ESP32

What is it

A ESP32 is a small IoT Chip often located on a development board to easily make make a wifi/radio/bluetooth IoT Project.

Setting up ESP32

This was made using the docs in start of 2025. Small changes my apply per board

Prereqs

```
sudo apt-get install git wget flex bison gperf python3 python3-pip python3-venv cmake ninja-build ccache libffi-dev
libssl-dev dfu-util libusb-1.0-0
```

ESP-IDF

> **Note**
>
> If using fish shell consult THIS

```
mkdir -p ~/esp
cd ~/esp
git clone -b v5.1.2 --recursive https://github.com/espressif/esp-idf.git
```

```
cd ~/esp/esp-idf
./install.sh esp32
```

```
cd ~/esp/esp-idf
./install.sh all
```

```
cd ~/esp/esp-idf
export IDF_GITHUB_ASSETS="dl.espressif.com/github_assets"
./install.sh
```

```
. $HOME/esp/esp-idf/export.sh
```

Add this to your .zshrc or .bashrc file

```
alias get_idf='. $HOME/esp/esp-idf/export.sh'
```

Hello world

```
cd ~/esp
cp -r $IDF_PATH/examples/get-started/hello_world .
```

Getting device

USB devices usually is /dev/ttyUSB*

Listing Devices

> To get the devices plugged in use:
>
> ```
> lsusb
> ```
>
> if you do not know the name of the device you can unplug and replug the usb device and use
>
> ```
> dmesg
> or
> sudo dmesg
> ```
>
> to get the port of the newest plugged in device

When device port is found, in my case /dev/ttyUSB0.
Configuring the project to use esp32

```
cd ~/esp/hello_world
idf.py set-target esp32
idf.py menuconfig
```

if the project does not know what idf.py is run

```
get_idf
```

now you should be able build

```
idf.py build
```

and then upload

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

it says it should be able to find the port automatically, never have it done that for me.

```
idf.py -p /dev/ttyUSB0 flash
```

> ⚠ **Caution**
>
> If sure that the port is correct, but the upload still fails
>
> > ## Permission Issues
> >
> > ```
> > sudo usermod -a -G dialout $USER
> > ```
> >
> > then reboot/logout

To see the result use the monitor command

```
idf.py -p /dev/ttyUSB0 monitor
```

you can flash and monitor at the same time

```
idf.py -p /dev/ttyUSB0 flash monitor
```