

PROCESO DEL PROYECTO

DEFINICIÓN Y PLANEACIÓN DEL PROYECTO

Inicialmente nos planteamos tres alternativas, dos de estas eran de entretenimiento o decoración (característica que los descarto):

1. Un cubo de Leds RGB dinámicos y programables.
2. Un reloj de cartas - Calendario de cartas.
3. Dispositivo para dar señalamiento lumínico a las bicicletas.

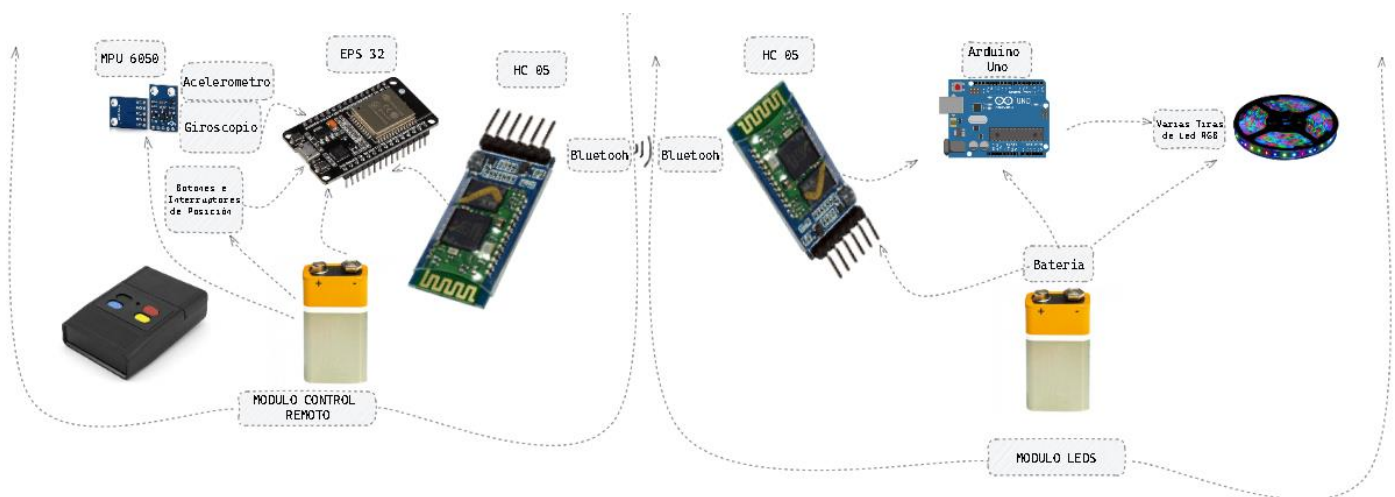
Si estás aquí sabrás que escogimos la tercera opción y empezamos con la determinación de las funcionalidades. Para esto realmente se mantuvo una pequeña conversación ya que estas funcionalidades eran bastantes obvias, siendo las siguientes:

- Poner iluminar la dirección a tomar (derecha o izquierda).
- Poner señalar un próximo detenimiento o parqueo.
- Poder señalar el proceso de frenado.
- Cuando es de noche, se debe de tener iluminación permanente respetando las funcionalidades previas.

También por tema de presupuesto se dejo que las funcionalidades quedaran en “iluminación”, ya que el dispositivo como se mencionara posteriormente podría tener un sistema de sonido, un sistema que mida distancias o un sistema GPS.

Diagrama de Caja Negra

Ya con las funcionalidades establecidas se empezó a establecer la tecnología a utilizar. Para esto se utilizo el diagrama de caja negra el cual es muy intuitivo y dinámico, teniendo el siguiente resultado:



Este diagrama se divide en dos partes, la parte del Control Remoto y la parte de las Luces (De aquí en adelante cada parte se llamará correspondientemente a lo dicho).

El Control Remoto posee un esp32 DEVKIT DOIT v1 como cerebro, una batería (18650 de 3,7V) como alimentación, un MPU 6050 para automatizar algunas señales y el HC-05 para la conexión Bluetooth con Luces, el cual a su vez contiene un hc.05, un Arduino UNO como cerebro, dos baterías como alimentación y una conexión a las tiras LED de iluminación.

Ahora ampliando lo dicho, queremos empezar con la siguiente pregunta: ¿Por qué se escogieron estas tecnologías? Iniciemos por el HC-05, este dispositivo nos permite generar una conexión Bluetooth entre ellos y además nos permite controlar las direcciones entre los módulos, contraseñas, nombres, controlar si es un Maestro o Esclavo (Si es host o no), etc. Con esto conseguimos una conexión entre partes remotas de un mismo dispositivo de manera segura, rápida y confiable. Sobre el MPU 6050 es muy poco lo que toca mencionar en este documento, este dispositivo es muy versátil a la hora de medir alteraciones en la velocidad lineal y en la velocidad angular, siendo perfecto para medir los cambios que pueden presentarse a la hora de usar una bicicleta o cualquier vehículo (Se recomienda leer sobre este dispositivo en internet). Para escoger el esp32 y el Arduino fue mas complicado el camino ya que inicialmente teníamos que el MPU 6050 funciona mejor bajo 3,3V mientras que el resto de los circuitos trabajaban a 5V, se sumaba el hecho de necesitar trabajar en el lenguaje C++ para el manejo de librerías próximas a escoger, junto a la necesidad de escoger entre Wi-Fi y Bluetooth en la interconexión (el primer componente a escoger de cada parte fueron los cerebros), el pequeño espacio que debían tener la PCB futuras y el tema del presupuesto. Pero con una leve investigación el esp32 era la elección más acertada ya que cumplía todas las características e incluía las formas de interconexión mencionadas (se escogió Bluetooth por la economía del HC-05 y no se usó el propio del esp32 por temas de seguridad), ¿pero entonces, donde aparece en juego el Arduino? El Arduino aparece por la imposibilidad de conseguir el segundo esp32, ya que los envíos estaban ciertamente caros y el Arduino ya se tenía en posesión antes de iniciado el proyecto. La única desventaja que esto representa es el tamaño de la placa.

Sobre las tiras LED RGB, se escogieron los ws2812b, ya que presentaban una alta gama de colores, las librerías son sencillas de utilizar, algunas webs aseguraban que eran a prueba de agua e iluminan bastante. Por economía se compraron los LED en si y no las cintas, ya que nuevamente los envíos son excesivamente caros en Colombia (el metro de la cinta es barato en contraste). Por último, en la parte de alimentación tenemos a las baterías 18650 de 3,7V que se escogieron por ser tanto recargables como duraderos (el dispositivo puede pasar bastante tiempo sin cargar), para cargar dichas baterías se usaron puertos Jack y módulos MD-CH42 que permiten hacer el proceso de carga-descarga de baterías dando 5V constantes al circuito.

Diagrama de flujo – Costos

Invitamos a ver o descargar los archivos adjuntos a este proyecto, ya que lo que se mencionara aquí son mínimos detalles.

Sobre el diagrama de flujo cabe resaltar que existen funcionalidades prioritarias, como los es el frenado y por esto mismo en el flujo a la hora de existir una desaceleración todas las otras funciones quedan bloqueadas.

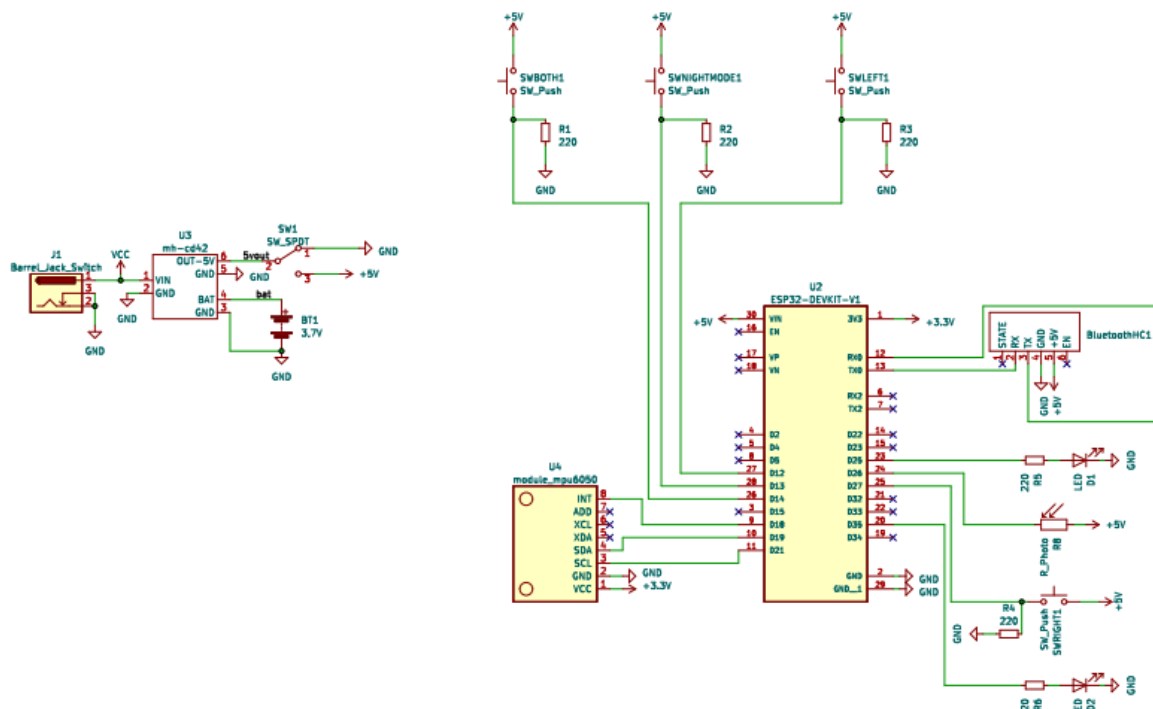
Sobre los costos se ha de resaltar que las compras fueron hibridas y que el presupuesto estaba alrededor de los \$200.000

Con esto la planeación había acabado y entramos a la producción.

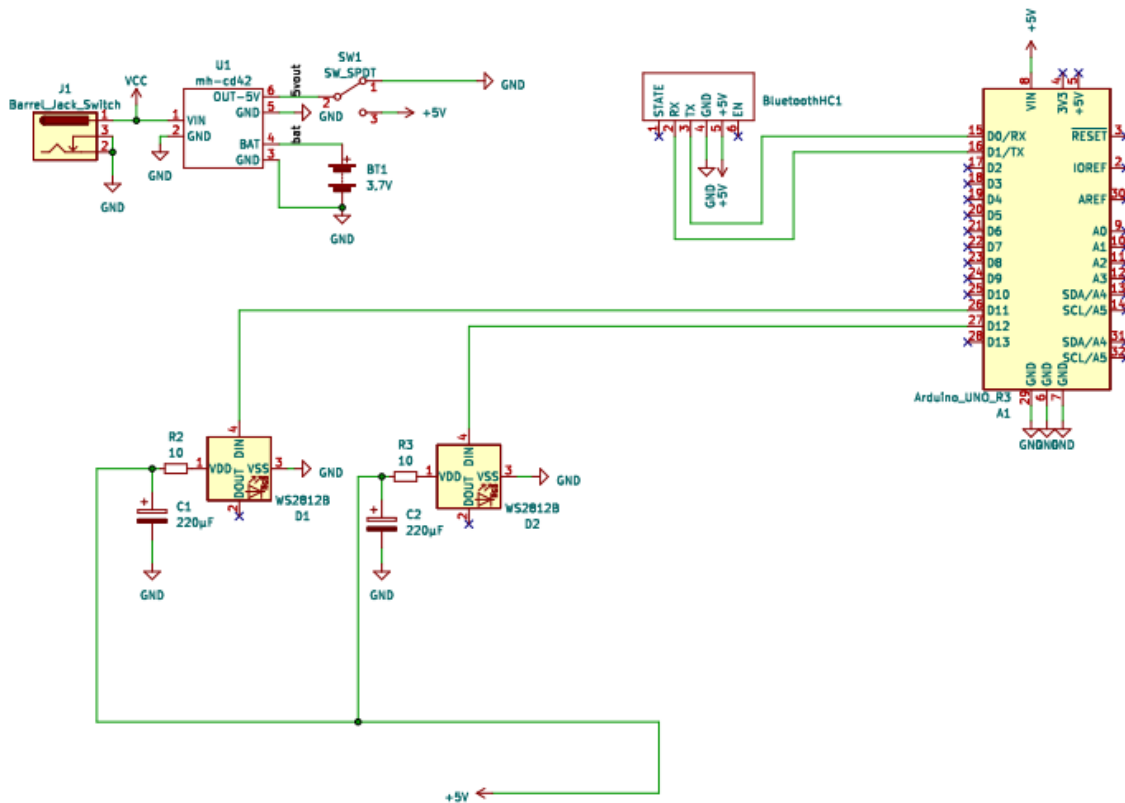
ESQUEMATICOS, CODIGO Y PCB

Para esta parte se usaron dos Software, KiCad para la creación de los esquemáticos y las PCB, y Arduino IDE para el código.

Empezando por lo esquemáticos, estos iniciaron con la búsqueda de las “imágenes” de cada componente para poder generar las conexiones, anotación y correcciones correspondientes. Este proceso fue el mas laborioso realmente, siendo sorprendente el hecho de que nos tocara hacer los esquemas para el HC-05 t el MPU-6050 por mano propia (usando KiCad), pero fue una experiencia valiosa. Al final se generaron los siguientes esquemáticos:



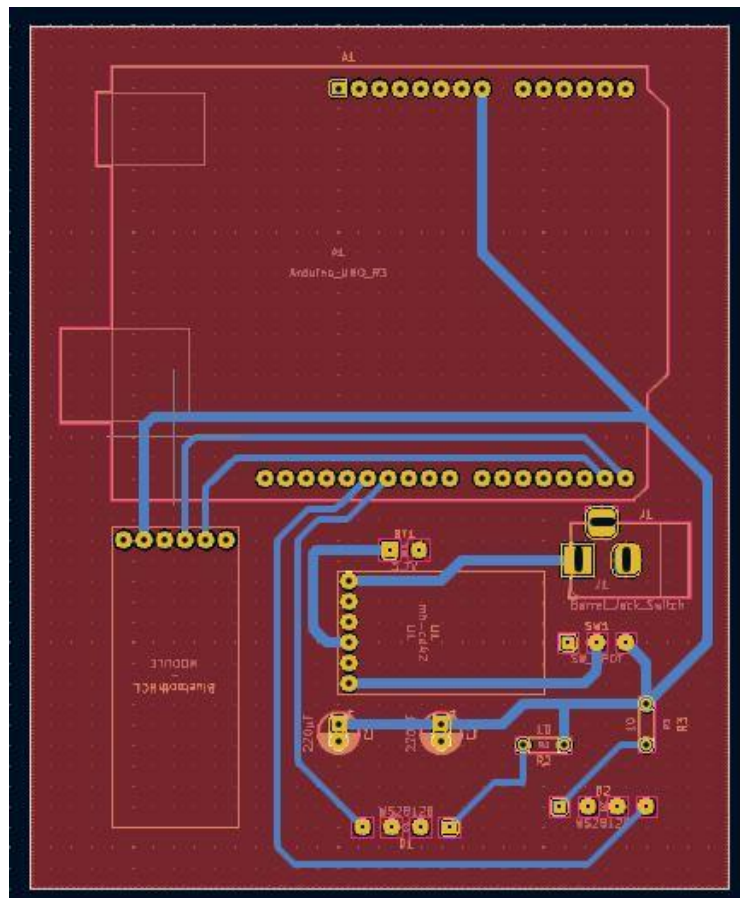
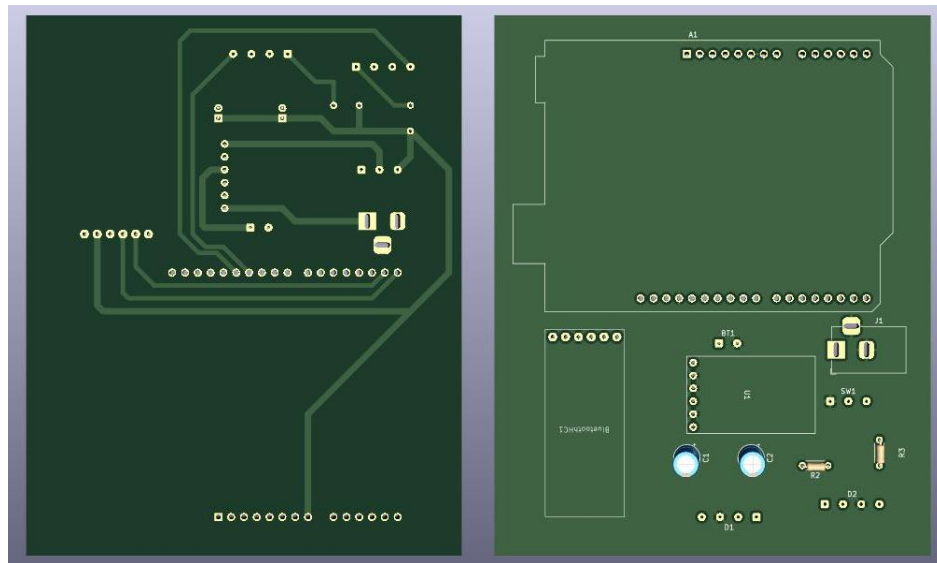
Esquemático Control Remoto



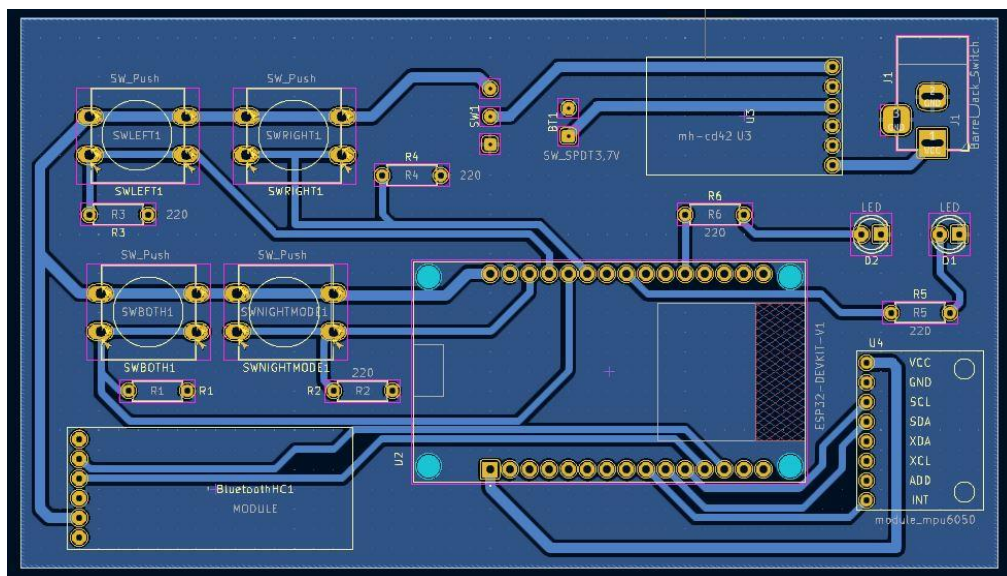
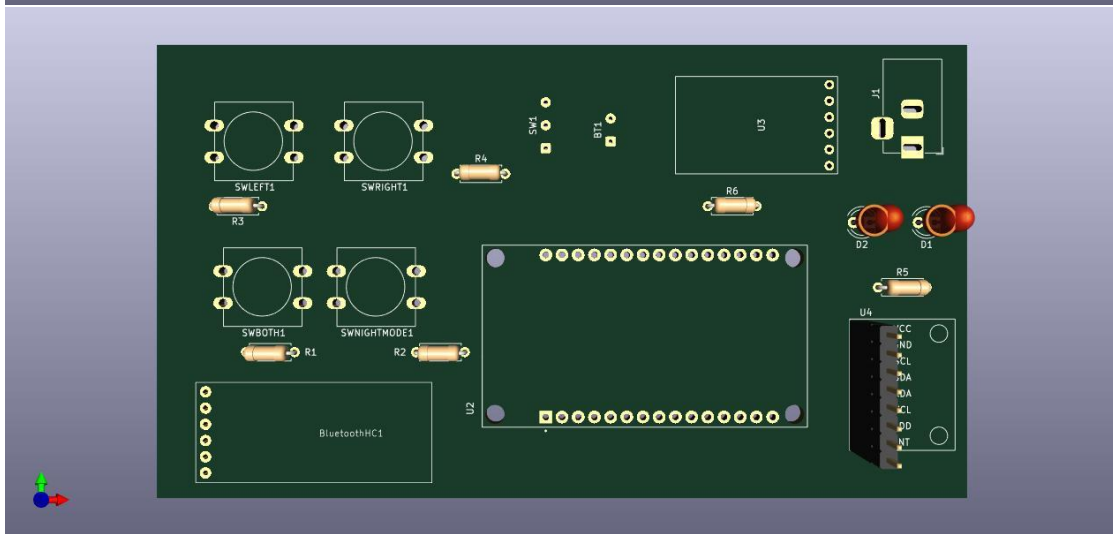
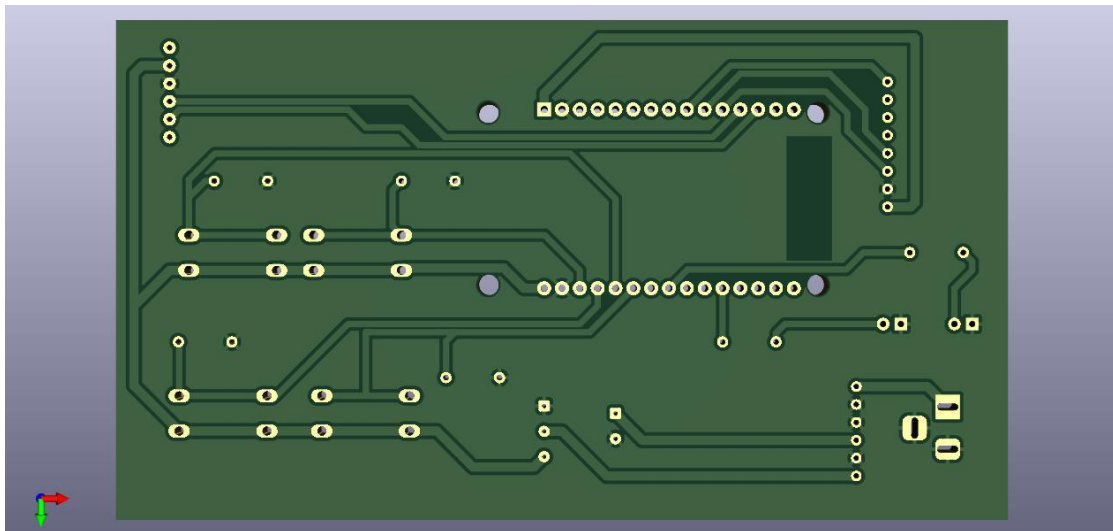
Esquemático Luces

Ya con los esquemáticos hechos asignamos las huellas (footprints) de cada componente, siendo para varias de ellas (como los ws2812b) simples “puntos de conexión”. Cabe aclarar que también toco realizar algunos footprints propios.

Después de esto seguía la creación de las dos PCB, pero como ya se tenía la mayoría hecho gracias a los esquemáticos, solo todo ordenar los componentes y realizar las conexiones (diferenciando los anchos de pista según el voltaje y la función de cada una, por ejemplo, las pistas de señal requieren menos grosor) y creando un campo GND. El resultado fue el siguiente:



PCB Luces



PCB Control Remoto

Con todo listo se empezó el desarrollo del código. Primero cabe aclarar que la parte de decir que LED encendía o como se enviaba información a través del Serial se deja a disposición del lector, por lo sencillo que ha de ser el código. Por lo que solo hablaremos de lo importante.

Primero se configuraron los HC-05, para estos se conectaron los módulos tanto al esp32 como al Arduino (igual que en el esquemático, pero mientras se conecta el 5V de los módulos se ha de oprimir el botón que estos poseen, así entran en modo de configuración) y se empezó una conexión con el Monitor Serial a 9600 baudios, después se inició la conexión con los módulos a 38400 baudios, permitiendo utilizar los comandos AT. Estos comandos permiten conocer los datos de los HC-05 como el nombre o la contraseña y a su vez los permite cambiar. La ecuación es esta, para conocer un dato se coloca AT+(NAME o PSWD o MODE, etc) y si se desea cambiar se escribe el mismo comando un igual y el dato nuevo. Por lo que con esto hecho, se convirtió el HC de las Luces en esclavo (solo una conexión a un maestro) y el HC del Control en el maestro (Host), además se configuró que solo se pudieran conectar entre ellos (cambie el CMODE, el BIND o dirección y el PSWD de ambos). Para saber más de este proceso recomendamos investigar en internet. Y por último recordar, que cuando acaben de configurar escriban el comando AT+RESET y esos 38400 baudios del código los pasen a 9600, si no, al enviar datos de dispositivo a otro, lo que llegara será un carácter de pregunta o un carácter de cuadrado al no reconocer la información correctamente.

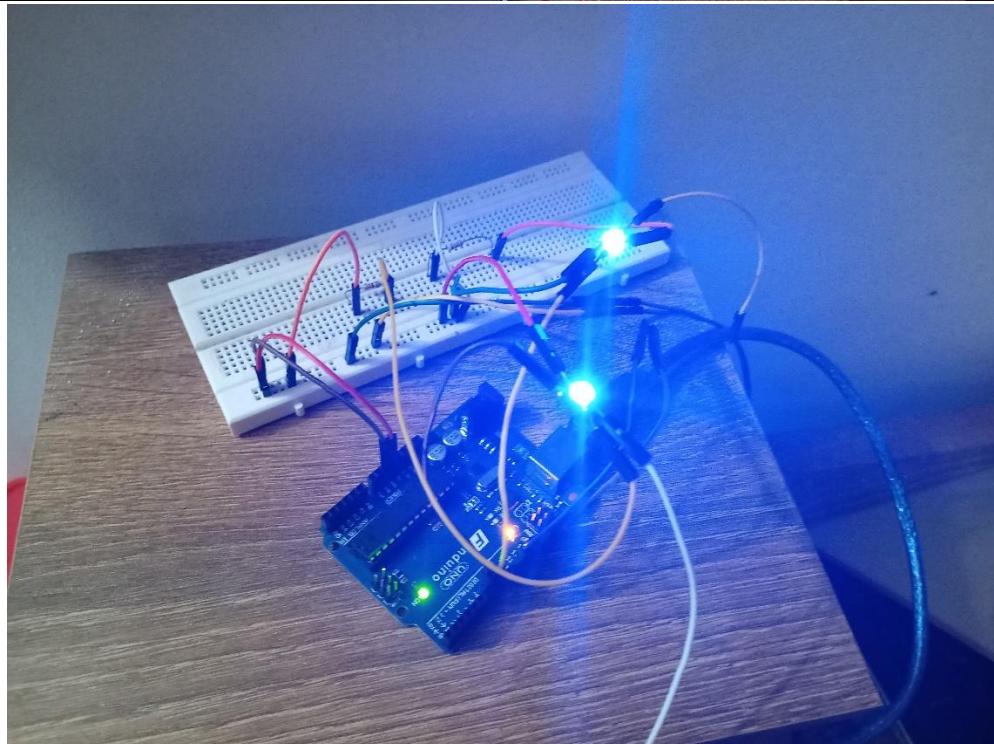
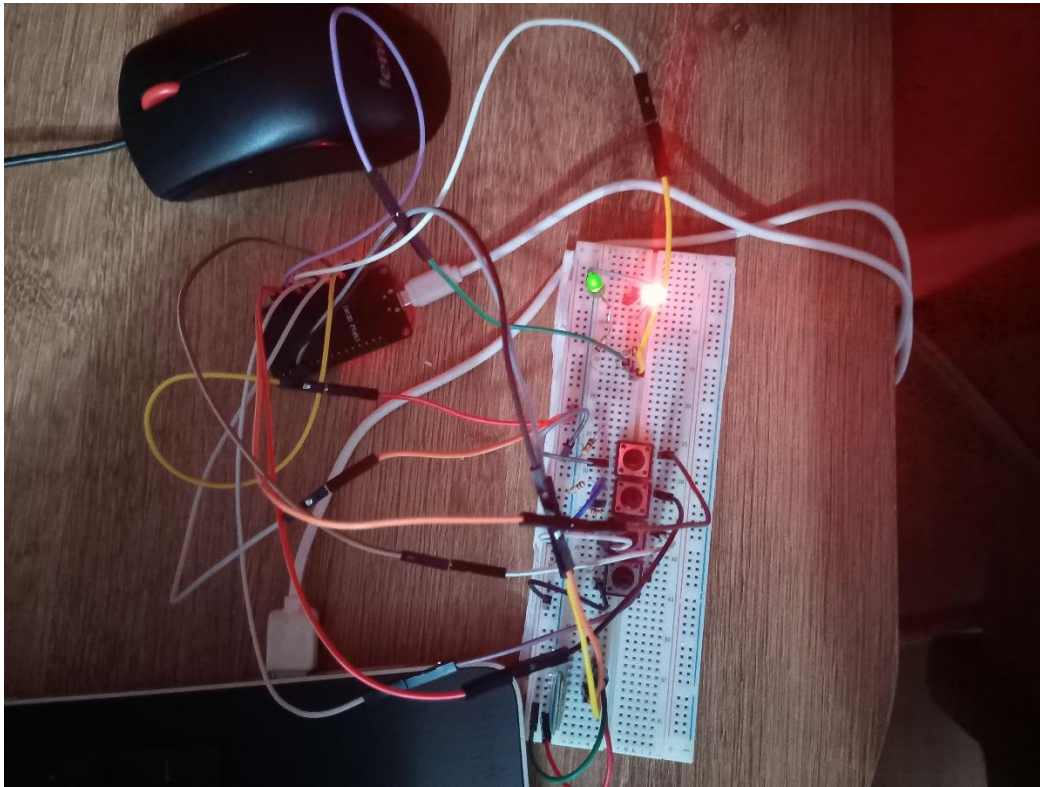
Sobre el control de los ws2812b se utilizó la librería `Adafruit_NeoPixel` el cual permitía en tres líneas inicializar los leds: se asigna el strip, se inicializa y se muestra. Para cambiar el color únicamente se llamaba al strip, al número del led y al color nuevo.

Sobre el MPU 6050 también recomendamos la lectura vía internet ya que de ahí se hizo un “copia y pega” y funcionó preliminarmente.

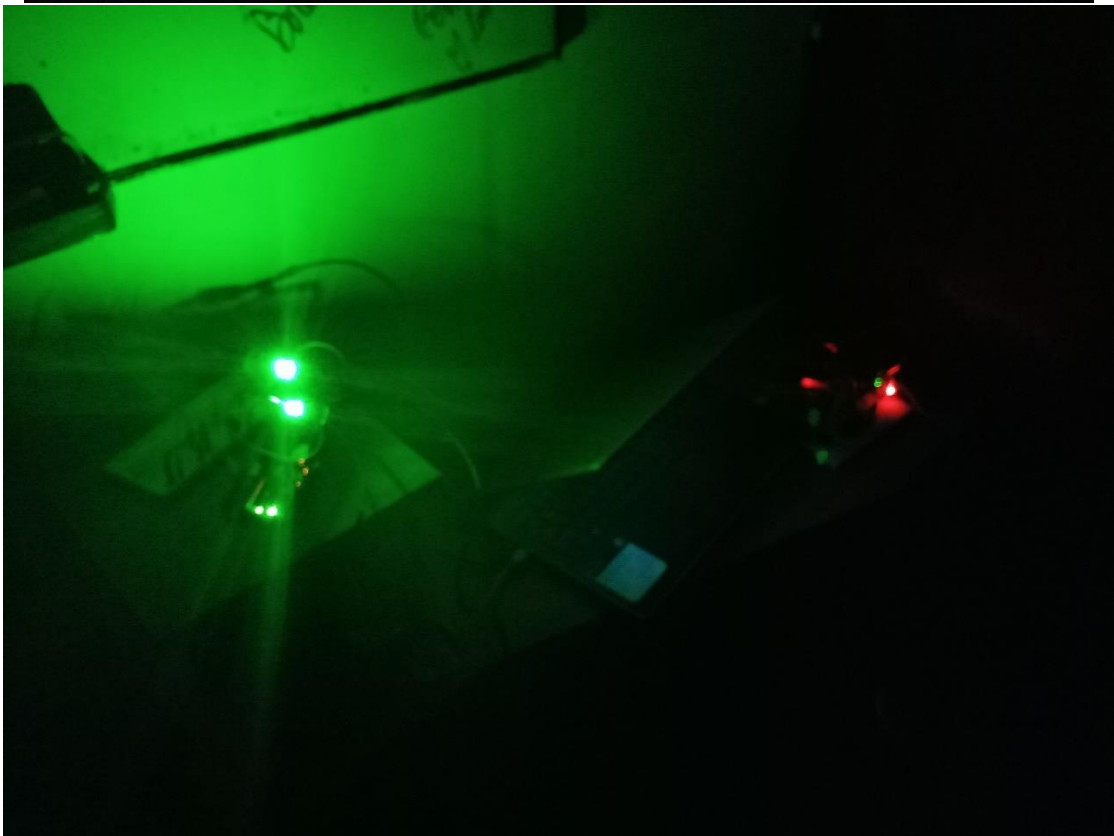
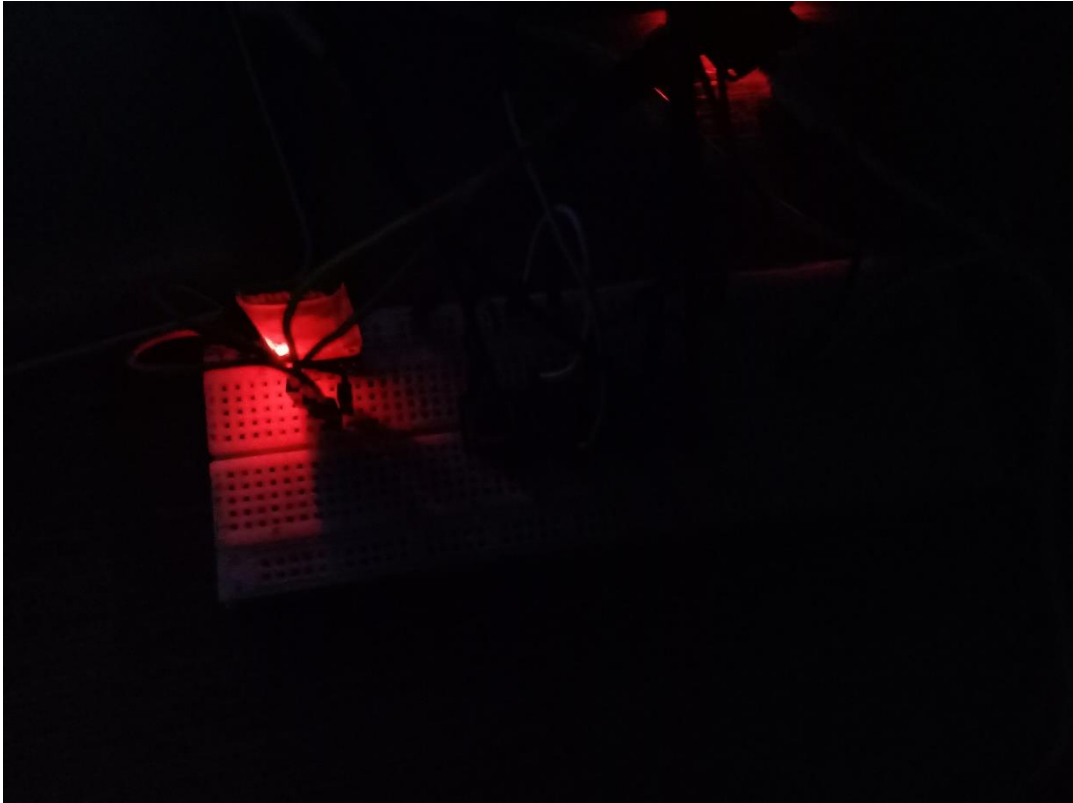
Pruebas del Prototipo

Realizamos las pruebas en ProtoBoards para poder saber que todo funcionaba correctamente, y así fue a excepción de un problema que tuvimos con el MPU. Este módulo al soldar sus pines terminó dañándose y fue imposible comprar su reemplazo, por lo que las pruebas fueron realizadas sin él.

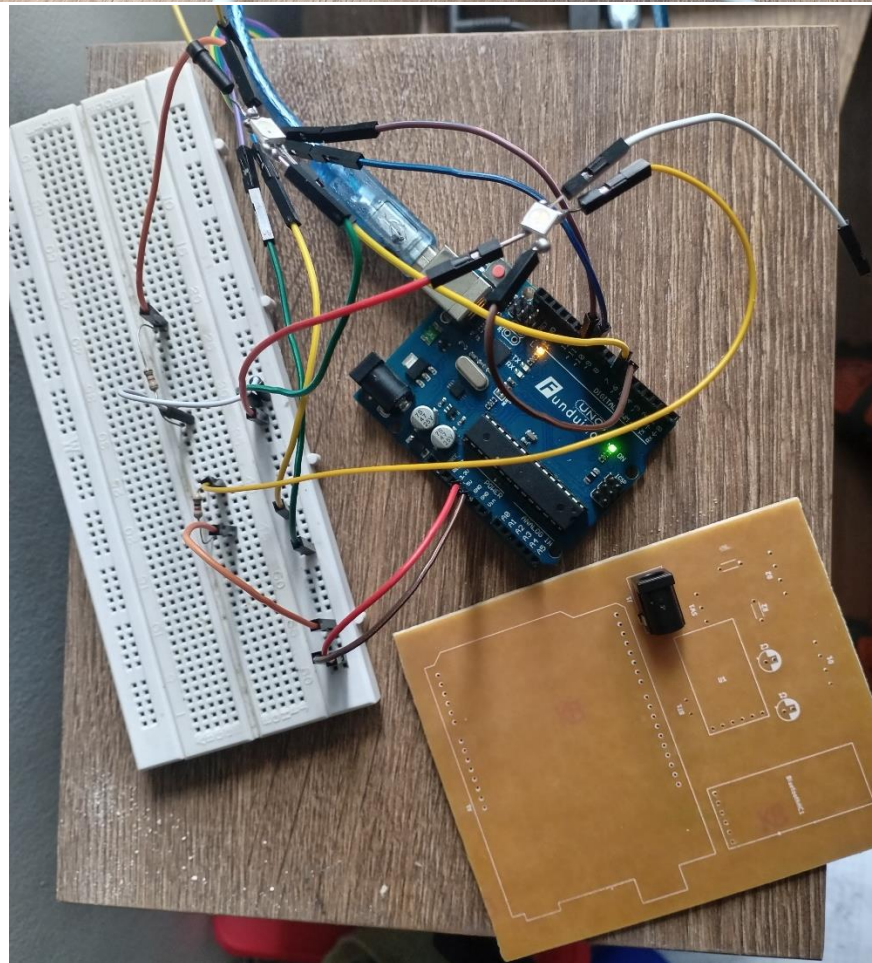
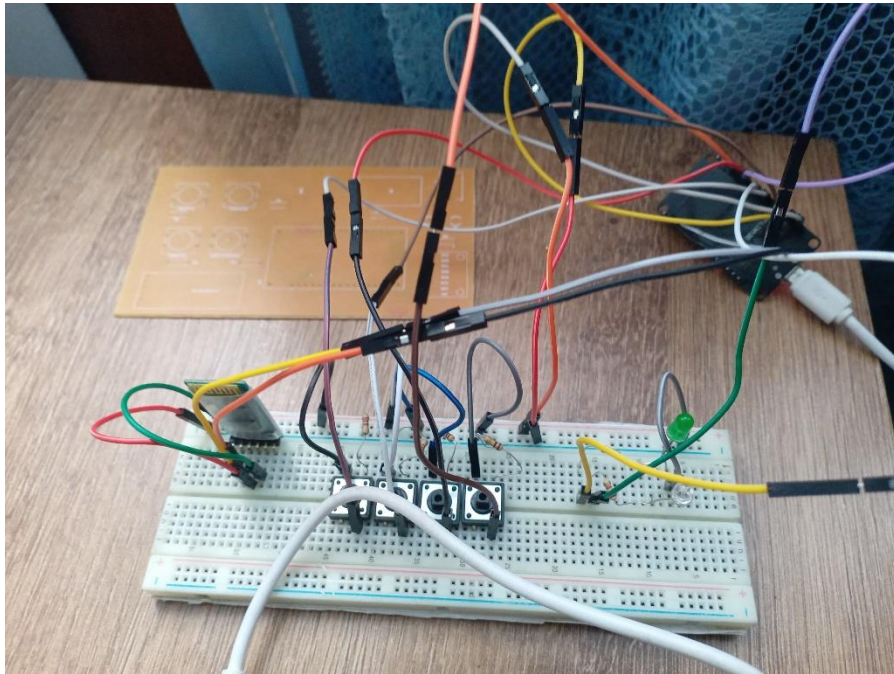
Se recomienda ver los videos de demostración.



Imágenes al Atardecer



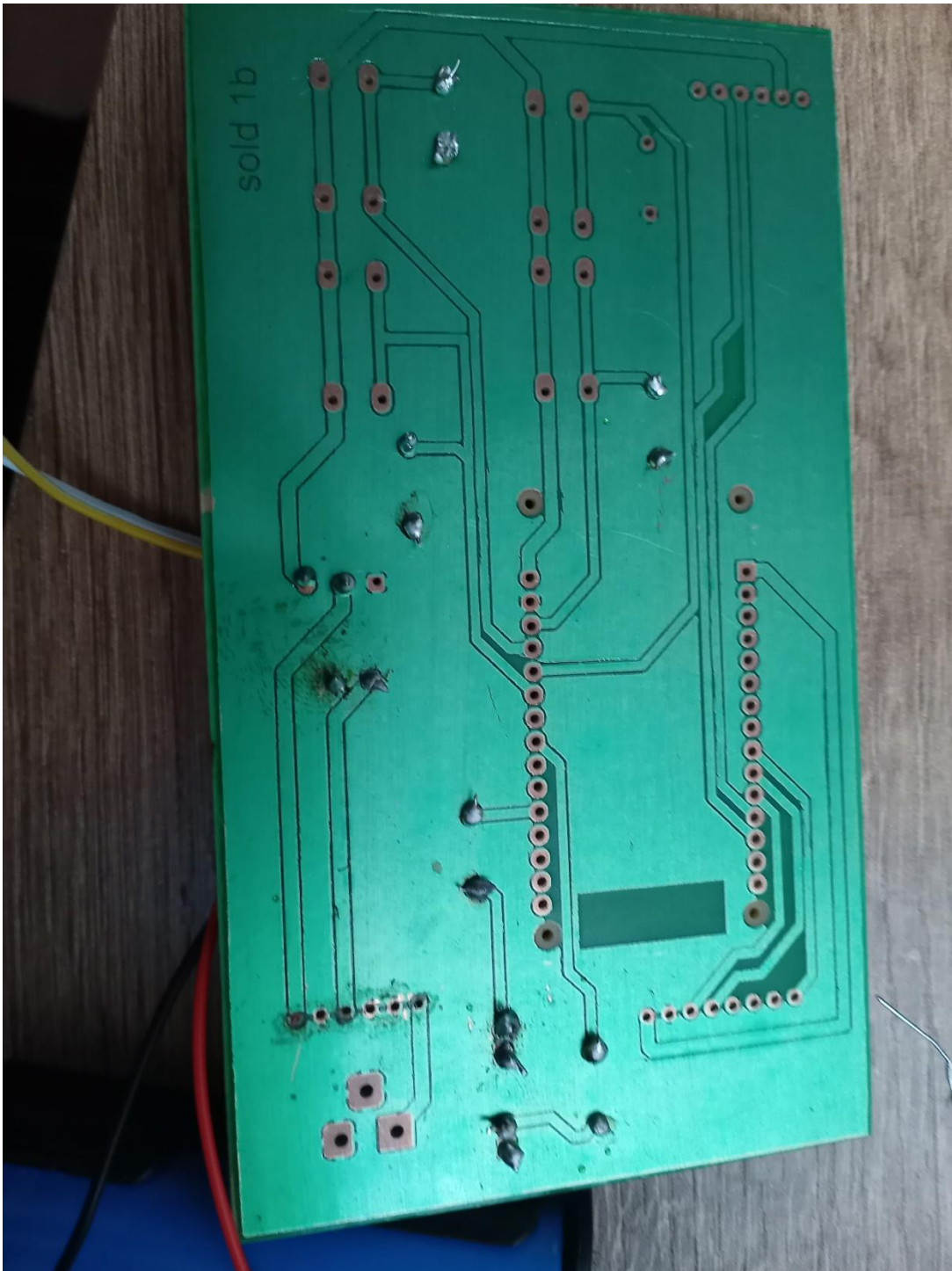
Imágenes en la Noche

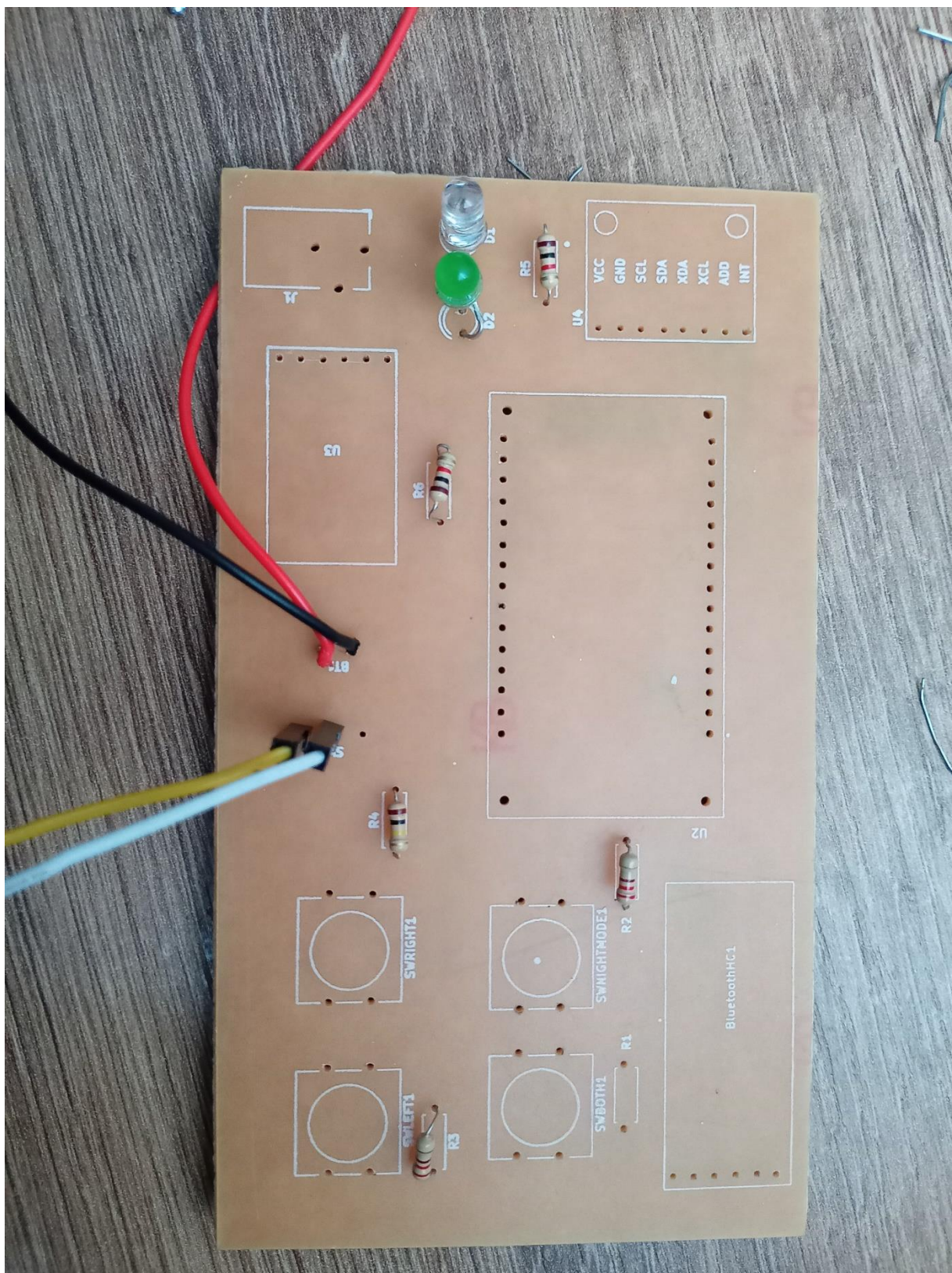


Imágenes en otra vista y con mas iluminación.

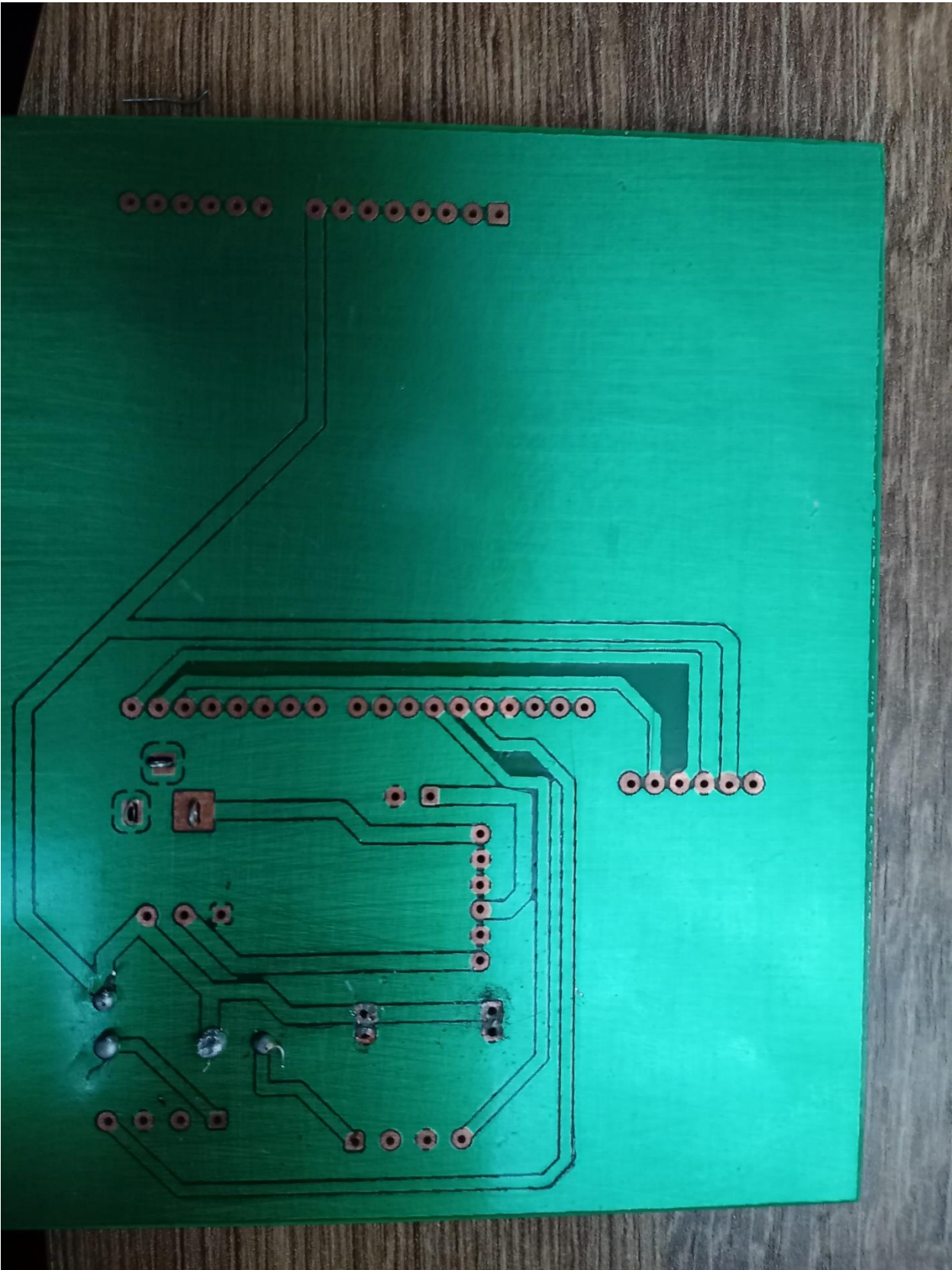
RESULTADOS FINALES

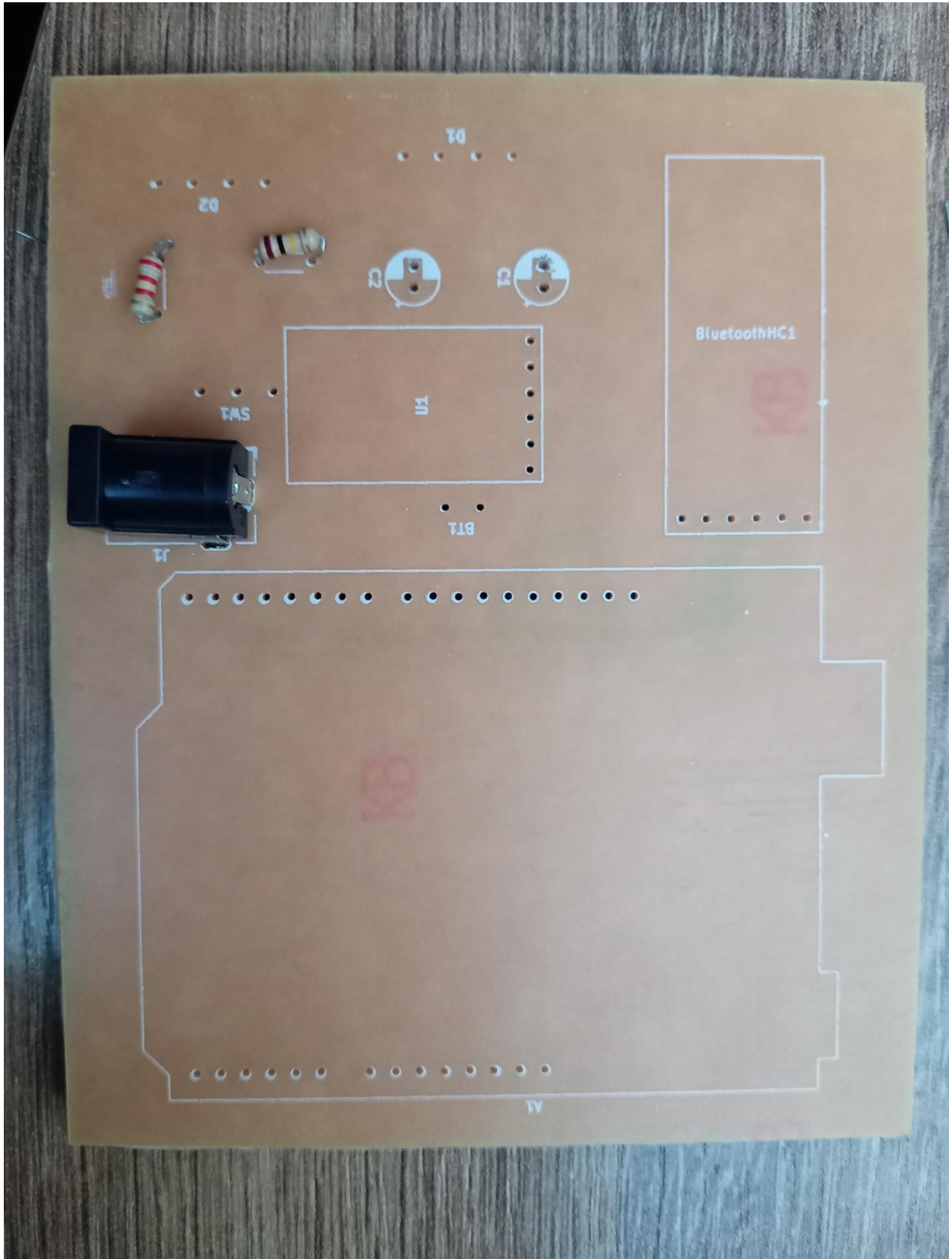
Primero quiero mostrarles los PCB.



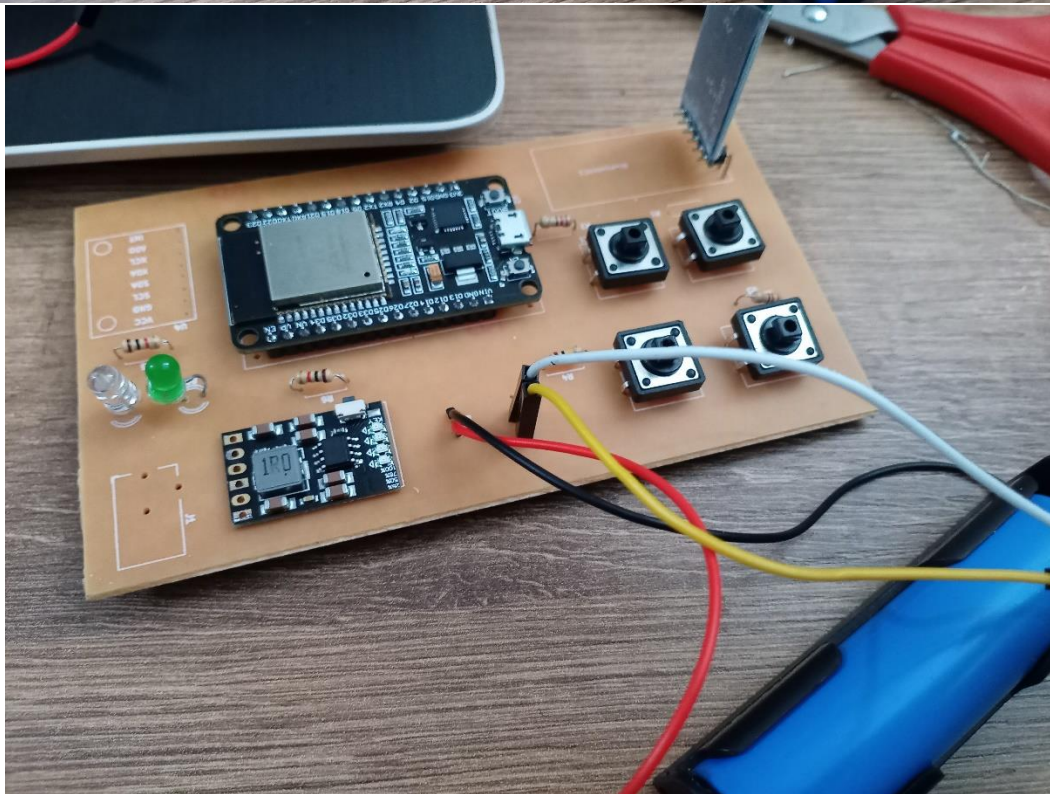
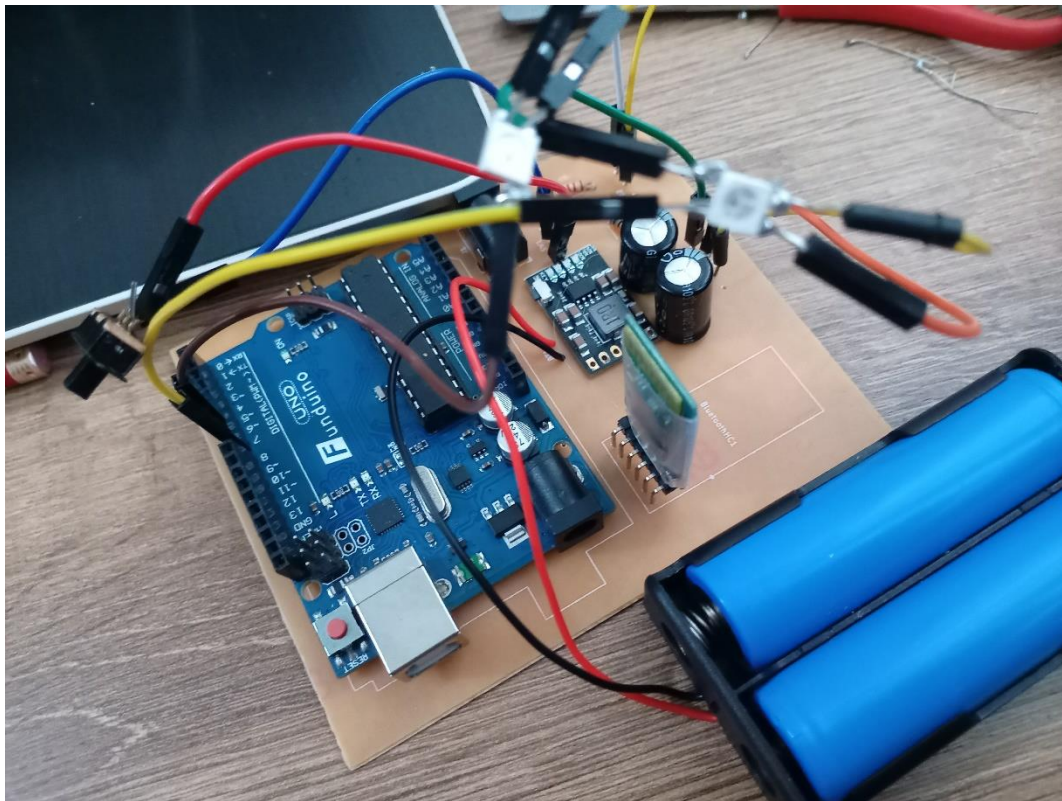


PCB Control Remoto





PCB Lucas



Visualización de como quedan conectadas

Como se puede ver, las PCB quedaron bastante estéticas y el dispositivo se ve bastante bien. Pero, lastimosamente las PCB no funcionaron, el ¿Por qué? Esta en la siguiente sección.

PROBLEMAS

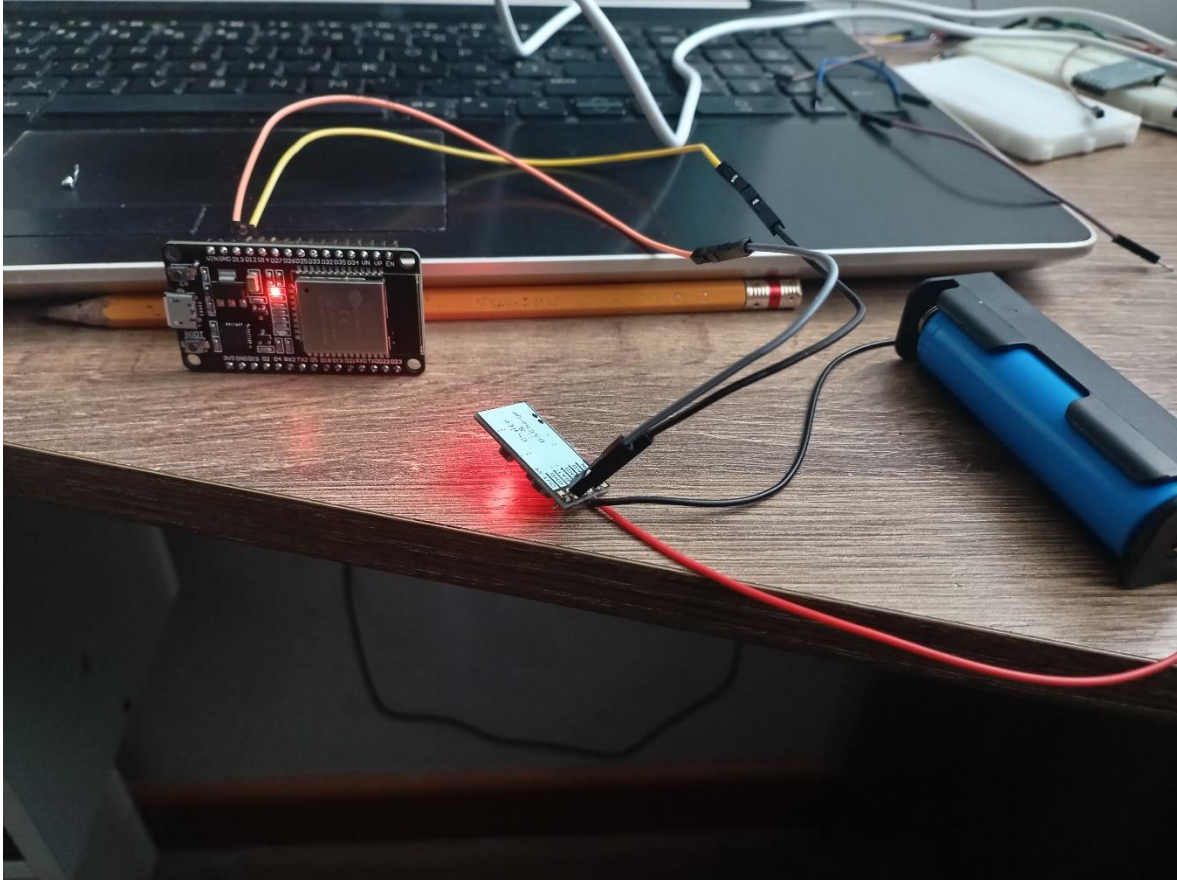


Imagen de pruebas del MH-CD42

Al conectar la PCB de Control remoto nos dimos cuenta de que algo malo estaba pasando, ya que ningún del rojo que indicara “encendidos” estaba alumbrando. Primero, tomamos un milímetro y medimos voltaje del interruptor al esp32 y nada, ceros. Después intentamos medir voltaje entre el Campo de GND de la placa y las primeras vías de alimentación (+5V, las de la batería, las del MH, etc), pero todas daban cero. Por lo que llegamos a dos hipótesis, o la PCB estaba dañada y no conducía o el MH estaba dañado (esta ultima ya era una pequeña muestra de esperanza ante la frustración). Como se ve en la imagen hicimos múltiples pruebas para la porta pilas, para el MH, y en todas funcionaban si no tocaban la PCB, por lo cual la primera hipótesis estaba casi corroborada. Por último, decidimos hacer pruebas en la PCB únicamente con los componentes de alimentación, colocando exceso de estaño, probando con Jumpers, etc. Pero nada funciono, la única manera de prender el MH era de forma directa como se debe ver en un video. También cabe

aclarar que los MH no vieron en perfecto estado, pero aun así fueron lo suficientemente útiles para determinar el problema.

Lastimosamente en estas pruebas, el esp32 se vio afectado y ahora este, aunque prende, los pines digitales y los pines Tx y Rx se dañaron, enviando señales o datos siempre, con lo cual la placa quedo inútil.

Así que, lastimosamente por la calidad de la PCB y/o la calidad de nuestras soldaduras terminaron por acabar todo en la recta final. Por lo cual, invitamos a todos los lectores a revisar y mejorar este proyecto, ya que este dispositivo tiene alta utilidad y podría ayudarnos a nosotros también, ya que, aunque haya llegado el deadline del proyecto, nosotros seguiremos mejorándolo.

A FUTURO

- Como fue por aprendizaje, usamos placas completas, pero esto ocupa mucho espacio, con lo cual, en un futuro de ser posible, quisiéramos implementar solo el chip del esp32 ya programado para el proyecto
- Ampliar la escala tecnológica del proyecto, ya que como nos enfocamos en solo las luces, no alcanzamos a explorar las posibilidades del ultrasonido, la temperatura, el sonido y demás utilidades que la electrónica otorga. Mejorar y profesionalizando el proyecto.
- Conseguir una PCB y el MPU para corroborar que lo hecho estaba bien hecho, o en su contrario encontrar alternativas que evitar los problemas mencionados.