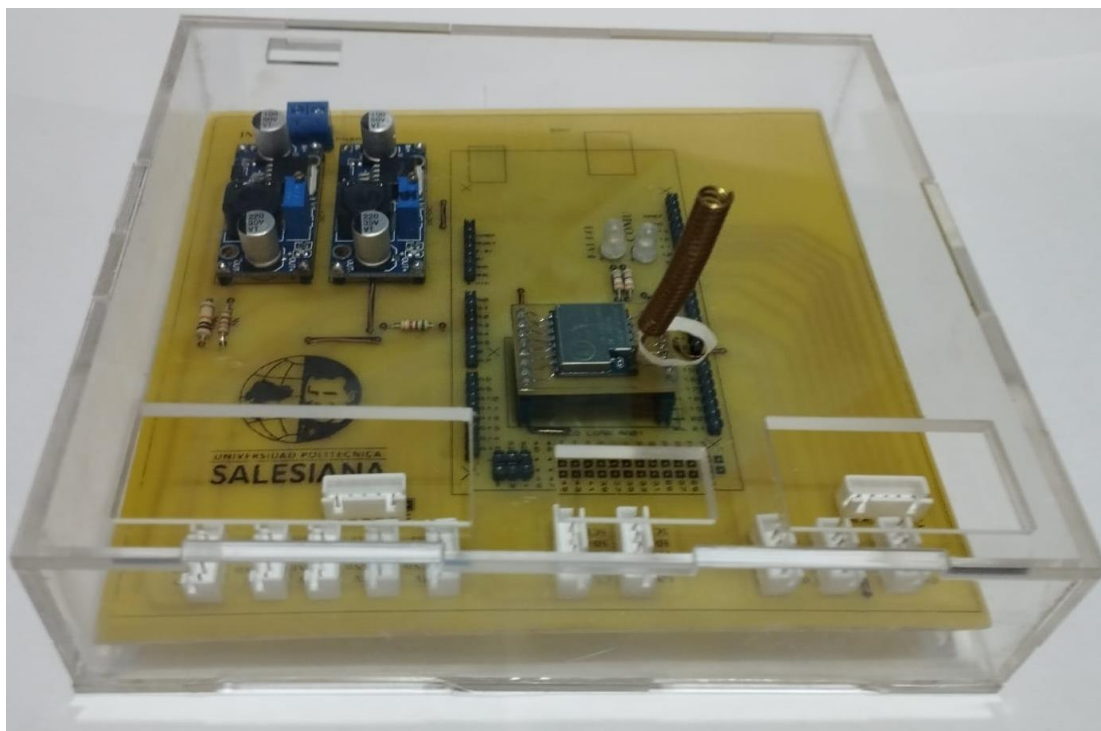
Fuente: Guato Brandon, Yáñez Mario

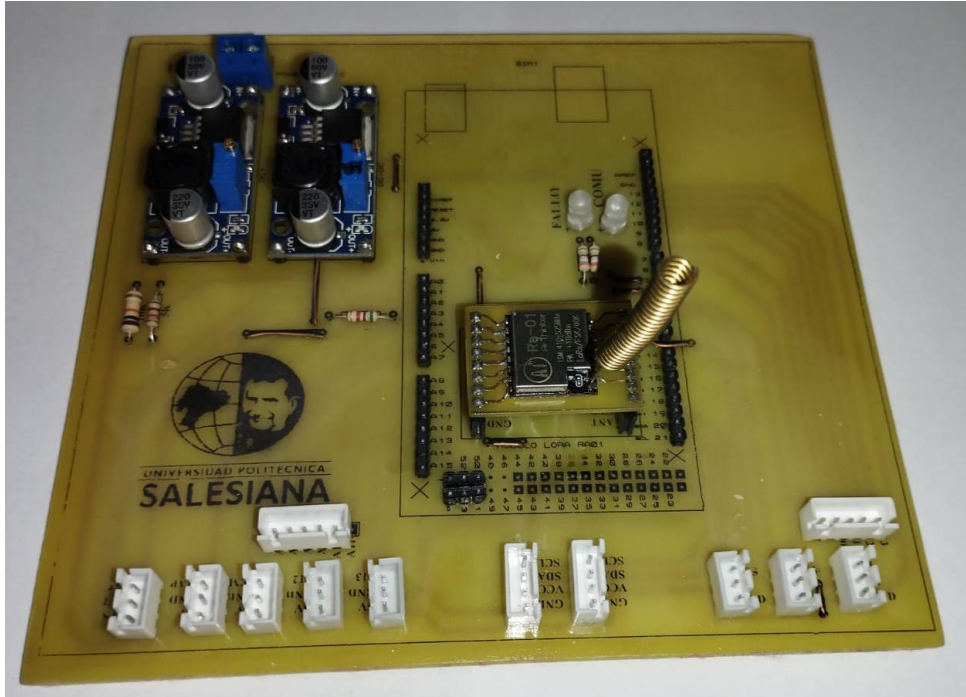
Figura 3.25. Diseño de la placa PCB del emisor.



Fuente: Guato Brandon, Yáñez Mario

Figura 3.26. Montaje de la placa PCB



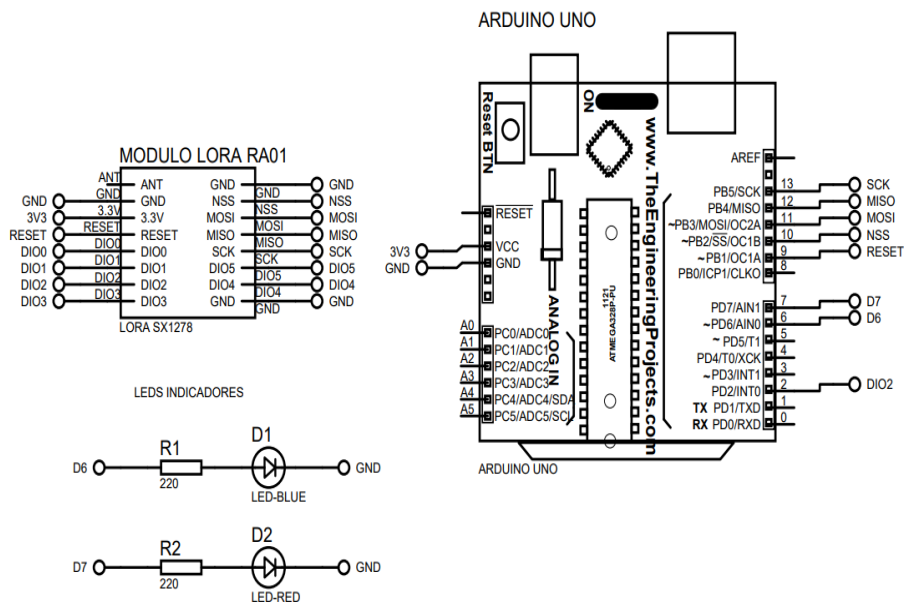


Fuente: Guato Brandon, Yáñez Mario

### 3.10.2 Diseño de placa PCB Receptor

Después de realizar las respectivas pruebas de funcionalidad del sistema receptor se procede al diseño del esquemático y la placa PCB y se presenta a continuación. Diseño de la placa PCB del receptor en 3D, (ANEXO X).

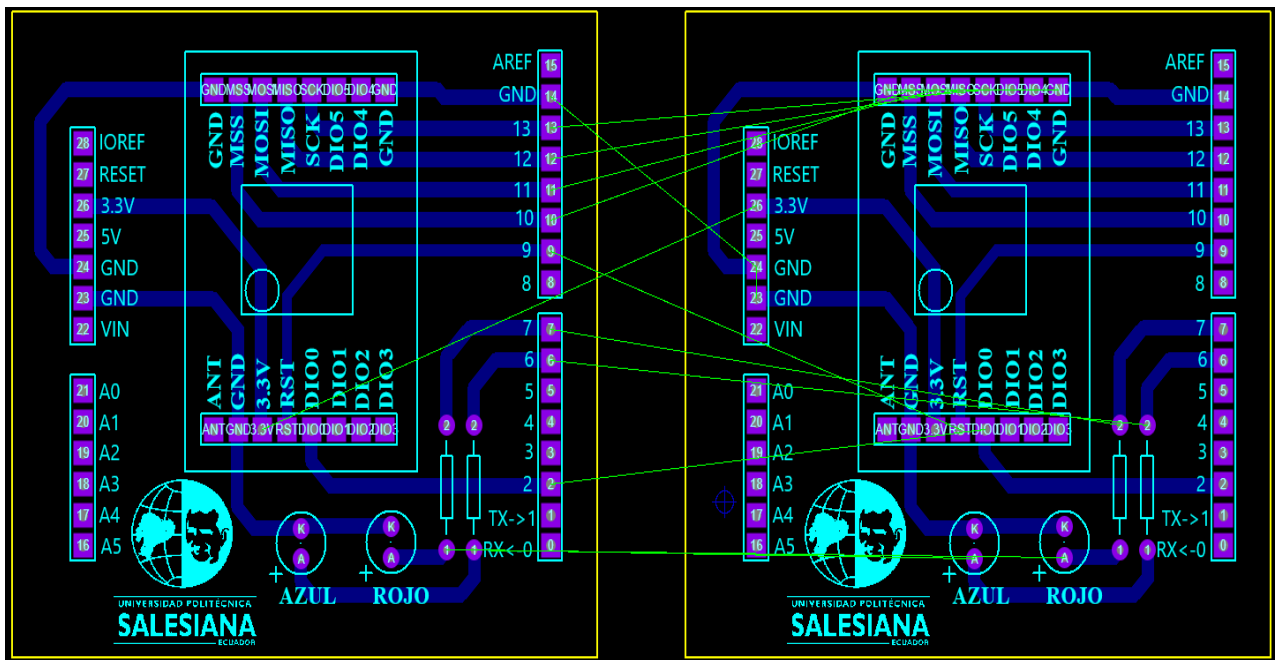
Figura 3.27. Esquematación del diagrama del Receptor



Fuente: Guato Brandon, Yáñez Mario

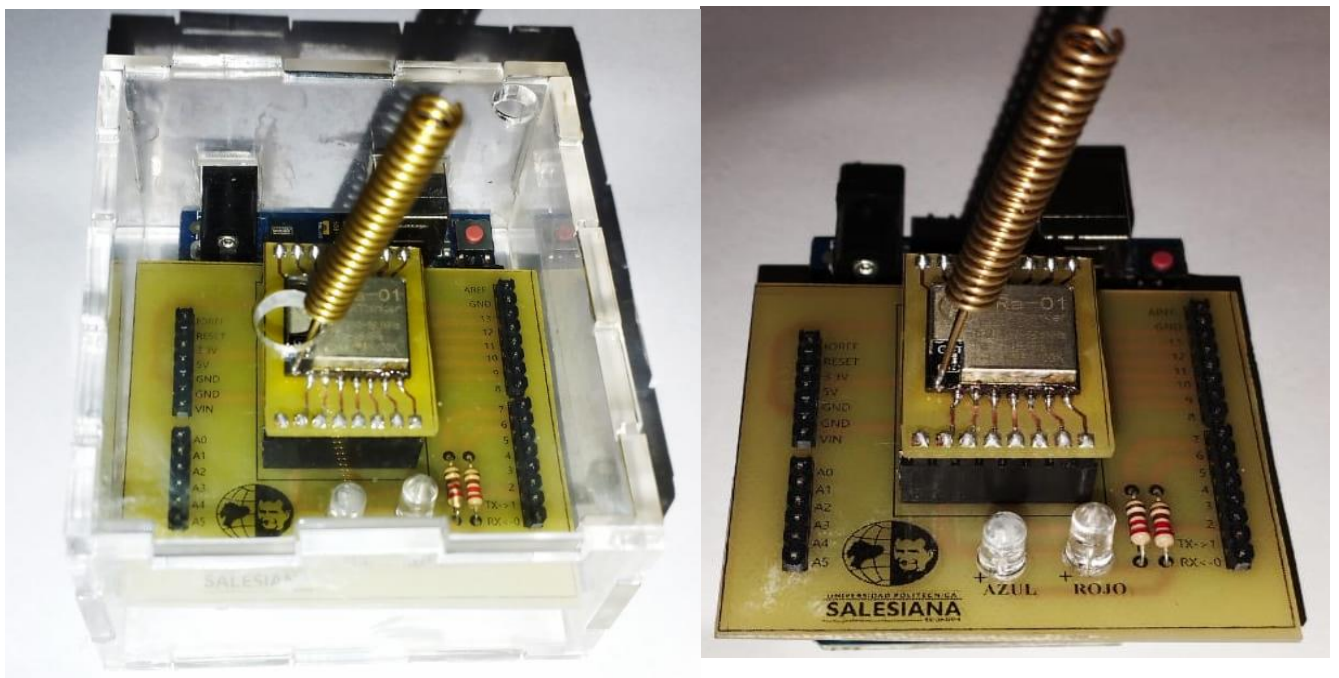


Figura 3.28. Diseño de la placa PCB.



Fuente: Guato Brandon, Yánez Mario

Figura 3.29. Montaje de la placa PCB.



Fuente: Guato Brandon, Yánez Mario

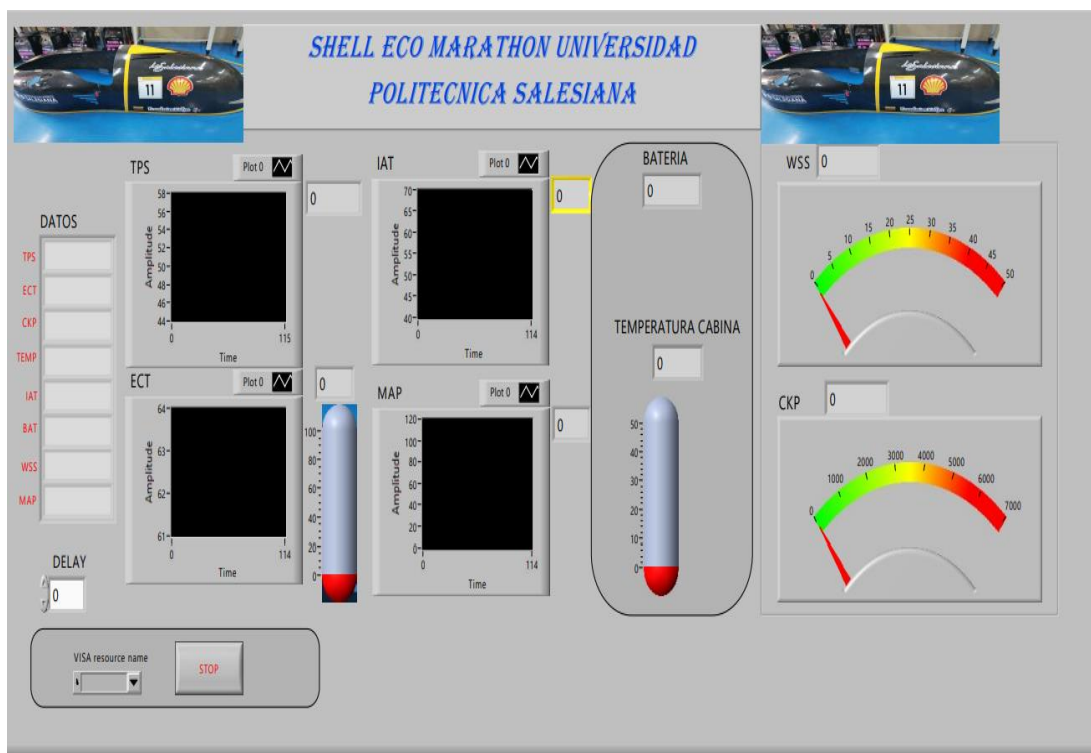
### 3.11 Diseño e Implementación Interfaz Gráfica en LabVIEW

Después de obtener los datos de los diferentes sensores del prototipo Shell Eco Marathon hacia el módulo receptor y este a su vez visualizar en el centro de monitoreo.

El desarrollo de la interfaz gráfica se realizó utilizando el software LabVIEW.

Software desarrollado para el diseño de sistemas en un lenguaje de programación grafica. Esta programación permite la visualización gráfica, además logra integrar hardware y software a su vez permitiendo diseñar interfaz personaliza para el usuario.

Figura 3.30. Interfaz gráfica LabVIEW.



Fuente: Guato Brandon, Yáñez Mario

#### 3.11.1 Desarrollo de la programación en LabVIEW

Para el desarrollo de la programación se lo realizo en el software de LabVIEW ya que este nos permite visualizar de una manera gráfica todos los componentes como indicadores, termómetros, gráficos de formas de onda, para la persona que se encuentra monitoreando estos datos se podrá entender de una manera más eficiente esta interfaz gráfica.



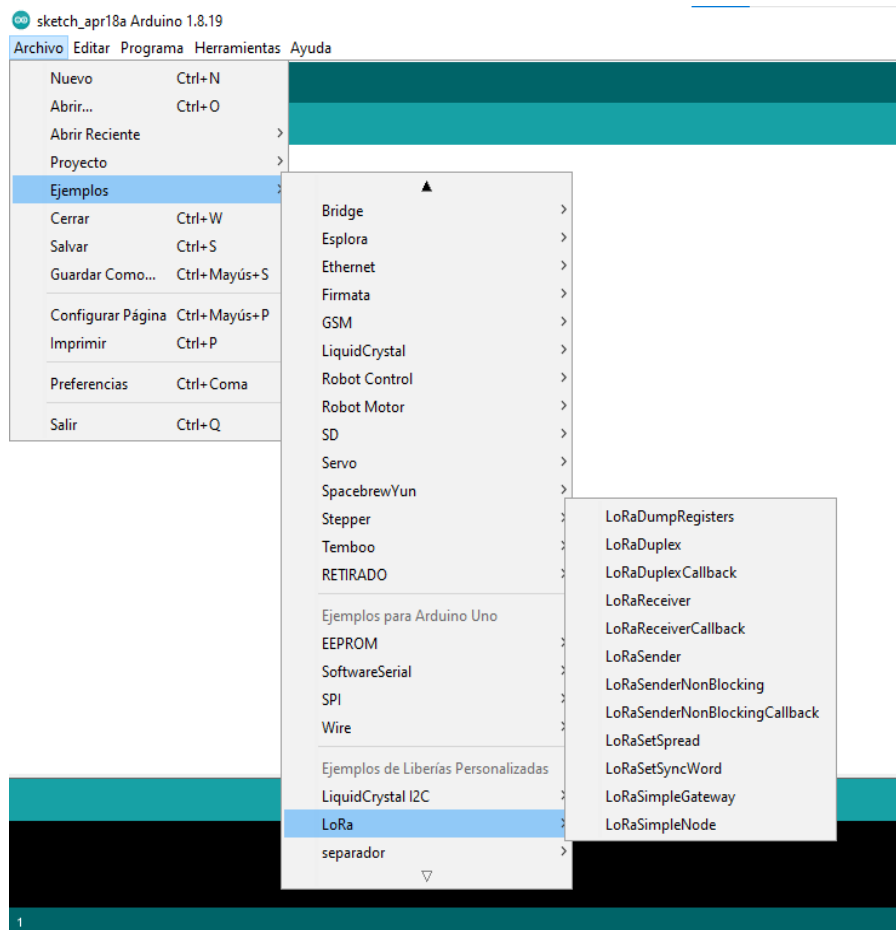
## CAPÍTULO IV

### ANÁLISIS DE RESULTADOS

#### 4.1 Pruebas de Comunicación

Para las pruebas de comunicación entre los módulos LoRa tomamos el ejemplo LoRaSender como se muestra en la Figura 4.1.

Figura 4.1. Ejemplo LoRaSender



Fuente: Guato Brandon, Yáñez Mario

LoRa utiliza tres frecuencias principalmente:

Para América 915 MHz.


Para Europa 868 MHz.

Para Asia y América del sur a 433 MHz.

Esto es muy importante ya que los dispositivos LoRa trabajan a ciertas frecuencias dependiendo el país de donde nos encontremos, Ecuador trabaja en los 433MHz esta banda esta libre disponible para radioaficionados que va desde los 430 MHz hasta los 440MHz.

A continuación, se presenta la programación del Ejemplo LoRaSender la misma que nos permite realizar envío de paquetes.

Figura 4.2. Código LoRaSender.

The image shows a screenshot of the Arduino IDE interface. At the top, the title bar reads "LoRaSender Arduino 1.8.19". Below the title bar, there are menu options: "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". A toolbar with icons for checkmark, back, grid, upload, and download is visible. The main editor area shows the following C++ code:

```
LoRaSender$
#include <SPI.h>
#include <LoRa.h>

int counter = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Sender");

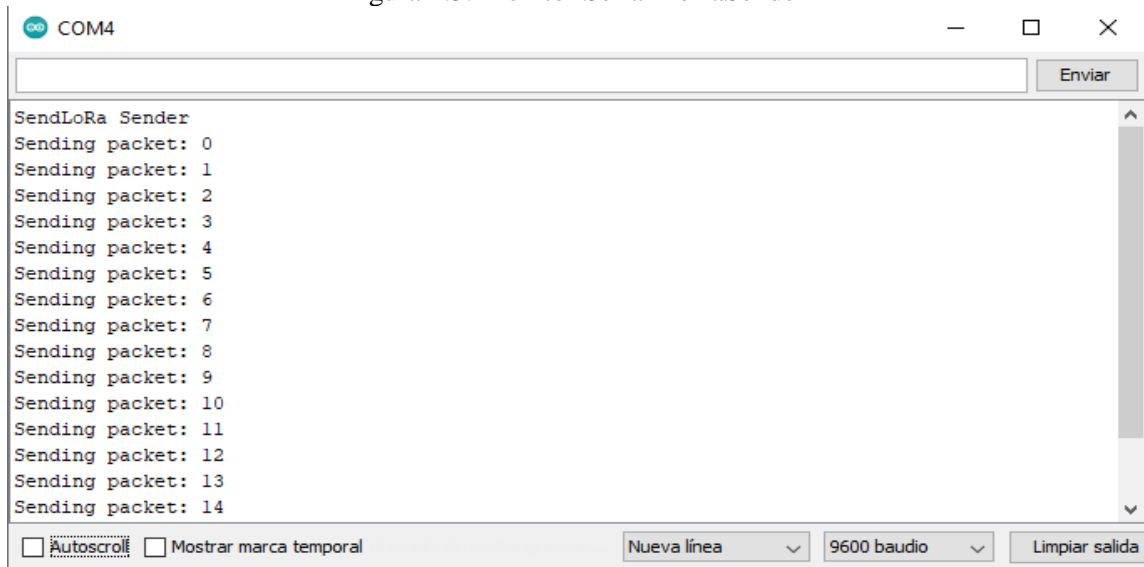
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}

void loop() {
  Serial.print("Sending packet: ");
  Serial.println(counter);
  // send packet
  LoRa.beginPacket();
  LoRa.print("hello ");
  LoRa.print(counter);
  LoRa.endPacket();
  counter++;
  delay(5000);
}
```

Fuente: Guato Brandon, Yáñez Mario

Abrimos el monitor Serial en donde observamos que se inicializa la conexión del LoRa y el contador de los paquetes enviados.

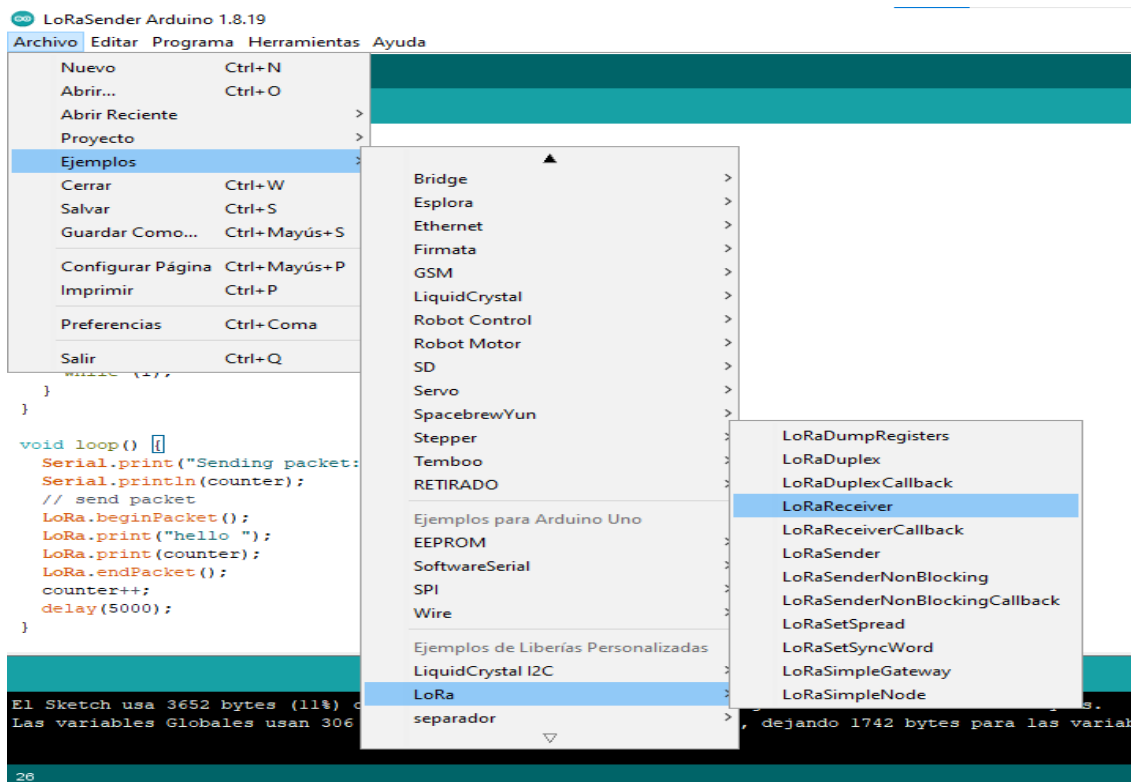
Figura 4.3. Monitor Serial LoRaSender



Fuente: Guato Brandon, Yáñez Mario

Para las pruebas de comunicación de recepción abrimos el Archivo, LoRa, Ejemplos, LoRaReceiver, como se muestra en la Figura 3.43

Figura 4.4. Ejemplo LoRaReceiver



Fuente: Guato Brandon, Yáñez Mario

A continuación, se presenta la programación del Ejemplo LoRaReceiver la misma que nos permite recibir los paquetes.

Figura 4.5. Código LoRaReceiver

The image shows a screenshot of the Arduino IDE interface. At the top, there is a title bar that says "LoRaReceiver Arduino 1.8.19". Below the title bar, there are menu options: "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". A toolbar with icons for checkmark, play, keyboard, upload, and download is visible. The main area shows a code editor with the following C++ code:

```
LoRaReceiver $
#include <SPI.h>
#include <LoRa.h>
void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver");

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}
void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet ");

    // read packet
    while (LoRa.available()) {
      Serial.print((char) LoRa.read());
    }
    // print RSSI of packet
    Serial.print("' with RSSI ");
    Serial.println(LoRa.packetRssi());
  }
}
```

Fuente: Guato Brandon, Yáñez Mario

Abrimos el monitor Serial en donde observamos que se inicializa la conexión del LoRa y el contador de los paquetes recibidos.

Figura 4.6. Monitor Serial LoRaSender

```
COM4  
  
LoRa Receiver  
LoRa Initializing OK!  
Received packet 'hello ' with RSSI -85  
Received packet 'hello ' with RSSI -85  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -85  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -87  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -86  
Received packet 'hello ' with RSSI -87  
Received packet 'hello ' with RSSI -87  
Received packet 'hello ' with RSSI -85
```

Fuente: Guato Brandon, Yáñez Mario

Mediante las pruebas de comunicación de los módulos LoRa que se aplicaron anteriormente se pudo evidenciar que los dos dispositivos tanto emisor como receptor están funcionando correctamente, ya que se pudo apreciar el envío y recepción de paquetes.

#### 4.2 Pruebas de distancia de recorrido

Para las pruebas de distancia entre los módulos LoRa tanto para el emisor como para el receptor se optó por realizar las pruebas en el parque Ecológico de Santa Ana en el sector de la Rodrigo de Chávez al sur de Quito obteniendo como resultados lo siguiente datos.

Parámetros LoRa configurados por software

Tabla 4.1. Parámetros LoRa.

	Factor de alcance (SF)	Ancho de banda (BW) kHz	Tasa de codificación de errores (CR)
Datos	12	500E3	5

Fuente: Guato Brandon, Yáñez Mario



Se usaron los parámetros mencionados, para el factor de propagación o alcance se configuro en 12 ya que acepta parámetros de configuración entre 7 a 12 esto permite una alta velocidad de bits o velocidad de datos.

Para el ancho de banda se configura en 500E3, ya que admite valores entre 7.8E3-10.4E3-15.6E3-20.8E3-31.25E3-41.7E3-62.5E3-125E3-250E3-500E3

Para la codificación de la radio se configuro en 5 ya que los valores admitidos están entre 1 y 5.

Es necesario aclarar ciertos conceptos para una correcta interpretación de los resultados los cuales serán mencionados a continuación:

RSSI en sus siglas define la intensidad de la señal recibida y es medida en dBm siempre es un valor negativo donde el RSSI mínimo es: -120 dBm y una señal que tendría un valor de -30 dBm se considera un valor eficiente (Dopazo, 2019).

A continuación, se presenta los resultados obtenidos del RSSI a varias distancias:

- **Prueba de distancia a 200 metros.**

Figura 4.7. Distancia de 200.61 metros Parque ecológico Santa Ana.



Fuente: Guato Brandon, Yáñez Mario

Figura 4.8. RSSI a 200 metros.

```
Received packet 'hello 0' with RSSI -111
Received packet 'hello 1' with RSSI -113
Received packet 'hello 0' with RSSI -109
Received packet 'hello 1' with RSSI -113
Received packet 'hello 2' with RSSI -111
Received packet 'hello 3' with RSSI -114
Received packet 'hello 4' with RSSI -114
Received packet 'hello 5' with RSSI -109
Received packet 'hello 6' with RSSI -113
Received packet 'hello 8' with RSSI -113
Received packet 'hello 9' with RSSI -110
Received packet 'hello 10' with RSSI -112
Received packet 'hello 11' with RSSI -110
Received packet 'hello 13' with RSSI -111
Received packet 'Xgl  088' with RSSI -110
Received packet 'hello 19' with RSSI -112
Received packet 'hello 20' with RSSI -111
```

Fuente: Guato Brandon, Yánez Mario

- **Prueba de distancia a 400 metros.**

Figura 4.9. Distancia de 400.55 metros Parque ecológico Santa Ana.



Fuente: Guato Brandon, Yánez Mario

Figura 4.10. RSSI a 400 metros.

```
Received packet 'hello 0' with RSSI -108
Received packet 'hello 1' with RSSI -108
Received packet 'hello 2' with RSSI -111
Received packet 'hello 3' with RSSI -107
Received packet 'hello 4' with RSSI -109
Received packet 'hello 5' with RSSI -107
Received packet 'hello 6' with RSSI -106
Received packet 'hello 7' with RSSI -108
Received packet 'hedlo 8' with RSSI -110
Received packet 'hello 9' with RSSI -109
Received packet 'hello 10' with RSSI -106
Received packet 'hello 11' with RSSI -107
Received packet 'hello 12' with RSSI -107
Received packet 'hello 13' with RSSI -107
Received packet 'hello 14' with RSSI -107
Received packet 'hello 15' with RSSI -105
Received packet 'hello 16' with RSSI -106
Received packet 'hello 17' with RSSI -105
Received packet 'hello 18' with RSSI -106
Received packet 'hello 19' with RSSI -106
Received packet 'hello 20' with RSSI -107
```

Fuente: Guato Brandon, Yánez Mario

- **Prueba de distancia a 600 metros.**

Figura 4.11. Distancia de 600.24 metros Parque ecológico Santa Ana.



Fuente: Guato Brandon, Yánez Mario

Figura 4.12. RSSI a 600 metros.

```
Received packet 'hello 0' with RSSI -113
Received packet 'hello 2' with RSSI -112
Received packet 'hello 3' with RSSI -109
Received packet 'hello 4' with RSSI -109
Received packet 'hello 5' with RSSI -109
Received packet 'hello 6' with RSSI -113
Received packet 'hello 7' with RSSI -114
Received packet 'hello 8' with RSSI -113
Received packet 'hello 9' with RSSI -113
Received packet 'hello 10' with RSSI -109
Received packet 'hello 11' with RSSI -110
Received packet 'hello 12' with RSSI -110
Received packet 'hello 13' with RSSI -109
Received packet 'hello 14' with RSSI -110
Received packet 'hello 15' with RSSI -110
Received packet 'hello 16' with RSSI -109
Received packet 'hello 17' with RSSI -110
Received packet 'hello 18' with RSSI -110
Received packet 'hello 19' with RSSI -110
Received packet 'hello 20' with RSSI -110
```

Fuente: Guato Brandon, Yáñez Mario

- **Prueba de distancia a 800 metros.**

Figura 4.13. Distancia de 800.28 metros Parque ecológico Santa Ana.



Fuente: Guato Brandon, Yáñez Mario

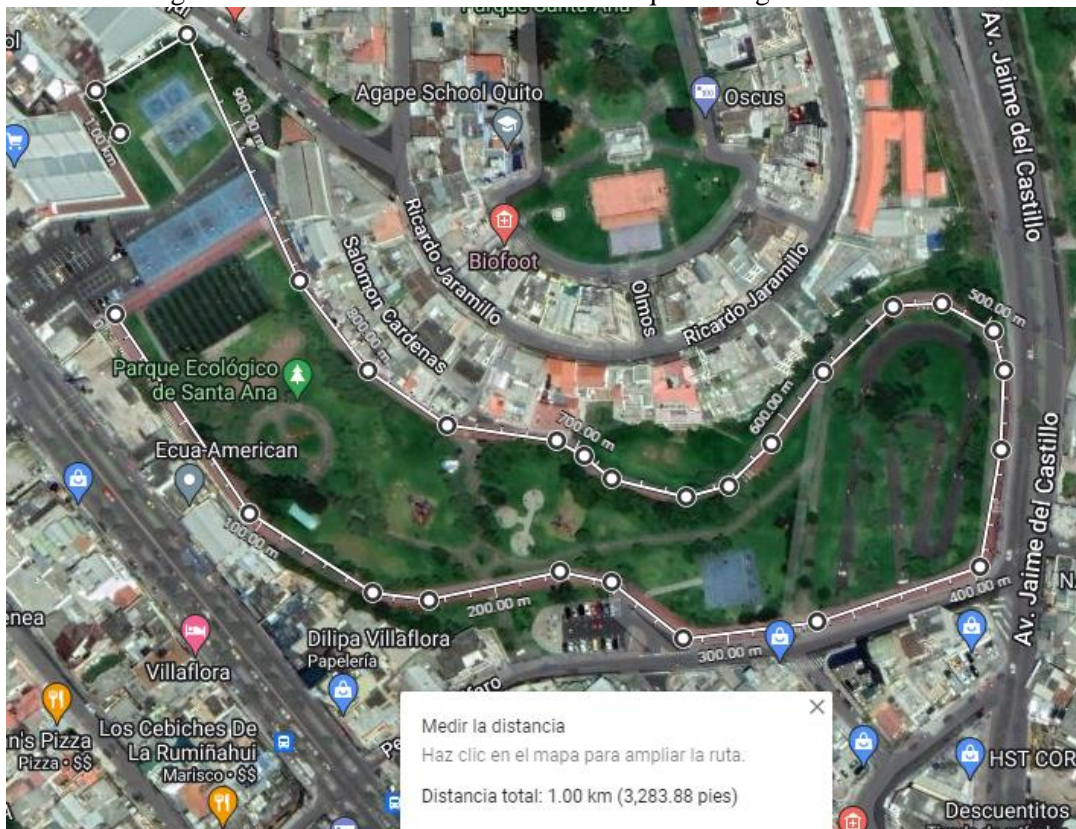
Figura 4.14. RSSI a 800 metros.

```
Received packet 'hello 0' with RSSI -113
Received packet 'hello 2' with RSSI -112
Received packet 'hello 3' with RSSI -109
Received packet 'hello 4' with RSSI -109
Received packet 'hello 5' with RSSI -109
Received packet 'hello 6' with RSSI -113
Received packet 'hello 7' with RSSI -114
Received packet 'hello 8' with RSSI -113
Received packet 'hello 9' with RSSI -113
Received packet 'hello 10' with RSSI -109
Received packet 'hello 11' with RSSI -110
Received packet 'hello 12' with RSSI -110
Received packet 'hello 13' with RSSI -109
Received packet 'hello 14' with RSSI -110
Received packet 'hello 15' with RSSI -110
Received packet 'hello 16' with RSSI -109
Received packet 'hello 17' with RSSI -110
Received packet 'hello 18' with RSSI -110
Received packet 'hello 19' with RSSI -110
Received packet 'hello 20' with RSSI -110
```

Fuente: Guato Brandon, Yáñez Mario

- **Prueba de distancia a 1000 metros.**

Figura 4.15. Distancia de 1000 metros Parque ecológico Santa Ana.



Fuente: Guato Brandon, Yáñez Mario

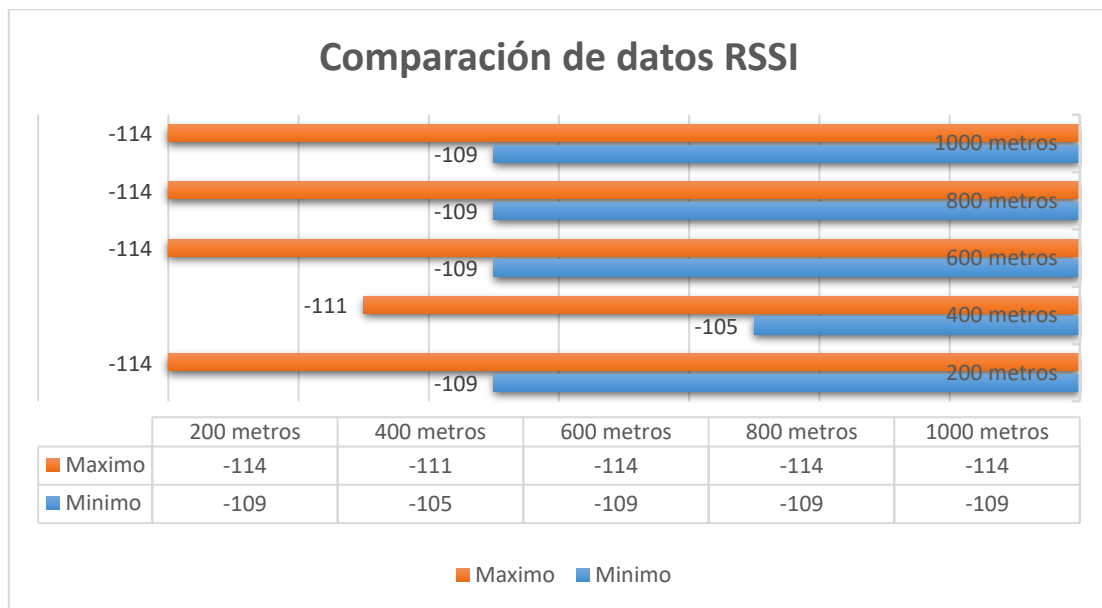
Figura 4.16. RSSI a 1000 metros.

```

Received packet 'hello 0' with RSSI -113
Received packet 'hello 2' with RSSI -112
Received packet 'hello 3' with RSSI -109
Received packet 'hello 4' with RSSI -109
Received packet 'hello 5' with RSSI -109
Received packet 'hello 6' with RSSI -113
Received packet 'hello 7' with RSSI -114
Received packet 'hello 8' with RSSI -113
Received packet 'hello 9' with RSSI -113
Received packet 'hello 10' with RSSI -109
Received packet 'hello 11' with RSSI -110
Received packet 'hello 12' with RSSI -110
Received packet 'hello 13' with RSSI -109
Received packet 'hello 14' with RSSI -110
Received packet 'hello 15' with RSSI -110
Received packet 'hello 16' with RSSI -109
Received packet 'hello 17' with RSSI -110
Received packet 'hello 18' with RSSI -110
Received packet 'hello 19' with RSSI -110
Received packet 'hello 20' with RSSI -110
    
```

Fuente: Guato Brandon, Yáñez Mario

Figura 4.17. Comparación de datos RSSI.



Fuente: Guato Brandon, Yáñez Mario

Como se puede apreciar en la Figura 4.17. se realizó una comparación de los datos mínimos y máximos del RSSI llegando a la conclusión que el envío de paquetes se mantiene entre -105 como mínimo a -114 como máximo, no se apreció una gran variación del RSSI a pesar de que se fue aumentando la distancia entre los módulos LoRa.

### 4.3 Pruebas De Funcionamiento en el prototipo Shell

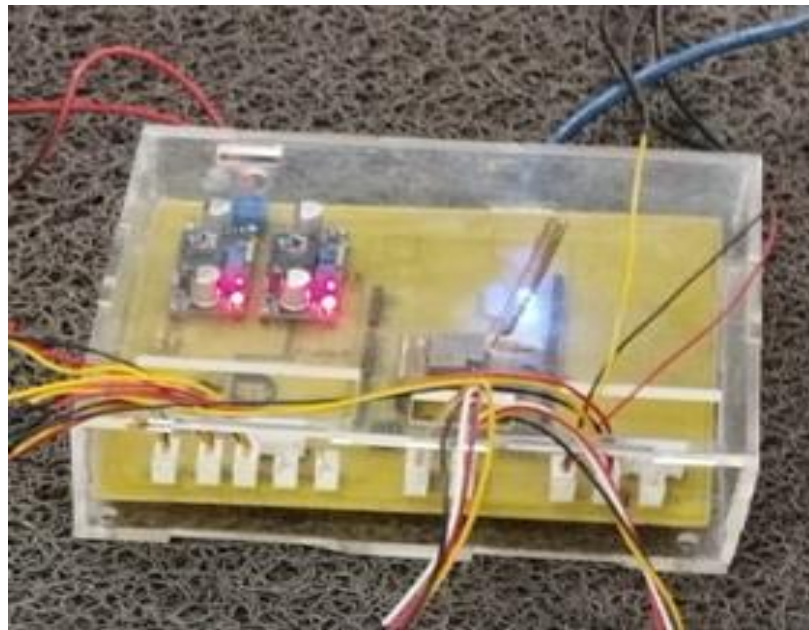
Para las respectivas pruebas de funcionamiento se las realizo con el prototipo Shell en modo estacionario, ya que se encontraba en remodelación en su estructura.

#### 4.3.1 Energización de los módulos

- **Módulo Emisor**

Durante la energización del módulo emisor se puede observar el encendido de los diferentes sensores y componentes electrónicos que están conectados a través de él, así como el Arduino Mega, el módulo LoRa, las pantallas LCDs de 16x2 y de 20x4 y los sensores WSS, LM35 y sensor MAP.

Figura 4.18. Módulo Emisor energizado

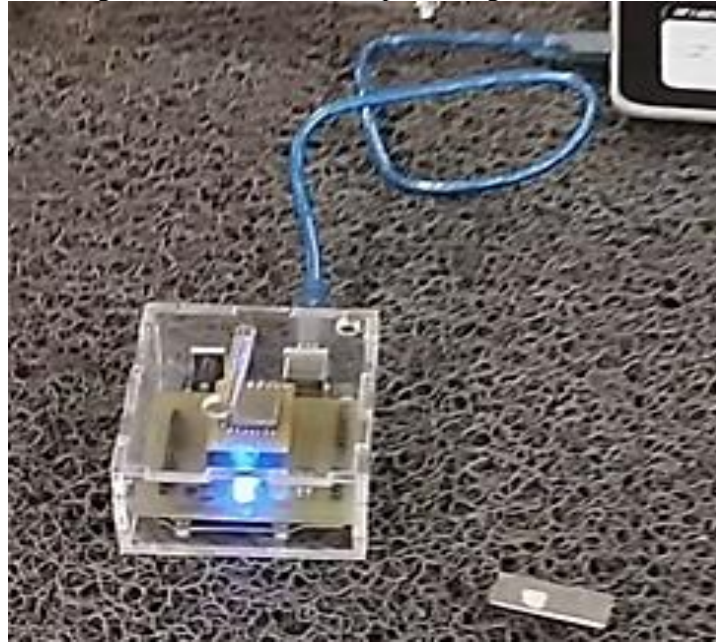


Fuente: Guato Brandon, Yáñez Mario

- **Módulo Receptor**

Durante la energización del módulo receptor se puede observar el encendido del Arduino Uno, el módulo LoRa y dos leds que están en la placa, los cuales nos permite conocer en azul una comunicación correcta y en rojo que presenta algún error.

Figura 4.19. Módulo Receptor energizado



Fuente: Guato Brandon, Yáñez Mario

Figura 4.20. Sistema de telemetría funcionando.



Fuente: Guato Brandon, Yáñez Mario



### 4.3.2 Funcionamiento de los diferentes sensores

Se realizó distintas pruebas de funcionamiento en los laboratorios de la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana en el campus Sur, en los cuales se observa el comportamiento de los diferentes sensores.

Tabla 4.2. Resultados obtenidos

Sensor	Prueba 1	Prueba 2	Prueba 3
CKP (RPM)	3700	4300	6400
ECT (°C)	20	38	56
MAP (kPa)	52	58	60
WSS (Km/h)			
TPS (%)	0	2	95
IAT (°C)	12	12	13
LM35 (°C)	12	16	19
BATERIA (V)	13.5	13	12.82

Fuente: Guato Brandon, Yáñez Mario

### Conexión del emisor hacia el prototipo Shell

Para obtener las señales de los sensores CKP, IAT, ECT y TPS, que están en el prototipo Eco Shell Marathon fue necesario realizar un empalme entre los pines de conexión de la Haltech y el Arduino Mega, revisando los pines de conexión de la Haltech con su debido diagrama y los pines del Arduino donde van a ingresar las mencionadas señales. (ANEXO XIV)

Para los sensores restantes que disponemos en el prototipo Eco Shell Marathon es decir MAP, LM35, WSS y sensor de batería la conexión fue directa hacia el Arduino puesto que estos sensores están siendo agregados, para una mejora en el prototipo.

## CAPÍTULO V

### GESTIÓN DEL PROYECTO

#### 5.1 Recursos Y Materiales

En este capítulo se analizará la factibilidad de costo-beneficio para el presente proyecto, a continuación, se detalla el costo de producción, talento humano y recursos materiales.

##### 5.1.1 Costo De Producción

Tabla 5.1. Costo de producción.

#	DETALLE	COSTO C/U	COSTO FINAL
1	Arduino Mega	\$25.00	\$25.00
1	Arduino Uno	\$15.00	\$15.00
2	Módulos Lora	\$20.00	\$40.00
4	Leds	\$0.5	\$0.20
12	Molex Macho	\$0.50	\$6.00
2	Lm2596	\$3.00	\$6.00
1	Bornera	\$0.50	\$0.50
7	Resistencias	\$0.10	\$0.70
1	Lcd 16x2	\$4.00	\$4.00
1	Lcd 20x4	\$10.00	\$10.00
2	Módulos I2C	\$2.50	\$5.00
1	Sensor Map	\$22.00	\$22.00
1	Racor Acople	\$3.00	\$3.00
2	Mangueras Aire	\$6.00	\$6.00
12	Molex Hembra	\$0.80	\$9.60
2	Cajas Protectoras	\$20.00	\$40.00
2	Protoboard	\$4.00	\$8.00
1	Cables Jumper	\$4.00	\$4.00
2	Case Acrílico LCDS	\$5.00	\$10.00
10	Cable Flexible 10metros	\$0.50	\$5.00
1	Sensor Efecto Hall	\$3.00	\$3.00
<b>Total</b>			<b>\$223</b>

Fuente: Guato Brandon, Yáñez Mario

##### 5.1.2 Recursos Materiales

Tabla 5.2. Recursos Materiales

EQUIPOS	CANTIDAD
Computador	2
Cautín	1
Destornilladores	2
Osciloscopio	1
Multímetro	2
Cortadora de cables	2
Pinzas	2

Fuente: Guato Brandon, Yáñez Mario

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1.1 CONCLUSIONES

- A través del diseño e implementación del sistema de telemetría en el prototipo Shell-Eco Marathon de la Universidad Politécnica Salesiana Sede Quito, se logró obtener datos y parámetros de los sensores como temperatura, presión, apertura de aceleración, velocidad entre otros; mismos que sirven para la evaluación del comportamiento y rendimiento del motor que antes no se tenían en el prototipo antes mencionado.
- En la implementación tecnológica se logró adaptar los módulos de comunicación, mismos que permiten el procesamiento, recepción y emisión de la información relacionada al funcionamiento de los diferentes sensores analizados; con el fin de mejorar la estrategia en la competencia Shell respecto al monitoreo de los datos en tiempo real, eficiencia en el uso de combustible y más información relevante para la toma de decisiones según los parámetros y exigencias permitidas.
- Al realizar una comparación entre el Costo de Producción del presente proyecto con el sistema de telemetría denominado ATLAS; mismo que es utilizado en la Formula 1; y que en la actualidad es considerado uno de los más costosos del mercado; se logró determinar que resulta conveniente a nivel de costos implementar nuestro diseño ya que cumple con las funciones más relevantes y brinda en proporción los mismos beneficios pero a un menor costo; haciendo que este sistema sea llamativo para los usuarios, debido a que se lo puede implementar y adaptar a varios sectores tales como: agrícola, minero, automotriz, petrolero , industrial, IoT.
- Mediante la respectiva programación en LabVIEW; se diseñó una interfaz gráfica amigable con el usuario, además se logró automatizar el tiempo de respuesta, mediante un Delay, el cual permite aumentar o disminuir, lo cual influye directamente sobre el envío y recepción de los diferentes datos en tiempo real. Obteniendo como mejor tiempo de respuesta entre 280 a 300 ms(milisegundos), por lo cual se pudo apreciar un leve retardo en el envío de datos.

### **6.1.2 RECOMENDACIONES**

- Para tomar las diferentes señales de los sensores debe compartir una tierra común en todo el sistema, tanto en el prototipo Shell, emisor y receptor, para que no existe señales con información errónea.
- Ubicar en una base fija el sistema emisor para evitar que las vibraciones que se generan en el prototipo Shell en funcionamiento, las cuales pueden afectar la información o hasta la desconexión del mismo sistema.
- Realizar una correcta calibración basándose en el software de la Haltech ya que permite visualizar los diferentes sensores en tiempo real, de esta manera podemos apreciar una información lo más exacta posible con el sistema emisor.

## BIBLIOGRAFÍA

- Alcaraz, M. (2014). Internet de las cosas. *Universidad Católica Nuestra Señora de la Asunción*.
- Alessandro, N. (2004). *Telemetria e conceitos relacionados*. Sao José dos Campos: Creative Commons Atribuição.
- Alliance, L. (Noviembre de 2015). *Lora Alliance*. Obtenido de Lora Alliance: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>
- Amazon. (2021). *Amazon*. Obtenido de Amazon: [https://www.amazon.com.mx/Circuitos-Integrados-transmisi%C3%B3n-propagaci%C3%B3n-inal%C3%A1brica/dp/B096PC1N1W/ref=sr\\_1\\_2?\\_\\_mk\\_es\\_MX=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=33ZS4DGK4X7P7&keywords=M%C3%B3dulo+Ra-01+LoRa+SX1278&qid=1641520039&srefix=m%C3%B3](https://www.amazon.com.mx/Circuitos-Integrados-transmisi%C3%B3n-propagaci%C3%B3n-inal%C3%A1brica/dp/B096PC1N1W/ref=sr_1_2?__mk_es_MX=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=33ZS4DGK4X7P7&keywords=M%C3%B3dulo+Ra-01+LoRa+SX1278&qid=1641520039&srefix=m%C3%B3)
- Aranda, D. (2014). *Electrónica: Plataformas Arduino y Raspberry Pi*. Buenos Aires: Fox Andina.
- arduino.cc. (2022). *arduino.cc*. Obtenido de arduino.cc: <https://www.arduino.cc/>
- Arroyo, D. A. (2020). *Sistema de telemetría aplicable en los prototipos de categoría gasolina participantes en la competencia shell-Eco-marathon*. Riobamba-Ecuador: Escuela Superior Politécnica de Chimborazo.
- Bartoletti, P., Paiotti, R., & Leca, R. (2019). *Proyectos con ESP32 y LoRa*. Sao Paulo : Editora NCB.
- Bonnick, A. W. (2001). *Automotive Computer Controlled Systems: Diagnostic Tools and Techniques*. Routledge.
- Candia, A., & Varela, L. N. (2017). *WSN para servicios públicos metropolitanos*. Argentina: Universidad de la PLata.
- Carpentiers, N. (11 de Diciembre de 2016). *F1 telemetria. la caerra de datos F1i*. Obtenido de <https://f1i.com/magazine/73067-f1-telemetry-data-race.html>
- Cornejo , A., & Tintin, J. (2010). *Diseño, construcción e implementación de un sistema de telemetría utilizando tecnología GSM; para el monitoreo de los parámetros de temperatura, presion de aceite, velocidad de giro del motor y la velocidad de desplazamiento de un vehículo*. Cuenca: Universidad Politecnica Salesiana.
- Denton, T. (27 de Enero de 2011). *Automotive Tecnology* . Obtenido de <http://automotive-technology.co.uk/?p=901>
- Díaz, A. (2017). *Sistema de telemetría para la Fórmula SAE*. Madrid: Universidad Politécnica de Madrid.
- Dopazo, G. L. (2019). *Test Y despliegue con tecnologia de comunicacion LoRa para aplicaciones de internet of things[Tesis de Grado,Universidad Politecnica de Madrid]*. Madrid: Repositorio Institucional.

- Morocho Arroyo, D. A. (2020). *Sistema de telemetría aplicable en los prototipos de categoría gasolina participantes en la competencia Eco-Shell Marathon*. Riobamba: Escuela Superior Politécnica de Chimborazo.
- Murillo, J. (2013). *Fundamentos de Radiación y Radiocomunicación*. Sevilla: Universidad de Sevilla .
- OnubaElectronica. (2021). *Radiofrecuencia*. Obtenido de OnubaElectronica: <https://www.onubaelectronica.es/la-radiofrecuencia/>
- Orenda. (1 de Julio de 2016). *Introduccion a Simplex, Half duplex y Full Duplex*. Obtenido de <https://medium.com/@fiberstoreorenda/introduction-to-simplex-half-duplex-and-full-duplex-fbda8d591e3a>
- Osio, J., Antonini, L., Aróztegui, W., & Rapallini, J. (2011). *Descripcion general de un microcontrolador (Módulos de comunicación)*. Argentina: CeTAD – Facultad de Ingeniería - UNLP.
- Rapallini, J. (2011). *Descripción General de un microcontrolador (Módulos de comunicación)*. Argentina: Universidad Nacional de la Plata.
- Reinoso , H., Salazar , A., & Enríquez , M. (2012). *Implementación de un modulo de control electrónico programable haltech en un motor corsa 1.3L cuatro tiempos a gasolina con inyección electrónica para el análisis de sus rendimientos*. Quito: Universidad Internacional del Ecuador.
- Rhetassi, H. B. (2019). *Desarrollo de una aplicación web de telemetría para el control de .* Cartagena: Universidad Politécnica de Cartagena.
- Satían&Cantuña. (2019). *Análisis de rendimiento de la tecnología LoRaWan aplicado al desarrollo de un sistema de monitoreo de calidad de aire en la unidad educativa santo tomás apóstol riobamba*. Riobamba: Escuela Superior Politécnica de Chimborazo.
- SHELL. (01 de enero de 2022). *Shell Eco-Marathon*. Obtenido de Shell Eco-Marathon: <https://www.shell.com.br/energia-e-inovacao/ecomarathon/shell-eco-marathon-brasil.html#>
- SITRACK. (20 de julio de 1999). *Todo sobre la Telemetría*. Obtenido de <http://landing.sitrack.com/telemetr%C3%ADa-y-sus-aplicaciones>
- Tapia, V. (2013). *Diseño e implementación de un sistema de transferencia de datos por telemetría para un vehículo de competencia con el uso de tecnología inalámbrica bluetooth (Trabajo de titulación).(Pre-grado)*. Riobamba: Escuela Politécnica de Chimborazo.

## ANEXOS

### ANEXO I. Especificaciones técnicas del Arduino Mega y Arduino UNO.

DETALLE	CARACTERÍSTICAS
Voltaje operativo	5V
Voltaje de entrada	7-12V
Voltaje de entrada límites	6-20V
Microcontrolador	Atmel ATmega2560 a 16 MHz
Memoria flash	256 KB
Intensidad de corriente por cada pin de E/S	40mA DC

### Especificaciones técnicas del Arduino UNO.

DETALLE	CARACTERÍSTICAS
Microcontrolador	ATmega328
Voltaje Operativo	5V
Voltaje de entrada	7 - 12V
Memoria flash	32 kB
Velocidad del reloj	16 MHz

### ANEXO II. Especificaciones técnicas del módulo LoRa Ra-01 SX1278.

DETALLE	CARACTERÍSTICAS
Voltaje de entrada	1.8V - 3.6V
Estándar inalámbrico	433 MHz
Rango de frecuencia	420 - 450MHz
Interfaz	SPI Half-duplex
Modulación	FSK, GFSK, MSK, GMSK, Lora
Velocidad de transmisión de datos	300 kbps
Sensibilidad	-136dbm
Potencia de salida	+20dBm -100mW
Distancia de alcance	10km (varía de acuerdo con la topografía y obstáculos que afecten la señal)
Temperatura de trabajo	-40°C - 85°C

### ANEXO III. Especificaciones técnicas del módulo regulador DC-DC LM2596.

DETALLE	CARACTERÍSTICAS
Corriente de salida máxima	3Amp
Voltaje de entrada	4V a 35V
Voltaje de salida	1.23V a 30V
Protección de cortocircuito	Hasta 5A
Potencia de salida	25W

### ANEXO IV. Especificaciones técnicas del Sensor LM35.

DETALLE	CARACTERÍSTICAS
Resolución	10mV por cada grado centígrado.
Voltaje de alimentación	4VDC hasta 20VDC
Tipo de medición	Salida analógica
Numero de pines	3 pines, GND, VCC y VSalida
Precisión	±1/4°C
Consumo de corriente	60 µA

**ANEXO V. Especificaciones técnicas de la Pantalla LCD 16X2 y LCD 20X4**

<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Formato de presentación	16 caracteres por 2 líneas
Interfaz de entrada	4-Bits / 8-Bits
Retroiluminación	LED azul
Controlador	HD44780
Voltaje de funcionamiento	4.5 – 5.5V
Consumo de corriente del LCD	aproximadamente 2 mA

**Especificaciones técnicas de la Pantalla LCD 20x4.**

<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Formato de presentación	20 caracteres por 4 líneas
Interfaz de entrada	4-Bits / 8-Bits
Retroiluminación	LED azul
Controlador	HD44780
Voltaje de funcionamiento	4.5 – 5.5V
Consumo de corriente del LCD	aproximadamente 2 mA

**ANEXO VI. Especificaciones técnicas de Módulo adaptador serial I2C.**

<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Referencia	Módulo I2C
Basado en el expansor	PCF8574
Interfaz	IIC/I2C
Pines	GND, VCC, SDA, SCL
Voltaje de funcionamiento	5VDC
Compatible con: pantallas LCD	16x2 o 20x4

**ANEXO VII. Especificaciones técnicas del sensor ECT.**

<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Voltaje de operación	0.5V a 4.9V DC
Voltaje máximo	+5V
Temperatura mínima	-40 ° C
Temperatura máxima	120 ° C

**ANEXO VIII. Especificaciones técnicas del sensor MAP Haltech.**

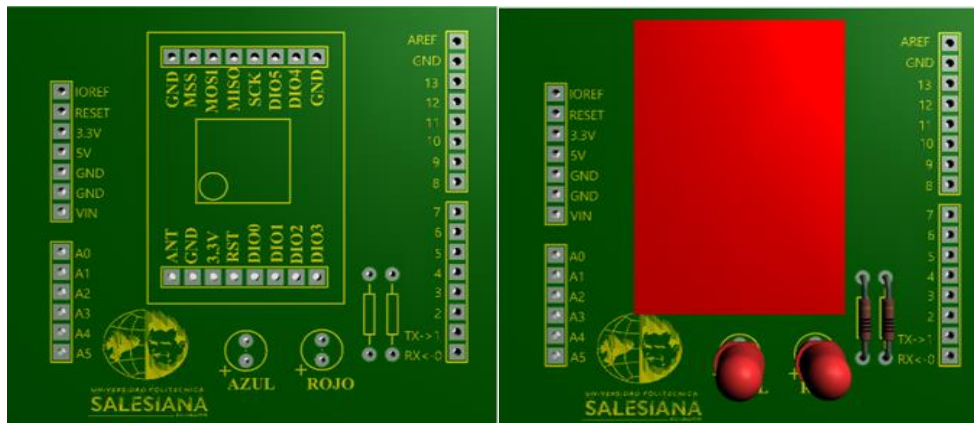
<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Voltaje de operación	0.5V a 4.9V DC
Voltaje máximo	+7V
Presión mínima	15 KPa
Presión máxima	200 KPa

**ANEXO IX. Señal del sensor WSS efecto Hall.**

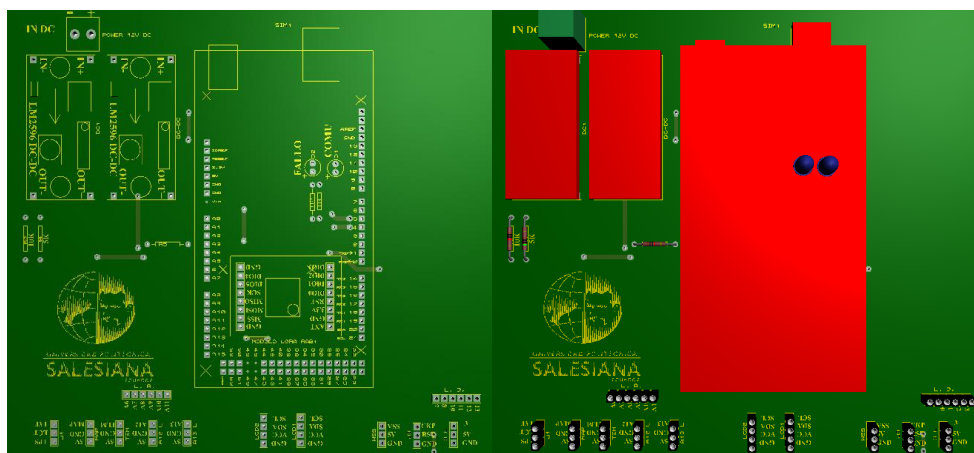
<b>DETALLE</b>	<b>CARACTERISTICAS</b>
Voltaje de operación	3.5V a 24V DC
Voltaje máximo	24V DC
Rango de temperatura de funcionamiento	-40 ° C a 125 ° C



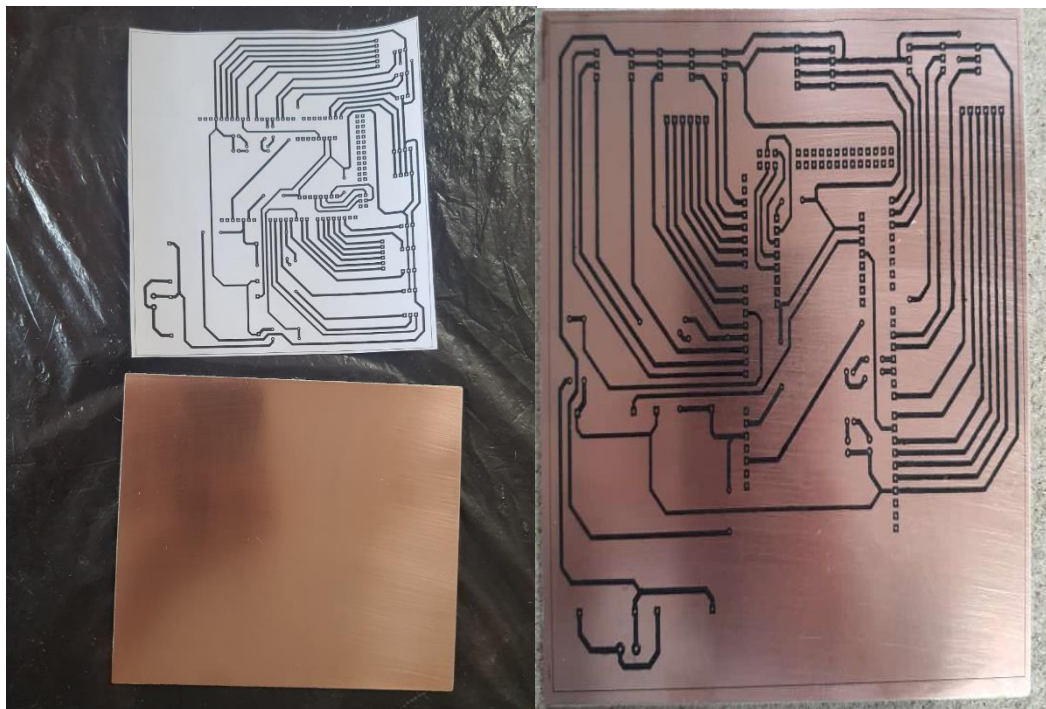
**ANEXO X.** Diseño de la placa PCB del receptor en 3D



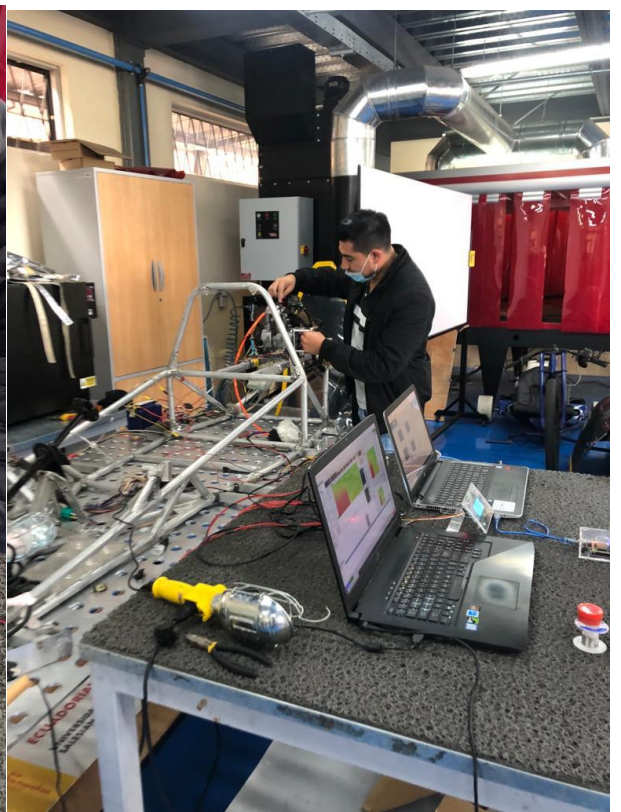
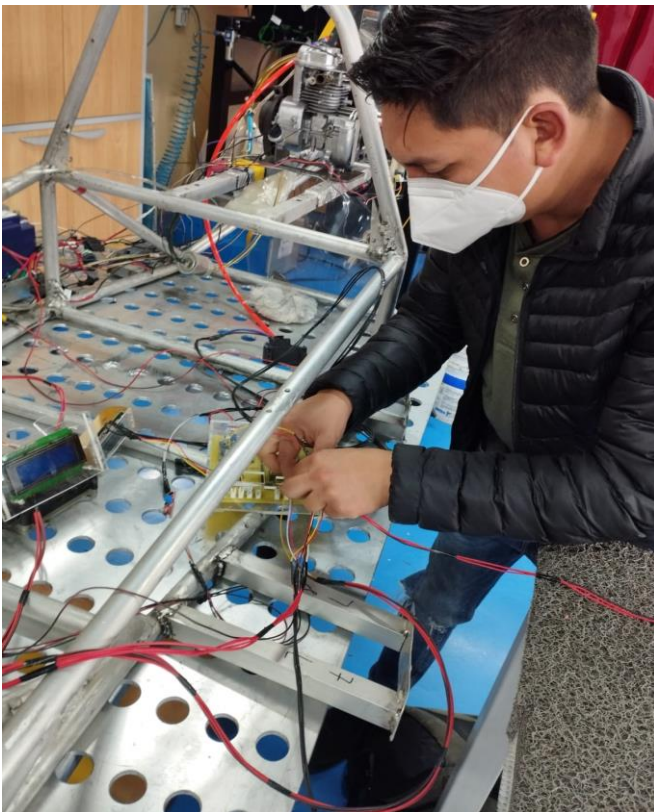
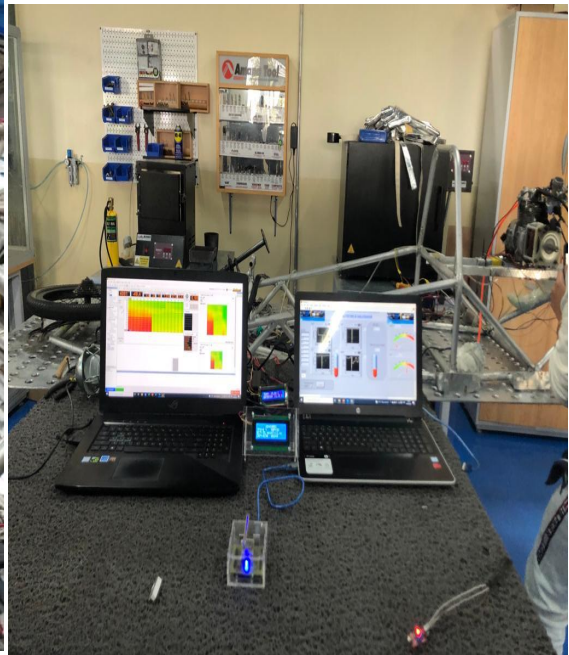
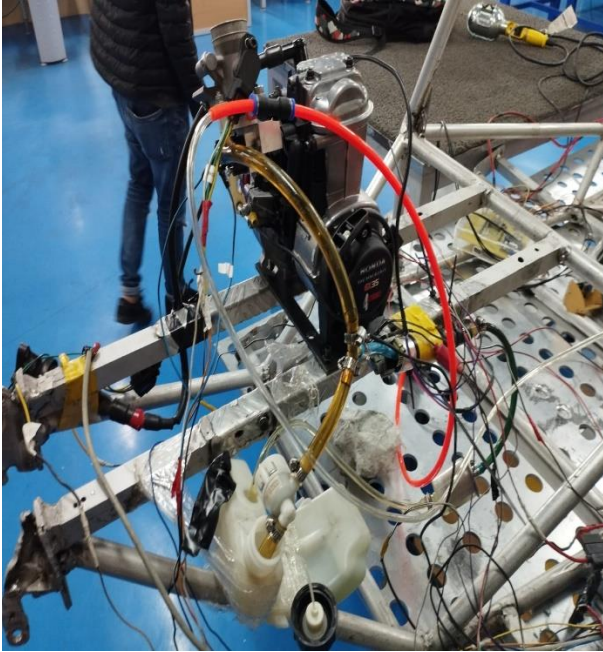
**ANEXO XI.** Diseño de la placa PCB del emisor en 3D



**ANEXO XII** Elaboración de la placa.



**ANEXO XIII. Conexión y calibración de equipos.**





ANEXO XIV. Diagrama de conexión del Ecu Haltech

