

# DigiKey

[Login](#)[Create Maker Profile](#)

We built a Drone using ESP32 | Now on Kickstarter



## A Step-by-Step Guide to Build a Low-Cost Drone using ESP32

0

2024-04-12 | By [Aswinth Raj](#)

License: GNU Lesser General Public License

[DC Motor](#)[Wifi](#)[ESP32](#)

Drones have rapidly evolved from niche hobbies to versatile tools with a wide range of applications, from photography to agriculture or even for defence and military purposes. Depending on the application, features, and payload capacity their price will vary from pocket changes to a few million. Even though the basic principle behind these drones may appear simple, there is a lot of technology and computation that goes behind it for proper operations and the desired result from these drones—for example, maintaining stability

while in the air requires precision sensors such as a Gyroscope and proper processing of reading taken with it. In this project, we are going to make a drone that can be easily controlled using our phones.

This DIY drone is small in size and can be built using easily available components such as ESP32 modules, MPU6050 IMU, coreless motors, and plastic propellers.

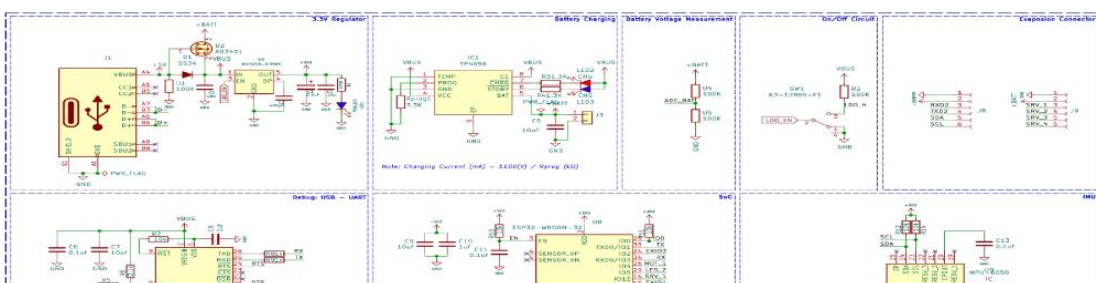
### Features of our DIY WiFi-Controlled Drone

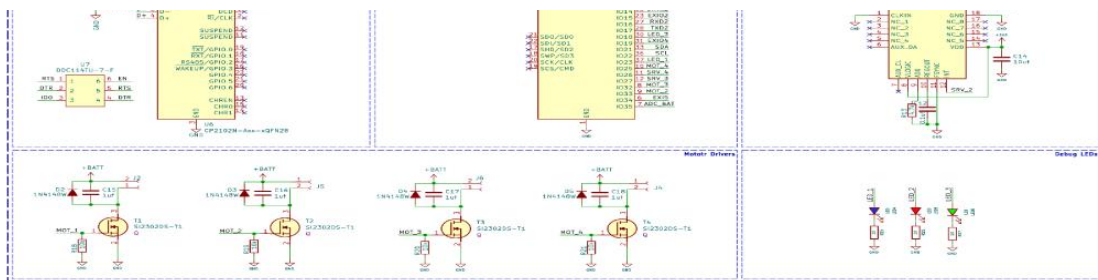
- WiFi controlled: This can be controlled using a smartphone.
- MPU6050 IMU for stability control.
- All-in-one PCB: Doesn't need any 3D printed parts or such.
- Easily upgradable: Additional features such as position hold or height hold can be added using external modules.
- Small size and lightweight.
- Built-in battery charger.
- Built-in USB interface for programming and debugging.
- Android and IOS apps.
- Open source



### DIY WiFi-Controlled Drone Complete Circuit Diagram

The complete circuit diagram for the **DIY WiFi Controlled Drone** is shown below.



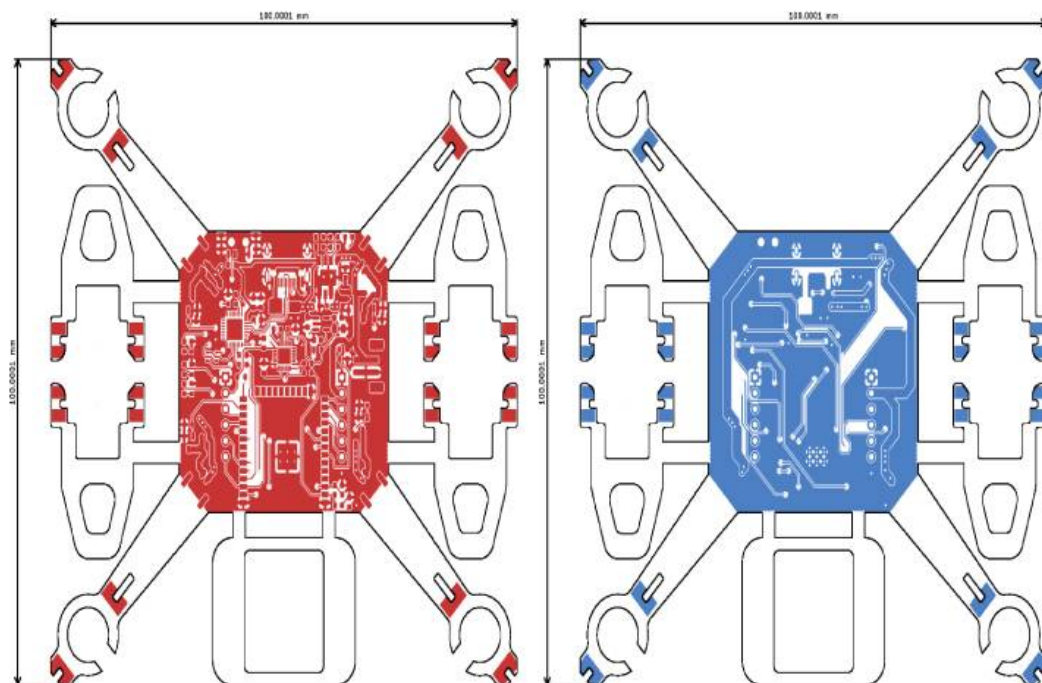


A Type C USB port serves both charging and programming functions. Its power path controller, utilizing a P-Channel MOSFET U2 and diode D1, allows seamless switching between USB power and battery power, with voltage regulation ensured by the MIC5219 3.3V LDO from Microchip. A slide switch activates the thermal camera while a TP4056 charge controller manages battery charging. Voltage sensing is facilitated by a classic voltage divider circuit. Moving to the programming section, an ESP32 SoC, CP2102 USB to UART controller, and MPU6050 PMU chip form the core. The CP2102, coupled with a dual MOSFET, simplifies ESP32 reset functionality, while standard circuitry surrounds the ESP32. The MPU6050 PMU aids in flight stabilization and motion control, interfaced with the ESP32 via GPIO pins. Finally, motor driver circuits employ SI2302 N-Channel MOSFETs for motor control, supplemented with protective diodes and resistors. Debugging LEDs, including a blue LED for calibration, a green LED for connection detection, and a red LED for low battery indication, enhance operational monitoring alongside power and charging indicators.

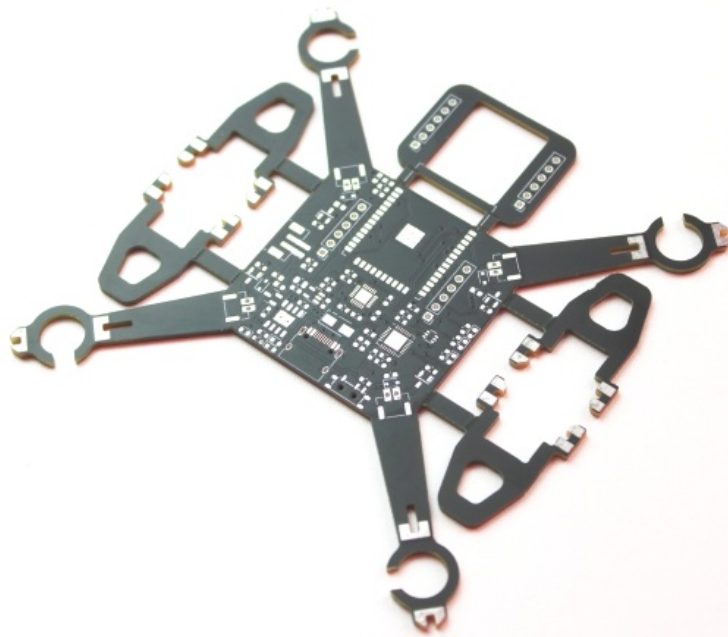
You can find more detailed explanations and other details on the project page [Let's Build a Low-Cost Drone using ESP32](#).

### PCB for DIY WiFi-Controlled Drone

For this project, we have decided to make a custom PCB. This will ensure that the final product is as compact as possible as well as easy to assemble and use. We have also designed the PCB in a way that the feet for the drone are also included in the PCB and Can be easily broken away from the main PCB. Here are the top and bottom layers of the PCB.



Here is the PCB.

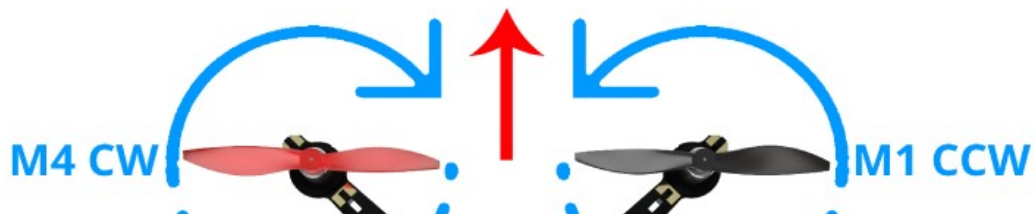


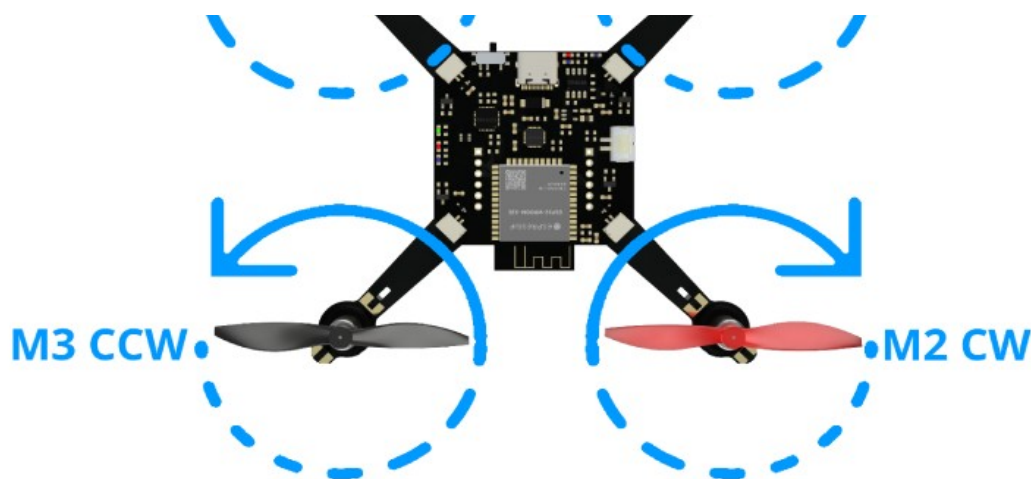
And here is the fully assembled drone.



### Propeller Direction

Install A and B propellers according to the figure below. During the power-on self-test, check if the propellers spin properly and are spinning in the correct direction.





### Firmware for the DIY WiFi-Controlled Drone

The firmware from our DIY drone is based on the ESP-drone firmware from **Espressif**. The code is written with ESP-IDF, and the version that is used to compile this code is ESP-IDF 4.4.5. Follow the following link to install and configure [ESP32 IDF version 4.4.5](#). You can either build the firmware from scratch using the source provided at the Github repo below or you can just flash the binary file provided within the same repo if you don't want the hassle. Make sure to use the source code provided within the below GitHub, because there are many modifications on the Espressif's code to suit our PCB design.

### Flashing the Firmware

To flash the code on our ESP32 Drone, you can follow any one of the below three methods.

#### Method 1: Building from Source using ESPIDF

- To start with install and configure ESP-IDF. Please follow the detailed instructions from [Espressif](#). Make sure to install version 4.4.x of ESP-IDF.
- Once the ESP-IDF is installed clone the ESP-Drone firmware repo using git and navigate to the firmware folder.

```
git clone https://github.com/Circuit-Digest/ESP-Drone.git cd ESP-Drone/Firmware/esp-drone
```

- You can change all the firmware configurations using menuconfig. For our use case, the current configuration is enough, and you don't need to change anything.

```
idf.py menuconfig
```

- Now to build and flash the firmware you can use the flash command. It will automatically build and flash the project. Replace PORT with your ESP32-S2 board's serial port name.

```
idf.py -p PORT flash
```

#### Method 2: Using ESPTOOL

To use the ESPtool, make sure you have installed the ESP IDF. You can find the instructions for that in the previous section. Once installed and configured open the terminal in the same folder as the firmware image and use the following command to flash the firmware.

```
esptool.py write_flash --flash_size detect 0x0 ESPDrone.bin
```

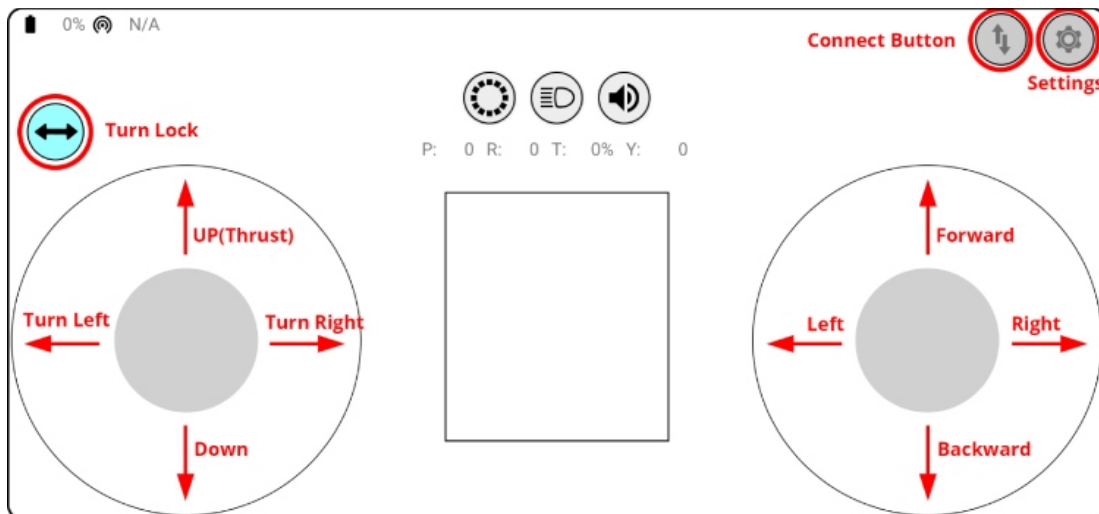
#### Method 3: Using ESP32 Flash Download Tool

- First download the ESP32 Flash Download Tool
- Extract it to a folder and double-click on the exe file to run it. When prompted select ESP32 in the chip type field and click on OK.
- Select the firmware file with the ESPDrone.bin file and add the address as 0x00. Select the proper com

- port and click on erase. Once the flash erase is complete click on START to flash the firmware.
- That's it. You are ready to use your DIY drone

## Using The Drone

Place the drone on a flat surface and turn it on. As soon as it's turned on the flight controller will create a WiFi Hotspot. Connect to it using the password 12345678 and open the app. For the IOS the app can be downloaded from the App Store, just search for ESP-Drone APP. For Android, you can download the app from the following [link](#). Keep in mind that the app is created and hosted by a third party. So, install them at your own will. The app interface will look like this.



Click on the connect button to start communicating with the drone. When the connection is established successfully between your drone and the APP, the LED on the drone blinks GREEN. The turn lock button can be used to lock the left controller for just up and down or up, down, left turn and right turn. Use the left stick to take off or land the drone. Use the right stick to control the movements. If the drone disconnects from the app or the drone itself is rebooting when trying to take off, that means the battery can't provide enough power. We have used a 1300mAh 30C battery. So please ensure to use a battery with a higher discharge rating.

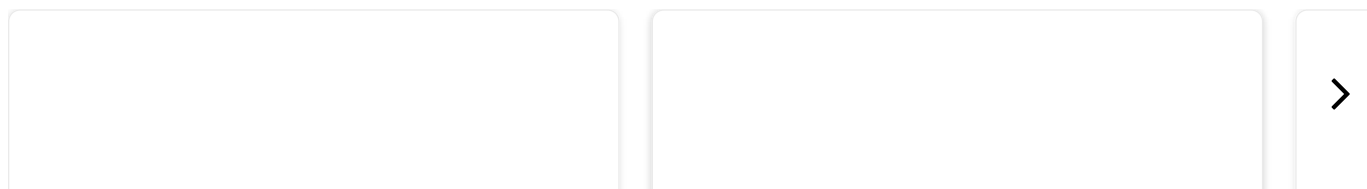
## Preflight Check

- Place the drone with its head on the front, and its tail (i.e., the antenna part) at the back.
- Place the drone on a level surface and power it up when the drone stays still.
- After the communication is established, check if the LED at the drone tail blinks GREEN fast.
- A blinking RED LED indicates battery LOW, Charge the battery if it happens.
- Slide forward the Trust controller slightly to check if the drone can respond to the command.
- Use the right controller to check if the direction control works well.

You can find more detailed explanations and other details on the project page [Let's Build a Low-Cost Drone using ESP32](#).

## Key Parts and Components

10 Items



<b>Mfr Part # ESP32-WROOM-32E-H4</b> RF TXRX MOD BT WIFI TH SMD Espressif Systems \$2.68 <a href="#">View More</a> <a href="#">Details</a>	<b>Mfr Part # MPU-6050</b> IMU ACCEL/GYRO 3-AXIS I2C 24QFN TDK InvenSense \$4.80 <a href="#">View More</a> <a href="#">Details</a>	<b>Mfr</b> IC U Silic \$5.9
--	--	--------------------------------------

[Add all DigiKey Parts to Cart](#)

## TechForum

Have questions or comments? Continue the conversation on [TechForum](#), DigiKey's online community and technical resource.

**Visit TechForum**

[Arduino](#) | [3D Printing](#) | [Raspberry Pi](#)

### More videos by Aswinth Raj

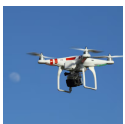


**Donald Bell Maker Update: The Ceiling's the Limit**

Maker.io Staff

1 year ago

### Related Content



**Drones: Weight is Everything**

By Maker.io Staff



**Teardown Syma X5C Drone**


By All About Circuits




### Hacksmith Industries: Security Drone


By The Hacksmith


[View All Projects and Tutorials](#)


 Chat

 1-800-344-4539

218-681-6674

 [sales@digikey.com](mailto:sales@digikey.com)

 Co-Browse

 United States | Copyright © 1995-2024, DigiKey. | All Rights Reserved.

Local Support: 701 Brooks Avenue South, Thief River Falls, MN 56701 USA

Do Not Sell / Do Not Share My Personal Information