MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SLOVENSKEJ REPUBLIKY
EURÓPSKA ÚNIA
Európsky sociálny fond
Európsky fond regionálneho rozvoja
OPERAČNÝ PROGRAM ĽUDSKÉ ZDROJE

Písomný výstup pedagogického klubu

1. Prioritná os	Vzdelávanie
2. Špecifický cieľ	1.1.1 Zvýšiť inkluzívnosť a rovnaký prístup ku kvalitnému vzdelávaniu a zlepšiť výsledky a kompetencie detí a žiakov
3. Prijímateľ	Stredná priemyselná škola strojnícka, Duklianska 1, Prešov
4. Názov projektu	Učitelia SPŠ strojníckej v Prešove inovujú
5. Kód projektu ITMS2014+	312011ADH9
6. Názov pedagogického klubu	Pedagogický klub elektroniky a informatiky
7. Meno koordinátora pedagogického klubu	Ing. Pavol Pavlanin
8. Školský polrok	február 2022 – jún 2022
9. Odkaz na webové sídlo zverejnenia písomného výstupu	https://spspo.edupage.org/a/projekt

Úvod:

Stručná anotácia

Výstup pedagogického klubu

Výstup sme zamerali na konkrétne zadania, ktoré je možné použiť na motiváciu žiakov učiť sa programovať alebo pripravené ukážky použiť na zatraktívnenie vyučovacieho procesu. Pripravili sme súťaž z programovania, s cieľom rozvíjať ich logické myslenie a zároveň zvýšiť záujem žiakov o programovanie. Súťaž pozostávala z teoretickej a praktickej časti. V rámci teoretickej časti sme vytvorili online test, ktorý žiaci riešili v prostredí EduPage. V praktickej časti žiaci riešili konkrétne dve zadania, ktorých výstupom mal byť funkčný program v jazyku Python.

V oblasti výmeny skúsenosti medzi jednotlivými členmi, sme priblížili vizualizáciu dát získaných z rôznych komponentov, prostredníctvom protokolu MQTT a vývojového prostredia Node-RED.

Kľúčové slová

súťaž, e-test, vývojový diagram, program, Python, animácia, MQTT, Node-RED, vizualizácia, broker, TinkerCad

Zámer a priblíženie témy písomného výstupu

Cieľom pedagogického klubu je zvýšiť vedomosti a zručnosti žiakov v oblasti elektroniky a informatiky.

Našim zámerom bolo aj inšpirovať ostatných učiteľov k realizácii súťaží a ponúknuť konkrétne ukážky obsahu súťaže zameranej na programovanie. V rámci inšpirácie sme pripravili možnosti vizualizácie dát získaných z rôznych komponentov prostredníctvom protokolu MQTT a vývojového prostredia Node-RED. Ponúkame aj rozpracovanú konkrétnu úlohu, ktorú je možné využiť na zatraktívnenie vyučovacieho procesu.

Jadro:
Popis témy/problém

- ***Súťaž v programovaní***

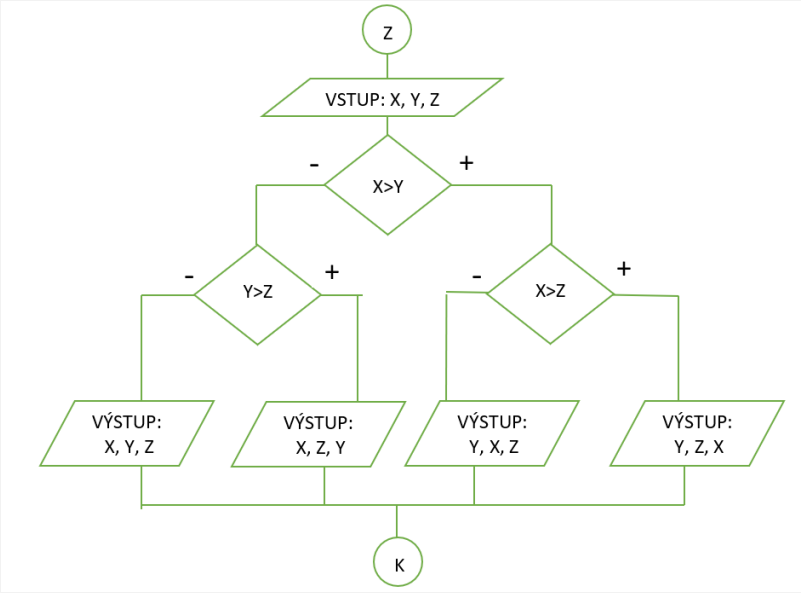
Zadania úloh súťaže ponúkame učiteľom ako ich motiváciu, organizovať rôzne súťaže a zapájať žiakov aj do mimoškolských aktivít. Možnosť zúčastniť sa súťaže žiaci prijali mimoriadne pozitívne. Súťaž bola realizovaná prezenčnou formou a pozostávala z teoretickej a praktickej časti. V teoretickej časti žiaci riešili e-test v EduPage, ktorý obsahoval úlohy zamerané na pochopenie vývojových diagramov, syntaxe jazyka Python a pochopenie rôznych logaritmických konštrukcií programovania. V praktickej časti žiaci riešili dve zadania na vytvorenie programu v Pythone pomocou modulu Tkinter.

Súťaž v programovaní - teoretická časť

E-test – vývojové diagramy, algoritmizácia, programovanie

1. Vstupom zadaného vývojového diagramu (VD) sú tri celé čísla. Podľa VD zistí, aký bude výstup pre jednotlivé vstupy.

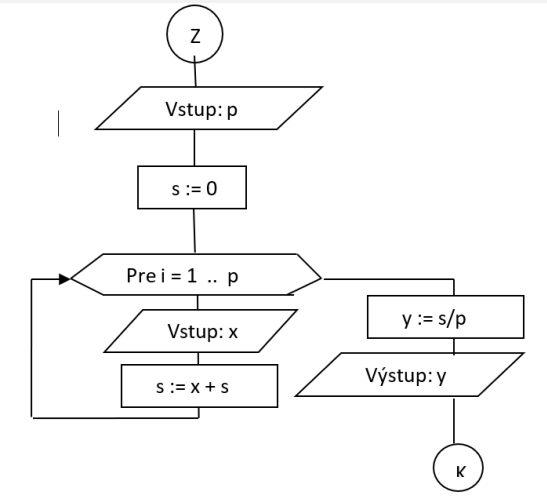
   a. VSTUP: 5; 25; 15 — VÝSTUP: SPRÁVNA ODPOVEĎ: **5; 15; 25**
   b. VSTUP: 10; 8; 3 — VÝSTUP: SPRÁVNA ODPOVEĎ: **8; 3; 10**
   c. VSTUP: 30; 20; 40 — VÝSTUP: SPRÁVNA ODPOVEĎ: **20; 30; 40**
   d. VSTUP: 18; 9; 15 — VÝSTUP: SPRÁVNA ODPOVEĎ: **9; 18; 15**



2. Medzi základné grafické znaky pre zápis algoritmu vo vývojovom diagrame patria:

a) kosodĺžnik - vstupné údaje — SPRÁVNA ODPOVEĎ
b) obdĺžnik - výstupné údaje
c) kosoštvorec – podmienka — SPRÁVNA ODPOVEĎ
d) kružnica – cyklus
e) obdĺžnik – príkazy priradenia — SPRÁVNA ODPOVEĎ

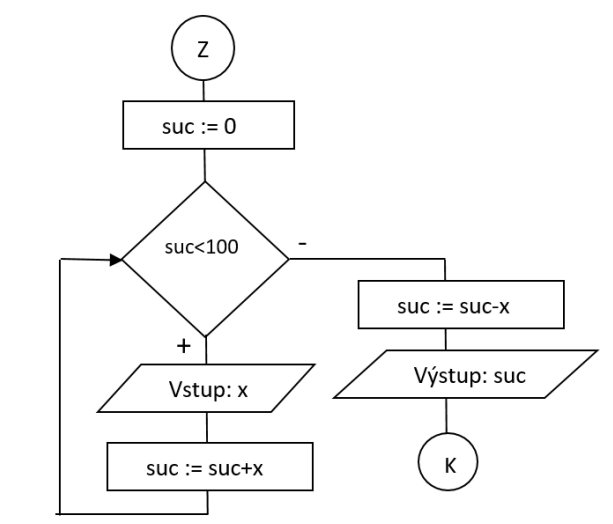
3. Napíšte, aký bude výstup zadaného VD, ak na vstupe budú postupne načítavané tieto čísla: 5; 2; 15; 10; 5; 3.



VAŠE VYPRACOVANIE:

SPRÁVNA ODPOVEĎ: 7

4. Označte možnosť, ktorá uvádza riešenie problému, ktorý počíta uvedený VD:



a) VD spočítava zadávané čísla, pokiaľ je súčet čísel menší ako 100, VD vypíše posledné zadané číslo.
b) VD načítava zadávané čísla, ak zadáme väčšie číslo ako 100, tak VD ho vypíše.
c) VD spočítava zadávané čísla, pokiaľ je súčet čísel menší ako 100, VD vypíše najvyšší súčet, ktorý neprekročí číslo 100. SPRÁVNA ODPOVEĎ
d) VD načítava zadávané čísla, výstupom bude súčet čísel, ktorý bude väčší ako 100.

5. Vyberte správny výpis nasledujúceho cyklu:

```
for i in range(1,50,5):
    print(i)
```

a) 1, 10, 20, 30, 40
b) 1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51
c) 1, 6, 11, 16, 21, 26, 31, 36, 41, 46 SPRÁVNA ODPOVEĎ
d) 1, 6, 11, 16, 21

6. Vyberte správny výpis nasledujúceho cyklu:

```
for i in range(5):
    print(i)
print('Python Cup')
```

a) 1, 2, 3, 4, 5, Python Cup
b) 1, 2, 3, 4, Python Cup
c) 1, Python Cup, 2, Python Cup, 3, Python Cup, 4, Python Cup, 5, Python Cup
d) 0, 1, 2, 3, 4, Python Cup SPRÁVNA ODPOVEĎ

7. Vyberte správny výpis nasledujúcej podmienky, ak vstupom bude x=50 a y=100:

```
if 0<x<50 and 50<y<150:
    print(x+y)
else
    print(x-y)
```

a) 50
b) 100
c) -50 SPRÁVNA ODPOVEĎ
d) 150

8. Vyberte správny výpis nasledujúcej podmienky, ak vstupom bude x=50 a y=100:

```
if 0<x<50 or 50<y<150:
    print(x+y)
else
    print(x-y)
```

a) 50
b) 100
c) -50
d) 150 SPRÁVNA ODPOVEĎ

9. Označte príkaz, ktorý vykreslí modrý štvorec so stranou dlhou 100b v strede grafického plátna s veľkosťou 500x500.

a) canvas.create_rectangle(200,200,300,300, fill='blue') SPRÁVNA ODPOVEĎ

b) canvas.create_squere(200,200,300,300, fill='blue')

c) canvas.create_squere(100,200,100,200, fill='blue')

d) canvas.create_rectangle(200,300,200,300, fill='blue')

10. Označte správny zápis na vloženie tlačidla do grafického plátna.

a) def button1():
príkazy

button1 = tkinter.Button(text = ‘text na tlačidle‘, command = button1_klik)
button1.pack

b) def button1_klik():
príkazy

button1 = tkinter.button(text = ‘text na tlačidle‘, command = button1_klik)
button1.pack

c) def button1_klik():
príkazy

button1 = tkinter.Button(text = ‘text na tlačidle‘, command = button1_klik)
button1.pack() SPRÁVNA ODPOVEĎ

d) def button1():
príkazy

button1 = tkinter.button(text = ‘text na tlačidle‘, command = button1_klik)
button1.pack()

11. Označte príkaz, ktorý vytvorí text „Súťaž v Pythone“ a umiestni ho v strede grafického plátna 300x500, font nastavte Arial, veľkosť 40.

a) canvas.create_text(100,200,text='Súťaž v Pythone', font='arial 40')

b) canvas.create_text(150,250, text='Súťaž v Pythone',font='Arial 40') SPRÁVNA ODPOVEĎ

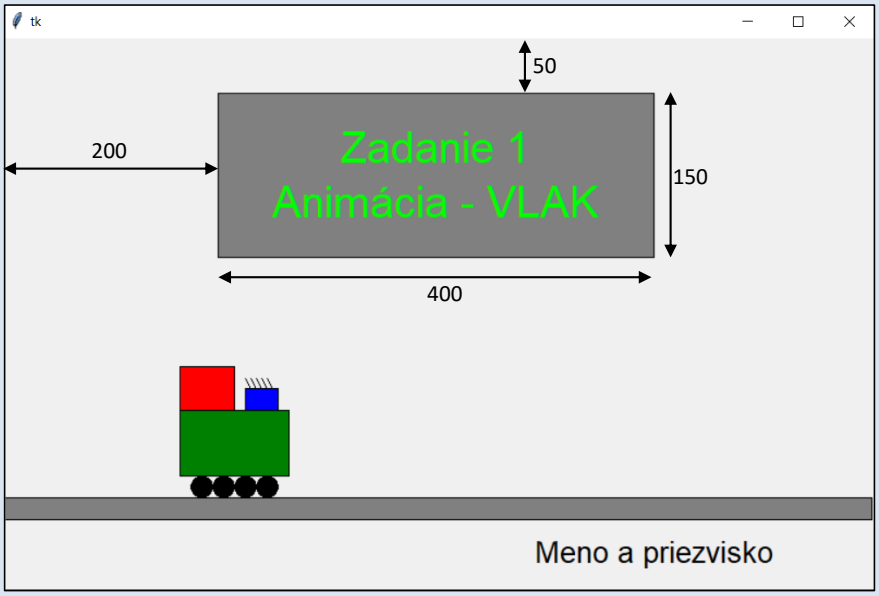
c) canvas.create_text(150,250, 'Súťaž v Pythone',Arial, 40)

d) canvas.create_text(100,200,200,300,text = 'Súťaž v Pythone', font= Arial, size=40)

Súťaž v programovaní – praktická časť

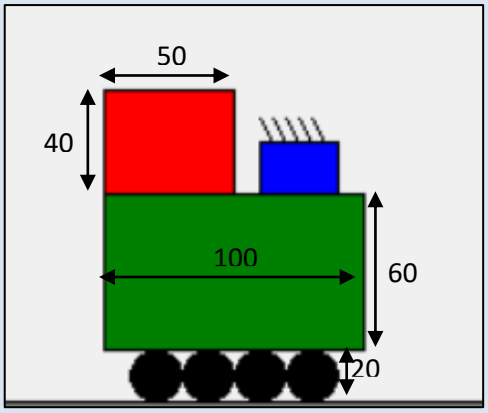
ZADANIE 1:

Vytvorte program Animácia vlaku v jazyku Python podľa zadaného obrázka:



Grafická stránka programu (všetko sa zobrazí po spustení programu):

- grafické plátno: veľkosť 800x500 px,
- texty „Zadanie 1“ a „Animácia – VLAK“: umiestnenie podľa obrázka do obdĺžnika veľkosti 400x150. Font textu nastavte: Arial 30, farba „lime“,
- text „Meno a priezvisko“ umiestnenie podľa zobrazeného obrázka a nastavte font písma na Arial 20. Meno a priezvisko uveďte svoje,
- vlak nakreslíte podľa nasledujúceho obrázku:



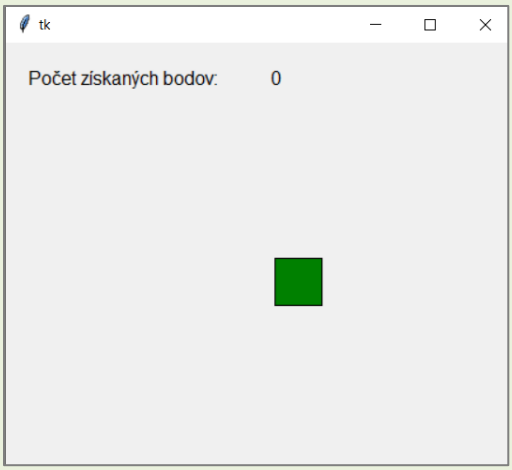
Popis fungovania programu:

- po spustení programu sa vlak zobrazí vľavo v grafickom plátne a pomaly sa bude posúvať smerom doprava,
- na posúvanie vlaku použite príkaz move a časovač,
- na označenie jednotlivých grafických príkazov na nakreslenie vlaku môžete použiť parameter tags.

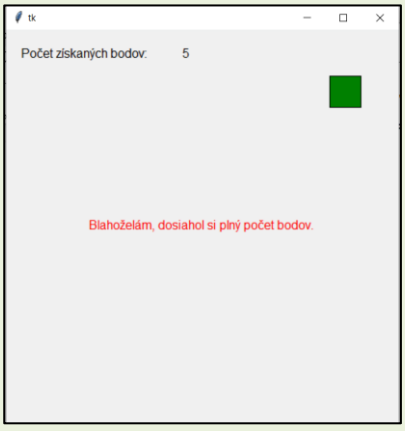
ZADANIE 2:

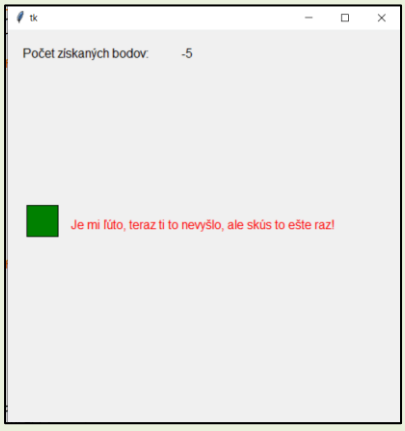
Vytvorte program Klikaj v jazyku Python podľa zadaného obrázka:

Vytvorte nasledujúci program:



- po štarte programu, sa na náhodnom mieste plochy bude každú sekundu vykresľovať stále jeden štvorec o veľkosti 40b a bude zelenej farby,
- úlohou hráča je kliknúť na náhodne vykresľovaný štvorec,
- ak hráč klikne na štvorec, pripočíta sa mu bod, ak klikne mimo plochy štvorca, bod sa odpočíta. Počet bodov sa vykresľuje v hornej časti grafickej plochy (viď. obrázok),
- ak hráč získa 5 bodov, vypíše sa text: „Blahoželám, dosiahol si plný počet bodov.“ a vykresľovanie sa zastaví (pozri obr.),
- ak hráč získa -5 bodov, vypíše sa text: „Je mi ľúto, teraz ti to nevyšlo, ale skús to ešte raz!“ a vykresľovanie sa zastaví (pozri obr.).





- **Zhodnotenie súťaže v programovaní**

Jednotlivé kolá súťaže sa konali v mesiaci február. Súťaže sa zúčastnilo 12 žiakov 3. ročníka odboru strojárstvo. Súťaž pozostávala z dvoch častí – teoretického testu a praktickej časti, ktorá pozostávala z dvoch zadaní. V každom zadaní mali žiaci vytvoriť program v Pythone.

- **Teoretická časť:**

číslo úlohy	**zameranie úlohy**	**úspešnosť**
1,2,3,4	vývojové diagramy	63,38%
5,6,7,8	logaritmické konštrukcie	72,05%
9,10,11	syntax jazyka Python	86,61%

Online test žiaci riešili v prostredí EduPage. Čas na riešenie bol vyhradený na 30 min, všetci žiaci stihli vyriešiť všetky úlohy v zadanom čase.

Jednotlivé úlohy sme rozdelili do 3 oblastí: logaritmické konštrukcie, vývojové diagramy a syntax jazyka Python. Najslabšie výsledky dosiahli z oblasti vývojových diagramov. Oblasť vývojových diagramov bola pre žiakov náročná aj počas vyučovacích hodín, chýbala im predstavivosť a samostatnosť pri tvorbe vývojových diagramov. Naopak, najlepšie výsledky boli z oblasti syntaxe jazyka Python.

Priemerná úspešnosť žiakov bola **74 %.**

- **Praktická časť:**

Pri riešení obidvoch zadaní zvládli žiaci grafickú stránku programu, aj keď v prvom zadaní neboli niektorí dôslední. Najväčší problém mali pri vytvorení animácie vlaku, a to niektorí žiaci ani nezvládli naprogramovať.

Žiaci lepšie zvládli druhé zadanie, niektorí nevedeli zastaviť vykresľovanie štvorca aj po dosiahnutí maximálneho alebo minimálneho počtu bodov.

Priemerná úspešnosť žiakov bola **88 %.**

- **Celkové výsledky:**

Súťažiaci dosiahli priemernú úspešnosť **81%.**

Najúspešnejší riešitelia dosiahli celkovú úspešnosť: 94%, 89% a 88%.

Výmena skúseností medzi jednotlivými členmi pedagogického klubu

Vizualizácia dát pomocou protokolu MQTT a prostredia Node-RED

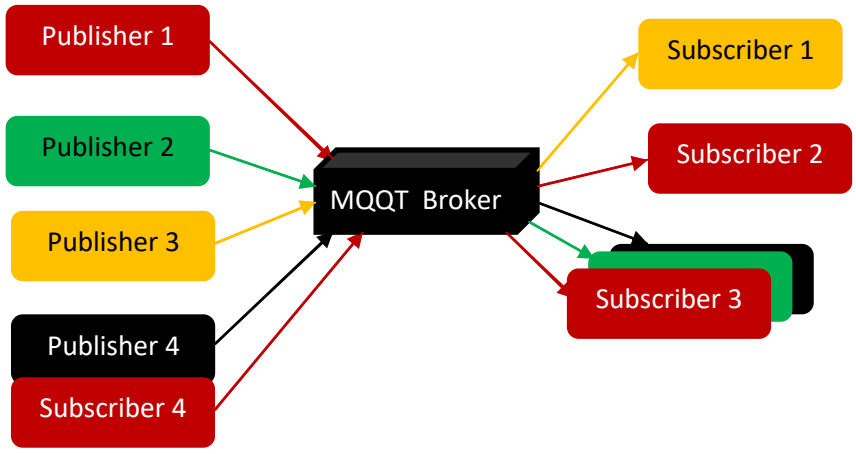
Protokol MQTT

- **MQTT** – **M**essage **Q**ueue **T**elemetry **T**ransport (Telemetrická preprava vo fronte správ)– MQTT je komunikačný protokol založený na systéme publikovania krátkych **správ** a prihlásenia odber **tém**. MQTT je jednoduchý na používanie a je vynikajúci pre projekty internetu vecí.

Základné pojmy MQTT:

- **Publisher** – poskytovateľ, ktorý odosiela údaje a nestará sa o to, kto ich prijíma.
- **Subscriber** – odberateľ, ktorý odoberá dáta, ku ktorým sa prihlásil.
- **Broker** – softvér, ktorý sa stará o sprostredkovanie predávania dát medzi publisherom a subscribrom.
- Správy sa skladajú z tzv. **topicu** (téma) a **payloadu** (hodnota, ktorá sa posiela).
- Na prenos správ sa využíva protokol TCP. Na otvorenú, nešifrovanú komunikáciu sa najčastejšie používa port 1883. Na šifrovanú komunikáciu sa používajú porty 8883 a 8884.
- **QoS** (Quality of Service) – nástroj, ktorý nám umožňuje nastaviť potvrdzovanie správ na nasledujúcich úrovniach:
  - *Level 0 – fire and forget* – správy sa nepotvrdzujú, takže sa nedá zaručiť ich doručenie.
  - *Level 1 – at least once* – zabezpečí, že správa bude doručená aspoň raz.
  - *Level 2 – exactly once* – zabezpečí, že správa bude doručená len raz.

Medzi poskytovateľmi a odberateľmi správ je vzťah many-to-many, čiže poskytovateľ môže poskytovať správy pre viac odberateľov a odberateľ môže odoberať správy od viacerých poskytovateľov. Jedno zariadenie môže byť zároveň aj poskytovateľ aj odberateľ.



Zápis správ:

- Topic je hierarchicky organizovaný podobne ako adresáre, napr.
  - hala1/dielna/tlacitka/tlacitko01/pocet_stlaceni:3

Topic

Payload

Tému si zvolí poskytovateľ (subscriber). Témy sa nemusia vopred konfigurovať, ak MQTT broker príjme od niektorého poskytovateľa správu s novou témou, automaticky túto tému zaradí. Odberateľ sa k odberu správ k danej téme prihlási poslaním špeciálnej správy „subscribe“ s názvom témy. Subscriber môže pri odbere správ použiť aj nasledujúce zástupné znaky :

- \+ (plus) – nahradí jednu úroveň
- hala1/dielna/tlacitka/+/pocet_stlaceni
- # - nahradí všetky nasledujúce úrovne a musí byť posledný
- hala1/#
- $ - špeciálny topic, väčšinou od brokera
- $sys/

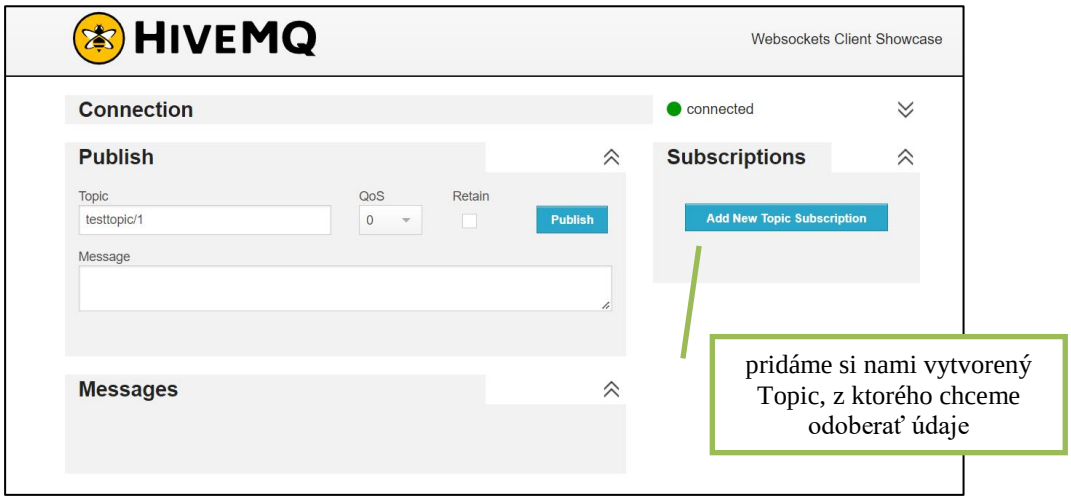
HiveMQ broker

Na odosielanie a odoberanie dát môžeme použiť ľubovoľného free online brokera, napr.:

- **HiveMQ – broker.hivemq.com**
- test.mosquitto.org
- iot.eclipse.org

Publikovanie a odoberanie dát prostredníctvom brokera HiveMQ

- link na pripojenie klienta: http://www.hivemq.com/demos/websocket-client/



Program, ktorým sa pripojíme na internet, aby sme mohli odosielať údaje cez MQTT na broker HiveMQ:

```
from machine import Pin
from umqtt.robust import MQTTClient
_ssid = 'skolenie'
_password = '123456789abc'

def do_connect(ssid, password):
    import network
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        wlan.connect(ssid, password)
        while not wlan.isconnected():
            pass
    print('network config:', wlan.ifconfig())

do_connect(_ssid, _password)
```

Definovanie klienta v programe, z ktorého odosielame údaje na broker:

- *klient = MQTTClient('meno odosielatela', 'nazov brokera')*

napr. *klient = MQTTClient('Adam123456', 'broker.hivemq.com')*

Príkaz sa nadviazanie spojenia s brokerom:

- *klient.connect()*

Príkaz na publikovanie dát:

- *klient.publish('Topic', Payloud)*

Príkaz na odpojenie klienta od brokera:

- *klient.disconnect()*

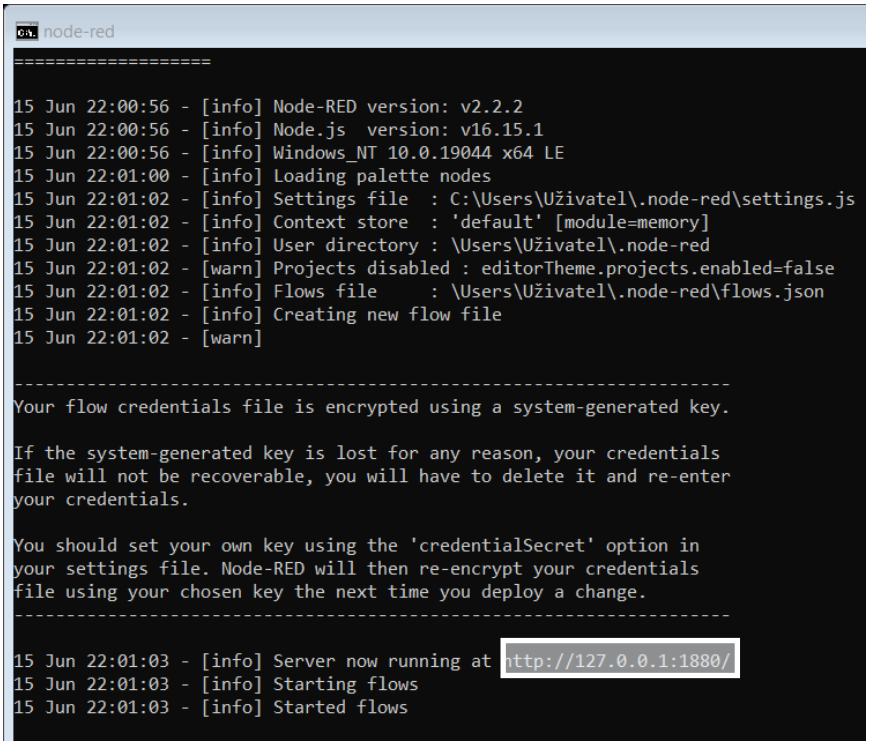
Node-RED- vizualizácia dát

- je programovací nástroj na prepojenie hardvérových zariadení, rozhraní API a online služieb novými a zaujímavými spôsobmi.
- Používa sa na vizuálne programovanie IoT riešení. Node-RED nám umožňuje vizualizovať rôzne údaje získané napr. aj z rôzneho brokera.
- Node-RED je postavený na Node.js a plne využíva jeho udalosťami riadený, neblokujúci model.
- Vstavaná knižnica vám umožňuje uložiť užitočné funkcie, šablóny alebo toky na opätovné použitie.

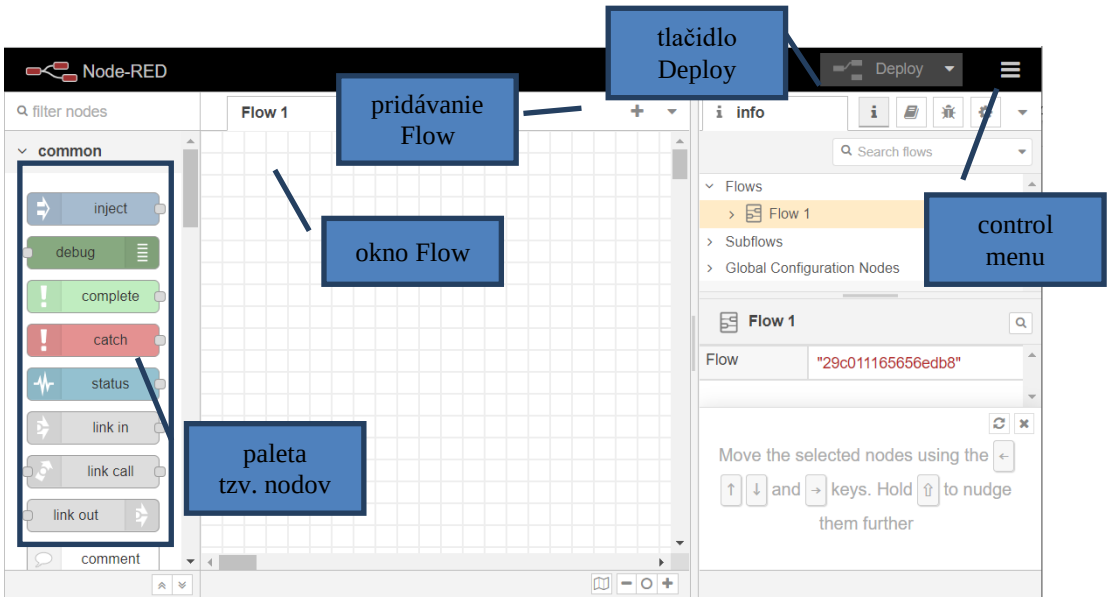
Postup inštaláciea spustenia Node-RED:

Node-RED si môžeme nainštalovať či už na lokálnom počítači, zariadení, ako je Raspberry Pi alebo v cloude. Inštalačný súbor si stiahneme z oficiálnej stránky nodered.org:
https://nodered.org/docs/getting-started/

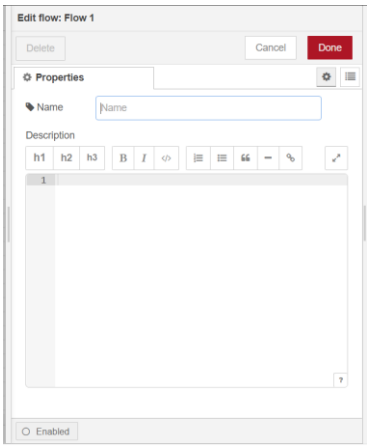
1. Spustíme inštaláciu
2. Po nainštalovaní si v príkazovom riadku spustíme Node-RED v termináli pomocou príkazu: node-red.
3. Počas celej práce v prehliadači musí byť Node-RED spustený v termináli.
4. Na zastavenie Node-RED môžete použiť Ctrl-C alebo zatvoriť okno terminálu.
5. So spusteným Node-RED otvoríme editor vo webovom prehliadači
6. Ak používame prehliadač na rovnakom počítači, na ktorom je spustený Node-RED, môžeme k nemu pristupovať pomocou adresy URL: http://localhost:1880 .
7. Ak používame prehliadač na inom počítači, budeme musieť použiť adresu IP počítača s Node-RED:



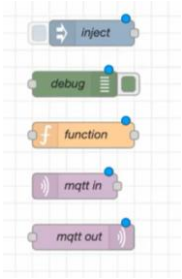
Prostredie Node-RED:



- Flow si môžeme premenovať v dialógovom okne, ktoré vyvoláme dvojklikom na názov Flow 1



Najzakladanejšie nody:



- Inject – input node
- Debug - Output node
- Function - Processing node
- MQTT in - subscribe node
- MQTT out - publish node

Node-RED dashboard:

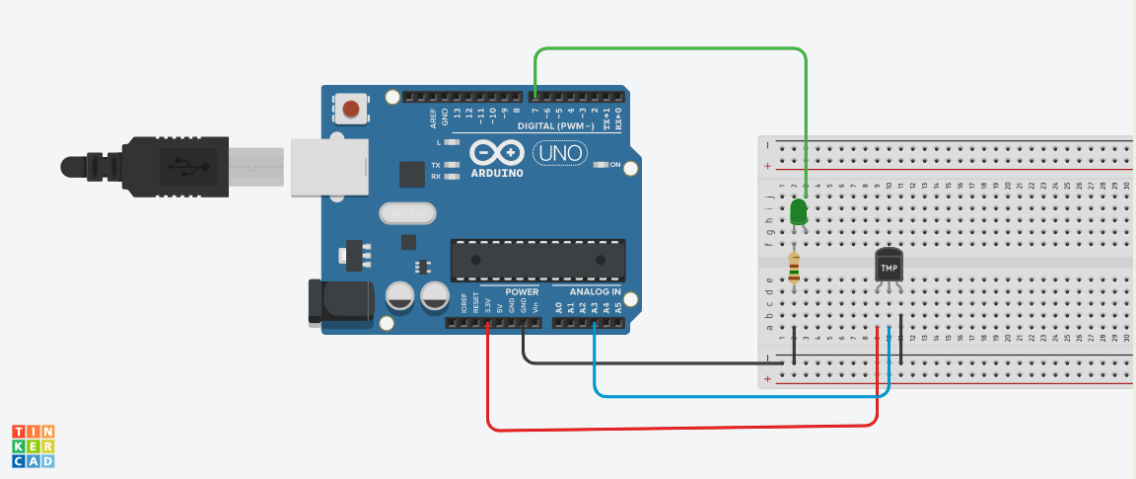
- Umožňuje vykresľovanie grafov či hodnôt
- Rýchle a jednoduché rozhranie
- Má vlastné nody

Programovanie v týchto vývojových prostrediach môžeme použiť vo všetkých vyučovacích predmetoch zameraných na programovanie, kde simulácia v prostredí ponúka žiakom konkrétne príklady z praxe.

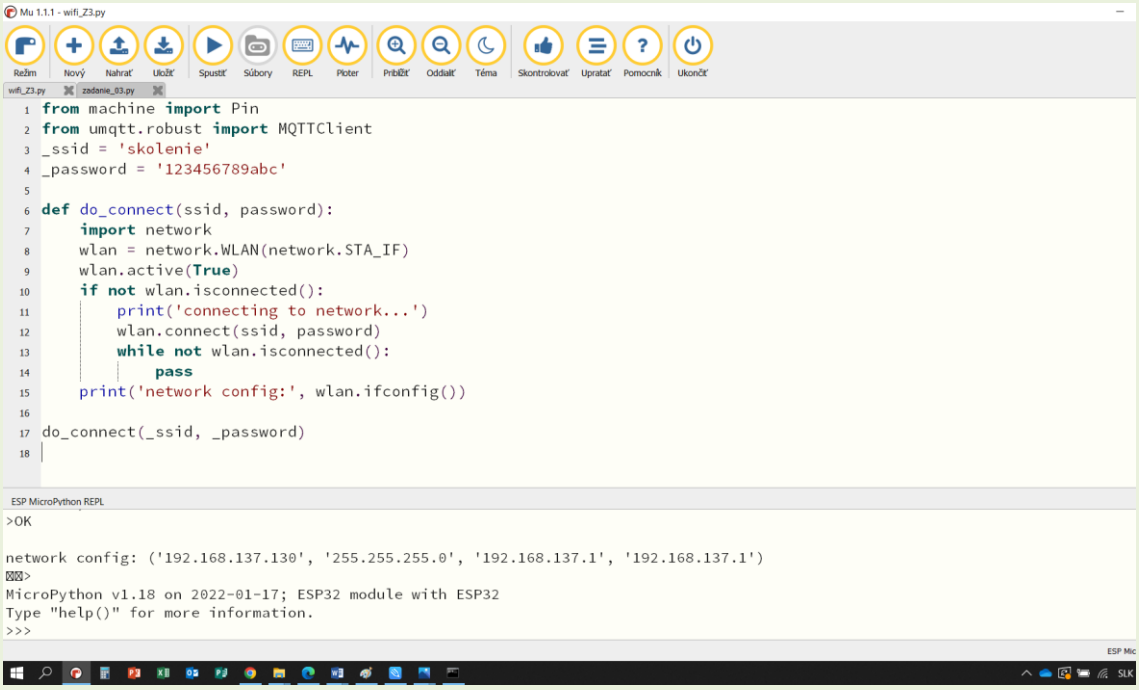
Praktická ukážka vizualizácie dát získaných z termistora

Zadanie: Vizualizácia údajov nameraných z termistora.

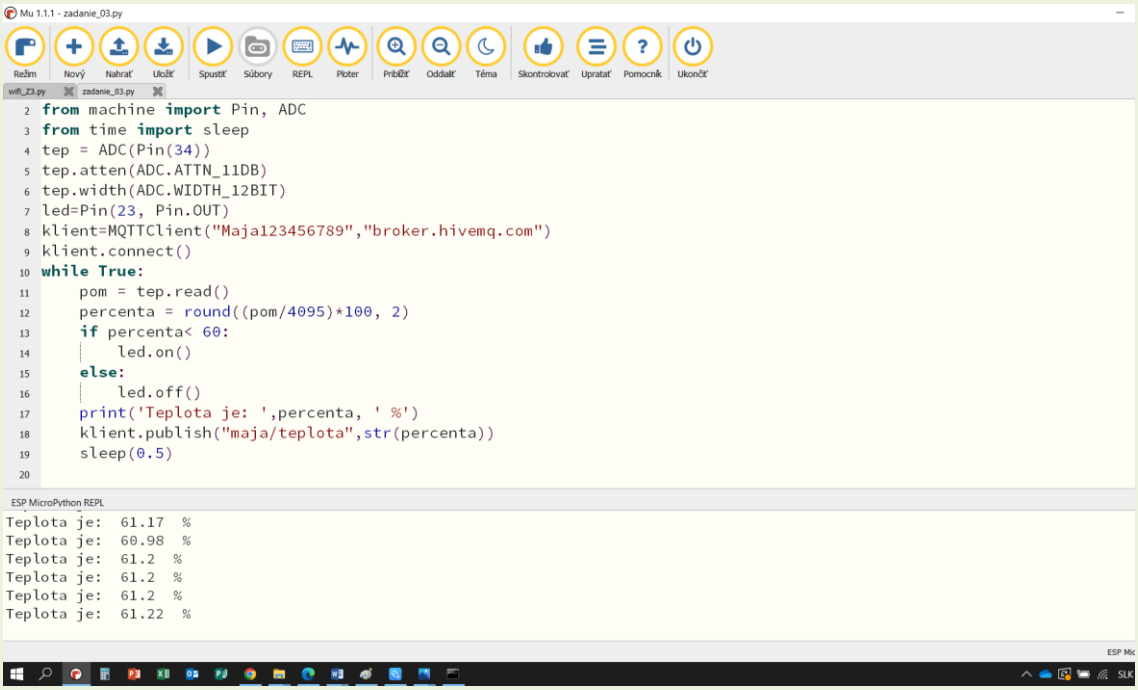
1. krok: Vytvorenie schémy v Tinkercade:



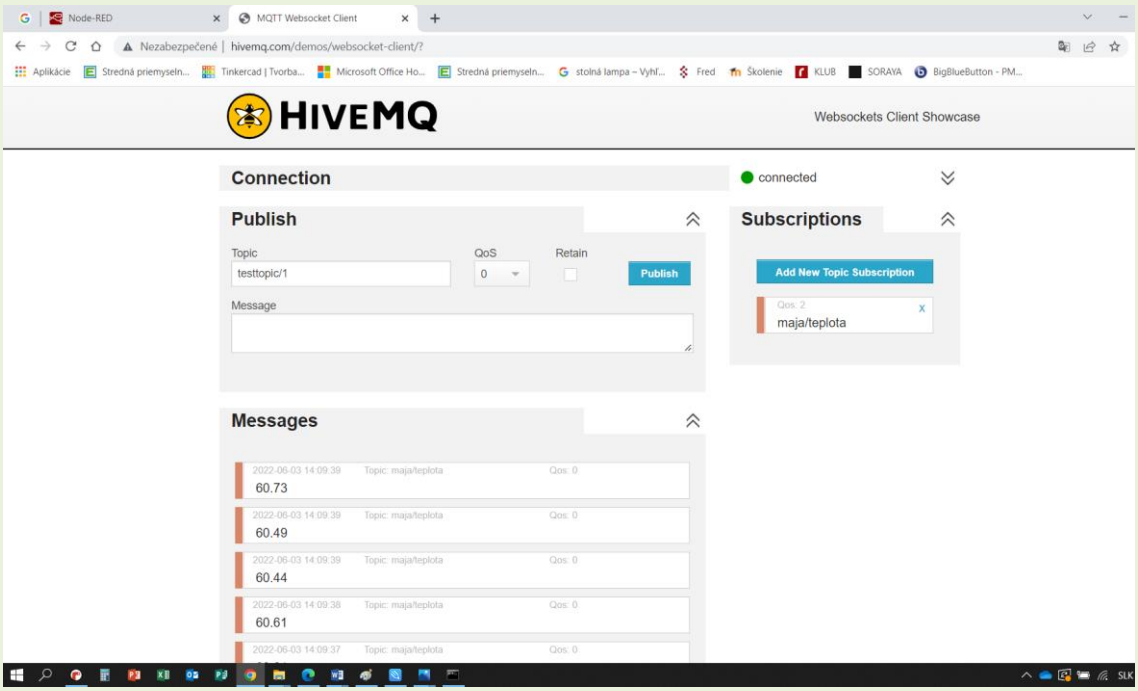
2. krok: Napísanie programu v editore Mú, ktorý pripojí PC k wifi



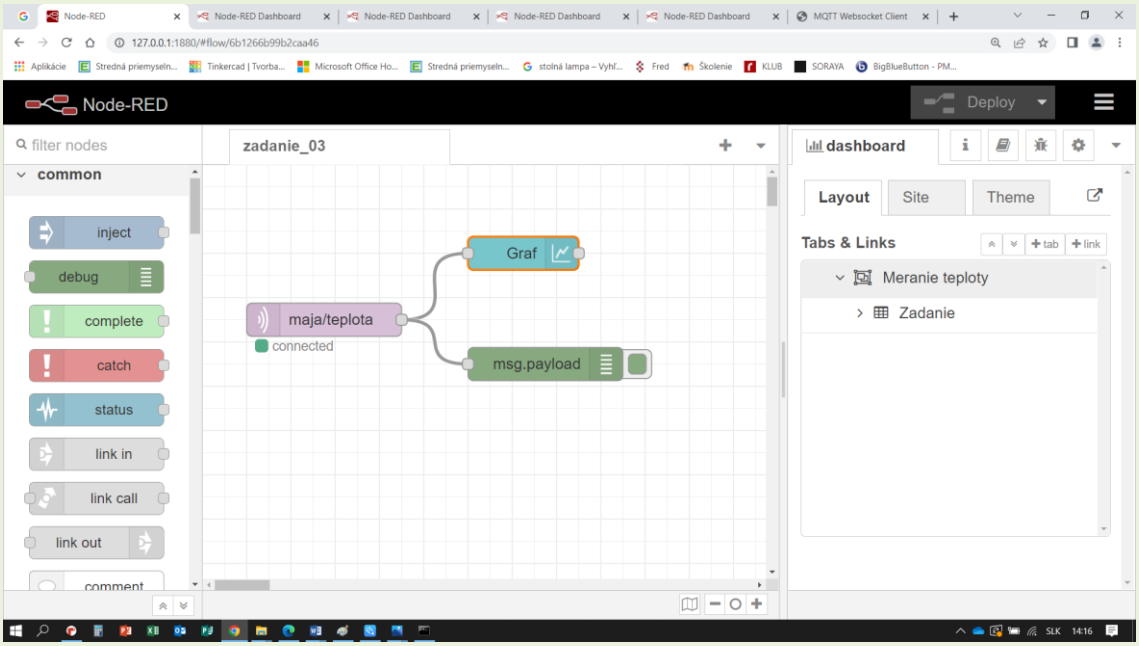
3. krok: Program na odosielanie dát na server



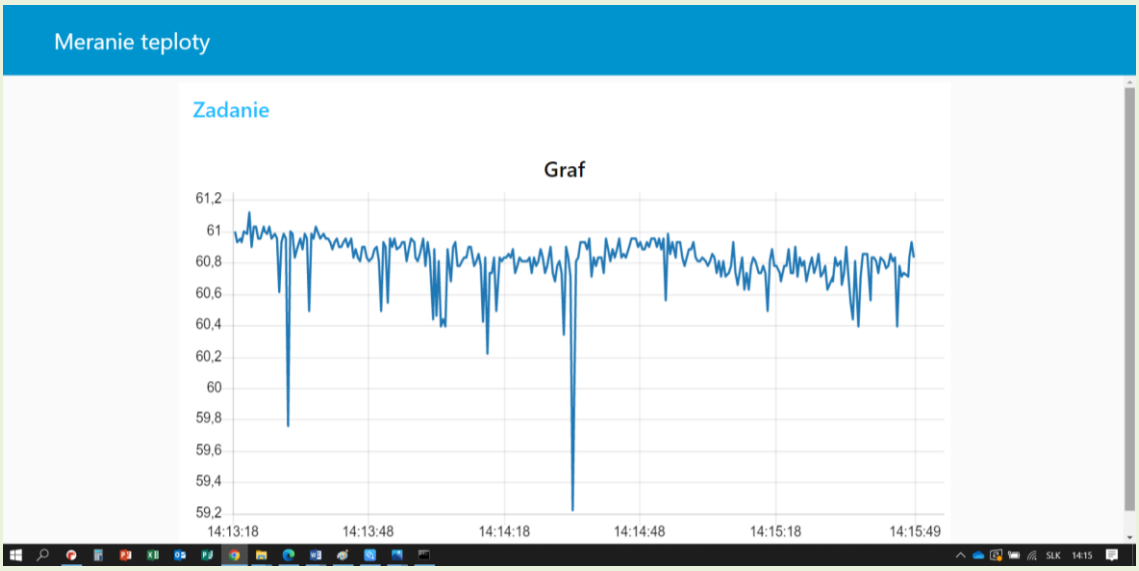
4. krok: Kontrola dát cez klienta



5. Node-RED

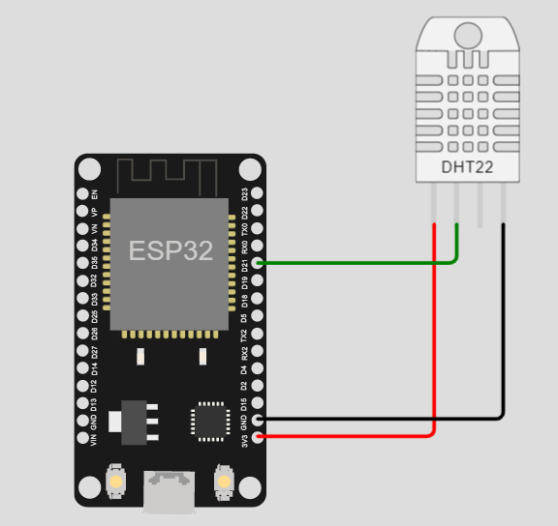


6. Vizualizácia získaných dát

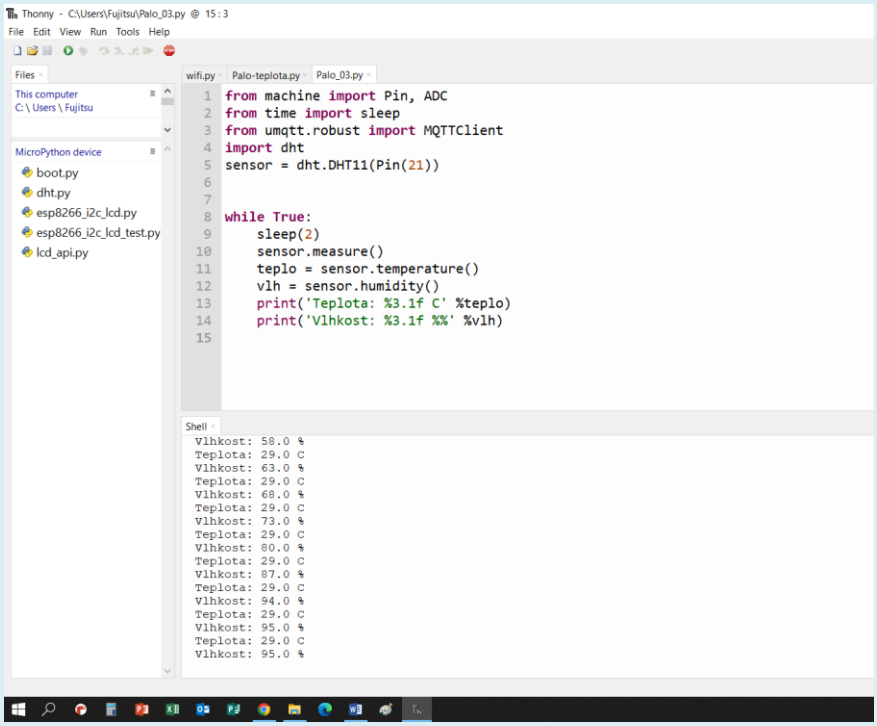


Praktická ukážka vizualizácie hodnôt získaných zo snímača teploty a vlhkosti

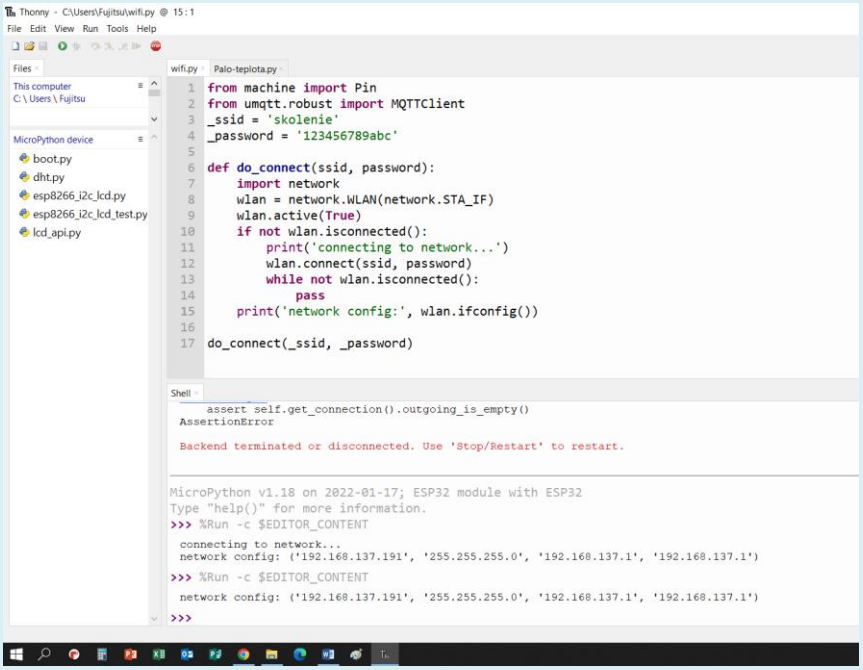
1. krok: Schéma zapojenia – obvod so snímačom teploty a vlhkosti



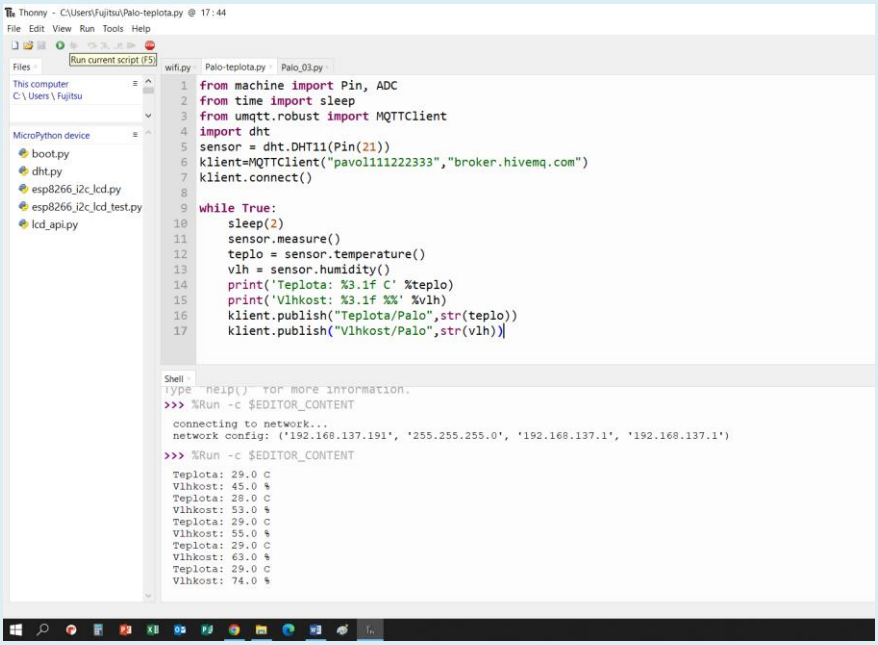
2. krok: Program na výpis nameraných hodnôt teploty a vlhkosti



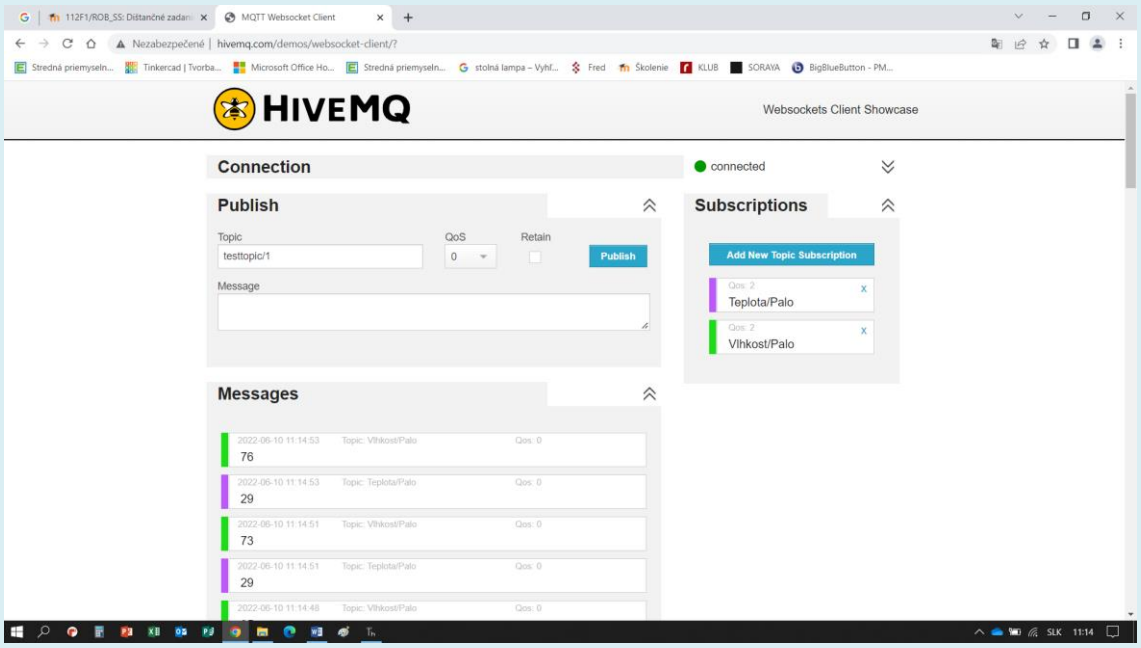
3. krok: Program na pripojenie na wifi



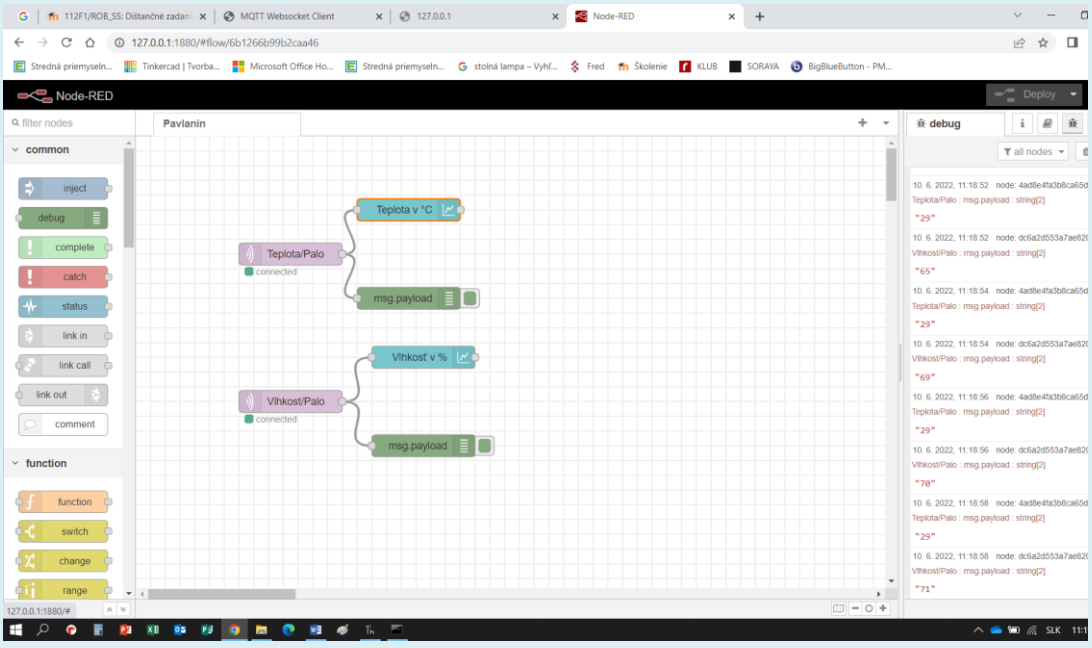
4. krok: Posielanie dát na broker



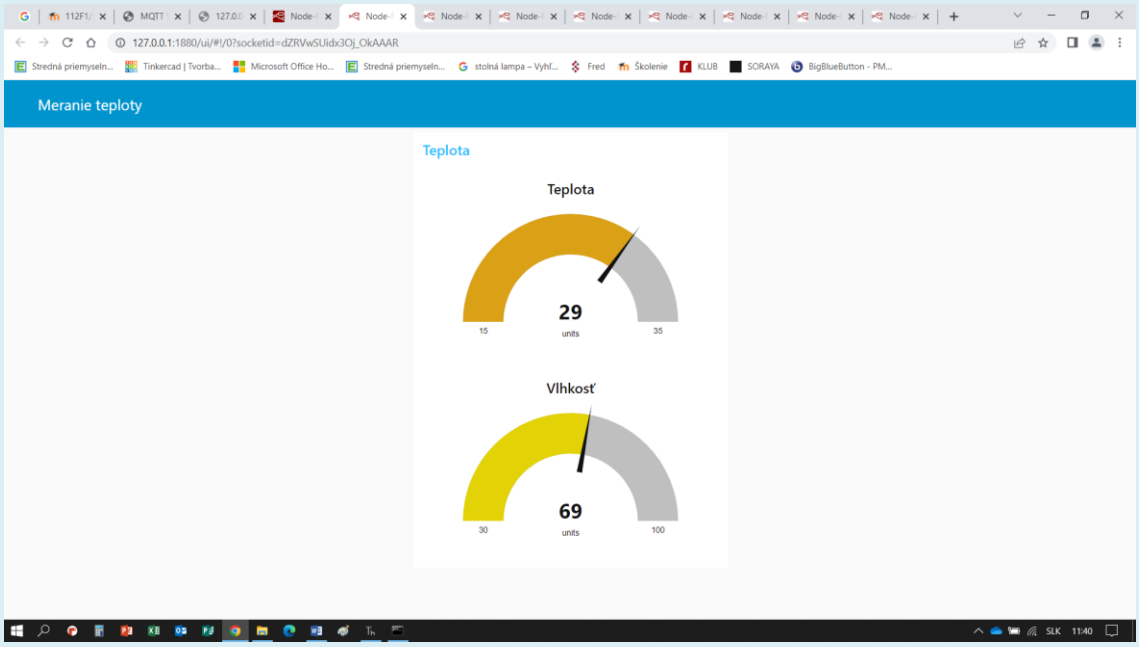
5. krok: Odoberanie dát cez HiveMQ



6. krok: Node-RED



7. krok: Vizualizácia dát



Záver:

Zhrnutie a odporúčania pre činnosť pedagogických zamestnancov:

- systematicky rozvíjať vedomosti a zručnosti žiakov v oblasti elektroniky, informatiky a programovania,
- motivovať žiakov k tvorbe projektov v oblasti elektroniky, informatiky a programovania,
- motivovať žiakov k aktívnemu zapájaniu sa do súťaží,
- organizovať praktické a pre žiakov zaujímavé súťaže hlavne prezenčnou formou,
- využívať rôzne nástroje pre simuláciu pre zatraktívnenie vyučovania,
- využívať vývojové prostredia na vizualizáciu dát získaných z rôznych senzorov.

11. Vypracoval (meno, priezvisko)	Mgr. Mária Forgáčová
12. Dátum	30.06.2022
13. Podpis	
14. Schválil (meno, priezvisko)	Ing. Pavol Pavlanin
15. Dátum	
16. Podpis	