```
# -*- coding: utf-8 -*-
import smbus
import time

class SO1602A():
    def __init__(self, sa0 = 0, cursor = False, blink = False): #確認2018/12/29
        self.bus = smbus.SMBus(1)       #smbusの実態を作成
        if (sa0 == 0):                  #SO1602Aのスレーブアドレスは、0x3C（SA0=Low）か0x3D（SA0=High）。
            self.addr = 0x3c
        else:
            self.addr = 0x3d
        self.clearDisplay()             #画面消去
        self.returnHome()               #ホームポジションに戻る
        self.displayOn(cursor, blink)   #カーソルと点滅を消去
        self.clearDisplay()
    def writeLine(self, str = '', line = 0, align = 'left'):    #確認2018/12/29
        while (len(str) < 16):          # 文字列が16文字に満たない場合空白で埋める
            if (align == 'right'):
                str = ' ' + str
            else:
                str = str + ' '
        if (line == 1):                 # カーソル位置をあわせる
            self.bus.write_byte_data(self.addr, 0x00, (0x80 + 0x20))
        else:
            self.bus.write_byte_data(self.addr, 0x00, 0x80)
        for i in range(len(str)):       # 1文字ずつ送信
            self.bus.write_byte_data(self.addr, 0x40, ord(str[i]))
    def clearDisplay(self):                                     #確認2018/12/29
        self.bus.write_byte_data(self.addr, 0x00, 0x01) #空白(0x20)で表示メモリを埋める

    def returnHome(self):                                       #確認2018/12/29
        self.bus.write_byte_data(self.addr, 0x00, 0x02) #左上の表示位置0x00に戻る　表示内容の変更はない。

    def displayOn(self, cursor = False, blink = False):         #確認2018/12/29
        cmd = 0x0c          #表示コマンド
        if (cursor):        #カーソル
            cmd += 0x02
        if (blink):         #点滅
```

```
            cmd += 0x01
        self.bus.write_byte_data(self.addr, 0x00, cmd)  #点灯コマンド送信

    def displayOff(self):                                       #動作未確認
        self.bus.write_byte_data(self.addr, 0x00, 0x08) #消灯コマンド送信

#def main():
#    oled = SO1602A(sa0 = 0)                                #SO1602Aの実態を作成
#    oled.writeLine(str = 'Hello LE-lab!', line = 0)      #上側の表示
#    time.sleep(0.5)
#    oled.writeLine(str = 'Have a nice day!', line = 1)  #下側の表示
#
#if __name__ == '__main__':
#    main()
#
#**********SMBus（システム管理バス）は、I2Cプロトコルのサブセットです。
**********
#I2Cデバイス用のドライバを書くとき、SMBusアダプタとI2Cアダプタの両方でデバイスドライバを使用することを可能にするので、
#可能であれば（デバイスがI2Cプロトコルのそのサブセットのみを使用する場合）SMBusコマンドを使用するようにしてください。
#注アドレスは、読み取り/書き込みビットを除いた7ビットアドレスです
#（読み取り/書き込みビットに追加されると、1ビット左にシフトされます）。
#long write_quick（int addr）
#   読み書きビットのみを送信する
#long read_byte（int addr）
#   デバイスレジスタを指定せずに、デバイスから1バイトを読み取ります。
#long write_byte（int addr、char val）
#   1バイトをデバイスに送信する
#long read_byte_data（int addr、char cmd）
#   バイトデータ読み取りトランザクション。
#long write_byte_data（int addr、char cmd、char val）
#   バイトデータ書き込みトランザクション。
#long read_word_data（int addr、char cmd）
#   Wordデータトランザクションを読み取ります。
#long write_word_data（int addr、char cmd、int val）
#   Word Dataトランザクションを書き込みます。
#long process_call（int addr、char cmd、int val）
#   コール処理トランザクション。
#long [] read_block_data（int addr、char cmd）
#   ブロックデータトランザクションの読み取り
```

```
#write_block_data（int addr、char cmd、long vals []）
#　デバイスに最大32バイトを書き込みます。この関数は、valls配列の前にvals配列の長さを示す最初のバイトを追加します。
#　代わりにwrite_i2c_block_dataを使用してください。
#long [] block_process_call（int addr、char cmd、long vals []）
#　ブロックプロセスコールトランザクション。
#
#I2Cアクセス機能
#
#long [] read_i2c_block_data（int addr、char cmd）
#　読み取りトランザクションをブロックします。
#write_i2c_block_data（int addr、char cmd、long vals []）
#　ブロック書き込みトランザクション。
#
#Code Example
#
##!/usr/bin/python
#
#import smbus
#
#bus = smbus.SMBus(1)    # 0 = /dev/i2c-0 (port I2C0), 1 = /dev/
i2c-1 (port I2C1)
#
#DEVICE_ADDRESS = 0x15      #7 bit address (will be left shifted to
add the read write bit)
#DEVICE_REG_MODE1 = 0x00
#DEVICE_REG_LEDOUT0 = 0x1d
#
##Write a single register
#bus.write_byte_data(DEVICE_ADDRESS, DEVICE_REG_MODE1, 0x80)
#
##Write an array of registers
#ledout_values = [0xff, 0xff, 0xff, 0xff, 0xff, 0xff]
#bus.write_i2c_block_data(DEVICE_ADDRESS, DEVICE_REG_LEDOUT0,
ledout_values)
#
#参考先
#http://raspberry-projects.com\
#/pi/programming-in-python/i2c-programming-in-python/using-the-i2c-
interface-2
#
**********************************************************************
*************
# 参考  https://gist.github.com/uhey22e/
2a4d676363722d59428a937ac729def0#file-so1602a-py-L2
```