Micropython & ESP8266 & (u)asyncio

serpulga [GitHub | Twitter | Gmail]

Supported boards

- PyBoard
- ESPxx
- WyPy
- STM32F4
- NUCLEO
- Espruino Pico

Micropython

Python Medellín

ESP8266

- WiFi
- 16 GPIO pins
- SPI
- I²C (software)
- I²S (shared pins)
- UART (dedicated pins)
- TX UART (GPIO2)
- 10-bit ADC

Micropython

Python Medellín

Flashing binaries

http://micropython.org/download

```
$ pip install esptool

$ esptool.py --port /dev/ttyUSB0 erase_flash

$ esptool.py --port /dev/ttyUSB0                    \
             --baud 115200                          \
             write_flash --flash_size=detect 0      \
             esp8266-xxxxxxxx-vx.x.x.bin
```

Micropython

Python Medellín

Interactive console [REPL]

```
$ picocom /dev/ttyUSB0 -b115200

>>> import sys
>>> sys.platform
'esp8266'
```

Micropython

Python Medellín

Uploading scripts

```
$ ampy -p /dev/ttyUSB0 -b 115200 put main.py
```

The file **main.py** will be run automatically after booting.

Micropython

Python Medellín

Semaphore mini-Project

A "smart" pedestrian crossing semaphore which prevents unnecessary stops by allowing automoviles pass indefinitely until there is an actual cross request.

https://github.com/serpulga/esphore

Micropython

Python Medellín

Why asyncio?

- Microcontrollers are single-core
- The event loop
- Event-driven is a perfect match

```
loop = asyncio.get_event_loop()
loop.create_task(...)
loop.run_forever()
```

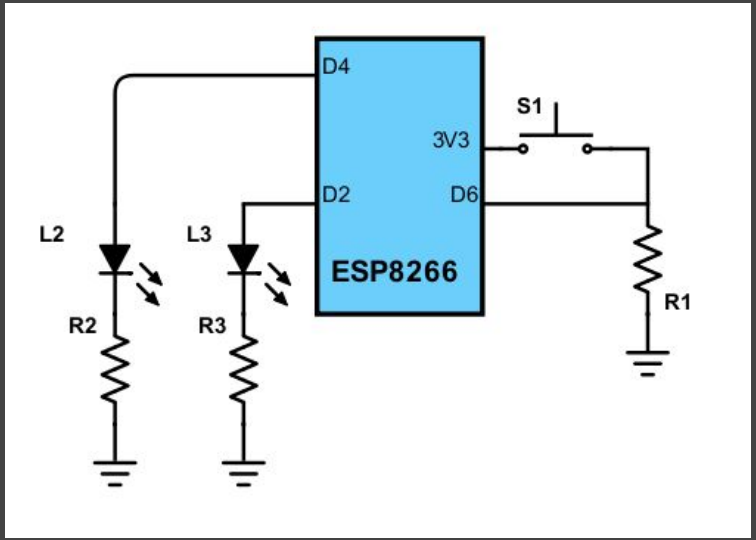
Micropython

Python Medellín

Semaphore mini-Project

Preparing hardware

R1, R2, R3 ?



Micropython

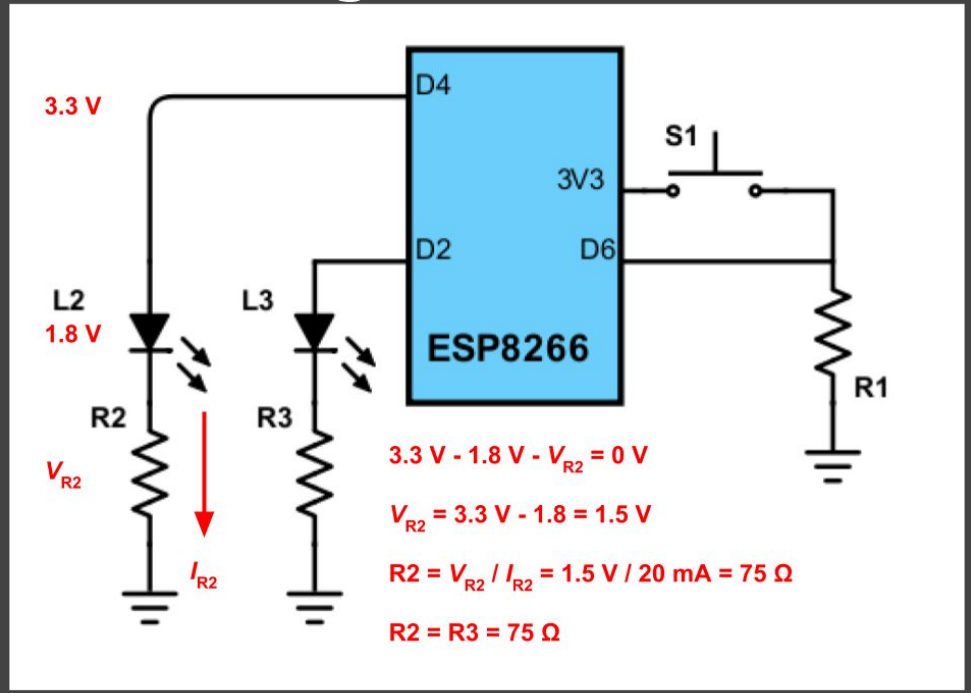
Python Medellín

Semaphore mini-Project

Datasheet:

I_{R2} = 20 mA

V_{L2} = 1.8 V



Micropython

Python Medellín

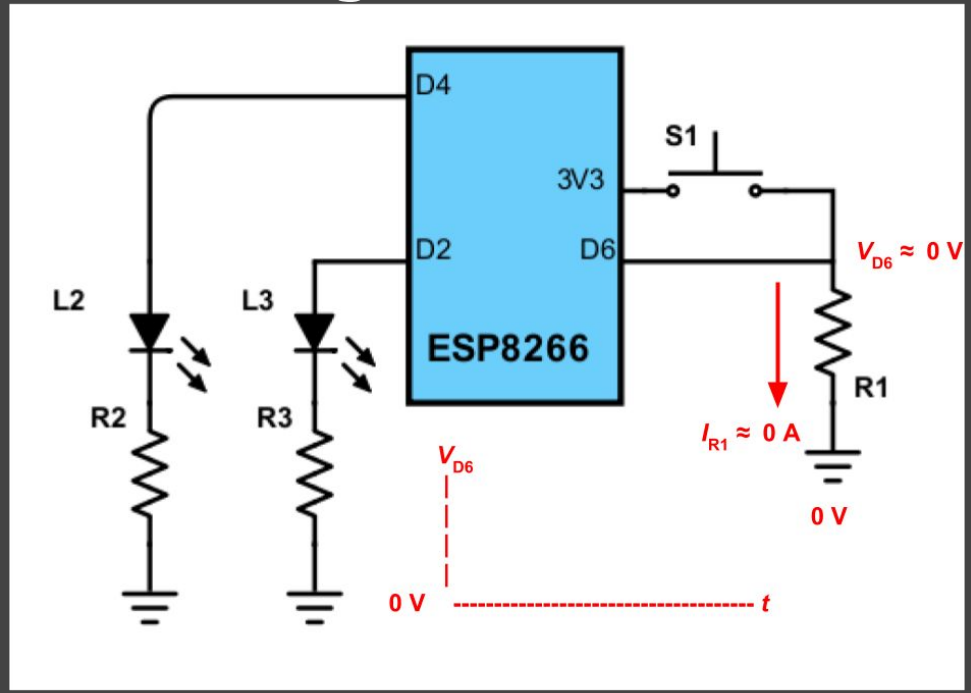
Semaphore mini-Project

Normal operation:

$I_{R1} \approx 0\ A$

$V_{R1} \approx 0\ V$ (0.018 V)

D6 => LOW



Micropython

Python Medellín

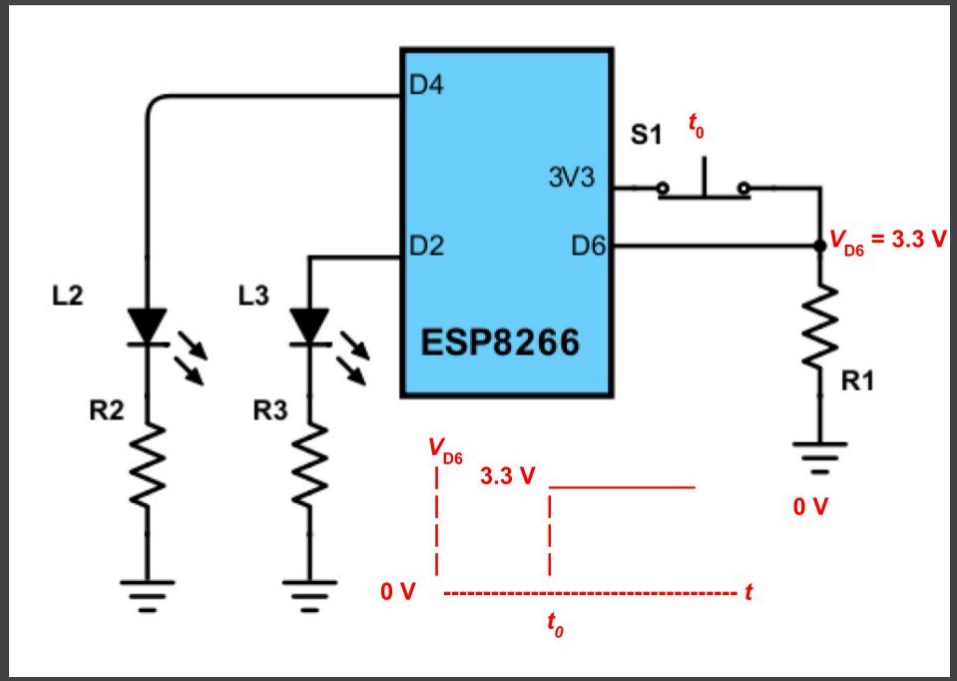
Semaphore mini-Project

Pressed button:

R1 = 47 KΩ

$I_{R1} = V_{D6} / R1$

$I_{R1} = 70\ \mu A$



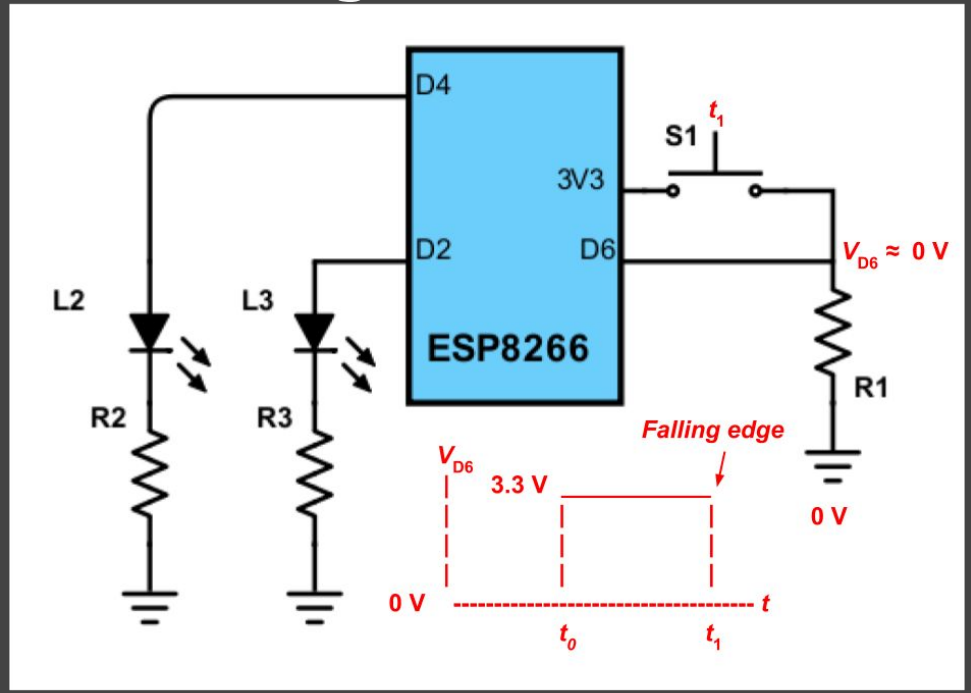
Micropython

Python Medellín

Semaphore mini-Project

Released button:

Falling edge is detected By PIN



Micropython

Python Medellín

Semaphore mini-Project

IRQ [Interrupt Request] is triggered by the pin

```
def callback(pin):
    ...

cross_pin = Pin(12, Pin.IN)
cross_pin.irq(trigger=Pin.IRQ_FALLING,
              handler=callback)
```

Micropython

Python Medellín