1 LED Blinking

```
import time
import RPi.GPIO as GPIO

TRUE = 1

led1 = 20
led2 = 21
led3 = 22
led4 = 23
led5 = 24
led6 = 25
led7 = 26
led8 = 27

myled=[led1,led2,led3,led4,led5,led6,led7,led8]

GPIO.setmode(GPIO.BCM)
GPIO.setup(led1,GPIO.OUT)
GPIO.setup(led2,GPIO.OUT)
GPIO.setup(led3,GPIO.OUT)
GPIO.setup(led4,GPIO.OUT)
GPIO.setup(led5,GPIO.OUT)
GPIO.setup(led6,GPIO.OUT)
GPIO.setup(led7,GPIO.OUT)
GPIO.setup(led8,GPIO.OUT)

def ledState(led,a):
  GPIO.output(myled[led],a)

try:
  for i in range(0,8):
    ledState(i,1)

  while TRUE:
    for i in range(0,8):
```

```
        ledState(i,1)
        time.sleep(0.5)
        ledState(i,0)
        time.sleep(0.5)



# If CTRL+C is pressed the main loop is broken
except KeyboardInterrupt:
    RUNNING = False
    print ("\program closed")

# Actions under 'finally' will always be called
finally:
    # Stop and finish cleanly so the pins
    # are available to be used again
    GPIO.cleanup()
```

2 Button

```
import time
from gpiozero import LED
from gpiozero import Button

button = Button(14)

led1 = LED(20)
led2 = LED(21)
led3 = LED(22)
led4 = LED(23)
led5 = LED(24)
led6 = LED(25)
led7 = LED(26)
led8 = LED(27)

led1.off()
led2.off()
led3.off()
led4.off()
led5.off()
led6.off()
led7.off()
led8.off()

while True:
  try:
    if button.is_pressed:
      led1.on()
      led2.on()
      led3.on()
      led4.on()
      led5.on()
      led6.on()
```

```
        led7.on()
        led8.on()
    else:
        led1.off()
        led2.off()
        led3.off()
        led4.off()
        led5.off()
        led6.off()
        led7.off()
        led8.off()

except KeyboardInterrupt:
    print("closing")
    exit()
```

3. Buzzer

```
import time
import RPi.GPIO as GPIO

TRUE = 1

buzzer = 3

GPIO.setmode(GPIO.BCM)
GPIO.setup(buzzer,GPIO.OUT)

def buzzerState(val):
  GPIO.output(buzzer,val)



try:
  while TRUE:
    buzzerState(1)
    time.sleep(1)
    buzzerState(0)
    time.sleep(1)



# If CTRL+C is pressed the main loop is broken
except KeyboardInterrupt:
  RUNNING = False
  print ("\program closed")

# Actions under 'finally' will always be called
finally:
  # Stop and finish cleanly so the pins
  # are available to be used again
  GPIO.cleanup()
```

4. IRLED

```
import time
import RPi.GPIO as GPIO

RUNNING = True

HIGH  = 1
LOW  = 0
DetectPin = 5
led = 8

def InitSystem():
  GPIO.setmode(GPIO.BCM)
  GPIO.setup(DetectPin,GPIO.IN,pull_up_down=GPIO.PUD_UP)
  GPIO.setup(led,GPIO.OUT)
  return


def DetectPerson():
  while True:
    input_state = GPIO.input(DetectPin)
    time.sleep(0.3)
    if input_state == 0:
      return LOW
    else:
      return HIGH




try:
  print ("\nCounting using IR LED\n")
  print  ("-------------------------------------------------\n")
  InitSystem()
  count =0;
```

```
  while RUNNING:
    state = DetectPerson()
    if state == LOW:
      count+=1
      print ("person count =%d" %count)
      GPIO.output(led,LOW)
      time.sleep(1)
      GPIO.output(led,HIGH)

# If CTRL+C is pressed the main loop is broken
except KeyboardInterrupt:
  RUNNING = False



# Actions under 'finally' will always be called
finally:
  # Stop and finish cleanly so the pins
  # are available to be used again
  GPIO.cleanup()
```

5. Temp_Sensor

```
import time
from gpiozero import LED
from w1thermsensor  import W1ThermSensor


sensor =  W1ThermSensor()


led1 = LED(20)
led2 = LED(21)
led3 = LED(22)
led4 = LED(23)
led5 = LED(24)
led6 = LED(25)
led7 = LED(26)
led8 = LED(27)


led1.off()
led2.off()
led3.off()
led4.off()
led5.off()
led6.off()
led7.off()
led8.off()


while True:
  temp  = sensor.get_temperature()
  print(" the temp is  %s celcius" %temp)
  if temp  >= 29:
    led1.off()
    led2.off()
    led3.off()
    led4.off()
    led5.off()
    led6.off()
```

```
    led7.off()
    led8.off()
    print("limit exceeded")
  else:
    led1.on()
    led2.on()
    led3.on()
    led4.on()
    led5.on()
    led6.on()
    led7.on()
    led8.on()
  time.sleep(1)
```