



Instituto Politécnico Nacional

Escuela Superior de Ingeniería Mecánica y Eléctrica

Unidad Zacatenco

Ingeniería en Comunicaciones y Electrónica

Proyecto: Dron



Microprocesadores

Avelar Mondragón Christian Miguel, 2021300304

Grupo: 6CM4

Profesor: Galicia Galicia Roberto

Fecha de entrega: Domingo 14 de Enero del 2024

Objetivos:

Al término de la presente proyectó, se procura:

- Diseñar un dron desde los componentes más básicos.
- Programar el funcionamiento de los dispositivos vinculados al Dron.
- Permitir el uso de Wifi para el funcionamiento y control de los mismos componentes que lo conforman

Introducción:

En el presente documento, pretendo dar la mayor información requerida que se implementó y utilizo con la finalidad de diseñar un dron propio para su uso controlado y medido. Se debe tener en cuenta que muchos de los materiales utilizados pueden ser cambiados por otros o incluso utilizar otro tipo de configuración para hacerlo funcionar. Pero en este caso será específico a ciertos materiales.

También proporcionaré y explicaré de forma resumida la programación, conexión y diseño del Dron, con el fin de que pueda utilizarse como referencia si se plantea mejorar.

También es importante mencionar que el Dron descrito permite el funcionamiento, movilidad y registro, lectura y escritura de datos enviados de forma serial a los motores y el módulo MP6050 (Giroscopio con acelerómetro). Pero no existe la funcionalidad de controlarse como control remoto. Así que es posible utilizarse para elevarse y descender, pero no hay forma de redirigir ni de nivelar. Puede ser riesgoso probarlo sin las mediciones adecuadas.

Desarrollo:

❖ Materiales y monto:

- Frame de 450mm, para cuadricóptero (F450): \$435 mx.
- Kit de motores (4) Brushless 1000kv A2212 con controlador ESC30A: \$996 mx.
- 3 Baterías LiPo 3.7V 2000mAh (503290): \$339 mx.
- Giroscopio con acelerómetro IMU MPU6050: \$49 mx.
- Arduino ESP32 WROOM: \$116 mx.

Monto total: \$1935 mx aprox.

❖ Explicación de los materiales:

Será importante definir la funcionalidad de muchos de estos materiales, para conocer su funcionalidad e importancia.

1. Frame:

Principalmente debemos saber que este será nuestra principal base en la que armaremos el circuito de nuestro dron, es importante tener en cuenta factores importantes a la hora de adquirirlo, pues al comprar el frame, también tendremos en cuenta los motores que vamos a utilizar, ya que estos deben tener la suficiente energía para poder permitir el ascenso del dron sin complicaciones, si adquirimos motores que no sean capaces de cargar un frame tan

grande, entonces todo nuestro proyecto jamás ascendería. Por eso es importante tener en cuenta si nuestros motores serán lo suficientemente capaces de elevar todos los componentes.

Y tomando en cuenta lo anterior también se involucra demasiado el peso. Pues de los factores que determinaran el equilibrio y capacidad que podrá tener nuestro dron de elevarse y mantenerse en el aire.

Por eso es importante distribuir adecuadamente el peso de los componentes que se coloquen dentro del dron.



2. Motores Brushless 1000kv A2212

Quizás una de las mayores complicaciones alrededor de este proyecto fue la adquisición de estos motores, debido a los escasos que había de ellos, retrasando mucho al mismo proyecto en sí.

Existe una gran importancia con estos motores, mejor conocidos como los motores sin escobillas. Los cuales son actualmente usados para evitar los problemas encontrados en motores eléctricos utilizados anteriormente (con escobillas). Estos motores sabemos que funcionan con la



corriente directa y se determinan por la cantidad de revoluciones por minuto que puede proporcionar conforme a la carga de tensión proporcionada en ellos.



Por si solos no tienen un funcionamiento PWM, para poder proporcionar esta particularidad es muy común el uso de los ESC30A, o mejor dicho, el controlador de velocidad electrónico, nos permite definir la velocidad de giro del motor,

esto gracias a que se le proporciona una señal de pulsos, la ideal para el funcionamiento de nuestro motores.

Todo este tipo de información sería bastante importante de leer en sus hojas de datos (datasheet). Se las proporcionaré en los siguientes links:

Motores Brushless 1000kv A2212: [https://www.openimpulse.com/blog/wp-content/uploads/wp-content/uploads/BLDC-A2212-13T-Datasheet.pdf](https://www.openimpulse.com/blog/wp-content/uploads/wp-content/uploads/wp-content/uploads/BLDC-A2212-13T-Datasheet.pdf)

ESC40A: https://graysonhobby.com/pdf/grayson-40A_manual.pdf

3. Baterías LiPo:



Lo más importante de todo es la energización de nuestro dron, mas que nada para los motores, ya que es de los mas importante. Deben ser baterías lo suficientemente utilices de una buena duración para que no sea un problema en tener que recargarlas. Estas baterías para el caso del dron deben de proporcionar 11.1 volts aproximadamente a los motores. Pero un solo batería solo proporciona 3.7, entonces para solucionar este problema, se tuvieron que colocar las 3 baterías en serie, con la finalidad de que la

suma de sus tensiones nos proporcionara los 11.1 volts que requerimos a la salida.

Con respecto a la energización del microcontrolador, lo que se tuvo que realizar es que se proporcionara la energía con otra fuente de alimentación, ya que utilizar las baterías podían generar fallos y caídas de voltajes a cada componente. Así que para solucionarlo utilizamos un Power Bank que se encarga de solo alimentar a la placa ESP32.

4. Giroscopio con acelerómetro

Este componente es un poco más difícil de explicar. En un principio el giroscopio se encarga de medir y mantener los movimientos de rotación por los que pase, estos son pequeños sensores que no consumen demasiada corriente y se utilizan para medir la velocidad angular, estos pueden ser de gran utilidad para equilibrar la rotación de algún autónomo y corregir los errores de inclinación enviando estas instrucciones al motor. Estos giroscopios pueden medir la velocidad angular en 3 ejes (x, y, z). Este tipo de mediciones son obtenidas gracias al efecto Coriolis. Ahora bien, estos 3 ejes son de rotaciones, es decir cuando se rota hacia una dirección entonces será detectado por alguno de los componentes, así es como se tiene en cuenta los valores cartesianos de un plano y el de un espacio (debido a z, la altitud). Entonces esto quiere decir que podemos medir las alteraciones o rotaciones de la altitud, longitud y latitud.

Ahora bien, la comunicación es realizada mediante el modulo I2C, el cual es trabajada en la mayoría de microcontroladores y con ellos también tiene su forma de ser controlado por un Arduino, permitiendo así la comunicación, escritura y lectura de los datos proporcionados por el Arduino y por el modulo MPU6050. Con todo respecto al giroscopio y como su rotación se medí, entonces tan solo es necesario agregar un sensor que determine el tiempo con el que este tipo de datos de alteración recibidos al modulo cambian, entonces tenemos ahora la aceleración.

Con esto es posible de realizar toda la configuración necesaria a nuestro dron y así obtener los datos correspondientes para enviarlos y controlarlos.

5. ESP32

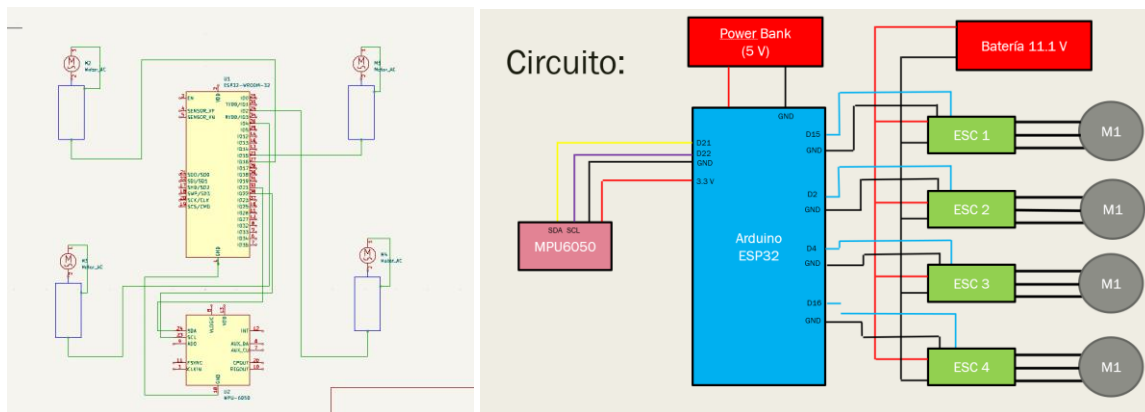
Será algo breve con este dispositivo. Este es un microcontrolador o Arduino que nos permite realizar una gran cantidad de acciones y actividades. Actualmente se ha vuelto muy popular

en la electrónica debido a su bajo costo y a algunos sensores o componentes que lo conforman que facilitan la mayor cantidad de tareas. También una de las mayores ventajas es por su memoria flash que contiene, permitiendo almacenar una gran cantidad de instrucciones.

Para este proyecto lo mas importante que lo hizo destacar para que podamos hacerle un uso, fue debido a sus módulos de Wifi y Bluetooth, los cuales ya vienen integrados en el microcontrolador y eso nos permite ahorrar en comprar algunos otros componentes y en poder proporcionar una funcionalidad de control mediante wifi y una página web las acciones que pueda realizar el dron, así como algunas lecturas importantes del MPU6050.

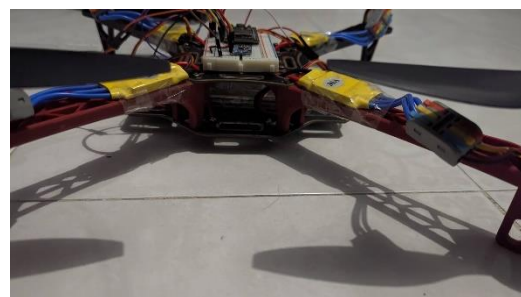
❖ Construcción del dron

Como primer punto para el diseño de nuestro dron era necesario visualizar la construcción del circuito que iba a contener y como todos los dispositivos necesitarían ir conectado uno con otro. Así que para realizar esto, se implementó el siguiente circuito:



Ahora que conocemos el circuito a implementar, es momento de ensamblar todo en cada componente.

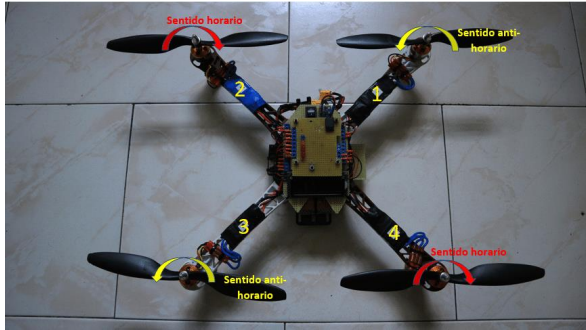
Para eso comenzaremos con el Frame, conociendo como esta conformada el circuito de energización en la placa que esta incluida, pues estos Frames nos ayudan a ya tener interconectado la energización de los motores, tanto los puntos positivos, como los negativos para que podamos colocar nuestras baterías de una forma mas cómoda y el cableado no sea tan problemático de pensar.



Una vez conociendo cual es positivo y negativo, se recomienda soldar todos los cables a conectar para las baterías y los motores, recordemos que las baterías ya deberían estar en serie, los cuales al final de todo solo seguiremos teniendo un positivo y un negativo que se conectara en una sola parte para que todos los demás puntos salgan con 11.1 V aproximadamente.

Como siguiente paso será momento de colocar cada uno de los motores a los extremos, ya tendremos los tornillos necesarios para asegurar los motores con seguridad, es importante que estén bien apretados para que no comiencen a vibrar demasiado a la hora de encenderlo.

Con respecto a los motores esto vendrá con su kit de hélices y su ESC. Primero que nada, las hélices tienen su importancia, ya que cada motor tiene 2 hélices, pero estas son diferentes entre sí, debido a que una se utiliza para un determinado sentido y la otra para el otro. Esto con el fin de que al moverse el motor el aire comience a bajar generando que este se impulse del suelo y permita el vuelo.



Para poder determinar la dirección de giro tendremos que pasar ahora a la conexión de los motores. Estos motores son trifásicos y tienen en realidad 3 puntas, siempre el cable negro va a ser GND y ese nunca se va a alterar en ningún motor. Pero tanto el cable rojo y amarillo estos van a intercambiarse al conectarse con el ESC, para determinar el

sentido del giro. Ahora es importante que si 1 motor lo pongo hacia una dirección el que sigue o el que este alado de el debe tener los cables intercambiados con respecto al anterior. Así tendremos un motor con un orden de cables y el siguiente estos cables estarán intercambiados y el siguiente como el primero y el ultimo intercambiados. Con esto determinaremos el giro. Ahora para las hélices debemos escoger el que permita que cuando se gire el motor el aire baje y no suba, esto se puede identificar en el relieve de las hélices y el siguiente paso será ajustar estas hélices con mucho cuidado y sin aplicar tanta presión por que podemos dañar los motores debido a que se les será difícil girar.

Ahora para finalizar con los motores solo es necesario conectar la energización del ESC. Si nos damos cuenta este cuenta con 3 cables. 2 de ellos son muy grandes y estos son la energización principal del componente, estos van a ir a los cables que soldamos previamente a la placa del Frame. Mientras que el cable pequeño se conforma por otros 3 cables. GND, VCC y Datos. Es importante tomar en cuenta que VCC no se va a conectar, solo se va a conectar GND y los Datos, ya que la energización ya la va a tener el motor.

Por último, utilizaremos algo para colocar la ESP32 y el giroscopio, para esto utiliza una protoboard que me ha parecido muy fácil de utilizar. Así era más fácil usar el carril negativo para interconectar todo. En la ESP32 debemos de conectar los motores tal cual esta descrito en el diagrama del circuito, son los pines establecidos en el programa y con ellos permiten la

comunicación serial con el controlador de velocidad. El GND podemos conectarlo en uno de los carriles así colocaremos los GND de los 4 motores.

Por otro lado, tendremos al giroscopio, procuraremos colocarlo en un lugar adecuado para pasar los cables. Este giroscopio debe conectarse en su VCC la salida de 3.3 V que nos proporciona la ESP32. Mientras que la conexión serial del I2C será a los pines explicados en el diagrama, ya que es la salida permitida de I2C de la ESP32. Se recomienda mucho soldar con cuidado el giroscopio a un header, el módulo puede quemarse con mucha facilidad, dificultad que fue posible notar demasiado.

Específicamente para nosotros no era fácil conectar la ESP32 a las mismas baterías de los motores, ya que las caídas de tensiones podían provocar serios problemas al Arduino y desconectar el modulo de Wifi siendo algo demasiado peligroso en realidad, así que por lo mientras se utilizo la energía proporcionada por una computadora. Para resolver este problema lo más recomendable es utilizar un Power Bank que suministre 5 V de corriente directa a la placa.

❖ Programación

Es momento de hablar de una de las partes más difíciles en este proyecto, sección la cual aun no esta terminada y actualmente se cuenta con lo más básico de funcionamiento para la lectura y escritura de datos que pasan por la ESP32 a los motores o al MPU6050.

Explicare conceptos muy generales con los que cuenta el programa, el cual también esta proporcionado en esta misma carpeta de archivos.

```
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Arduino_JSON.h>
#include <ESP32Servo.h>
#include "SPIFFS.h"
```

Comenzaremos por las bibliotecas que contiene nuestro programa, en ellos podemos ver 3 secciones muy importantes:

- ESPAsyncWebServer:

Esta biblioteca nos va a permitir generar un servidor con el ESP32, este servidor va a poder conectarse mediante Wifi con su respectiva biblioteca. Es importante definir

varios parámetros para la conexión web de nuestra ESP32 a alguna conexión Wifi. Para esto fue preciso investigar algunos conceptos básicos con respecto a la conectividad de red, declarando el nombre de nuestra red y la contraseña para que pueda conectarse a internet. Ahora bien la ESP32 puede configurarse en diferentes modos, se estuvo en duda si utilizar una red wifi en modo AP (Access Point) o STA (Station), pero en esta ocasión fue preferente usarlo en el modo STA, ya que nos permitía hacer uso de funciones de callback con la página. El Arduino en general estará enviando constantemente los datos y recibéndolos de los componentes, pero estos datos son enviados a una página web cargada y creada por la misma ESP32. Para poder realizar esta parte utilizaremos una función nueva que es necesario agregar o implementar en nuestro Arduino IDE. Este es conocido como el SPIFFS o el Sketch Data Upload. Esta funcionalidad nueva nos permitirá ingresar datos a la memoria flash de

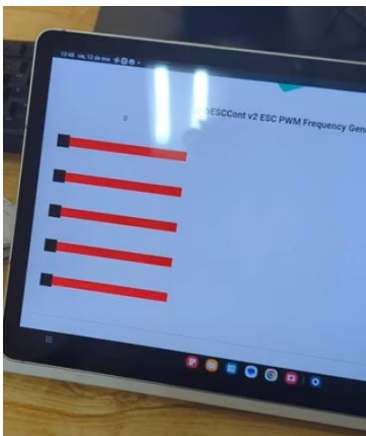
nuestra ESP32, en este caso lo que vamos a agregar es la página web, junto con su hoja de estilos y su script.

Para esto, debemos acabar de colocar todo el programa y cargarlo, en un principio no va a conectar ni abrir ningún tipo de página al cargarse. Pues en ese caso no va a generar una IP a la cual

conectarse, al menos no aparecerá nada en la navegación. Para completar este paso debemos de crear una carpeta en el mismo lugar que la de nuestro programa, esta carpeta debe llamarse “data”. Dentro de esta carpeta se encontrará el archivo html, la hoja de estilo y el script. Ya que se instala ahora cada que la ESP32 este encendida se conectara a internet y va a cargar la página web que contiene en su memoria en alguna IP que proporcione nuestro modem a la ESP32.

| | | | |
|----------------|------------------------|-----------------------|-------|
| data | 12/01/2024 12:29 a. m. | Carpeta de archivos | |
| PruebaCompleta | 10/01/2024 10:19 p. m. | Arduino file | 12 KB |
| index | 10/01/2024 10:12 p. m. | Opera GX Web Do... | 3 KB |
| script | 10/01/2024 10:10 p. m. | Archivo de origen ... | 5 KB |
| style | 10/01/2024 10:10 p. m. | Archivo de origen ... | 2 KB |

- ESP32Servo:



Esta biblioteca no es tan complicada en realidad, lo que nos proporciona es poder tomar control del ESC. En sus salidas podemos generar una señal de pulso que estará siendo enviada constantemente a estos componentes, permitiendo así el movimiento controlado de los motores brushless generando una función PWM de los mismos, útil para la velocidad y control de movimiento y altura de cada uno de los motores. Asignando una salida específica a cada salida y para su control podemos realizarlo mediante “sliders”. Cada que alteremos uno de las sliders significará un tipo de movimiento o velocidad del tiempo de los pulsos que le estará llegando a los

motores.

Esta comunicación ya esta definida en el ESP32 y la página web, funcionando consecutivamente dando escritura a los motores y lectura de los valores que están actualmente en funcionamiento.

También otra de las cosas adicionales que puede realizar es calibrar a los motores cuando se intente encender estos mismos, para que no haya problemas de que se bloquee su giro o falle.

- Adafruit

Esta biblioteca, aunque tiene muchas funciones, principalmente permite la comunicación I2C de la placa y el modulo MUP6050. En general estas bibliotecas nos permitirán controlar la velocidad, las distancias y los cambios con respecto al tiempo, también hacemos uso de un sensor de temperatura para conocer la temperatura de nuestra ESP32 al mismo tiempo. Este

tipo de datos comenzarán a ser leídos y enviados a la ESP32, por consiguiente, estos serán enviados a la página web.

Recopilar todos estos datos conlleva de mucha información, la cual ya expliqué muy por encima cuando explico acerca del módulo y como obtener algunos de esos valores o porque pueden ser tan importantes.

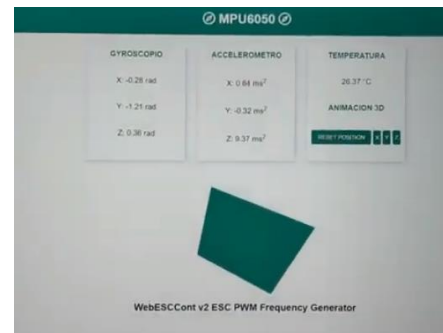
También tiene su propia calibración para las mediciones, pero fue posible notar que existe un fallo muy frecuente cuando se obtienen los cálculos del eje x. Pero para eso es necesario pulir lo mejor posible los datos y calibración de los valores leídos por el módulo, todo esto será necesario de leer y comprender a profundidad.

Algo consiguiente de esto es que todo este tipo de información y datos van a ser útiles posteriormente para el control total del Dron, así pueda incluso direccionarse a una ubicación deseada en específico. Pero este tipo de información y programación se va a posponer debido a la alta complejidad y tiempo requerido en hacer uso de esta información.

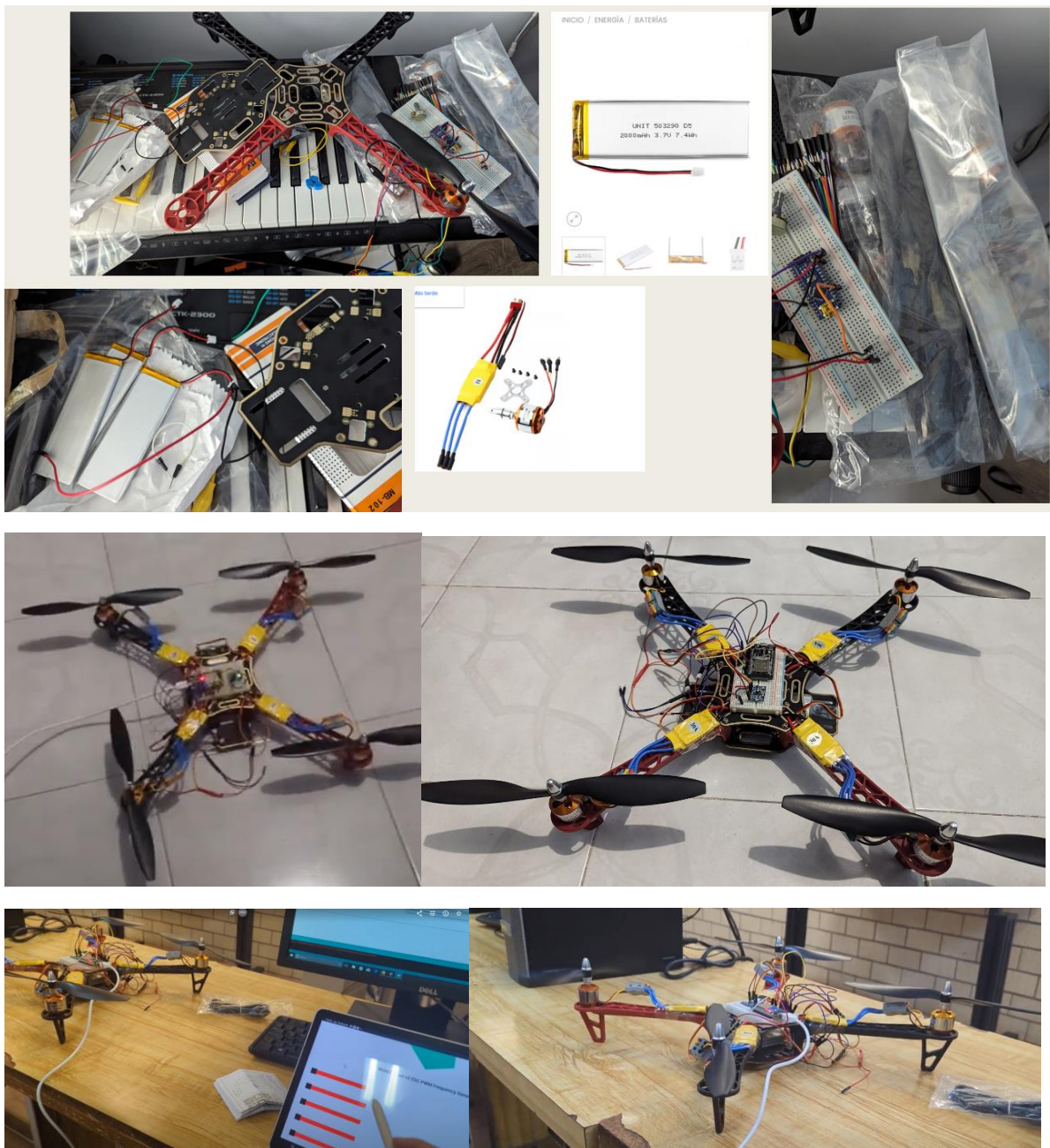
Todos estos códigos implementados también fueron proporcionados por diferentes usuarios y paginas que fueron consultados para la conexión e implementación completa de nuestro programa. Principalmente los más importantes fueron:

Para Servo Motores: Ya no se encuentra el video habilitado, pero fue realizado por Ankush Sheoran.

Para MPU6050: <https://www.youtube.com/watch?v=516be4dHADc&t=125s>



❖ Evidencias



Conclusiones y propuestas a futuro:

Al final de este proyecto logre darme cuenta de la complejidad e importancia en la realización, diseño y programación de un dron que funcione de acuerdo a las necesidades solicitadas. Este es un proyecto que se involucra en sectores muy importantes en la actualidad y que se están desarrollando con la finalidad de darles un uso muy importante en el futuro. Realizar este tipo de proyectos no es una tarea fácil y tampoco es fácil conseguir en tan poco tiempo, conlleva mucho conocimiento y aprendizaje continuamente para el ajuste necesario a todos los problemas que puedan irse encontrando con el tiempo.

Fue bastante agradable realizar uno de estos proyectos, aunque a decir verdad conllevan un coste demasiado elevado y una dedicación de tiempo bastante extensa por lo que lo hace no muy factible para demasiadas personas. Pero es por esta misma razón que buscamos una manera de que todo salga a un menor costo y a una gran eficiencia, aunque parece un reto muy difícil de conseguir puede ser posible conforme pase el tiempo se ira actualizando y mejorando.

A decir verdad, en un futuro se espera poder lograr el control realizado desde una página web, tener la habilidad de poder controlar la dirección y velocidad proporcionada a nuestro dron podremos implementar otros factores y algo muy importante de estos factores es la seguridad, ya que debe ser tan importante como la terminación del proyecto, debemos asegurar que pueda ser usado sin que represente ningún tipo de riesgo a los civiles. Además, que es importante seguir las leyes que se rigen en el país para el uso medido y controlado de los drones, ya que esto ya está siendo implementado actualmente y se encuentra regulado en el gobierno.

También es importante implementar otras funciones que pueden ser interesantes o de mucha utilidad, por el momento se plantea el uso de una cámara, pero también, me gustaría implementar algún tipo de bocina o sonido que pueda reproducirse a través del dron, esto ultimo puede generar demasiadas dificultades debido a ciertos parámetros u obstáculos que impiden el uso de ondas de radio.

Este reporte es generado por si alguien desea mejorar o actualizar las funciones del dron, implementando un sistema económico y que pueda realizar una funcionalidad efectiva con un Arduino que pocas veces a sido utilizado para este tipo de proyectos (ESP32)

Bibliografía:

- Introducción al giroscopio. (s. f.). https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=13
- Tutorial MPU6050, acelerómetro y giroscopio. (s. f.). Naylamp Mechatronics - Perú. https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html
- Colaboradores de Wikipedia. (2023, 24 mayo). Motor eléctrico sin escobillas. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Motor_el%C3%A9ctrico_sin_escobillas
- Como hacer un drone con arduino | Drone arduino casero paso a paso. (2023, 12 junio). ArduProject.es. <https://arduprject.es/>
-