UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS DE UBICACIÓN Y CONSUMO DE ENERGÍA APLICADO A UNA MOTO ELÉCTRICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica

AUTORES: ELMER PATRICIO BERNAL ESPINOZA
ARTURO TEÓFILO GRANDA PIZHA
TUTOR: ING. JOHNNY XAVIER SERRANO GUERRERO, PHD.

Cuenca - Ecuador
2023

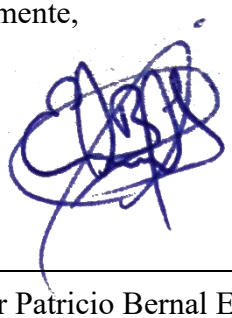
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

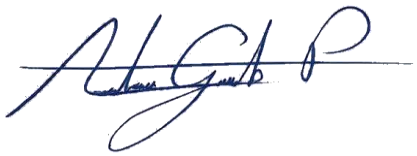
Nosotros, Elmer Patricio Bernal Espinoza con documento de identificación N° 0928642297 y Arturo Teófilo Granda Pizha con documento de identificación N° 0302902150; manifestamos que:

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.

Cuenca, 28 de marzo del 2023

Atentamente,





Elmer Patricio Bernal Espinoza
0928642297

Arturo Teófilo Granda Pizha
0302902150

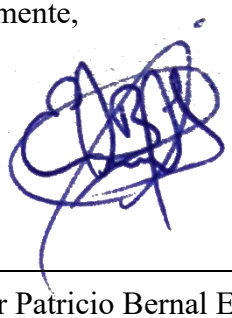
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

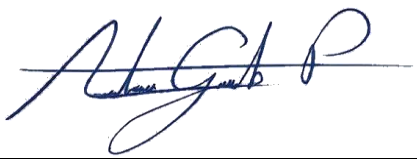
Nosotros, Elmer Patricio Bernal Espinoza con documento de identificación N° 0928642297 y Arturo Teófilo Granda Pizha con documento de identificación N° 0302902150, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: "Implementación de un sistema de adquisición de datos de ubicación y consumo de energía aplicado a una moto eléctrica de la Universidad Politécnica Salesiana", el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, 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 la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 28 de marzo del 2023

Atentamente,





Elmer Patricio Bernal Espinoza
0928642297

Arturo Teófilo Granda Pizha
0302902150

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Johnny Xavier Serrano Guerrero con documento de identificación N° 0104983382, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS DE UBICACIÓN Y CONSUMO DE ENERGÍA APLICADO A UNA MOTO ELÉCTRICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, realizado por Elmer Patricio Bernal Espinoza con documento de identificación N° 0928642297 y por Arturo Teófilo Granda Pizha con documento de identificación N° 0302902150, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 28 de marzo del 2023

Atentamente,

Ing. Johnny Xavier Serrano Guerrero, PhD.

0104983382

● AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a Dios y a mis padres por brindarme la fuerza, motivación, confianza y apoyo necesarios para continuar con mis estudios a pesar de los errores cometidos. Asimismo, agradezco de manera especial a los profesores de la Universidad, quienes han sido clave en mi formación académica y preparación para el mundo laboral, y aquellas amistades que han confiado en mí desde un inicio.

Elmer Patricio Bernal Espinoza

Primeramente, agradezco al todopoderoso por darme la vida y la salud, por otorgarme una familia maravillosa quienes me han apoyado en todo momento. Además, quiero reconocer principalmente a mis padres quienes siempre han soñado que sea un profesional y una persona digna para la sociedad, que han luchado y han sacrificado por mí, a mis hermanos que son ejemplo de vida, y a todas las personas cercanas las cuales tuvieron paciencia y confiaron hasta el último momento dándome consejos y alientos de ánimos. Finalmente, agradezco por el apoyo que me brindó el programa de becas Senescyt hasta culminar mi carrera, y a los todos docentes quienes he conocido durante mi carrera, agradezco de corazón por inculcarme y brindarme su juicio

Arturo Teófilo Granda Pizha

● DEDICATORIAS

Este trabajo de titulación está dedicado de manera especial a mis padres, quienes han sido mi principal apoyo en el camino hacia mi formación académica y personal. Su confianza en mí ha sido una motivación constante para seguir adelante en todos los ámbitos de mi vida, y gracias a su guía y ejemplo he aprendido los valores que me permitirán ser una persona trabajadora y honrada. Les agradezco profundamente por su incondicional apoyo y amor.

Elmer Patricio Bernal Espinoza

Este presente trabajo de titulación dedico a mis queridos padres, sin ellos ni existiera en esta vida, que gracias a vosotros soy quien debo ser y a las personas que me apoyaron siempre os recordaré. Por ello aludió estas letrillas, Dios solo guíame mi vida, acércate a mí con todos los deseos divinos para mi familia y mi futuro, cuídame y protégeme de todo mal, dilos que solo vivo para amarlos, antepasados los honro y trataré de vivir con toda la dignidad que me enseñaron.

Arturo Teófilo Granda Pizha

● ÍNDICE GENERAL

● ÍNDICE DE FIGURAS

● ÍNDICE DE TABLAS

● RESUMEN

La implementación de un sistema para adquirir datos del consumo de energía de una batería de motoneta utilizando un sensor para medir la corriente y voltaje es factible con la tecnología actual, y puede ser una herramienta útil para optimizar el rendimiento y la eficiencia energética de la moto.

Además, si se realiza una integración de la potencia en un periodo de tiempo determinado, se puede calcular la energía total consumida en ese periodo. Es importante tener en cuenta que el consumo de energía de una motoneta puede variar en función de diferentes factores, como la velocidad, la carga que transporta, la inclinación del terreno, la temperatura ambiente, entre otros. Por lo tanto, para obtener una estimación precisa del consumo de energía en una situación específica, es necesario tomar en cuenta estos factores y realizar mediciones en diferentes condiciones. La integración de un módulo GPS en el circuito permite conocer la ubicación de la moto eléctrica en tiempo real. Además, al conectar el circuito a Internet mediante un módulo WiFi, es posible enviar la información de consumo de energía y ubicación a una nube en línea para su registro y visualización. De esta manera, se puede monitorear y analizar el consumo de energía y la ubicación de la moto eléctrica en tiempo real, lo que puede ser de gran utilidad para mejorar la eficiencia energética de la moto y para tomar decisiones informadas sobre su uso y mantenimiento.

Al enviar los datos de potencia y ubicación a través de un módulo WiFi a la nube Adafruit, se puede visualizar la información de la batería de la moto desde cualquier dispositivo con acceso a la cuenta de la nube. La extracción de la información de la plataforma de Adafruit y su almacenamiento en una hoja de Excel puede ser una herramienta valiosa para analizar y comparar los datos en diferentes momentos, lo que puede ayudar a identificar patrones y tendencias en el consumo de energía y la ubicación de la moto. De esta manera, se puede tomar decisiones informadas para mejorar la eficiencia energética de la moto y optimizar su uso.

Palabras claves: *WiFi, Adafruit, comunicación serial, potencia, consumo.*

- **ABSTRACT**

The implementation of a system to acquire data on the energy consumption of an electric motorcycle battery using a sensor to measure current and voltage is feasible with current technology, and can be a useful tool to optimize the performance and energy efficiency of the battery. motorcycle. By obtaining the energy consumption information of the electric motorcycle using a sensor to estimate current and voltage, it is possible to know the power that the motorcycle consumes in different circumstances, such as at different speeds or with different weights on it. Likewise, by recording energy consumption at different times, it is possible to know how much energy the motorcycle consumes in a given period. The integration of a GPS module in the circuit allows knowing the location of the electric motorcycle in real time. In addition, by connecting the circuit to the Internet through a WiFi module, it is possible to send the information of energy consumption and location to an online cloud for recording and viewing. In this way, the energy consumption and location of the electric motorcycle can be monitored and analyzed in real time, which can be very useful to improve the energy efficiency of the motorcycle and to make informed decisions about its use and maintenance.

By sending power and location data via a WiFi module to the Adafruit cloud, motorcycle battery information can be viewed from any device with access to the cloud account. Extracting the information from the Adafruit platform and storing it in an Excel sheet can be a valuable tool for analyzing and comparing data at different times, which can help identify patterns and trends in energy consumption and location of the motorcycle In this way, informed decisions can be made to improve the energy efficiency of the motorcycle and optimize its use.

Keywords: *WiFi, Adafruit, serial communication, power, consumption.*

INTRODUCCIÓN

El presente trabajo de titulación pretende implementar un sistema de adquisición de datos en una moto eléctrica de la Universidad Politécnica Salesiana sede Cuenca, el cual permite leer parámetros de voltajes, corrientes, potencia, consumo de energía y ubicación en tiempo real, los mismos que son monitoreados desde una computadora a través de una web Adafruit. Se plantea un diseño para monitorear datos de cada sensor, mediante la programación en el Arduino, lo cual permite la integración y control de estos datos en una plataforma en línea como Adafruit, lo que permite garantizar una medición precisa y confiable de estos parámetros.

El primer capítulo debe establecer una base sólida de conocimientos teóricos y describir detalladamente los componentes (cables, conectores, terminales aislantes, sensor de corriente y voltaje, Arduino, modulo GPS ublox Neo-6m-v2, módulo ESP32) utilizados en el proyecto para garantizar que se pueda lograr una medición precisa y confiable de los parámetros eléctricos y la ubicación de la motocicleta eléctrica.

Dentro del capítulo dos, es necesario enfocar en las características de la plataforma web, como el uso, el funcionamiento, la compatibilidad de dispositivos que se utilizará para visualizar los datos, con la finalidad de describir la implementación del sistema para cumplir a cabalidad los objetivos planteados en este proyecto.

En el tercer capítulo se realizan mediciones de variables importantes como; la potencia, el consumo de energía la batería de la motocicleta y se verifica la ubicación o localización de la misma en un periodo dado. Además, se realiza análisis de los resultados obtenidos para evaluar la efectividad del sistema de adquisición de datos implementado y determinar si se ha cumplido los objetivos del proyecto.

● ANTECEDENTES DEL PROBLEMA DE ESTUDIO

La operación de una máquina de corriente continua al acoplarse a un sistema eléctrico está condicionada por su modo de funcionamiento, ya sea como motor o generador, y esto se debe a la dirección del flujo de energía. La eficiencia de los vehículos eléctricos es aproximadamente 3 veces más que los de combustión ya que no consume energía cuando no se encuentra en movimiento o cuando no ejerce una fuerza, a diferencia de los motores actuales, que continúan funcionando incluso estando en ralentí. Los vehículos eléctricos presentan tiempos de recarga prolongados en comparación con el proceso de llenado de combustible de un tanque, lo que hace evidente la necesidad de mejorar continuamente su autonomía. Una motocicleta eléctrica es un vehículo que se impulsa mediante un motor eléctrico ubicado en la rueda delantera o trasera, y utiliza una batería para almacenar la energía necesaria para su funcionamiento. Su principal rasgo distintivo radica en su diseño sencillo y fácil uso para el usuario. [3].

El desarrollo tecnológico en la monitorización de maquinarias y vehículos es de gran importancia a nivel nacional debido a que permite una gestión más eficiente de los recursos y una optimización de los procesos productivos. Al contar con sistemas de monitoreo en tiempo real, se pueden obtener datos precisos sobre el estado de las maquinarias y vehículos en diferentes ámbitos, como el consumo de combustible, la temperatura del motor, el rendimiento y la productividad. Al conocer el estado actual de las maquinarias y vehículos en tiempo real, se pueden tomar decisiones informadas sobre su mantenimiento y reparación, lo que reduce el tiempo de inactividad y aumenta la eficiencia y la productividad [5].

El desarrollo de prototipos tecnológicos para el monitoreo y la medición del estado de las motocicletas es de gran importancia para mantenerlas en óptimas condiciones y prolongar su vida útil. Estos prototipos pueden estar equipados con sensores que midan parámetros como la temperatura del motor, la presión del aceite, la velocidad, la aceleración y la posición geográfica de la motocicleta, con esta información en tiempo real, los usuarios pueden tomar decisiones informadas sobre el mantenimiento y la reparación del vehículo, lo que reducirá el tiempo de inactividad y aumentará su

eficiencia y vida útil. Por otro lado, el sistema de rastreo mediante GPS también es de gran importancia, ya que permite conocer la ubicación en tiempo real [8].

- # JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

Actualmente dentro de las industrias del Ecuador usan varios sistemas de medición de energía en diferentes vehículos, con el fin de registrar parámetros eléctricos y otros datos deseados, los mismo que nos permiten determinar el consumo de la batería o combustible y prevenir un sin número de fallas que se pueden evitar. El sistema de rastreo ha tomado mucha importancia en estos últimos lustros, por la alta tasa de delincuencia, lo que permite que el usuario pueda observar las condiciones y la localización del vehículo, con el propósito de mantener la seguridad.

Este proyecto permite ejecutar un gran aporte tecnológico, innovador aplicando directamente a la ingeniería, ya que se puede visualizar todos los parámetros que se planteó y presentar los datos en tiempo real. La mayor parte de las industrias es obtener datos instantáneos para encontrar fallas comunes. Las soluciones fueron desarrolladas mediante la programación de Arduino IP en lenguaje C++ y el desempeño en el uso de los módulos de sensores ocupada en el proyecto, y con los conocimientos adquiridos en el transcurso de la carrera.

● OBJETIVOS

o OBJETIVO GENERAL

- Implementar un sistema para adquirir datos del consumo de energía y ubicación a una moto eléctrica de la Universidad Politécnica Salesiana.

o OBJETIVOS ESPECÍFICO

- Diseñar un sistema de adquisición de los parámetros de voltaje y corriente de la batería y la ubicación de la moto eléctrica.
- Diseñar una plataforma Web basada en un servidor permitiendo capturar los datos del sistema de adquisición diseñado.
- Analizar los parámetros de información de la plataforma.

● CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA O ESTADO DEL ARTE

1.1 FUNDAMENTOS TEÓRICOS

1.1.1 CARACTERÍSTICAS GENERALES DE LA MOTOCICLETA TRINITY VENUS

Figura 1.1 Motocicleta eléctrica Trinity Venus

El Venus es un scooter eléctrico del startup de scooter eléctrico de marca Trinity proveniente de Alemania. La compañía fundada en el año 2013 se dedica a la movilidad de ahorro de recursos, asequible, silenciosa y potente. La compañía tiene sus propias instalaciones de I + D en Alemania y puede personalizar vehículos bajo pedido, tiene un potente motor eléctrico de 2 kW (2,7 hp) para una velocidad máxima de 45 km / h, con una batería LiNiMnCo (NMC) de litio intercambiable para un rango de conducción de 90 km [4].

Tabla 1.1 Características de la moto modelo trinity Venus

Características de la moto	
Velocidad	25 y 45 km/h
Potencia	2 kw(2.7hp)

Recarga	LiNiMnCo(NMC)
Peso	75kg
Durabilidad	1000 ciclos
Distancia	90 km
Voltaje de batería	60V
Precio	2.968 $

1.2.1 CORRIENTE CD

Se denomina corriente directa (CD) o corriente continua (CC) al flujo de una carga eléctrica a través de un material conductor desplazando una cierta cantidad de electrones a lo largo de una estructura molecular. Para el caso de la corriente continua, esta se distingue por tener el mismo sentido de circulación.

La corriente continua se refiere al flujo ininterrumpido de carga eléctrica entre dos puntos de un conductor con potenciales y cargas eléctricas diferentes, sin variar con el tiempo. Esta corriente se define en términos de polaridad de carga, no de intensidad. Por ejemplo, una batería que se agota con una carga baja sigue siendo de corriente continua si la dirección del flujo eléctrico permanece constante, es decir, del polo positivo al negativo.

Todo circuito eléctrico cuenta con dos polos (positivo y negativo) y si los distingue por colores (rojo y negro, respectivamente), con el fin de obstruir que la fuente eléctrica se introduzca en sentido contrario y haya una inversión en la polaridad ya que en ese caso se puede dañar el circuito si no cuenta con una protección para éste. Por eso las baterías de un aparato como en la moto eléctrica debe ir en el orden polar correcto para que funcione.

A diferencia de la corriente continua pues, la corriente alterna (CA) se caracteriza por tener una variación regular y cíclica de su magnitud y sentido en el tiempo [5].

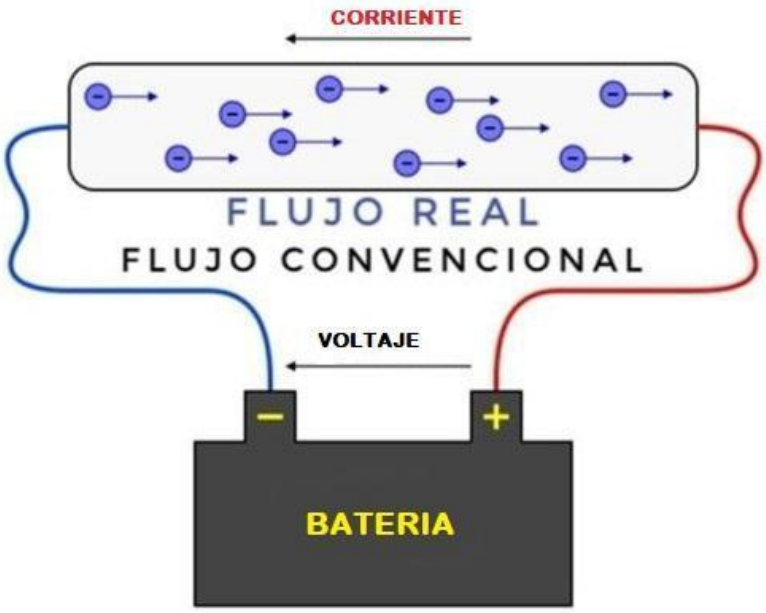


Figura 1.2 Corriente DC y Voltaje DC.

1.3.1 VOLTAJE DC

El voltaje o tensión de corriente continua (CC) o directa (CD) inducen la corriente continua, pues las ondas que son en una sola dirección y la magnitud de la tensión siempre es constante, es decir no presenta cambios con el tiempo, su polaridad siempre se mantendrá igual.

El voltaje de corriente continua induce la corriente continua entre dos puntos, este puede partir de un celular o una batería. La diferencia entre el voltaje alterno y el continua es que el alterno tiene una frecuencia (depende del país) que normalmente se utiliza entre 50 y 60 Hz, y el voltaje continuo su frecuencia se convierte en cero [6].

1.4.1 MÓDULO SENSOR CORRIENTE 50A ACS758LCB-050B-PFF-T-50A.

La importancia del módulo ACS758 es un sensor de corriente puede ser usado como salidas analógicas para detección de corriente CA o CC, dando un rendimiento de líder en la industria por su diseño de amplificadores y filtros patentados, las aplicaciones típicas incluyen como: control de motor, detección y

administración de carga, fuente de alimentación y control de CC a CC, control de inverso y detección de fallas por sobrecorriente [7].

El dispositivo consiste en un circuito de Hall lineal de baja desviación de precisión con una trayectoria de conducción de cobre genera un campo magnético y convierte en un voltaje proporcional. Este dispositivo se optimiza a través de la contigüidad de la señal magnética al conductor. Las características esenciales del módulo, tal que reduce el acoplamiento capacitivo del conductor de corriente a la matriz debido a las altas señales dV/dt, y evita la desviación de desplazamiento en aplicaciones de alto voltaje. Mejora el error de salida total a través de la ganancia de ajuste compensado sobre la temperatura. Suministra un máximo de corriente de 13,5 mA.

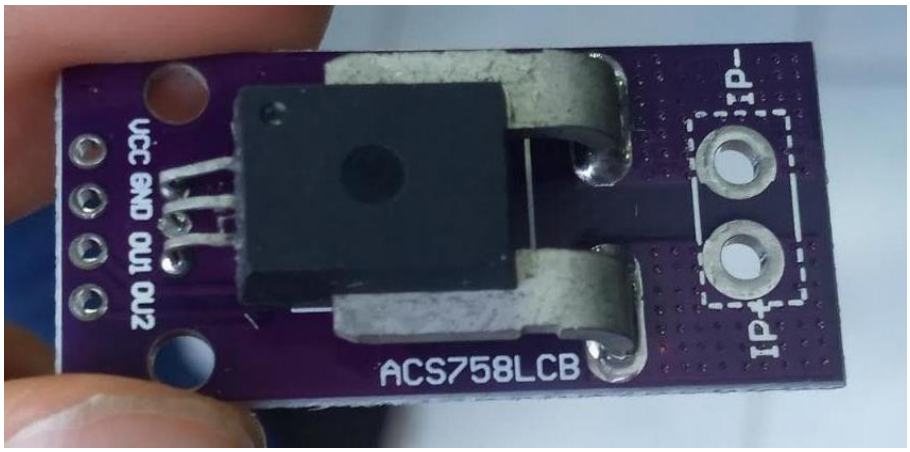


Figura 1.3 Sensor de corriente 50A ACS758LCB-050B-PFF-T-50A.

1.5.1 ARDUINO MEGA 2560

El Arduino Mega 2560 es un microcontrolador basado en el ATmega2560. Tiene 54 pines de E/S (tal que los 15 se pueden usar como salidas PWM), las 16 como entradas analógicas, 4 UART (puertos seriales), un oscilador de cristal de 16MHz, una conexión USB, conexión de alimentación, cabezal ICSP y un botón de reinicio. Además, es compatible con todos los microcontroladores y la mayoría de los escudos diseñados para la placa Arduino Uno y sus versiones anteriores.

Puede funcionar con una amplia gama de voltajes de entrada, con un voltaje de entrada de 7-12V, lo cual se puede utilizar una amplia variedad de fuentes de

alimentación. La importancia del uso del Arduino Mega es que tiene gran cantidad de memoria lo que nos permite almacenar y procesar grandes cantidades de datos favorecidos al programador [8].

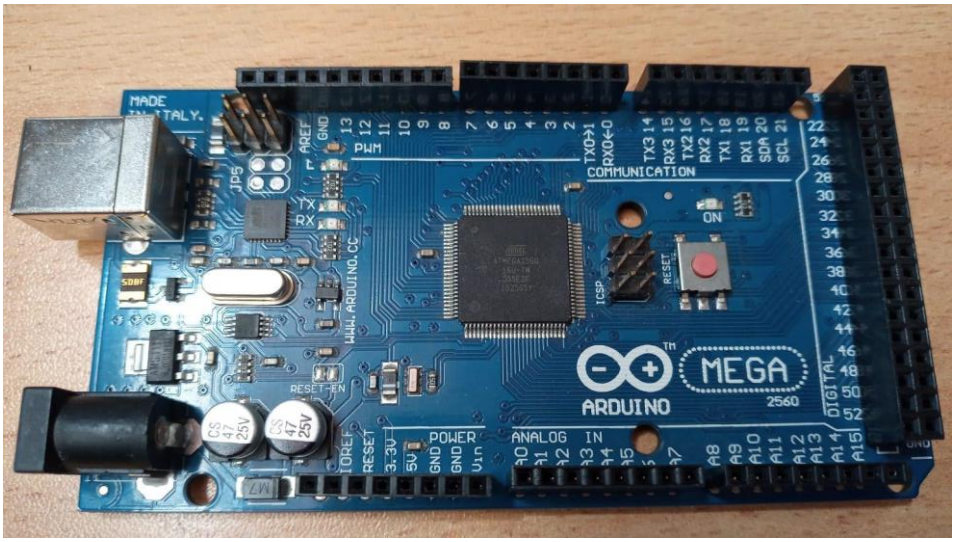


Figura 1.4 Arduino Mega 2560

Tabla 1.2 Características de Arduino Mega 2560 implementado en el proyecto

Características técnicas Arduino Mega 2560	
Voltaje de operación	5v
Pines de entrada-salida digital	14, las cuales 6 se puede usar como salidas PWM
Entradas analógicas	16 pines
DC corriente pin I/O	20 mA
DC corriente pin 3.3v	50 mA
Memoria flash	256 kB
SRAM	8 kB
EEPROM	4 kB
Frecuencia de reloj	16MHz

1.6.1 CABLE CHOGORI

Se puede conectar a la batería M23, es a prueba de agua, puede ser usado en bicicleta eléctrica, motor eléctrico, vehículo eléctrico, cargador de batería, etc., tiene una estructura de bloque por empuje avanzado fácil de montar, contiene puerto de carga y descarga y soporta un máximo de 70 amperios en corriente nominal [9].

Figura 1.5 Cable conector de batería 2+1+5 IP67

Fuente: *(Alibaba, 2022)*

1.7.1 BATERÍA LINIMCCO (NMC) DE LITIO

Se encuentran en una caja metálica para su protección. Estas baterías son ligeras con más resistencia a la descarga y esta ofrece una tensión constante en un ciclo de arranque en frío; también son conocidas como baterías de óxido de litio-manganeso-cobalto (NMC) que se encuentran en la misma celda y está optimizado para una potencia alta, producen entre 3,6 y 3,7 voltios en paralelo y en serie entregan 60 voltios nominal. Una de sus desventajas es que tienen un bajo nivel de funcionamiento en bajas temperaturas (debajo de los cero grados). Otras de sus ventajas es que sufren menos calentamiento, su carga es más rápida, son más resistentes a cargas parciales, tiene una mayor vida útil, reparación sencilla, es fácil el desmonte de esta, entre otras [10].

Especificaciones:

- El voltaje nominal es de 60V.

- Corriente 23Ah.
- Tecnología de la batería NMC
- Voltaje mínimo: 54.4V
- Voltaje máximo: 71.4V
- Corriente de descarga máxima: 30A
- Corriente de carga máxima: 10A

Figura 1.6 Batería de la moto eléctrica.

1.8.1 ESP32-WROOM-32

Este módulo es una solución completa para el desarrollo de productos de IoT, ya que integra WIFI y Bluetooth. Al utilizar WIFI, es posible establecer comunicación de mediano alcance y conectarse a una red LAN, y mediante un Modem Router, acceder a internet. Además, el Bluetooth permite conectar el módulo a dispositivos como smartphones.

Este dispositivo cuenta con dos núcleos de CPU que se pueden controlar de manera independiente, con una frecuencia de reloj ajustable que varía desde los

80 MHz hasta los 240 MHz. Además, es posible apagar uno de los núcleos para utilizar el co-procesador de baja potencia con el fin de supervisar los periféricos y detectar cambios de estado. El módulo también dispone de una amplia gama de periféricos integrados, tales como sensores táctiles capacitivos, sensores Hall, amplificadores de bajo nivel de ruido, interfaz para SD, Ethernet, SPI, UART, I2S e I2C.

Para alimentar este módulo se requieren 3.3 voltios de tensión y es importante evitar la alimentación con 5 voltios, ya que podría dañar el dispositivo. Se recomienda colocar un capacitor de 100uF en paralelo con la fuente de alimentación, con el fin de filtrar los picos de corriente. Además, es importante tener en cuenta que los pines de entrada y salida (GPIO) funcionan con 3.3 voltios, lo que significa que se deben utilizar conversores de nivel para conectarlos a sistemas que funcionan con 5 voltios [11]. El módulo ESP32 consume 80-180 mA con funcionamiento WiFi.

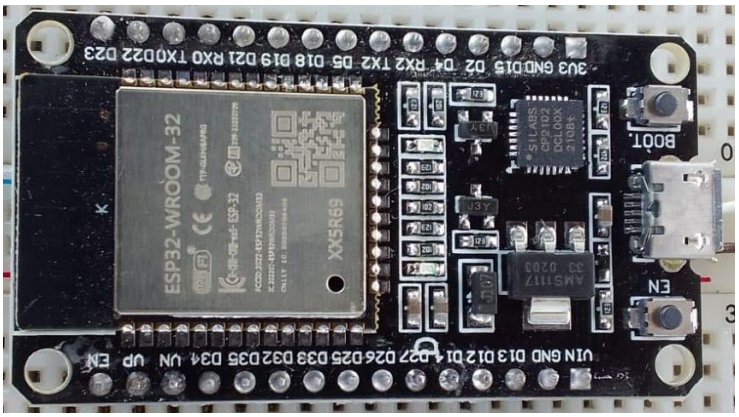


Figura 1.7 ESP32-WROOM-32

1.9.1 MÓDULO GPS NEO-6M V2 APM2.5

El módulo Neo-6 ofrece un alto rendimiento de U-Blox 6 motor de posicionamiento, son ideales para vehículos autónomos como aeromodelismo, quadcopters, helicópteros, robots móviles, así también como para realizar control de vuelo o control de recorridos. Es de comunicación serial por lo que son leídos fácilmente por cualquier microcontrolador o tarjeta de desarrollo [12].

Especificaciones técnicas:

- Voltaje de funcionamiento de la placa: 3V a 5V.
- EEPROM que guardar datos de configuración.
- Indicador de señal por LED.
- Tasa de actualización: 5Hz.
- Velocidad de transmisión: 4800, 9600, 38400, 57600.
- Dimensiones de la antena: 2.5 x 2.5 [cm].
- Dimensiones módulo: 3.5 x 2.5 [cm].

Figura 1.8 Módulo GPS NEO-6M V2 APM2.5

1.10.1 CONVERTIDOR VOLTAJE DC-DC STEP-DOWN 3A LM2596

Su función consiste en proporcionar un voltaje de salida eficiente y variable, el cual es menor que el voltaje de entrada. El rango de voltaje de entrada va desde 4.5 voltios hasta 40 voltios en corriente continua, mientras que el rango de voltaje de salida oscila entre 1.23 voltios y 37 voltios en corriente continua. Además, este dispositivo tiene una capacidad máxima de corriente de salida de 3 amperios y se puede utilizar para diversas aplicaciones, como regulador de voltaje para baterías de automóviles de 12V o 24V, fuente de alimentación, robótica móvil, drones y cargadores USB para teléfonos móviles [13].

Especificaciones Técnicas:

- Corriente de salida: máximo de 3A, 2.5 recomendado (usas disipador para corrientes mayores de 2 amperios).
- Potencia de salida: 25W.
- Eficiencia de conversión: 92%.
- Frecuencia de trabajo: 150 KHz.
- Protección limitadora de corriente: si.
- Dimensiones: 43 x 21 x 13 [mm].

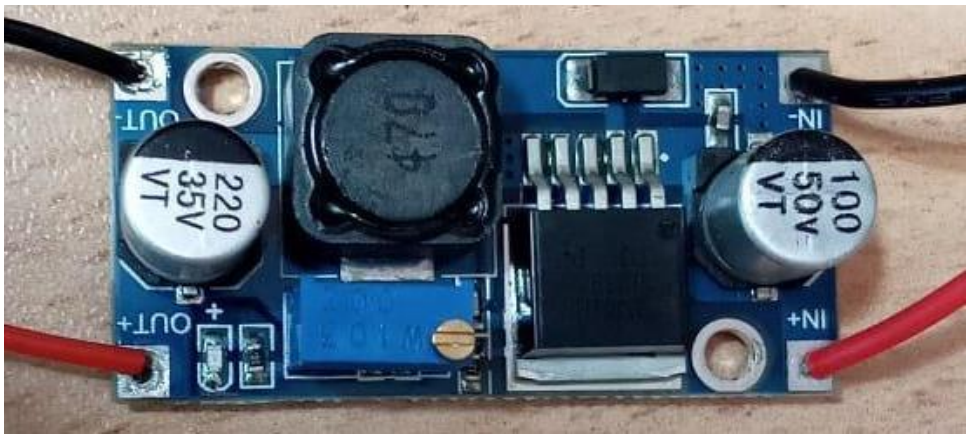


Figura 1.9 Convertidor de voltaje DC-DC Step-Down LM2596

1.11.1 PORTAFUSILES IMPERMEABLE ATC/ATO, 12AWG

Es un dispositivo encargado de salvaguardar en su interior un fusible, tiene un cable resistente de 12 AWG con la función de protección al circuito electrónico, se adapta a distintos fusibles de hoja de diferentes amperajes tipo ATC/ATO [14].

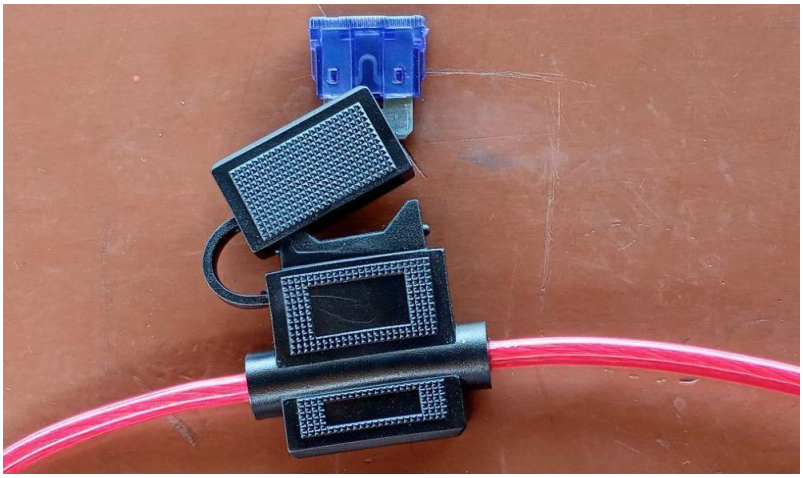


Figura 1.10 Portafusible en línea.

1.1.1.11 Fusible

Los fusibles en todo vehículo son unas pequeñas piezas que protegen todos los sistemas eléctricos, en este caso va a proteger el circuito eléctrico de la moto eléctrica. Su misión es fundirse o estropearse cuando detecta una subida de tensión o corriente excesiva, actúa como seguro de vida para el sensor de corriente ya que el arranque de una moto da de resultado unos picos de corriente que supera su corriente nominal. Cuando el fusible descubre una intensidad de corriente muy alta y que al no cortar se provocará una avería peligrosa en el circuito eléctrico pues, de forma automática se funde para impedir el paso de corriente dañina y el circuito queda a salvo. Los fusibles se diferencian por el color dependiendo el amperaje máximo que soporta [15].

Figura 1.11 Fusible.

1.12.1 COMUNICACIÓN EN SERIE

Para la comunicación en serie usando la placa de Arduino, se puede elegir los pines Seriales estándar como Rx Tx del software Arduino, desde la UART dentro de la placa Arduino, por lo que también se llama Serial TTL.

La comunicación Serial permite conectar dos dispositivos diferentes para enviar y recibir datos entre ellos, dispositivos como entre el Arduino mismo y un Módulo ESP32, o Módulo ESP8266, Módulo GPS, entre otros.

El puerto Serial TTL puede transformarse a un estándar industrial como con el RS232 y RS458. Al usar RS232 se trabaja con un rango de voltaje estandarizado

que son los 5 voltios que ocupan las placas de Arduino. En cambio, si se usa el puerto RS485 se puede configurar una red de comunicaciones entre dispositivos en la que uno trabaje como MAESTRO (Master) conectado a varios dispositivos como ESCLAVOS (Slave), el número de dispositivos a usar esta entre 2 a 32 con una distancia máxima entre los dispositivos de 1220 metros.

El puerto serie UART se puede usar tanto en Arduino como en un PLC basado en Arduino, para añadir esta comunicación entre dispositivos se utiliza la biblioteca llamada softwareserial.h [16].

1.13.1 PROTOCOLO MQTT

MQTT es una abreviatura de Transporte de Telemetría del Servidor de Cola de Mensajes. Este protocolo de mensajería se caracteriza por ser liviano, lo que lo hace ideal para casos en los que los clientes requieren un código con una huella pequeña y están conectados a redes poco confiables o con recursos de ancho de banda limitados. Es utilizado principalmente para la comunicación entre máquinas (M2M) o para conectar dispositivos de Internet de las cosas a redes poco confiables para el usuario, como nubes.

MQTT se ejecuta sobre TCP/IP que utiliza la topología PUSH/SUBSCRIBE que es cliente/publicador, el publicador recibe comunicaciones de los clientes y las envía a otros clientes, cada cliente puede ser un editor, un suscriptor o incluso ambos [17].

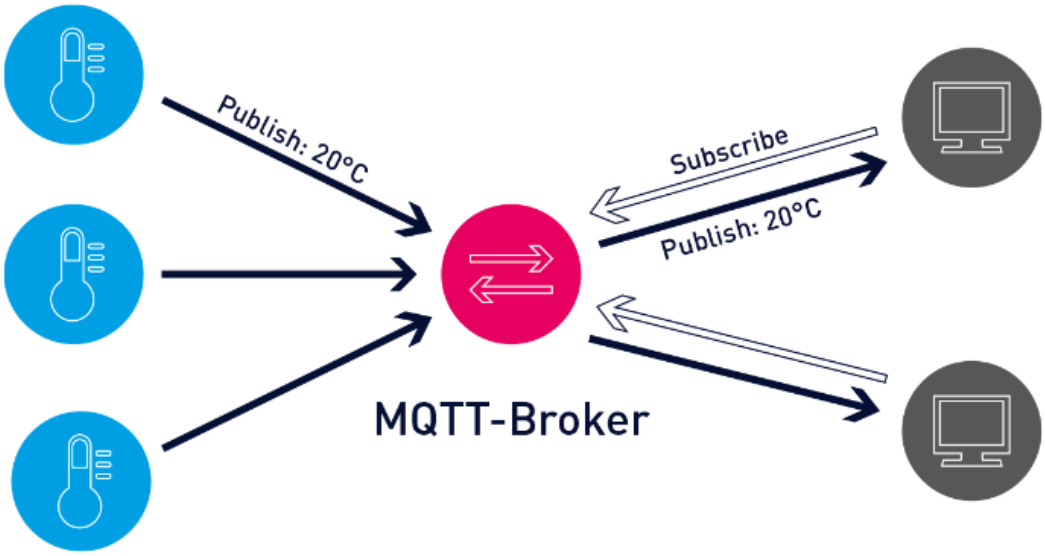


Figura 1.12 Comunicación MQTT.

Fuente: *(Paessler,2023)*

1.14.1 ADAFRUIT IO

Es una plataforma que permite conectar todo un sistema IoT recogiendo datos desde los dispositivos que se encuentran conectados, almacenarlos y analizarlos en tiempo real, en Adafruit se puede agregar, almacenar y visualizar cierto tipo de datos en tiempo real en la nube desde dispositivos IoT conectados. También permite interactuar con los dispositivos desde paneles de control conocidos como dashboards basados en la web.

Este sistema es compatible con dispositivos como Arduino, Raspberry Pi, ESP8266, ESP32, entre otros. Además, es capaz de trabajar con APIs REST y MQTT para una mayor flexibilidad. Adafruit ofrece una versión gratuita que permite enviar hasta 30 datos por minuto, almacenarlos durante un máximo de 30 días, y generar disparadores o alarmas cada 15 minutos. También es posible crear hasta 5 paneles con 10 campos o feeds por panel para una mayor personalización y organización de los datos.

Al crear una cuenta en adafruit en la página: https://io.adafruit.com obtendremos nuestro nombre de usuario y la clave (“My key”) que nos asigna la página mismo, este es necesario ya que al tener una conexión con un dispositivo

IoT necesitaremos de estos datos para enviar la información a nuestra cuenta, y en nombre de nuestro feeds o campos para visualizar los datos en tiempo real [17].

1.15.1 WiFi

WiFi es una tecnología inalámbrica utilizada para conectar computadores, teléfonos inteligentes, módulos de IoT entre otros dispositivos a internet. WiFi es la señal de radio enviada desde un router inalámbrico a un dispositivo electrónico cercano en la que traduce la señal en datos que se pueda ver o usar.

Hay opciones de conectarse a internet, una de ellas es sin cables como se puede realizar mediante una red móvil. La más usada y popular entre dispositivos móviles es la del punto de acceso Hotspot, que se usa de manera segura mientras viaja, en la actualidad cualquier smartphone o tablet puede usarse como un hotspot temporalmente siendo una excelente opción si se necesita ocasionalmente, para ello se necesita de batería y de datos móviles para su conexión, incluso ofrece un mayor alcance de WiFi [18].

1.16.1 Buzzer

El zumbador, también conocido como buzzer, es un pequeño transductor que convierte la energía eléctrica en sonido. Su conexión es sencilla, ya que solo se necesita conectar el polo positivo al signo más y el polo negativo al signo menos en una batería u otra fuente de corriente directa. El buzzer se basa en el efecto piezoeléctrico de los materiales, el cual funciona de tal manera que, al aplicar un voltaje, el volumen del material cambia ligeramente y se mantiene en ese estado hasta que se quita el voltaje aplicado. De esta forma, es posible generar sonidos simples o melodías en diferentes dispositivos electrónicos.

Con el fin de generar un sonido continuo, es necesario que las placas del buzzer vibren constantemente. Para lograr esto, se instala un oscilador que hace que los materiales cambien de estado repetidamente, permitiendo que el buzzer emita varios ciclos de sonido y así producir un audio perceptible [19]. La corriente máxima que consume el buzzer es de 150mA.

Figura 1.13 Buzzer.

1.17.1 VENTILADOR DC

Este ventilador axial cuenta con conmutación electrónica completamente integrada y ofrece un flujo de aire constante y fluido para el dispositivo que está refrigerando, en nuestro caso a la batería de la caja y al encapsulado del sensor del corriente en caso de el aumento de temperatura, por el uso constante o la prueba que se realiza.

Figura 1.14 Ventilador DC.

1.18.1 JSON (JAVASCRIPT OBJECT NOTATION)

JSON es un formato ligero de intercambio de datos, este es de una fácil lectura y escritura para el usuario, también es fácil de analizar y generar por parte de las máquinas. El estándar de programación es ECMA-262 3a Edición - Diciembre de 1999. Los lenguajes de programación que utiliza son de la familia C que incluyen C, C++. C#, Java, JavaScript, Perl, Python, entre otras.

Para la conversión de tipo de datos, JSON admite valores tipo serie, número y booleanos, pero no admite octales y hexadecimales. También maneja matrices y objetos nulos y vacíos; JSON tiene un valor especial denominado null que puede establecerse en cualquier tipo de datos incluyendo matrices, objetos, tipos numéricos y booleanos [20].

1.19.1 MICROSD CARD ADAPTER

Un lector de tarjetas microSD es un dispositivo que permite almacenar información en una tarjeta SD y se puede integrar en proyectos de electrónica y Arduino. En ambos tipos de lectores, la lectura se puede realizar mediante el bus SPI. Aunque algunos lectores pueden disponer de otros interfaces, como bus I2C o UART, normalmente se prefiere utilizar SPI debido a su alta tasa de transferencia. De esta forma, es posible acceder y transferir datos de manera rápida y eficiente desde la tarjeta microSD a otros dispositivos electrónicos.

El módulo de lectura de tarjetas SD o microSD suele tener una tensión de alimentación de 3.3V, sin embargo, en la mayoría de los casos se incluye la electrónica necesaria para que se pueda conectar fácilmente a Arduino, lo que suele incluir un regulador de voltaje que permite alimentarlo directamente con 5V. El uso de una tarjeta SD o microSD en conjunto con Arduino ofrece la ventaja de proporcionar una memoria casi ilimitada para nuestros proyectos, que además es no volátil, lo que significa que la información se mantiene incluso cuando se elimina la alimentación. Además, estas tarjetas pueden ser extraídas y conectadas a un ordenador con facilidad para transferir datos.

Una desventaja significativa de utilizar una tarjeta SD o microSD con Arduino es que puede suponer una carga importante para el microcontrolador. El programa en sí puede ocupar hasta el 40% de la memoria Flash y casi el 50% de la memoria dinámica, lo que puede ser bastante exigente para el procesador [21].

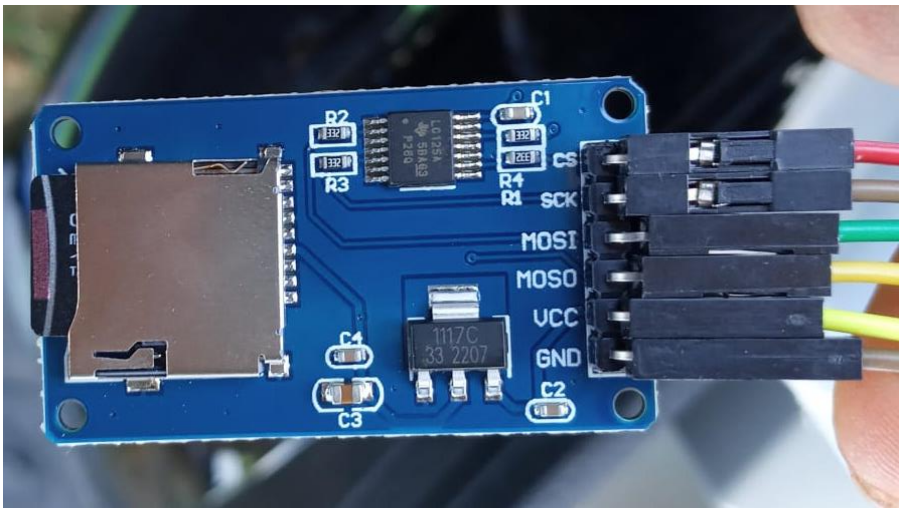


Figura 1.15 Adaptador microSD.

Fuente: *Autores*

• CAPÍTULO 2: MARCO METODOLÓGICO

Este proyecto de titulación tiene como objetivo implementar un sistema de lectura del consumo de energía en vehículos, incluyendo el consumo de combustible, velocidad recorrida, potencia, etc. Las mediciones se mostrarán en un manómetro ubicado en un lugar apropiado. Las pruebas se realizan mediante análisis y pruebas, teniendo en cuenta que el rendimiento del motor puede variar debido a diversos factores internos y externos a la conducción.

El enfoque de potencia específico propuesto en este trabajo puede predecir y caracterizar las tasas de consumo de energía en diferentes tipos de terreno y modos de operación del vehículo. Además, el uso del análisis de eficiencia de energía transformada desde la fuente primaria hasta la batería o tanque de combustible (WTW) de motocicletas para ciclos de conducción específicos, y la comparación entre tecnologías convencionales y eléctricas, puede indicar caminos para implementar soluciones de transporte más sostenibles.

o 2.1 DISEÑO DE SISTEMA DE ADQUISICIÓN DE DATOS Y UBICACIÓN

Para medir el voltaje y la corriente de una motocicleta eléctrica, es necesario utilizar un conector auxiliar que permita conectar el sistema eléctrico de la motocicleta al dispositivo de medición. Para medir la corriente, es necesario interrumpir el circuito en serie, lo que requiere cortar la cubierta del conector y conectar un sensor de corriente cortando un cable alimentador.

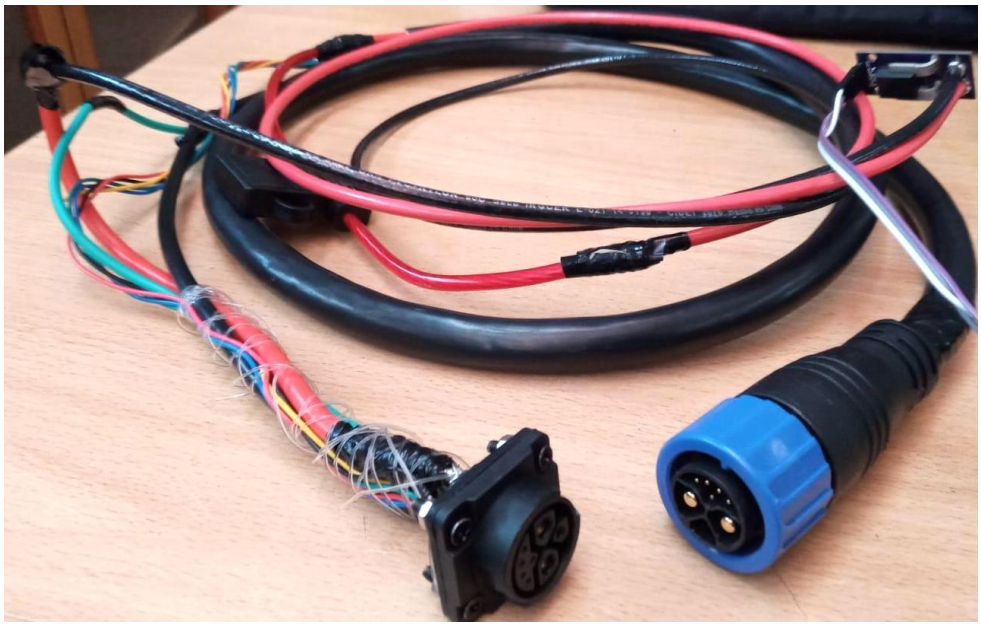


Figura 2.1. Cable auxiliar chogori.

Figura 2.2. Entrada de cable chogori M23 a la batería.

Para medir el voltaje y la corriente en la adquisición de datos, se utiliza el sensor de corriente ACS758LCB, que cuenta con una librería en el software Arduino

IDE, lo que permite una programación menos compleja. El sensor tiene un rango de medición de 0 a 50 amperios en corriente continua. Sin embargo, para proteger el sensor y el circuito eléctrico, se coloca un fusible para los saltos de corriente. Además de medir la corriente, el sensor también permite medir el voltaje continuo y, gracias a esto, se puede obtener el voltaje y la corriente de la batería de la moto eléctrica.

Figura 2.3. Medición de voltaje y corriente de la batería.

Luego se comprueba el voltaje de la batería mediante un multímetro para comparar los valores con las especificaciones del catálogo de la motoneta y con el resultado de la programación.

Figura 2.4. Comprobación de voltaje de batería.

El sensor obtiene el resultado del voltaje y la corriente de la batería, al arrancar la moto la corriente aumenta en el arranque y una vez mantenida la velocidad se mantiene la corriente constante. Al dejar de acelerar la corriente desciende a cero amperios y se puede visualizar este cambio de corriente mediante el monitor serial del software Arduino IDE. Sin embargo, el voltaje se mantiene constante y va disminuyendo mientras se agota la batería de la motoneta.

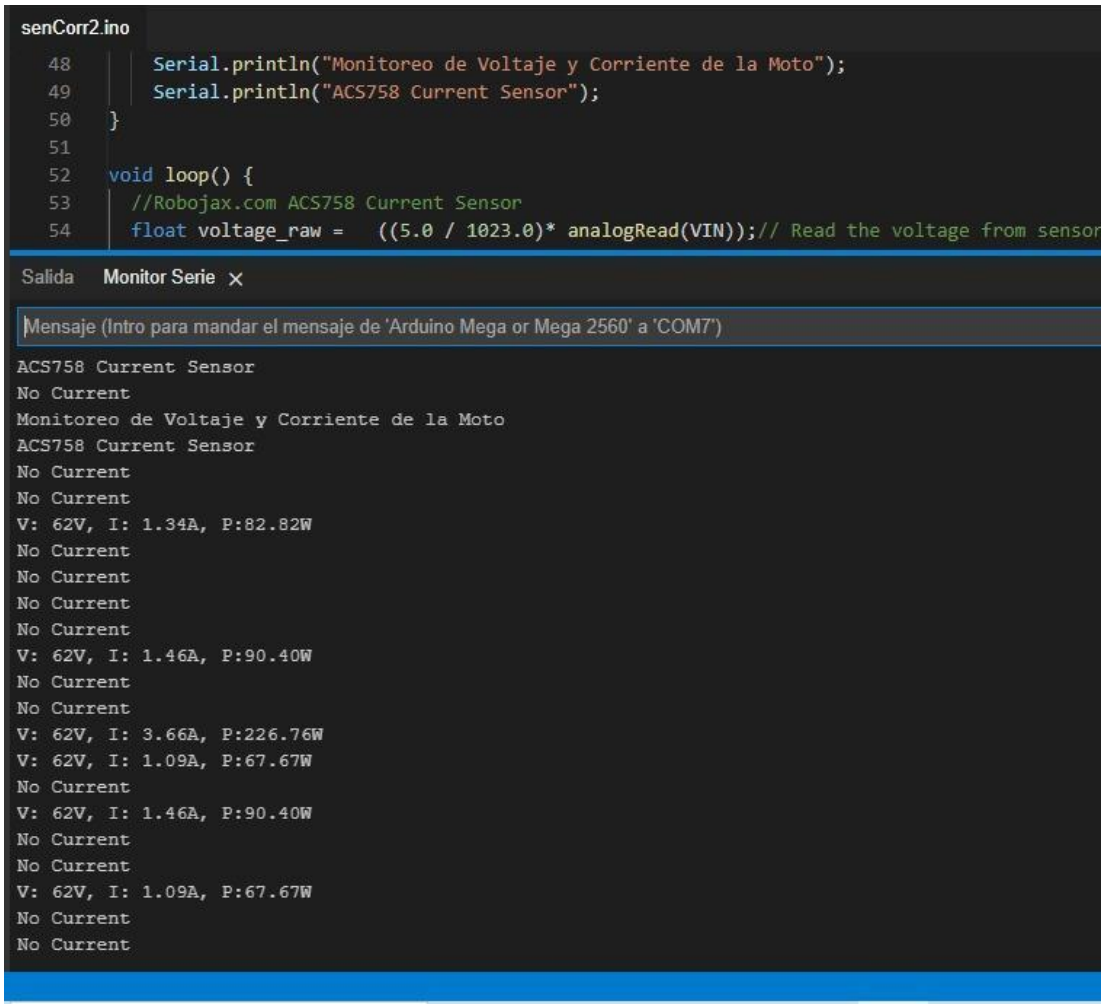


Figura 2.5. Lectura del sensor en arranques sin carga.

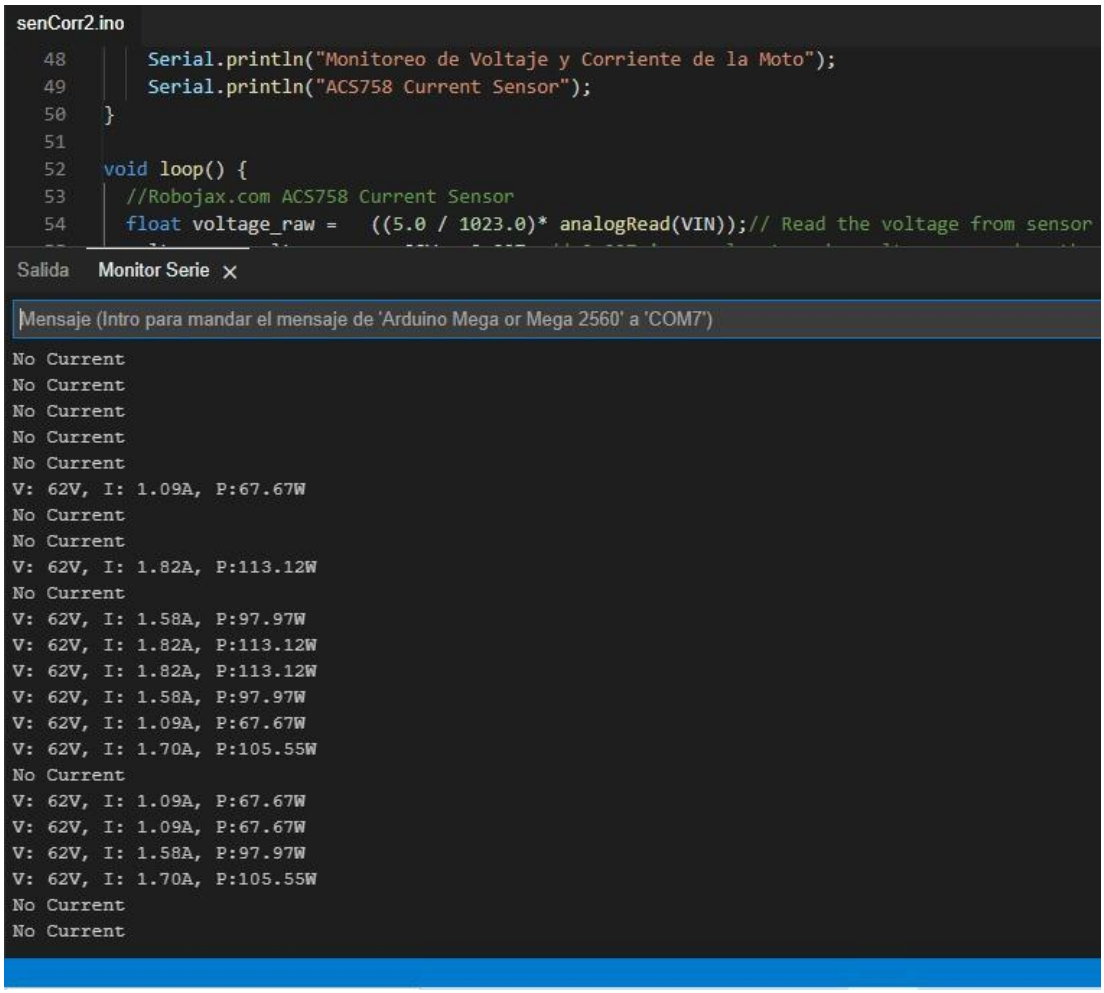


Figura 2.6. Lectura del sensor en aceleración constante y suelto del acelerador.

Para la ubicación de la moto se utiliza un módulo GPS, este mismo módulo cuenta con una librería en arduino para la programación de este mismo y obtener las coordenadas en la que se encuentre ubicado el dispositivo. Se incluye las librerías a utilizar para el GPS como el <TinyGPS++.h>, y en el código arduino se obtiene la altitud, latitud y longitud para las coordenadas.

```
void gpsLoop(){
  while(Serial1.available() > 0){
    if(gps.encode(Serial1.read())){
      if(gps.location.isValid()){
        latitud = gps.location.lat();
        longitud = gps.location.lng();
        altura = gps.altitude.meters();
        sendData();
      }
    }
  }
}
```

Figura 2.7. Programación para datos de las coordenadas de GPS.

Se utiliza un formato ligero de intercambio de datos llamado JSON que es de fácil lectura y escritura para los usuarios para poder enviar los datos las coordenadas del gps y la lectura de la potencia de la motoneta cada cierto tiempo.

```
void sendData(){
  if(millis()-lastSend>500){
    lastSend = millis();
    // Serial.println(latitud);
    DynamicJsonDocument doc(200);
    doc["value"]["sensor-1"] = analogLecture;
    doc["value"]["sensor-2"] = altura;
    doc["lat"] = latitud;
    doc["lon"] = longitud;
    doc["ele"] = altura;
    String toReturn = "";
    serializeJson(doc,Serial);
    Serial.println();
  }
}

#endif
```

Figura 2.8. Programación para enviar datos mediante JSON

Al comprobar los datos enviados del microcontrolador Arduino al módulo WIFI ESP32 se puede visualizar en el monitor serial de la programación en el arduino.

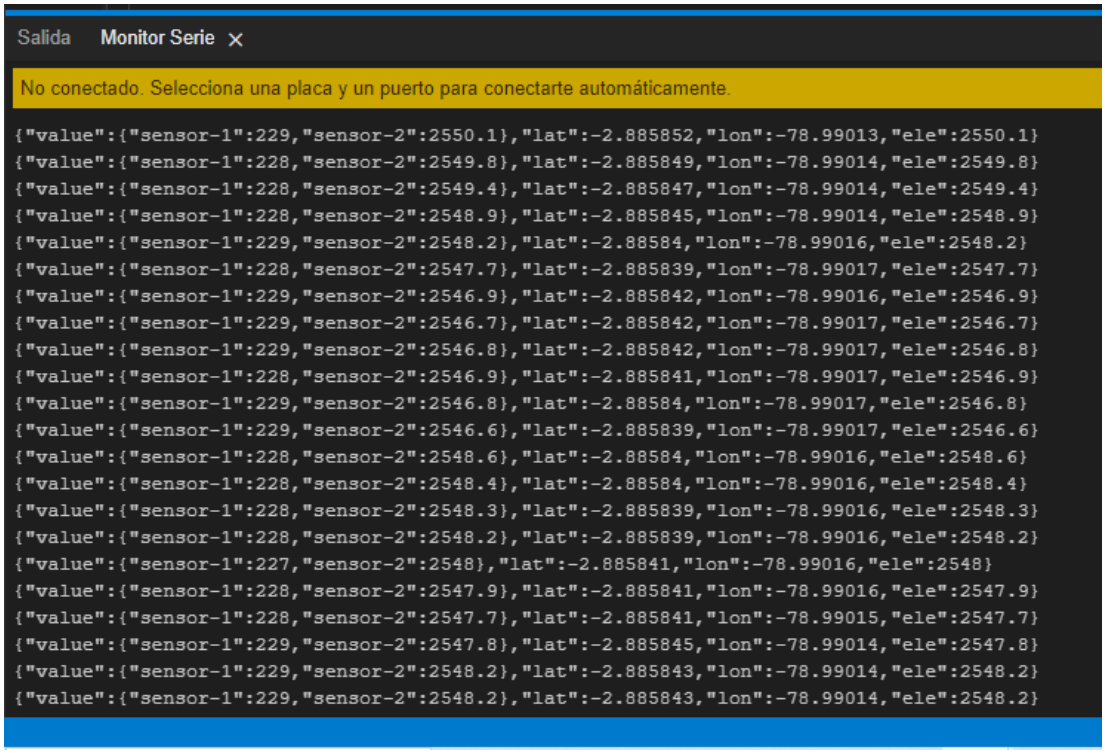


Figura 2.9. Visualización de envío de datos por monitor serial.

En el módulo ESP32 se publican los datos a la nube, se obtienen del microcontrolador y se publica a la nube cada 5 segundos.

```
unsigned long now = millis();
if (now - lastMsg > 5000 && lectura != "")
{
  String toSend ="";
  lastMsg = now;
  //serializeJsonPretty(doc,Serial);
  analogLecture = doc["value"]["sensor-1"];
  altitud = doc["value"]["sensor-2"];
  doc["value"] = analogLecture;
  serializeJson(doc,toSend);
  Serial.println("Publishing analog:" + toSend);
  client.publish(topicAnalog, toSend.c_str());
  doc["value"] = altitud;
  toSend = "";
  serializeJson(doc,toSend);
  Serial.println("Publishing altitud:" + toSend);
  client.publish(topicAltitud, toSend.c_str());
}
```

Figura 2.10. Programación de publicación de datos a la nube

Visualización de publicación de datos enviados desde el módulo WIFI a la nube Adafruit.io.

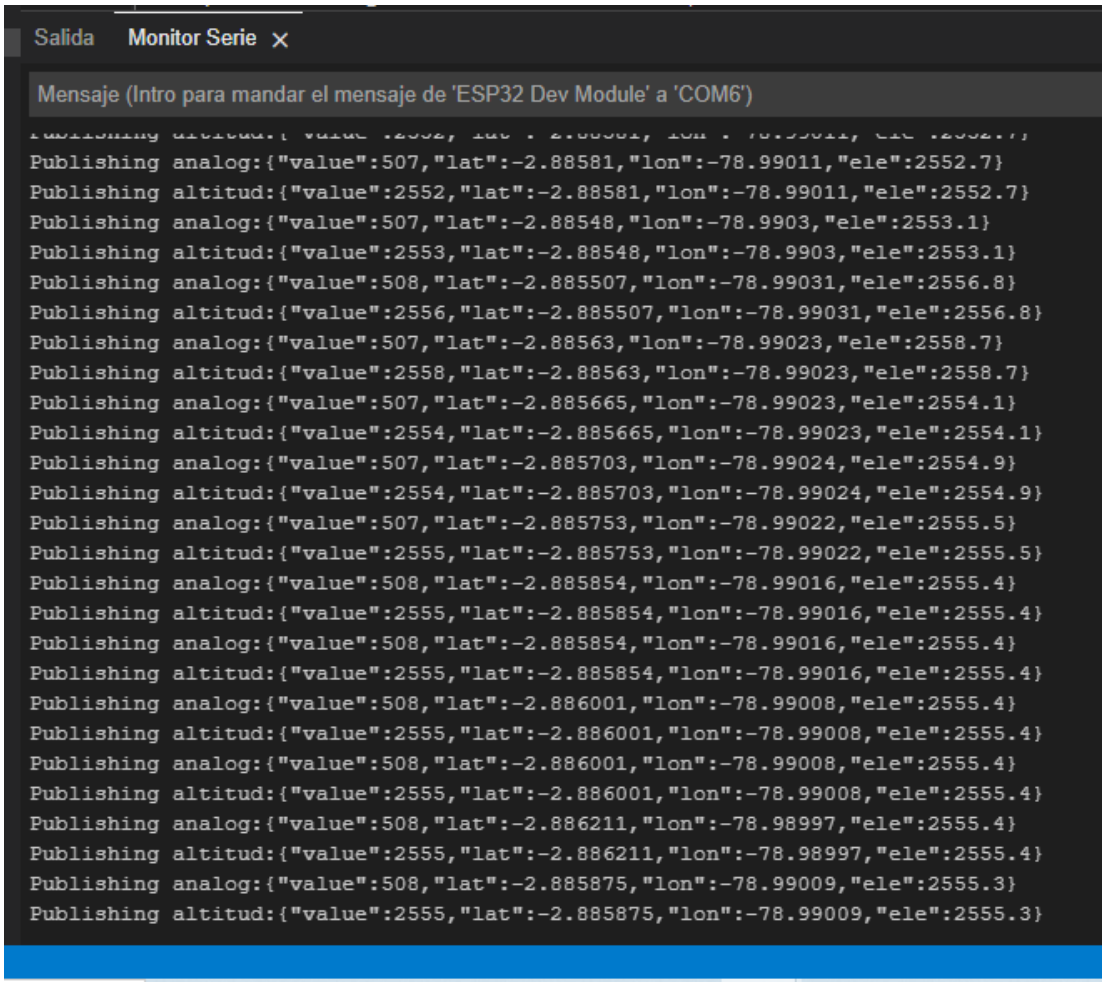


Figura 2.11. Datos publicados desde el módulo WiFi hacia la nube Adafruit.

Se le agrega un buzzer al circuito como señal de que el circuito está encendido correctamente o que se encuentra con batería. En el ESP32 se programa una secuencia de sonido al buzzer.

```
void loop()
{
  ledcWriteNote(BUZZER_CHANNEL, (note_t)NOTE_A, 8);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_C, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_D, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_E, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_F, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_G, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_A, 4);
  delay(500);
  ledcWriteNote(BUZZER_CHANNEL, NOTE_B, 4);
  delay(500);
  ledcDetachPin(13);
```

Figura 2.12. Programación de buzzer en ESP32.

Al cable que alimenta a la batería se le realizó ciertos arreglos para que quede dentro el baúl de la motoneta y se pueda cerrar el asiento sin inconvenientes.

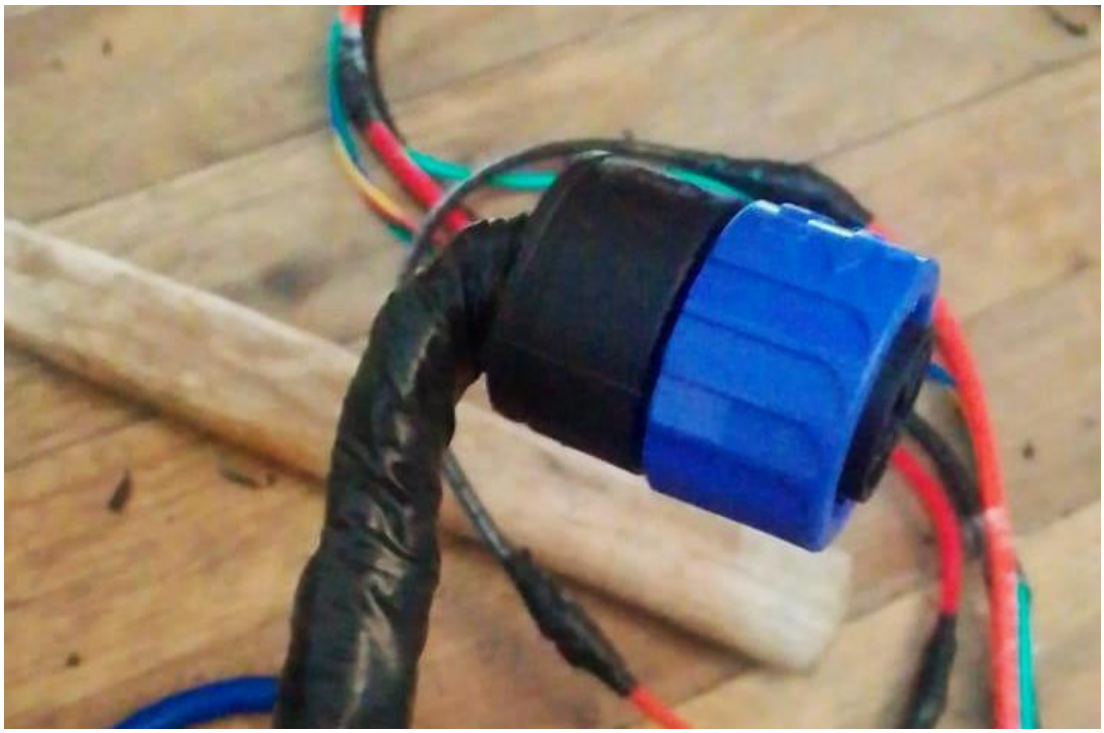


Figura 2.13. Ajustes en el cable de alimentación a la motoneta.

Para comenzar el diseño de la placa, es necesario ubicar la conexión de los componentes del circuito y proceder con el diseño en base a esta información.

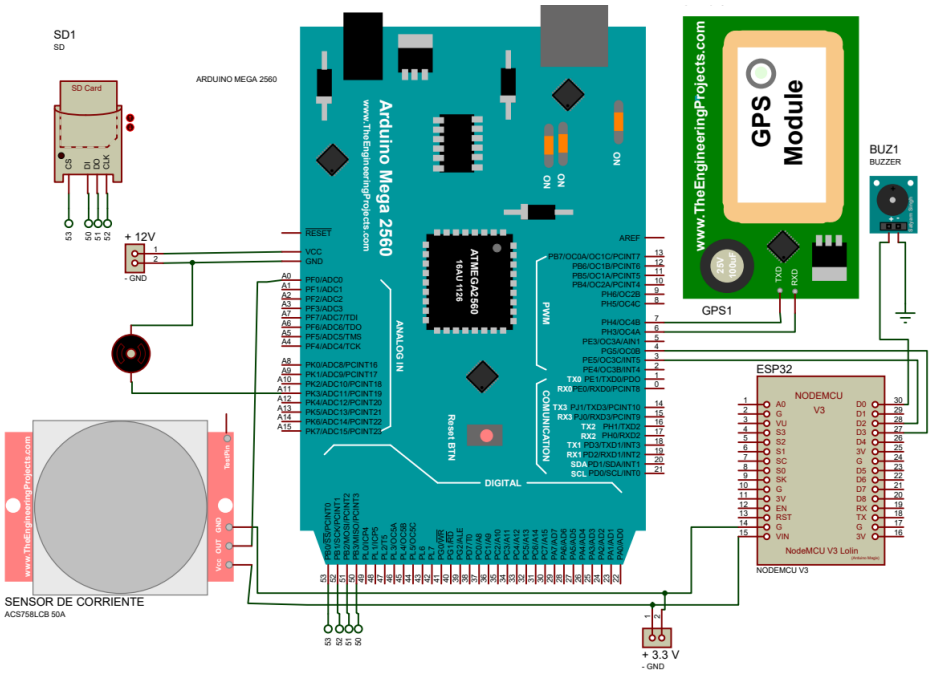


Figura 2.14. Circuito de adquisición de datos.

Para la alimentación del circuito se debe considerar el consumo de cada elemento electrónico a utilizar, ya que si no abastece la corriente para todo elemento el circuito no funcionará correctamente y, por lo tanto, los datos no se publicarán en la nube.

Tabla 2.1 Consumo máximo de corriente del circuito

Elementos Electrónicos	**Consumo máximo de corriente [mA]**
Sensor de corriente	13.5
Arduino Mega 2560	50
Módulo ESP32	180
Módulo GPS	67
Buzzer	150

Ventilador	100
Lector microSD	100
Total de corriente a consumir	***660***

El consumo máximo del circuito es de 660 mA, por lo cual es necesario adquirir una fuente que supere los 800 mA para evitar fallos del circuito y abastecer con la alimentación de todo el circuito.

Se utiliza acrílico para la cubierta del circuito debido a que este material es resistente a diversos factores climáticos, tales como el viento, el frío, el agua y especialmente al calor, ya que se encuentra ubicado debajo de la moto, donde la temperatura es considerablemente alta. El acrílico puede resistir temperaturas de hasta 180 grados centígrados, ya que, a temperaturas superiores a ésta, el material empieza a derretirse.

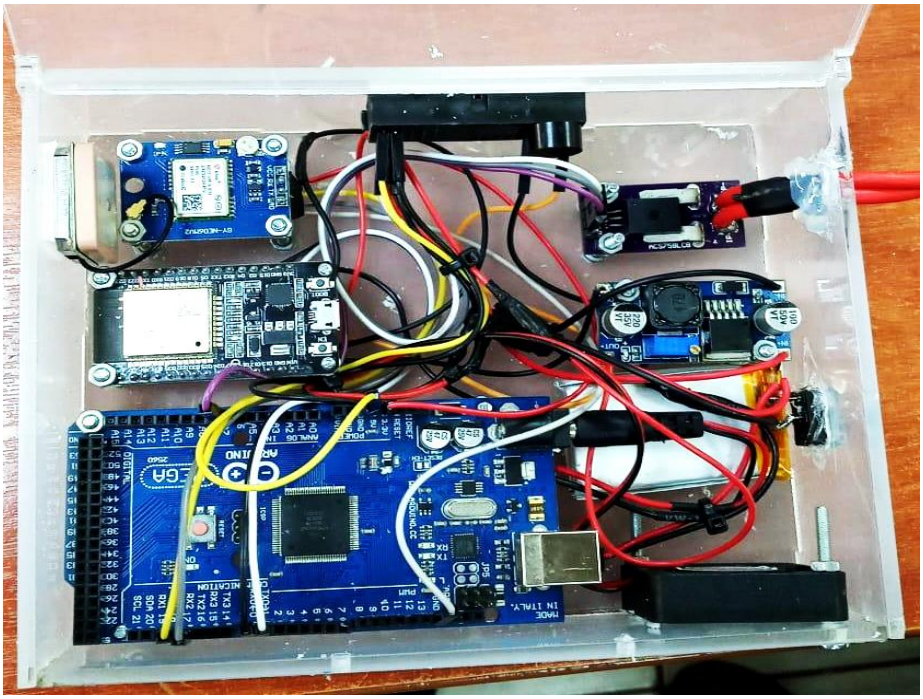


Figura 2.15. Cubierta del circuito, vista frontal con tapa abierta

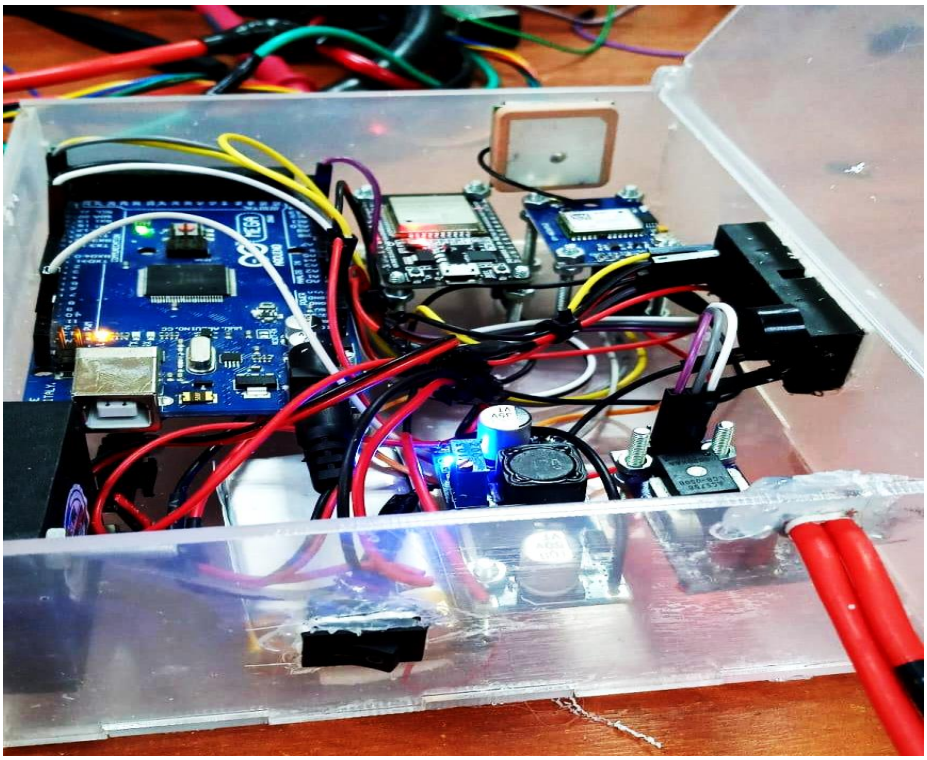


Figura 2.16. Cubierta del circuito, vista lateral con tapa abierta

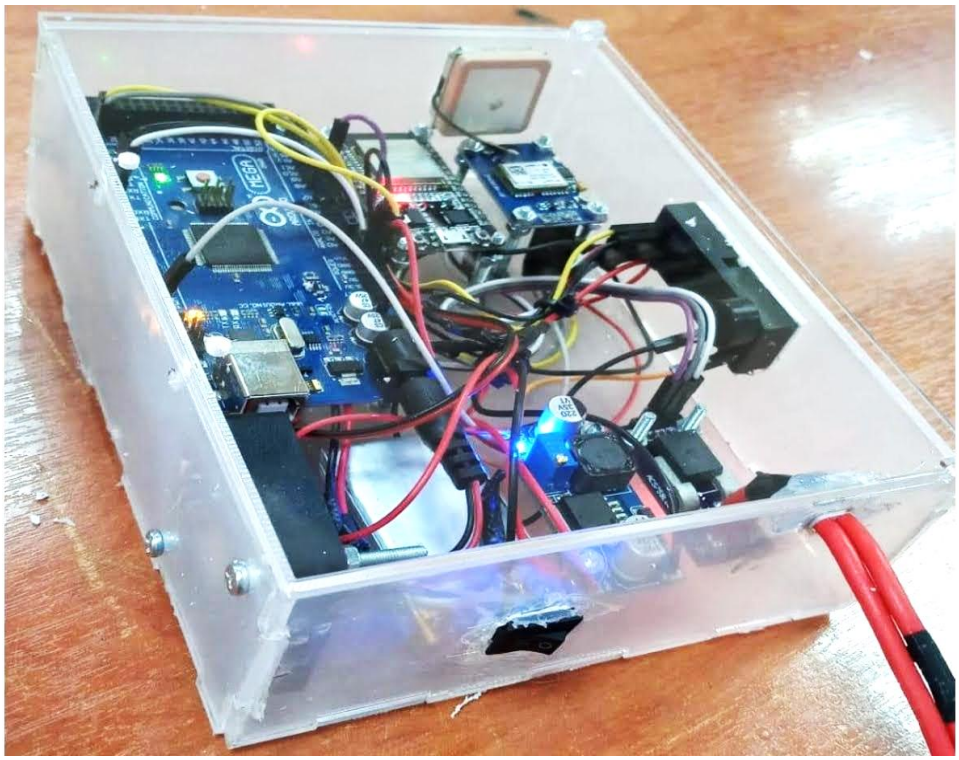


Figura 2.17. Cubierta del circuito, vista lateral con tapa cerrada

Para programar la tarjeta SD, es necesario conectar los pines del módulo lector de memoria al Arduino. De esta forma, mediante una programación específica, el Arduino puede comunicarse con la tarjeta SD y almacenar los parámetros leídos por el sensor de corriente y voltaje.

De este modo, se permite el registro de consumo de energía de la batería sin necesidad de estar conectado a una red WiFi, lo que proporciona un respaldo confiable en caso de que el usuario de la motoneta no tenga acceso a internet.

```
void setup() {
  Serial.begin(9600);
  aReadInit();
  gpsInit();
  lastSend = millis();
////////////////////7
  pinMode(led, OUTPUT);
  Serial.print("Iniciando SD card...");
  pinMode(53, OUTPUT);
  pinMode(led, OUTPUT);
  digitalWrite(53, HIGH);
  if (!SD.begin(53)) {
  Serial.println("Fallo comunicacion o no existe SD");
  digitalWrite(led, HIGH);
  return;}
  digitalWrite(led, LOW);
  Serial.println("SD Iniciada.");

  dataFile = SD.open("datalog.txt", FILE_WRITE);
```

Figura 2.18. Programación de comunicación y apertura con la tarjeta SD.

Al establecerse la comunicación entre el Arduino y la tarjeta SD, se abre un archivo de tipo .txt, se almacena la información y se cierra el archivo. Este proceso se repite cada 5000 milisegundos, permitiendo el registro continuo de la información leída por el sensor de corriente y voltaje. Este proceso se mantendrá activo mientras el circuito esté conectado y tenga suministro de energía.

```
  String dataString = "";
  sensor = analogRead(analogLecture);  // Lee el sensor
  dataString += String(sensor);   //Almacena en la cadena

dataFile = SD.open("datalog.txt", FILE_WRITE);

  if (dataFile) {
    digitalWrite(led, LOW);
    dataFile.println(dataString);
    delay(1000);
    dataFile.close();
    Serial.println(dataString);
  }
  else {
    Serial.println("Fallo comunicacion.txt");
    Serial.println("Revise conecxion");
    digitalWrite(led, HIGH);
    delay(5000); }
}
```

Figura 2.19. Lectura y almacenamiento de potencia de la batería de la moto en la tarjeta SD.

o 2.2 Diseño en la plataforma Web Adafruit IO

Para el diseño del servidor web, se utiliza la plataforma en la nube Adafruit IO, la cual permite agregar, almacenar y visualizar los datos enviados desde el microprocesador. En esta plataforma, se diseña un panel de control que recibe datos como la potencia, altitud, latitud y longitud (para la ubicación) de la programación, los cuales pueden ser visualizados en el servidor.

Adafruit IO es una plataforma de acceso libre en la cual se puede crear una cuenta y comenzar a utilizar la nube de forma gratuita.

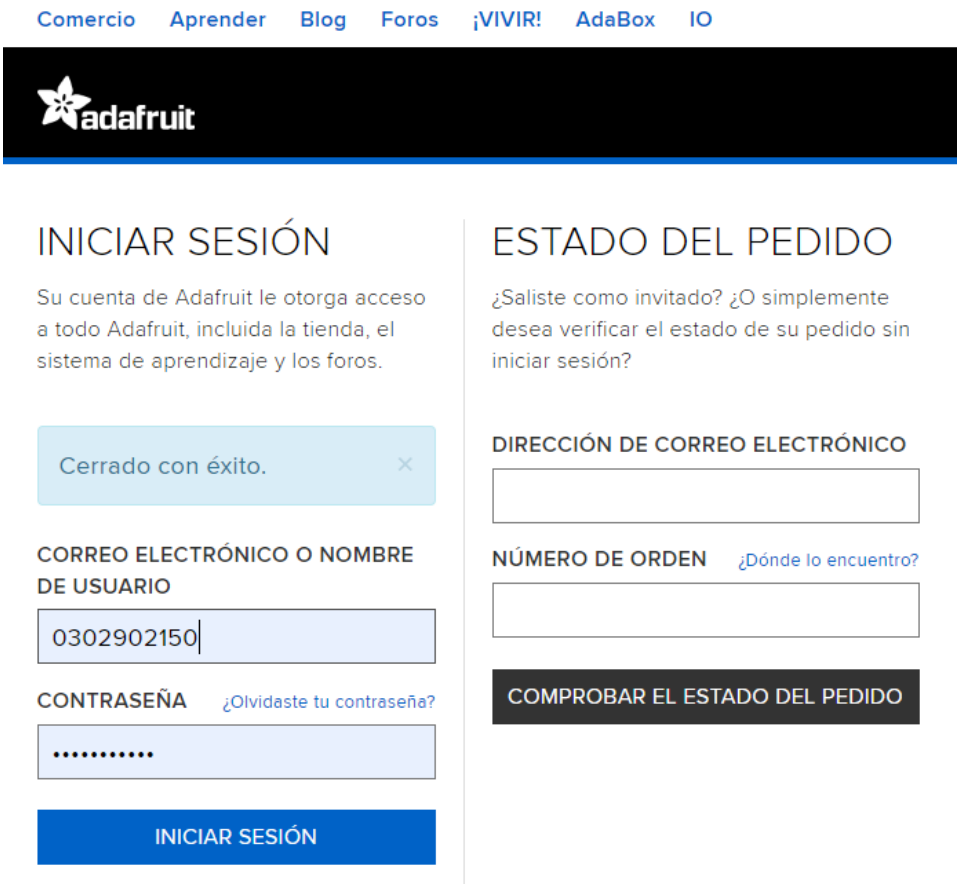


Figura 2.20. Inicio de cuenta de la nube Adafruit.IO

Para realizar un dashboard o un tablero se da clic en la ventana de 'IO' y se crea tanto los feeds como el tablero.

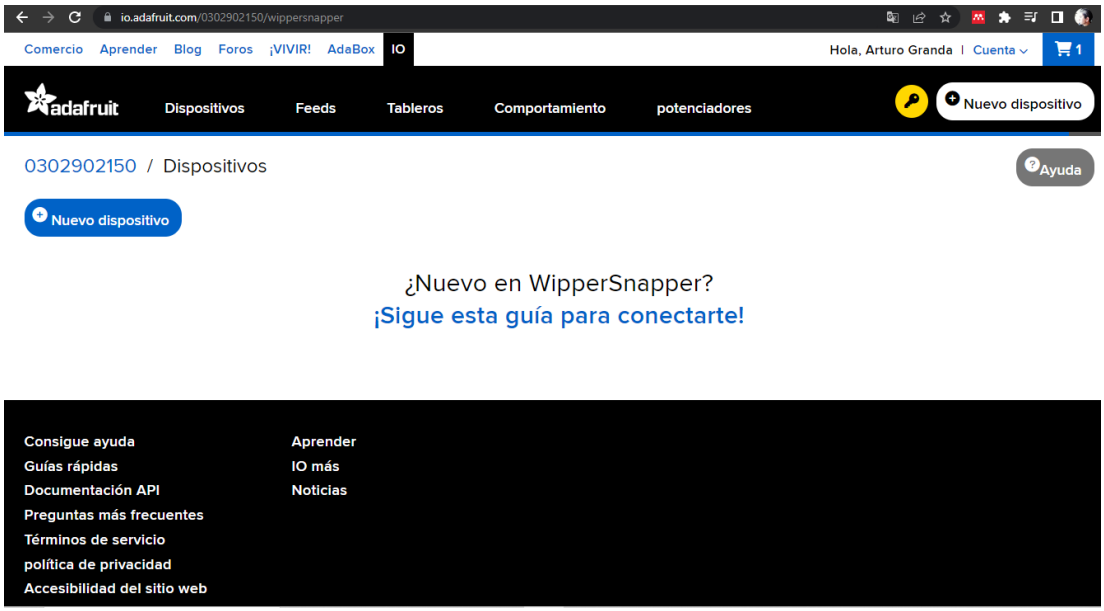


Figura 2.21. Ventana IO para crear feeds y tableros en la nube

Se empieza creando los campos o los feeds que es en donde se almacenará la información enviada a la nube.

Figura 2.22. Creación de feeds en la nube.

Se crean dos Line Chart para visualizar el consumo de la energía de la batería mientras transcurre el tiempo y de la misma manera se visualiza la altitud sobre el nivel del mar en la que se encuentra la moto mientras transcurre el mismo tiempo del consumo de la batería.

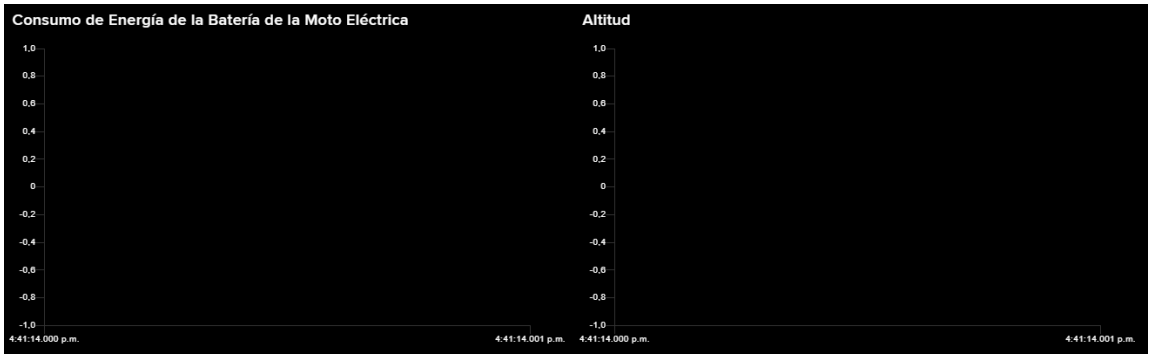


Figura 2.23. Diseño del tablero de la nube (Line Chart).

De la misma manera se crean dos bloques para visualizar la potencia de la batería y un mapa para poder tener un seguimiento de la moto mientras va recorriendo una trayectoria.

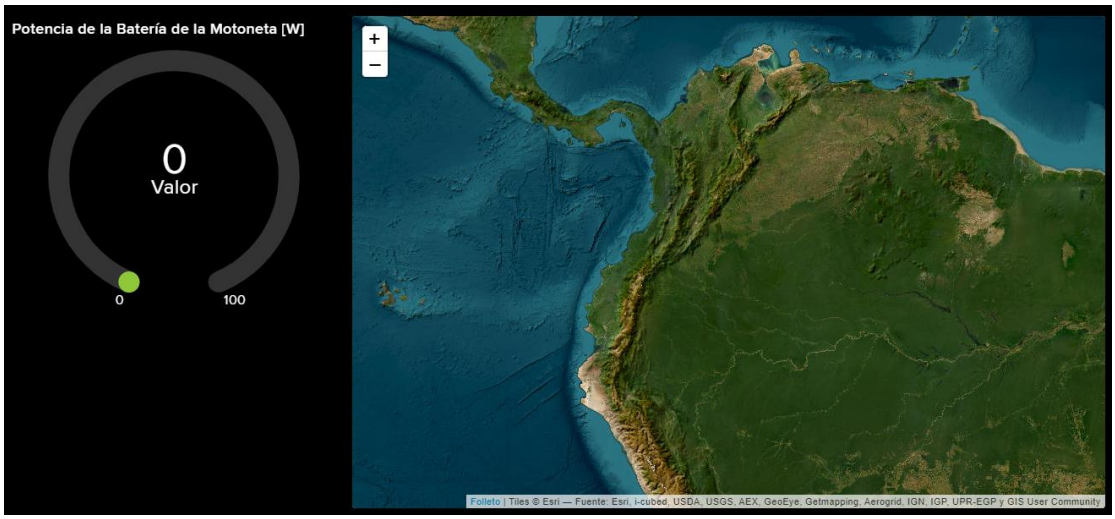


Figura 2.24. Diseño del tablero y mapa de la nube (Gauge and Map).

Al encender el circuito, se obtiene la lectura del consumo de la batería en la parte izquierda del dispositivo, la cual varía dependiendo de la aceleración de la moto. En la parte derecha, se visualiza la altitud del GPS, la cual aumenta en valor a medida que se actualizan los datos de la ubicación del GPS.

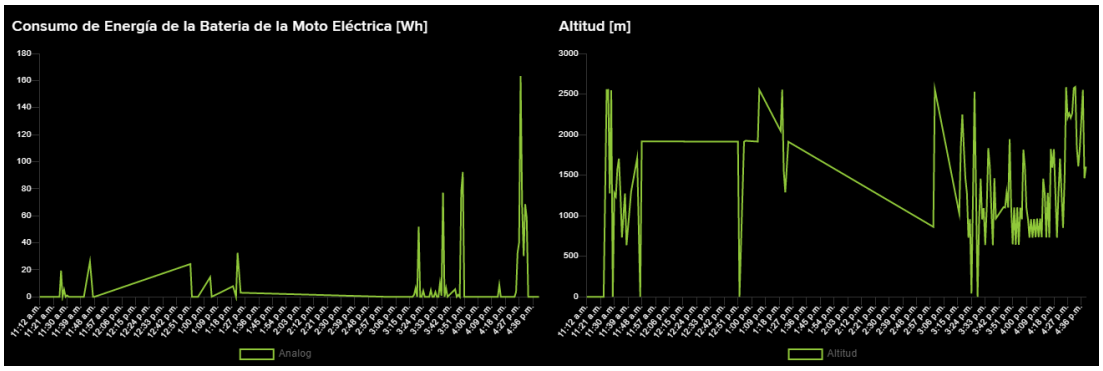


Figura 2.25. Estabilidad de valores en la nube Adafruit.

La señal de ubicación del GPS puede tener un margen de error significativo, incluso al estar estático en un lugar, ya que puede variar entre varios puntos cercanos. En la figura 2.23 se muestra el margen de error al moverse en la moto en una distancia corta, sin superar los 40 metros. Sin embargo, una vez que la señal se estabiliza, se puede ver la ubicación actual en el panel de control de la nube. En este caso, la ubicación mostrada es cerca de la cancha de fútbol de la Universidad Politécnica Salesiana Sede Cuenca.

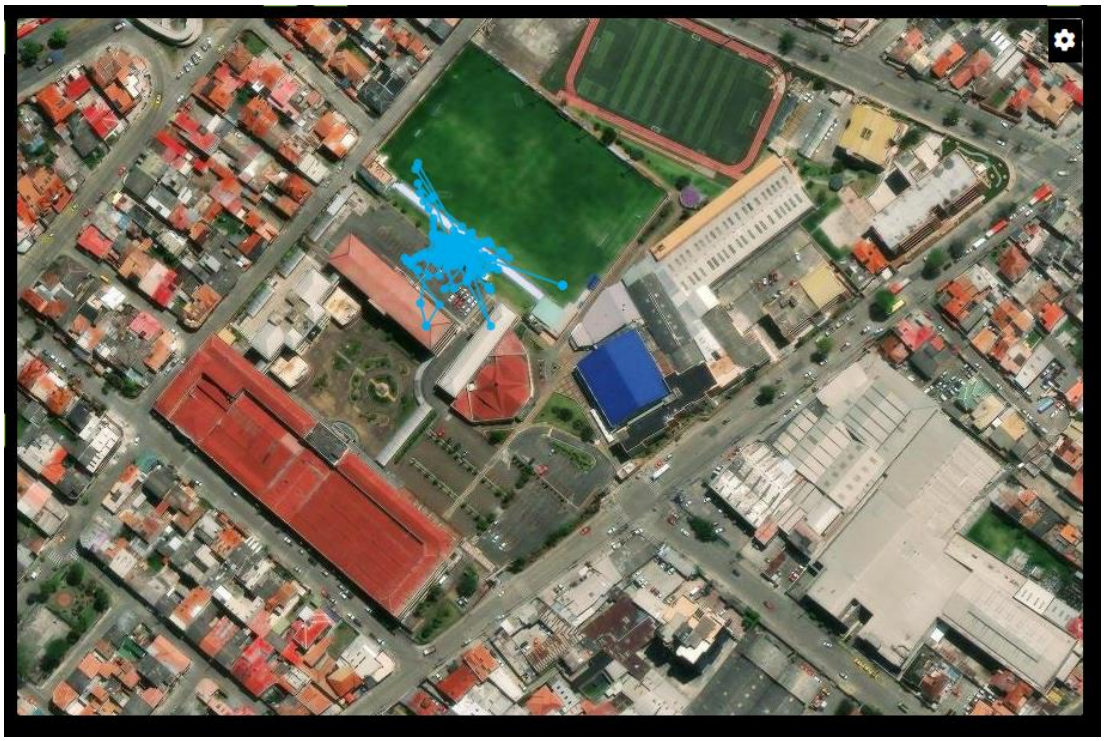


Figura 2.26. Ubicación de la moto en la Universidad Politécnica Salesiana Sede Cuenca.

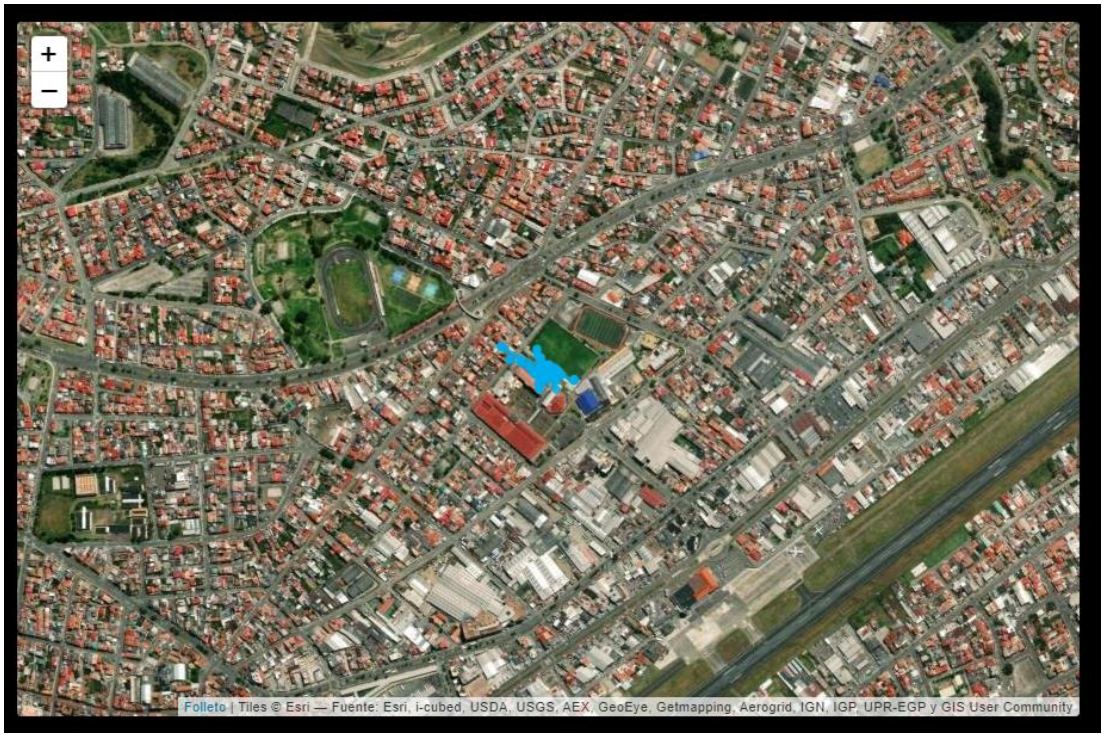


Figura 2.27. Ubicación de la moto con una vista alejada a la figura 2.23.

Al encender el circuito, el módulo GPS puede tardar unos minutos en actualizar su ubicación. Sin embargo, una vez que el GPS está funcionando correctamente, si se

desconecta la fuente de alimentación, el módulo dispone de una batería interna que puede mantener la última ubicación registrada durante unos minutos, aunque no más de una hora.

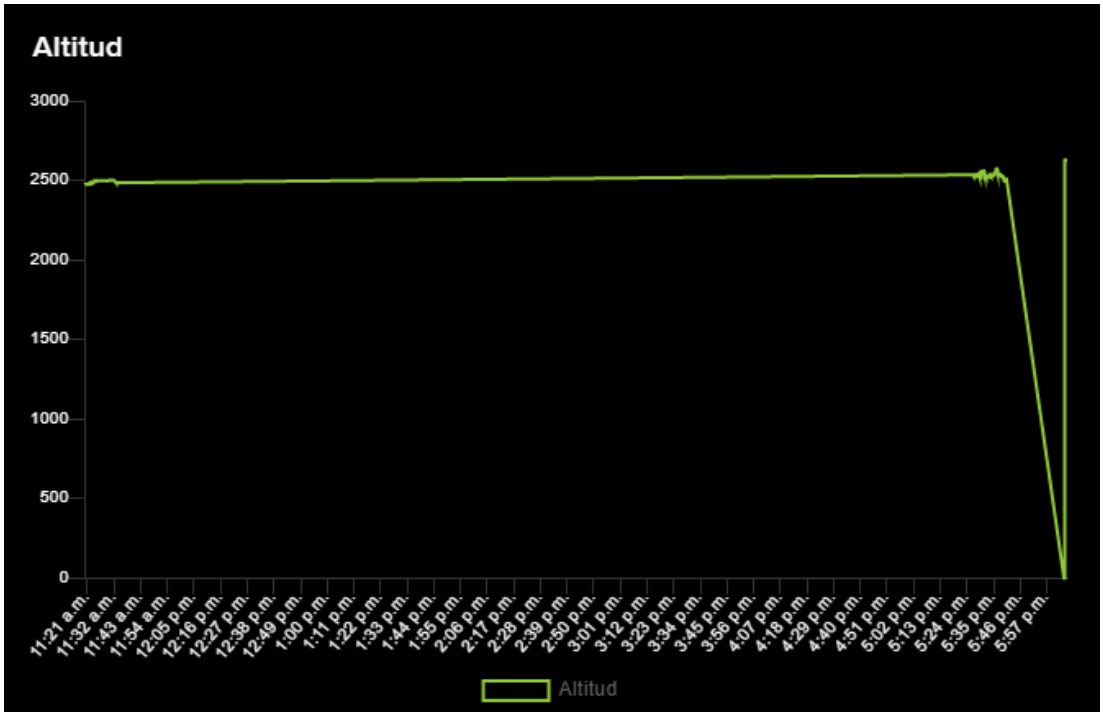


Figura 2.28. Comportamiento de la altitud al desconectar y conectar la fuente al circuito.

Para verificar la altitud del GPS, se puede comparar con datos de mapas topográficos disponibles en línea para la ciudad de Cuenca y otras áreas del país.

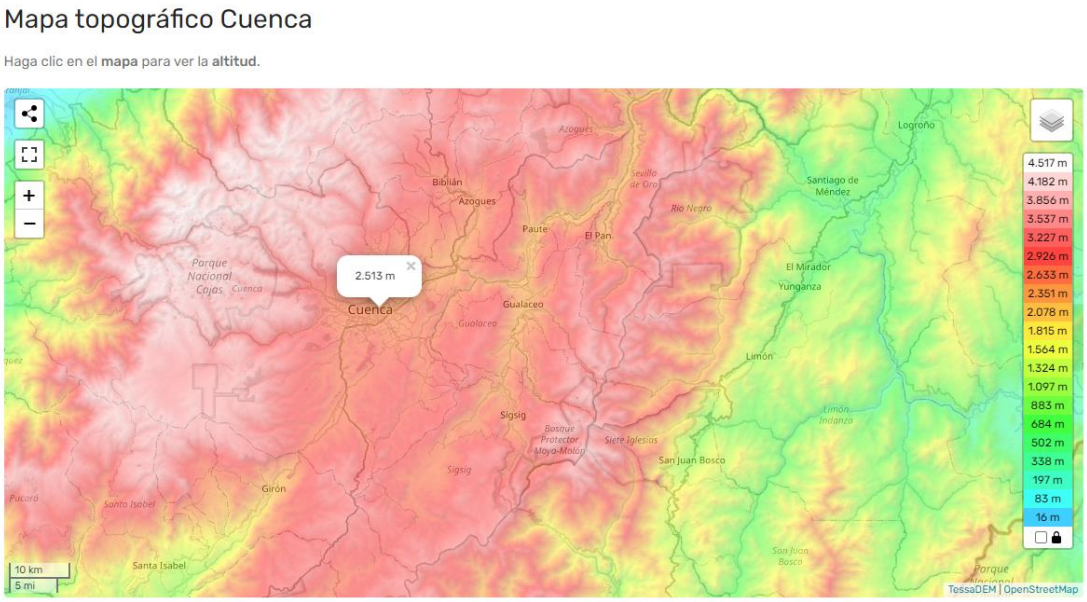


Figura 2.29. Mapa topográfico en Cuenca (altitud 2514 msnm).

En el caso de latitud y longitud se compara los datos que publica el módulo GPS a la nube (latitud: -2.885874, longitud: -78.99009) de la figura 2.28, con el dato del mapa topográfico de Cuenca (latitud: -2.93616, longitud: -79.07118)de la figura 2.29, se puede visualizar que los datos concuerdan, existe una pequeña diferencia, pero es debido a las posiciones actuales.

```
Publishing analog:{"value":507,"lat":-2.885703,"lon":-78.99024,"ele":2554.9}
Publishing altitud:{"value":2554,"lat":-2.885703,"lon":-78.99024,"ele":2554.9}
Publishing analog:{"value":507,"lat":-2.885753,"lon":-78.99022,"ele":2555.5}
Publishing altitud:{"value":2555,"lat":-2.885753,"lon":-78.99022,"ele":2555.5}
Publishing analog:{"value":508,"lat":-2.885854,"lon":-78.99016,"ele":2555.4}
Publishing altitud:{"value":2555,"lat":-2.885854,"lon":-78.99016,"ele":2555.4}
Publishing analog:{"value":508,"lat":-2.885854,"lon":-78.99016,"ele":2555.4}
Publishing altitud:{"value":2555,"lat":-2.885854,"lon":-78.99016,"ele":2555.4}
Publishing analog:{"value":508,"lat":-2.886001,"lon":-78.99008,"ele":2555.4}
Publishing altitud:{"value":2555,"lat":-2.886001,"lon":-78.99008,"ele":2555.4}
Publishing analog:{"value":508,"lat":-2.886001,"lon":-78.99008,"ele":2555.4}
Publishing altitud:{"value":2555,"lat":-2.886001,"lon":-78.99008,"ele":2555.4}
Publishing analog:{"value":508,"lat":-2.886211,"lon":-78.98997,"ele":2555.4}
Publishing altitud:{"value":2555,"lat":-2.886211,"lon":-78.98997,"ele":2555.4}
Publishing analog:{"value":508,"lat":-2.885875,"lon":-78.99009,"ele":2555.3}
Publishing altitud:{"value":2555,"lat":-2.885875,"lon":-78.99009,"ele":2555.3}
```

Figura 2.30. Datos de latitud y longitud del Módulo GPS utilizado.

Figura 2.31. Datos de latitud y longitud del mapa topográfico de Cuenca revisado por Internet.

El GPS incluye una batería interna que permite almacenar su última ubicación cuando el circuito se desconecta. Al encenderlo nuevamente, mostrará la nueva ubicación siempre y cuando no hayan transcurrido más de una hora, ya que la batería se descarga y el GPS se reinicia desde cero.

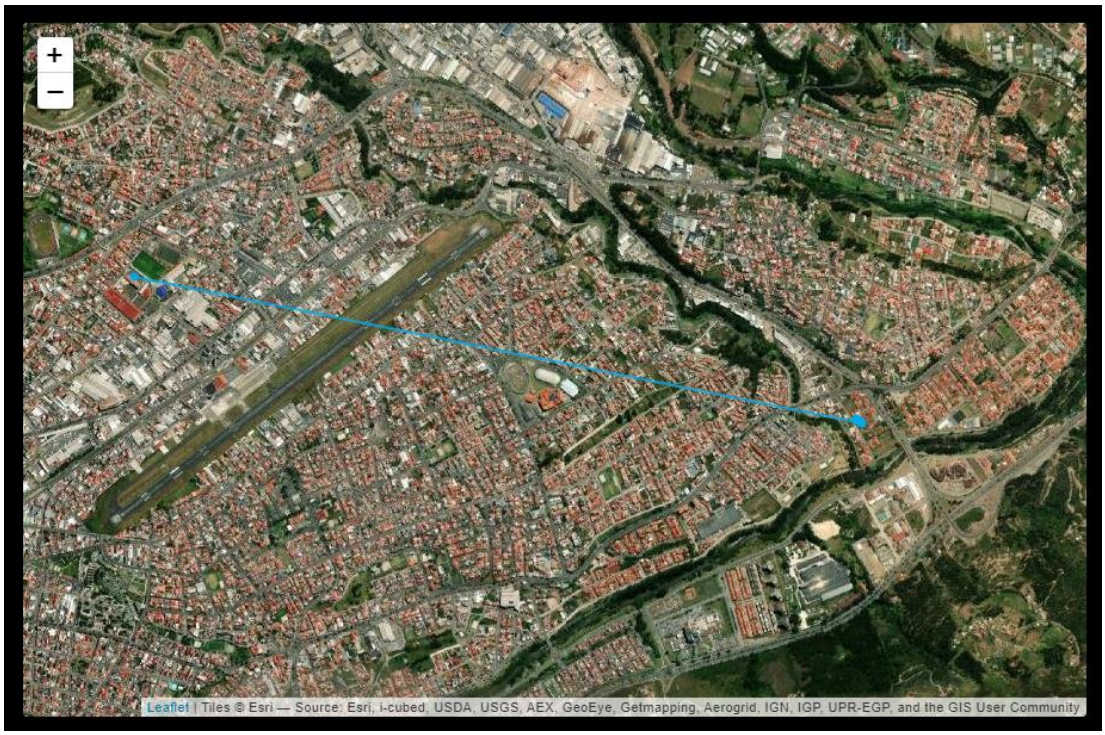


Figura 2.32. Actualización del GPS después de encendido y apagado del circuito.

Figura 2.33. Antena GPS .

● CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

○ 3.1 ANÁLISIS DE LOS PARÁMETROS EN LA NUBE ADAFRUIT IO, DE LA MOTO ELÉCTRICA

Para el análisis de los parámetros de la moto, es importante considerar varios factores, tales como el peso de los pasajeros, el terreno en el que se conduce (subidas o bajadas) y los arranques. En el primer caso, se debe realizar una medición sin carga o pasajeros, manteniendo la moto en un punto fijo. En esta situación, al encender el circuito y la moto, se mide la altitud y la potencia, siendo esta última un valor bajo debido a que no se requiere de una gran fuerza al estar sin carga. La potencia va cambiando al acelerar la moto y esta variación debe ser registrada y analizada para obtener un mejor entendimiento del rendimiento de la moto en diferentes situaciones.

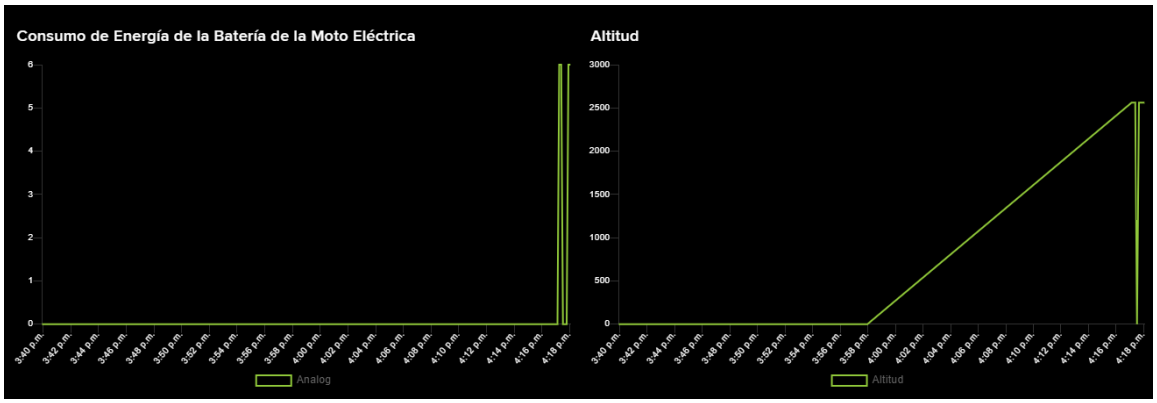


Figura 3.1. Altitud y consumo de energía al arrancar la motoneta sin carga de pasajero.

Al cargar los datos del GPS se estabiliza la altitud en el dashboard de la nube, como se encuentra en una misma altura, el valor de la altitud es constante.

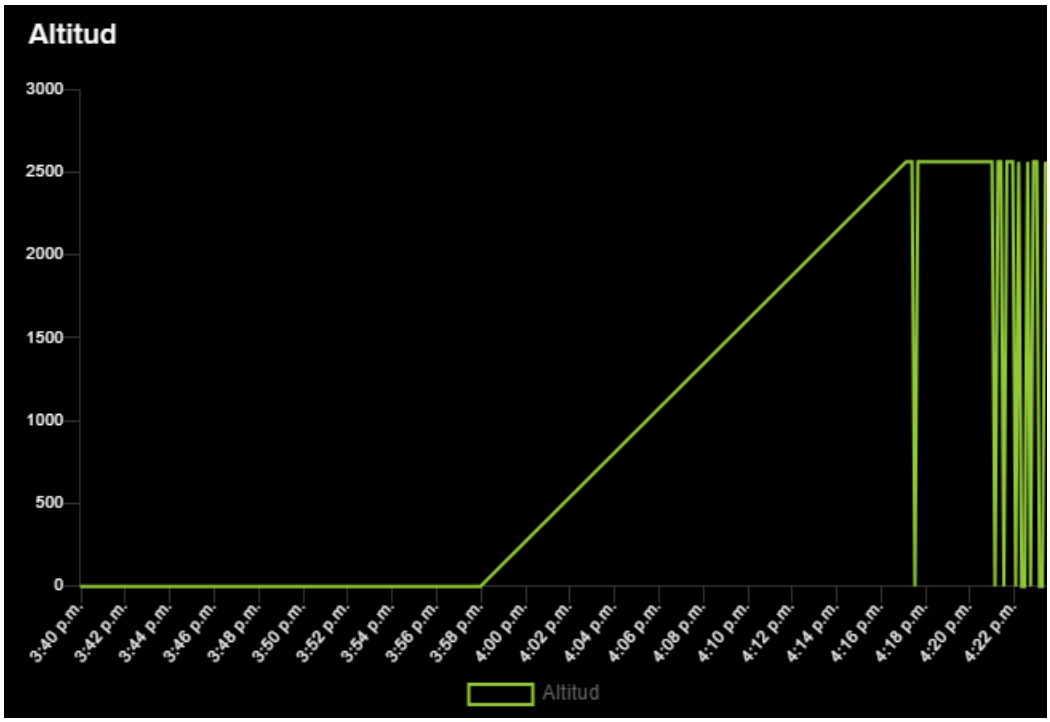


Figura 3.2. Altitud de la ubicación de la motoneta.

La potencia sin carga en la motoneta es baja, solo en el caso de tener arranques bruscos el valor de la potencia tendrá picos altos y en pocos casos supera los 350 vatios como se ha logrado obtener en la figura 3.5.

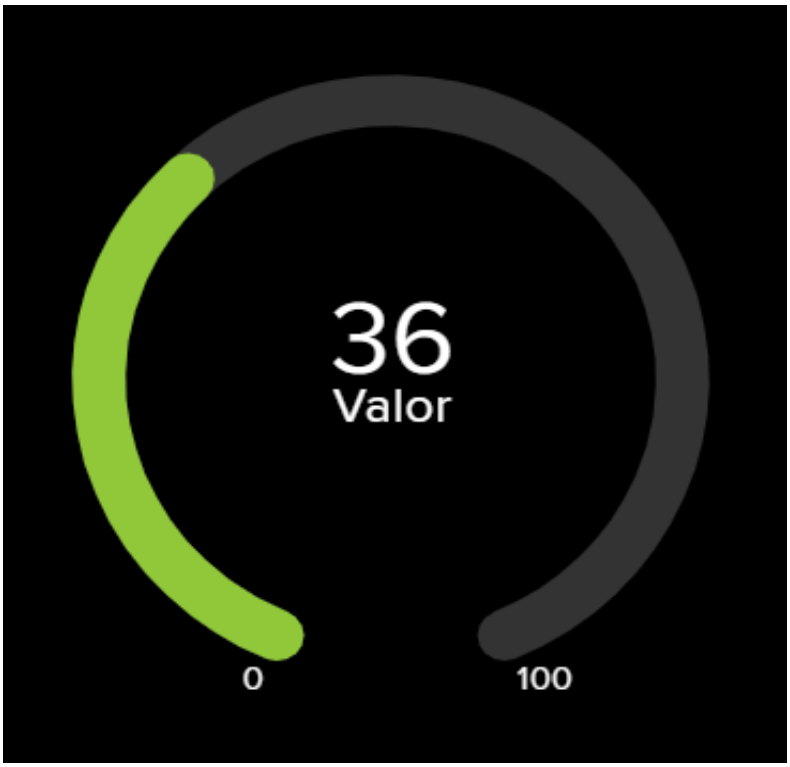


Figura 3.3. Potencia de la batería de la motoneta en arranque sin pasajero.

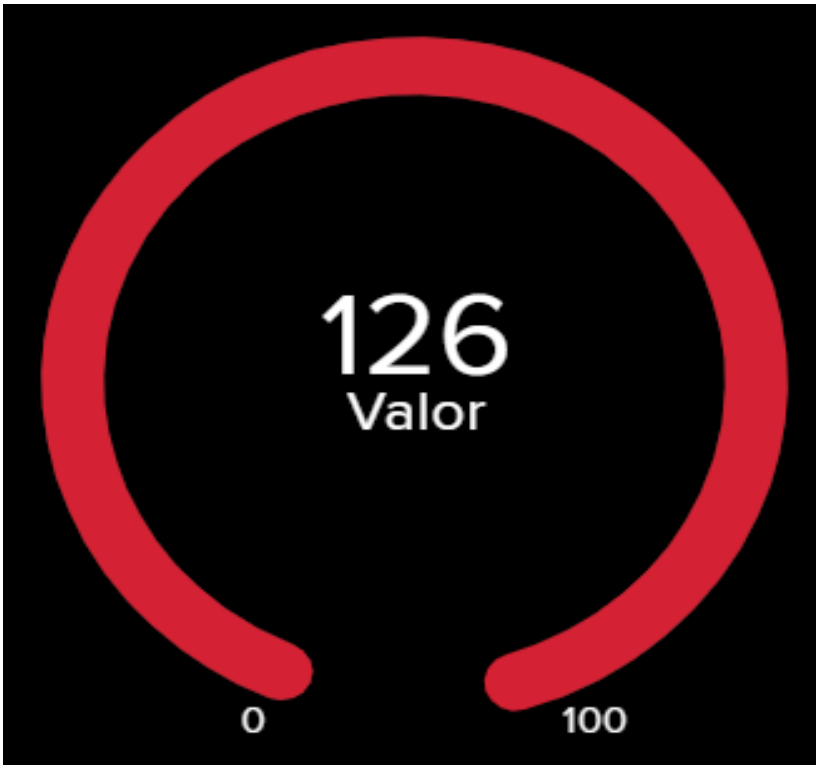


Figura 3.4. Potencia de la batería de la motoneta en arranque brusco sin pasajero.

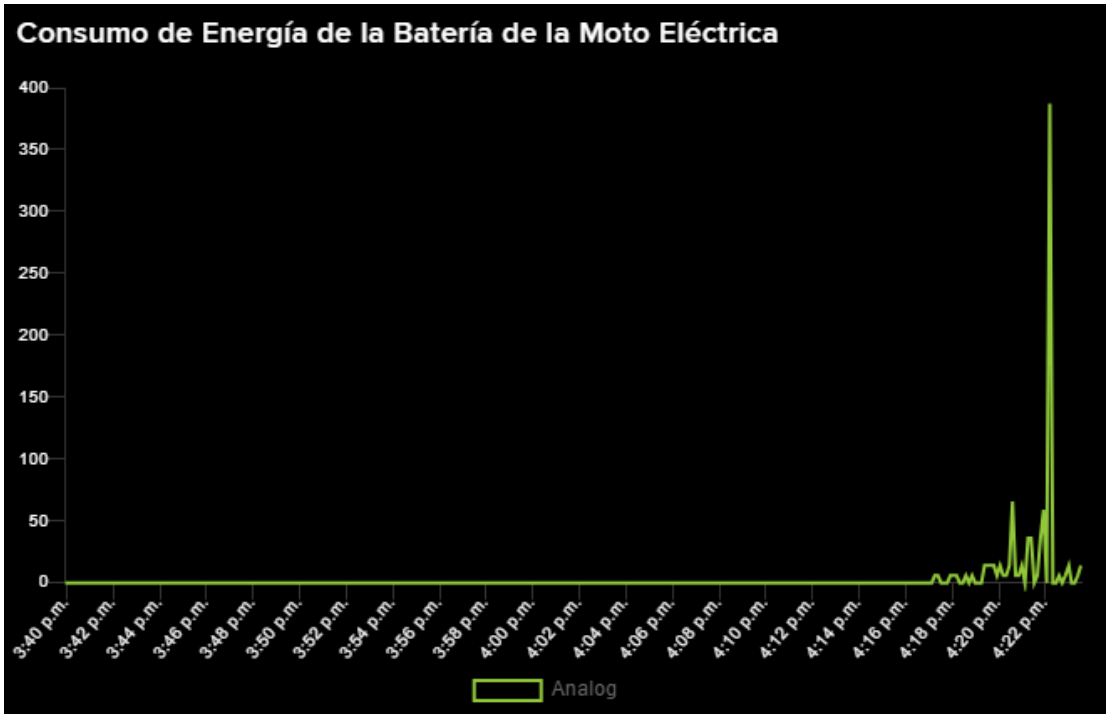


Figura 3.35. Consumo de energía sin carga y con carga consecutivamente.

En la captura de datos del GPS se puede visualizar la ubicación de la motoneta en la Universidad Politécnica Salesiana Sede Cuenca en los estacionamientos de vehículos.

Figura 3.6. Ubicación de la moto desde su punto de partida.

El arranque de la motoneta con peso de 63 kg de una persona adulta va a dar de resultado una mayor potencia por tener una mayor fuerza de arranque tanto en reposo como en velocidad.

Figura 3.7. Potencia de arranque de la moto con carga.

Se realizan las pruebas con el peso del pasajero en la motoneta durantes cortas trayectorias dentro de la universidad.

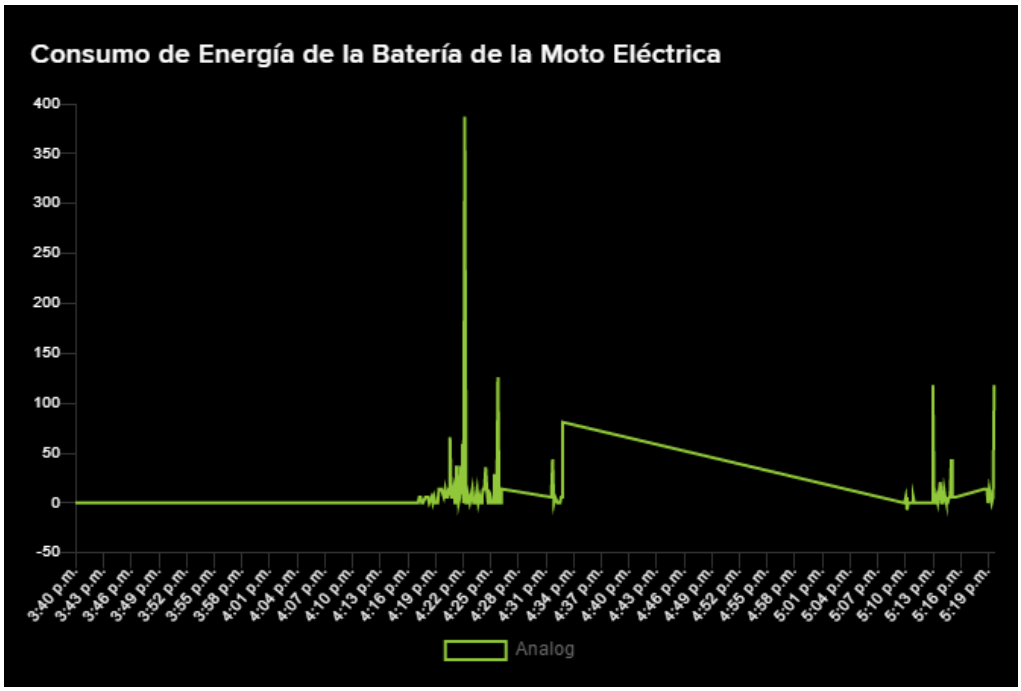


Figura 3.8. Gráfica del consumo de batería de la moto en cortos recorridos.

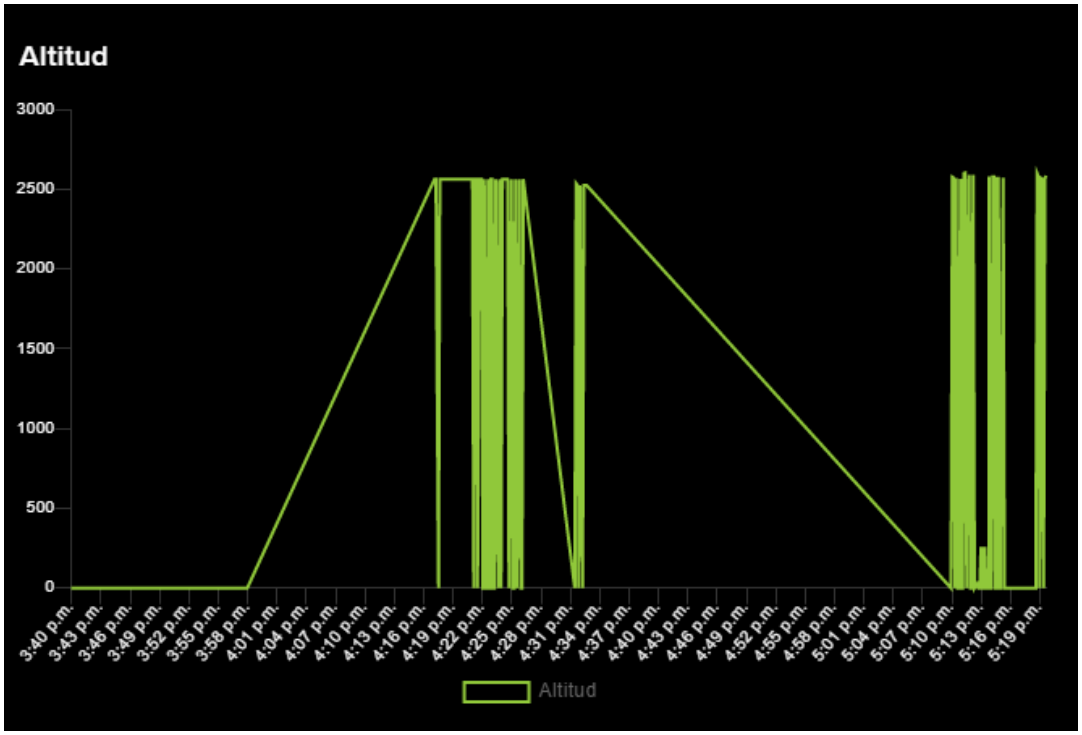


Figura 3.9. Gráfica de la altitud de la ubicación en cortos recorridos.

Figura 3.10. Potencia en el arranque en rompe velocidad.

Figura 3.11. Potencia de arranque brusco con pasajero.

Para obtener un análisis en un recorrido más extenso se agrega un bloque Stream (arroyo) para visualizar en la nube los datos que llegan del Módulo al WiFi y poder comparar con los bloques del dashboard tanto en potencia como en altitud, latitud y longitud.

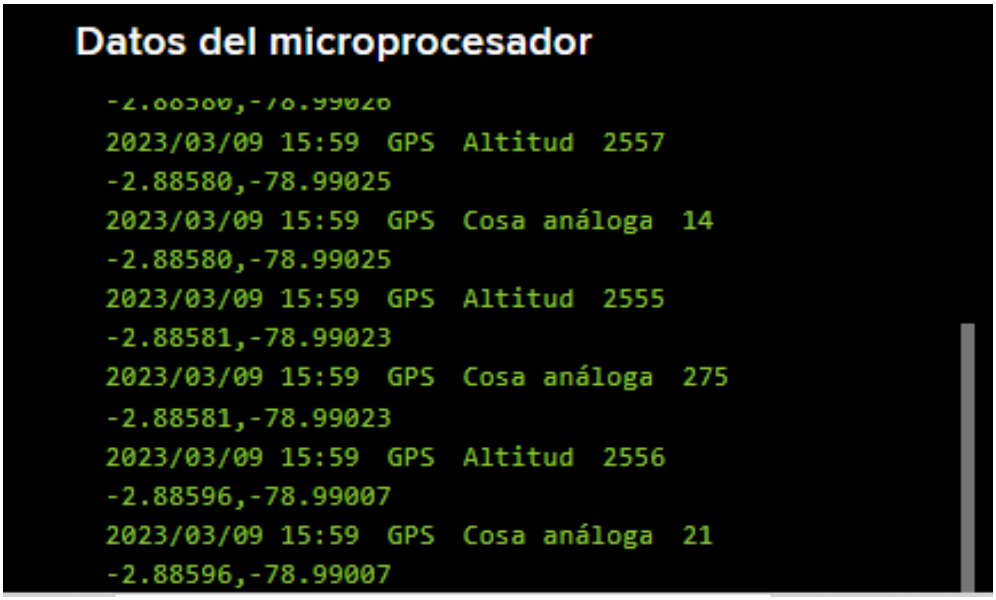


Figura 3.12. Parámetros recibidos en la nube Adafruit

Se recorre una pequeña trayectoria en el estacionamiento de la universidad.

Figura 3.13. Dashboard del recorrido en el estacionamiento de la parte trasera de la universidad.

Se recorre una trayectoria y los puntos de ubicación del GPS quedan registrados en el mapa de la nube Adafruit.

Figura 3.14. Rastreo de la motoneta en el primer recorrido.

Figura 3.15. Rastreo de la motoneta en un segundo recorrido.

Figura 3.16. Rastreo de la motoneta en un tercer recorrido.

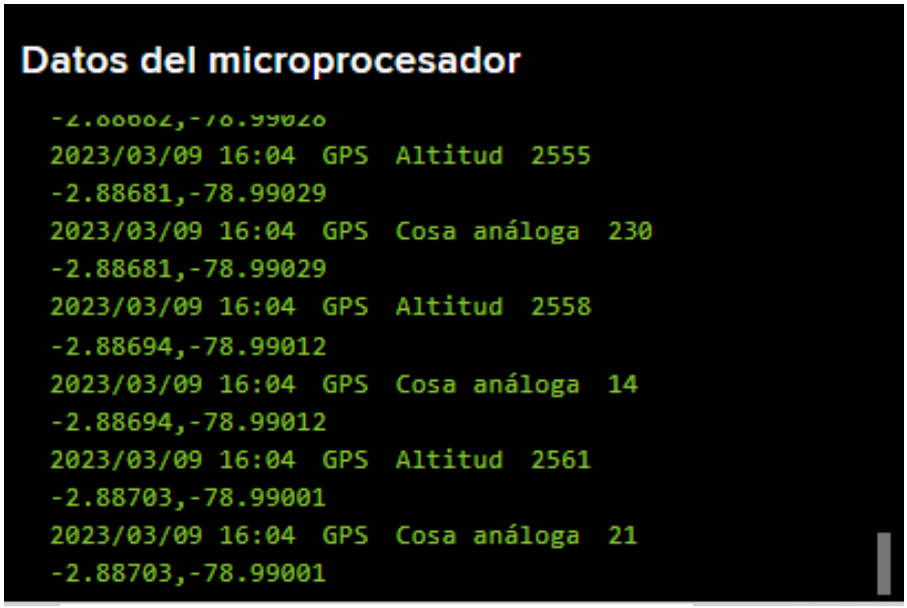
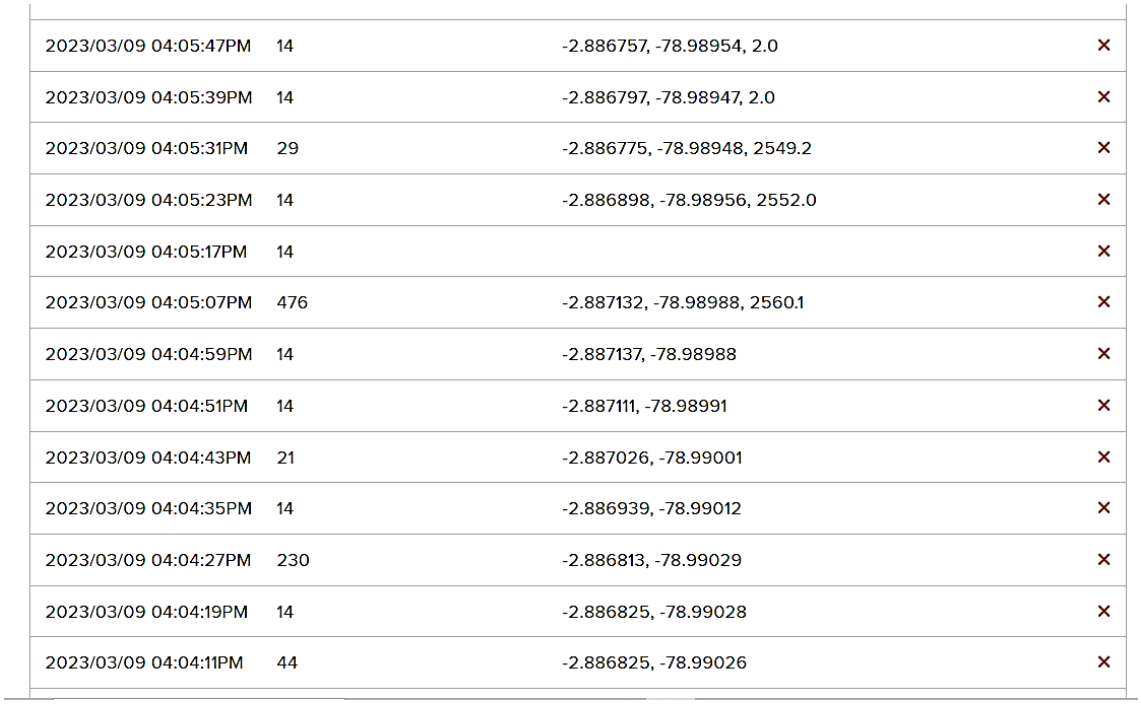


Figura 3.17. Muestra de datos en recorrido de la figura 3.16.

La nube de Adafruit ofrece la ventaja de almacenar datos durante períodos personalizables, que pueden ir desde una hora hasta un mes, lo que permite una mayor flexibilidad en la gestión de los datos. Además, si se requiere un almacenamiento de datos por un tiempo más prolongado, se puede adquirir una licencia de la nube para obtener aún más opciones y beneficios.

En cuanto a la adquisición de datos de la motoneta, se almacenan los parámetros de los recorridos realizados durante 24 horas, con cada uno de estos

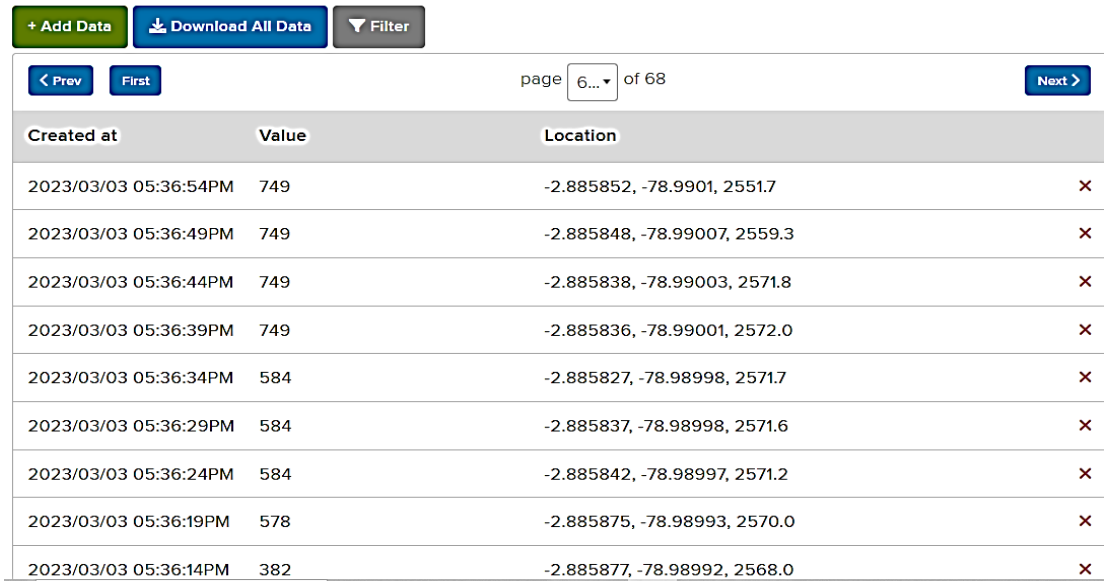
parámetros siendo guardado en la nube cada ocho segundos, tal como se puede apreciar en la siguiente figura.



2023/03/09 04:05:47PM	14	-2.886757, -78.98954, 2.0	×
2023/03/09 04:05:39PM	14	-2.886797, -78.98947, 2.0	×
2023/03/09 04:05:31PM	29	-2.886775, -78.98948, 2549.2	×
2023/03/09 04:05:23PM	14	-2.886898, -78.98956, 2552.0	×
2023/03/09 04:05:17PM	14		×
2023/03/09 04:05:07PM	476	-2.887132, -78.98988, 2560.1	×
2023/03/09 04:04:59PM	14	-2.887137, -78.98988	×
2023/03/09 04:04:51PM	14	-2.887111, -78.98991	×
2023/03/09 04:04:43PM	21	-2.887026, -78.99001	×
2023/03/09 04:04:35PM	14	-2.886939, -78.99012	×
2023/03/09 04:04:27PM	230	-2.886813, -78.99029	×
2023/03/09 04:04:19PM	14	-2.886825, -78.99028	×
2023/03/09 04:04:11PM	44	-2.886825, -78.99026	×

Figura 3.18. Parámetro del consumo de la batería y de ubicación de la moto.

Se puede observar que en los distintos recorridos realizados en la motoneta se han almacenado un total de 68 páginas de parámetros.



Created at	Value	Location	
2023/03/03 05:36:54PM	749	-2.885852, -78.9901, 2551.7	×
2023/03/03 05:36:49PM	749	-2.885848, -78.99007, 2559.3	×
2023/03/03 05:36:44PM	749	-2.885838, -78.99003, 2571.8	×
2023/03/03 05:36:39PM	749	-2.885836, -78.99001, 2572.0	×
2023/03/03 05:36:34PM	584	-2.885827, -78.98998, 2571.7	×
2023/03/03 05:36:29PM	584	-2.885837, -78.98998, 2571.6	×
2023/03/03 05:36:24PM	584	-2.885842, -78.98997, 2571.2	×
2023/03/03 05:36:19PM	578	-2.885875, -78.98993, 2570.0	×
2023/03/03 05:36:14PM	382	-2.885877, -78.98992, 2568.0	×

Figura 3.19. Registro de páginas almacenadas en la nube.

De igual manera se puede descargar todas las páginas.

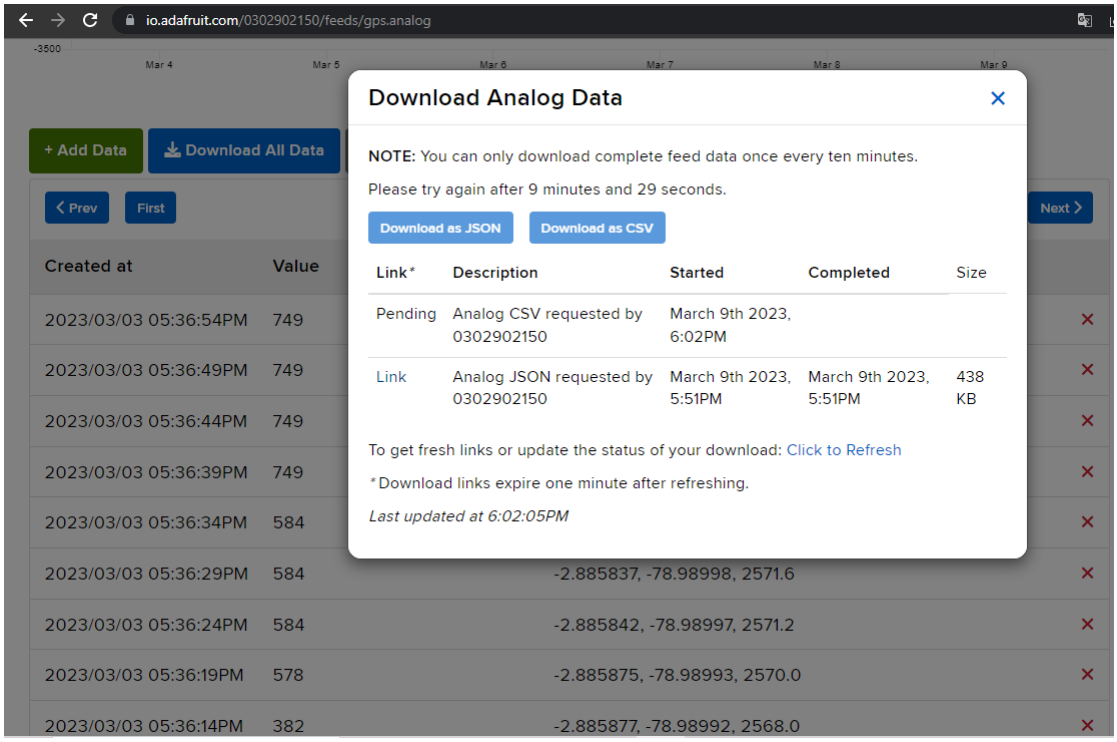


Figura 3.20. Descarga de páginas de parámetros almacenados.

Asimismo, se ha registrado el parámetro de altitud del GPS junto con la latitud y longitud, generando un total de 64 páginas almacenadas en la nube de Adafruit. Además, se cuenta con la posibilidad de descargar toda la información almacenada en la nube.

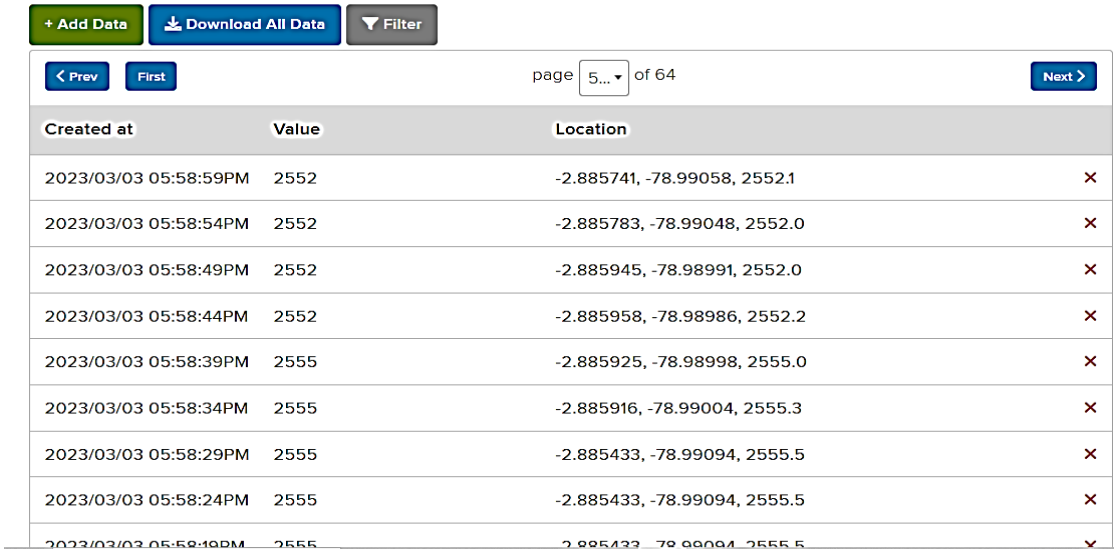


Figura 3.21. Parámetros de la altitud de la ubicación de la motoneta.

Además de poder visualizar los parámetros de la nube Adafruit desde una PC, también es posible hacerlo mediante un dispositivo móvil. Con solo acceder al enlace de la página de la nube, se puede ver la información desde cualquier dispositivo con conexión a internet.

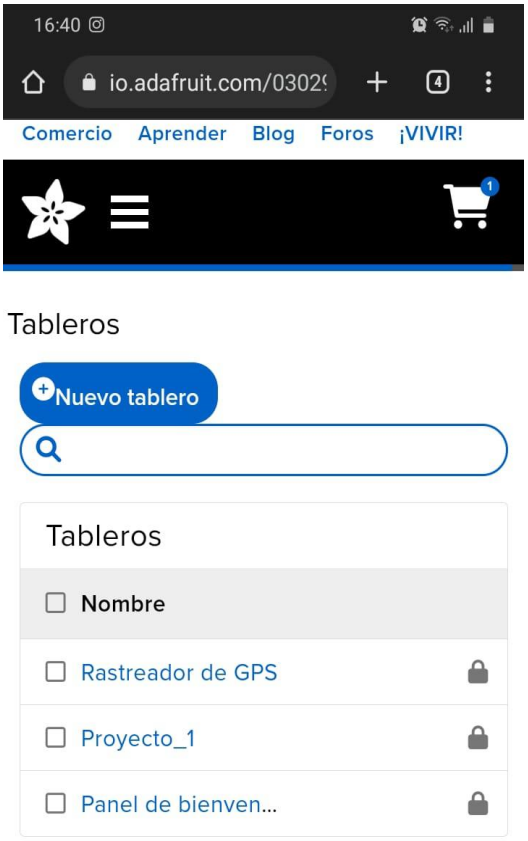


Figura 3.22. Ingreso del tablero de la nube mediante un dispositivo móvil.

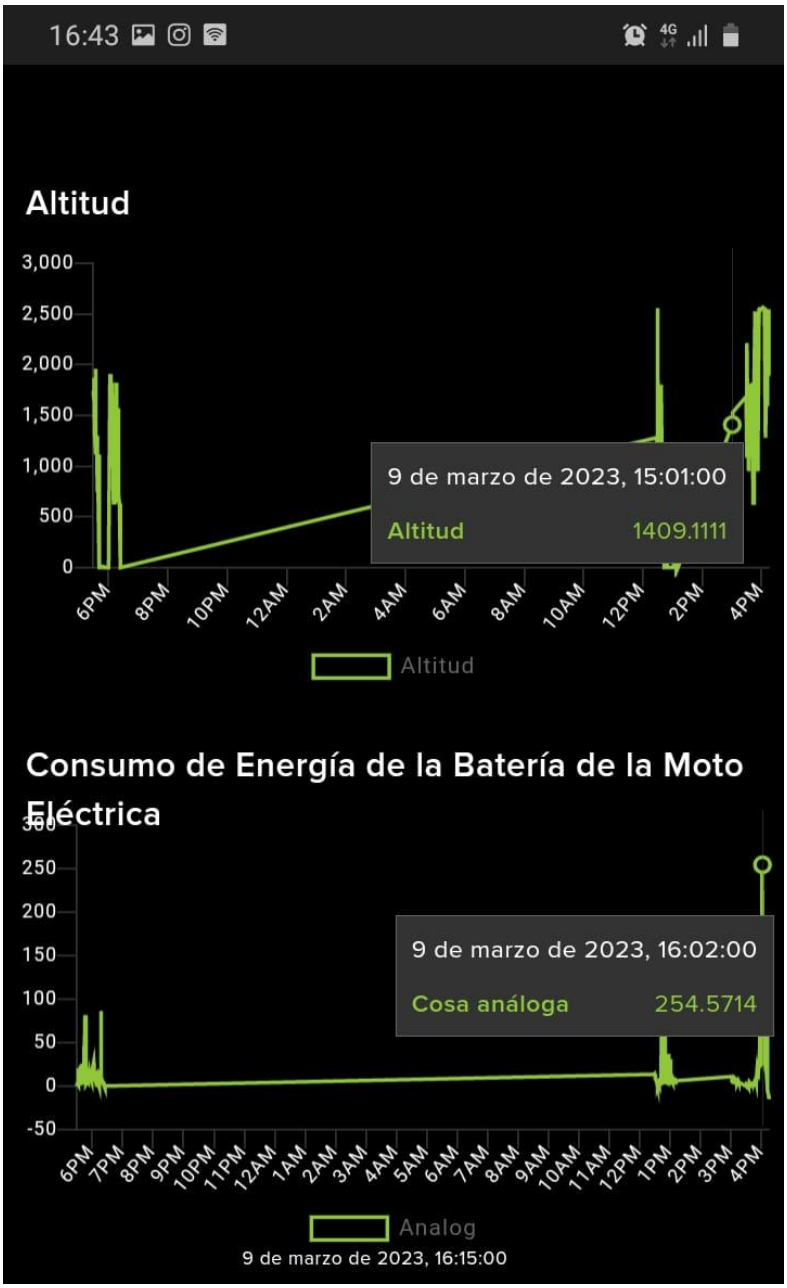


Figura 3.23. Plataforma Adafruit desde un dispositivo móvil.

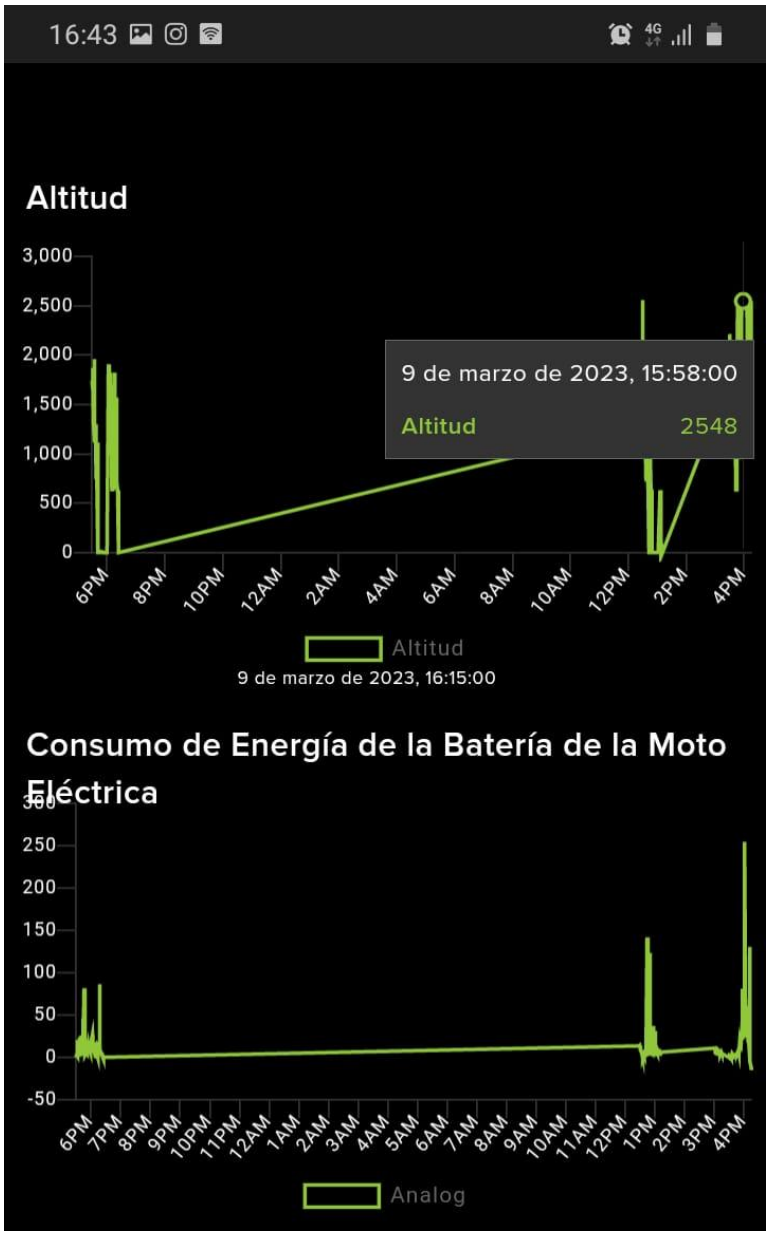


Figura 3.24. Plataforma Adafruit con vista de registro de altitud.

Figura 3.25. Plataforma Adafruit con vista del mapa y potencia de la moto

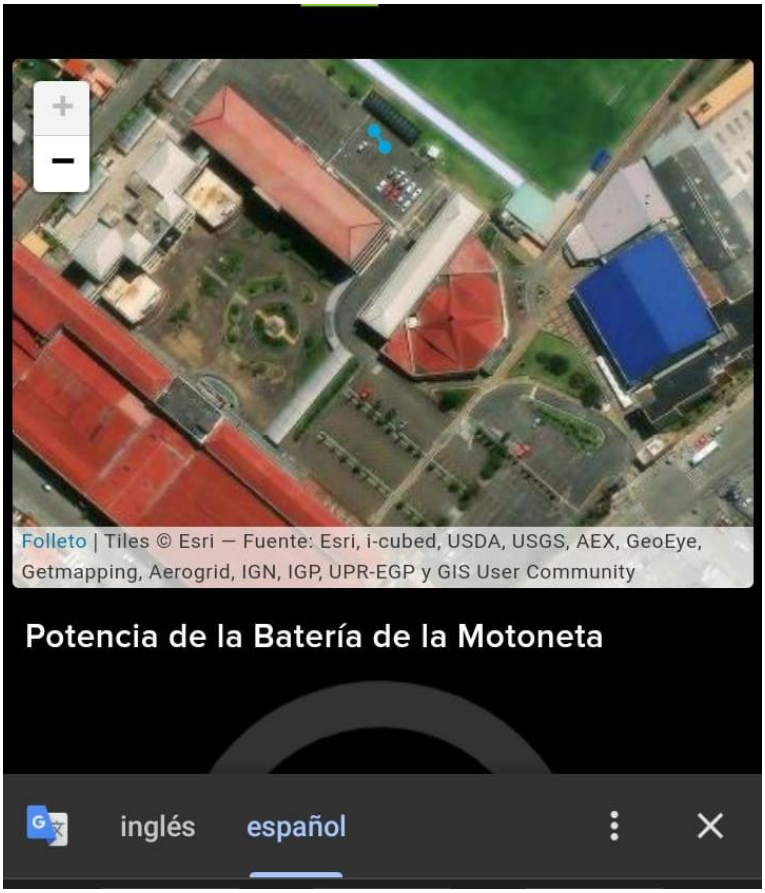


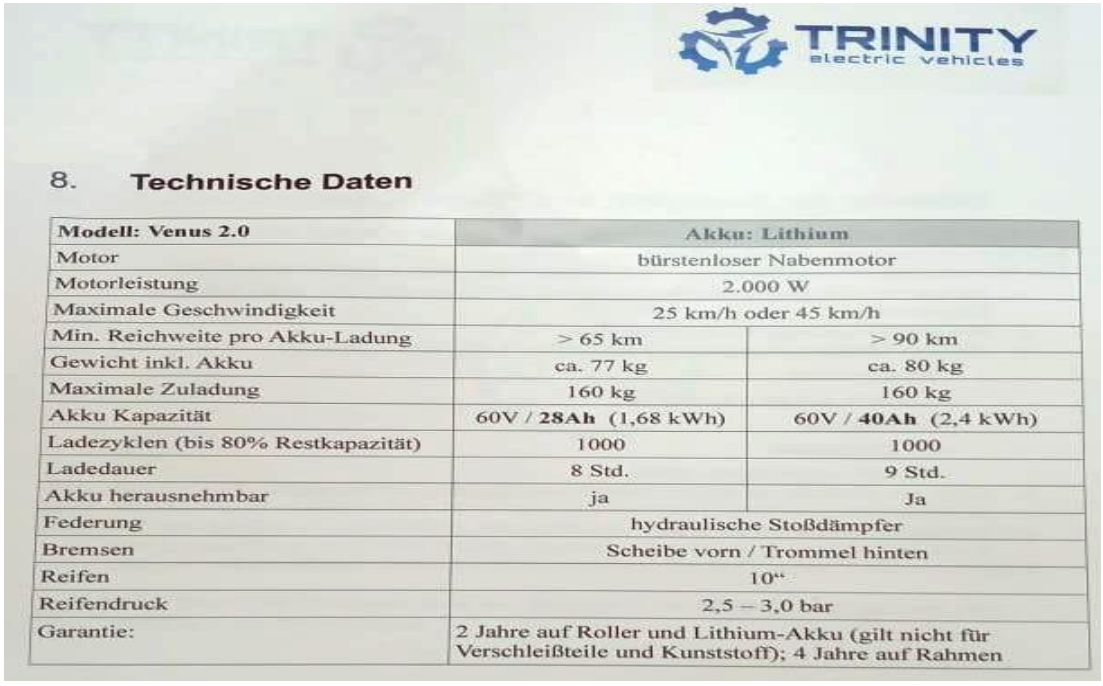
Figura 3.26. Plataforma Adafruit con ubicación del GPS

Figura 3.27. Plataforma Adafruit con ubicación del GPS en un corto recorrido.

.

La motoneta tiene un límite máximo de carga, el cual, si se supera, puede provocar que la potencia exceda su valor nominal y, por lo tanto, que la corriente

sobrepase su límite máximo. Esta información se puede encontrar en los datos técnicos del vehículo.

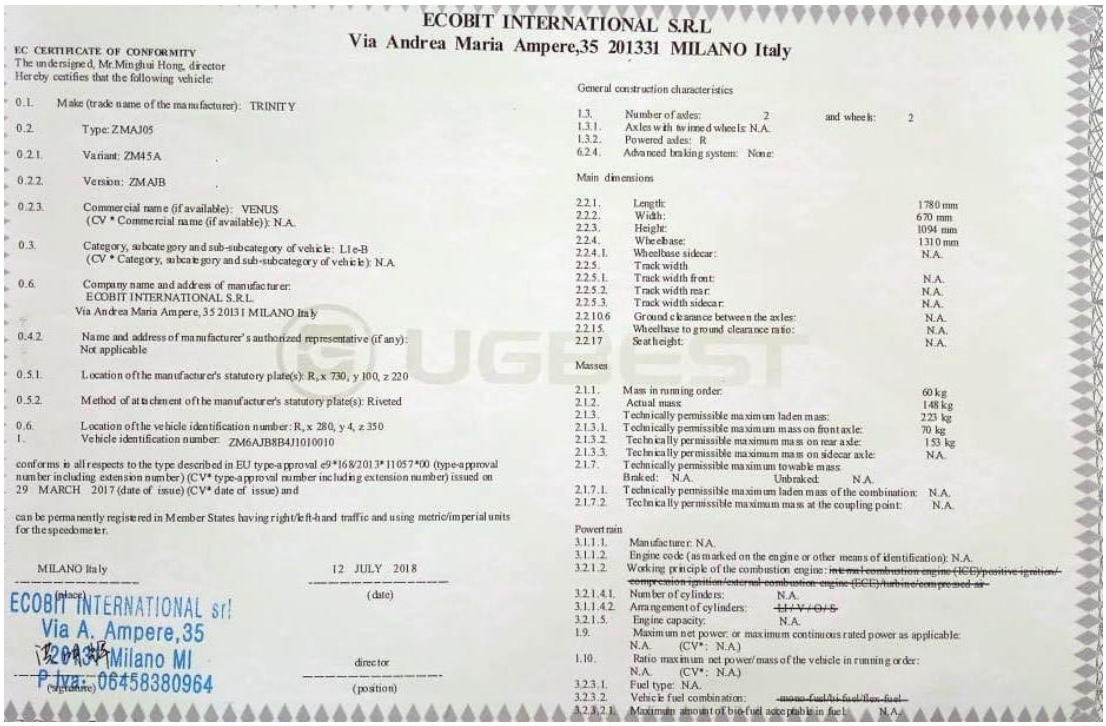


TRINITY
electric vehicles

8. **Technische Daten**

Modell: Venus 2.0	**Akku: Lithium**	
Motor	bürstenloser Nabenmotor	
Motorleistung	2.000 W	
Maximale Geschwindigkeit	25 km/h oder 45 km/h	
Min. Reichweite pro Akku-Ladung	> 65 km	> 90 km
Gewicht inkl. Akku	ca. 77 kg	ca. 80 kg
Maximale Zuladung	160 kg	160 kg
Akku Kapazität	60V / **28Ah** (1,68 kWh)	60V / **40Ah** (2,4 kWh)
Ladezyklen (bis 80% Restkapazität)	1000	1000
Ladedauer	8 Std.	9 Std.
Akku herausnehmbar	ja	Ja
Federung	hydraulische Stoßdämpfer	
Bremsen	Scheibe vorn / Trommel hinten	
Reifen	10“	
Reifendruck	2,5 – 3,0 bar	
Garantie:	2 Jahre auf Roller und Lithium-Akku (gilt nicht für Verschleißteile und Kunststoff); 4 Jahre auf Rahmen	

Figura 3.28. Datos técnicos de la moto eléctrica Trinity.



ECOBIT INTERNATIONAL S.R.L
Via Andrea Maria Ampere,35 201331 MILANO Italy

EC CERTIFICATE OF CONFORMITY
The undersigned, Mr.Minghui Hong, director
Hereby certifies that the following vehicle:

0.1. Make (trade name of the manufacturer): TRINITY
0.2. Type: ZMAJ05
0.2.1. Variant: ZM45A
0.2.2. Version: ZMAJB
0.2.3. Commercial name (if available): VENUS
(CV * Commercial name (if available)): N.A.
0.3. Category, subcategory and sub-subcategory of vehicle: L1e-B
(CV * Category, subcategory and sub-subcategory of vehicle): N.A.
0.6. Company name and address of manufacturer:
ECOBIT INTERNATIONAL S.R.L.
Via Andrea Maria Ampere, 35 20131 MILANO Italy
0.4.2. Name and address of manufacturer's authorized representative (if any):
Not applicable
0.5.1. Location of the manufacturer's statutory plate(s): R, x 730, y 100, z 220
0.5.2. Method of attachment of the manufacturer's statutory plate(s): Riveted
0.6. Location of the vehicle identification number: R, x 280, y 4, z 350
1. Vehicle identification number: ZM6AJB8B4J1010010

conforms in all respects to the type described in EU type-approval e9*168/2013*11057*00 (type-approval number including extension number) (CV* type-approval number including extension number) issued on 29 MARCH 2017 (date of issue) (CV* date of issue) and

can be permanently registered in Member States having right/left-hand traffic and using metric/imperial units for the speedometer.

MILANO Italy (place) — 12 JULY 2018 (date)

ECOBIT INTERNATIONAL srl
Via A. Ampere,35
20131 Milano MI
P.Iva: 06458380964

(signature) — director (position)

General construction characteristics

1.3. Number of axles: 2 and wheels: 2
1.3.1. Axles with twinned wheels: N.A.
1.3.2. Powered axles: R
6.2.4. Advanced braking system: None

Main dimensions

2.2.1. Length: 1780 mm
2.2.2. Width: 670 mm
2.2.3. Height: 1094 mm
2.2.4. Wheelbase: 1310 mm
2.2.4.1. Wheelbase sidecar: N.A.
2.2.5. Track width
2.2.5.1. Track width front: N.A.
2.2.5.2. Track width rear: N.A.
2.2.5.3. Track width sidecar: N.A.
2.2.10.6 Ground clearance between the axles: N.A.
2.2.15. Wheelbase to ground clearance ratio: N.A.
2.2.17 Seat height: N.A.

Masses

2.1.1. Mass in running order: 60 kg
2.1.2. Actual mass: 148 kg
2.1.3. Technically permissible maximum laden mass: 223 kg
2.1.3.1. Technically permissible maximum mass on front axle: 70 kg
2.1.3.2. Technically permissible maximum mass on rear axle: 153 kg
2.1.3.3. Technically permissible maximum mass on sidecar axle: N.A.
2.1.7. Technically permissible maximum towable mass:
Braked: N.A. Unbraked: N.A.
2.1.7.1. Technically permissible maximum laden mass of the combination: N.A.
2.1.7.2. Technically permissible maximum mass at the coupling point: N.A.

Powertrain

3.1.1.1. Manufacturer: N.A.
3.1.1.2. Engine code (as marked on the engine or other means of identification): N.A.
3.2.1.2. Working principle of the combustion engine: ~~internal combustion engine (ICE)/positive ignition/compression ignition/external combustion engine (ECE)/turbine/compressed air~~
3.2.1.4.1. Number of cylinders: N.A.
3.1.1.4.2. Arrangement of cylinders: ~~I/I-V/O/S~~
3.2.1.5. Engine capacity: N.A.
1.9. Maximum net power: or maximum continuous rated power as applicable:
N.A. (CV*: N.A.)
1.10. Ratio maximum net power/mass of the vehicle in running order:
N.A. (CV*: N.A.)
3.2.3.1. Fuel type: N.A.
3.2.3.2. Vehicle fuel combination: ~~mono-fuel/bi-fuel/flex-fuel~~
3.2.3.2.1. Maximum amount of bio-fuel acceptable in fuel: N.A.

Figura 3.29. Datos técnicos de velocidad, masa, entre otras, de la moto eléctrica Trinity.

Para asegurarse de que la corriente eléctrica de la moto no supere la máxima especificada en el catálogo, es necesario calcular el peso máximo de la carga y el

pasajero combinados. Este valor se obtiene restando la carga útil máxima, que incluye el peso de ambos, del límite técnico de carga admisible para la moto.

Tabla 3.1 Carga máxima de carga para la moto eléctrica.

	Valores
Carga útil máxima	160 kg
Masa de la moto	75 kg
Total de peso	**85 kg**

Se llevó a cabo una prueba con dos pasajeros cuyo peso combinado supera los 85 kg. La corriente eléctrica máxima de la moto, limitada a 28A en su velocidad máxima, resultó insuficiente para soportar la carga, lo que generó una sobrecarga eléctrica.

El sensor de corriente de la moto tiene una capacidad máxima de 50kg, por lo que se instaló un fusible de 35A para proteger el circuito. Sin embargo, al superar este límite, el fusible se quemó y cortó el suministro eléctrico, preservando la integridad del sensor y del circuito

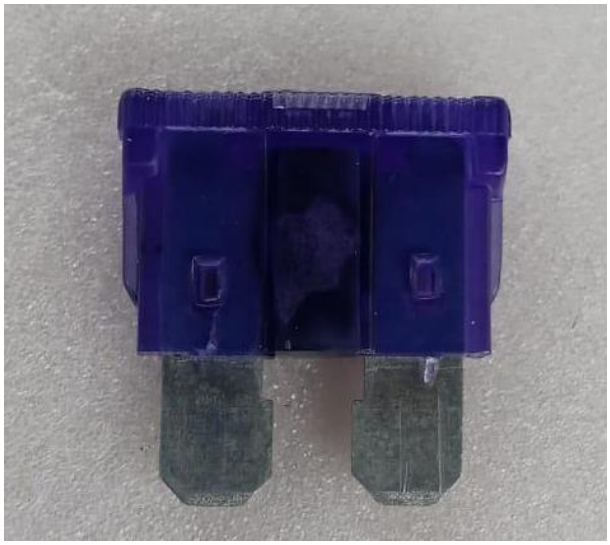


Figura 3.30. Fusible quemado

3.2 Análisis de los parámetros en la Tarjeta de memoria microSD

Con el objetivo de mejorar el rendimiento del proyecto, se ha integrado una tarjeta microSD que almacena los parámetros de consumo de energía de la moto. En caso de que el usuario no tenga acceso a internet para conectar al módulo Wifi, este circuito cuenta con un lector de microSD conectado al Arduino que permite guardar una reserva de los parámetros de la moto.

Para recopilar datos, se realizaron pruebas con la moto en reposo y se registraron los datos del arranque en un archivo de texto en la microSD. Posteriormente, se transfirieron los datos a una hoja de Excel para crear una gráfica de dispersión y visualizar el comportamiento del arranque de la moto.

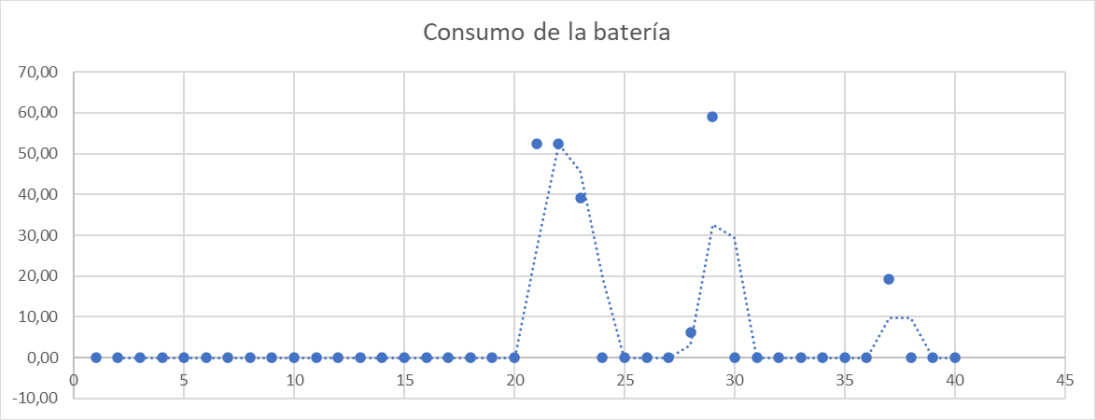


Figura 3.31. Arranque de la motoneta primer estado, Potencia vs Tiempo.

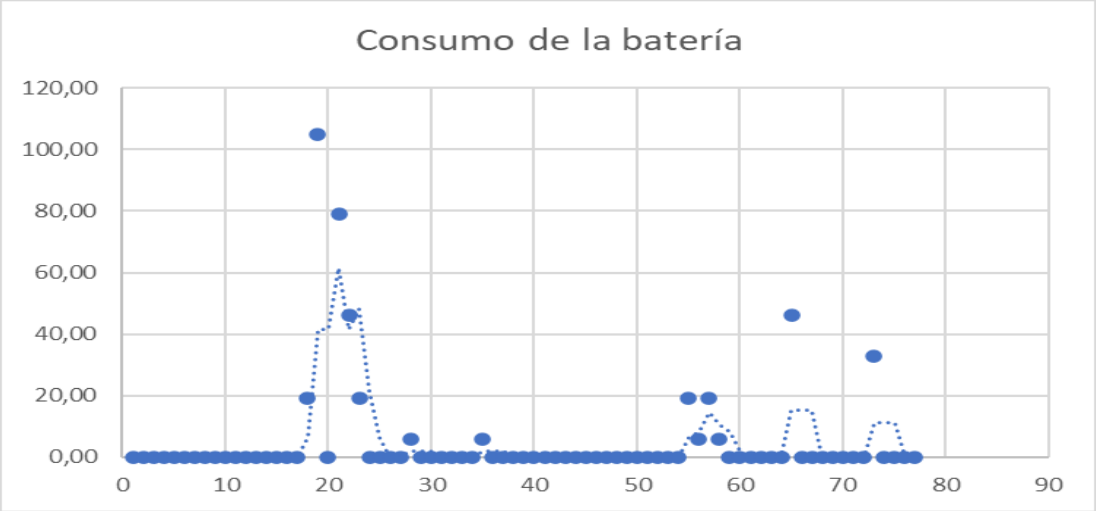


Figura 3.32. Arranque de la motoneta segundo estado, Potencia vs Tiempo.

Según se puede observar en las figuras 3.31 y 3.32, durante el análisis se encontró que cuando la motoneta no está arrancando, su potencia es cero watts, ya que la corriente que consume también es cero. Por otro lado, al realizar arranques, la potencia puede superar los cien watts. Es importante mencionar que estos resultados fueron obtenidos con la motoneta sin carga de pasajeros, por lo que su potencia no es muy alta.

Por siguiente, se ha obtenido los parámetros de la potencia de la batería con el peso del conductor en un recorrido de tres minutos.

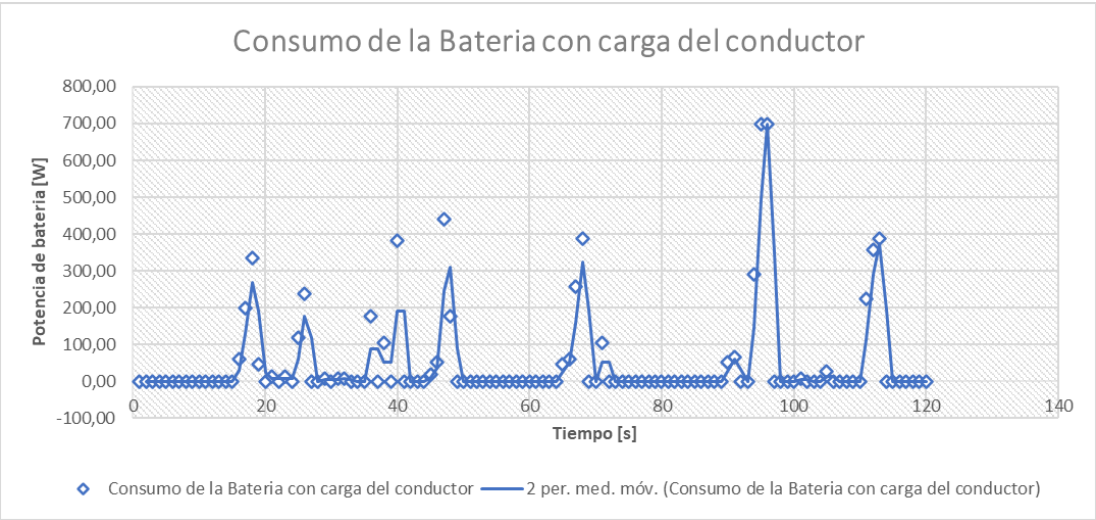


Figura 3.33. Comportamiento del consumo de la batería de la motoneta con carga.

o 3.3 ANÁLISIS Y RESULTADOS DE LOS PARÁMETROS EN UN RECORRIDO.

Al recorrer la motocicleta por diferentes trayectorias de la universidad se ha logrado registrar y enviar información en tiempo real sobre la ubicación en diferentes puntos y consumo de la batería de la motocicleta. Además, la visualización de los datos en la nube Adafruit permite un monitoreo y análisis detallado de la información para mejorar el rendimiento y eficiencia de la motocicleta eléctrica, para realizar una serie de análisis.

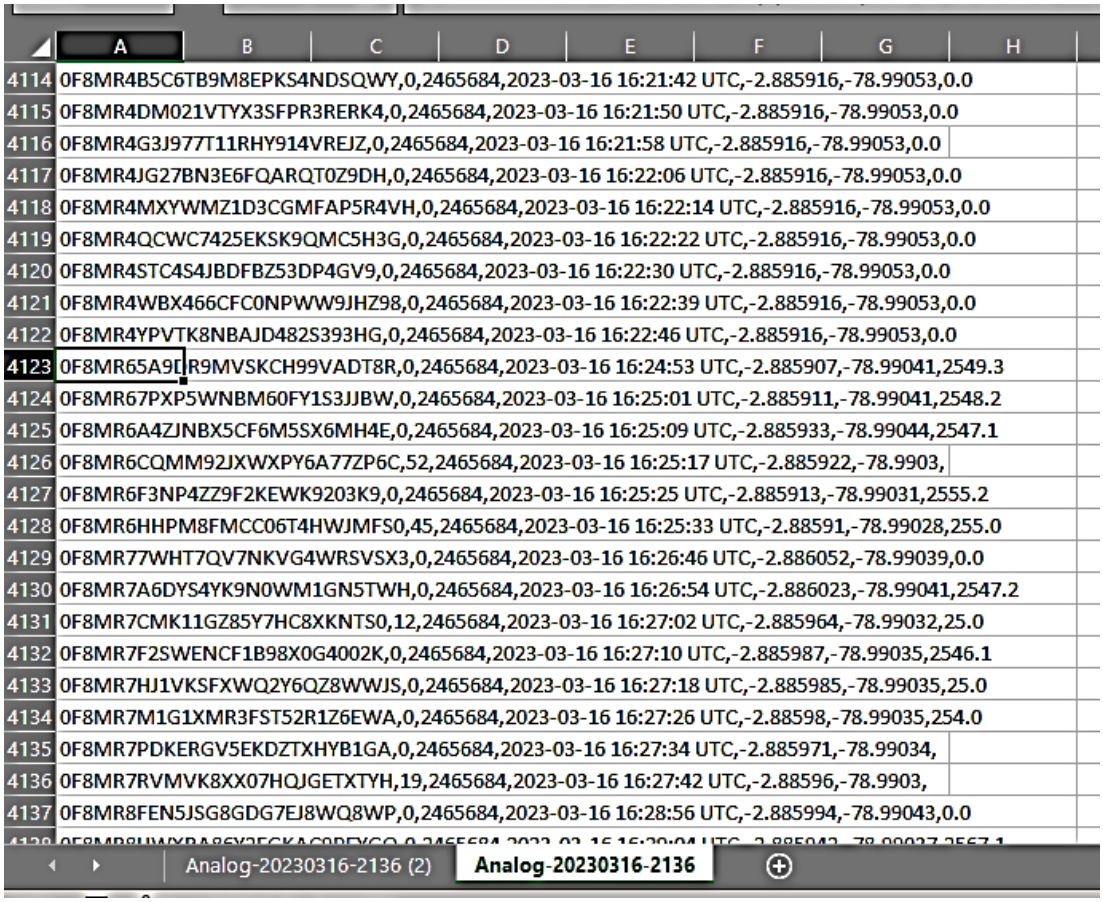
Figura 3.34. Ubicación de la motoneta en su último recorrido.

Al registrar de ubicación y consumo de batería se almacenan en la nube Adafruit con un registro de fecha y hora, permite una monitorización y análisis más preciso del desempeño de la motocicleta eléctrica en distintas situaciones y recorridos. Además, estos datos pueden ser visualizados y descargados en diferentes formatos para su posterior análisis.

2023/03/16 16:27:27	256	-2.885551, -7	×
2023/03/16 04:27:19PM	652	-2.885245, -7	×
2023/03/16 04:27:11 p. m.	0	-2.885228, -78.989	×
2023/03/16 04:27:03PM	190	-2.885198, -78.98901, 2582.8	×
2023/03/16 04:26:54PM	0	-2.885198, -78.98881, 2573.5	×
2023/03/16 04:26:46 p. ...	0	-2.885058, -78.98892	×
2023/03/16 04:26:38 p. ...	0	-2.884966, -78.98907, 2570.9	×
2023/03/16 16:26:30	157	-2.884831, -78.98943, 258	×
2023/03/16 04:26:22PM	0		×
2023/03/16 04:26:14PM	124	-2.884882, -78.98	×
2023/03/16 04:26:06 p. ...	0	-2.884963, -78.98984, 2579	×
2023/03/16 04:25:58 p. ...	91	-2.885123, -78.98	×
2023/03/16 04:25:50PM	131	-2.884912, -78.98	×

Figura 3.35. Parámetros almacenados en la variable.

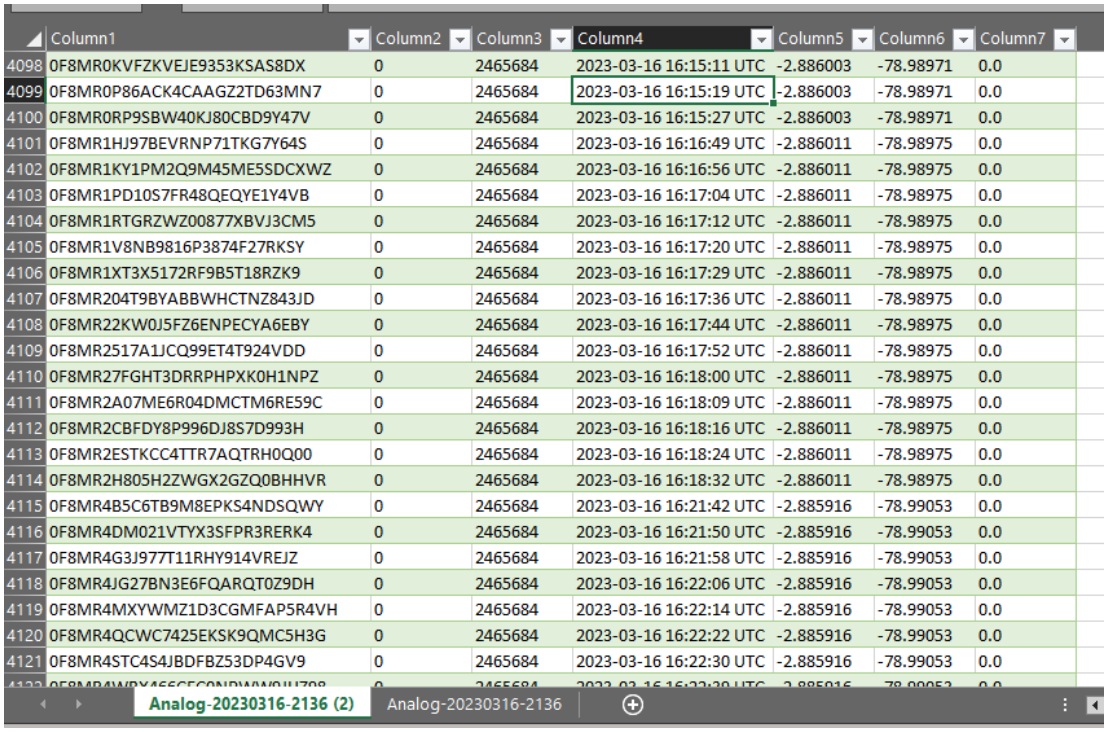
Al obtener un sin número de datos sobre los parámetros más eficiente y visual, se procede a descargar todos los datos en archivo CSV, lo cual, se puede crear tablas dinámicas y gráficos que permite analizar y visualizar los datos de manera más clara y entendible.



	A	B	C	D	E	F	G	H
4114	0F8MR4B5C6TB9M8EPKS4NDSQWY,0,2465684,2023-03-16 16:21:42 UTC,-2.885916,-78.99053,0.0							
4115	0F8MR4DM021VTYX3SFPR3RERK4,0,2465684,2023-03-16 16:21:50 UTC,-2.885916,-78.99053,0.0							
4116	0F8MR4G3J977T11RHY914VREJZ,0,2465684,2023-03-16 16:21:58 UTC,-2.885916,-78.99053,0.0							
4117	0F8MR4JG27BN3E6FQARQT0Z9DH,0,2465684,2023-03-16 16:22:06 UTC,-2.885916,-78.99053,0.0							
4118	0F8MR4MXYWMZ1D3CGMFAP5R4VH,0,2465684,2023-03-16 16:22:14 UTC,-2.885916,-78.99053,0.0							
4119	0F8MR4QCWC7425EKSK9QMC5H3G,0,2465684,2023-03-16 16:22:22 UTC,-2.885916,-78.99053,0.0							
4120	0F8MR4STC4S4JBDFBZ53DP4GV9,0,2465684,2023-03-16 16:22:30 UTC,-2.885916,-78.99053,0.0							
4121	0F8MR4WBX466CFC0NPWW9JHZ98,0,2465684,2023-03-16 16:22:39 UTC,-2.885916,-78.99053,0.0							
4122	0F8MR4YPVTK8NBAJD482S393HG,0,2465684,2023-03-16 16:22:46 UTC,-2.885916,-78.99053,0.0							
4123	0F8MR65A9E R9MVSKCH99VADT8R,0,2465684,2023-03-16 16:24:53 UTC,-2.885907,-78.99041,2549.3							
4124	0F8MR67PXP5WNBM60FY1S3JJBW,0,2465684,2023-03-16 16:25:01 UTC,-2.885911,-78.99041,2548.2							
4125	0F8MR6A4ZJNBX5CF6M5SX6MH4E,0,2465684,2023-03-16 16:25:09 UTC,-2.885933,-78.99044,2547.1							
4126	0F8MR6CQMM92JXWXPY6A77ZP6C,52,2465684,2023-03-16 16:25:17 UTC,-2.885922,-78.9903,							
4127	0F8MR6F3NP4ZZ9F2KEWK9203K9,0,2465684,2023-03-16 16:25:25 UTC,-2.885913,-78.99031,2555.2							
4128	0F8MR6HHPM8FMCC06T4HWJMFS0,45,2465684,2023-03-16 16:25:33 UTC,-2.88591,-78.99028,255.0							
4129	0F8MR77WHT7QV7NKVG4WRSVSX3,0,2465684,2023-03-16 16:26:46 UTC,-2.886052,-78.99039,0.0							
4130	0F8MR7A6DYS4YK9N0WM1GN5TWH,0,2465684,2023-03-16 16:26:54 UTC,-2.886023,-78.99041,2547.2							
4131	0F8MR7CMK11GZ85Y7HC8XKNTS0,12,2465684,2023-03-16 16:27:02 UTC,-2.885964,-78.99032,25.0							
4132	0F8MR7F2SWENCF1B98X0G4002K,0,2465684,2023-03-16 16:27:10 UTC,-2.885987,-78.99035,2546.1							
4133	0F8MR7HJ1VKSFXWQ2Y6QZ8WWJS,0,2465684,2023-03-16 16:27:18 UTC,-2.885985,-78.99035,25.0							
4134	0F8MR7M1G1XMR3FST52R1Z6EWA,0,2465684,2023-03-16 16:27:26 UTC,-2.88598,-78.99035,254.0							
4135	0F8MR7PDKERGV5EKDZTXHYB1GA,0,2465684,2023-03-16 16:27:34 UTC,-2.885971,-78.99034,							
4136	0F8MR7RVMVK8XX07HQJGETXTYH,19,2465684,2023-03-16 16:27:42 UTC,-2.88596,-78.9903,							
4137	0F8MR8FEN5JSG8GDG7EJ8WQ8WP,0,2465684,2023-03-16 16:28:56 UTC,-2.885994,-78.99043,0.0							

Analog-20230316-2136 (2) | Analog-20230316-2136

Figura 3.36. Datos tipo CSV descargados de la nube.



	Column1	Column2	Column3	Column4	Column5	Column6	Column7
4098	0F8MR0KVFZKVEJE9353KSAS8DX	0	2465684	2023-03-16 16:15:11 UTC	-2.886003	-78.98971	0.0
4099	0F8MR0P86ACK4CAAGZ2TD63MN7	0	2465684	2023-03-16 16:15:19 UTC	-2.886003	-78.98971	0.0
4100	0F8MR0RP9SBW40KJ80CBD9Y47V	0	2465684	2023-03-16 16:15:27 UTC	-2.886003	-78.98971	0.0
4101	0F8MR1HJ97BEVRNP71TKG7Y64S	0	2465684	2023-03-16 16:16:49 UTC	-2.886011	-78.98975	0.0
4102	0F8MR1KY1PM2Q9M45ME5SDCXWZ	0	2465684	2023-03-16 16:16:56 UTC	-2.886011	-78.98975	0.0
4103	0F8MR1PD10S7FR48QEQYE1Y4VB	0	2465684	2023-03-16 16:17:04 UTC	-2.886011	-78.98975	0.0
4104	0F8MR1RTGRZWZ00877XBVJ3CM5	0	2465684	2023-03-16 16:17:12 UTC	-2.886011	-78.98975	0.0
4105	0F8MR1V8NB9816P3874F27RKSY	0	2465684	2023-03-16 16:17:20 UTC	-2.886011	-78.98975	0.0
4106	0F8MR1XT3X5172RF9B5T18RZK9	0	2465684	2023-03-16 16:17:29 UTC	-2.886011	-78.98975	0.0
4107	0F8MR204T9BYABBWHCTNZ843JD	0	2465684	2023-03-16 16:17:36 UTC	-2.886011	-78.98975	0.0
4108	0F8MR22KW0J5FZ6ENPECYA6EBY	0	2465684	2023-03-16 16:17:44 UTC	-2.886011	-78.98975	0.0
4109	0F8MR2517A1JCQ99ET4T924VDD	0	2465684	2023-03-16 16:17:52 UTC	-2.886011	-78.98975	0.0
4110	0F8MR27FGHT3DRRPHPXK0H1NPZ	0	2465684	2023-03-16 16:18:00 UTC	-2.886011	-78.98975	0.0
4111	0F8MR2A07ME6R04DMCTM6RE59C	0	2465684	2023-03-16 16:18:09 UTC	-2.886011	-78.98975	0.0
4112	0F8MR2CBFDY8P996DJ8S7D993H	0	2465684	2023-03-16 16:18:16 UTC	-2.886011	-78.98975	0.0
4113	0F8MR2ESTKCC4TTR7AQTRH0Q00	0	2465684	2023-03-16 16:18:24 UTC	-2.886011	-78.98975	0.0
4114	0F8MR2H805H2ZWGX2GZQ0BHHVR	0	2465684	2023-03-16 16:18:32 UTC	-2.886011	-78.98975	0.0
4115	0F8MR4B5C6TB9M8EPKS4NDSQWY	0	2465684	2023-03-16 16:21:42 UTC	-2.885916	-78.99053	0.0
4116	0F8MR4DM021VTYX3SFPR3RERK4	0	2465684	2023-03-16 16:21:50 UTC	-2.885916	-78.99053	0.0
4117	0F8MR4G3J977T11RHY914VREJZ	0	2465684	2023-03-16 16:21:58 UTC	-2.885916	-78.99053	0.0
4118	0F8MR4JG27BN3E6FQARQT0Z9DH	0	2465684	2023-03-16 16:22:06 UTC	-2.885916	-78.99053	0.0
4119	0F8MR4MXYWMZ1D3CGMFAP5R4VH	0	2465684	2023-03-16 16:22:14 UTC	-2.885916	-78.99053	0.0
4120	0F8MR4QCWC7425EKSK9QMC5H3G	0	2465684	2023-03-16 16:22:22 UTC	-2.885916	-78.99053	0.0
4121	0F8MR4STC4S4JBDFBZ53DP4GV9	0	2465684	2023-03-16 16:22:30 UTC	-2.885916	-78.99053	0.0

Analog-20230316-2136 (2) | Analog-20230316-2136

Figura 3.37. Datos convertidos de CVS a xlsx.

Durante la obtención de los parámetros se procede a calcular la distancia entre dos puntos, (se puede calcular la velocidad dividiendo la distancia recorrida por el tiempo transcurrido entre los dos puntos). para encontrar la distancia en el eje x, y, z; entonces, se calcula el consumo de energía de la batería, se puede utilizar la fórmula: consumo de energía = potencia x tiempo. Con esta información, se pueden crear tablas y gráficos en Excel para visualizar la distancia recorrida, la velocidad, el consumo de energía y otros parámetros relevantes. Esto permite realizar un análisis detallado del rendimiento de la motocicleta eléctrica y tomar decisiones informadas sobre su uso y mantenimiento.

Se empieza obteniendo los parámetros de potencia y ubicación como se encuentra en la siguiente tabla.

Tabla 3.2 Valores de fecha, potencia y ubicación de la moto eléctrica.

N°	Fecha	Hora	Potencia	Ubicación
1	2023/03/16	16:24:54	19	-2.885991, -78.98994, 25

2	2023/03/16	16:25:34	39	-2.885057, -78.989
3	2023/03/16	16:25:50	131	-2.884912, -78.98
4	2023/03/16	16:25:58	91	-2.885123, -78.98
5	2023/03/16	16:26:14	124	-2.884882, -78.98
6	16/3/2023	16:26:30	157	-2.884831, -78.98943, 258.0
7	2023/03/16	16:27:03	190	-2.885198, -78.98901, 2582.8
8	2023/03/16	16:27:19	652	-2.885245, -7.0
9	16/3/2023	16:27:27	256	-2.885551, -7.0
10	16/3/2023	16:27:35	164	
11	2023/03/16	16:27:51	32	-2.886538, -78.98936, 25.0
12	2023/03/16	16:27:59	12	-2.886571, -78.98914, 25.0
13	16/3/2023	16:28:39	85	-2.885998, -78.98864, 2571.0
14	16/3/2023	16:28:47	72	-2.886032, -78.98862, 2572.0
15	16/3/2023	16:28:55	309	-2.885887, -78.98843, 2578.3
16	2023/03/16	16:29:03	98	-2.885957, -78.98845, 2586.0
17	2023/03/16	16:29:11	6	-2.886009, -78.98856, 2593.3
18	16/3/2023	16:29:27	6	
19	2023/03/16	16:29:59	131	-2.886753, -78.98904
20	16/3/2023	16:30:15	138	-2.886544, -78.98932
21	16/3/2023	16:30:31	171	-2.886898, -78.9894
22	2023/03/16	16:30:55	171	-2.886734, -78.9899
23	2023/03/16	16:31:11	65	-2.886953, -78.98962, 2576.5
24	16/3/2023	16:31:19	296	-2.88683, -78.98981, 2577.0

Por siguiente se obtiene la latitud y longitud en grados, minutos, segundos (GMS); y también la latitud y longitud en grados decimales. Este proceso es necesario para obtener la distancia recorrida entre dos puntos.

Tabla 3.3 Valores de latitud y longitud de la ubicación de la moto.

Hora	Latitud (GMS)	Longitud (GMS)	Latitud (Decimal)	Longitud (Decimal)
16:24:54	2° 53' 9,53"	78° 59' 23.81"	2.88598	78.9899
16:25:34	2° 53' 6.238'	78° 59' 22.241"	2.88507	78.9895
16:25:50	2° 53' 5,658"	78° 59' 20.41"	2.9849	78.989
16:25:58	2° 53' 6.432"	78° 58' 44.399"	2.9851	78.979
16:26:14	2° 53' 5,561"	78° 58' 47.964"	2.9849	78.9967
16:26:30	2° 53' 5,366"	78° 59' 21.966"	2.8848	78.9894
16:27:03	2° 53' 6.724"	78° 59' 20.41"	2.8952	78.9890

16:27:19	2° 53' 6.918"	78° 59' 20.41"	2.8953	78.9890
16:27:27	2° 53' 7.984"	78° 59' 20.41"	2.8956	78.9890
16:27:35				
16:27:51	2° 53' 11.576"	78° 59' 21.674"	2.8865	78.9894
16:27:59	2° 53' 11.674"	78° 59' 20.896"	2.8866	78.9891
16:28:39	2° 53' 9.636"	78° 59' 19.147"	2.8860	78.9887
16:28:47	2° 53' 9.733"	78° 59' 19.049"	2.8860	78.9886
16:28:55	2° 53' 9.15"	78° 59' 18.369"	2.8859	78.9884
16:29:03	2° 53' 9.442"	78° 59' 18.466"	2.8860	78.9885
16:29:11	2° 53' 9.636"	78° 59' 18.855"	2.8860	78.9886
16:29:27				
16:29:59	2° 53' 12.35"	78° 59' 20.507"	2.8868	78.9890
16:30:15	2° 53' 11.576"	78° 59' 21.577"	2.8893	78.9893
16:30:31	2° 53' 12.836"	78° 59' 21.868"	2.8869	78.9894
16:30:55	2° 53' 12.253"	78° 59' 23.618"	2.8867	78.9899
16:31:11	2° 53' 13.031"	78° 59' 22.646"	2.8870	78.9896
16:31:19	2° 53' 12.545"	78° 59' 23.326"	2.8898	78.9898

Se calcula la distancia recorrida en grados entre dos puntos P y Q en cierto tiempo, para ello nos ayudamos de la siguiente Ecuación (1), este cálculo se realiza usando datos geoespaciales [22]:

$$Ecuacion\ (1)\quad \left|\overline{P\,Q}\right|[°] = (q_x - p_x)^2 + \left(q_y - p_y\right)^2$$

El radio R de la tierra se determina para transformar los grados en metros. Las coordenadas están referenciadas a un elipsoide terrestre específico. El elipsoide de tierra más utilizado es el elipsoide WGS 84, que tiene un radio polar de 6.356.752,314 m y un radio ecuatorial de 6.378.137,0 m [22]. El radio en cada área Ra se puede interpolar según la latitud de la siguiente manera:

$$Ecuacion\ (2)\qquad R_a = R_{eq} - \left(\frac{R_{eq} - R_{po}}{90°} - Lat_a\right)$$

Así, se obtiene la distancia en metros:

$$Ecuacion\ (3)\qquad \left|\overline{P\ Q}\right|[m] = \frac{\left|\overline{P\ Q}\right|[°]}{360°}.\ 2\pi.\ R_a$$

Y para obtener la velocidad recorrida en cierta aquella distancia se obtiene con la siguiente fórmula:

$$Ecuacion\ (4) \qquad v = \frac{d}{\Delta t}$$

Tabla 3.4 Valores de distancia y velocidad recorrida por la moto.

Hora	Distancia entre dos puntos [m]	Velocidad [m/s]
16:24:54	0.99	0.04970
16:25:34	99.983	2.773090337
16:25:50	10.00	1.250249975
16:25:58	17.70	1.106320619
16:26:14	100.37	6.272864427
16:26:30	10.41	0.3153845292
16:27:03	0.10	0.00625
16:27:19	0.30	0.0375
16:27:27		
16:27:35		
16:27:51	0.32	0.03952847075
16:27:59	0.72	0.01802775638
16:28:39	0.10	0.0125
16:28:47	0.22	0.02795084972
16:28:55	0.14	0.01767766953
16:29:03	0.10	0.0125
16:29:11		
16:29:27		
16:29:59	2.52	0.1573709789
16:30:15	2.40	0.1501301519
16:30:31	0.54	0.0224381867
16:30:55	0.42	0.02651650429
16:31:11	2.81	0.3508917212
16:31:19		

Las gráficas realizadas a partir de los parámetros obtenidos es una forma útil de visualizar el comportamiento de la moto eléctrica y su desempeño en diferentes situaciones. Por ejemplo, se puede graficar la potencia en función del tiempo, lo que

permitiría observar los momentos en que se requiere mayor o menor potencia para desplazarse.

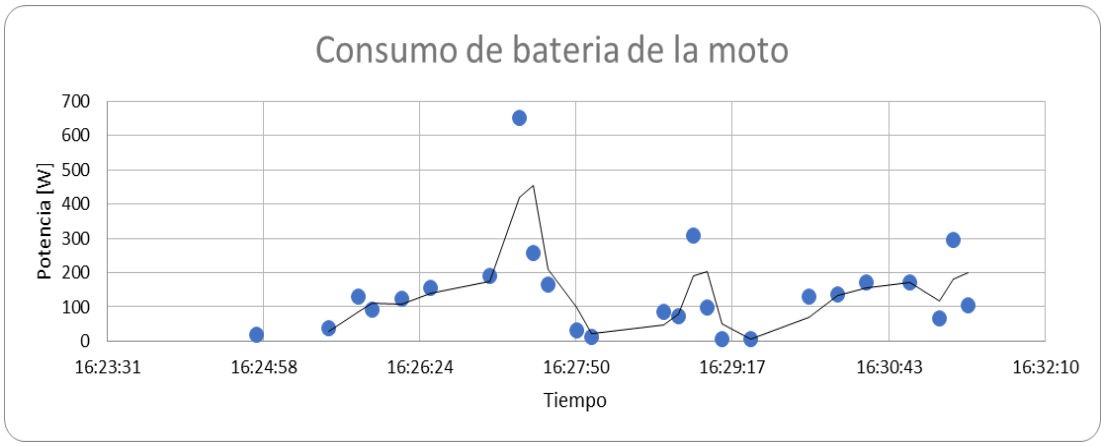


Figura 3.38. Curva de la potencia de la batería respecto al tiempo transcurrido.

De igual manera, se puede graficar la distancia recorrida en función del tiempo para observar la velocidad a la que se desplaza la moto y los momentos en que se detiene o acelera. También se pueden realizar gráficas de la energía consumida en función de la distancia recorrida, lo que permitiría determinar la eficiencia energética de la moto y cómo ésta varía en diferentes situaciones de conducción. Estas gráficas pueden ser útiles para tomar una decisión.

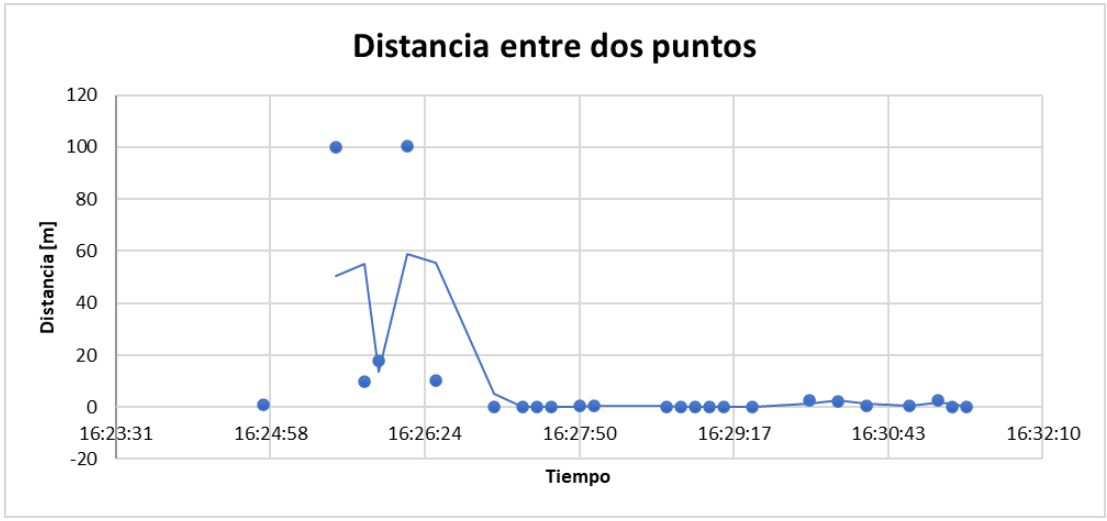


Figura 3.39. Curva de la distancia recorrida por la moto eléctrica.

En esta Figura 3.39, se puede observar que el comportamiento de la moto está relacionado directamente con la potencia que necesita para recorrer cierta distancia a cierta velocidad y que las pendientes en el camino afectan el consumo de energía de la batería.

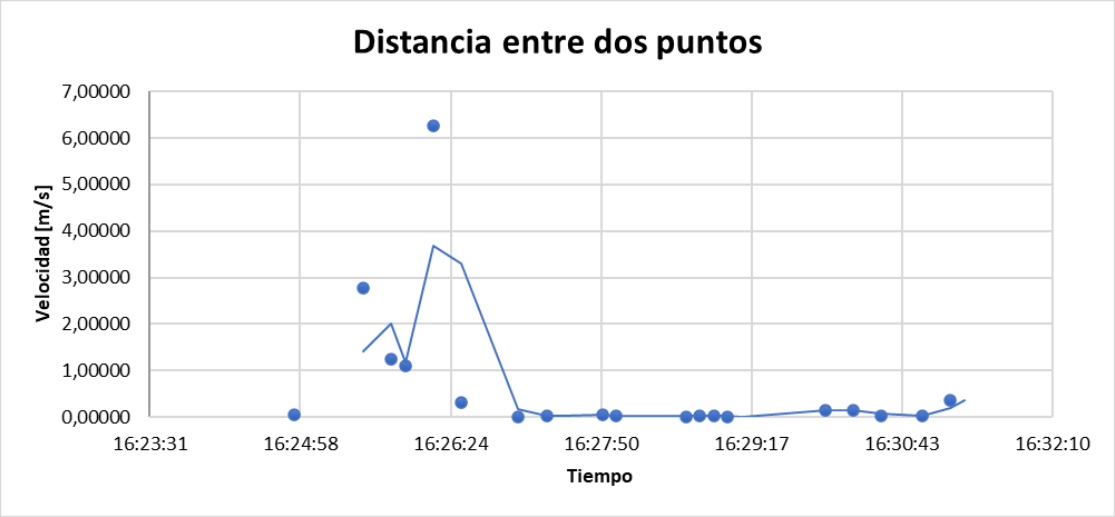


Figura 3.40. Curva de la velocidad de la moto.

En efecto, en determinadas condiciones, el motor puede alcanzar su potencia máxima y a partir de ahí, aunque se acelere más, la potencia no aumentará más debido a que la curva de potencia entra en su tramo descendente. La eficiencia del motor comienza a disminuir y se produce una pérdida de potencia. Por lo tanto, es importante conocer la curva de potencia del motor para poder optimizar su rendimiento y obtener el máximo aprovechamiento del consumo de batería.

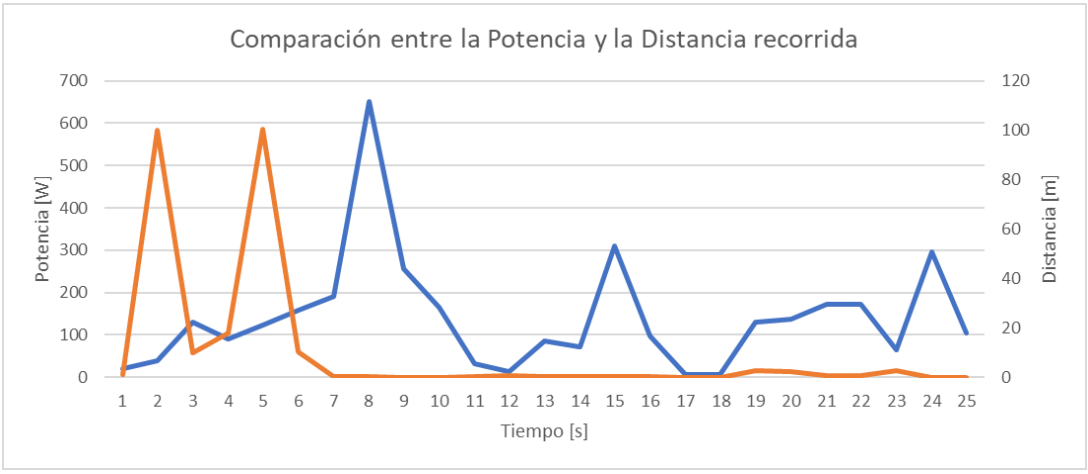


Figura 3.41. Curva de comportamiento de la potencia junto a la distancia en m/s.

En estas condiciones, aunque se acelere más la velocidad del motor, éste no es capaz de entregar más potencia dado que la curva entra en su tramo descendente.

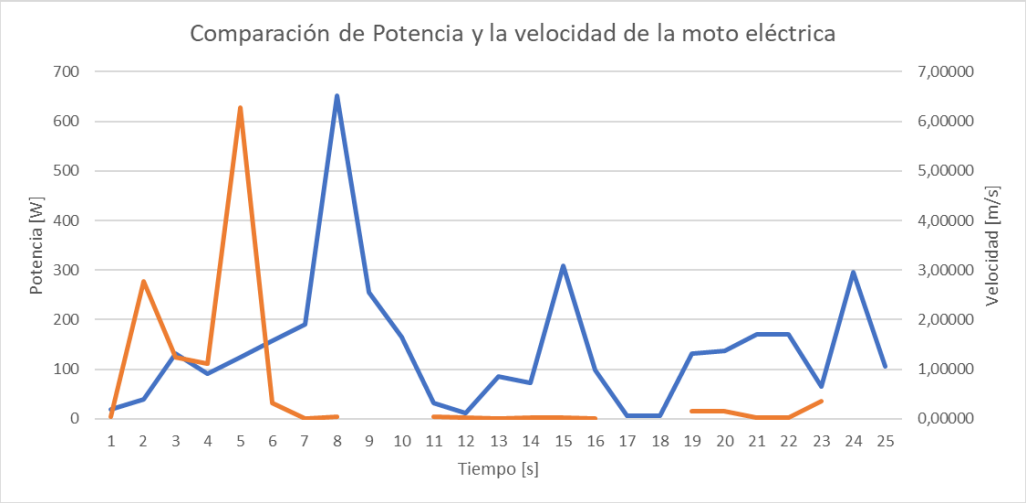


Figura 3.42. Curva de comportamiento de Potencia con la velocidad de la moto en m/s.

En la Figura 3.43, -vemos que la tendencia de la altitud en el recorrido dentro de la universidad no tiene variaciones significativas, por lo que se mantiene en valores cercanos a un mapeo lineal. La importancia del comportamiento de la potencia para cierta velocidad radica en que nos permite conocer el consumo de energía de la batería en diferentes condiciones de conducción, lo que puede ayudarnos a optimizar el rendimiento y el consumo de energía de la motocicleta eléctrica.

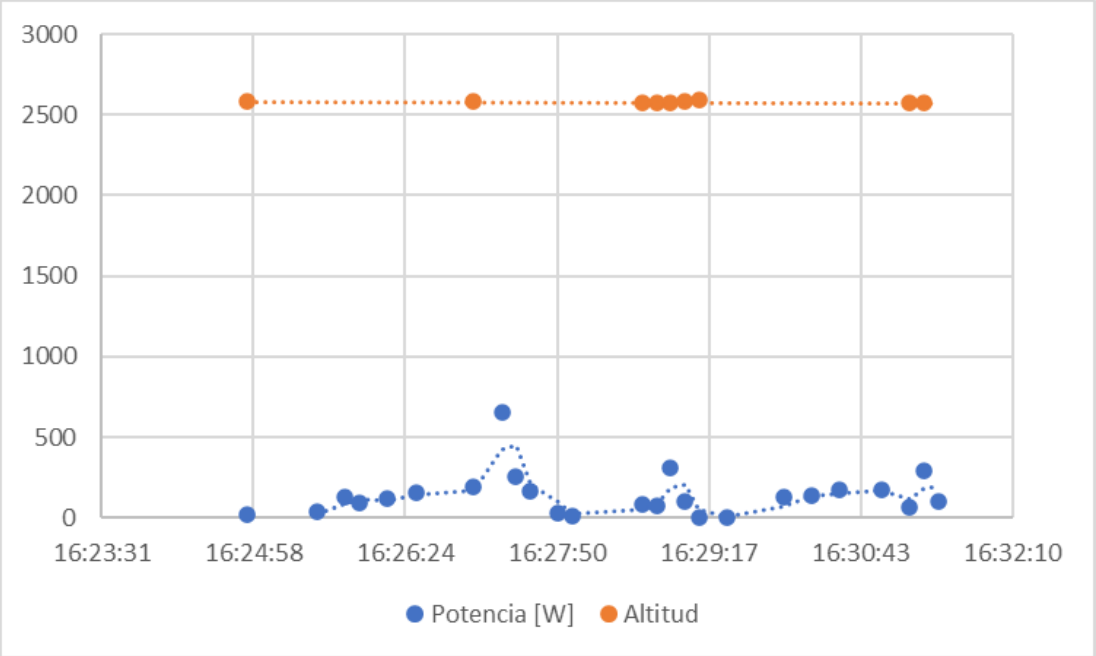


Figura 3.43. Curva de la potencia de la batería junto a la altitud de ubicación del GPS

● CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

o 4.1 CONCLUSIONES

En la medición de los parámetros de voltaje, aunque exista un pequeño margen de error en comparación con la medición mediante un multímetro, esto puede variar dependiendo de la herramienta utilizada y la programación empleada debido a las diferentes fórmulas utilizadas. No obstante, la diferencia de valores es insignificante. Por otro lado, en la medición de corriente, el margen de error es ligeramente mayor debido al margen de error inherente al sensor utilizado. A pesar de que la construcción de la placa es corta debido al número reducido de componentes, se debe tener en cuenta que se necesitará espacio adicional para alojar la fuente de energía para cada dispositivo, ya que se requieren dos fuentes en total, tanto para la placa de Arduino como como la del módulo ESP32.

En el desarrollo de la plataforma, es fundamental prestar especial atención a la programación utilizada, ya que se emplean diversos protocolos para la transmisión de datos. Uno de estos protocolos es MQTT, el cual es utilizado en la página de Adafruit y permite la publicación y suscripción de datos con dispositivos que tienen limitaciones de recursos. Asimismo, es importante considerar las librerías utilizadas, como la comunicación serial entre el microcontrolador y los diferentes módulos empleados, así como la comunicación entre el módulo WiFi y el programa Adafruit IO

Para llevar a cabo el análisis de los datos adquiridos de la batería y el seguimiento de la motocicleta, es necesario realizar diversas pruebas que involucran diferentes situaciones. Estas pruebas incluyen la evaluación de la motocicleta mientras se encuentra en reposo, en movimiento con conductor o carga, y recorriendo diferentes espacios a distintas velocidades, considerando los obstáculos en la vía, tales como rampas, huecos, desniveles, entre otros. Además, el GPS debe ser capaz de rastrear la ubicación de la motocicleta, mostrando su longitud, latitud y altitud.

Es importante tener en cuenta la cantidad de datos que se publican en la nube Adafruit, ya que la publicación de una gran cantidad de datos en un corto periodo de

tiempo puede congestionar la nube. Por este motivo, se estableció un intervalo de 5 segundos entre cada dato enviado para evitar la transmisión simultánea de varios datos y asegurar el correcto funcionamiento de la plataforma.

En los resultados obtenidos a partir de las curvas de potencia, se pueden observar los cambios bruscos en la potencia debidos a la fuerza que ejerce la moto eléctrica al subir o bajar pendientes. Si el usuario no acelera la moto, la potencia es cero, ya que no se realiza ninguna acción para acelerarla. Además, la velocidad aumenta a medida que la potencia de la moto se incrementa, permitiendo una mayor velocidad de la misma.

El uso de la web Adafruit permitió la visualización gráfica y fácil reconocimiento de los parámetros establecidos en la monitorización como son: voltaje, corriente, potencia, consumo de energía y ubicación por GPS. Además, favoreció la implementación de gráficos, las cuales son guardadas en un archivo CSV. La finalidad de generar los datos en los gráficos nos permite mantener informado al usuario cuándo se generan grandes consumos de energía.

En la tabla 3.2 de los resultados, se puede observar que la potencia varía en función del recorrido, especialmente en las zonas de subida y bajada. Cuando la moto requiere un mayor esfuerzo para subir una pendiente, consume una mayor cantidad de potencia, lo que resulta en un aumento en el consumo de la batería.

En la Tabla 3.4 de los resultados, se puede apreciar que la distancia recorrida aumenta a medida que aumenta el tiempo de recorrido. En particular, en los casos en que se recorrieron 100 metros, se observó una mayor velocidad, lo que permitió cubrir una distancia tan larga. En conclusión, se puede afirmar que la velocidad es proporcional a la distancia recorrida.

o 4.2 RECOMENDACIONES

Aumentar la tasa de baudios: la tasa de baudios es el número de bits que se transmiten por segundo. Al aumentar la tasa de baudios se puede lograr una mayor velocidad de transmisión. Sin embargo, es importante tener en cuenta que a medida que se aumenta la tasa de baudios, la probabilidad de errores de transmisión también

aumenta. Reducir la cantidad de datos transmitidos: si es posible, se pueden reducir los datos que se transmiten para lograr una mayor velocidad. Por ejemplo, se pueden enviar solo los datos esenciales y evitar enviar datos redundantes.

Optimizar el protocolo de comunicación: se puede optimizar el protocolo de comunicación para reducir el tiempo de espera y mejorar la eficiencia de la transmisión. Por ejemplo, se puede implementar un protocolo que permita la transmisión de múltiples paquetes de datos en paralelo.

Utilizar un hardware más avanzado: si se está utilizando un hardware obsoleto o de baja calidad, se puede considerar la actualización a un hardware más avanzado que tenga una mayor capacidad de procesamiento y una velocidad de transmisión más alta.

En la medición de corriente, se utiliza un sensor con un rango tres veces mayor a la corriente nominal de la batería para prevenir daños en el circuito eléctrico durante arranques o situaciones en las que se produzcan picos de corriente debido al peso de la motocicleta u otras circunstancias. Además, se recomienda el uso de fusibles para evitar problemas similares.

Cuando se utiliza la plataforma Adafruit, es importante tener en cuenta que existen versiones gratuitas y de pago. En este proyecto, se utilizó la versión gratuita, la cual tiene ciertos límites, como la transmisión de datos, que está limitada a treinta datos por segundo. Es importante considerar que, al enviar datos de potencia y ubicación, se debe alargar el tiempo de envío para evitar el congelamiento de la nube, ya que el envío de grandes cantidades de datos puede congestionar la plataforma, lo que requiere cierto tiempo para volver a recopilar los datos recibidos.

Para garantizar el correcto funcionamiento del prototipo adecuado, se pretende diseñar un convertidor DC a DC desde la propia batería de la motocicleta, lo cual se evita la necesidad de utilizar fuentes externas, lo que reduce el costo y el tiempo de instalación.

Con la finalidad de continuar con la línea de investigación se plantea graficar la velocidad y la distancia en la web de Adafruit con la finalidad de obtener más información relevante sobre la motocicleta. Con la finalidad de mantenernos

informados sobre la optimización del uso de energía y el desempeño de la moto eléctrica.

● REFERENCIAS BIBLIOGRÁFICAS

[1] Ubysz, A. (s. f.). EFFECTIVE EFFICIENCY IN CAR ENGINE AS A FUNCTION OF REDUCTION RATIO DURING ENERGY-SAVING SPEED CONTROL (Journal of KONES Powertrain and Transport, No. 4 2010). Vol. 17.

[2] P. L. Castro Verdezoto, J. A. Vidoza y W. L. R. Gallo, "Analysis and projection of energy consumption in Ecuador: Energy efficiency policies in the transportation sector", Brazil, Volume 134, noviembre de 2019. Accedido el 19 de noviembre de 2021. [En línea]. Disponible: https://reader.elsevier.com/reader/sd/pii/S030142151930535X?token=48E7BB77BF1C59 943F6145980C8B3997C9F1F4C8D11D9531CB8696B0798916195592D2CE2CF0697D F1F21C5F6C600041&originRegion=us-east1&originCreation=2021111922461

[3] K. Arai y R. Mardiyanto, "Eyes Based Eletric Wheel Chair Control System- - I (eye) can control Electric Wheel Chair -", International Journal of Advanced Computer Science and Applications, vol. 2, n.º 12, 2011. Accedido el 20 de noviembre de 2021. [En línea]. Disponible: https://doi.org/10.14569/ijacsa.2011.021215

[4] "TRINITY Venus". ec.e-scooter.co – Comprar Scooter eléctrico 2023. https://ec.e-scooter.co/trinity-venus/ (accedido el 31 de enero de 2023).

[5] "Corriente continua - Concepto y diferencia con la alterna". Concepto. https://concepto.de/corriente-continua/ (accedido el 31 de enero de 2023).

[6] "Diferencia entre voltaje AC y DC". Alles über Elektrik und Elektronik. https://illustrationprize.com/es/51-difference-between-ac-amp-dc-voltage.html (accedido el 31 de enero de 2023).

[7] "Amazon.com". Amazon.com. Spend less. Smile more. https://www.amazon.com/-/es/HiLetgo-ACS758-ACS758LCB-050B-PFF-T-corriente-ACS758LCB/dp/B0832PHKB5 (accedido el 31 de enero de 2023).

[8] "Arduino Mega 2560 Rev3". Arduino Online Shop. https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us (accedido el 31 de enero de 2023).

[9] "Conector De Batería Impermeable Para Motor,M23,8 Pines,Gran Oferta - Buy Waterproof Battery Connector,Chogori Stecker Plug,Chogori 8pin Kabel Product on Alibaba.com". Alibaba - la plataforma de comercio entre empresas en línea más grande del mundo. https://spanish.alibaba.com/product-detail/Hot-sale-M23-8pin-waterproof-battery-62374800157.html (accedido el 31 de enero de 2023).

[10] "Batería litio 60V 20AH NMC caja metálica - Moviltronics". Moviltronics. https://moviltronics.com/tienda/bateria-litio-60v-20ah/ (accedido el 31 de enero de 2023).

[11] "Módulo ESP-WROOM-32 ESP32 WiFi". Naylamp Mechatronics - Perú. https://naylampmechatronics.com/espressif-esp/382-modulo-esp-wroom-32-esp32-wifi.html (accedido el 31 de enero de 2023).

[12] "Módulo GPS NEO-6M V2 APM2.5 + Antena para Arduino RPi PIC - Tecnopura". Tecnopura. https://www.tecnopura.com/producto/modulo-gps-neo-6m-v2-apm2-5-antena-para-arduino-rpi-pic/ (accedido el 31 de enero de 2023).

[13] "Convertidor Voltaje DC-DC Step-Down 3A LM2596". Naylamp Mechatronics - Perú. https://naylampmechatronics.com/conversores-dc-dc/196-convertidor-voltaje-dc-dc-step-down-3a-lm2596.html (accedido el 31 de enero de 2023).

[14] "Amazon.com". Amazon.com. https://www.amazon.com/-/es/MCIGICM-Portafusibles-fusible-cuchilla-paquete/dp/B081DHT8Y7 (accedido el 31 de enero de 2023).

[15] "Uso de la Librería de Software Serial en la automatización". Boot & Work Corp. S.L. https://www.industrialshields.com/es_ES/blog/blog-industrial-open-source-1/post/como-utilizar-la-libreria-de-software-serial-en-el-controlador-industrial-arduino-plc-99 (accedido el 31 de enero de 2023).

[16] "What is MQTT? Definition and Details". Paessler - The Monitoring Experts. https://www.paessler.com/it-explained/mqtt (accedido el 31 de enero de 2023).

[17] "Aprende a utilizar la plataforma Adafruit IO para tus dispositivos IoT (parte 1) | MK Electronica". MK Electronica. https://mkelectronica.com/aprende-a-utilizar-la-plataforma-adafruit-io-para-tus-dispositivos-iot-parte-1/#:~:text=Para%20terminar,%20debemos%20decir%20que,difunde%20entre%20los%20clientes%20suscritos. (accedido el 31 de enero de 2023).

[18] "¿Qué es Wi-Fi? | Definición, significado y explicación". verizon.com. https://espanol.verizon.com/articles/internet-essentials/wifi-definiton/ (accedido el 31 de enero de 2023).

[19] "Que es el buzzer y cómo funciona (zumbador) - Ingeniería Mecafenix". Ingeniería Mecafenix. https://www.ingmecafenix.com/electronica/el-buzzer/ (accedido el 31 de enero de 2023).

[20] "IBM Documentation". IBM - Deutschland | IBM. https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format (accedido el 5 de marzo de 2023).

[21] "What Is Python Used For? A Beginnerâs Guide". Coursera. https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python (accedido el 6 de marzo de 2023).

[22] Wenz, K. P., Serrano-Guerrero, X., Barragán-Escandón, A., González, L. G., & Clairand, J. M. (2021). Route prioritization of urban public transportation from conventional to electric buses: A new methodology and a study of case in an intermediate city of Ecuador. *Renewable and Sustainable Energy Reviews*, *148*, 111215.

APÉNDICES

APÉNDICE A: DIMENSIONES DEL CONECTOR DE BATERÍA

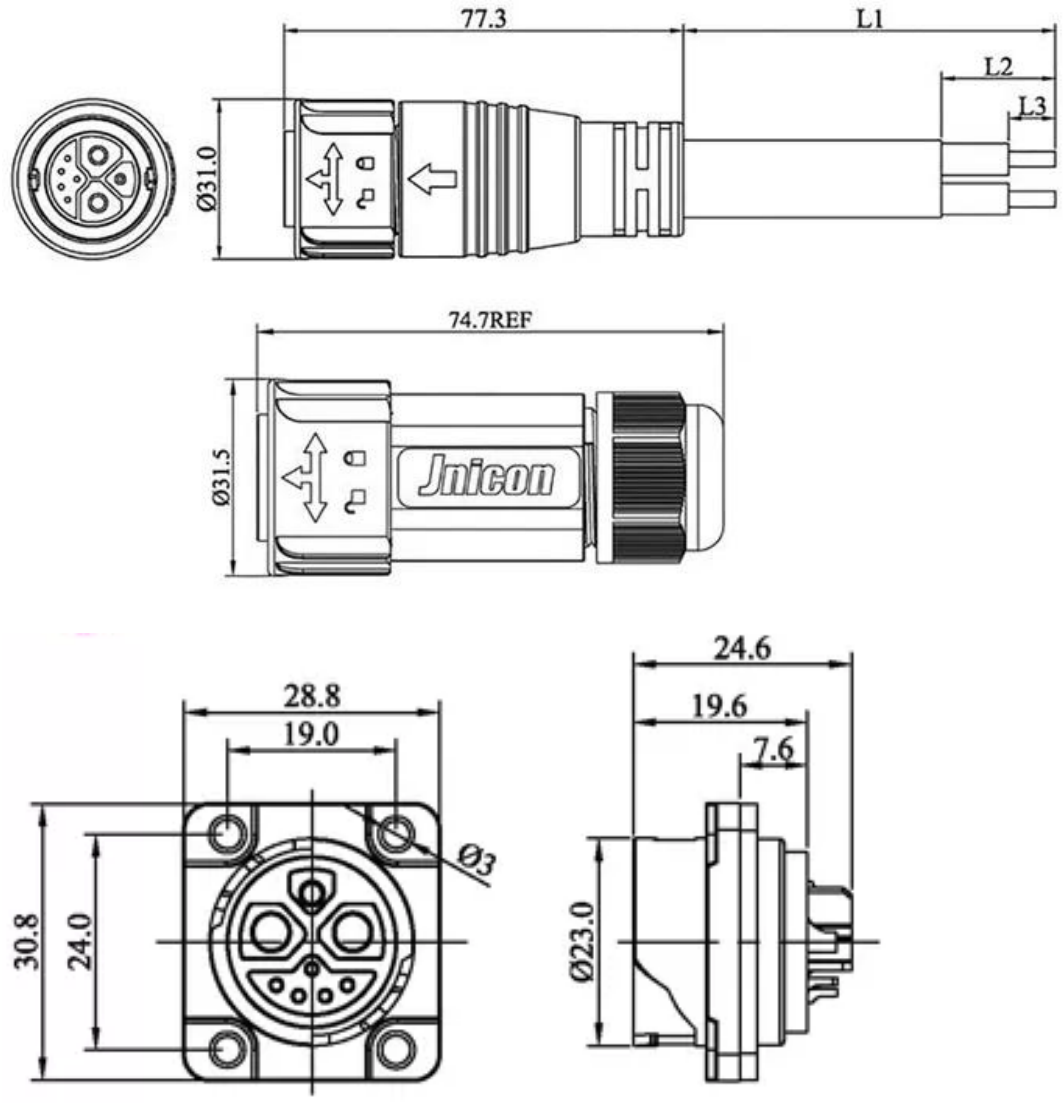


Figura A.1. Dimensiones del cable chogori

APÉNDICE B: CUBIERTA DEL CIRCUITO

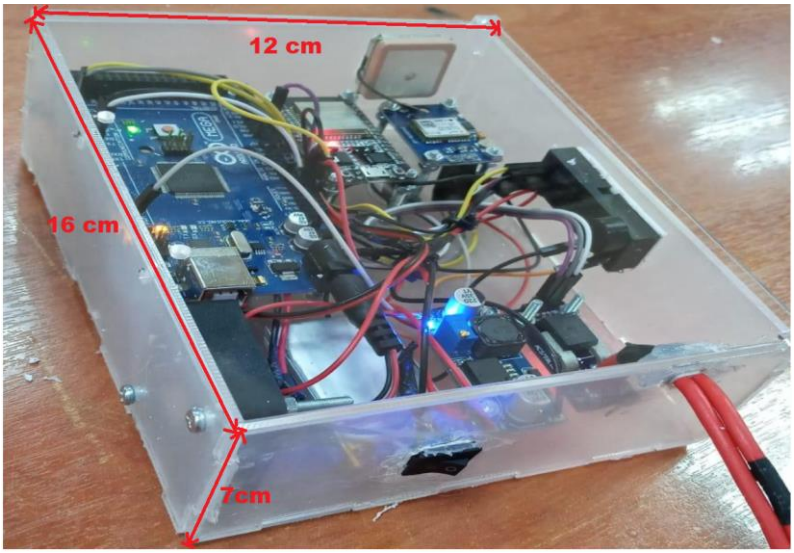


Figura B.1. Dimensiones de la caja del circuito

ANEXOS

Algoritmo 1: Captura de parámetros de potencia para almacenar en tarjeta SD

```
//Principal, librerías
#include "variables.h"
#include "messenger.h"
#include "AnalogicRead.h"
#include "GPS.h"
#include <SD.h>


//Variables hoja principal
int sensor = 0;
File dataFile;

void setup() {
  Serial.begin(9600);
  aReadInit();
  gpsInit();
  lastSend = millis();
  pinMode(53, OUTPUT);
  if (!SD.begin(53)) {
  return;}
  dataFile = SD.open("datalog.txt", FILE_WRITE);

  if (dataFile) {
    dataFile.println("Escribiendo la informacion...");
    dataFile.println("");
    dataFile.println("Programo: Elmer y Teofilo");
    dataFile.println("");
    dataFile.println("<< Datalogger >>");
    dataFile.close();}
}


void loop() {
  aReadLoop();
  gpsLoop();
  String dataString = "";
  sensor = analogRead(analogLecture);
  dataString += String(analogLecture);

dataFile = SD.open("datalog.txt", FILE_WRITE);

  if (dataFile) {
    dataFile.println(dataString);
    delay(1000);
    dataFile.close();
  }
```

}

Algoritmo 2: Captura de parámetros de ubicación, corriente y voltaje en la placa Arduino

```
//AnalogicRead.h
#ifndef A_READ_H
#define A_READ_H
#include "variables.h"

#define VIN A0
const float VCC   = 5.0;
const int model = 0;
float cutOffLimit = 1.00;

/* //DATOS PARA DIFERENTES CORRIENTES
        "ACS758LCB-050B",// for model use 0
        "ACS758LCB-050U",// for model use 1
        "ACS758LCB-100B",// for model use 2
        "ACS758LCB-100U",// for model use 3
        "ACS758KCB-150B",// for model use 4
        "ACS758KCB-150U",// for model use 5
        "ACS758ECB-200B",// for model use 6
        "ACS758ECB-200U"// for model use  7
*/
float sensitivity[] ={
        40.0,// for ACS758LCB-050B
        60.0,// for ACS758LCB-050U
        20.0,// for ACS758LCB-100B ////20
        40.0,// for ACS758LCB-100U
        13.3,// for ACS758KCB-150B
        16.7,// for ACS758KCB-150U
        10.0,// for ACS758ECB-200B
        20.0,// for ACS758ECB-200U
       };


float quiescent_Output_voltage [] ={
        0.5,// for ACS758LCB-050B
        0.12,// for ACS758LCB-050U
        0.5,// for ACS758LCB-100B //0.5
        0.12,// for ACS758LCB-100U
        0.5,// for ACS758KCB-150B
        0.12,// for ACS758KCB-150U
        0.5,// for ACS758ECB-200B
        0.12,// for ACS758ECB-200U
        };
const float FACTOR = sensitivity[model]/1000;
const float QOV =   quiescent_Output_voltage [model] * VCC;
float voltage;// internal variable for voltage
float cutOff = FACTOR/cutOffLimit;
```

```
void aReadLoop(){
  float voltage_raw =   ((5.0 / 1023.0)* analogRead(VIN));//
  voltage =  voltage_raw - QOV + 0.007 ;//
  float current = voltage / FACTOR;

  analogLecture = float(VIN)*current;
  if (analogLecture <= 0.00) {
    analogLecture =0.00;
  }
}
#endif
```

Algoritmo 3: Programación para control del GPS

```
GPS.h
#ifndef GPS_H
#define GPS_H

// Bibliotecas
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include "variables.h"

// Creacion de instacia TinyGPS
TinyGPSPlus gps;

// Inicializacion de gps
void gpsInit(){
  Serial1.begin(9600);
}


void gpsLoop(){
  while(Serial1.available() > 0){
    if(gps.encode(Serial1.read())){
      if(gps.location.isValid()){
        latitud = gps.location.lat();
        longitud = gps.location.lng();
        altura = gps.altitude.meters();
        sendData();
      }
    }
  }
}
#endif
```

Algoritmo 4: Publicación de datos a otra placa microcontroladora

```
//messenger.h
#ifndef MESSENGER_H
#define MESSENGER_H
```

```
#include <ArduinoJson.h>
#include "variables.h"

void sendData(){
  if(millis()-lastSend>500){
    lastSend = millis();
    DynamicJsonDocument doc(200);
    doc["value"]["sensor-1"] = analogLecture;
    doc["value"]["sensor-2"] = altura;
    doc["lat"] = latitud;
    doc["lon"] = longitud;
    doc["ele"] = altura;
    String toReturn = "";
    serializeJson(doc,Serial);
    Serial.println();
  }
}
#endif

//variables.h
#ifndef VARIABLES_H
#define VARIABLES_H
float analogLecture;
double latitud, longitud, altura;
unsigned long lastSend;
#endif
```

Algoritmo 5: Captura de parámetros de la placa principal y publicación de parámetros a Internet, con buzzer de referencia de conectividad

```
#if defined(ESP8266)
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial Serial2(D5, D5); // RX, TX
#elif defined(ESP32)
#include <WiFi.h>
#else
#error Architecture not supported.
#endif
#include <ArduinoJson.h>
#include <PubSubClient.h>

const char *ssid = " "; // usuario WIFI
const char *password = " "; //contraseña WIFI
const char *mqtt_server = "io.adafruit.com";
#define user " " // Cuenta ADAFRUIT
#define key " "
#define topicAnalog " "
#define topicAltitud " "
```

```
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (70)
char msg[MSG_BUFFER_SIZE];
String lectura = "";
DynamicJsonDocument doc(200);
int analogLecture, altitud;
int BUZZER = 13;
int BUZZER_CHANNEL = 0;

void setup_wifi()
{
 delay(10);
 Serial.println();
 Serial.print("Connecting to ");
 Serial.println(ssid);

 WiFi.mode(WIFI_STA);
 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED)
 {
  delay(500);
  Serial.print(".");
 }

 randomSeed(micros());

 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 Serial.println(WiFi.localIP());
}

void reconnect()
{
 // Loop until we're reconnected
 while (!client.connected())
 {
  Serial.print("Attempting MQTT connection...");
  String clientId = "ESP8266Client-";
  clientId += String(random(0xffff), HEX);
  if (client.connect(clientId.c_str(), user, key))
  {
   Serial.println("connected");
  }
  else
  {
   Serial.print("failed, rc=");
```

```
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup()
{
 Serial.begin(115200);
 Serial2.begin(9600);
 setup_wifi();
 client.setServer(mqtt_server, 1883);
 ledcAttachPin(BUZZER, BUZZER_CHANNEL);
}

void loop()
{
 ledcWriteNote(BUZZER_CHANNEL, (note_t)NOTE_A, 8);
 delay(500);
 ledcWriteNote(BUZZER_CHANNEL, NOTE_C, 4);
 delay(500);
 ledcWriteNote(BUZZER_CHANNEL, NOTE_D, 4);
 delay(500);
 ledcWriteNote(BUZZER_CHANNEL, NOTE_E, 4);
 delay(500);
 ledcDetachPin(13);
 if (!client.connected())
 {
  reconnect();
 }

 if (Serial2.available())
 {
  lectura = Serial2.readStringUntil('\n');
 }
 DeserializationError error = deserializeJson(doc, lectura);

 client.loop();
 unsigned long now = millis();
 if (now - lastMsg > 5000 && lectura != "")
 {
  String toSend ="";
  lastMsg = now;
  analogLecture = doc["value"]["sensor-1"];
  altitud = doc["value"]["sensor-2"];
  doc["value"] = analogLecture;
  serializeJson(doc,toSend);
  Serial.println("Publishing analog:" + toSend);
  client.publish(topicAnalog, toSend.c_str());
```

```
    doc["value"] = altitud;
    toSend = "";
    serializeJson(doc,toSend);
    Serial.println("Publishing altitud:" + toSend);
    client.publish(topicAltitud, toSend.c_str());
  }
}
```