INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

RESIDENCIA PROFESIONAL

INGENIERÍA ELECTRÓNICA

"DISEÑO DE UN SISTEMA ELECTRÓNICO DE MONITOREO Y DIAGNÓSTICO DE UN VEHÍCULO ELÉCTRICO."

PRESENTA:

GUTIERREZ GORDILLO JORGE FRANCISCO

PÉREZ MEJÍA BILGAI HILARIO

Asesor:

Dr. Francisco Ronay López Estrada.

Ciclo escolar:

Enero-junio 2019

Tuxtla Gutiérrez, Chiapas 11 de Junio del 2019

INDICE

1.1 INTRODUCCIÓN 4

1.2 PLANTEAMIENTO DEL PROBLEMA. 4

1.3 Justificación 5

1.4 ANTECEDENTES 6

..... **6**

1.5 OBJETIVOS 11

1.5.1 Objetivos generales. **11**

1.5.2 Objetivos específicos: **11**

1.6 alcance y limitaciones 12

1.6.1 alcances **12**

1.6.2 limitaciones **12**

1.7 Lugar en el cual se obtenían los datos. 12

2.1 MARCO TEORICO 13

2.1.1 Udoo-X86 ..... **13**

2.1.2 Arduino **15**

2.1.3 Matlab **17**

2.1.4 Python **18**

2.1.5 Sensor de corriente de efecto Hall en lazo cerrado **20**

2.1.6 Motor *brushless* **21**

2.3.7 ESC: Variador de motor **22**

2.1.8 BATERIAS **23**

3.1 DESARROLLO 25

3.2 Cronograma de actividades propuesto. **25**

3.3 Descripción de actividades. **25**

3.4 Uso de la tarjeta UDOO-X86 ..... **26**

3.5 Diseño de placa para adquisición de datos. 27
3.6 PROGRAMACIONES PARA ADQUIRIR DATOS Y GRAFICAR 31
3.7 Programación en Arduino..... 31
3.8 Programación Matlab..... 32
3.9 Interfaz grafica 37
3.10 conclusión 40
3.11 REFENRENCIAS 41

1.1 INTRODUCCIÓN

En el laboratorio de diseño electromecánico de la maestría en ciencias en ingeniería mecatrónica se realizó el proyecto, “Desarrollar un sistema que permita saber el estado de carga del banco de baterías de un vehículo eléctrico”, del cual se presenta este reporte que conforme se irá desarrollando el proyecto.

En el presente informe se ira desarrollando partes comenzando con la primera que ocupa los planteamientos principales del problema, la justificación, como parte de los antecedentes que se hará mención sobre el vehículo eléctrico. Siguiendo el orden pasaremos con el marco teórico que más bien será un breve repaso de los materiales usados, sensores, librerías, lenguaje de programación que se implementará, tanto como los elementos de hardware como del software. Al termino se tomarán temas de los aspectos generales tales de donde partimos para comenzar con el desarrollo, la instrumentación, los sensores que usaríamos para poder tomar los datos requeridos para el análisis de la carga de la batería.

1.2 PLANTEAMIENTO DEL PROBLEMA.

Nos adentramos a una de las situaciones más relevantes del mundo actual que es la contaminación por medio del dióxido de carbono que emiten los vehículos a gasolina, por lo cual se está viendo la manera minorizar la contaminación que actualmente hay en la sociedad; por ello se está tomando alternativas de mejora para poder aprovechar las fuentes de energía autosustentables, de esta manera se están implementando los vehículos eléctricos que funcionan con energía limpia es menos contaminante que un motor a gasolina. Actualmente en el mercado existen diversos modelos de vehículos eléctricos que son alternativas para el uso de transporte, carga etc. que contamina menos, podemos encontrar modelos como autos en 4 ruedas, motocicletas, entre otras opciones que tenemos para poder elegir. Actualmente en el municipio de Ocozocoautla de Espinosa, Chiapas; lugar donde se encuentra la fábrica de motos eléctricas INVEMEX están diseñando motos

que ayudan al ambiente a no contaminar. La cual nos da la problemática de que el vehículo eléctrico de la compañía INVEMEX no cuenta con un sistema de monitoreo de la carga de la batería, la corriente consumida.

1.3 JUSTIFICACIÓN

El transporte es la actividad comercial más demandante actualmente ya sea público o privado a nivel mundial, además de su consumo energético que esta genera no tan lejos también encontramos la industria energética; que gran parte de este consumo es llevada a cabo por vehículos de motores de combustión o motores a gasolina; cierto uso remota desde los años anteriores según estadísticas, que en aumento a la demanda automotriz va en aumento la cual trae consigo grandes emisiones de efecto invernadero que afecta nuestra capa de ozono.

En México según SEMARNAT 22% es el transporte y 21% es la industria energética, que juntas es más del 40% del invernadero nacional, la masa de los vehículos con motores a gasolina es 15 o 20 veces mayor que la carga que transporta en pasajeros, lo que significa una eficiencia menos del 1%.

Para cual caso se están desarrollando nuevas y mejores alternativas de para poder eliminar el efecto invernadero que cada día está en aumentó debido a las enormes cantidades de carros que circulan diariamente por las calles. Y para enfrentar la situación los VE (vehículos eléctricos), que tienen hasta 4 veces más eficiencia que los vehículos a gasolina, además de cero emisiones. Un motor de un VE es sometido a diferentes condiciones de trabajo que se pueden emplear en la vida cotidiana como transporte, carga, entre otros.

Los vehículos eléctricos son alternativas que día con día se vuelven una de las soluciones más acertadoras ya que no por común no contaminan y no generan dióxido de carbono que daña nuestra capa de ozono. Debido a eso es una posible solución al problema de la contaminación, además se puede hacer que los vehículos sean más efectivos y tengan un óptimo rendimiento para que en un futuro sea una opción viable.

1.4 ANTECEDENTES

El inicio de la historia del vehículo eléctrico es anterior a la de su homólogo de combustión interna. Era el siglo XIX, y los avances en electromagnetismo, así como los experimentos de Ányos Jedlik, que en 1828 desarrolló el primer motor eléctrico formado por un estator, un rotor y un conmutador, o Joseph Henry, pronto facilitaron que algunos hombres adelantados a su época probaran a montar este tipo de motores a sus coches de caballos, y así fue como el norteamericano Thomas Davenport construyó el que se dice que es el primer vehículo eléctrico de la historia, adelantándose 50 años al nacimiento del motor de combustión (Escorza Rodríguez, 2007).

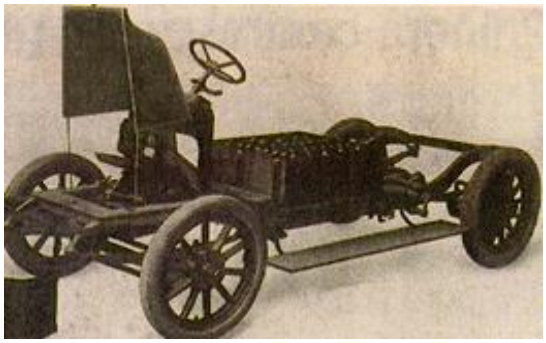


Ilustración 1.1 Primer vehículo eléctrico

A principios del Siglo XX, existían en el mundo 19 fabricantes de vehículo eléctrico, y los datos de ventas en Estados Unidos reflejaban la tendencia dominante; de los 4.200 vehículos vendidos, el 38% eran alimentados por electricidad, mientras que un 40% eran de vapor y el 22% restante recurría a la gasolina. Precisamente fue este país el que recogió el testigo de líder en movilidad eléctrica.

Ilustración 1.2 Fabrica vehículos de combustión

Grandes fabricantes americanos de la época fueron

En 1852, Gaston Planté inventó la batería recargable de plomo y ácido. Pero su fabricación a nivel industrial no era posible. Fue en 1880 que Camille Faure inventó un procedimiento electroquímico llamado masa activa que aumentaba la capacidad de carga de la batería de Planté. La fabricación a nivel industrial de la batería recargable de plomo y ácido sería a partir de entonces una realidad comercial. El

poder recargar la batería hizo que el coche eléctrico se impusiera como el automóvil por excelencia a principios del siglo XX.

El coche eléctrico es tan antiguo como el propio automóvil y a principios del siglo XX parecía ser el futuro.

En la década de 1890 en Europa, el fabricante austriaco de carruajes Jacob Lohner estaba convencido que la era de los carruajes tirados por caballos llegaba a su fin. Lo tuvo claro al volver de un viaje a Estados Unidos y deseaba convertir su empresa en fabricante de automóviles, tanto eléctricos como de motores de combustión interna. Así, le encargó a un joven ingeniero que trabajaba en Viena, un tal Ferdinand Porsche, la creación de lo que sería esencialmente un coche eléctrico.

Ilustración 1.3 Lohner Porsche eléctrico (y tracción delantera) de 1898.

Lohner pensaba que se vendería mejor un coche eléctrico, pues a muchos clientes potenciales no les gustaban los humos ni el ruido de los primeros coches con motor de combustión interna. En 1898, Ferdinand Porsche desveló lo que sería su primer coche, el Egger-Lohner P1. Era capaz de alcanzar 34 km/h y recorrer hasta 79 km con una carga. El P1 sería todo un éxito para Lohner. Le seguirían una multitud de modelos eléctrico e incluso híbridos, como el Semper Vivus, fruto de la colaboración de Ferdinand Porsche con Lohner.

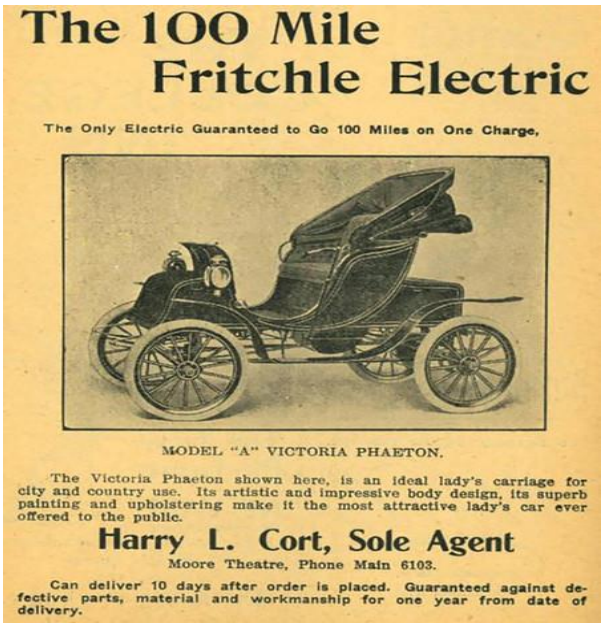
Ilustración 1.4 El "One Hundred Mile Fritchle" o el Tesla Model S de 1908

Lo que vio Jacob Lohner al otro lado del Atlántico fue el auge imparable de los coches eléctricos. El primero de ellos, se vendió en 1890 por William Morrison of Des Moines, Iowa. Pero el líder indiscutible del mercado de la época era Fritchle, fundada por Oliver O. Fricthle, un químico instalado en Denver. Fritchle ganó fama al arreglar las baterías de los automóviles de la zona, pero también se dio cuenta que podría mejorar las baterías que le traían y por tanto crear un mejor coche.

Fritchle vendió su primer coche en 1906 y en 1908 la Oliver P. Fritchle Company abrió su primera tienda en Colfax Avenue, Denver. Para darse a conocer, Fritchle aseguraba que su coche podía recorrer hasta 100 millas (160 km) en llano tras recargar su batería toda la noche. Como nadie le creyó, se montó su particular road-trip demostrando así la veracidad de su anuncio. Los pedidos para el "One Hundred Mile Fritchle" empezaron a llegar desde todos los rincones del país.

Ilustración 1.5 Oliver P. Fritchle frente a uno de sus coches. (Foto: History Colorado)

Fritchle había diseñado su coche pensando en las damas de la alta sociedad de las Rocky Mountains. Y es que, dicen, que las mujeres preferían los coches eléctricos a los de gasolina porque eran sencillamente más limpios. Así, el coche de Fritchle era espacioso y podía subir las fuertes pendientes de la región de Denver. De ahí su excelente autonomía de hasta 100 millas en llanuras. En 1912, Fritchle abrió una tienda en la Quinta Avenida de Nueva York. Su factoría todavía no había empezado a producirlos en serie que ya había una lista de espera para hacerse con un modelo. Eso sí, los Fritchle eran coches de lujo. Mientras que un Ford con motor de combustión de la época costaba el equivalente de 14.000 dólares actuales, un Fritchle costaba el equivalente de unos 105.000 dólares actuales.



Por cierto, si esta historia te suena es porque básicamente tiene unas extrañas (¿o preocupantes?) similitudes con la historia de Tesla (la autonomía y los viajes que hace posible, el precio, la lista de espera o de reservas, etc). Vamos, que no hay nada nuevo. Los progresos del automóvil eléctrico se hicieron patentes con el récord de velocidad del belga Camille Jenatzy y su coche eléctrico “La Jamais Contente” conseguido en 1899: fue el primer ser humano en superar los 100 km/h y a partir de ahí empezaría una lucha por ser el más rápido sobre ruedas. En 1900 se fabricaron 4.192 coches en Estados Unidos y el 28 % de esa producción eran coches eléctricos.

Actualmente, los combustibles más utilizados don los derivados del petróleo y el gas natural como la gasolina. Aunque existen otros tipos de combustibles que son utilizados como los biocombustibles (bioetanol o biodiesel).

Algunas de las ventajas que podemos encontrar en los vehículos a gasolina son los siguientes:

- Posibilidad de obtener grandes potencias de torque y arrastre.
- Gran autonomía de funcionamiento debido a que funcionan a través de hidrocarburos.

Por otro lado, los vehículos eléctricos los motores son eléctricos en lugar de un motor a gasolina, por eso tiene una gran diferencia entre ambos vehículos.

- El motor de gasolina es remplazado por uno eléctrico

- El motor es mucho más pequeño con respecto a uno de gasolina.
- Tiene un controlador que hace las variaciones de cambio de velocidad mediante pulsos.
- Contiene un banco de baterías para su alimentación y entrega la potencia requerida.

Algunas ventajas que tenemos de los vehículos eléctricos son:

- Muchas fuentes energéticas.
- Cero emisiones casi nulas.
- Alta eficiencia.

Pero por otro lado todo lo bueno también tiene sus desventajas.

- Tenemos muy poca autonomía.
- Poca demanda del producto.
- Su costo es un poco elevado, pero su mantenimiento no.

Actualmente tenemos varias opciones de donde podemos adquirir un vehículo eléctrico ya que muchas han comenzado con esa iniciativa de fabricarlos tal el nuestro caso que en Chiapas tenemos uno fábrica muy conocida que se ubica en Ocozocoautla de Espinosa, Chiapas. Actualmente fabrica bajo pedidos y ajuste a lo que el cliente pida es decir que pueden diseñar el modelo requerido. Entre los mas comunes podemos encontrar para recolector de basura, taxis, transporte de carga, entre otros.

Ilustración 1.6 Carro eléctrico recolector de basura

1.5 OBJETIVOS

1.5.1 Objetivos generales.

Desarrollar un sistema que permita saber el estado de carga del banco de baterías de un vehículo eléctrico.

1.5.2 Objetivos específicos:

- Determinar mediante el filtro de Kalman la estimación real de la carga de la batería.
- Estimar el porcentaje de voltaje y corriente del banco de baterías.
- Desarrollar una interfaz que permita ver el estado del banco de baterías.
- Incluir un giroscopio para poder medir la inclinación en ciertos terrenos donde se realizan las pruebas correspondientes.
- En base a los valores obtenidos poder graficarlo mediante Matlab o Python para ver cual es el comportamiento de la carga de las baterías.

1.6 ALCANCE Y LIMITACIONES

1.6.1 alcances

- se obtendrán mediciones de consumo de corriente, voltaje e inclinación del vehículo del terreno donde se realizará las pruebas.
- Se diseñará el sistema SCADA mediante la UDOO-X86.
- Se tendrá un modelo de un mototaxi proporcionado por la empresa INVEMEX que nos facilitará la recolección de datos.

1.6.2 limitaciones

- Tiempo comprendido en el que se realiza la residencia profesional.

1.7 LUGAR EN EL CUAL SE OBTENÍAN LOS DATOS.

Actualmente en el Instituto Tecnológico de Tuxtla Gutiérrez, se tiene con suficiente espacio en el cual se tomo muestras; considerando que se trabajó en el laboratorio que corresponde a la Maestría en Ciencias en Ingeniería Mecatrónica, el laboratorio cuenta con el equipo necesario para poder realizar nuestros practicas e investigación.La universidad se encuentra ubicada en el centro de Chiapas, la dirección exacta es carretera panamericana Km. 1080 C.P. 29050.

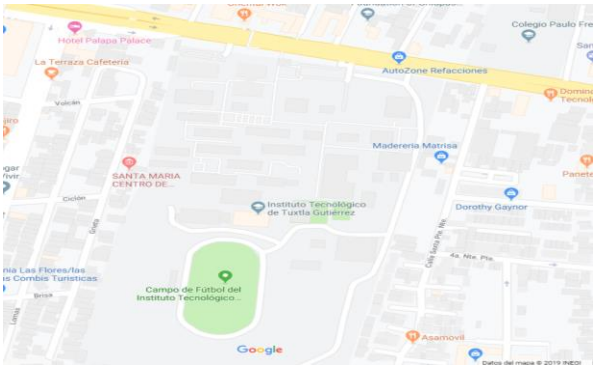


Ilustración 1.7 Lugar donde se realizó el recorrido

2.1 MARCO TEORICO

2.1.1 Udoo-X86

UDOO X86 Single Board son computadoras poderosas, con tecnología Intel, Linux / Windows / Android con un módulo compatible con Arduino 101 integrado en la misma placa. El X86 puede ejecutar software disponible para el mundo de las PC, incluidos juegos, transmisión de video, editores gráficos y plataformas de desarrollo profesional. El X86 también puede ejecutar todo el software para el mundo Arduino ™ 101, incluidos todos los bocetos, bibliotecas y el IDE oficial de Arduino 101.

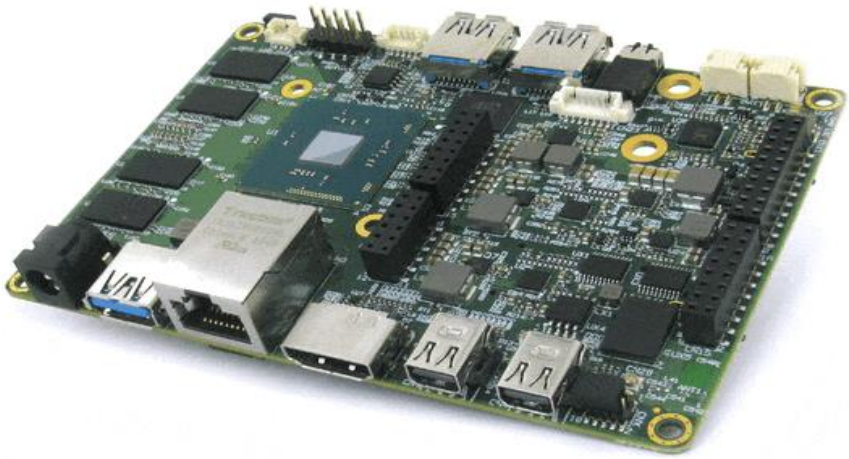


Ilustración2.1 Udoo-x86

El X86 se basa en los procesadores x86 de nueva generación Quad-Core de 64 bits fabricados por Intel. Este procesador fue diseñado para el dominio de PC. El UDOO X86 es ideal para un reproductor multimedia muy bueno y asequible, pero las posibilidades son ilimitadas. El UDOO X86 ULTRA y PLUS vienen con 32 GB de almacenamiento eMMC.

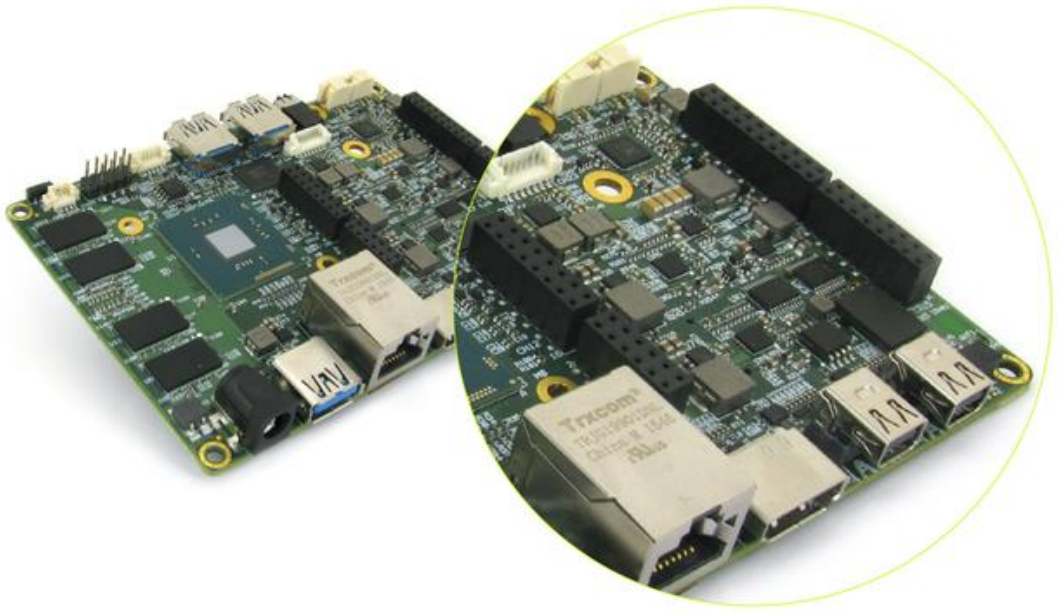


Ilustración 2.2 Udoo-x86, pines de arduino 101

El UDOO X86 incorpora el nuevo Arduino ™ 101, con todas las actualizaciones, incluido el acelerómetro y giroscopio de 6 ejes y la conectividad Bluetooth de baja energía. La plataforma a bordo Arduino ™ compatible está conectada con el procesador principal a través de un puerto USB interno. En términos de software, UDOO X86 es compatible con el IDE oficial de Arduino 101 y con todos los bocetos, tutoriales y recursos disponibles en la comunidad de Arduino ™ 101.

Hablando de hardware, UDOO X86 tiene el mismo diseño PINOUT de Arduino 101. En otras palabras, UDOO X86 es compatible con todos los escudos, sensores y actuadores compatibles con Arduino 101. El voltaje de operación de la placa y la E / S son 3.3V, pero todas las clavijas también están protegidas contra una sobretensión de 5V.

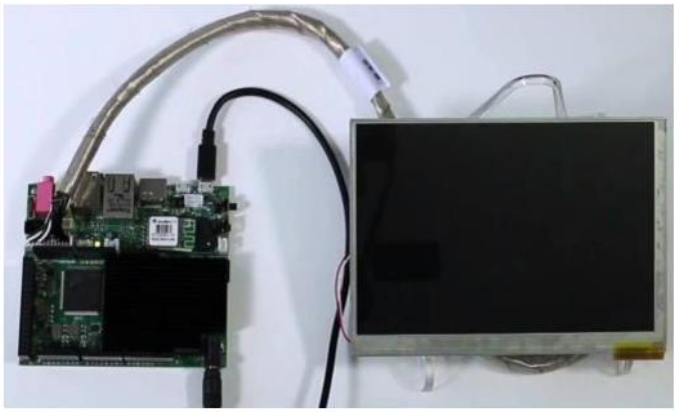


Ilustración 2.2 udoo x86 conexión a pantalla

2.1.2 Arduino

Arduino es un hardware libre. Los diseños de referencia de hardware se distribuyen bajo licencia Creative Commons Attribution Share-Alike 2.5 y están disponibles en el sitio web de Arduino. Los esquemáticos y archivos de montaje de componentes (PCBs) para algunas versiones de placas también están disponibles.

La mayoría de las placas Arduino constan de un microcontrolador AVR Atmel-8 bits (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560), cada microcontrolador consta de diversas cantidades de memoria flash, pines y funciones. Las placas utilizan pines/cabezales hembra de una o dos hileras que facilitan las conexiones e incorporación en otros circuitos.

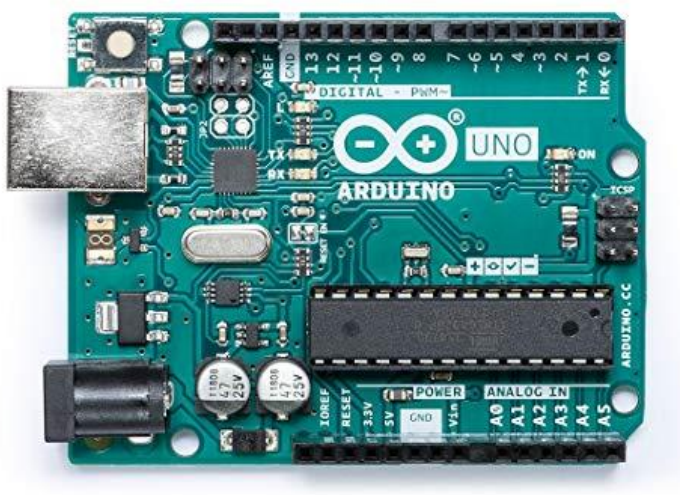
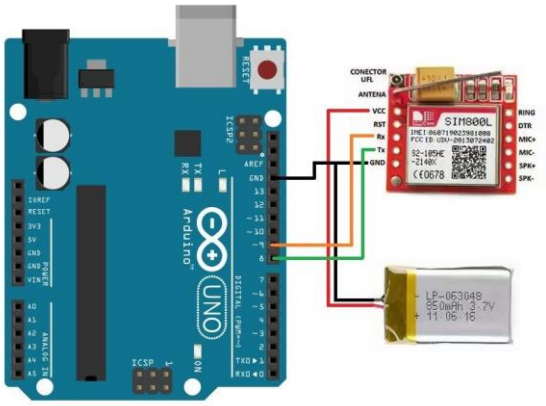


Ilustración 2.3 Arduino uno

Las placas Arduino pueden conectarse con módulos adicionales denominados *shields* (*escudos*, por su traducción al español), dichos *shields* aumentan las características técnicas de la placa Arduino en uso, debido a que poseen circuitos específicos que añaden una o más funcionalidades extras a la placa Arduino nativa en la cual se utilice, también se les conoce como *placas de expansión*. La mayoría de estos shields se conectan a través de un bus serie I^2C, aunque existen también aquellas que emplean conexión mediante el

bus UART (Universal Asynchronous Receiver-Transmitter, por su traducción al español *Transmisor-Receptor Asíncrono Universal*), así como con el bus SPI (Serial Peripheral Interface, por su traducción al español *Interfaz Periférica Serie*).



La mayoría de las placas incluyen un regulador lineal de 5 V y un oscilador de cristal de 16 MHz, o un resonador de cerámica según sea el caso. Algunos diseños, como el LilyPad, funcionan a 8 MHz y prescinden del regulador de voltaje a bordo debido a restricciones de factor/tamaño de forma específicas.

Ilustración 2.4 Arduino con modulo GMS

Los modelos de Arduino se categorizan en placas de desarrollo, placas de expansión (*shields*), kits, accesorios e impresoras 3D.

- Placas: Arduino Galileo, Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre (En Desarrollo), Arduino Zero, Arduino Micro, Arduino Esplora, Arduino Mega ADK, Arduino Ethernet, Arduino Mega 2560, Arduino Robot, Arduino Mini, Arduino Nano, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, LilyPad Arduino USB, Arduino Pro Mini, Arduino Fio, Arduino Pro, Arduino MKR1000/Genuino MKR1000, Arduino MICRO/Genuino MICRO, Arduino 101/Genuino 101, Arduino Gemma.
- Placas de expansión (*shields*): Arduino GSM Shield, Arduino Ethernet Shield, Arduino Wi-Fi Shield, Arduino Wireless SD Shield, Arduino USB Host Shield, Arduino Motor Shield, Arduino Wireless Proto Shield, Arduino Proto Shield.
- Kits: The Arduino Starter Kit, Arduino Materia 101.Accesorios: Pantalla LCD TFT, Adaptador USB/Serie y MiniUSB/Serie, Arduino ISP.
- Impresoras 3d: Arduino Materia 101.

2.1.3 Matlab

MATLAB (abreviatura de **MATrix LABoratory**, "laboratorio de matrices") es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Ilustración 2.5 Logotipo de Matlab

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink(plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas* (*toolboxes*); y las de Simulink con los *paquetes de bloques* (*blocksets*).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

2.1.4 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Ilustración 2.6 Símbolo de Python

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostiles a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

En la interfaz de Python tenemos algunas de sus librerías mas usadas que son las más conocidas actualmente. Python se ha convertido en uno de los lenguajes con más demanda laboral hoy en día. Muchas empresas buscan candidatos que sepan programar en este lenguaje de programación tan popular. Y es que una de las peculiaridades de Python es su flexibilidad. Con este lenguaje seremos capaces de

crear tanto aplicaciones de escritorio como aplicaciones web y todo bajo el mismo entorno.

1. Request: La librería HTTP más famosa que está desarollada por Kenneth Reitz. Es un must-have para todos los desarrolladores de Python.
2. Scrapy: Si estás involucrado en webscraping, esta es una biblioteca que debes manejar sí o sí. Después de utilizar esta biblioteca no utilizarás ninguna otra relacionada.
3. wxPyhton: Un gui toolkit para Python. A esta, en mis propios proyectos, la he priorizado por encima de tkinter. Simplemente, la adorarás.
4. Pillow: Un amistoso fork de PIL (Python Imaging Library). Es mucho más sencillo de utilizar que la propia PIL y se convierte en toda una necesidad para aquellos programadores que trabajen con imágenes.
5. SQLAlchemy: Una biblioteca muy polémica para gestionar bases de datoss. Muchos la aman y muchos la odian. La decisión es cosa tuya.
6. BeautifulSoup: Sé que es lenta, pero esta librería para parsear código de xml y html es muy útil para aquellos que están comenzando a programar en Python.
7. Twisted: La herramienta más importante para cualquier desarrollador de aplicaciones de red. Cuenta con una API muy, pero que muy bonita y es utilizada por una gran cantidad de desarrolladores de Python famosos.
8. NumPy: De esta librería es muy difícil escapar. Proporciona algunas funcionalidades matemáticas avanzadas para Python.
9. SciPi: Si hablamos de NumPy entonces tenemos que hablar también de SciPy. Es una biblioteca de algoritmos y herramientas matemáticas que ha ocasionado que muchos científicos se cambien de Ruby a Python.
10. matplotlib: Una biblioteca de trazado numérico. Es muy útil para cualquier científico de datos o cualquier analizador de datos.
11. Pygame: ¿A qué programador no le gusta echarse unas partiditas a un videojuego? Con la librería Pygame podrás desarrollar juegos en 2D a la antigua usanza.

12. Pyglet: Un motor de animación y creación de juegos en 3D. Este es el motor con el que se desarrolló el juego que más dinero ha generado en Youtube, Minecraft.
13. PyQt: Un conjunto de herramientas GUI para Python. Es mi segunda elección después de wxPython para el desarrollo de interfaces gráficas de usuario para mis scripts en Python.
14. PyGTK: Otra biblioteca GUI para Python. Es la misma biblioteca con la que se creó el famoso cliente Bittorrent.
15. Scapy: Un sencillo analizador de Python desarrollado con Python. Todo muy meta.
16. pywin32: Una biblioteca de Python que proporciona algunos métodos y clases útiles para interactuar con el sistema operativo Windows.

2.1.5 Sensor de corriente de efecto Hall en lazo cerrado

Ilustración 2.7 Sensor T201DHC100

T201DCH100 es aislado, sin contacto transductor actual accionado un lazo de AC/DC con el tamaño muy compacto (dimensiones totales menos de 96.5 x 68 x 26 milímetros). La función y la mirada del dispositivo es muy similares a ésas de un estándar activo CT, pero con la característica notable de medir el componente de la C.C. y de CA de TRMS de la corriente.

Para su resistencia eléctrica, las dimensiones de empleos fáciles y compactos, el transductor T201DCH100 cumplió cada clase de medida actual hasta 100 ADC/Aac y varios usos (baterías, cargadores de batería, los paneles solares, unidades de energía, C.C. y cargas de CA etc.) (Direct INDUSTRY, 2014).

Salida de voltaje de señal con respecto a la corriente de entrada.

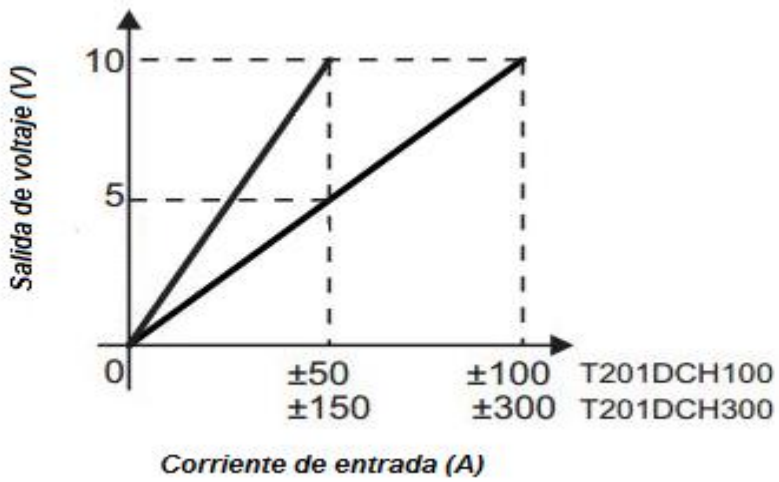


Ilustración 2.8 Respuesta del sensor

2.1.6 Motor *brushless*

Los motores *brushless* están compuestos por una parte móvil que es el rotor, que es donde se encuentran los imanes permanentes, y una parte fija, denominada estator o carcasa, sobre la cual van dispuestos los bobinados de hilo conductor. La imagen refleja una sección de uno de estos motores en donde puede verse la disposición de los bobinados y los imanes permanentes (que en este caso son de neodimio).

Ilustración 2.9 Motor brushless 60V

En este tipo de motor la corriente eléctrica pasa directamente por los bobinados del estator o carcasa, por lo tanto, aquí no son necesarios ni las escobillas ni el colector que se utilizan en los *brushed*. Esta corriente eléctrica

genera un campo electromagnético que interacciona con el campo magnético creado por los imanes permanentes del rotor, haciendo que aparezca una fuerza que hace girar al rotor y por lo tanto al eje del motor. No se tiene ni escobillas, ni colector y delgas; por lo que el elemento que controla que el rotor gire, sea cual sea su posición, es el variador electrónico; lo que hace básicamente es ver en qué posición se encuentra el rotor en cada momento, para hacer que la corriente que le llegue sea la adecuada para provocar el movimiento de rotación que le corresponde.

2.3.7 ESC: Variador de motor

Los variadores son los elementos que se encargan de manejar los motores eléctricos y se les denomina por las siglas **ESC** (*Electronic Speed Controller*).

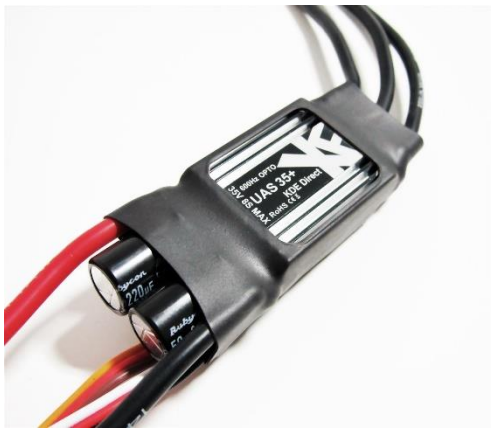


Ilustración 2.10 Variador de velocidad (ESC)

Los motores brushless que cuentan con un bobinado especial semejante al de los motores industriales trifásicos y aplicando mucha más frecuencia; estos motores no se pueden conectar directamente a la batería, sino que requieren de un circuito electrónico que los maneje: el variador ESC.

Los variadores pueden llegar a ser muy sofisticados y manejar capacidades de corriente muy elevadas (llegan a alcanzase los 200A en modelos extremos).

Ilustración 2.11 ESC 2000W

2.1.8 BATERIAS

Las baterías que se emplean en este carro eléctrico don de Ion Litio T-1275 PLUS que son baterías recargables de larga duración la cual en el banco de baterías se tenían 5 de 12V cada una, sumando un total de 60V todas conectadas en serie.

Ilustración 2.12 Batería empleada Trojan

La marca Trojan undada en 1925 por George Godber y Carl Speer, Trojan Battery Company es la fábrica líder a nivel mundial de baterías de ciclo profundo. Desde baterías de electrolito líquido de ciclo profundo a baterías de Gel y AGM de Ciclo Profundo, Trojan ha modelado la tecnología del mundo de las baterías de ciclo profundo con más de 90 años de experiencia en la fabricación de baterías.
Con la invención de la batería para carros de golf para el vehículo Autoette en 1952, Trojan fue el primero en promover el desarrollo de la tecnología de baterías de ciclo profundo para la industria del golf e introdujo de manera exitosa la movilización en el juego de golf. Para Trojan, esto inició un legado de liderazgo e innovación que prevalece hasta hoy en los mercados globales del segmento de aplicaciones de ciclo profundo para plataformas aéreas, transporte, energía renovable, golf, máquinas para limpieza de Suelos, marina y vehículos de recreo. En la actualidad, las baterías Trojan están disponibles en todo el mundo a través de nuestra red global de distribuidores maestros

Con sede en Santa Fe Springs, CA, las operaciones de Trojan incluyen plantas de fabricación con certificación ISO 9001:2008 en California y Georgia, dos centros de investigación y desarrollo de tecnología avanzada dedicada de manera exclusiva a las tecnologías de batería de ciclo profundo y oficinas internacionales localizadas en Europa, Emiratos Árabes Unidos y Asia. Trojan es miembro del Consejo Internacional de Baterías (BCI por sus siglas en inglés) y coopera en investigaciones técnicas con la Academia de Ciencias de Bulgaria. Todos los procesos de Trojan se apegan a los estándares del BCI y de la Comisión Internacional Electromecánica (IEC por sus siglas en inglés).

PRODUCT SPECIFICATIONS

BCI GROUP SIZE	TYPE	CAPACITY [A] Minutes			CAPACITY [B] Amp-Hours (AH)				ENERGY (kWh)	TERMINAL Type [E]	DIMENSIONS [C] Inches (mm)			WEIGHT lbs. (kg)
		@25 Amps	@56 Amps	@75 Amps	5-Hr Rate	10-Hr Rate	20-Hr Rate	100-Hr Rate	100-Hr Rate		Length	Width	Height [D]	
12 VOLT DEEP CYCLE BATTERY - with T2 TECHNOLOGY™														
N/A	T-1275 Plus	280	102	70	120	134	150	166	1.99	1	12.96 (329)	7.13 (181)	10.71 (272)	82 (37)

Ilustración 2.13 Especiaciones de la batería

Conderando que las baterias son recarbles con un cargador especial que acontinuacion les damos unas recomendaciones.

CHARGING INSTRUCTIONS

CHARGER VOLTAGE SETTINGS (AT 77°F/25°C)				
System Voltage	12V	24V	36V	48V
Daily Charge	14.8	29.6	44.4	59.2
Float	13.2	26.4	39.6	52.8
Equalize	15.5	31.0	46.5	62.0

Ilustración 2.14 Ajustes del voltaje del cargador

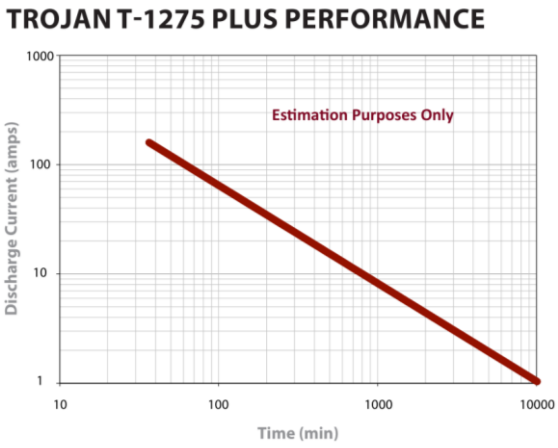


Ilustración 2.15 Desempeño de la batería Trojan

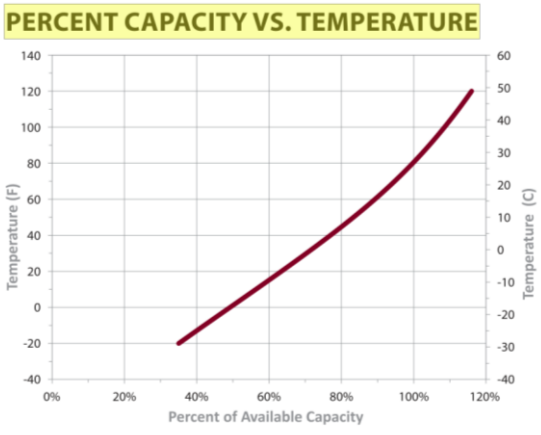


Ilustración 2.16 PORCEN. DE CAPACIDAD VS. TEMPERATURA

3.1 DESARROLLO

3.2 Cronograma de actividades propuesto.

Actividad	Semana															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.-Investigación bibliográfica																
2.- Diseño del sistema de estimación de carga basado en filtro de Kalman																
3.- Construcción del prototipo de tarjeta de diagnóstico																
4.- Diseño del sistema de medición de velocidad y posicionamiento del VE																
5.- Diseño del sistema SCADA																
6.- Pruebas y corrección de errores																

3.3 Descripción de actividades.

1. Investigación bibliográfica: conocer las características de los giroscopios electrónicos
2. Diseño del sistema de estimación de carga basado en filtro de Kalman para estimar en tiempo real la carga de la batería

3. Construcción del prototipo de tarjeta de diagnóstico. Se diseñarán las tarjetas electrónicas con la instrumentación requeridas para los sistemas de monitoreo y diagnóstico. -
4. Diseño del sistema de medición de velocidad y posicionamiento del VE.
5. Conectar el giroscopio al sistema SCADA mediante una tarjeta de adquisición de datos en Python.
6. Realizar pruebas de funcionamiento del giroscopio, a diferentes condiciones de operación del vehículo.

3.4 Uso de la tarjeta UDOO-X86

La tarjeta UDOO –X86 como la tarjeta para adquisición de datos directamente de los sensores con lo que cuenta la moto. El sistema no es tan desconocido ya que es un sistema operativo Ubuntu con el cual ya se está familiarizado. Esta tarjeta es más potente que una rapsberry y una tarjeta Odroid xu4, la tarjeta Udoo-X86 tiene las siguientes características:

- Cuenta con un procesador Quad-Core de 64 bits de Intel.
- Está integrado con un arduino 101 en la misma tarjeta.
- Es compatible con todos actuadores, sensores de arduino 101.
- Viene con 32 GB de almacenamiento eMMC
- Tiene incluido el acelerómetro y giroscopio de 6 ejes y la conectividad Bluetooth de baja energía.
- El arduino está conectado al microprocesador vía USB internamente lo cual facilita el uso del mismo arduino
- El voltaje de operación de la placa y la E / S son 3.3V, pero todas las clavijas también están protegidas contra una sobretensión de 5V.
- Tiene la capacidad de conectar teclado, mouse y pantalla.

3.5 Diseño de placa para adquisición de datos.

Se usó el IDE de Arduino para realizar la toma de las señales provenientes de los sensores que cuenta la moto, así como el giroscopio instalado en la tarjeta. La UDOO X86 no soporta voltajes arriba de los 3.3v de entrada así que se tuvo que diseñar una placa con divisores de voltaje para que pueda leer la señal los pines del Arduino.

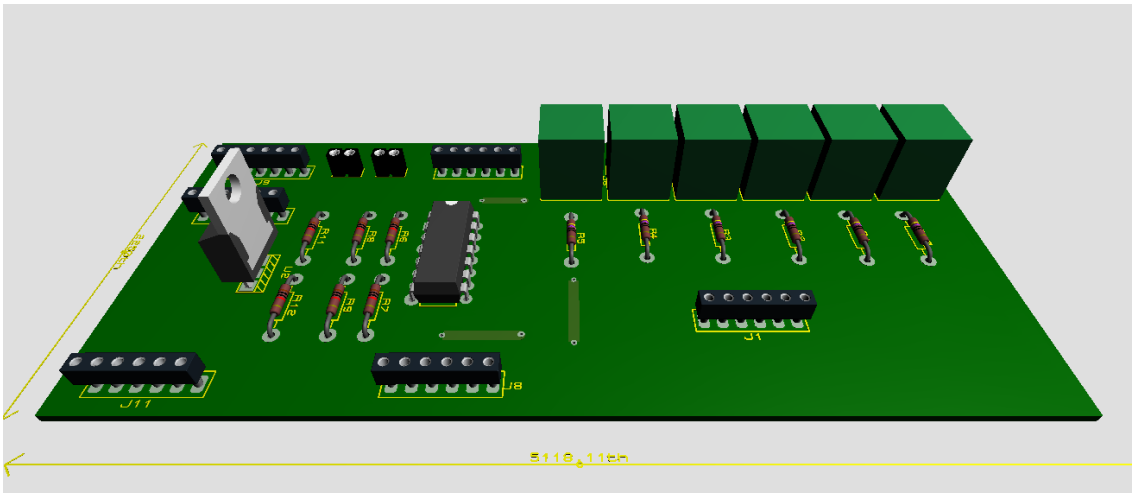


Ilustración 3.1 Placa para adquisición de datos

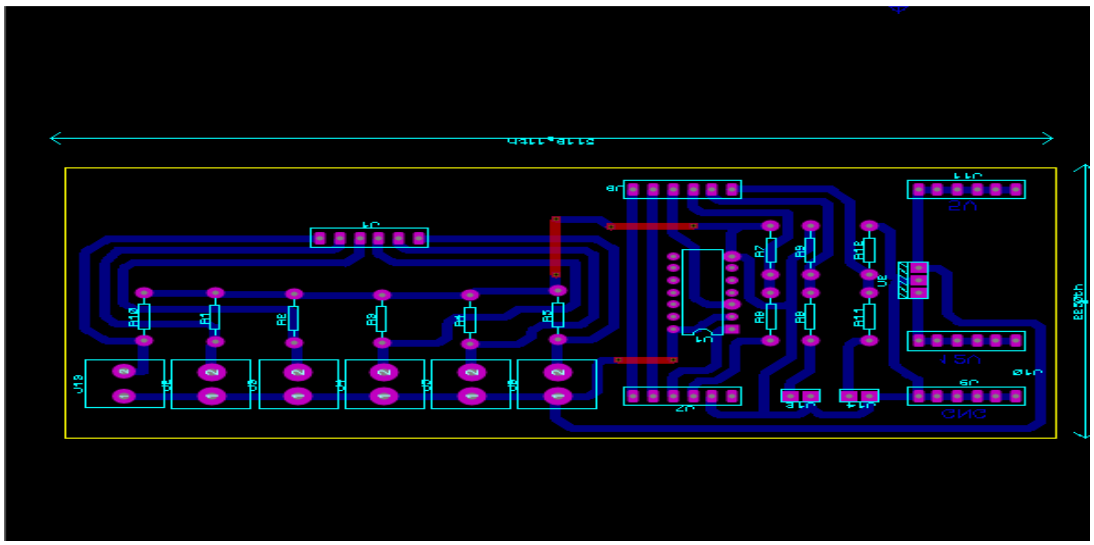


Ilustración 3.2 Diseño del PCB

La placa también se utilizó para usar señales PWM provenientes de la programación de Arduino y así controlar la velocidad de la moto y calcular los picos de corriente que suele tener al momento de acelerar.

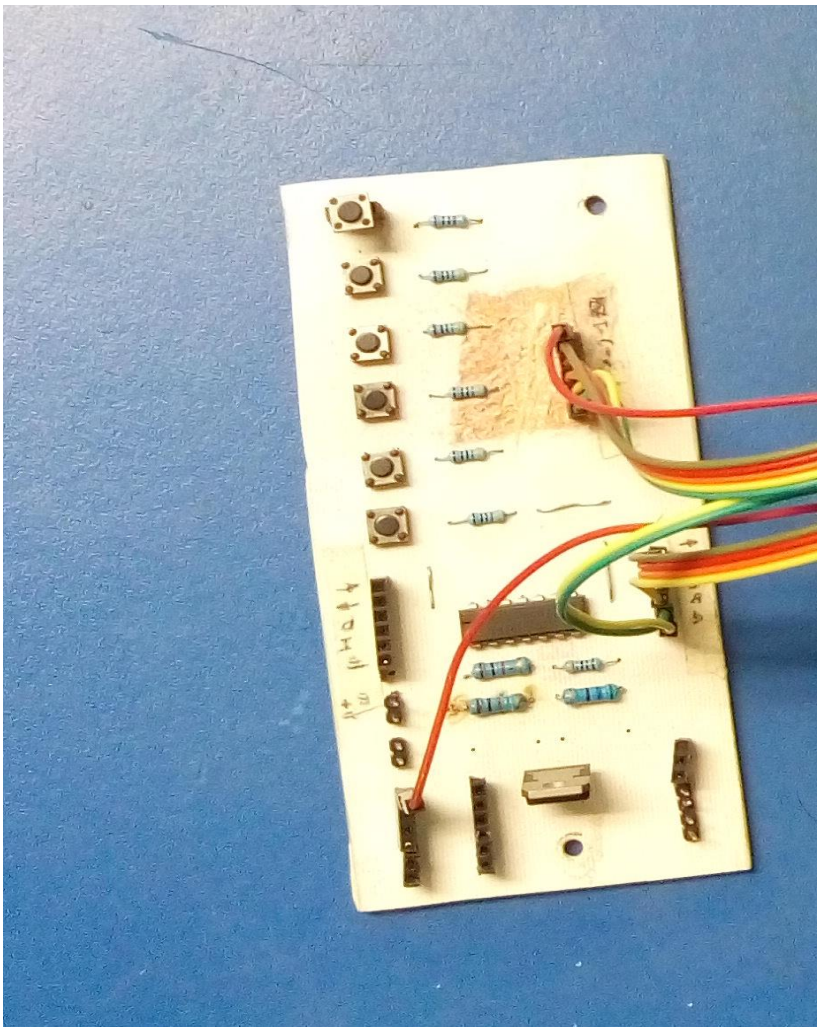


Ilustración 3.3 Placa para ajuste de señales y control de velocidad

La placa está compuesta por dos divisores de voltaje que tiene una entrada de 60v y sale 3.3v para que los pines del Arduino, otros divisores que viene de los sensores que mandan una señal de 10V. Ademas se la incluido unos reguladores de voltaje tales que de la misma placa podemo obtener voltajes, de 5V, 12V para poder emplerar otros sensores posibles que en algun futuro se puedan emplear. Se le an agregado 5 botones que son los encargador de activar las señales PWM, para poder tener un control de la placa mas preciso y suave sin tener que tener presionado el acelerador.

Se realizo los divisores necesarios para que la UDOO pueda leer los valores y no tenga problemas, de dañarse.

Divisor de voltaje

$V_{out}=\frac{V1R2}{(R1+R2)}$

Valores de resistencias propuestos para el divisor a partir de una entrada de 60V

Ventrada= 60v

R1=11800Ω

R2=680Ω

$V_{out}=\frac{(60)(680)}{(11800+680)}$

V_{out}=3.26v

Valores de resistencias propuestos para el divisor a partir de una entrada de 10V.

Ventrada= 10v

R1=1000Ω

R2=470Ω

$V_{out}=\frac{(10)(470)}{(100+470)}$

V_{out}=3.19v

En la placa se puso un regulador de voltaje de 5 volts para alimentar los push botton que sirven para hacer las pruebas con PWM. Todos los sensores van directos al Arduino 101 y de este mismo se abre un programa en Python para adquirir datos este genera un archivo .txt donde están todos los datos y con estos mismo se hace una gráfica en Matlab y en la interfaz en Python.

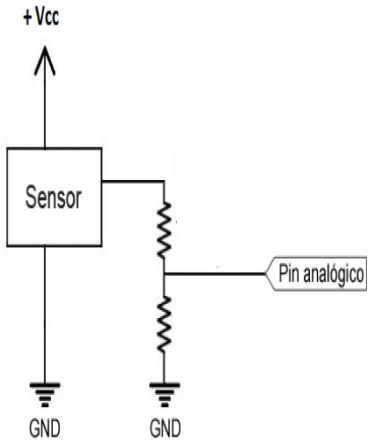


Ilustración 3.4 Ejemplo del divisor de voltaje

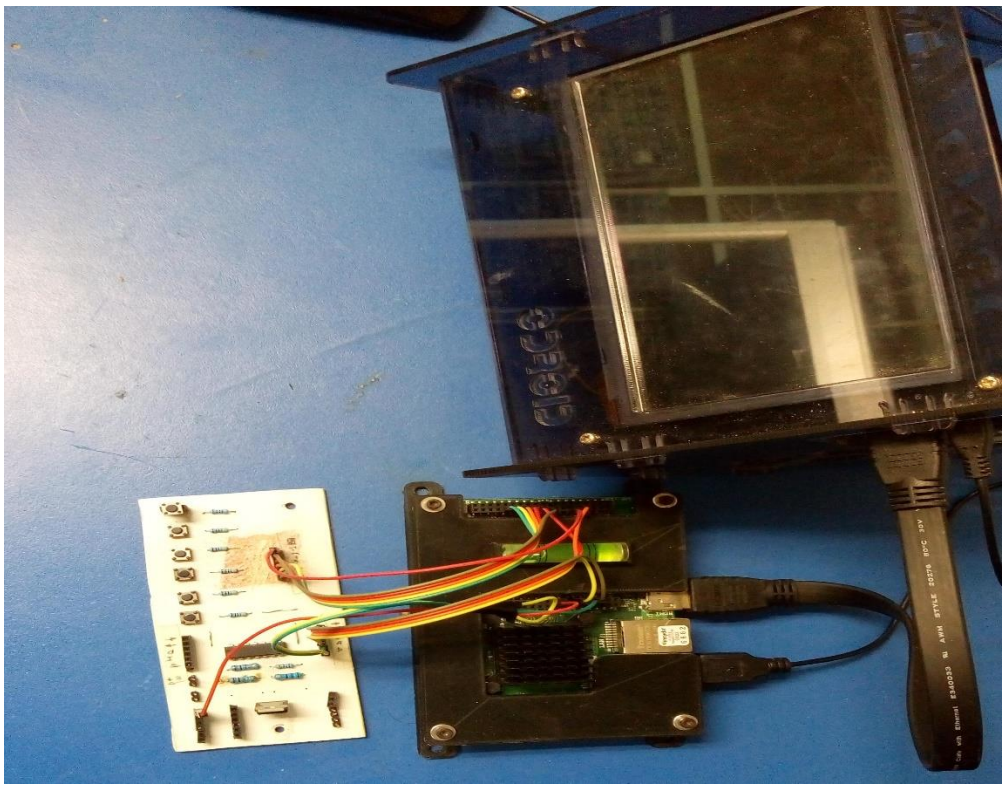


Ilustración 3.5 Tarjeta UDOO-x86 con la placa diseñada

3.6 PROGRAMACIONES PARA ADQUIRIR DATOS Y GRAFICAR

Para poder hacer una gráfica donde se muestre la corriente y el voltaje del banco de baterías se hizo una serie de programaciones con Python, Arduino y Matlab.

3.7 Programación en Arduino.

El Arduino se utilizó para leer las señales de los sensores de la moto, así como la corriente, el voltaje de la batería, temperatura y posición del vehículo en ciertos recorridos que se hizo.

Para poder obtener los datos y el archivo .txt primero ejecutamos el ide de Arduino en el 101 y después se ejecuta la programación en Python, este genera el archivo .txt con los datos obtenidos en las pruebas con la moto.

Los datos generados por el programa de Arduino y de Python, en la tercera y cuarta columna son la corriente y voltaje respectivamente. Se puede observar las variaciones conforme se va tomando las muestras.

0.00	148	0.00	4.93	39.38	1.86	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.96	39.38	1.88	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.68	39.39	1.86	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.87	39.40	1.97	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.90	39.43	1.88	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	5.03	39.44	1.87	0.00	2.00	0.00	3.14	0.00
0.00	148	0.00	5.05	39.46	1.83	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.71	39.48	1.88	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.81	39.49	1.84	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.50	39.52	1.82	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.57	39.54	1.78	0.00	2.00	0.00	3.14	0.00
0.00	148	0.00	4.33	39.56	1.72	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.53	39.59	1.69	0.00	1.20	1.00	1.88	0.00
0.00	148	0.00	4.39	39.62	1.70	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.12	39.64	1.65	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	4.29	39.65	1.64	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.76	39.68	1.67	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.68	39.70	1.81	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.59	39.72	1.68	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.46	39.74	1.65	0.00	1.20	1.00	1.88	0.00
0.00	148	0.00	3.41	39.76	1.66	0.00	1.60	1.00	2.51	0.00
0.00	148	0.00	3.38	39.78	1.68	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.40	39.79	1.59	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.44	39.81	1.69	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.06	39.81	1.78	0.00	1.20	1.00	1.88	0.00
0.00	148	0.00	3.08	39.80	1.72	0.00	1.60	1.00	2.51	0.00
0.00	148	0.00	2.97	39.82	1.67	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	3.15	39.84	1.68	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.79	39.86	1.69	0.00	1.20	1.00	1.88	0.00
0.00	148	0.00	2.66	39.85	1.67	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.78	39.86	1.67	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.73	39.81	1.59	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.57	39.83	1.68	0.00	1.20	1.00	1.88	0.00
0.00	148	0.00	2.55	39.86	1.69	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.41	39.88	1.67	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.34	39.90	1.68	0.00	1.60	0.00	2.51	0.00
0.00	148	0.00	2.31	39.89	1.69	0.00	1.20	0.00	1.88	0.00

Ilustración3.6 Datos obtenidos en tiempo real

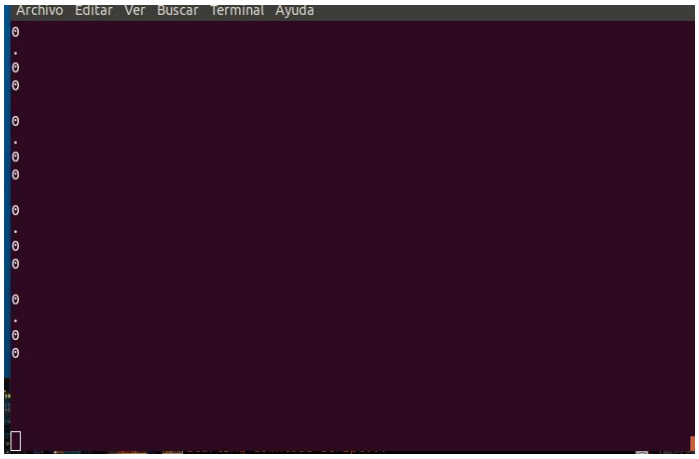


Ilustración3.7 Ejecutable en Python donde toma muestras

Esta es la toma de datos por parte del voltaje y la corriente, estos datos complementan a los tomados por los sensores, tanto cuanto estaba en movimiento o cuando se estaba cargado para poder realizar la estimación de carga, el programa estaba diseñado para recolectar los datos a partir del IDE de Arduino y almacenarlas en un archivo txt, y eso poder graficarlo en el software Matlab o en el mismo Python.

3.8 Programación Matlab.

Como habíamos visto anteriormente la programación anterior, que nos recolectaba los datos obtenidos en la moto después no los guardaba en un archivo. Los archivos .txt generado previamente por las pruebas hechas, estos datos recopilados se grafican con la ayuda de Matlab, ya que es un programa que nos permite graficar datos de valores enteros, para eso se creó el siguiente programa que a continuación les mostraremos.

En esta parte del código se manda a traer el archivo txt. Escribimos la dirección y ubicación del archivo, además del nombre dado caso que se asigne uno.

```
%OTROS DATOS
a21p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  11_55_18
2019.txt');
a22p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  11_56_29
2019.txt');
a23p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  11_58_52
2019.txt');
a31p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  11_59_49
2019.txt');
a32p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_01_06
2019.txt');
a33p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_01_52
2019.txt');
a41p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_03_44
2019.txt');
a42p240=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_04_28
2019.txt');
t=[1:1:300];  adq1=[59:1:358];  adq2=[22:1:321];  adq3=[28:1:327];  adq4=[23:1:322];
adq5=[97:1:396]; adq6=[19:1:318];
adq7=[16:1:315]; adq8=[18:1:317];

a21p230=load('C:\Users\HOME\Desktop\300419\Tue Apr 30 12_06_48 2019.txt');
a22p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_07_22
2019.txt');
%a23p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_08_26
2019.txt');
a31p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_10_39
2019.txt');
a32p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_12_49
2019.txt');
a33p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_13_33
2019.txt');
a41p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_14_11
2019.txt');
a42p230=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_15_03
2019.txt');
a21p220=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_33_56
2019.txt');
a22p220=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_34_46
2019.txt');
ad1=[28:1:327];  ad2=[15:1:314];  ad3=[13:1:312];  ad4=[16:1:315];  ad5=[16:1:315];
ad6=[22:1:321];
ad7=[24:1:323]; ad8=[23:1:322]; ad9=[54:1:353];

adq23=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_35_40
2019.txt');
adq24=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_36_30
2019.txt');
adq25=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_37_38
2019.txt');
adq26=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_38_11
2019.txt');
adq27=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_39_22
2019.txt');
adq28=load('D:\3er  SEMESTRE\Proyectito\importante\300419\Tue  Apr  30  12_40_08
2019.txt');
```

En esta parte graficamos los datos de las variables de interés para este caso son las siguientes.

```
subplot(422);plot(t,MaDatos(t2,1));title('avance');
subplot(427);plot(t,MaDatos(t2,2));title('pwmXD');
subplot(423);plot(t,MaDatos(t2,3));title('corriente');
subplot(424);plot(t,MaDatos(t2,4)*-1);title('imuINT');
subplot(421);plot(t,MaDatos(t2,5));title('VOLTAbat');
subplot(428);plot(t,MaDatos(t2,6));title('veldes');
subplot(425);plot(t,MaDatos(t2,8));title('RPS2');  %8
subplot(426);plot(t,MaDatos(t2,10));title('posicionINST2'); %10
```

Se hace un código con la función “load” para llamar archivos .txt este separa los datos adquiridos y después gráfica. Para este caso es de la corriente y de esa manera se realiza con las otras variables que antes mecionamos.

```
subplot(423);hold on
plot(t,a21p240(adq1,3)); %plot(t,a22p240(adq2,3)); plot(t,a23p240(adq3,3));
plot(t,a31p240(adq4,3)); %plot(t,a32p240(adq5,3));plot(t,a33p240(adq6,3));
%plot(t,a41p240(adq7,3)); %
plot(t,a42p240(adq8,3));
%plot(t,a21p230(ad1,3));      plot(t,a22p230(ad2,3));      plot(t,a31p230(ad3,3));
plot(t,a32p230(ad4,3)); plot(t,a33p230(ad5,3));
%plot(t,a41p230(ad6,3));      plot(t,a42p230(ad7,3));      plot(t,a21p220(ad8,3));
plot(t,a22p220(ad9,3));
title('corriente'); hold off
```

Programa que grafica el voltaje de la bateria.

```
subplot(421); hold on
plot(t,a21p240(adq1,5)); %plot(t,a22p240(adq2,5)); plot(t,a23p240(adq3,5));
plot(t,a31p240(adq4,5)); %plot(t,a32p240(adq5,5)); plot(t,a33p240(adq6,5));
plot(t,a41p240(adq7,5)); %plot(t,a42p240(adq8,5));
%plot(t,a21p230(ad1,5));      plot(t,a22p230(ad2,5));      plot(t,a31p230(ad3,5));
plot(t,a32p230(ad4,5)); plot(t,a33p230(ad5,5));
%plot(t,a41p230(ad6,5));     plot(t,a42p230(ad7,5));     %plot(t,a21p220(ad8,5));
plot(t,a22p220(ad9,5));
title('voltaBAT');hold off
```

Se manda a llamar la variable y con la función “plot” se grafica los datos del .txt

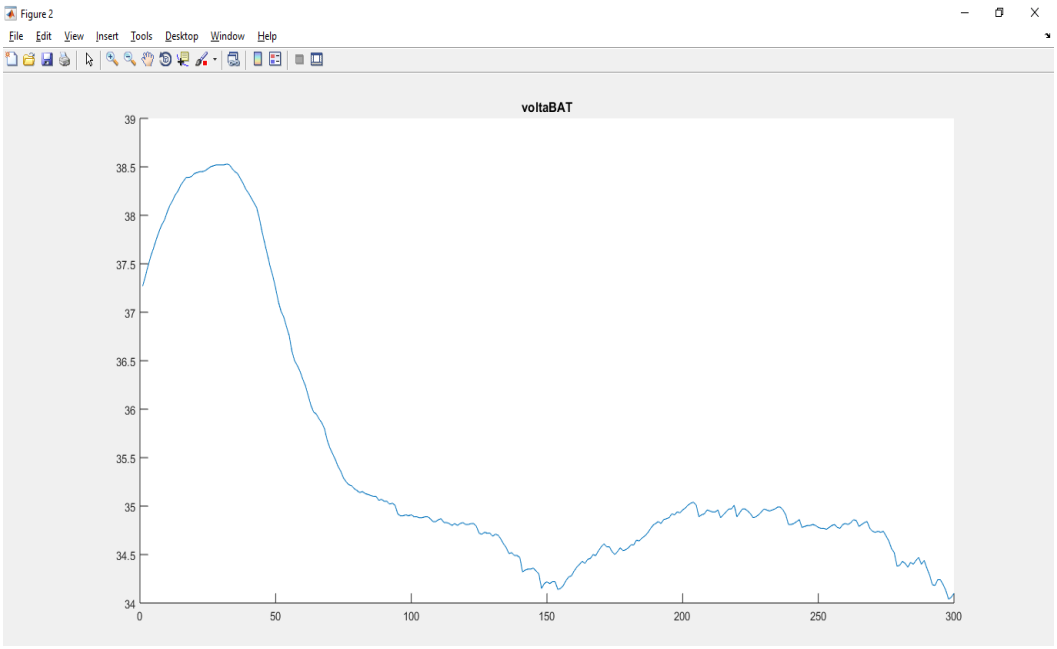


Ilustración 3.8 Grafica de voltaje

En la gráfica de voltaje uno se puede observar la descarga de la batería en un tiempo determinado de muestras, se observa que se descarga en el voltaje límite del fabricante para el buen funcionamiento de las baterías. En esta grafica la señal ya está filtrada.

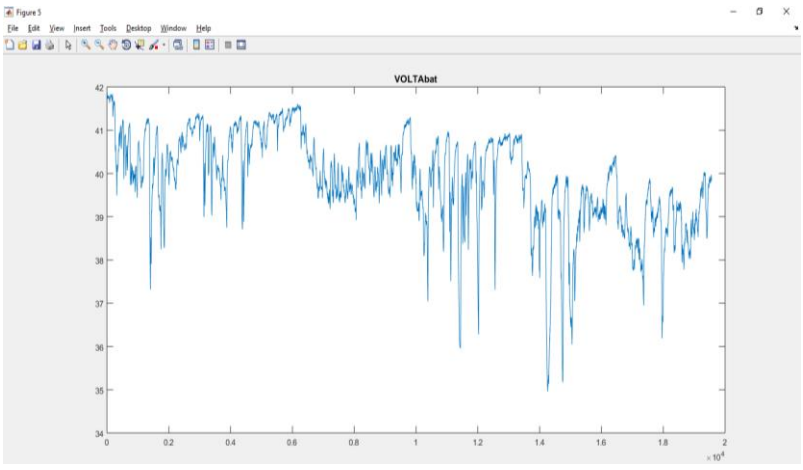


Ilustración 3.9 Grafica del voltaje antes de ser filtrada

Esta grafica del voltaje dos la batería está más cargada y se nota como de apoco se va descargando conforme se va tomando datos. Esta grafica se hizo con más muestras de datos y sin un filtro alguno.

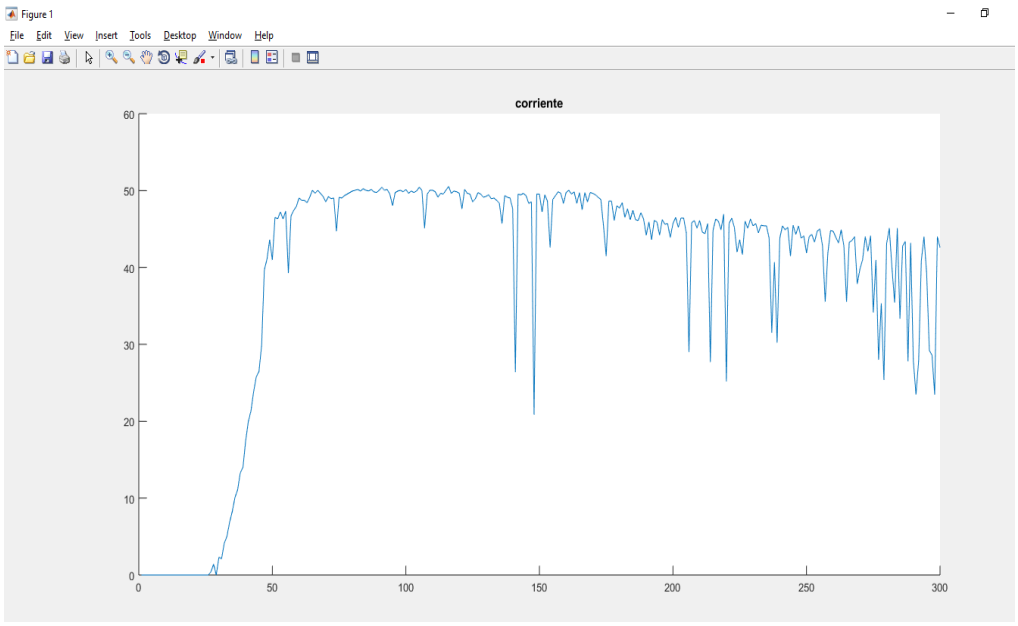


Ilustración 3.10 Grafica de la corriente con sumida filtrada

En la gráfica de corriente se observa las variaciones de la corriente conforme un tiempo de muestras al igual que la de batería, se observa las variaciones de la corriente según se iba descargando las baterías.

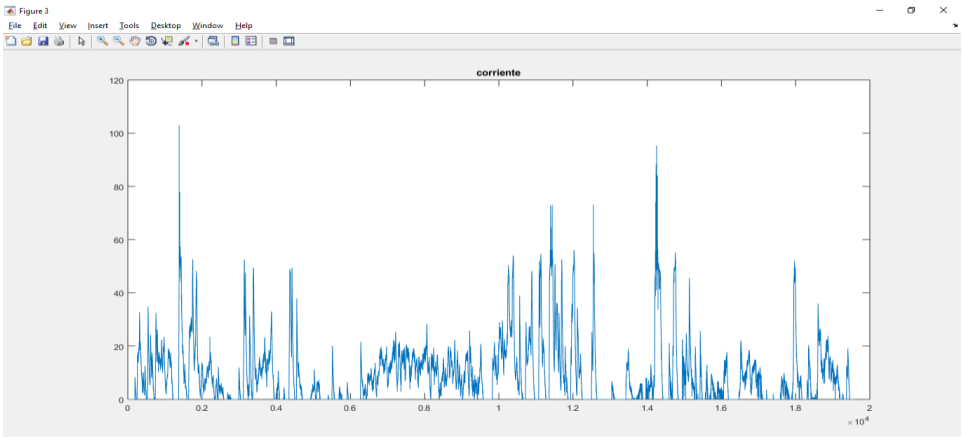


Ilustración 3.11 Grafica de la corriente consumida sin filtro

Al igual que la gráfica de voltaje dos, se realizó con más datos y sin filtro de señal

3.9 Interfaz grafica

La interfaz se realizó con Tkinter, Tkinter es una librería de Python que permite realizar programación de interfaces con orientación a eventos; los elementos de botones, cuadros de dialogo, imágenes, etiquetas, slider, color de fondo, tamaño de letra, color de letras, ubicación, se realiza a través de líneas de código en el que se especifica el formato de cada uno de ellos.

Para parte se mandadan a importar datos de otros codigo que ne neseitaron y las librerias adecuadas para el correcto fucionamiento del codigo.

```
import wiringpi2 as wpi
from Tkinter import*
import pandas as pd
from matplotlib import pyplot as plt
import bateri as pr2
import bateri as br1
import tkMessageBox
import thread
import time

#Obtener los datos de los cuadros de texto entry
def obtener():
        namearchi=dia.get()
        return namearchi

def obtener2():
        try:
                nodatos=muestras.get()
                return nodatos
        except ValueError:
                tkMessageBox.showwarning('')
#Graficar
def obtener3():
```

Ilustración 3.12 Se importan librerías y programas necesarios

Cuando tenemos listo lo anterior en la cual añadimos las librerías y todo lo necesario para el funcionamiento del nuestro código damos paso a la ejecución, es decir que daremos la orden de cada una de las variables que antes se han declarado.

```
#Boton iniciar
def ejecutar():
        global aver
        pr2.adqdatos1(obtener())
        pr2.adqdatos2(obtener2())
        xd2=int(obtener2())

#detiene la toma de muestras
def stop():
        pr2.stop2(1)
#grafica archivos recientes, registros actuales
def actual():
        grafica.set(dia.get())
#Da la orden de leer el archivo prbs
def prbs():
        br1.prbs2(1,0)
        pr2.adqdatos1(obtener())
        pr2.adqdatos2(obtener2())
        xd2=int =100(obtener2())

def prbstop():
        br1.prbsss()
def exit():
        br1.prbs2(0,1)
```

Ilustración 3.13 Se dan instrucciones para la ejecución

En esta seccion podemos ver que este cogido se crean la ventana principal y los botones que se utilizaran dado es este caso se le asignara el nombre del la ventana principal. Cuando se crea un boton o una etiqueta tenemos la opcion de acomodar a nuestro gusto con los comando (place(x=,y=) solo escribimos la coordenadas que queremos para acomodar nuestros botones.

```
ventana=Tk()
ventana.geometry('1080x720')
ventana.configure(bg='silver')
ventana.title('Bateria')
Label(ventana,text='    Analisis de bateria    ',fg='black',bg='silver',font=('Ravie',15)).place(x=150, y=10)

fecha= time.strftime('%d %b %y %H: %M')
dia=StringVar()
dia.set(fecha)
Label(ventana,text='Dia: ',fg='red',bg='silver').place(x=150, y= 100)
Entry(ventana,textvariable=dia,bg='white').place(x=300, y=100)

muestras=IntVar()
muestras.set(8640)
Label(ventana,text='N-o de Muestra x Dia: ',fg='red',bg='silver').place(x=150, y= 170)
Entry(ventana,textvariable=muestras,bg='white').place(x=300, y=170)

grafica=StringVar()
grafica.set(' ')
Label(ventana,text='Graficar datos .txt: ',fg='red',bg='silver').place(x=150, y= 270)
Entry(ventana,textvariable=grafica,bg='white').place(x=300, y=270)
Button(ventana,text='Graficar',fg='black',command=obtener3).place(x=300, y=295)

Button(ventana,text='Iniciar',fg='black',command=ejecutar).place(x=400, y=125)
Button(ventana,text='Llamar txt',fg='black',command=actual).place(x=380, y=295)
```

Ilustración 3.14 Creando los botones necesarios y las etiquetas

A lo mismo, en esta seccion se teminan de crear los botonoes dandole coordenadas, tanto como para X y Y de esa manera ubicamos, al igual insertamos las imágenes o logos que mas deseamos, como comandos (imagen=PhotoImage(file=)), de esta manera inssertamos a nuestra interfaz las imágenes, escribiendo su ubicación, nombre y formato.

```
muestras.set(8640)
Label(ventana,text='N-o de Muestra x Dia: ',fg='red',bg='silver').place(x=150, y= 170)
Entry(ventana,textvariable=muestras,bg='white').place(x=300, y=170)

grafica=StringVar()
grafica.set(' ')
Label(ventana,text='Graficar datos .txt: ',fg='red',bg='silver').place(x=150, y= 270)
Entry(ventana,textvariable=grafica,bg='white').place(x=300, y=270)
Button(ventana,text='Graficar',fg='black',command=obtener3).place(x=300, y=295)

Button(ventana,text='Iniciar',fg='black',command=ejecutar).place(x=400, y=125)
Button(ventana,text='Llamar txt',fg='black',command=actual).place(x=380, y=295)
Button(ventana,text='Salir',fg='black', width=5,height=3, command=quit).place(x=390, y=380)
Button(ventana,text='Finalizar',width=6,fg='black',command=stop).place(x=390, y=195)

imagen=PhotoImage(file='/home/odroid/Pictures/xd3.png')
Label(ventana,image=imagen).place(x=10, y=15)

nivel=IntVar()
nivel.set('')
Label(ventana,text='Porcentaje:',fg='red',bg='silver').place(x=150, y= 350)
Entry(ventana,textvariable=nivel,bg='white').place(x=300, y=350)

ventana.mainloop()
```

Ilustración 3.15 Se inserta las imágenes deseadas

La programación mostrada es parte de la interfaz gráfica que aparecerá a continuación realizada para tomar la estimación de la carga del banco de baterías.

Ilustración 3.16 Interfaz terminada

3.10 CONCLUSIÓN

Es posible que hoy en la actualidad, pasar de que haya mucha contaminación por los motores a gasolina, poder emplear los vehículos eléctricos ya que es una alternativa para poder apoyar a la ecológica. Ya que actualmente la tecnología cada día avanza y podemos aprovecharla en beneficio de nuestra sociedad y para el mismo ambiente, con las herramientas que tenemos el software, los hardware que existen se puede logran hacer que los vehículos eléctricos sean más eficientes para el uso cotidiano.

Podemos aprovechar los programas computacionales para las mejoras que se le puedan hacer al mismo vehículo para que en un futuro próximo seamos más ecologistas, hacer conciencia de que le hacemos daño al planeta.

3.11 REFENRENCIAS

Coches RC.com (2013). *Motor eléctrico Brushless: Funcionamiento y características* [Online]. Recuperado de: http://www.cochesrc.com/motor-electrico-brushless-funcionamiento-y-caracteristicas-a3607.html

Migue Ángel Álvarez (2015). *¿Qué es Python?* [Online]. Recuperado de: https://desarrolloweb.com/articulos/1325.php

TROJAN BATTERY COMPANY
https://www.trojanbattery.com/es/

Fabrica de vehículos eléctricos en Chiapas INVEMEX
http://www.invemex.org.mx/

Método de estimación de carga para baterías de Litio Polímero basado en Filtro Kalman Extendido Adaptativo (AEKF)
https://www.researchgate.net/publication/292156744_Metodo_de_estimacion_de_carga_para_baterias_de_Litio_Polimero_basado_en_Filtro_Kalman_Extendido_Adaptativo_AEKF?fbclid=IwAR02L_U-3sUAsLYGoZsHKwxYIYLeTtEi5cgXXFlYBpbVuazA7NKIOZH7Vd8

Cómo calcular el tiempo de carga de una batería
https://techlandia.com/calcular-carga-bateria-como_47860/?fbclid=IwAR2SxVTKvukipqoQUZMGOgDEccpfJTk5pHoUTo2qkX_oylfHTz24j5PWPK0

métodos de estimación del estado de carga de baterías electroquímicas.
Autor: Jaime Martin Beltrán

Un modelo realista para la predicción del estado de carga de la batería en herramientas de simulación de gestión de energía.
https://www.sciencedirect.com/science/article/pii/S0360544218325064?fbclid=IwAR1yE5V4rg8qTxfV3b8W44l68ALSBTiP-4DfF7fJuXhR7R3CgayiB6m66JM