Installation

To use MicroPython with a real ESP32 board, you will need to follow these steps:

- Download MicroPython firmware
- Flash the firmware
- Connect to the Board's Serial REPL and interact with MicroPython
- Transfer files to the ESP32 board

There are several very good tutorials how to install and use MicroPython on an ESP microcontroller, such as this one for Windows. The following text was tested under Linux-based operating system.

1. Install Python.

2. Open terminal (typically `Ctrl+Alt+T`) and install `esptool`:

   ```
   pip install esptool
   ```

   Connect your ESP board and test the `esptool`:

   ```
   # Get the version
   esptool.py version

   # Read chip info, serial port, MAC address, and others
   # Note: Use `dmesg` command to find your USB port
   esptool.py --port /dev/ttyUSB0 flash_id

   # Read all eFuses from the chip
   espefuse.py --port /dev/ttyUSB0 summary
   ```

ESP32

3. Download the latest firmware for your target device, such as `esp32-20230426-v1.20.0.bin` for Espressif ESP32.

4. Erase the Flash of target device (use your port name):

   ```
   esptool.py --chip esp32 --port /dev/ttyUSB0 erase_flash
   ```

5. Deploy the new firmware:

   ```
   esptool.py --chip esp32 --port /dev/ttyUSB0 write_flash -z 0x1000
   esp32-20230426-v1.20.0.bin
   ```

ESP8266

3. Download the latest firmware, such as `esp8266-20230426-v1.20.0.bin`.

4. Erase the Flash before deploying the firmware:

   ```
   esptool.py --chip esp8266 --port /dev/ttyUSB0 erase_flash
   ```

5. Deploy the firmware:

   ```
   esptool.py --chip esp8266 --port /dev/ttyUSB0 write_flash --flash_mode
   dio --flash_size 4MB 0x0 esp8266-20230426-v1.20.0.bin
   ```

Usage

Test MicroPython via PuTTY or directly in terminal by `screen`. You need to press on-board reset button:

```
screen /dev/ttyUSB0 115200
```

> **Note:** To exit the `screen`, press `Ctrl+A`, followed by `K` and `Y`.

```
# Print string to a Shell
>>> print("Hi there!")
Hi there!

# Operators used for the different functions like division,
# multiplication, addition, subtraction, ...
>>> 10/3
3.333333
>>> 10//3
3
>>> 10%3
1
>>> 10*3
30
>>> 10**3
1000

# Integers, floats, strings
>>> type(10)
<class 'int'>
>>> type(10.0)
<class 'float'>
```

```
>>> pi = 3.1415
>>> pi_str = str(pi)
>>> type(pi_str)
<class 'str'>
>>> len(pi_str)
6

# `ord` returns unicode code of a specified character
>>> ord("A")
65
>>> ord("a")
97
>>> ord("0")
48

>>> print(pi_str)
3.1415
>>> ord(pi_str[0])
51
>>> ord(pi_str[-1])
53
```

See MicroPython tutorials, such as MicroPython Programming Basics with ESP32 and ESP8266 for detailed explanation.

Test some other useful commands from Quick reference for the ESP32:

```
# A platform identifier
>>> import sys
>>> sys.platform
'esp32'

# Get the current frequency of the CPU and RTC time
>>> import machine
>>> help(machine)
>>> machine.freq()
>>> machine.RTC().datetime()

# Get Flash size in Bytes
>>> import esp
>>> esp.flash_size()

# Read the internal temperature (in Fahrenheit)
>>> import esp32
>>> esp32.raw_temperature()
# FYI: temp_c = (temp_f-32) * (5/9)
#      temp_f = temp_c * (9/5) + 32
```

Useful information

- ESP32 brief overview (YouTube video)
- FireBeetle board

References

1. Peter Kazarinoff. How to install MicroPython on an ESP32 microcontroller
2. Getting started with MicroPython on the ESP32
3. Rafael Aroca. ESP32, Camera, MicroPython and NO esptool!

Tested on operating systems

Version	Result (yyyy-mm-dd)	Note
Windows 10	OK (2023-09-18)	Lab SC 6.61
Linux Mint 20.3 (Una)	OK (2022-05-23)	Laptop

```
# FYI: How to check OS version in Linux
$ cat /etc/os-release

# Or by Neofetch
$ neofetch
```