

Making An ESP32 Microcontroller Board

The ESP32 series of microcontrollers from Espressif have become very popular among hobbyists and in the astronomy sphere. This page goes over the basics of making a custom PCB that uses an ESP32. You can apply the information here to your projects, whether it might be a custom OnStep controller or an electronic autofocuser.

Selecting a chip

[Sirawit Moonrinta](#) made a [comparison chart](#) on GitHub that can help you select an ESP32, but it boils down to the following:

	ESP32	ESP32-S2	ESP32-S3	ESP32-C3	ESP32-C6
Processor	240 MHz single or dual core	240 MHz single core	240 MHz dual core	160 MHz single core	160 MHz single core
WiFi	WiFi 4	WiFi 4	WiFi 4	WiFi 4	WiFi 6
Bluetooth	BLE 4.2	x	BLE 5	BLE 5	BLE 5
USB	x	✓	✓	✓	✓
GPIOs	34	43	44	22	22

For a lot of applications you don't need a whole lot and you want to simplify the design, in that case the C3 or C6 might be a good option. If you need the speed or a lot of GPIOs, then an S2 or S3 would be a good option. The original ESP32 is currently the only tested option for OnStep. Issues with USB communication in the OnStep driver cause the ESP32s with built-in USB connectivity to not work correctly. They may work with a USB-to-serial bridge, but that has not been tested yet.

The different packages



ESP32 can be bought in many different packages. What that means is that they can be bought as a chip or as a module with easy to solder castellated holes. In this case using the chip itself doesn't have many advantages, so we'll only be looking at using the modules.

The following types of modules are available:

- WROOM: This is the most common type of module package you will see on microcontroller board, has castellated holes that are easy to solder.
- WROVER: Available for some ESP32s, these are similar but larger than WROOM modules and break out a bit more functionality.
- PICO/MINI: These have a smaller footprint and cannot be soldered with a soldering iron because they have the pads on the bottom. So you will either need a reflow plate, reflow oven or have to be assembled by a PCB house.

Lastly, many of these modules can be bought with either a PCB antenna or a uFL connector. The modules with a uFL connector are usually smaller, but otherwise have the same footprint and pinout as the PCB antenna versions.

The other components we need

The ESP32 doesn't work by itself, we need some extra passive components and chips to make it all work.

- Decoupling/buffer capacitors: These act as little buckets of energy that can supply the chip when there is a fast change in current that the power supply can't supply through long wires. These capacitors also provide a low impedance path for high frequency noise to go to GND.
- LDO Voltage regulator: In a lot of cases we want to use 5V, but the ESP32 can only accept 3.3V, so a voltage regulator is needed. The simplest option is going to be a Low DropOut regulator. These voltage regulators are not ideal for stepping down from higher voltages (>9V) because of how inefficient they get at those larger voltage differences.
- USB-to-serial bridge: If we are using an normal ESP32, a USB-to-serial bridge is needed to program and talk to the microcontroller. Since it doesn't have native USB, the bridge will need to convert to UART.
- Reset & Boot button: to simplify programming and debugging, a button to reset the chip (restart, not clear the memory) and a button to get the chip into the firmware download mode might be a good idea.

Updated 20 January 2025 13:23:24 by Admin