10xIoT

Python GPIO Library Examples

OCTOBER 2019 – THIS PAGE IS WORK IN PROGRESS.

This page contains Python examples for various Python GPIO Libraries. There is also a comparison table (https://10xiot.com/python-gpio-library-comparison-table/) available that summarises and compares the features of these libraries.

The source code for the examples on this page is available at **~~http://github.com/10xiot/python-gpio-library-examples (http://github.com/10xiot/python-gpio-library-examples)~~**

There are code samples for the following GPIO libraries.

- GPIOZero (#GPIOZero)
- Pi.GPIP (Coming Soon)
- Python Periphery (Coming Soon)
- PiGPIO (#PiGPIO)
- WiringPi (#WiringPi)
- Blinka (CircuitPython) (#BlinkaCircuitPython)

If you know of another Python GPIO library, or you can suggest improvements to any of the code shown below, please post a comment at the end of the page. I welcome all feedback and suggestions.

Technical Requirements

- All code samples below are **expected to be ran on a Raspberry Pi**. They may work on similar devices like a Banana or Orange Pi, however, this has not been confirmed. They will not work on a generic Windows, Mac or Linux PC.
- The code was tested against **Python 3.5**
- It is highly recommended that you use a **Python Virtual Environment** to run these examples. We have an ~~Introduction to Python Virtual Environments (http://TODO)~~ if you would like to learn more.

Reference Circuit

The following Python code examples work with the following circuit. The LED will turn on and off with each button press.

- The LED is connected to GPIO 21
- The Button is connected to GPIO 23
- The Resistor is 200Ω (or a *higher* value up to ~1kΩ will be suitable)

Figure 1 – LED & Button Circuit on Raspberry Pi Model B

Figure 2 – Semantic Diagram for Figure 1

> 💡 ~~**Learn the reasons**~~ ~~**(http://TODO)**~~ behind why the Resistor above is 200Ω.

Python Examples

The following code samples implement a small program where pressing the button toggles the LED on and off. When possible with a GPIO library, interrupt or callback handlers are used to intercept button presses, else we perform polling in a `while` loop.

(#)

GPIOZero

Documentation & API Reference: https://gpiozero.readthedocs.io (https://gpiozero.readthedocs.io)

LED & Button Python Example

```
"""
Turn on and off an LED with a Button using gpiozero.
Dependencies:
  pip install gpiozero
Written and tested with Python version 3.5
"""
from gpiozero import Device, LED, Button
import signal

LED_GPIO_PIN = 21
BUTTON_GPIO_PIN = 23

def pressed():
    led.toggle()
    state = 'on' if led.value == 1 else 'off'
    print("Button pressed: LED is " + state)

led = LED(LED_GPIO_PIN)
led.off()

button = Button(BUTTON_GPIO_PIN, pull_up=True, bounce_time=0.1)
button.when_pressed = pressed

print("Press button to turn LED on and off.")

signal.pause() # Stops main thread (program) from exiting.
```

Run Example

```
# Install dependencies
$ pip3 install gpiozero

# Run example
$ python3 gpiozero_led_button.py
```

(#)

PiGPIO

PiGPIO Resources

- PiGPIO Web Site: http://abyz.me.uk/rpi/pigpio (http://abyz.me.uk/rpi/pigpio)
- Python API Documentation: http://abyz.me.uk/rpi/pigpio/python.html (http://abyz.me.uk/rpi/pigpio/python.html)
- PiGPIO GitHub: https://github.com/joan2937/pigpio (https://github.com/joan2937/pigpio)

- PiGPIO Python Examples: http://abyz.me.uk/rpi/pigpio/examples.html#Python%20code (http://abyz.me.uk/rpi/pigpio/examples.html#Python%20code)

LED & Button Python Example

> The following example requires the PiGPIO daemon to be running on your Raspberry Pi (~~Instructions~~ ~~(http://todo.com)~~)

```
"""
Turn on and off an LED with a Button using pigpio.
Dependencies:
  pip install pigpio
Written and tested with Python version 3.5
"""
import pigpio
import signal

LED_GPIO_PIN = 21
BUTTON_GPIO_PIN = 23

pi = pigpio.pi()

# LED provides 'Output'
pi.set_mode(LED_GPIO_PIN, pigpio.OUTPUT)
pi.write(LED_GPIO_PIN, 0) # LED Off

# Button provides 'Input'
pi.set_mode(BUTTON_GPIO_PIN, pigpio.INPUT)
pi.set_pull_up_down(BUTTON_GPIO_PIN, pigpio.PUD_UP)
pi.set_glitch_filter(BUTTON_GPIO_PIN, 10000) # microseconds ~= debounce

# Button pressed handler
def pressed(gpio_pin, level, tick):
    # Get current pin state for LED.
    led_state = pi.read(LED_GPIO_PIN)

    if led_state == 1:
        # LED is on, so turn it off.
        pi.write(LED_GPIO_PIN, 0) # 0 = Pin Low = Led Off
        print("Button pressed: Led is off")
    else: # 0
        # LED is off, so turn it on.
        pi.write(LED_GPIO_PIN, 1) # 1 = Pin High = Led On
        print("Button pressed: Led is on")

# Register button handler.
pi.callback(BUTTON_GPIO_PIN, pigpio.FALLING_EDGE, pressed)

print("Press button to turn LED on and off.")
```

```
42.
43.     signal.pause() # Stops program from exiting.
```

Run Example

```
# Start PiGPIO daemon if not already running
$ pigpio &

# Install dependencies
$ pip3 install pigpio

# Run example
$ python3 pigpio_led_button_example.py
```

(#)

WiringPi

The following code sample uses the Python port of the popular C WiringPi library. See the “WiringPi C Library” web site for full details and API regarding the library.

WiringPi Resources:

- WiringPi Python Port (GitHub): https://github.com/WiringPi/WiringPi-Python (https://github.com/WiringPi/WiringPi-Python)
- WiringPi C Library (Web Site): http://wiringpi.com (http://wiringpi.com/)

LED & Button Python Example

```
1.  import wiringpi
2.  import time
3.  import signal
4.
5.  wiringpi.wiringPiSetupGpio() # BCM GPIO pin numbering
6.
7.  LED_GPIO_PIN = 21
8.  BUTTON_GPIO_PIN = 23
9.
10. # Configure LED Pin
11. wiringpi.pinMode(BUTTON_GPIO_PIN, wiringpi.INPUT)
12. wiringpi.digitalWrite(LED_GPIO_PIN, wiringpi.LOW)
13.
14. # Configure Push Button Pin
15. wiringpi.pinMode(LED_GPIO_PIN, wiringpi.OUTPUT)
16. wiringpi.pullUpDnControl(BUTTON_GPIO_PIN, wiringpi.GPIO.PUD_DOWN)
17.
```

```
18.  # Debouncing variables
19.  debounce_timer = 0
20.  debounce_threshold = 0.25 # Increase if button is erratic
21.
22.  def button_pressed():
23.      global debounce_timer
24.
25.      if (time.time() - debounce_timer) < debounce_threshold:
26.          return;
27.
28.      state = wiringpi.digitalRead(LED_GPIO_PIN) # 0 (LOW) or 1 (HIGH)
29.      state = not state
30.
31.      wiringpi.digitalWrite(LED_GPIO_PIN, state) # Toggle LED On/Off
32.
33.      if state:
34.          print("LED is on")
35.      else:
36.          print("LED is off")
37.
38.      debounce_timer = time.time()
39.
40.  # Register Push Button Handler. GPIO.INT_EDGE_FALLING => Button Pressed
41.  wiringpi.wiringPiISR(BUTTON_GPIO_PIN, wiringpi.GPIO.INT_EDGE_FALLING,
     button_pressed)
42.
43.  signal.pause()
```

Run Example

```
# Install dependencies
$ pip3 install wiringpi

# Run example
$ python3 wiringpi_led_button_example.py
```

(#)

Blinka (CircuitPython)

The following code example uses Circuit Python and the CPython compatibility layer Blinka.

Circuit Python & Blinka Resources:

- CircuitPython web site: https://circuitpython.org (https://circuitpython.org/)

- CircuitPython API Documentation: https://circuitpython.readthedocs.io (https://circuitpython.readthedocs.io/)
- CircuitPython Introduction: https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython (https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython)
- Blinka GitHub: https://github.com/adafruit/Adafruit_Blinka (https://github.com/adafruit/Adafruit_Blinka)

LED & Button Python Example

```
import digitalio
import board
from digitalio import DigitalInOut, Direction, Pull
import time

LED_GPIO_PIN = board.D21
BUTTON_GPIO_PIN = board.D23

# Configure LED
led = digitalio.DigitalInOut(LED_GPIO_PIN)
led.direction = digitalio.Direction.OUTPUT

# Configure Push Button
button = DigitalInOut(BUTTON_GPIO_PIN)
button.direction = Direction.INPUT
button.pull = Pull.UP

pressed = False

while True:
    if not pressed and button.value == False:

        # Button is pressed
        pressed = True

        led.value = not led.value

        if led.value:
            print("LED is on")
        else:
            print("LED is off")

    elif pressed and button.value == True:

        # Button released
        pressed = False

    time.sleep(0.01) # debounce button
```

Run Example

```
# Install dependencies. Blinks has a dependency on RPi.GPIO
$ pip3 install rpi.gpio adafruit-blinka

# Run example
$ python3 blinka_example.py
```

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

post comment