

Datalogger Opgave

Import opsætning

Til at starte med importere jeg nogle biblioteker, så jeg kan bruge deres funktioner senere. Import csv gør mig i stand til at lave csv filer, derefter hentes sense hattens funktioner ind, efterfulgt af datetime, som kan give mig en dato og et klokkesæt.

```
5
6     #Importer det moduler der skal bruges til at kunne lave csv filer, timestamp, sensor læsning etc.
7     import csv
8     from sense_hat import SenseHat
9     from datetime import datetime
10    from time import sleep
11
```

Variable opsætning

Her sættes sense op til at bruge sense hatten.

```
12    #Sense sættes til sense hatten så den ved det er den som skal bruges
13    sense = SenseHat()
14    sense.clear()
15
```

Initialisering

Initialisere det hele med en ordbog af ting jeg kommer til at exportere til de filer som bliver lavet med informationen senere.

```
17
18    #Initialisere det hele med en "ordbog" af ting jeg kommer til at exportere til filerne
19    def __init__(self):
20        self.data_dict = {}
21
```

Data indsamling

Indsamling af de forskellige sensors data sker her, ved at bruge self.data_dict laves en fil med et label som er givet i [' '], den data som vil blive skrevet er filerne kommer efter = tegnet, så det skriver en dato, et klokkesæt, og den information som bliver hentet fra sensorerne.

```
22    #Indsamling af de forskellige sensors data
23    def collect_data(self):
24        #self.data dict laver en fil med et label som er givet i [' ']
25        #og dataen som kommer i den fil er så det information den henter efter = tegnet
26        #så dataen bliver datetime.now() som giver dato og klok, og den information som kan læses fra givne sensor temp, fugt, tryk...
27        self.data_dict['temperatur'] = (datetime.now(), sense.get_temperature())
28        self.data_dict['fugtighed'] = (datetime.now(), sense.get_humidity())
29        self.data_dict['tryk'] = (datetime.now(), sense.get_pressure())
30
```

Konsol udskrivning

En simple funktion til at udskrive informationen i konsol vinduet, i en let læsligt formateret måde, til hurtig inspektion.

```
30
31 #Konsol print af data i formateret strings så det er let læsligt
32 def print_data(self):
33     print("-"*120)
34     print("~ {0:%d-%m-%Y, %H:%M:%S} ~" .format(*self.data_dict['temperatur']))
35     print("Temperatur: {1:,.2f}" .format(*self.data_dict['temperatur']))
36     print("Fugtighed: {1:,.2f}" .format(*self.data_dict['fugtighed']))
37     print("Trykhed: {1:,.2f}" .format(*self.data_dict['trykhed']))
38
```

Data exportering

Her skrives dataen i csv filer, filen er gemt i en undermappe med navnet "data" og filen er angivet som en csv fil, der skal åbnes i append mode, for at kunne tilføje data til det som allerede er skrevet i filen, hvis der skulle stoppes for loggeren og startes igen.

```
39 #Skriver dataen i csv filer
40 #filen åbnes i append mode så den kan tilføje data til det data der allerede er der
41 #hvis den skulle lukkes og startes op igen
42 def log_data(self):
43     for file, data in self.data_dict.items():
44         with open('data/' + file + '.csv', 'a+', newline='') as f:
45             writer = csv.writer(f)
46             writer.writerow(data)
47
```

Loop

Et simpelt loop som kører til den stoppes. Looped læser dataen, skriver den ud i konsol vinduet, og i en fil hvert 5 sekund, hvilket kan justeres i sleep funktionen.

```
47
48 #Køre et loop som læser data, og skriver det på konsol og i fil hvert 5 sekund
49 def main():
50     while True:
51         logger = Logger()
52         logger.collect_data()
53         logger.log_data()
54         logger.print_data()
55         sleep(5)
56
```

Start

Her bliver programmet bedt om at køre, eftersom det hele sat op i looped "def main".

```
57 #Kør main aka alt eftersom det hele er i min def main
58 main()
```

Konsol vinduet

```
Shell x
~ 17-04-2020, 09:43:01 ~
Temperatur: 32.16
Fugtighed: 24.12
Trykhd: 1,017.54
-----
~ 17-04-2020, 09:43:06 ~
Temperatur: 32.13
Fugtighed: 24.61
Trykhd: 1,017.52
```

Temperatur csv filen

```
temperatur.csv - Mousepad
File Edit Search View Document Help
2020-04-17 09:38:57.588860,32.164207458496094
2020-04-17 09:39:02.608531,32.19987487792969
2020-04-17 09:39:07.626616,32.306884765625
2020-04-17 09:43:01.681214,32.164207458496094
2020-04-17 09:43:06.706559,32.128536224365234
```

Fugt csv filen

```
fugtighed.csv - Mousepad
File Edit Search View Document Help
2020-04-17 09:38:57.595439,23.09792137145996
2020-04-17 09:39:02.611106,23.708942413330078
2020-04-17 09:39:07.628220,23.808490753173828
2020-04-17 09:43:01.687917,24.120866775512695
2020-04-17 09:43:06.708824,24.611743927001953
```

Tryk csv filen

