Application to blink all RED LEDs from 3 RGB LEDs

```
import time
import sys
sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-legacy/Adafruit_MCP230xx')
from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)
mcp.config(0, mcp.OUTPUT)
mcp.config(3, mcp.OUTPUT)
mcp.config(6, mcp.OUTPUT)
while True:
    mcp.output(0, 1) #RED LED1 ON
    mcp.output(3, 1) #RED LED2 ON
    mcp.output(6, 1) #RED LED3 ON
    time.sleep(1)
    mcp.output(0, 0) #RED LED1 OFF
    mcp.output(3, 0) #RED LED1 OFF
    mcp.output(6, 0) #RED LED1 OFF
    time.sleep(1)
```

Application to read and print the status of both PUSH BUTTON

```
import time

import sys

sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-legacy/Adafruit_MCP230xx')

from Adafruit_MCP230XX import Adafruit_MCP230XX

mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)

mcp.config(9, mcp.INPUT)

mcp.pullup(9, 1)

mcp.config(10, mcp.INPUT)

mcp.pullup(10, 1)

while (True):

    print "Pin 9 = %d" % (mcp.input(9))

    print "Pin 10 = %d" % (mcp.input(10))

    time.sleep(2)
```

Application to ON/OFF the BUZZER

```
import time
import sys
sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-legacy/Adafruit_MCP230xx')
from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)
mcp.config(11, mcp.OUTPUT)
while (True):
    mcp.output(11, 1) #BUZZER ON
    time.sleep(1)
    mcp.output(11, 0) #BUZZER OFF
    time.sleep(1)
```

Application to connect a PIR(Digital) sensor and print the status of Human presence.

```
import sys
sys.path.append('/home/pi/Adafruit-Raspberry-Pi-
Python-Codelegacy/
Adafruit_MCP230xx')
from Adafruit_MCP230XX import
Adafruit_MCP230XX
import time
mcp = Adafruit_MCP230XX(busnum = 1, address =
0x21, num_gpios = 16)
mcp.config(0, mcp.INPUT)
mcp.pullup(0, 1)
while (True):
    i = mcp.input(0)
    time.sleep(1)
    if i == 1:
        print "person detect"
    if i == 0:
        print "person not detect"
```

Application to connect a ULTRASONIC(Digital) sensor and print the distance of object.

```
import sys
import time
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO_TRIGGER = 16 ##connect with RPI16
GPIO_ECHO = 18 ##connect with RPI18
GPIO.setup(GPIO_TRIGGER,GPIO.OUT) # Trigger
GPIO.setup(GPIO_ECHO,GPIO.IN) # Echo
GPIO.output(GPIO_TRIGGER, False)
time.sleep(0.5)
while True:
    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
    #start = time.time()
    while GPIO.input(GPIO_ECHO)==0:
    start = time.time()
    while GPIO.input(GPIO_ECHO)==1:
        stop = time.time()
    elapsed = stop-start
    distance = elapsed * 34300
    distance = distance / 2
    print "Distance : %.1f" % distance
    #GPIO.cleanup()
```

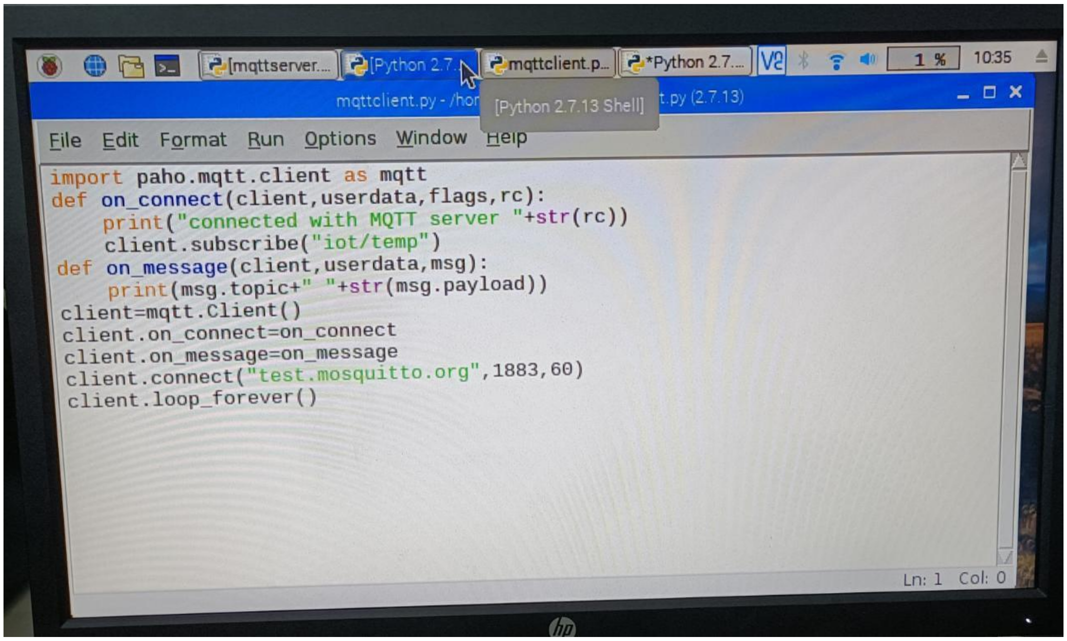
```
# Application to connect a SERVO Motor and Rotate 180°
mode of operation.
import RPi
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(22, GPIO.OUT)
pwm=GPIO.PWM(22,100)
pwm.start(5)
angle1=10
duty1= float(angle1)/10 + 2.5
angle2=160
duty2= float(angle2)/10 + 2.5
ck=0
while ck<=5:
      pwm.ChangeDutyCycle(duty1)
      time.sleep(0.8)
      pwm.ChangeDutyCycle(duty2)
      time.sleep(0.8)
      ck=ck+1
time.sleep(1)
GPIO.cleanup()
```

```
    #Soil Moisture Program
    import RPi.GPIO as GPIO
     import spidev
    import time
    spi = spidev.SpiDev()
    spi.open(0,0)
    def ReadChannel(channel):
        adc = spi.xfer2([1,(8+channel)<<< 8) + adc[2]
        return data

    def ConvertVolts(data,places):
        volts = (data * 3.3) / float(1023)
        volts = round(volts,places)
        return volts

    moisture_channel = 0 #CONNECT ANALOG INPUT A0
    delay = 3
    while True:
     moisture_level = ReadChannel(moisture_channel)
    moisture_volts = ConvertVolts(moisture_level,2)
    print
"____________________________________________" print
("Moisture: {} ({}V)".format(moisture_level,moisture_volts))
```

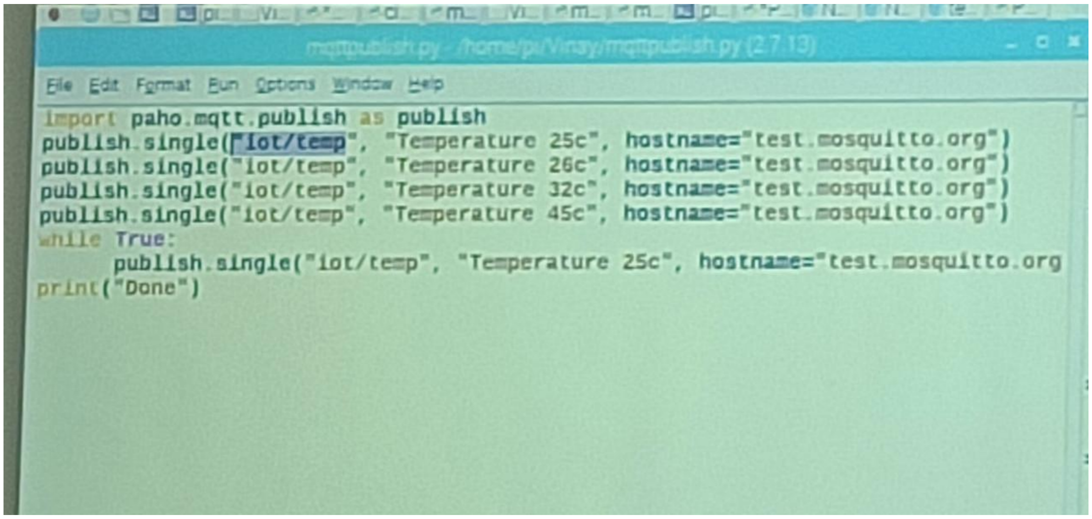
```
#Bluetooth Server
import bluetooth
import time
import sys
sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-legacy/Adafruit_MCP230xx')
from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)
mcp.config(0, mcp.OUTPUT)
server_sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
port = 1
server_sock.bind(("",port))
server_sock.listen(1)

client_sock, address = server_sock.accept()
print("Accepted connection from :",address)

while True :
    data = client_sock.recv(1024)
    print("Received : ",data)
    if data == "ON" :
        mcp.output(0,1)
        print("RED LED ON")
    if data == "OFF" :
        mcp.output(0,0)
        print("RED LED OFF")
        time.sleep(0.5)
    input = (mcp.input(9))
    if input == 512 :
        text = "Input based on push button"
        client_sock.send(text)
    time.sleep(0.5)
client_sock.close()
server_sock.close()
```

```
#Bluetooth-Client
import bluetooth
import time
bd_addr = "B8:27:EB:1B:47:09"
port = 1
sock = bluetooth.BLuetoothSocket(bluetooth.RFCOMM)
sock.connect((bd_addr, port))
while True :
    text = raw_input("Enter your message : ")
    sock.send(text)
    time.sleep(1)
    data = sock.recv(1024)
    print("Received : ",data)
    time.sleep(1)
sock.close()
```

```
#Bluetooth-Commands
sudo bluetoothctl
power on
agent on
default-agent
discoverable on
scan on
pair < ADDR>
```



```
import paho.mqtt.client as mqtt
def on_connect(client,userdata,flags,rc):
    print("connected with MQTT server "+str(rc))
    client.subscribe("iot/temp")
def on_message(client,userdata,msg):
    print(msg.topic+" "+str(msg.payload))
client=mqtt.Client()
client.on_connect=on_connect
client.on_message=on_message
client.connect("test.mosquitto.org",1883,60)
client.loop_forever()
```



mqttpublish.py - /home/pi/Vinay/mqttpublish.py (2.7.13)
File Edit Format Run Options Window Help

```
import paho.mqtt.publish as publish
publish.single("iot/temp", "Temperature 25c", hostname="test.mosquitto.org")
publish.single("iot/temp", "Temperature 26c", hostname="test.mosquitto.org")
publish.single("iot/temp", "Temperature 32c", hostname="test.mosquitto.org")
publish.single("iot/temp", "Temperature 45c", hostname="test.mosquitto.org")
while True:
    publish.single("iot/temp", "Temperature 25c", hostname="test.mosquitto.org
print("Done")
```