

Fejlesztési dokumentáció

Napfény monitorozó, világítás vezérlő

A program leírása:

A program ötlete az ember ébredéséhez köthető. Az ember fényre könnyebben felébred, könnyebben felkel, mint teljes sötétségben. (Például a nem teljesen leengedett redőny által beengedett fény is segít.) Ezáltal a napja is könnyebben indul. Így egy olyan programot készítettünk, amely a reggeli esetlegesen nem elégséges fényt, fokozatosan kipótolja egy lámpa segítségével. A programot eredetileg csak reggeli ébresztéshez gondoltuk, de használható egész napos fény szabályozásra is. (így akár komfort növelésre, és költség csökkentésre is).

A „**Felhasználói Dokumentáció**” -ban leírtak szerint csatlakoztatni kell a fényérzékelő szenzort, illetve fel kell telepíteni az ott leírtak alapján a szoftvereket.

Program részeinek működése:

Szükséges importálni az alábbi Python könyvtárakat

```
import board
import digitalio
import busio
import adafruit_tsl2561
import csv
import time
import logging
import datetime
```

Ezután lehetőségünk van beállítani a szobában elérni kívánt lux értéket:

```
cellux = 500
```

A program indulásakor a lámpa alap esetben kikapcsolva van, azaz 0 értéket vesz fel:

```
lampaertek = 0
```

Opcionálisan tudjuk állítani, hogy a szenzor milyen sűrűn mérjen értéket, ez az érték másodpercben adható meg.

(default a 60 másodperc, mert a webes megjelenítés erre van konfigurálva, ha kevesebb mint 60 másodpercet szeretnénk, akkor feltétlen szükséges a webes monitorozó hozzáigazítása):

```
meresidokoz = 60
```

Lehetőség van állítani a programban ,hogy hány alkalommal történjen mérés.
(jelen esetben ,hogy hány percig működjön vezérlés)

```
waketime = 3
```

Az adafruit könyvtárban található teszt parancsokkal lehetőség van tesztelni a Raspberry,
és a fényszerezor közötti kommunikációt, és ki is lehet iratni.

```
pin = digitalio.DigitalInOut(board.D4)
print("Digital IO ok")
i2c = busio.I2C(board.SCL, board.SDA)
print("I2C ok!")
spi = busio.SPI(board.SCLK, board.MOSI, board.MISO)
print("SPI ok!")
```

A fényszenzor bekapcsolása, és inicializálása, és az aktuális értékek bekérése:

```
i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tsl2561.TSL2561(i2c)
```

Illetve az aktuális értékek kiírására is van lehetőség, melyekből mi, a Lux értéket
használjuk a továbbiakban:

```
print('Lux: {}'.format(sensor.lux))

print('Broadband: {}'.format(sensor.broadband))
print('Infrared: {}'.format(sensor.infrared))
print('Luminosity: {}'.format(sensor.luminosity))
```

Mivel az adatok CSV fájlba lesznek mentve, ezért ha excelbe szeretnénk megnézni az
adatokat, érdemes fejléccel ellátni az oszlopokat. Melyet a következő sorokkal tehetünk
meg:

(A webes monitorozó nem feltétlen indokolná)

```
with open('data1.csv', 'w', newline='') as f:
    thewriter = csv.writer(f)
    thewriter.writerow(['aktualis_kinti_feny', 'aktualis_lampa_ertek',
        'lampaszazalek', 'datum'])
    f.close()
```

Ciklus segítségével ütemezzük a mérések darabszámát, annyiszor fut le , amennyiben a
„waketime” változót meghatároztuk:

```
for k in range(0, waketime) :
```

Az aktuális rendszeridőt lekérjük, melyet a CSV-be megfelelő formátumba írunk bele:

```
date = datetime.datetime.now()
now = str(date.hour) + ':' + str(date.minute)
```

A szenzor által mért aktuális érték elmentése az „actlux” változóba, majd sztringgé alakítása, hogy a CSV-be kiírható legyen.

```
actlux = sensor.lux
str(actlux)
```

A program fő része, ami a vezérlés logikáját adja.

Ha szükséges elkezdi emelni a lámpának az értékét. 100 Lumenenként, illetve ha a cél és a mért érték különbsége kisebb ,mint 100 Lumen, akkor pontos értékkel növeli a lámpa értéket.

```
if actlux > cellux:
    lampaertek = 0
else:
    print('lampaertek',lampaertek)
    if (lampaertek + actlux) < 400:
        lampaertek = lampaertek + 100
    else:
        lampaertek = (cellux - actlux)
```

A lámpa érték százalékká alakítása (500 luxos lámpával számolva):

```
lampaszazalek = (lampaertek / 500) * 100
```

A CSV-be kiírandó értékek változóba mentése, a további könnyebb kezelhetőségért:

```
a = str(actlux)
b = str(lampaertek)
c = str(lampaszazalek)
```

Az értékek kiírása ugyanabba a CSV-be, aminek már előzetesen kitöltöttük a fejlécét. A CSV-t minden új érték beolvasás után le kell zárni , hogy belekerüljön a fájlba, így a monitorozó program real-timeba , valós időbe tudja mutatni a mért , és számított értékeket.

```
with open('data1.csv', 'a', newline='') as csvfile:

    thewriter = csv.writer(csvfile)
    thewriter.writerow([actlux, lampaertek, lampaszazalek, str(now)])
    csvfile.close()
```

A program futása során a képernyőre is kiirathatjuk az aktuális értékeket:

```
print('aktualis_kinti_feny' ,actlux)  
print('aktualis_lampa_ertek' ,lampaertek)
```

A „meresidokoz” változóban meghatározott ideig , szünetelteti a program következő lépését:

```
time.sleep(meresidokoz)
```

A forráskód:

```
import board

import digitalio
import busio
import adafruit_tsl2561
import csv
import time
import logging
import datetime

cellux = 500
lampaertek = 0

'''masodpercben'''
meresidokoz = 60
''' A fény kezelése ennyi percig fog működni'''
waketime = 3

'''adatkommunikáció teszt'''
pin = digitalio.DigitalInOut(board.D4)
print("Digital IO ok")
i2c = busio.I2C(board.SCL, board.SDA)
print("I2C ok!")
spi = busio.SPI(board.SCLK, board.MOSI, board.MISO)
print("SPI ok!")

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_tsl2561.TSL2561(i2c)

'''
print('Lux: {}'.format(sensor.lux))
print('Broadband: {}'.format(sensor.broadband))
print('Infrared: {}'.format(sensor.infrared))
print('Luminosity: {}'.format(sensor.luminosity))
'''

k = 0
with open('data1.csv', 'w', newline='') as f:
    thewriter = csv.writer(f)
    thewriter.writerow(['aktualis_kinti_feny', 'aktualis_lampa_ertek',
                        'lampaszazalek', 'datum'])
    f.close()

    for k in range(0, waketime) :
        date = datetime.datetime.now()
        now = str(date.hour) + ':' + str(date.minute)
        actlux = sensor.lux
        str(actlux)
```

```
if actlux > cellux:
    lampaertek = 0
else:
    print('lampaertek',lampaertek)
    if (lampaertek + actlux) < 400:
        lampaertek = lampaertek + 100
    else:
        lampaertek = (cellux - actlux)

lampaszazalek = (lampaertek / 500) * 100
a = str(actlux)
b = str(lampaertek)
c = str(lampaszazalek)

with open('data1.csv', 'a', newline='') as csvfile:
    thewriter = csv.writer(csvfile)
    thewriter.writerow([actlux, lampaertek, lampaszazalek, str(now)])
    csvfile.close()

print('aktualis_kinti_feny' ,actlux)
print('aktualis_lampa_ertek' ,lampaertek)

time.sleep(meresidokoz)
```