Using Edger without a browser

- The current stable release of Edger offers built in support for accessing I2C peripherals with web API operations. (GPIO is supported but not discussed here)
- On Linux either shell+curl or Python scripts can be used to access I2C peripherals connected to the ESP32 of an Edger system using HTTP operations.
- The Linux system must be on same subnet as the Edger system.

9/20/2023 r1

Edger Ant Web API

- The ESP32 plus Edger firmware combination called “Ant” offers a restful web server.
- A Python script with the requests module doing an HTTP patch to an Ant URL for the first Ant instance starts like this:

  https:esp-home.local/api/v1/i2c/…
- The explicit IP address of the instance shown in IDF monitor output can be used instead of esp-home.local.

Patch parameters

- The rest of the URL contains parameters that specify details:
  - The I2C device address
  - The offset in the I2C register space to use
  - Read or write is requested (actually and/or, but we’re keeping thing simple to start)
  - How much data to transfer

A Python HTTP request to read data via I2C

```
try:
    response = requests.patch(
        ENDPOINT_URL + '?get={"address":' +
            str(i2caddr) + ',"index":' +
            str(index) + ',"length":' + str(length) + '}',
        headers=headers, data=DATA,
    )
```

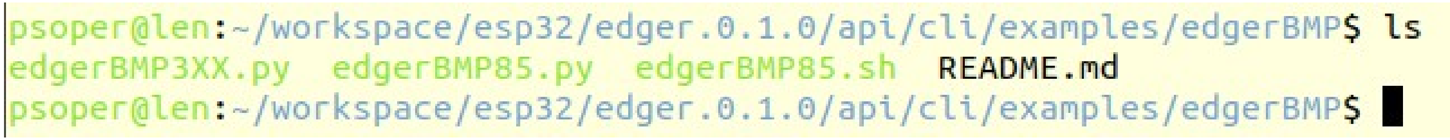
Patch request details

- ENDPOINT_URL is a pseudo-const string with the first part of the endpoint URL as already described
- I2caddr, index and length are integer variables specifying the peripheral I2C address, the device register address and the length (number of bytes) of data to get
- The headers parm specifies text/plain content type

Extracting the returned data as a function value

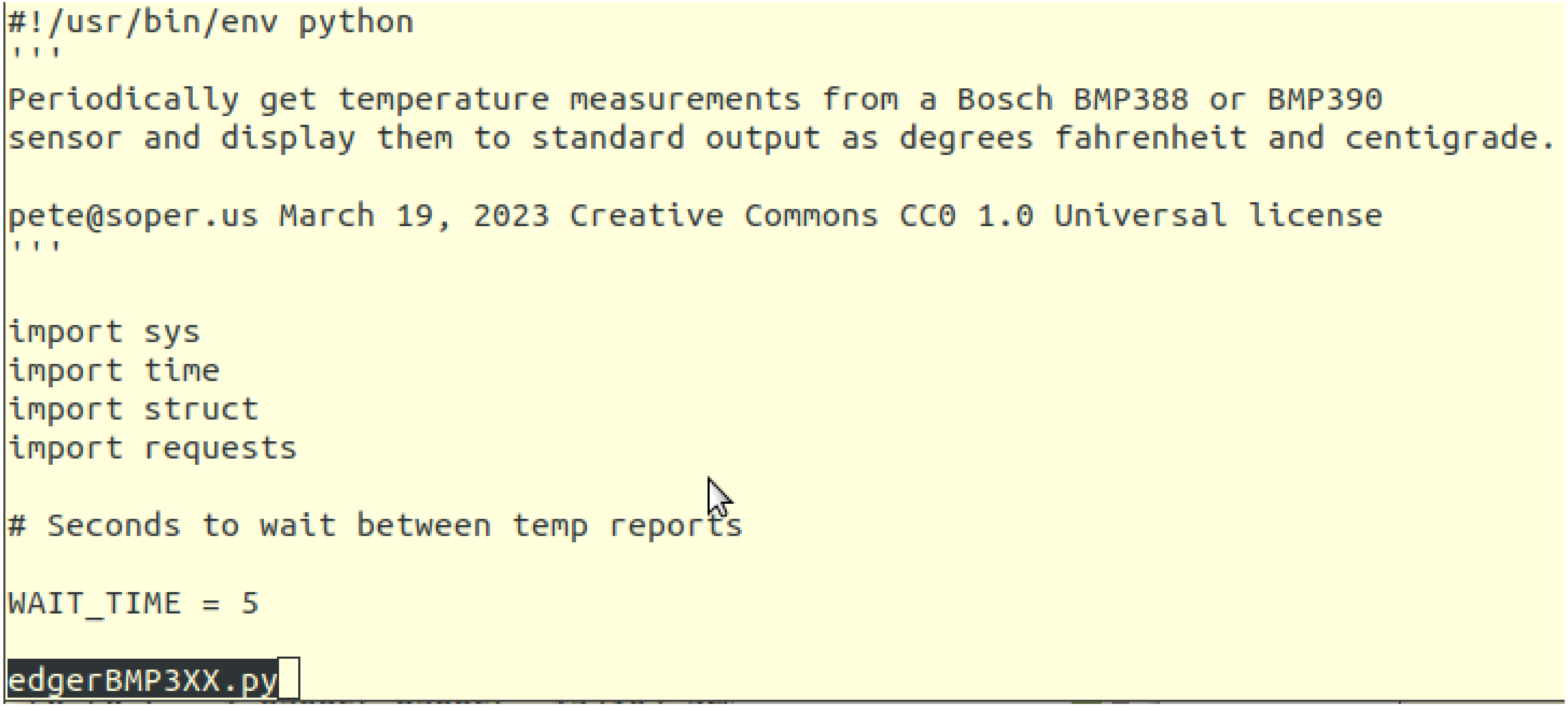
```
response_json = response.json()

return response_json['i2c'][0]['get']
```

Edger
script
examples



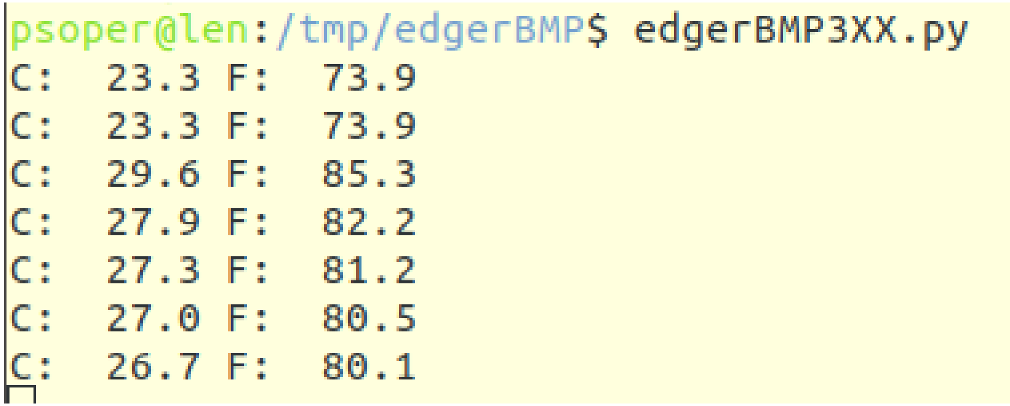
```
psoper@len:~/workspace/esp32/edger.0.1.0/api/cli/examples/edgerBMP$ ls
edgerBMP3XX.py  edgerBMP85.py  edgerBMP85.sh  README.md
psoper@len:~/workspace/esp32/edger.0.1.0/api/cli/examples/edgerBMP$ 
```

Python script start



```
#!/usr/bin/env python
'''
Periodically get temperature measurements from a Bosch BMP388 or BMP390
sensor and display them to standard output as degrees fahrenheit and centigrade.

pete@soper.us March 19, 2023 Creative Commons CC0 1.0 Universal license
'''


import sys
import time
import struct
import requests

# Seconds to wait between temp reports

WAIT_TIME = 5

edgerBMP3XX.py
```

Script output (finger pressed on sensor and released)



```
psoper@len:/tmp/edgerBMP$ edgerBMP3XX.py
C:  23.3 F:  73.9
C:  23.3 F:  73.9
C:  29.6 F:  85.3
C:  27.9 F:  82.2
C:  27.3 F:  81.2
C:  27.0 F:  80.5
C:  26.7 F:  80.1
```