

**Slovenská technická univerzita v Bratislave**  
**Fakulta elektrotechniky a informatiky**

# **Monitorovanie signálov získaných z reálnych senzorov prostredníctvom webovej aplikácie**

**Semestrálne zadanie**

Autor: Adam Ščevko

Predmet: Pokročilé informačné technológie

Dátum vyhotovenia: 2.6.2020

# Vývojárska príručka

## Zadanie

Cieľom zadania je monitorovať signály získané z reálnych senzorov. Monitorovanie sa má uskutočňovať prostredníctvom webovej aplikácie, aby bola koncepcia IoT naplnená.

Úlohou je vytvoriť webovú aplikáciu v jazyku Python, ktorá bude realizovať nasledovné funkcie:

1. Tlačidlom Open má spustiť aplikáciu, resp. inicializovať systém, nadviazať spojenie a aktivovať senzory.
2. Nastavenie parametrov monitorovania
3. Tlačidlom Start začať monitorovanie údajov
4. Výpis monitorovaných údajov vo forme záznamu v prehliadači klienta
5. Výpis monitorovaných údajov vo forme grafov v prehliadači klienta
6. Výpis monitorovaných údajov vo forme ručičkových ukazovateľov (ciferníkov) v prehliadači klienta
7. Ukladanie monitorovaných údajov do databázy s možnosťou ich výpisu a vykreslenia
8. Ukladanie monitorovaných údajov do súboru s možnosťou ich výpisu a vykreslenia
9. Tlačidlom Stop ukončiť monitorovanie údajov
10. Ukončenie aplikácie tlačidlom Close, resp. deaktivácia systému a ukončenie spojenia

Ku každému bodu zadania je potrebné vytvoriť serverovú a klientskú časť. Úlohou serverovej časti je zvyčajne monitorovanie údajov, pokým úlohou klientskej časti je vizualizácia.

# Vypracovanie zadania

## Micro:bit

Na vypracovanie tohto zadania bol použitý elektronický modul micro:bit. Micro:bit je najjednoduchší programovateľný plošný spoj dostupný na súčasnom trhu. Bol vyvinutý anglickou televíziou BBC v rámci kampane na podporu výučby programovania vo Veľkej Británii v roku 2016. Na doske sú priamo integrované:

- 25 (matica 5×5) samostatne nastaviteľných LED diód (možné použiť aj ako jednoduchý svetelný senzor)
- 2 tlačidlá
- vstupno výstupné piny na periférne zariadenia (napr. reproduktory, alebo iné rozširujúce dosky, 3 kapacitné kontakty použiteľné ako tlačítka)
- teplomer
- pohybový senzor (kompas a senzor naklonenia)
- možnosť bezdrôtového spojenia cez rádio alebo Bluetooth
- USB rozhranie

Je možné ich programovať vo viacerých jazykoch, ako čo sú Blockly (grafické programovanie – skladanie blokov), JavaScript (ES6 + TypeScript), Python (návody v slovenčine na programovanie micro:bitu v Pythone) a C / C++. Každý z týchto jazykov je mierne upravený / obohatený o špecifické funkcie micro:bitu, avšak jadro jazyka vždy ostáva.

Micro:bit bol programovaný v Online editore MakeCode, ktorý pôsobí pomerne dosť intuitívne a hravo a v maximálnej možnej miere uľahčuje vývoj. Program je možné priebežne testovať v simulátore, po dokončení stačí program stiahnuť (už skompilovaný hex súbor) a nahráť ho na micro:bit cez USB rozhranie. Do micro:bitu bol načítaný program, pomocou ktorého sa na základe nasnímaných hodnôt svetelného jasú (light level v rozsahu 0 až 255) rozsvietia LED žiarovky na určitý spôsob. V tomto prípade LED žiarovky predstavujú aj senzor aj aktuátor. Stupeň jasú som rozdelil na 5 úrovní. Pomocou sériovej komunikácie sa na počítač posiela údaj o úrovni svetelného jasú. resp. hodnoty 1 až 5 podľa toho, v akom rozsahu je okamžitá snímaná hodnota svetelného jasú. Konkrétne ak je snímaná hodnota svetelného jasú v rozsahu:

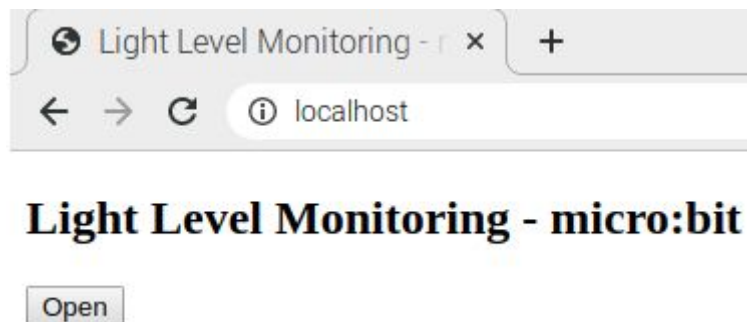
1. 0 - 40 => nesvieti žiadna LED dióda;
2. 40 - 85 => svieti jedine dióda nachádzajúca sa v strede;
3. 85 - 130 => LED diódy svietia v podobe znaku +;
4. 130 - 175 => svieti stredné pole diód v rozmeroch 3x3;
5. 175 - 255 => LED diódy sú rozsvietené v tvare slnka;

## 1. Tlačidlo Open

Stlačením tlačidla Open sa zavolá funkcia *Open()*, pomocou ktorej sa nadviaže spojenie medzi snímacím zariadením, resp. senzorom a počítačom. Okrem toho prepíname sa do nového bloku (tlačidlo skrýva samé seba, resp. celý tzv. *Open Block*, ktorý vidno na obr. 1), kde si následne volíme parametre - *Rate Block*.

Na nadviazanie spojenia bol použitý príkazový riadok:

```
socket = io.connect(location.protocol + '//' + document.domain + ':' + location.port + namespace);
```



Obr. 1 Open Block

## 2. Nastavenie parametrov monitorovania

Ohľadom tohto bodu zadania som vyhotovil blok pod názvom Rate Block (obr. 2) na nastavenie periódy, ktorou sa dáta budú zaznamenávať. Pomocou tohto bloku užívateľ má možnosť si zvoliť ako často chce, aby sa údaje zo senzora zaznamenávali, čo je užitočné ak napr. užívateľ má na ukladanie dát ohraničenú pamäť (vtedy si volí väčšiu periódu vzorkovania), alebo ak chce dáta zaznamenávať s väčšou presnosťou (môže si zvoliť menšiu periódu vzorkovania).

V tomto bloku sa nachádza posuvník, resp. slider, pomocou ktorého si užívateľ má možnosť zvoliť periódu vzorkovania v rozsahu 1 až 5 násobok pracovného cyklu (duty cycle). Pomocou funkcie *UpdateRate()* sa vykoná výpis hodnoty posuvníka do textového poľa, ktoré sa nachádza vedľa posuvníka. Stlačením tlačidla *Apply* sa zavolá funkcia *Open2()*, pomocou ktorej sa hodnota z posuvníka zapíše do premennej *sampleRate*, ktorá bude následne použitá v kóde. Zaznamenávanie údajov predurčenou periódou sa principiálne vykonáva pomocou príkazového riadku: *if (time % sampleRate == 0) => zaznamenávaj*, kde *time* je premenná, ktorá sa inkrementuje na konci pracovného cyklu. Taktiež sa stlačením tlačidla *Apply* prepíname do nového bloku.

## Light Level Monitoring - micro:bit

### Choose sampling period



Obr. 2 Rate Block

### 3. Tlačidlo Start

Tlačidlo Start sa spolu s takmer všetkými ostatnými užívateľskými komponentami tejto aplikácie (výber spôsobu zaznamenávania (záznam, graf, ciferník), tlačidlo *Stop*, tlačidlo *Database*, tlačidlo *Close*) nachádza v bloku pod názvom *Receive Block* (obr. 3). Stlačením tlačidla *Start* sa volá funkcia *Start()*, ktorou sa začne zaznamenávanie snímaných údajov. Na serverovej časti údaje sa zaznamenávajú do premennej *ser* príkazom

```
ser = serial.Serial('/dev/ttyACM0', 115200);
```

kde prvý argument funkcie predstavuje spojenie so sériovým portom, a druhý argument znamená baud rate.

Funkcia *Start()* v sebe zahŕňa volanie rôznych iných metód, či funkcií zo servera ako čo sú *connect*, *my\_event*, *my\_response*, ktoré sprostredkujú prenášanie nasnímaných údajov na webovú aplikáciu. Serverovská funkcia *connect()* zavolá príkaz, ktorým sa údaje začnú zaznamenávať (*ser.write("start\n")*).

## Light Level Monitoring - micro:bit



Obr. 3 Receive Block

## 4. Záznam

Zobrazenie nasnímaných údajov vo forme záznamu vykonáva stlačením tlačidla *List*, ktoré zavolá funkciu *ShowList()*, ktorá nám zobrazí želaný záznam. Záznam sa však začína vytvárať vo funkcii *Start()* príkazovým riadkom

```
$('#log').append('Received #'+porCislo+' : '+msg.data+'<br>').html();
```

Pomocná premenná *porCislo* bola vytvorená so zámerom aby priradila poradové číslo každej zaznamenanej vzorke. Táto pomocná premenná sa používa aj pri zobrazení časového priebehu snímanej hodnoty vo forme grafu. Zobrazenie nasnímaných údajov vo forme záznamu vidno na obr. 4.

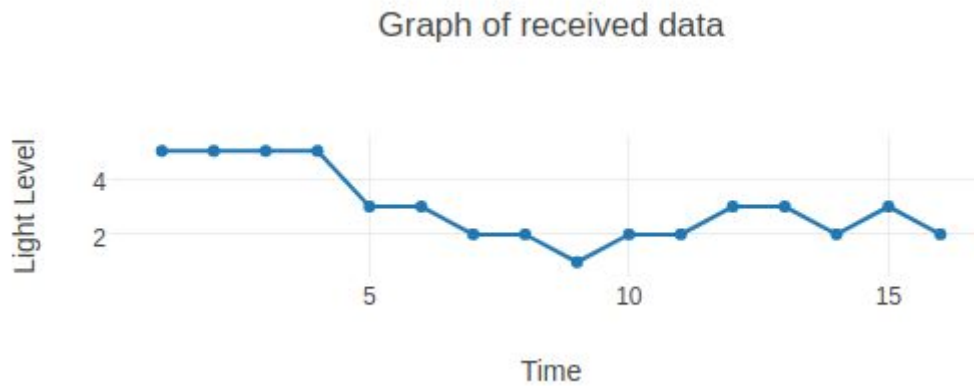
### List of received data:

```
Received #1: 5  
Received #2: 5  
Received #3: 5  
Received #4: 5  
Received #5: 3  
Received #6: 3  
Received #7: 2  
Received #8: 2  
Received #9: 1  
Received #10: 2  
Received #11: 2  
Received #12: 3  
Received #13: 3  
Received #14: 2
```

Obr. 4 Zobrazenie snímaných údajov vo forme záznamu

## 5. Graf

Zobrazenie nasnímaných údajov vo forme grafu vykonáva stlačením tlačidla *Graph*, ktoré zavolá funkciu *ShowGraph()*, ktorá nám zobrazí časový priebeh snímanej veličiny vo forme x-y grafu. Samotný graf sa vytvára zavolaním funkcie *Start()* a nevykreslí sa pokým sa tlačidlo *Graph* nestlačí. Vo funkcii *Start()* vieme nastavovať parametre grafu, ako čo sú údaje x-ovej osi, údaje ypsilonovej osi a názvy osí. Zobrazenie nasnímaných údajov vo forme grafu vidno na obr. 5.



Obr. 5 Zobrazenie snímaných údajov vo forme grafu

## 6. Ciferník

Zobrazenie okamžitej snímanej hodnoty vieme realizovať vo forme ručičkových ukazovateľov, resp. ciferníkov. V tejto aplikácii sa to vykonáva stlačením tlačidla *Gauge*, ktoré zavolá funkciu *ShowGauge()*, ktorá nám zobrazí želaný ciferník. Ciferník sa však začína vytvárať vo funkcii *Start()*, kde je možné meniť jeho parametre ako čo sú šírka a dĺžka objektu, názov jednotky snímanej veličiny, minimálna/maximálna hodnota, zobrazené hodnoty medzi nimi, rýchlosť animácie...



Obr. 6 Zobrazenie okamžitej snímanej hodnoty pomocou ručičkového ukazovateľa

## 7. Databáza

Vzhľadom na to, že som mal problémy s nainštalovaním databázy klasickým spôsobom, na vyhotovenie tohto bodu zadania som použil bezplatné služby stránky 000webhost.com, ktorá obsahuje súborový server a databázu. Najskôr som si tam vytvoril účet. Následne som vytvoril tabuľku, ktorá obsahuje stĺpce id, count, data a dátum. Id je primárny kľúč ktorý sa inkrementuje automaticky. Do súborového serveru som pridal 3 .php súbory potrebné na prácu s databázou.

1. Pomocou súboru database\_connect.php sa pripájame k databáze
2. Súbor pod názvom database\_addLightLevel.php slúži na pridávanie záznamov do tabuľky
3. Súbor database\_cleanTable.php vyčistí databázu pri reštartovaní stránky

Ich obsah možno pozrieť na githube. Adresa úložiska je uvedená na konci vývojarskej časti dokumentácie zadania.

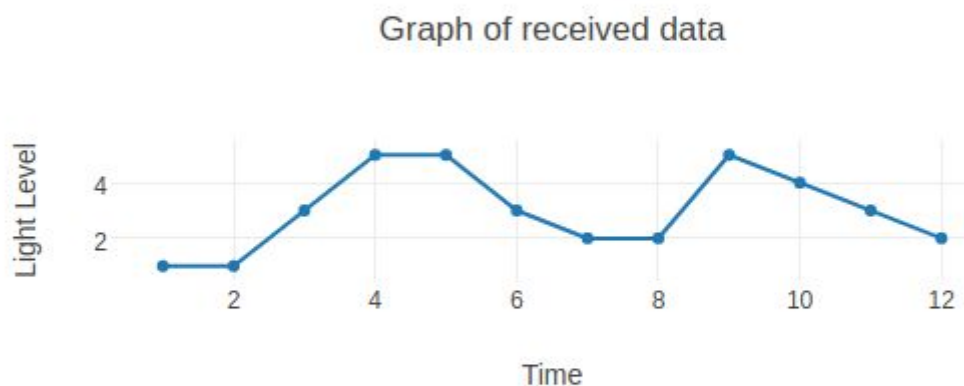
Tým, že sa dáta začnú zaznamenávať začne sa aj zapisovanie do databázy funkciami *CleanDatabase()* a *WriteDatabase()*, pomocou ktorých sa čerpajú zaznamenané údaje z databázy (obr. 7) zadáním konkrétnej webovej adresy. Výpis a vykresľovanie údajov z databázy sa vykonáva zavolaním súboru database.html, ktorý v sebe obsahuje potrebné funkcie ako čo sú *GetData()* a *WriteData()*. Zadáním do prehliadača localhost/db si údaje zaznamenané do databázy zobrazíme, tým že tak zavoláme súbor database.html.

| id  | count | data | datum               |
|-----|-------|------|---------------------|
| 406 | 1     | 5    | 2020-06-02 16:01:09 |
| 407 | 2     | 5    | 2020-06-02 16:01:10 |
| 408 | 3     | 5    | 2020-06-02 16:01:10 |
| 409 | 4     | 5    | 2020-06-02 16:01:11 |
| 410 | 5     | 3    | 2020-06-02 16:01:11 |
| 411 | 6     | 3    | 2020-06-02 16:01:12 |
| 412 | 7     | 2    | 2020-06-02 16:01:12 |
| 413 | 8     | 2    | 2020-06-02 16:01:13 |
| 414 | 9     | 1    | 2020-06-02 16:01:14 |
| 415 | 10    | 2    | 2020-06-02 16:01:14 |
| 416 | 11    | 2    | 2020-06-02 16:01:15 |
| 417 | 12    | 3    | 2020-06-02 16:01:15 |
| 418 | 13    | 3    | 2020-06-02 16:01:16 |
| 419 | 14    | 2    | 2020-06-02 16:01:16 |
| 420 | 15    | 3    | 2020-06-02 16:01:17 |

Obr. 7 Vykreslenie hodnôt do databázy, resp. tabuľky na stránke 000webhost.com



## Light Level Monitoring - Database



### List of received data:

Received #1: 1  
Received #2: 1  
Received #3: 3  
Received #4: 5

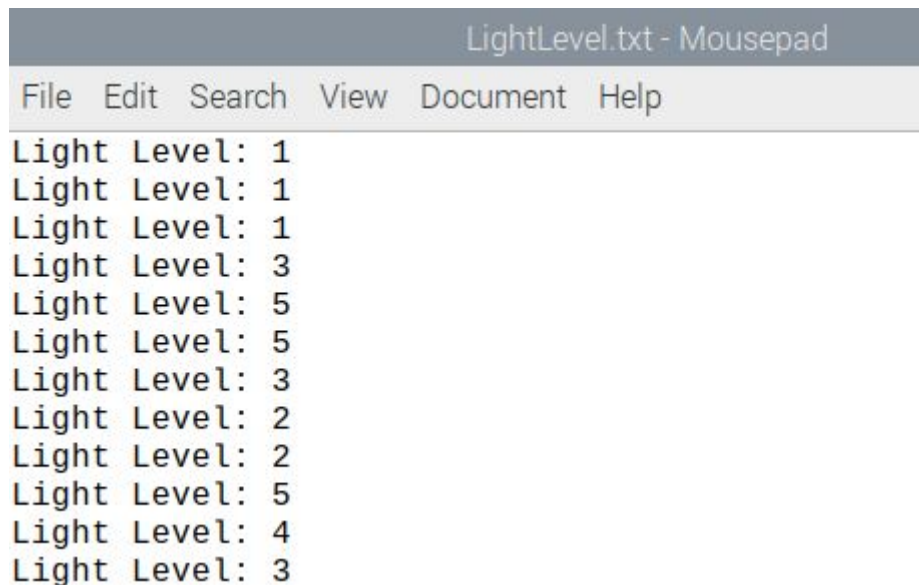
Obr. 8 Výpis a vykresľovanie dát na adrese localhost/db

## 8. Súbor

Táto webová aplikácia umožní aj zapisovanie snímaných dát do súboru. Táto časť zadania je splnená len zo serverovskej strany príkazmi

```
fo = open("static/files/LightLevel.txt", "a+")  
fo.write("Light Level: %s\r\n" % read_ser)
```

Údaje sa zapisujú do priečniku /static/files do súboru LightLevel.txt. Na začiatku každého zaznamenávania sa obsah súboru vymaže. Vypísané údaje do súboru vidno na obr. 9.



```
LightLevel.txt - Mousepad
File Edit Search View Document Help
Light Level: 1
Light Level: 1
Light Level: 1
Light Level: 3
Light Level: 5
Light Level: 5
Light Level: 3
Light Level: 2
Light Level: 2
Light Level: 5
Light Level: 4
Light Level: 3
```

Obr. 9 Výpis údajov do súboru

## 9. Tlačidlo Stop

Tlačidlom *Stop* sa zaznamenávanie údajov pozastaví. Vykoná sa to tak, že sa zavolá funkcia *Stop()*, ktorá servera volá funkciu zo servera *disconnect\_request*. Prenos údajov sa pozastaví nasledovným príkazovým riadkom (tento riadok sa nachádza vo funkcii *disconnect\_request*).

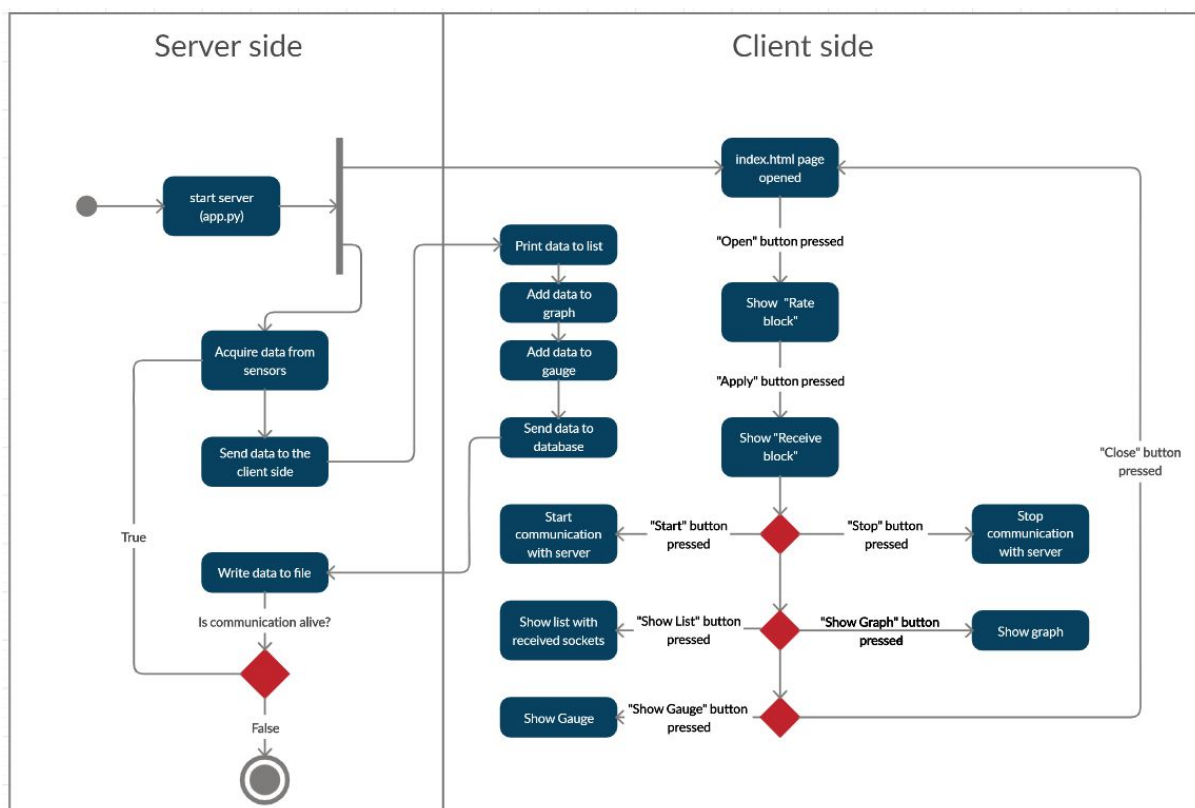
```
ser.write("stop\n")
```

## 10. Tlačidlo Close

Stlačením tlačidla *Close* sa užívateľ dostane do začiatočného stavu. Tým, že sa obnoví stránka tým, že sa zavolá príkaz *location.reload()*; Vymažú sa zaznamenané dáta a tak užívateľ má možnosť zaznamenávať dáta do prázdneho záznamu.

## UML Diagram

Pre lepšie pochopenie fungovania aplikácie som vyhotovil UML, resp. stavový diagram na stránke app.creately.com, ktorý je vidno na obr. 10.



Obr. 10 Stavový diagram webovej aplikácie

Adresa úložiska všetkých zdrojových súborov použité na vytvorenie tejto aplikácie

<https://github.com/scevkoadam/LightLevelMonitoring>

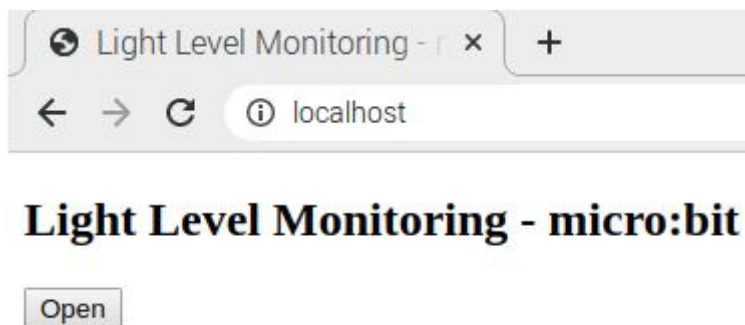
# Užívateľská príručka

## Light Level Monitoring via micro:bit

Cieľom aplikácie je monitorovať signály získané z reálneho senzora. Monitorovanie sa uskutočňuje prostredníctvom webovej aplikácie. Týmto je koncepcia IoT naplnená.

Používateľ má možnosť pomocou tejto webovej aplikácie okamžite zaznamenávať a monitorovať stupeň svetelného jas (light level), ktorý je snímaný elektronickým modulom micro:bit, konkrétne LED diódami. Stupeň svetelného jas predstavuje hodnotu 1 až 5. Prenos údajov beží pomocou sériovej komunikácie.

Spustením webovej aplikácie užívateľ sa dostane do začiatočného stavu, kde jeho jediná možnosť je stlačiť tlačidlo Open. Stlačením tohto tlačidla sa nadviaže spojenie medzi senzorom a počítačom. Pozri obr. 1.

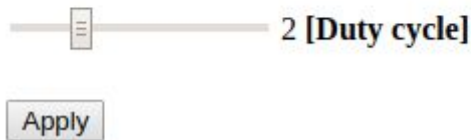


Obr. 1 Open Block

Následne si užívateľ má možnosť pomocou posuvníka zvoliť periódu, akou chce údaje zaznamenávať. Pomocou tohto bloku užívateľ má možnosť si zvoliť ako často chce, aby sa údaje zo senzora zaznamenávali, čo je užitočné ak napr. užívateľ má na ukladanie dát ohraničenú pamäť (vtedy si volí väčšiu periódu vzorkovania), alebo ak chce dáta zaznamenávať s väčšou presnosťou (môže si zvoliť menšiu periódu vzorkovania). Zvolením želanej periódy a stlačením tlačidla Apply (pozri obr. 2) sa užívateľ dostáva do zaznamenávacieho bloku.

## Light Level Monitoring - micro:bit

Choose sampling period



Obr. 2 Rate Block

Zaznamenávací blok ponúka užívateľovi možnosť zvoliť si spôsob vizualizácie dát. Užívateľ si môže voliť medzi záznamom, grafom a ciferníkom. Vid' obr. 3, 4 a 5.

## Light Level Monitoring - micro:bit

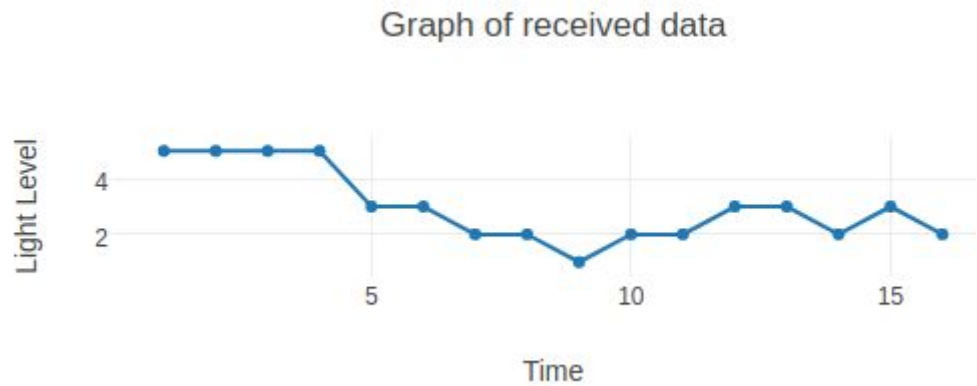


Obr. 3 Receive Block

### List of received data:

Received #1: 5  
Received #2: 5  
Received #3: 5  
Received #4: 5  
Received #5: 3  
Received #6: 3  
Received #7: 2  
Received #8: 2  
Received #9: 1  
Received #10: 2  
Received #11: 2  
Received #12: 3  
Received #13: 3  
Received #14: 2

Obr. 4 Zobrazenie snímaných údajov vo forme záznamu



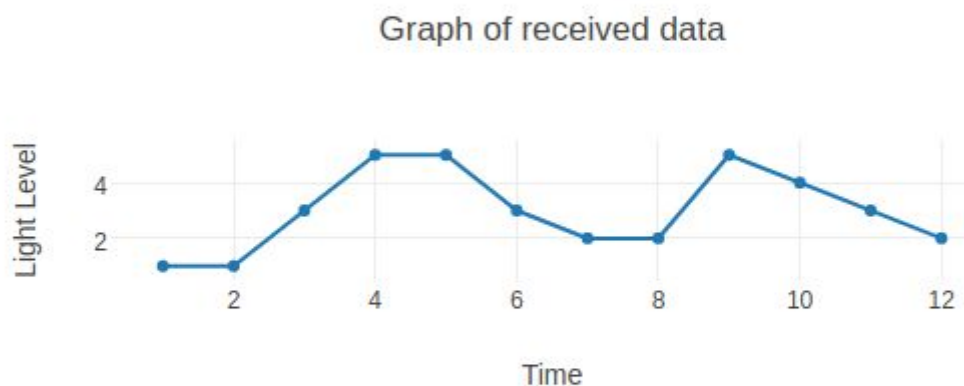
Obr. 5 Zobrazenie snímaných údajov vo forme grafu



Obr. 6 Zobrazenie okamžitej snímanej hodnoty pomocou ručičkového ukazovateľa

Užívateľ si stlačením tlačidla Start údaje ukladá aj do databázy, z ktorej ich následne vie vykresliť. Zadaním adresy localhost/db sa užívateľovi objaví zoznam zaznamenaných údajov do databázy a ich vykreslenie vo forme grafu ako je vidno na obr. 7.

## Light Level Monitoring - Database



### List of received data:

Received #1: 1  
Received #2: 1  
Received #3: 3  
Received #4: 5

Obr. 7 Výpis a vykresľovanie dát na adrese localhost/db

V zaznamenávacom bloku užívateľ má ešte možnosť stlačením tlačidla Stop zastaviť zaznamenávanie údajov. Stlačením tlačidla Close stránka sa obnoví, čím sa obnoví aj záznam, graf, ciferník, databáza aj súbor.