Flight Controller PCB Design

Dip Ghodmare, Rhythm Gaidhani, Pratik Chahande, Shashank Yadav
Vidyadhar Kamble & Sunil Kahalekar
Makerspace, IIT Bombay

Table of Contents

1. **Introduction**
2. **PCB Overview**
3. **Setting Up a New Project**
4. **Creating the Schematic**
   - Adding Components
   - Wiring Components
   - Annotating the Schematic
5. **Designing the PCB**
   - Converting Schematic to PCB
   - Placing/arranging Components
   - Routing the Traces
   - Adding Copper Areas and Silk Screen
   - Design Rule Check (DRC)
6. **Generating Gerber Files**
7. **Soldering and Assembly Guide**
8. **Software Setup**
9. **Code Uploading**

References

---

1. Introduction

A flight controller PCB is the brain of a UAV or drone, integrating components that manage stabilization, navigation, and communication. This document provides an overview of the PCB, its key components, how they connect, and their role in achieving a functional flight controller.

2. PCB Overview

The PCB is designed as a compact, efficient board that houses all essential components required for controlling a drone. It includes connectors for peripherals like ESCs, power supply, and sensors while maintaining optimized signal flow and minimal interference.

The present drone PCB has two microcontrollers, one is the NANO which is used as flight controllers and the second is ESP-32, which acts as a wireless receiver from external devices such as mobile or remote controllers.

The drone PCB includes a power circuit that supplies 12 V to all ESCs. A diode is used to prevent the reverse current flow, protecting the system from potential damage. After the diode, the power is fed to a UBEC (step-down converter), which regulates 12 V to a stable 5 V for the logic circuit. This 5 V is being used to power the microcontroller and onboard sensors.

The ESP32 board is used as a wireless receiver for signals from external devices like mobile phones or remote controllers. It processes the received signals and transmits appropriate outputs to the NANO via pins X1 to X4, as shown in Fig. 2. A decoupling capacitor is used to minimize noise and interference on the PCB.

The NANO board is used as the flight controller in the drone PCB, collecting 3-axis motion data from the MPU-6050 (mounted on the drone frame) and signals from the ESP32. This data is used to calculate the required RPM for each motor and to adjust the drone's position via I2C communication. An onboard LED indicates the drone's battery status and provide feedback of battery capacity to prevent deep discharge during the flight. The PCB includes decoupling capacitors to minimize noise in signal and power lines.

3. Setting Up a New Project

1. **Create an Account:**
   - Go to EasyEDA's website and sign up for a free account.
   - Open the **Pro version** of it.
   - After signing in, the user will be directed to the EasyEDA workspace.

2. **Start a New Project:**
   - Click on **"New Project"** in the middle of the screen.
   - A new window will pop up to create a new project.
   - Enter a name, path and description for the project.
   - Click **"Save"** to start the project.

4. Creating the Schematic

Adding Components

1. **Open the Schematic Editor:**
   a. Schematic window as default, if not then select the project on the left side and select schematic.

2. **Search for Components**:
   a. To add the component, select the component from the top menu. A new pop-up window will open as a Device block, use the search option for the components as shown in Fig. 1.
   b. Copy the name of the components as given below in Table. 1 and paste it in the search engine of EasyEDA and match the footprint name. If not found, then select **Filter** through "**System, Project, OR Public**" to match the footprint name.
   c. If the Component is in the common library, then select the common library tab from left. (as shown in Fig. 1).
   d. Select the component with a matching footprint and place it in the Schematic.

3. **Components Placing**:
   a. Place the component in the Schematic as mentioned above for all the components.
   b. Arrange the components logically, keeping related components close to each other as shown in Fig. 2.
   c. Make sure that spacing between the components is at least 0.3 mm.

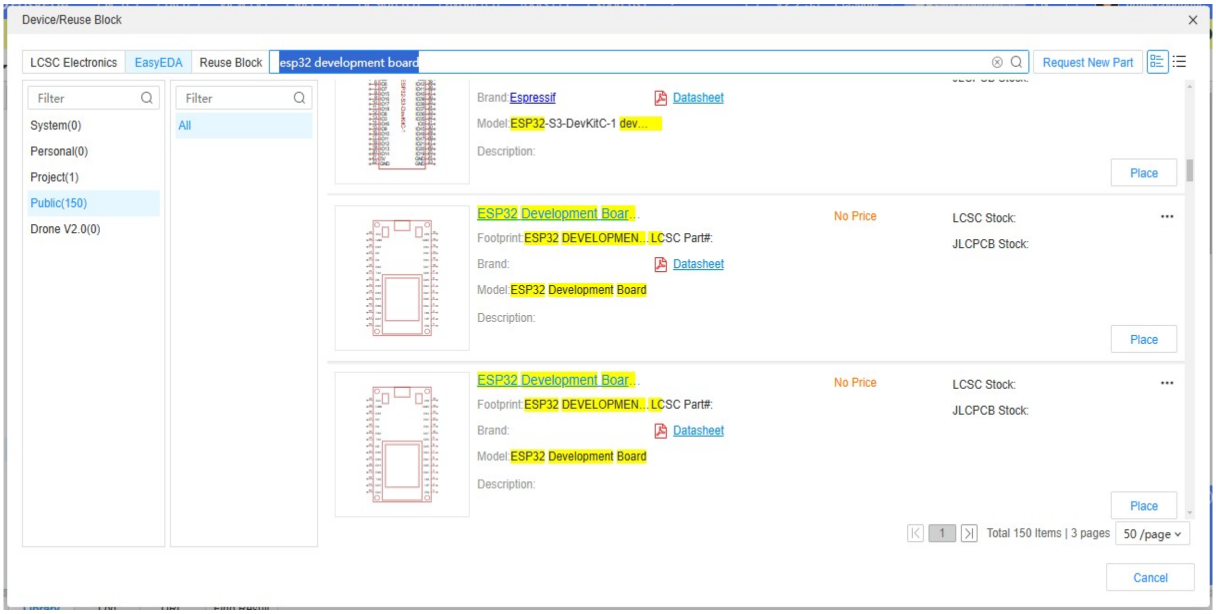


Fig. 1 Selection of Components for Placement in Schematic.

List of Components:

Table 1: List of Components with footprint name and source

Sr. No.	Component	Specifications/Values	Quantity	UoM	Designator	Footprint	Common Library	Project	Public	System
1	ESP32	ESP32 Development Board 30 pins, micro USB type	1	Nos	ESP32	ESP32 DEVELOPMENT BOARD ABUOMAR	-	✓	✓	-
2	Arduino nano	Arduino Nano, type C	1	Nos	Nano1	ARDUINO_NANO	-	✓	✓	-
3	Power connector-Type 1 & Type 2 pair	XT30 connectors Type 1- XT30PW-M Type 2- XT30U-F	4	pairs	ESC_PS1, ESC_PS2, ESC_PS3, ESC_PS4	CONN-TH_XT30PW-M	-	-	✓	-
4	Power connector-Type 3	XT60 connectors	1	Nos	BATT1	CONN-TH_XT60	-	-	✓	-
5	Diode	1N4007	1	Nos	D1	DO-41_BD2.4-L4.7-P8.70-D0.9-RD	-	-	-	✓
6	Resister	16kΩ, 1/4W	1	Nos	R4	RES-TH_BD3.3-L9.0-P13.00-D0.6	✓	-	-	-
7	Resister	10kΩ, 1/4W	1	Nos	R5	RES-TH_BD3.3-L9.0-P13.00-D0.6	✓	-	-	-
8	Resister	330Ω	1	Nos	R6	RES-TH_BD3.3-L9.0-P13.00-D0.6	✓	-	-	-
9	LED	LED_TH-R_5mm	1	Nos	LED1	LED-TH_BD5.8-P2.54-FD	✓	-	-	-
10	Capacitor	100uF Electrolytic capacitor	1	Nos	C1	CAP-TH_L5.8-W2.4-P2.54-D0.6	-	-	-	✓
11	Capacitor	10uF Electrolytic capacitor	1	Nos	C2	CAP-TH_BD6.3-P2.50-D1.0-FD	-	-	-	✓
12	Capacitor	0.1uF/100nF ceramic capacitor	5	Nos	U1, U2, U3, U4, U5	CAP-TH_L4.2-W3.8-P2.54-D0.5	-	-	-	✓
13	Female pin header	HDR-F_2.54_1x4P	2	Nos	MPU_6050, UBEC	HDR-TH_4P-P2.54-V-F	-	-	✓	-
14	Male pin header	HDR-M_2.54_1x3P	4	Nos	ESC-1,ESC-2, ESC-3,ESC-4	HDR-TH_3P-P2.54-V-M	-	-	✓	-

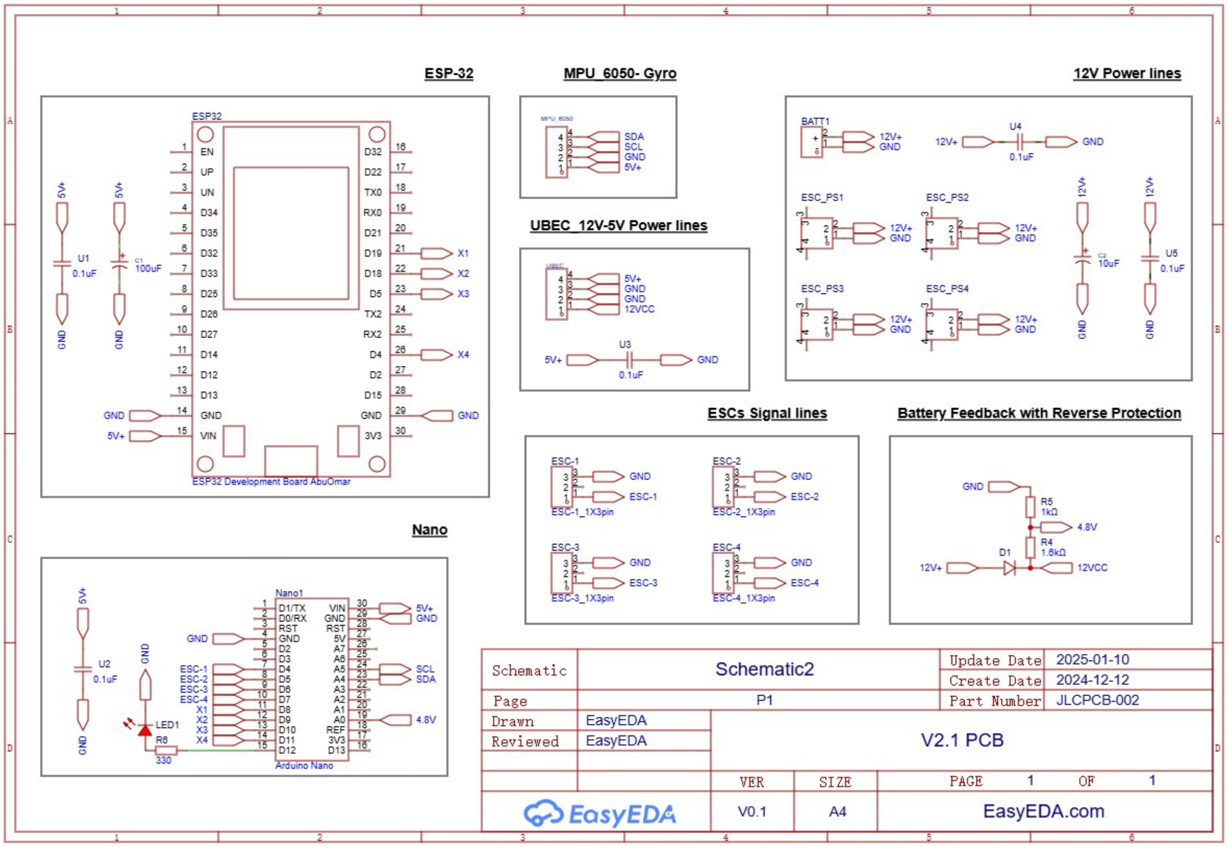


Fig. 2 Example of Component Placement in Schematic

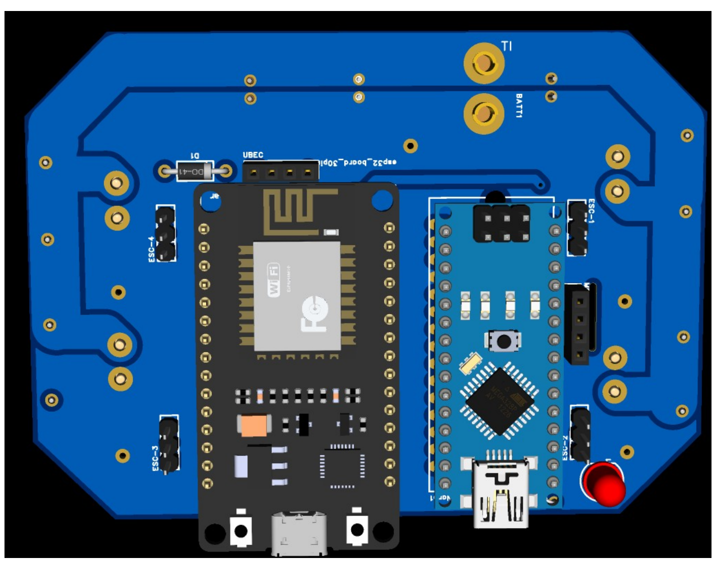


Fig. 3 Example of Component Placement in PCB.

Wiring Components

1. **Connect Components**:
    a. Make a connection as given in the schematic example shown in Fig. 2.
    b. Use the **"Wire"** tool from the toolbar to connect the components.
    c. Click on one pin of a component and drag the wire to the corresponding pin of another component.
    d. The "**Pin label IN/OUT**" as shown in Fig. 4 allows for cleaner and more organized routing by creating tracks instead of direct wire connections, helping to avoid clutter and unwanted interconnections, as shown in Fig. 2.

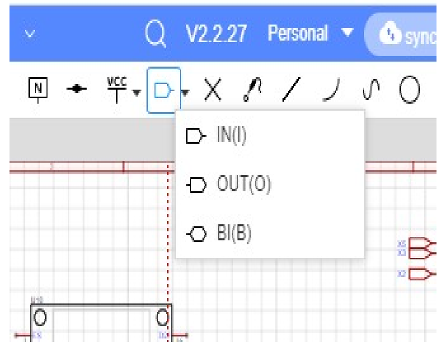


Fig. 4: Example of pin label

2. Add Power and Ground:

- Use the power and ground symbols from the library to connect the power supply to the circuit.
- Double-check the power connection while connecting
- Before switching to the PCB view make sure that the connections are proper.

Annotating the Schematic

1. **Add Labels and Text**:
    - Use the **"Text"** tool to add labels to components, making the schematic more understandable.
2. **Assign Reference Designators**:
    - Use the **"Annotate"** tool to automatically assign reference designators (e.g., R1, C1 OR X1) to the components.

12V Power Line

These are the main 12 V power connection which provide power to ESCs (Electronics Speed Controller) and UBEC (Universal Battery Eliminator Circuit). It also has a decoupling capacitor to reduce the noise in the power line.(Refer Fig. 5 and Fig. 6)

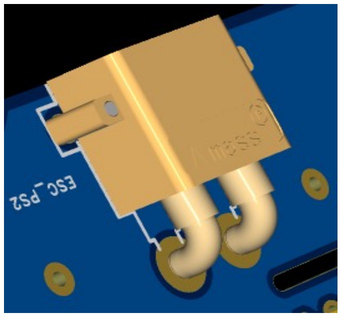


Fig.5. 12 V Connector (XT30 connector)

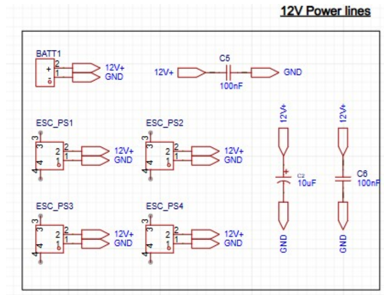


Fig.6 12 V power line

ESP32

The ESP32 board has a microcontroller with built-in Wi-Fi and Bluetooth capabilities, enabling wireless communication with external devices like remote controllers or mobile apps. It processes the received signals and convert them into appropriate commands for the flight controller. A decoupling capacitor is included to minimize power line noise, ensuring stable operation. ESP32 board and Schematic is shown in Fig. 7 and Fig. 8 respectively.

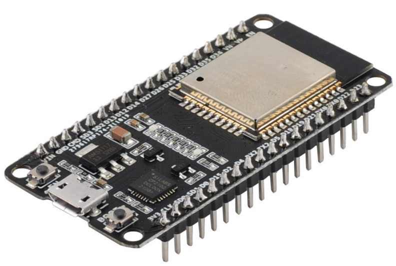


Fig.7. ESP32 Board [3]

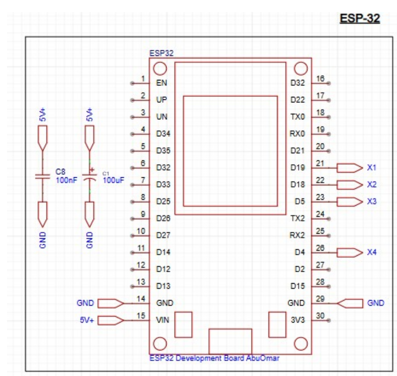


Fig.8. ESP32 Schematic [4]

MPU 6050

The MPU6050 is a 3-axis motion tracking device that integrates a 3-axis gyroscope and a 3-axis accelerometer in a single chip. It communicates with other devices using the I2C protocol. The gyroscope measures angular velocity, while the accelerometer detects linear acceleration. MPU6050 board and schematic are shown in Fig. 9 and Fig. 10 respectively.

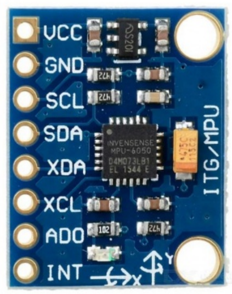


Fig. 9. MPU6050 Board [5]

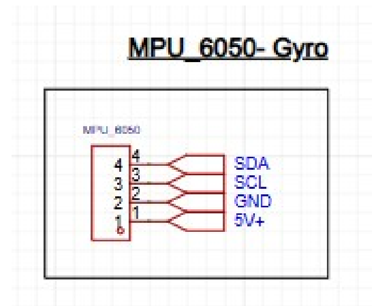


Fig. 10. MPU6050 Schematic

Battery Feedback Circuit

The battery feedback circuit shown in Fig. 11 is a straightforward yet effective voltage regulation system designed to step down the voltage to a safe level suitable for microcontroller operation. It includes a reverse protection diode to prevent unintended reverse current flow, safeguarding both the power source and the microcontroller from potential damage.

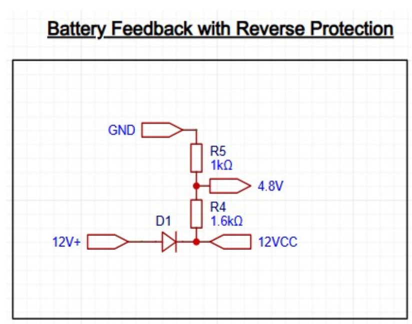


Fig. 11. Battery Feedback with Reverse Protection

UBEC

A UBEC (Universal Battery Eliminator Circuit) is a efficient electronic circuit that converts 12 V to 5 V, providing a stable power supply for logic circuits. It is ideal for powering components such as an Arduino Nano, ESP32, and various sensors. To integrate the UBEC, its input and output connections must be properly soldered to the corresponding power pins, ensuring reliable operation of the circuit.

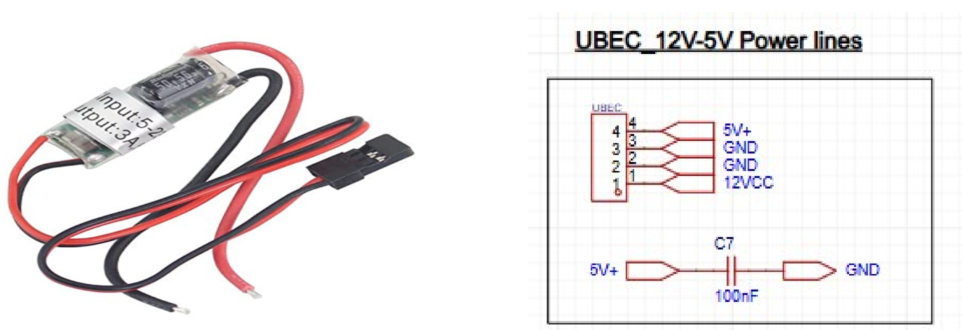


Fig. 12. UBEC [6]

NANO

The Arduino Nano microcontroller, controls all thefour ESCs, receives inputs from the ESP32, motion sensor, and a battery feedback circuit. It runs flight controller software for seamless operation.

Fig.13. Arduino Nano Board

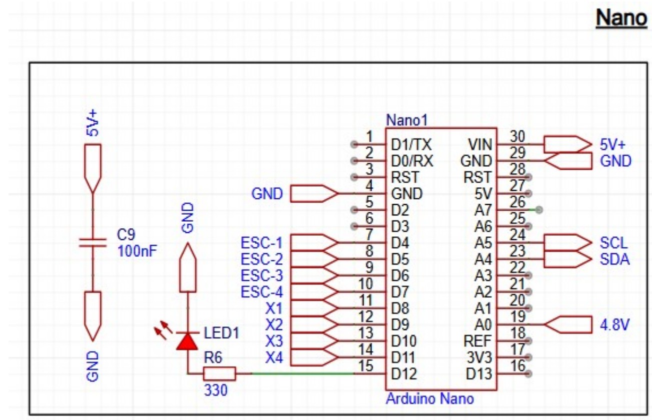


Fig.14. Arduino Nano Schematic [7]

ESC’s Signal Line

This 3-pin male connector as shown in Fig.15 is used to interface with each ESC, providing the signal and ground reference needed to drive the BLDC motors. The signal controls the RPM of each motor individually for precise operation.

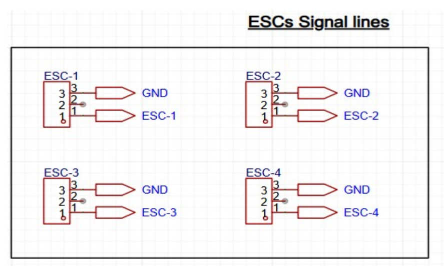


Fig.15. ESCs Signal Lines

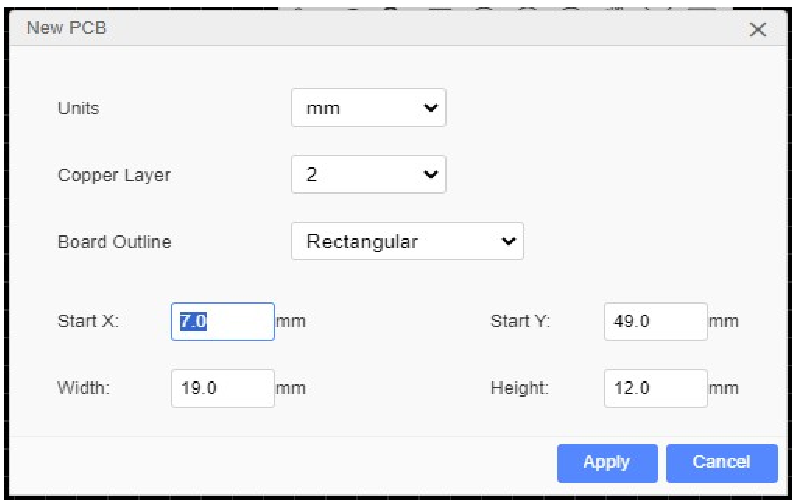
5. Designing the PCB

Converting Schematic to PCB

- After completing the schematic, click on save and then click **"Design"**>**"Convert to PCB"**.
- A new window (Fig. 16) will pop up as a new PCB, select the units in mm, copper layer "2", board outline as "**Rectangular**", select the width and height of the PCB as 98 mm x 71 mm or select "**Custom**" board outline to select dimension as per the designer requirement and select "Apply".

Fig. 16. New PCB Creation.

Placing Components



The components will automatically be placed on a new PCB canvas, once it gets converted from schematic to PCB.

Arrange Components

- Drag the components to the desired locations on the PCB as shown in Fig. 17.
- Place components logically to minimize trace lengths and avoid overlap.

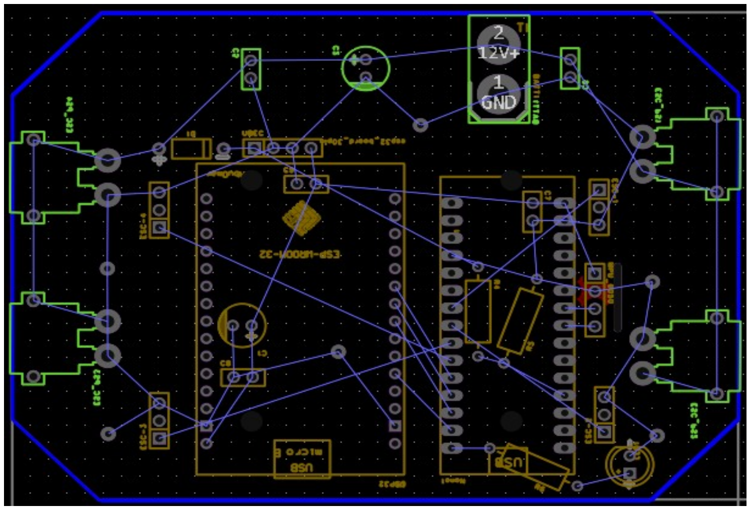


Fig.17. Example of Converting Schematic to PCB & components arrangement

Traces Routing

1. **Manual Routing**:
   - Use the **"Route"** tool to manually draw the traces between component pads.
   - Ensure traces (track) do not cross and avoid sharp turns for better signal integrity.
   - Maintain track width at least 1.25 mm and 1.5mm for the signal wire and the power supply respectively and make sure that the main **12V power** track width is at least **7.5mm**(Refer to Fig. 18.). (or adjust based on current requirement) (Refer to Fig. 18 for details.)

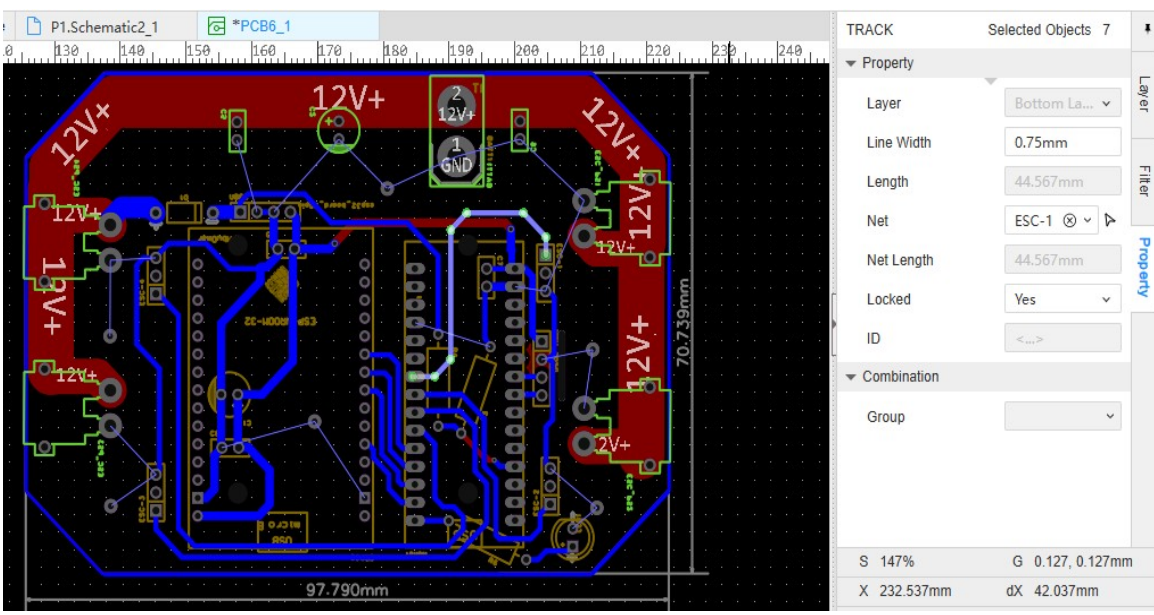


Fig.18. Track width

- The pad size should be >1.6 mm. It makes soldering a lot easier, click on component pad, go to the property (on left hand side) > Size > edit diameter to 1.6 or greater. (As shown in Fig. 18)
- Try to make the PCB in the bottom layer only. (As shown in Fig. 20)

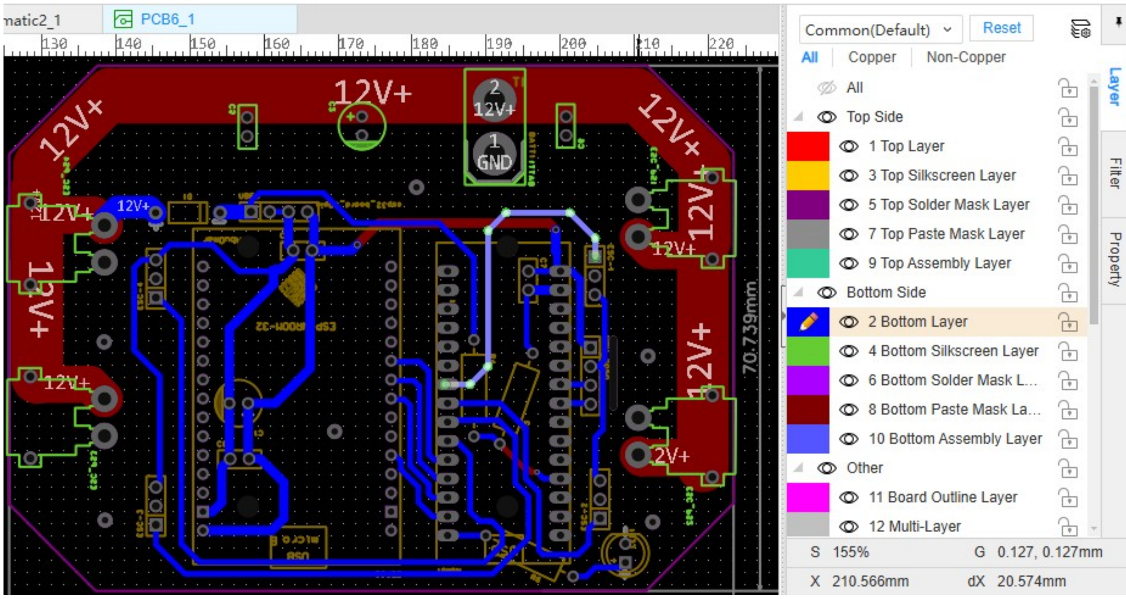


Fig.21. Layers of the PCB.

- Add mounting pads to secure the PCB, ensuring stability.

2. **Auto-Routing**
   - Alternatively, **"Auto Router"** tool can be used to automatically route all connections.
   - Review the auto-routed traces for any potential issues.
   - **It is better not to use auto-routing** and do manual routing as shown in Fig. 21.

3. **Lock Components**
   - Once satisfied with the placement, right-click on a component and choose **"Lock"** to fix its position.

4. **Add Mounting holes**
   - Add 4 mounting holes to the PCB which are **3mm Diameter**, Center to Center distance between them is **35mm** it can be changed according to your design.
   - Use a 3mm hole size in each pad to accommodate 3mm nuts and bolts for reliable mounting.

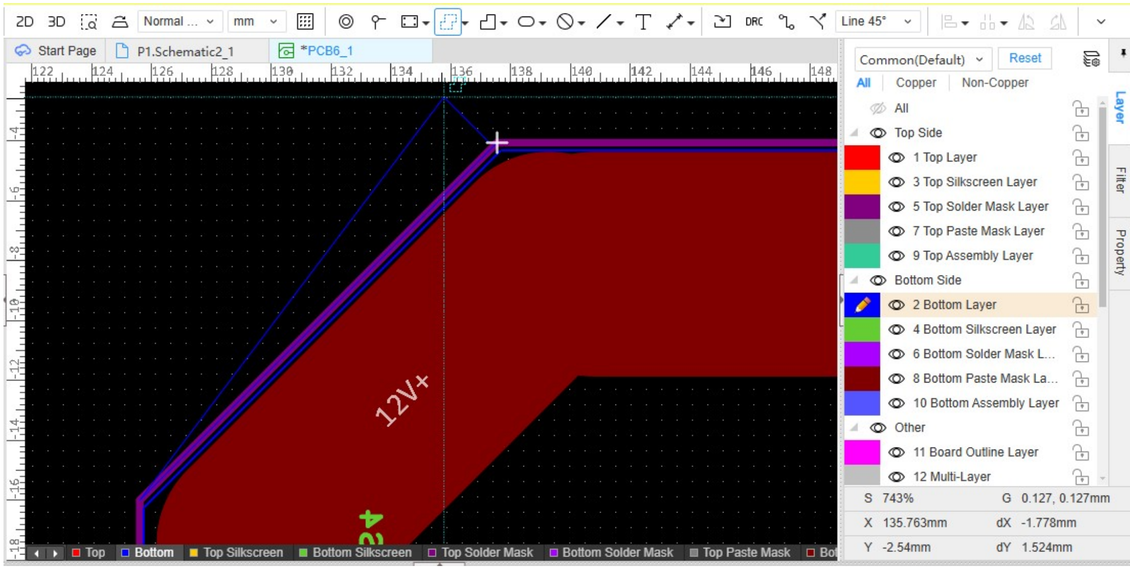


Fig.19. Copper area selection.

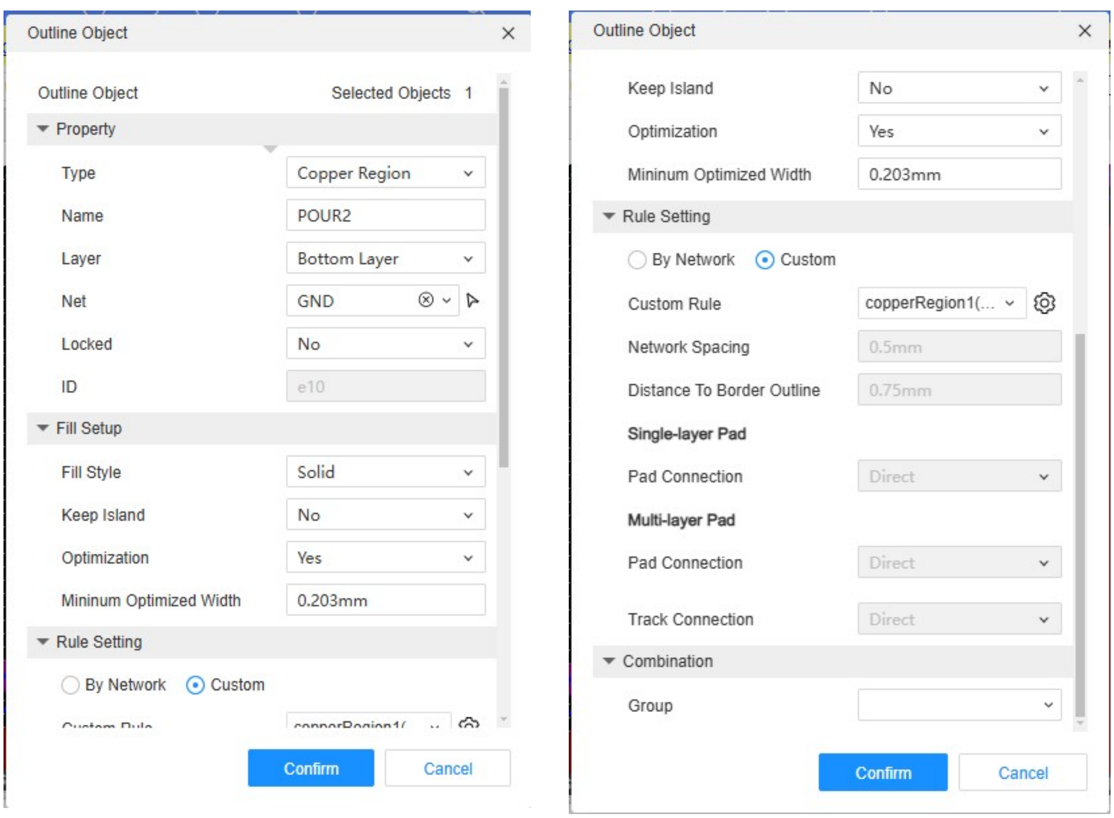


Fig.20. PCB Outline

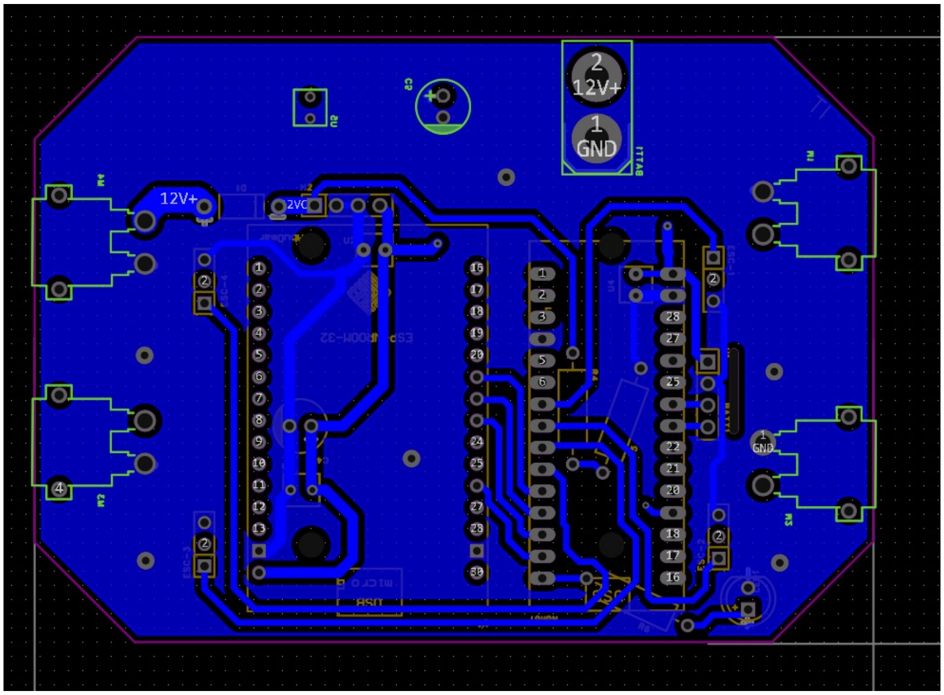


Fig.22 Example of manually Routed PCB.

Adding Copper Areas and Silk Screen

1. **Copper Areas**:
   - Draw the copper area on the desired layer (e.g., top layer, bottom layer).
   - Go to the **copper area** and select the area as per your requirements **“Type = Copper Region”** and **“Net = GND”**(Refer to Fig. 19 & 20 for details.)

Design Rule Check (DRC)

1. **Run DRC:**
   - Click on **"Design Rule Check (DRC)"** to check for any design rule violations.
   - Fix any errors or warnings before proceeding.

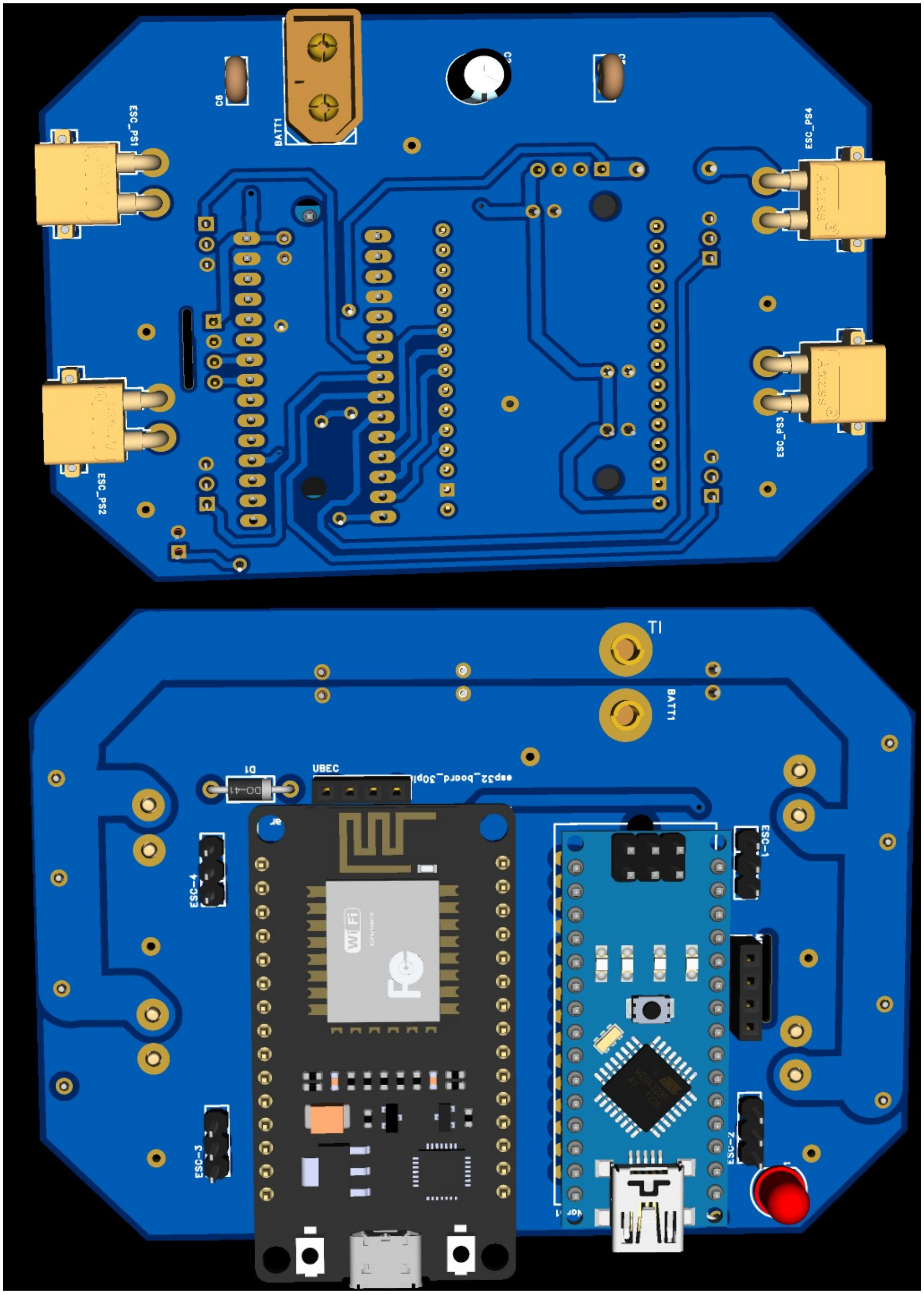
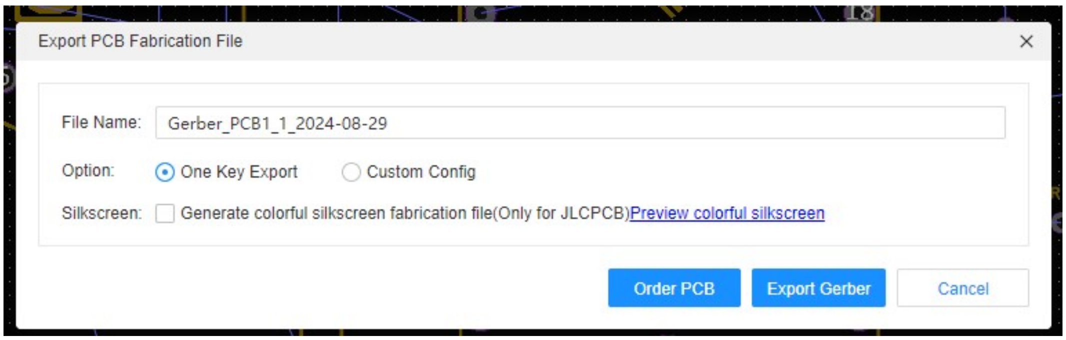


Fig.23 3D PCB Model.

6. Generating Gerber Files

1. **Generate Gerber Files**:
    - Once the PCB design is complete, save the project and click **"Export(R)>PCB Fabrication File (Gerber)."**>**"Export Gerber"** as shown in Fig.26.
    - Review the layers and settings, then click **"Generate Gerber"**.

2. **Download Gerber Files**:
    - After the files are generated, download them to the computer.



    - These files can now be sent to a PCB manufacturer for production.

Fig.26 Example of Gerber File Generation

3. **Download 3D file.**
    - After downloading the Gerber file, **Go to Files>Export(E)> "3D file"** as a step file as shown in Fig.
    - This step file is further useful for creating the 3D case of the remote in Fusion 360.

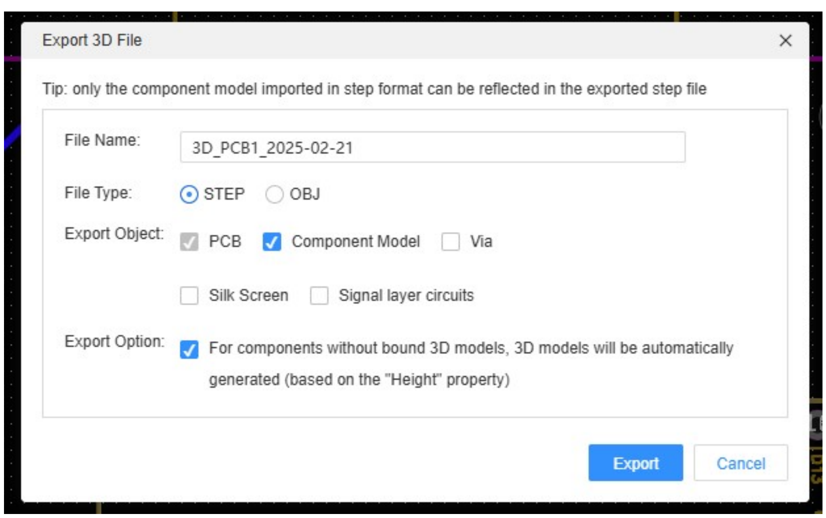


Fig. 27 Export in 3D file

4. Open the fusion then Go to **File > Upload** >Once uploaded right click on the step file and click add to the current design to initiate the 3D case design.

7. Soldering and Assembly Guide

Quick Description of the Soldering Process

Soldering is the process of joining electronic components by melting solder around the connection. Follow these steps for proper soldering:

1. Prepare the Workspace: Ensure a clean, well-ventilated area and gather tools (soldering iron, solder, flux, sponge, safety goggles).

2. Heat the Soldering Iron: Set the soldering iron to **320°C** (ideal range: **300°C–400°C**).

3. Clean & Tin the Tip: Wipe on a damp sponge, then apply a small amount of solder for better heat transfer.

4. Position Components & Align with the Silkscreen Layer: Align with the **PCB silkscreen**, ensuring correct placement and orientation of polarized components (diodes, capacitors, etc.).

5. Apply Flux: Apply flux to the component leads and PCB tracks to improve solder flow.

6. Apply Heat & Solder: Heat the component lead and pad, then apply solder (not directly on the iron)

7. Cool & Inspect the Joint: Let joints cool naturally; they should be shiny, smooth, and concave.

8. Clean Up: Turn off the iron, clean the tip, and store it safely.

Soldering Sequence of the Components

1. Small Components (Resistors & Capacitors)

- Start with **100 nF, 100 µF capacitors**, and resistors.
- Ensure components are **flush with the PCB and aligned with the silkscreen layer**.

2. Solder Female Header Strips

- Solder female headers for **ESP and Nano** and then solder **3-pin male headers** for ESC signal wires.

3. Solder Remaining Small Components

- Attach **LEDs, diodes**, **UBEC** and other small parts.

4. Solder Large Components

- Solder the **XT-30 connector** on the PCB (from the bottom side).
- Solder the **XT-60** battery Connector.

⚠Warning

Limit soldering iron contact time to 5 seconds per pad to prevent damage.

How to De-solder using a De-soldering Pump

Tools Needed: Soldering iron, De-soldering pump (solder sucker), and Tweezers

Steps:

1. **Heat the Solder Joint**
   - Use a soldering iron (320-350°C) to melt the solder from the particular pad.
2. **Use the De-soldering Pump**
   - Place the pump near the molten solder.
   - Press and release the plunger to suck up the solder (As per the Fig 27.5).
3. **Repeat & Remove Component**
   - If needed, reheat and use the pump again.
   - Use tweezers to remove the component once the solder is cleared.

Tips:

- Work quickly to avoid overheating.
- Keep the soldering iron tip clean.
- De-Soldering Wick can be used for Fine cleaning.

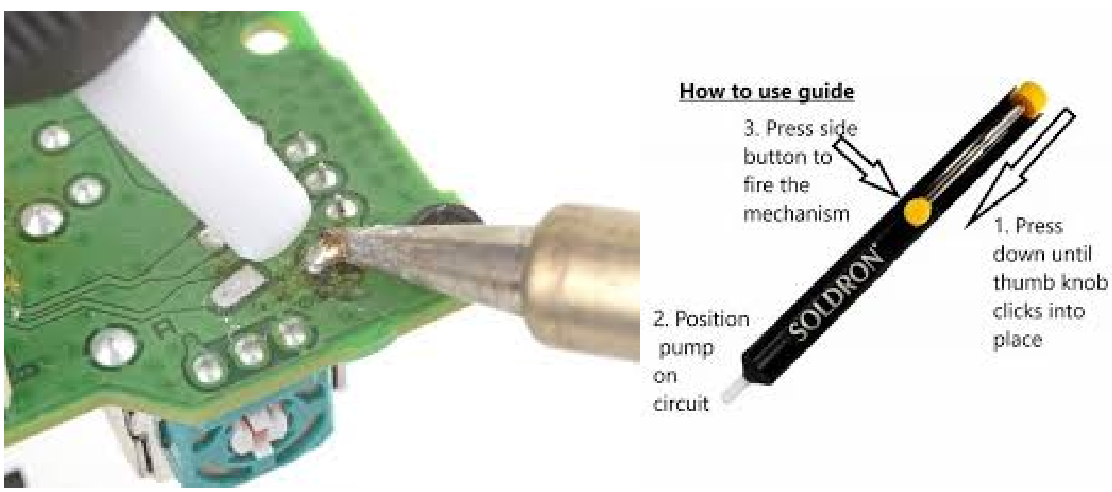


Fig. 27.5 De-Soldering Pump.

8. Software Setup

Required Tools and Libraries

- **Arduino IDE**

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application available for **Windows, macOS, and Linux**, designed for programming Arduino-compatible boards using **C and C++ functions**. It features a user-friendly code editor with **syntax highlighting, brace matching, and automatic indentation**, enhancing code readability and efficiency. Additionally, the IDE includes a **console for text output** from the Arduino board and a **toolbar** with essential functions such as **compiling and uploading code**, making development seamless and efficient.

The Arduino IDE simplifies the process of coding and interfacing with microcontrollers, making it accessible for beginners and professionals alike. It supports a wide range of Arduino boards and modules and provides an extensive library system to extend its functionality.

Download Link

- Arduino IDE Download and install it.

- **How to Install a Library in Arduino IDE**

Using the Library Manager

1. **Open Arduino IDE**: Launch the software.
2. **Open Library Manager**: Go to **Sketch > Include Library > Manage Libraries**
3. **Search**: Type the library name in the search bar.
4. **Select**: Click on the desired library.
5. **Install**: Click the **Install** button.
6. **Include**: Use #include <LibraryName.h> in your sketch.

Manual Installation

1. **Download Library**: Obtain the .zip file from a trusted source.
2. **Open Arduino IDE**: Launch the software.
3. **Add . ZIP Library**: Go to **Sketch > Include Library > Add .ZIP Library...**.
4. **Select File**: Choose the downloaded .zip file.
5. **Include**: Use #include <LibraryName.h> in your sketch.

- **Library in Arduino IDE Should be pre-installed**

1. **ESP-NOW Library**
   a. Library for ESP-NOW communication
   b. Include in code: #include <esp_now.h>
2. **WiFi Library**
   a. Library for WiFi functionality
   b. Include in code: #include <WiFi.h>

3. **ESP32Server Library**
    a. Library for ESP32 server functionality.
    b. Include in code: #include <ESP32Server.h>
4. **Wire Library**
    a. Library for ESP WiFi functionality
    b. Include in code: #include <Wire.h>
5. **EEPROM Library**
    a. Library for EEPROM (Electrically Erasable Programmable Read-Only Memory) functionality
    b. Include in code: #include <EEPROM.h>

- **Installation Links**

These libraries can be installed manually through the Library Manager in the Arduino IDE or by downloading them from official sources. The following libraries are readily available within the Arduino IDE:

- **ESP-NOW Library**
- **WiFi Library**
- **ESP-WiFi Library**
- **ESP-ADC Driver Library**
- **EEPROM Library**

9. Code Uploading

Pairing Instructions

Step 1: Import the Sketch in Arduino IDE

1. **Open Arduino IDE**: Launch the Arduino IDE on the computer.
2. **Open the Sketch**:
   - Go to File > Open...
   - Navigate to the directory where the sketch file (usually with a .ino extension) is located.
   - Select the sketch file and click Open.

Step 2: Select the Correct Board and Port

1. **Select the Board**:
   - Go to Tools > Board > ESP32 Arduino > DoIT ESP32 DEVKIT V1.
   - Ensure that the correct board is installed. If the file "DoIT ESP32 DEVKIT V1" is not in the list, install the ESP32 boards package. Follow these steps:
     - Go to File > Preferences.
     - In the "Additional Boards Manager URLs" field, add the following URL: https://dl.espressif.com/dl/package_esp32_index.json
     - Click OK.
     - Go to Tools > Board > Boards Manager.
     - Search for esp32 and install esp32 by Espressif Systems.
2. **Select the Port**:
   - Connect the ESP32 board to the computer using a USB cable.
   - Go to Tools > Port.
   - Select the port that corresponds to the "DoIT ESP32 DEVKIT V1" board (e.g., COM3, COM4, etc.). The correct port usually appears after connecting the board, in this case as (USB).

Step 3: Upload the Sketch

1. **Verify the Sketch**:
   - Click the checkmark icon in the top-left corner of the Arduino IDE to verify the sketch.
   - Ensure there are no compilation errors. If there are errors, check the code and resolve any issues.
2. **Upload the Sketch**:
   - Click the right-arrow **(Upload icon -** next to the checkmark) to upload the sketch to the ESP32 board.
   - Wait for the upload process to complete. The messages indicating the progress in the console window at the bottom of the Arduino IDE will be displayed.
   - If there is a problem regarding the uploading of the code contact to the concern TA.

Troubleshooting Tips

- **Common Issues**:
  - **Compilation Errors**: Ensure all the required libraries are installed. Go to Sketch > Include Library > Manage Libraries and install any missing libraries as shown in Fig. 24 and Fig. 25.

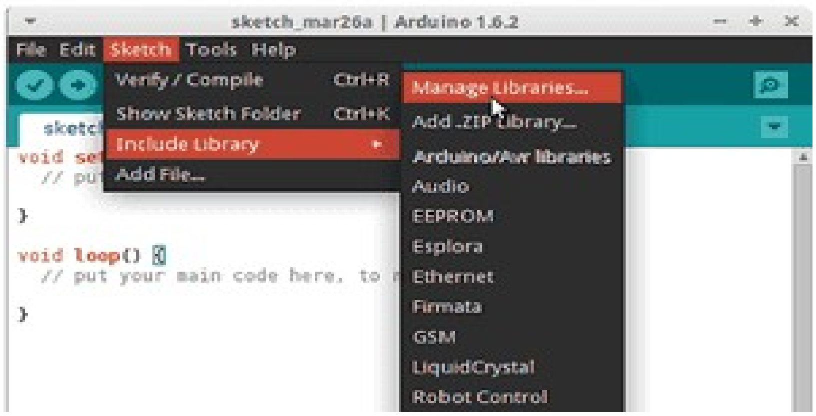


Fig.24. Library Installation

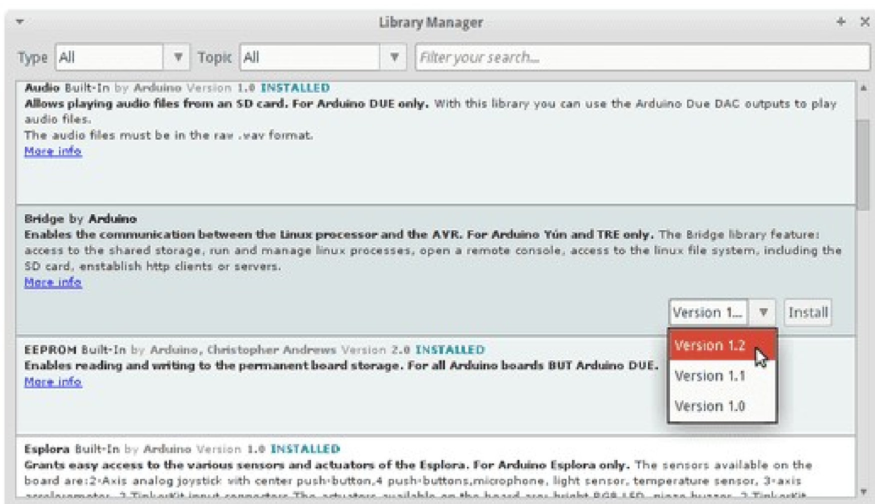


Fig.25. Library Manager (for searching and installing particular library)

  - **Upload Failures**: Double-check the selected port and board. Make sure the ESP32 **(DoIT ESP32 DEVKIT V1)** board is properly connected to the computer. Try a different USB cable or port if necessary.

- **Contacting Support**:
    - If you encounter any issues during the upload process that cannot be resolved, please contact to concern TA or the staff member for assistance. Provide details about the problem and any error messages that you receive.

Acknowledgement

We, the authors, would like to express our sincere gratitude to the management of the Makerspace for giving us the opportunity to be part of the Makerspace, an innovative learning environment.

We are deeply grateful to the dedicated staff of the Makerspace, whose tireless efforts and expertise have been instrumental in maintaining and optimizing this creative hub.

Our sincere gratitude goes to the faculty associated with the Makerspace, whose guidance and mentorship have been invaluable in fostering a culture of innovation and hands-on learning.

In addition, we would like to acknowledge Technoventor Innovations Pvt Ltd, for their pivotal role in setting up this state-of-the-art lab and allowing us to learn and contribute in the Makerspace academic activities.

References:

1. Designing PCBs with Easy EDA (Intro to Prototyping)
2. BEGINNERS PCB DESIGN USING EASY EDA...
3. https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/index.html
4. https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DEVKITM-1_V1_20210310A.pdf
5. https://makeradvisor.com/tools/mpu-6050-3-axis-accelerometer-and-gyroscope-sensor/
6. https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2371/1385_Web.pdf
7. https://www.electronicshub.org/arduino-nano-pinout/