

LED BLINKING

29 lines (24 sloc) 1.04 KB

```
1  import RPi.GPIO as GPIO # RPi.GPIO can be referred as GPIO from now
2  import time
3
4  ledPin = 22    # pin22
5
6  def setup():
7      GPIO.setmode(GPIO.BOARD)      # GPIO Numbering of Pins
8      GPIO.setup(ledPin, GPIO.OUT)   # Set ledPin as output
9      GPIO.output(ledPin, GPIO.LOW)  # Set ledPin to LOW to turn Off the LED
10
11  def loop():
12      while True:
13          print 'LED on'
14          GPIO.output(ledPin, GPIO.HIGH)  # LED On
15          time.sleep(1.0)                  # wait 1 sec
16          print 'LED off'
17          GPIO.output(ledPin, GPIO.LOW)   # LED Off
18          time.sleep(1.0)                  # wait 1 sec
19  def endprogram():
20
21      GPIO.output(ledPin, GPIO.LOW)      # LED Off
22      GPIO.cleanup()                     # Release resources
23
24  if __name__ == '__main__':             # Program starts from here
25      setup()
26      try:
27          loop()
28      except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the destroy() will be executed.
29          endprogram()
```



33 lines (33 sloc) 797 Bytes

TRAFFIC LIGHT

```
1  import RPi.GPIO as GPIO
2  import time
3  import signal
4  import sys
5  GPIO.setmode(GPIO.BCM)
6  GPIO.setup(9, GPIO.OUT)
7  GPIO.setup(10, GPIO.OUT)
8  GPIO.setup(11, GPIO.OUT)
9  def allLightsOff(signal, frame):
10     GPIO.output(9, False)
11     GPIO.output(10, False)
12     GPIO.output(11, False)
13     GPIO.cleanup()
14     sys.exit(0)
15  signal.signal(signal.SIGINT, allLightsOff)
16  while True:
17     # Red
18     GPIO.output(9, True)
19     time.sleep(3)
20     # Red and amber
21     GPIO.output(10, True)
22     time.sleep(1)
23     # Green
24     GPIO.output(9, False)
25     GPIO.output(10, False)
26     GPIO.output(11, True)
27     time.sleep(5)
28     # Amber
29     GPIO.output(11, False)
30     GPIO.output(10, True)
31     time.sleep(2)
32     # Amber off (red comes on at top of loop)
33     GPIO.output(10, False)
```