

Name	SID	Checkoff

Objectives: Flash the MicroPython firmware to the ESP32 and setup the programming environment. We will also solder headers to the Huzzah32 board.

Note: We have tested these instructions on the computers in the EE49 laboratory. However, are quite general. If you own computer (e.g. PC, Mac) you can do so using these same instructions. Please understand that while we will try to support a range of setups, we cannot diagnose problems specific to your computer.

Parts and Tools

For this laboratory, you will need the IoT controller, a **solder-less prototyping board**, and a soldering station (Figure 1).

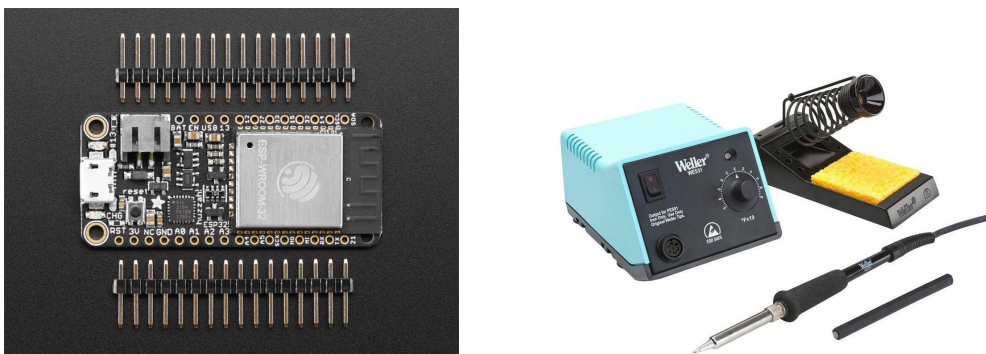


Figure 1 Huzzah32 IoT Controller & Soldering station

Prelab

Read

- This guide
- Soldering guide at <https://learn.adafruit.com/adafruit-guide-excellent-soldering>
- How to use a breadboard, <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>
- Instructions at <https://github.com/bboser/IOt49/blob/master/doc/install.md>

You also need to be familiar with Python programming at the level of Homework 1.

Lab

Command Prompt

For this (and subsequent) labs you will need to access the “command prompt” of the computer you are using. This is a textual interface to start programs. Instructions for getting a command prompt can be found on the web. E.g. for windows, <https://www.wikihow.com/Open-the-Command-Prompt-in-Windows>, or Mac, <https://www.wikihow.com/Get-to-the-Command-Line-on-a-Mac>.

Programming Environment

First we setup the MicroPython programming environment for the Huzzah32. Remove the Huzzah32 board from its packaging and connect it to the host computer using a micro-USB cable. Be careful not to touch any conductive surfaces (e.g. metals) with the Huzzah32 boards to prevent electrical damage to the board. After the board receives power through the USB connection, a yellow LED will start flashing rapidly. This is normal.

Follow the instructions at <https://github.com/bboser/IoT49/blob/master/doc/install.md> to flash the MicroPython firmware to the Huzzah32. You can do this either on the lab computer or on your own hardware.

Hello World—The IoT Way!

Create a text file `blink.py` with the following content:

```
from board import LED
from machine import Pin
from time import sleep

led = Pin(LED, mode=Pin.OUT)

while True:
    led(1)
    sleep(1)
    led(0)
    sleep(1)
```

In the folder with the file, start `shell49`. At the command prompt, type `ls` and verify that the file is present. You can use the `cd` command to change to a different directory. Then at the `shell49` prompt type

```
run blink.py
```

Type `Ctrl-C` to abort the program.

Show the blinking light to the instructor.

Checkoff:

Solder Headers to the Huzzah32 Board

Now that the Huzzah32 is working as expected, we are going to solder headers to the board. These will later be used to connect sensors and actuators to the controller.

Follow the instructions in section “Soldering in Plain Headers” of the guide at <https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/assembly#soldering-in-plain-headers>. Before soldering, watch the demo by the instructor. **Proceed carefully as to not damage the board and do not hesitate to ask questions.** Beware: do not touch the hot end of the soldering iron!

From now on use the board inserted in a protoboard to protect it from accidental electrical short circuits.

WiFi

To use the Huzzah32 on the EECS wireless network, it must be registered. With the device connected, type

```
mac
```

The output will look similar to this:

```
'3a:bf:a4:31:82:1b'
```

The hexadecimal number between the quotes is the devices' "MAC" address (**media access control address**).

Submit the hostname you chose for your ESP32 and the mac address on bcourses, assignment "WiFi Connection". Staff will use this information to enable WiFi service for the EECS-PSK network. Be careful to accurately copy the mac address, otherwise your device will not be able to connect to the EECS WiFi!

SOS

Modify the blink-program to send out an **SOS distress message**.

Checkoff: