

# RASPBERRY PI 4: DATALOGGER

## Úvod

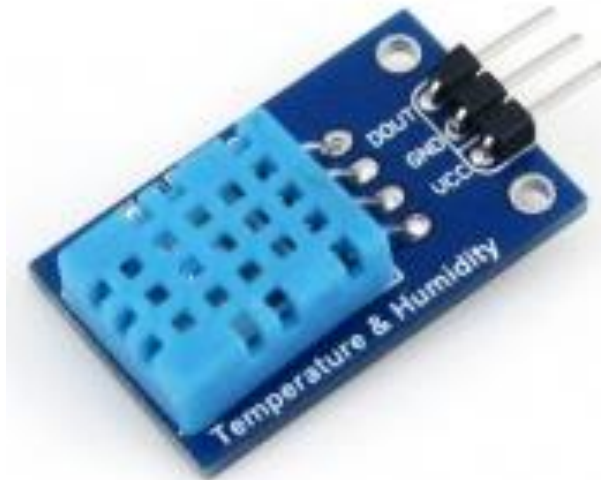
Toto je štvrtá metodika zo série 6 metódik, ktoré sú venované problematike mikropočítača **Raspberry Pi**. Tematicky sa venuje využitiu **Raspberry Pi** na využitie získavania dát zo senzorických vstupov, ich ukladanie do súborov a zobrazovanie grafických priebehov.

V rámci metodiky sa žiaci oboznámia s niekoľkými novými knižnicami pre prácu v Pythone na správu GPIO pinov a používaných senzorov. Modul **gpiozero** je určený pre jednoduchšiu prácu s GPIO pinmi a ďalšími hardvérovými súčasťami **Raspberry Pi**. V tejto metodike bude využívaná inštrukcia **CPUTemperature**, ktorá slúži na získavanie aktuálnej teploty CPU na doske **Raspberry Pi**.

Na vykresľovanie grafických priebehov bude používaný modul **matplotlib**, ktorý je potrebné vopred nainštalovať v príkazovom/konzolovom režime príkazom:

```
sudo apt-get install python3-matplotlib
```

Na ukážku využitia externých senzorov budeme používať **senzor vlhkosti a teploty DHT11** s kalibrovaným digitálnym výstupom<sup>1</sup> (spolu s vhodnými prepojovacími vodičmi je súčasťou senzorickej sady Waveshare, ktoré je súčasťou IT Science Labu):



Senzor má tri piny, ktoré sa pomocou prepojovacích vodičov pripájajú k **Raspberry Pi** nasledovne:

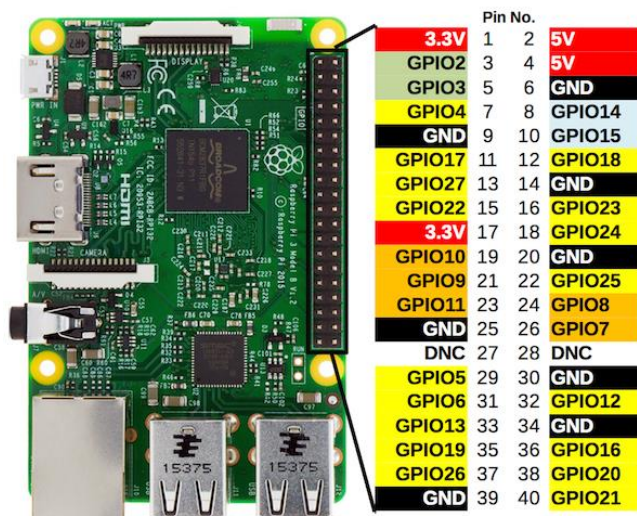
- **pin V<sub>cc</sub>** pripojíme na ktorýkoľvek napájací GPIO pin s hodnotou 3,3V alebo 5V
- **pin GND** pripojíme na ktorýkoľvek uzemňovací GPIO pin (označený **GND**)
- **pin DOUT** pripojíme na ktorýkoľvek univerzálny GPIO pin (označený **GPIOčíslo**, napr. GPIO21)

Je dôležité nepomýliť sa pri programovaní – v programe sa adresuje univerzálny GPIO pin prostredníctvom jeho identifikačného čísla, nie umiestnenia na doske, teda napr. pin **GPIO21** sa pri

---

<sup>1</sup> Bližšie informácie k senzoru je možné nájsť na stránke [https://www.waveshare.com/wiki/DHT11\\_Temperature-Humidity\\_Sensor](https://www.waveshare.com/wiki/DHT11_Temperature-Humidity_Sensor)

programovaní adresuje **číslo 21**, teda nie číslom 40, ktoré je dané jeho umiestnením – pozri obrázok s rozmiestnením a označením GPIO pinov:



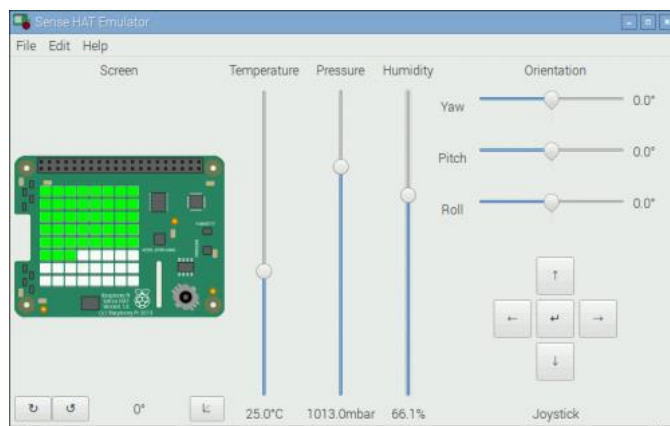
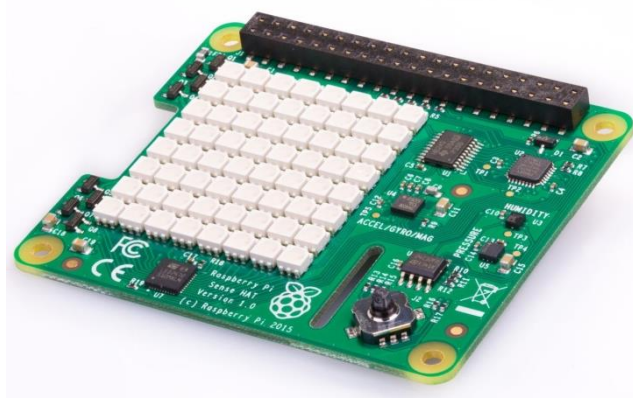
Pre zapojenie v úlohe v metodike je odporúčané jednoduché umiestnenie potrebných pinov na pripojenie spomínaného senzora – prvý okrajový pin číslo 1 (**5V**), posledný okrajový pin číslo 40 (**GPIO21**) a oproti nemu ležiaci pin číslo 39 (**GND**), ale v zmysle vyššie uvedených poznámok je možné aj iné zapojenie.

Pred prvým použitím senzora DHT11 je potrebné vykonať inštaláciu ovládacích knižníc, čo zaberie nejaký čas, preto by to malo byť zabezpečené pred tým, ako budeme realizovať výučbu. Postup pre inštaláciu v príkazovom/konzolovom režime:

1. Pokiaľ do Raspbianu ešte nebol predtým nainštalovaný Git, je nutné začať jeho inštaláciou (v opačnom prípade tento krok môžeme preskočiť):  
`sudo apt-get install git-core`
2. Stiahneme knižnicu Adafruit DHT11 pre **Python**:  
`git clone https://github.com/adafruit/Adafruit_Python_DHT.git`
3. Zmeníme adresár:  
`cd Adafruit_Python_DHT`
4. Nainštalujeme vývojové nástroje:  
`sudo apt-get install build-essential python-dev`
5. Napokon nainštalujeme samotnú knižnicu Adafruit DHT11 pre **Python3**:  
`sudo python3 setup.py install`

Metodika taktiež využíva emulátor senzorickej dosky **SenseHAT**, ktorá je súčasťou programátorských nástrojov OS Raspbian. **SenseHat** vo fyzickom prevedení je rozšírenie **Raspberry Pi**, ktoré obsahuje snímače teploty, atmosférického tlaku, vlhkosti vzduchu, akcelerometer/magnetometer/gyroskop. Okrem toho ponúka 8x8 RGB LED displej a malý joystick. Projekty stredoškôľakov určené pre **SenseHAT** sa môžu stať súčasťou medzinárodnej súťaže **Astro Pi** a realizovať snímania senzorickej dát priamo na

vesmírnej stanici ISS<sup>2</sup>. Fyzickú SenseHAT dosku je možné zakúpiť online v rôznych e-shopoch, avšak na zapojenie do programu **Astro Pi**, nie je nevyhnutná, nakoľko **SenseHAT** emulátor je plne postačujúci. Ak však škola má túto rozširujúcu dosku k **Raspberry Pi**, tak je vhodné ju žiakom v rámci hodiny ukázať. Pre programovanie s využitím emulátora bude potrebné na začiatku programu importovať modul **sense\_emu** (v prípade, ak by program mal bežať na fyzickej doske **SenseHAT**, tak miesto uvedeného modulu sa len importuje modul **sense\_hat**).



Žiaci majú k dispozícii pracovný list, ktorý obsahuje zadania úloh, miesto na žiacke riešenie a miesto pre poznámky. Odporúčame, aby učiteľ žiakom pri každej fáze vyučovania uviedol zoznam úloh z pracovného listu, ktoré budú aktuálne riešiť.

## PRIEBEH VÝUČBY

Osnova vyučovacej hodiny (podľa modelu 5E):

- **Zapojenie (5 minút)** – diskusia
- **Skúmanie (5 minút)** – práca vo dvojiciach s pracovným listom (úloha 1)
- **Vysvetlenie (10 minút)** – ukážka zápisu dát do súboru a ich vykresľovanie do grafu
- **Rozpracovanie (15 minút)** – práca vo dvojiciach, programovanie náročnejších úloh (úlohy 2 až 5 z pracovného listu)
- **Hodnotenie (5 minút)** – sebahodnotiaci test

## ZAPOJENIE (CCA 5 MIN.):

Hodinu začneme krátkou diskusiou. Žiakov vyzveme, aby uviedli čo najviac senzorov, ktoré majú pri sebe alebo sú v ich okolí – pokiaľ by ich táto otázka zmatla, môžeme ju konkretizovať – senzory sú

---

<sup>2</sup> Viac o súťaži je možné nájsť na stránke <https://astro-pi.org/>

obsiahnuté v každom mobilnom telefóne (dotykový senzor, GPS senzor, dotykový senzor, senzor natočenia, ...) alebo tablete. Žiaci môžu uviesť aj iné aplikácie, kde sa využívajú senzory. Položíme im otázku, či sa vedia k snímaným dátam nejakým spôsobom dostať, či ich už niekde videli. Ako príklad môžeme uviesť stránku Slovenského hydrometeorologického ústavu (<http://www.shmu.sk>), kde môžeme nájsť niektoré dáta získané meraním zo siete meteostaníc po celom Slovensku. Ktoré meteodáta sú zaznamenávané?

Na základe tejto diskusie predstavíme žiakom ciele hodiny:

- načítavať vstupné hodnoty zo senzorov,
- ukladať dáta do súboru,
- zobrazovať grafický priebeh dát.

## SKÚMANIE (CCA 5 MIN.):

V tejto časti hodiny žiaci pracujú vo dvojiciach s pracovnými listami (úloha 1). Nezasahujeme do práce žiakov, len ju monitorujeme, v prípade potreby im poskytneme konzultáciu alebo vysvetlíme časť, v ktorej im niečo nie je jasné.

### Úloha 1 Zapojte *Raspberry Pi*.

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

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

- b) Doplníte 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**.

## VYSVETLENIE (CCA 10 MIN.):

Na základe riešenia predošlej úlohy vyzveme žiakov, aby vyskúšali na chvíľu sa dotknúť CPU na doske **Raspberry Pi** a sledovali zmenu, ktorú tento dotyk spôsobí. Pri sledovaní snímanej hodnoty teploty na CPU sa spýtame žiakov, aký vplyv môže mať vyššia teplota (spravidla vyššia aj ako 50°C) na prevádzku počítača a ako by riešili problém s pomerne vysokou teplotou – môžeme im ukázať reálne ukážky pasívnych alebo aktívnych chladičov.

Spoločne so žiakmi si ukážeme, ako ukladať nasnímané hodnoty do súboru – pre numerické dáta môžeme použiť formát súboru **.csv** (ale postup je možné aplikovať aj na vytváranie textových súborov vo formáte **.txt**). Použijeme inštrukciu **with open()** na otvorenie súboru, ktorý bude v premennej **zaznam**, napr.:

```
with open("teplota.csv", "a") as zaznam:
```

Parametre inštrukcie **with open()** sú dva – meno súboru, s ktorým budeme pracovať a režim pre otvorenie tohto súboru ("a" predstavuje režim zapisovania s pridávaním na koniec predošlého záznamu, t.j. z anglického slova *append*).

Na samotný zápis do súboru potom postačí inštrukcia:

```
zaznam.write(str(hodnota)+"\n")
```

Žiakov upozorníme, že číselné dáta je potrebné pred ukladaním prekonvertovať na reťazce (príkazom **str()**). Na zápis do súboru si vytvoríme novú funkciu:

```
def zapis_hodnoty(hodnota):  
    with open("teplota.csv", "a") as zaznam:  
        zaznam.write(str(hodnota)+"\n")
```

Celý upravený zdrojový kód bude vyzeráť (je dostupný aj v súbore **monitoring\_teploty2\_riesenie.py**):

```
from gpiozero import CPUTemperature  
from time import sleep  
  
def zapis_hodnoty(hodnota):  
    with open("teplota.csv", "a") as zaznam:  
        zaznam.write(str(hodnota)+"\n")  
  
while True:  
    cpu = CPUTemperature()  
    print(cpu.temperature)  
    zapis_hodnoty(cpu.temperature)  
    sleep(2)
```

Necháme žiakov spustiť program, následne ho ukončiť, vyhľadať a otvoriť súbor **teplota.csv**. Všimnime si, že tento súbor predtým nebol vytvorený a teraz existuje – pre potreby ukladania dát bol tento súbor novovytvorený a nachádza sa v tom istom priečinku ako náš **.py** súbor. Dátový súbor môžeme uzavrieť a vyskúšame program na zisťovanie teploty spustiť nanovo. Skontrolujeme dáta v súbore **teplota.csv**, ktorý už tentokrát existoval pred spustením programu. Vidíme, že dáta sú v ňom aj teraz, ale predošlé dáta boli doplnené novými dátami.

Python ponúka aj možnosť vykresľovať grafický priebeh snímaných dát pomocou knižnice **matplotlib** inštrukciou **pyplot** – by sme nemuseli neustále vypisovať **matplotlib.pyplot** naimportujeme si ho ako **kresli**, teda:

```
import matplotlib.pyplot as kresli
```

Pri vykresľovaní budeme používať premenné **data\_x** a **data\_y**, čo budú zoznamy nameraných hodnôt, ktoré budeme vykresľovať na príslušných osiach. Najprv si spustíme interaktívny režim vykresľovania grafu a vytvoríme prázdne zoznamy **data\_x** a **data\_y**, do ktorých budeme potom dáta ukladať:

```
kresli.ion() #interaktívny režim vykresľovania grafu počas načítavania dát  
data_x = []  
data_y = []
```

Pre vykresľovanie grafu vytvoríme novú funkciu:

Hodnoty teploty budú na osi y a čas (získame inštrukciou **time()**, ktorú však musíme doimportovať z modulu **time**) bude zobrazovaný na osi x – výsledný zdrojový kód bude vyzeráť takto (je dostupný aj v súbore **monitoring\_teploty3\_riesenie.py**):

```
from gpiozero import CPUTemperature
from time import sleep, time
import matplotlib.pyplot as kresli

def zapis_hodnoty(hodnota):
    with open("teplota.csv", "a") as zaznam:
        zaznam.write(str(hodnota)+"\n")

def graf(hodnota):
    data_y.append(hodnota)
    data_x.append(time())
    kresli.clf() #vymazanie predošlého grafu
    kresli.plot(data_x,data_y) #vytvorenie grafu s dátami pre dané osi
    kresli.draw() #vykreslenie/zobrazenie grafu
    kresli.pause(0.01)

kresli.ion() #interaktívny režim vykresľovania grafu počas načítavania dát
data_x = []
data_y = []
while True:
    cpu = CPUTemperature()
    print(cpu.temperature)
    zapis_hodnoty(cpu.temperature)
    graf(cpu.temperature)
    sleep(2)
```

## ROZPRACOVANIE (CCA 15 MIN.):

V tejto časti hodiny žiaci riešia samostatne úlohy 2 až 5 z pracovného listu. Svoje programy si ukladajú do samostatných súborov vo svojom pracovnom priečinku. Sledujeme individuálnu prácu žiakov, podporujeme a povzbudzujeme ich. Pred úlohou 2 rozdáme žiakom senzory DHT11 aj s prepojovacími vodičmi a inštruujeme ich ako pripojiť senzory k **Raspberry Pi** (bolo vysvetlené v úvodnom komentári k metodike):

- **pin Vcc** pripojíme na napájací GPIO pin číslo 1 (**5V**) – t.j. prvý okrajový pin,
- **pin DOUT** pripojíme na univerzálny GPIO pin číslo 40 (označený **GPIO21**) – t.j. posledný okrajový pin,
- **pin GND** pripojíme na uzemňovací GPIO pin číslo 39 (označený **GND**) – t.j. posledný okrajový pin oproti pinu číslo 40.

### Úloha 2

- Otvorte pracovný súbor **vlhkost\_teplo.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**.
- Experimentujte so senzorom – vyskúšajte naň dýchnuť alebo ho chytiť do dlane a sledujte zmeny meraných veličín.
- Doplňte do programu vykresľovanie grafu vlhkosti.

**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)
```

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

**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.

**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.

## HODNOTENIE (CCA 5 MIN.):

Žiaci riešia samostatne sebahodnotiaci test z pracovného listu, zameraný na porozumenie použitým príkazom a postupom a sumarizáciu poznatkov. Pri jeho riešení môžu použiť všetky svoje programové kódy z hodiny. Na záver s nimi zhrnieme riešenia.



### Sebahodnotiaci test

- a) Ktoré moduly/knižnice a nové inštrukcie z nich ste použili na prácu so senzormi a vykresľovanie dát?
- b) Pomocou ktorých dvoch inštrukcií ste realizovali zápis dát do súboru?
- c) Ktoré štyri grafické metódy boli použité na kreslenie interaktívneho grafu?
- d) Aký je význam parametrov použitých pri načítavaní hodnôt zo senzora DHT11 pomocou inštrukcie **read\_retry()**?
- e) Akými inštrukciami/metódami je možné získať dáta zo sensorovej dosky **SenseHAT**?