

RASPBERRY PI 4: DATALOGGER

PRACOVNÝ LIST

SKÚMANIE

Úloha 1 Zapojte **Raspberry Pi**.

- a) Vytvorte a otestujte program **monitoring_tploty.py**, ktorý bude zistiť teplotu CPU na doske Raspberry Pi:

```
from gpiozero import CPUTemperature
cpu = CPUTemperature()
print(cpu.temperature)
```

- b) Dopĺňte program, aby vypisoval teplotu vždy s odstupom 2 sekúnd – budete potrebovať importovať z modulu **time** inštrukciu **sleep**.

POZNÁMKA: Použite nekonečný cyklus, ktorý môžete v Raspbiane kedykoľvek zrušiť stlačením CTRL+C.

Riešenie: Ukážka riešenia je v súbore **monitoring_tploty_riesenie.py**.

ROZPRACOVANIE

Úloha 2

- a) Otvorte pracovný súbor **vlhkost_tplota.py** a preštudujte si jeho zdrojový kód, v ktorom je použitý modul **Adafruit_DHT** na obsluhu senzora DHT11, ktorý je pripojený cez pin GPIO21. Spustite program. Upravte program tak, aby dáta neustále každé dve sekundy načítaval a ukladal do súboru **dataVT.csv**.
- b) Experimentujte so senzorom – vyskúšajte naň dýchnuť alebo ho chytiť do dlane a sledujte zmeny meraných veličín.
- c) Dopĺňte do programu vykresľovanie grafu vlhkosti.

Riešenie: Ukážka riešenia je v súbore **vlhkost_tplota_riesenie.py**.

Úloha 3

Spustite emulátor senzorovej dosky **SenseHAT** (nájdete ho medzi programátorskými nástrojmi v Raspbiane), ktorý predstavuje virtuálny model dosky **SenseHAT**, ktorá sa experimentálne používa na medzinárodnej kozmickej stanici ISS na snímanie dát. Preskúmajte nasledujúci zdrojový kód, vytvorte v **Pythone** vlastný súbor **astropi.py**, spustite ho a otestujte použitie emulátora na zmenu hodnoty tlaku:

```
from sense_emu import SenseHat
sense = SenseHat()
sense.clear()
tlak = sense.get_pressure()
print(tlak)
```

Dopĺňte program, aby sa tlak načítaval každých 5 sekúnd a vykresľoval aj jeho priebeh. Spustite program a použitím emulátora simulujte a sledujte zmenu tlaku.

Riešenie: Ukážka riešenia je v súbore **astropi_riesenie.py**.

Úloha 4 Upravte svoj program z predošlej úlohy tak, aby načítaval aj hodnoty vlhkosti (pomocou metódy **get_humidity()**) a teploty (pomocou metódy **get_temperature()**). Program nebude poskytovať grafický výstup, len výpis nameraných hodnôt do konzoly. Program uložte do súboru **astropi2.py**. Spustite program a použitím emulátora simulujte zmeny sledovaných veličín.

Riešenie: Ukážka riešenia je v súbore **astropi2_riesenie.py**.

Úloha 5 Podľa dokumentácie sú na vesmírnej stanici ISS normálne hodnoty sledovaných veličín:

teplota: 18.3 až 26.7°C

tlak: 979 až 1027mBar

vlhkosť: okolo 60%

Doplňte svoj program z predošlej úlohy tak, aby okrem načítavania a v aj testoval, či sú v norme a vypísal výsledky do konzoly (v norme/odchýlka od normy). Namerané dáta uložte do súboru **astro.csv**. Spustite program a použitím emulátora simulujte zmeny sledovaných veličín a overte ich zápis do dátového súboru.

Riešenie: Ukážka riešenia je v súbore **astropi3_riesenie.py**.

HODNOTENIE

Sebahodnotiaci test

- Ktoré moduly/knižnice a nové inštrukcie z nich ste použili na prácu so senzormi a vykresľovanie dát? **Použili sme CPUTemperature z modulu gpiozero, time z modulu time, pyplot z modulu matplotlib, read_retry z modulu Adafruit_DHT a SenseHat z modulu sense_emu.**
- Pomocou ktorých dvoch inštrukcií ste realizovali zápis dát do súboru? **Na zápis do sboru používame inštrukcie with open() a write()**
- Ktoré štyri grafické metódy boli použité na kreslenie interaktívneho grafu? **ion() clf() plot() draw()**
- Aký je význam parametrov použitých pri načítavaní hodnôt zo senzora DHT11 pomocou inštrukcie **read_retry()**? **Parametre predstavujú druh senzora (DHT11) a pin, na ktorý je pripojený k Raspberry Pi.**
- Akými inštrukciami/metódami je možné získať dáta zo sensorovej dosky **SenseHAT**? **get_pressure(), get_temperature() a get_humidity()**