

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Monitorovanie/riadenie napätia u_{R2} v RLC obvode

Záverečné zadanie - POIT

Úvod

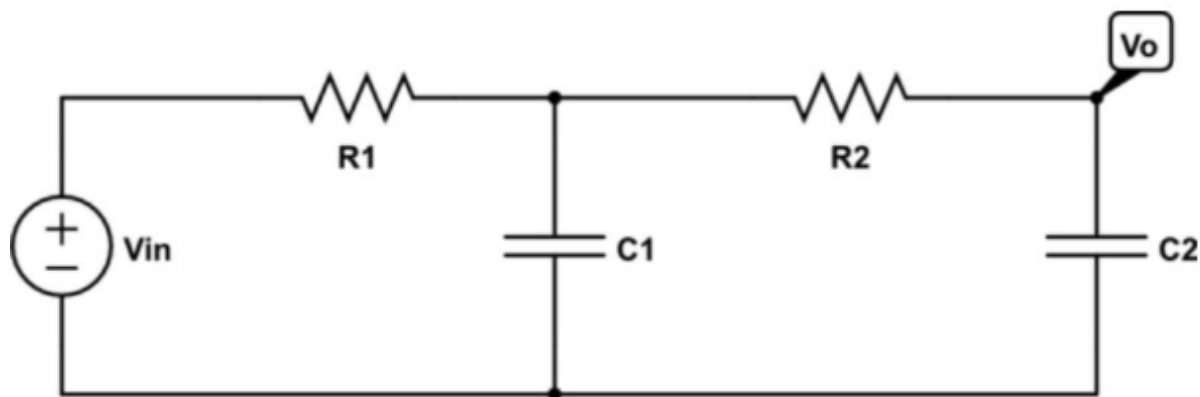
Cieľom zadania je monitorovať resp. riadiť signály získané z reálnych senzorov resp. simulačných a virtuálnych prostredí. Monitorovanie resp. riadenie sa má uskutočňovať prostredníctvom webovej aplikácie, aby bola naplnená koncepcia IoT.

Hardvér a softvér je podľa individuálnych možností. Preferuje sa využitie reálnej platformy Arduino a Raspberry. Štandardnou úlohou bolo vytvorenie webovej aplikácie v jazyku Python na platforme Raspberry Pi, ktorá by mala bude realizovať funkcie podľa zadania.

Vypracovanie zadania

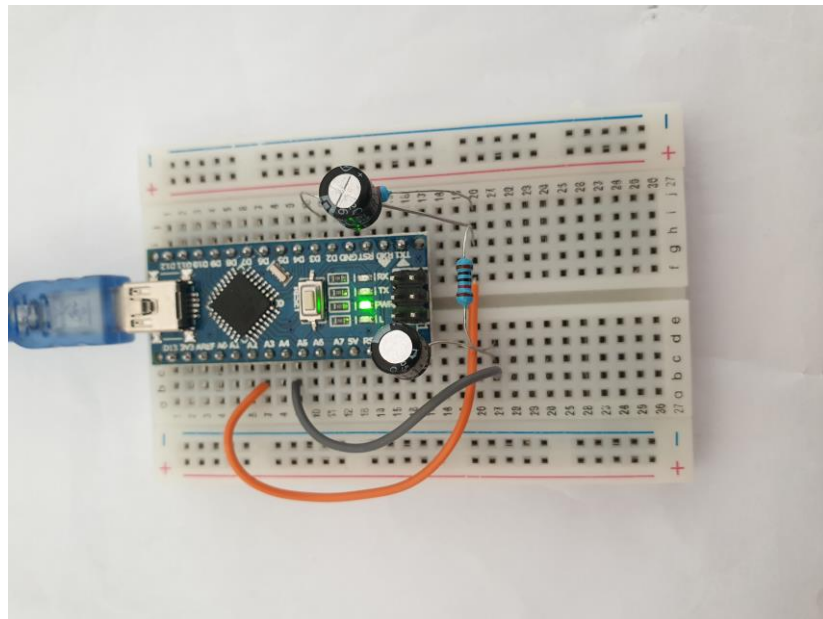
Kódy z vypracovaného zadania nájdete tu: <https://github.com/stetakova/poit>

Mojou úlohou bolo riadenie napätia u_{R2} za podmienky, že $R1 > R2$, $C1 > C2$. Pri skladaní obvodu som sa riadila nasledujúcou schémou. Údaje som získavala odpočítaním hodnôt u_{R2} - u_{R1} .



Serverová a klientska časť vrátane hardvéru – návrh, zapojenie a realizácia (vývojárska príručka):

Zapojenie reálneho obvodu:



Napísali sme si kód pre Arduino v Arduino IDE a pomocou pripojenia cez USB sme kód nahrali do zariadenia.

```
#define vystup 3
float value = 5;
|
void setup()
{
  Serial.begin(9600);
  pinMode(vystup, OUTPUT);
}

void loop(){
  float voltage1, voltage2, value;
  int low=1;

  voltage1 = (float)analogRead(A2)*5/1023;
  voltage2 = (float)analogRead(A4)*5/1023;
  float rozdiel=voltage1-voltage2;

  delay(500);
  if(rozdiel<0.02){
    digitalWrite(vystup, LOW);
    int low=1;
  }
  if(rozdiel == 0.00 && low==1){
    digitalWrite(vystup, HIGH);
  }
  delay(500);
  Serial.println(rozdiel);

  if(Serial.read() != -1){
    value = Serial.read();
  }
}
```

```
Done Saving.
avrdude: input file C:\Users\steta\AppData\Local\Temp\arduino_build_75766/sketch_apr25a.ino.he
avrdude: reading on-chip flash data:

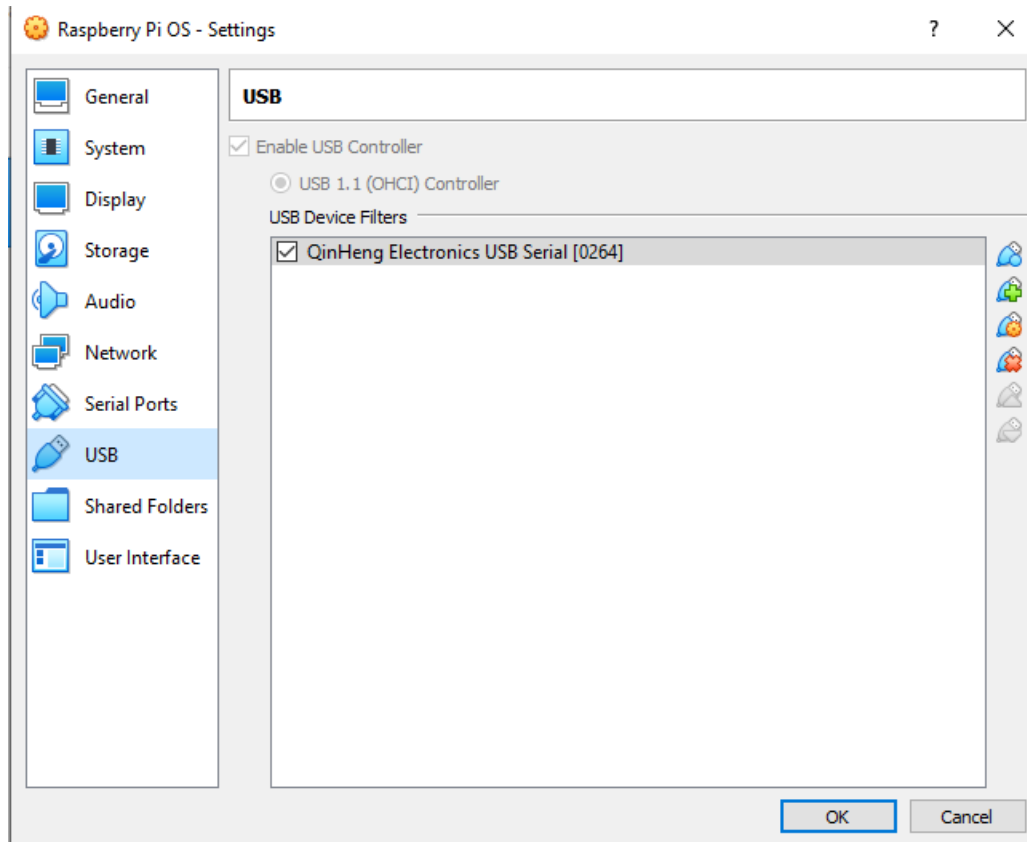
Reading | ##### | 100% 0.82s

avrdude: verifying ...
avrdude: 3818 bytes of flash verified

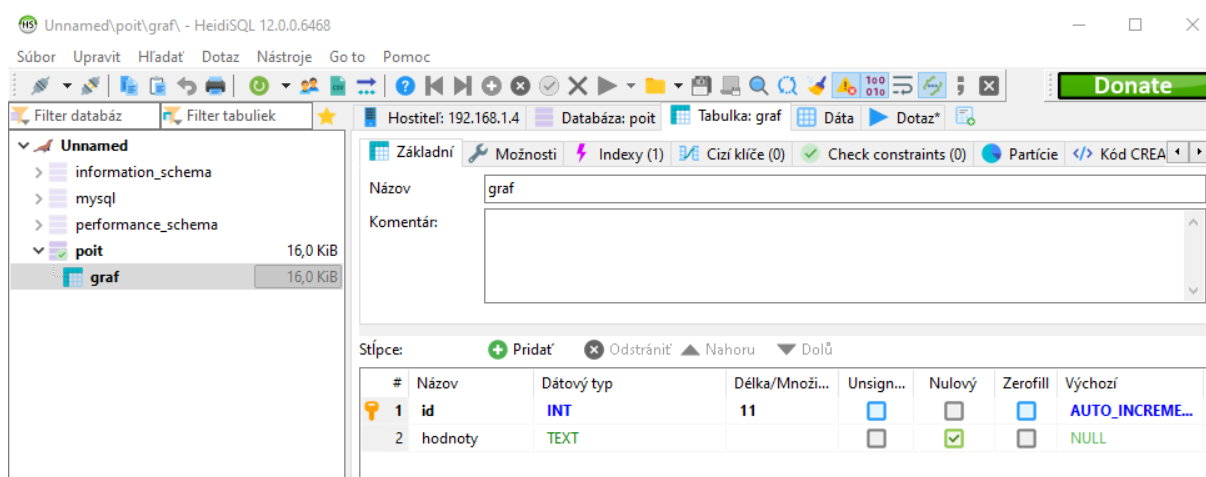
avrdude done. Thank you.
```

3 Arduino Nano. ATmega328P (Old Bootloader) on COM8

Arduino s Raspberry OS vo virtuálke sme prepojili cez USB QinHeng Electronics USB Serial.



Po prepojení sme si navrhli a vytvorili kódy k serverovej (app.py) a klientskej časti (index.html). Vytvorili sme si v HeidiSQL databázu poit a v nej tabuľku graf so stĺpcami id (auto_increment typu int) a hodnoty (typu text).

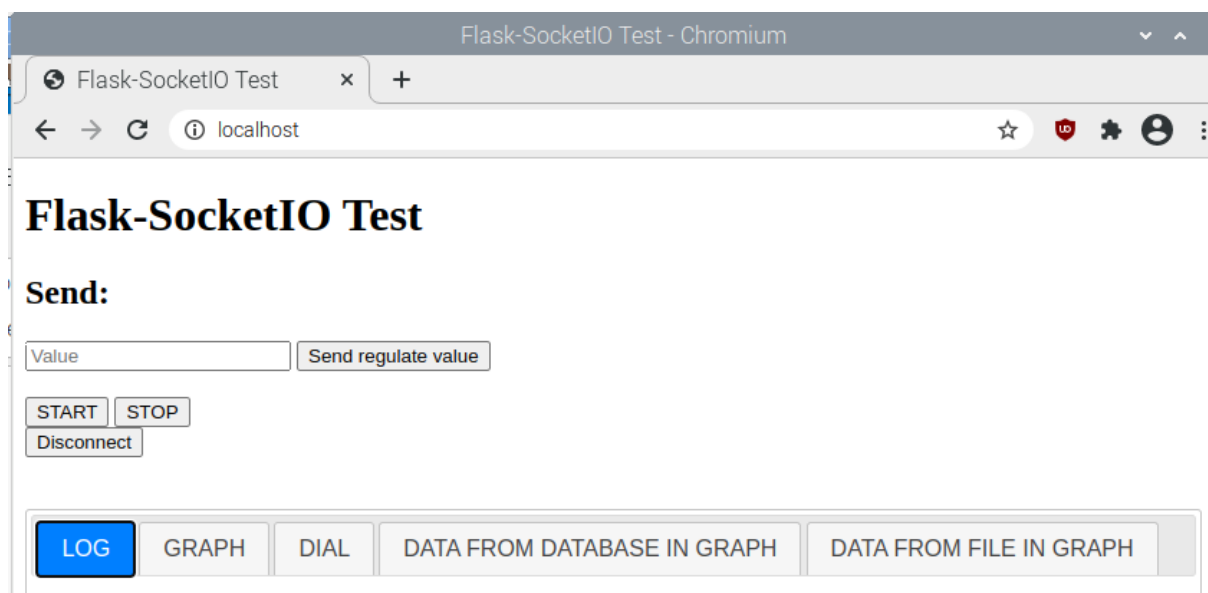


Tiež sme vytvorili priečinok static/files, kde je uložený súbor s názvom test.txt s dátami vo formáte JSON. Každý záznam sa uloží do poľa v jednom riadku a neskôr si ich vieme vykresliť do grafu práve na základe čísla riadku.

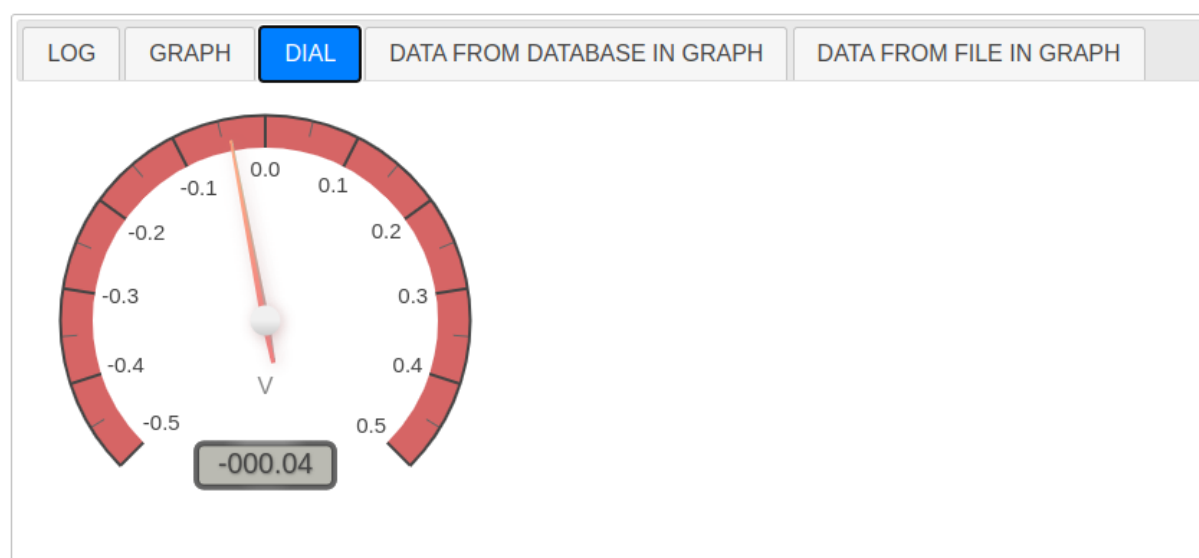
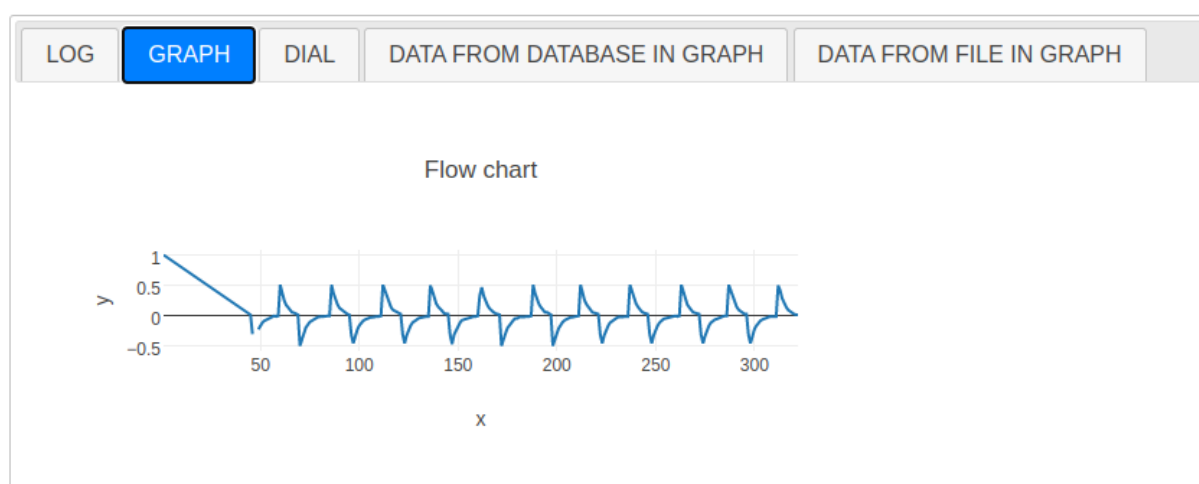
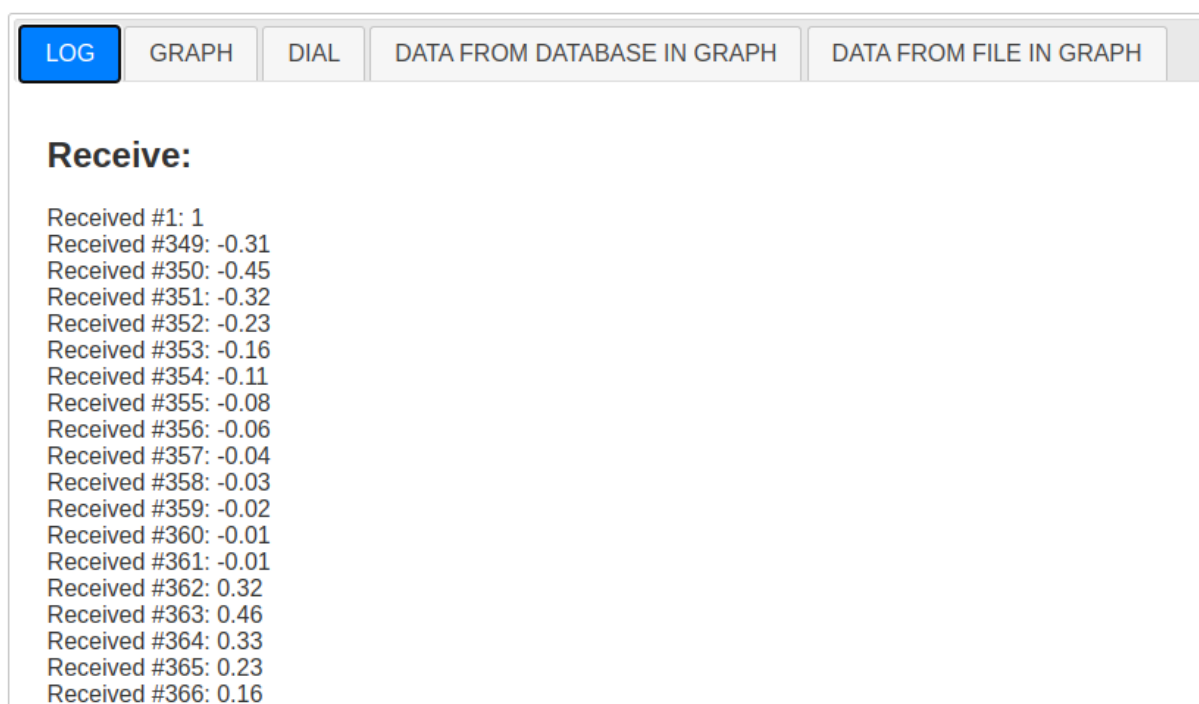
```
test.txt - Mousepad
File Edit Search View Document Help
[{"y": 0.6551787400492523, "x": 1, "t": 1522016547.531831},
```

Frontend a funkcionality (používateľská príručka):

Po pripojení Arduina s nahratým kódom a spustení serveru príkazom „*sudo pyhon3 app.py*“ spustíme vo webovom prehliadači localhost. Na úvodnej stránke aplikácie sa nachádza sekcia Send s poľom na vpísanie premennej, ktorá sa posiela do Arduina, tlačidlá START, STOP a Disconnect. Po stlačení tlačidla START sa spustí vypisovanie hodnôt z Arduina. Po stlačení tlačidla STOP sa vypisovanie zastaví a taktiež sa vymedzia hodnoty, ktoré sa zapíšu do databázy a do súboru. Tlačidlom Disconnect odpojíme webovú aplikáciu od servera a tým zastavíme posielanie dát. Ďalej sa tam nachádzajú taby LOG, GRAPH, DIAL, DATA FROM DATABASE IN GRAPH a DATA FROM FILE IN GRAPH.



V tabe LOG sa vypisujú prichádzajúce dáta, v tabe GRAPH môžeme pozorovať zakreslenie prichádzajúcich hodnôt do grafu, v tabe DIAL sa tieto isté hodnoty vykresľujú v ciferníku.



Hodnoty, ktoré sa zapísali do databázy a súboru test.txt si vieme v časti DATA FROM DATABASE IN GRAPH a DATA FROM FILE IN GRAPH vykresliť do grafu.

