Day 01-3

Assignment:

- Homework 01, Due Thursday
-

Today's Topics:

- Python GPIO
- /sys/

01-3 – Blink an LED the Easy Way

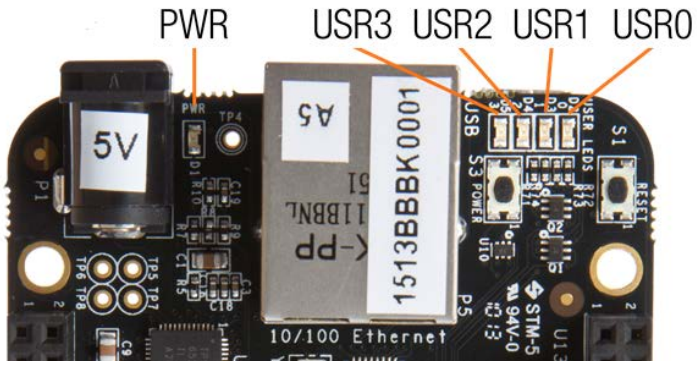
Much of this is from

BeagleBone Cookbook

Adafruit BBIO Python Library

- https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/overview

Blink an LED




```
#!/usr/bin/env python3
import Adafruit_BBIO.GPIO as GPIO
import time
LED = "USR0"
delay = 0.25
GPIO.setup(LED, GPIO.OUT)

while True:
    GPIO.output(LED, 1)
    time.sleep(delay)
    GPIO.output(LED, 0)
    time.sleep(delay
```

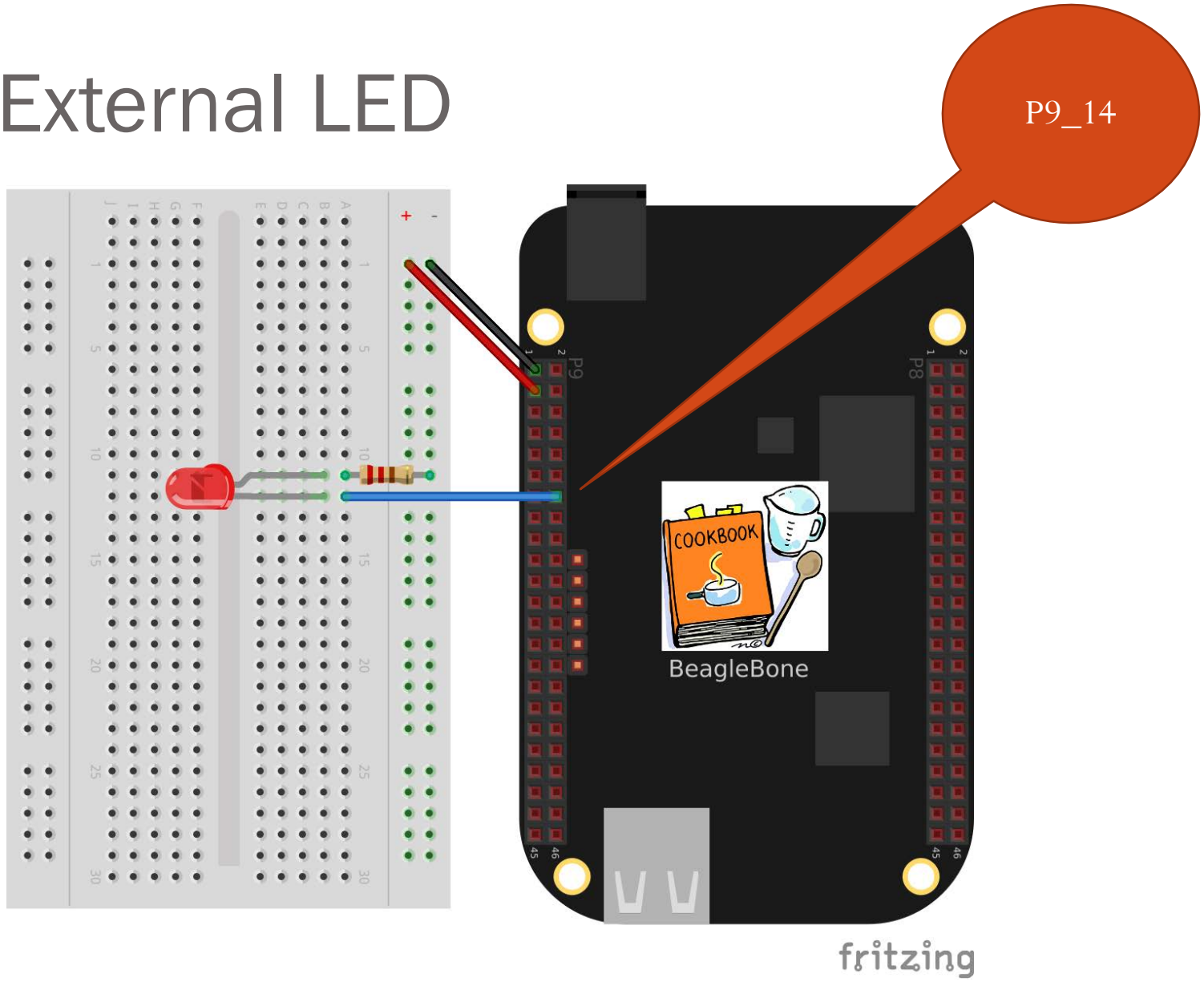
exercises/displays/blue/blink1led.py

Running python

- Use Cloud9 debugger
- From command line
- If the first line is: **`#!/usr/bin/env python3`**

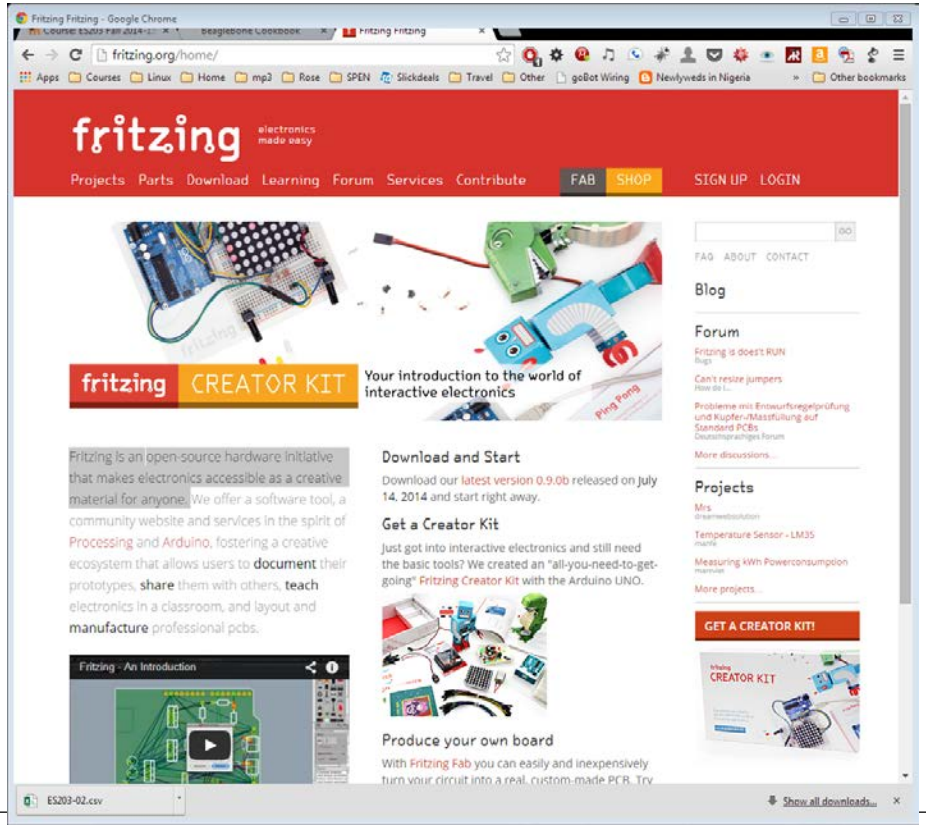
```
bone$ chmod +x ./blink1led.py
bone$ ./blink1led.py
```

External LED



Fritzing

- http://fritzing.org/home/
- Fritzing is an *open-source hardware initiative* that makes electronics accessible as a creative material for anyone.



Blink an LED

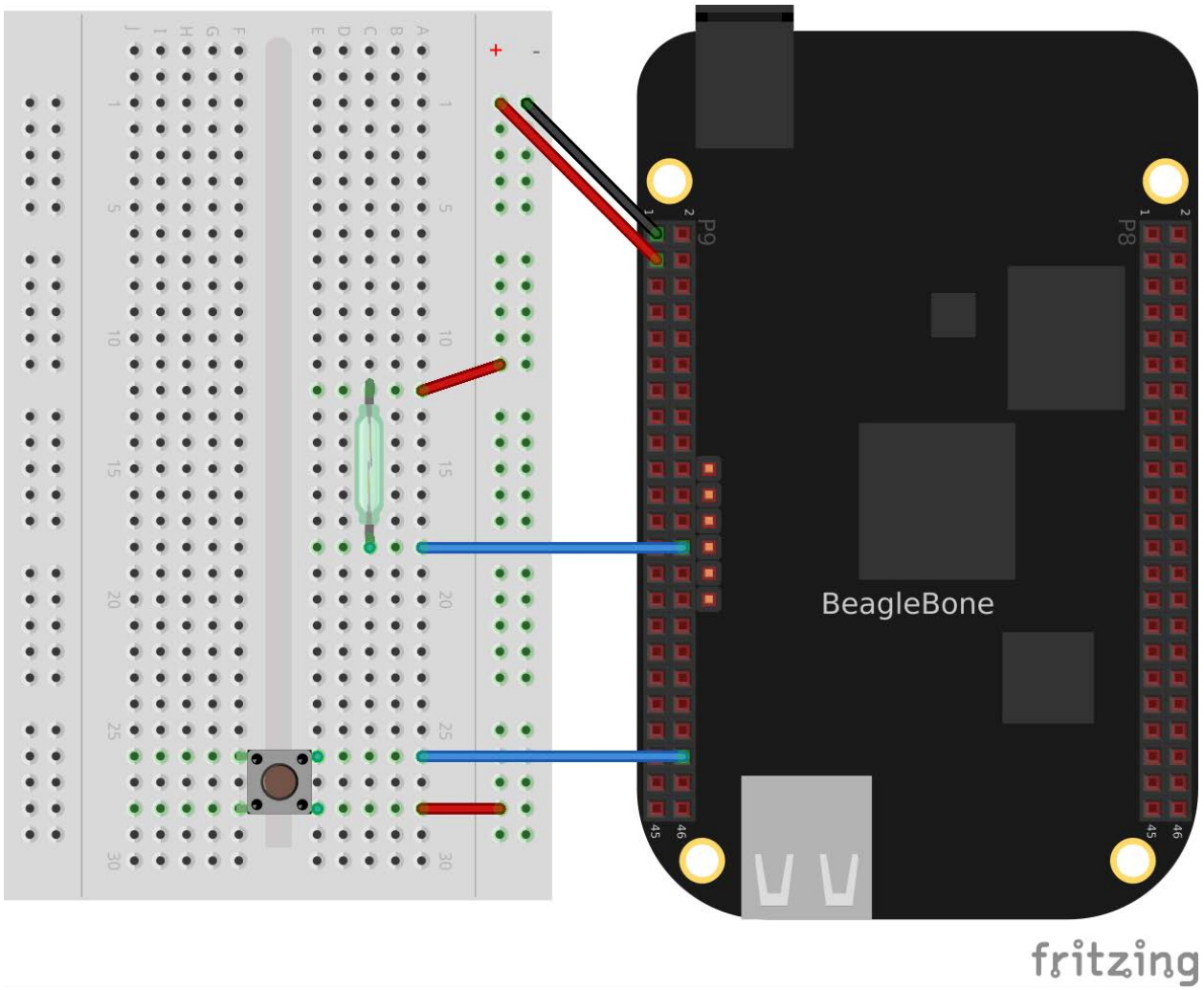
External

```
#!/usr/bin/env python3
import Adafruit_BBIO.GPIO as GPIO
import time

LED = “GP9_14"
delay = 0.25

GPIO.setup(LED, GPIO.OUT)

while True:
    GPIO.output(LED, 1)
    time.sleep(delay)
    GPIO.output(LED, 0)
    time.sleep(delay)
```

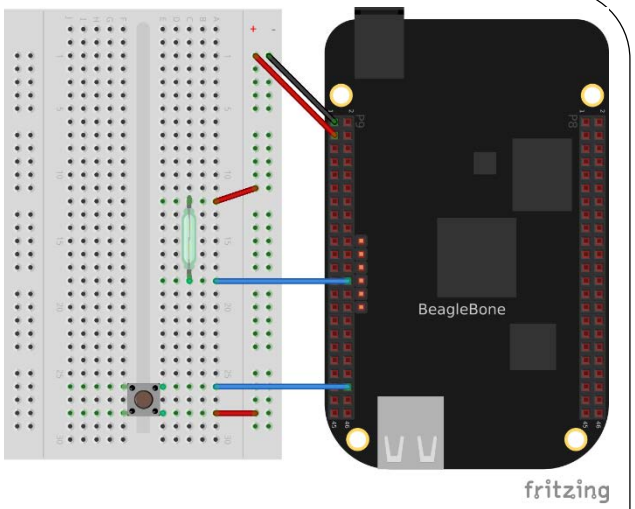
Internal

```
#!/usr/bin/env python3
import Adafruit_BBIO.GPIO as GPIO
import time

LED = "USR0"
delay = 0.25

GPIO.setup(LED, GPIO.OUT)

while True:
    GPIO.output(LED, 1)
    time.sleep(delay)
    GPIO.output(LED, 0)
    time.sleep(delay)
```

Read a button



Button




```
#!/usr/bin/env python3
import Adafruit_BBIO.GPIO as GPIO
import time
button="P9_42"
LED   ="USR3"
# Set the GPIO pins:
GPIO.setup(LED,    GPIO.OUT)
GPIO.setup(button, GPIO.IN)

while True:
  state = GPIO.input(button)
  GPIO.output(LED, state)

  GPIO.wait_for_edge(button, GPIO.BOTH)
  print("Pressed")
```

Careful when pasting code

exercises/displays/blue/button.py

Button - Events

```
#!/usr/bin/env python3
import Adafruit_BBIO.GPIO as GPIO
import time

buttonP="P9_26"  # PAUSE or MODE
buttonM="P9_42"

LEDp   ="USR0"
LEDm   ="USR1"

# Set the GPIO pins:
GPIO.setup(LEDp,    GPIO.OUT)
GPIO.setup(LEDm,    GPIO.OUT)
GPIO.setup(buttonP, GPIO.IN)
GPIO.setup(buttonM, GPIO.IN)

# Turn on both LEDs
GPIO.output(LEDp, 1)
GPIO.output(LEDm, 1)
```

```
# Map buttons to LEDs
map = {buttonP: LEDp, buttonM: LEDm}

def updateLED(channel):
    print("channel = " + channel)
    state = GPIO.input(channel)
    GPIO.output(map[channel], state)
    print(map[channel] + " Toggled")

print("Running...")

GPIO.add_event_detect(buttonP, GPIO.BOTH, callback=updateLED)
# RISING, FALLING or BOTH
GPIO.add_event_detect(buttonM, GPIO.BOTH, callback=updateLED)

try:
    while True:
        time.sleep(100)   # Let other processes run

except KeyboardInterrupt:
    print("Cleaning Up")
    GPIO.cleanup()
GPIO.cleanup()
```

exercises/displays/blue/buttonEvent.py

65 possible digital I/Os

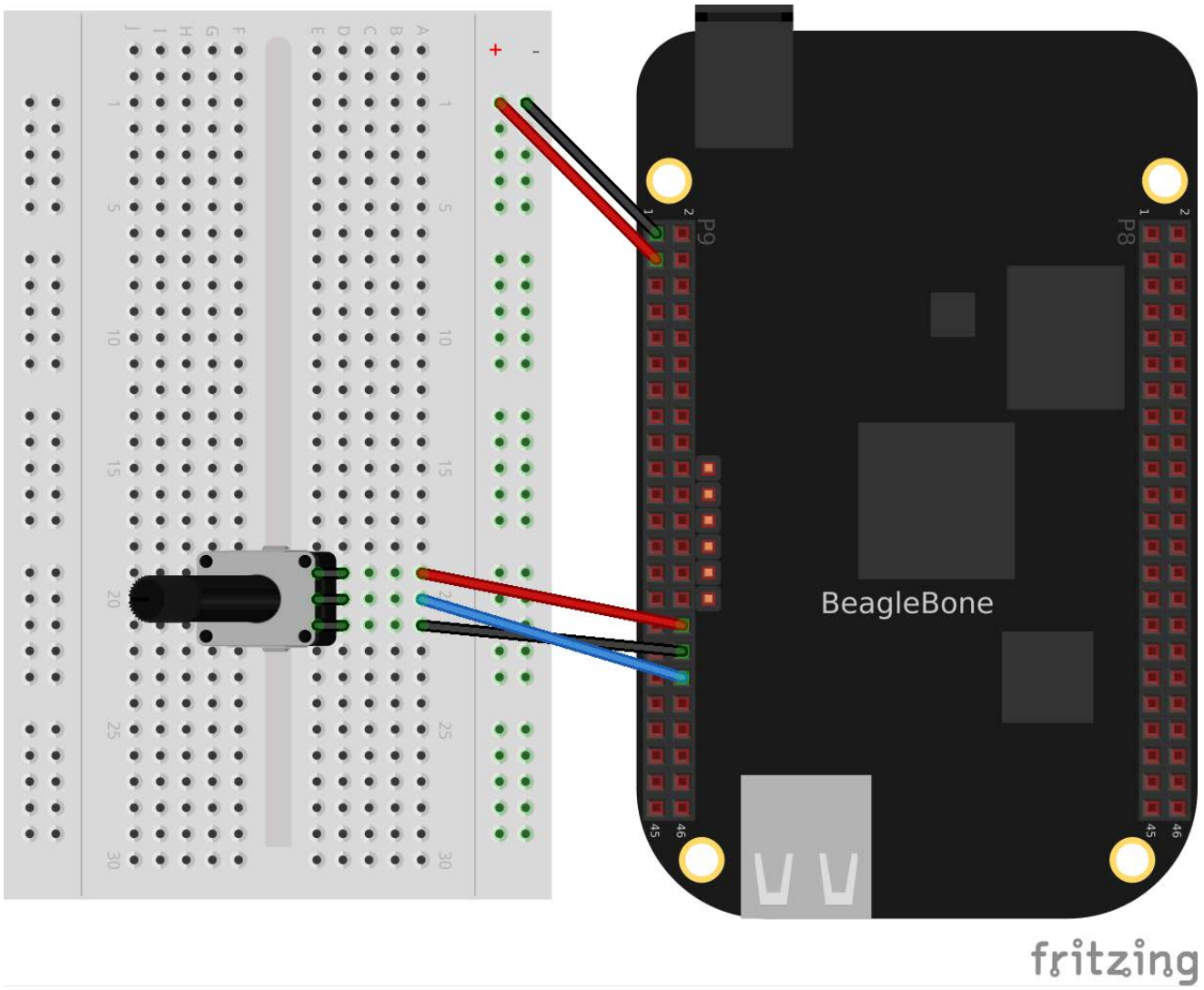
P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

Analog in



Analog Code

```
#!/usr/bin/env python3
# From: https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/adc
import Adafruit_BBIO.ADC as ADC
ADC.setup()
value = ADC.read("P9_40")
voltage = value * 1.8 #1.8V
```

exercises/displays/blue/analog.py

7 analog inputs (1.8V)

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETn
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_123
GPIO_121	29	30	GPIO_122
GPIO_120	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

Pulse Width Modulation (PWM)

```
#!/usr/bin/env python3
# From: https://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/pwm
import Adafruit_BBIO.PWM as PWM
#PWM.start(channel, duty, freq=2000, polarity=0)
PWM.start("P9_14", 50)

#optionally, you can set the frequency as well as the polarity from their defaults:
PWM.start("P9_14", 50, 1000, 1)

PWM.set_duty_cycle("P9_14", 50)
PWM.set_frequency("P9_14", 100)
```

exercises/displays/blue/pwm.py

External LED