

Miscellaneous DIY Information

- [Connectors & Cables](#)
- [Discord Servers](#)
- [Making An ESP32 Microcontroller Board](#)

Connectors & Cables

Doing things right is hard, especially when it comes to connectors. That is why this summary was created to help out.

Power connectors

- Barrel jacks are extremely common because they are simple, can handle high currents and can survive many mating cycles. Downside is that they are easy to accidentally disconnect and the thickness of wire you can get into the connector is limited. Furthermore, if you solder these connectors yourself then they will have very little strain relief, so unplugging it by pulling on the cable is an extremely bad idea.
- GX12/16-2 connectors are also very common and come with the advantage that they can be screwed on to prevent accidental unplugging. These connectors can also handle high currents like barrel jacks. Their main downside is that they can be picky when it comes to cable thickness and their strain relief isn't very good.
- XT-series connectors are more common in the world of batteries in RC toys, but since they are easy to solder and can handle extremely high currents they also have a place in the world of astronomy. Chassis and PCB mounted connectors are harder to find than the previously mentioned connectors, but they do exist. They can be difficult to disconnect when grabbing the connector itself, so make sure you have good solder connectors if you do have to pull on the cable.
- (mini)XLR connectors can't handle as much current as the other connectors, so in this case they are more of an honorable mention. They are easy to solder, generally offer good strain relief and are easy to get for cheap.
- The least common would probably be speakON connectors. Like XLR these are mainly used in the world of audio, but they can handle high currents, are easy to solder and also offer good strain relief. They are a bit more expensive than XLR.

It is recommended to standardise the connectors you use on your rig. If you ever forget a cable, you might have a spare one.

Always use the correct thickness wire for the amount of current you expect to pass through. Take a safety margin of at least 25%. Use heat shrink where possible to prevent short circuit connection if something comes loose.

Mini-XLR connectors have a limited current rating, they are not recommended for power or motor connectors.

Motor connectors

- GX-series connectors are the most popular option. As mentioned before they can be screwed on to prevent accidental unplugging, can handle high currents but come with the downside that their strain relief isn't very good.
- XLR connectors are also a good option, but 4 pole connectors aren't very common.

RJ45/RJ12 connectors aren't recommended because of their low current rating and inability to handle many mating cycles.

USB connectors

Use USB-A and B as much as possible and use off the shelf cables. This is the most robust option. USB-C can be used, but it is harder to implement in some applications and aren't as robust as A and B. USB-C is also hard to plug in when it is dark.

Discord Servers

There are a number of Discord servers where people are happy with answer DIY related questions.

General Astro-related DIY:

- AstroBiscuit: <https://discord.gg/XD2HvWPYH6>

Telescope making:

- Observational Astronomy: <https://discord.gg/YDveQ2W6mQ>

OpenAstroTech hardware:

- OpenAstroTech: <https://discord.gg/rKUDhCDJ4S>

General DIY:

- Hackaday: <https://discord.gg/anrPbwhq>
- Adafruit: <https://discord.gg/mUNhbgJmjY>

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.