

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Author: Matt Hawkins
4  # Author's Git: https://bitbucket.org/MattHawkinsUK/
5  # Author's website: https://www.raspberrypi-spy.co.uk
6  # http://elecrow.com/
7
8  import smbus
9  import time
10
11 bus = smbus.SMBus(1)
12
13 class LightSensor():
14
15     def __init__(self):
16
17         # Define some constants from the datasheet
18
19         self.DEVICE = 0x5c # Default device I2C address
20
21         self.POWER_DOWN = 0x00 # No active state
22         self.POWER_ON = 0x01 # Power on
23         self.RESET = 0x07 # Reset data register value
24
25         # Start measurement at 4lx resolution. Time typically 16ms.
26         self.CONTINUOUS_LOW_RES_MODE = 0x13
27         # Start measurement at 1lx resolution. Time typically 120ms
28         self.CONTINUOUS_HIGH_RES_MODE_1 = 0x10
29         # Start measurement at 0.5lx resolution. Time typically 120ms
30         self.CONTINUOUS_HIGH_RES_MODE_2 = 0x11
31         # Start measurement at 1lx resolution. Time typically 120ms
32         # Device is automatically set to Power Down after measurement.
33         self.ONE_TIME_HIGH_RES_MODE_1 = 0x20
34         # Start measurement at 0.5lx resolution. Time typically 120ms
35         # Device is automatically set to Power Down after measurement.
36         self.ONE_TIME_HIGH_RES_MODE_2 = 0x21
37         # Start measurement at 1lx resolution. Time typically 120ms
38         # Device is automatically set to Power Down after measurement.
39         self.ONE_TIME_LOW_RES_MODE = 0x23
40
41     def convertToNumber(self, data):
42
43         # Simple function to convert 2 bytes of data
44         # into a decimal number
45         return ((data[1] + (256 * data[0])) / 1.2)
46
47     def readLight(self):
48
49         data = bus.read_i2c_block_data(self.DEVICE, self.ONE_TIME_HIGH_RES_MODE_1)
50         return self.convertToNumber(data)
51
52 def main():
53
```

```
54     sensor = LightSensor()
55     try:
56         while True:
57             print("Light Level : " + str(sensor.readLight()) + " lx")
58             time.sleep(0.5)
59     except KeyboardInterrupt:
60         pass
61
62 if __name__ == "__main__":
63     main()
64
```