

Projekt zu Micropython: Helligkeitssensor

Sebastian Stigler

1 Messen der Helligkeit mit dem BH1750 i²c Sensor

Einige wichtige Zahlen aus dem Datenblatt:

i²c-Adresse (ADDR Pin ist Low) 0x23

i²c-Bus Taktfrequenz 400kHz

Wartezeit bei *niedriger* Auflösung zwischen Mess-Kommando und der Abfrage des Ergebnisses
: 24 ms

Wartezeit bei *hoher* Auflösung zwischen Mess-Kommando und der Abfrage des Ergebnisses
: 180 ms

Im Hochauflösungsmodus 2 muss das errechnete Ergebnis durch zwei geteilt werden.

Im Datenblatt finden Sie auf der Seite 4 den Zustandsautomaten der Sensors. Auf Seite 5 eine Tabelle mit den i²c Kommandos. Schließlich finden Sie auf Seite 7 im Beispiel 1 (ex 1) die Abfrage Sequenz für eines der Mess-Kommando.

```
bh1750.py
class BH1750
  Class Variables
  PWR_OFF = 0x00
  PWR_ON = 0x01
  RESET = 0x07

  CONT_HIRES_1 = 0x10
  CONT_HIRES_2 = 0x11
  CONT_LOWRÉS = 0x13

  ONCE_HIRES_1 = 0x20
  ONCE_HIRES_2 = 0x21
  ONCE_LOWRÉS = 0x23

  Attributes
  bus : I2C
  addr : int

  __init__(bus : I2C, addr : int = 0x23)
  set_mode(mode : int)
  on()
  off()
  reset()
  luminance(mode : int) -> float
```

Abbildung 1: bh1750.py

Schreiben Sie eine Klasse **BH1750** die beim Initialisieren ein instantiiertes I2C-Objekt und optional die i²c-Adresse des Sensors bekommt, wenn diese von **0x23** abweicht. Es gilt:

Klassenvariablen:

Hier werden die Kommando Codes aus der Tabelle auf Seite 5 im Datenblatt im Hex-Format gespeichert. Die Kommandos zum Kalibrieren des Sensors werden nicht implementiert. Entnehmen Sie die Werte und Variablenamen dem obigen UML Diagramm.

Attribute:

bus speichert das I2C Objekt ab, über das die Interaktion mit dem i²c-Bus stattfindet.

addr ist die Adresse des Sensors.

Methoden:

__init__(bus, addr=0x23) initialisiert die Attribute und sendet ein **off** und dann ein **reset** an den Sensor.

set_mode(mode) sendet einen Kommando Code über i²c an den Sensor.

on() führt das *Power On* Kommando aus.

off() führt das *Power Down* Kommando aus.

reset() führt das *Reset* Kommando aus. Beachten Sie den Hinweis in der Tabelle im Datenblatt!

luminance(mode) führt eine Messung im angegebenen Modus durch und berechnet aus der Ausgabe des Sensors den Wert in Lux und gibt diesen als float zurück. Betrachten Sie dafür das erste Beispiel auf Seite 7 im Datenblatt. Tipp: Die Funktion **sleep_ms** aus dem Modul **utime** könnte nützlich sein.

Testen

Testen Sie Ihren Code mit **testbh1750.py**.

Die Datei **bh1750.py** enthält die Klasse **BH1750**.