

ZÁPADOČESKÁ UNIVERZITA  
FAKULTA APLIKOVANÝCH VĚD  
KATEDRA KYBERNETIKY

# Dokumentace k projektu ITE-YELLOW

*Martin Hamar, Radek Kaupe, Samuel Kokoška*

# 1 Úvod

## 1.1 Zadání

## 2 Jednotlivé moduly

Projekt je rozdělen na několik modulů, které spolu přirozeně spolupracují. Tyto moduly se dají hrubě rozdělit na hardwarovou část a na softwarovou část. Hardwarová část zahrnuje krabičku a mikropočítač ESP8266 a jeho nastavení. Softwarovou část zahrnuje nastavení databáze, subscribéra a spuštění backendu a webové stránky.

### 2.1 Krabička

Obsah sekce "Krabička".

### 2.2 ESP8266

#### 2.2.1 Funkce

Kompletní zařízení disponuje funkcemi:

- Měření teploty, vlhkosti a osvětlení.
- Možnost zastoupení dedikovaného senzoru teploty v případě jeho poruchy méně přesným senzorem vlhkosti.
- Posílání hodnot indikujících poruchu senzorů.
- Synchronizace času s NTP serverem.
- Posílání naměřených dat přes MQTT s Quality Of Service 1 - zpráva je doručena alespoň jednou.
- Detekce selhání zaslání zprávy a tedy ztráty spojení.
- Znovupřipojení se k Wi-Fi a MQTT brokeru v případě ztráty spojení.
- Archivace naměřených dat v případě ztráty spojení a jejich znovuposlání po obnovení spojení.
- Poslání tracebacku a samovolný reset v případě chyby v hlavní smyčce.

#### 2.2.2 Setup

Pro použití na Vašem pracovišti je potřeba změnit v souboru main.py konstanty

- NET\_NAME - název Vaší Wi-Fi sítě
- NET\_PASS - heslo k této síti
- NTP\_HOST - adresa serveru pro synchronizaci času
- BROKER\_IP - IP adresa MQTT brokera
- BROKER\_UNAME - uživatelské jméno pro připojení k brokeru
- BROKER\_PASSWD - heslo pro připojení k brokeru
- TOPIC - topic, na který budou zprávy zasílány

Volitelně je zde možnost měnit parametry

- `PERIOD_SEC` - perioda měření a posílání v sekundách
- `TIMEOUT` - maximální doba čekání na potvrzení přijetí zprávy od brokera v sekundách - po jejím uplynutí začne archivace a pokusy o znovupřipojení
- `RECON_PERIOD` - perioda pokusů o znovupřipojení se na Wi-Fi a brokera v milisekundách
- doporučuje se změnit/vymazat ze struktury zprávy položku `team_name`

Společně s `main.py` je potřeba nahrát do zařízení následující knihovny, pokud tak již nebylo učiněno:

- `temp_sensor.py`
- `light_sensor.py`
- `umqtt.py`

Dále stačí jen zapojit zařízení do napájení. Pokud je vše v pořádku, tak by po připojení se na brokera měla zhasnout LED na pinu 16 (D0 na fyzickém zařízení) a měly by se začít posílat zprávy na daný topic.

## 2.3 Databáze

K realizaci projektu bude třeba založit PostgreSQL databázi na stroji, kde chcete nechat běžet pythonské skripty. Uživatele, heslo, hosta a název databáze je třeba napsat do `.env` složky tímto způsobem:

```
DB_USER=uzivatel
DB_PASSWORD=heslo
DB_HOST=host
DB_NAME=nazev_databáze
```

Po tomto kroku můžete spustit skript `db.py`, který vytvoří potřebné tabulky: `sensor_data`, `teams`, `sensor_data_outliers`. Tabulka `sensor_data` je hlavní tabulka, kde se ukládají veškerá validní data. V tabulce `teams` jsou uložena jména a ID jednotlivých týmů a do tabulky `sensor_data_outliers` se ukládá informace, zda záznamy přesahují Aimtecem vymezené hranice. Tabulka `sensor_data_test` je určena pro debugovací a testovací účely. Ukládají se zde například i záznamy z budoucnosti či se zápornou illuminací, aby se tyto případy mohli identifikovat a na straně ESP8266 opravit.

## 2.4 Subscriber a připojení k Aimtec AWS

Před samotným spuštěním subscribera je třeba vyřešit credentials týkající se připojení na Aimtec. je hlavní tabulka, kde se ukládají veškerá validní data

### 2.4.1 Aimtec

Pro zajištění posílání dat na Aimtec upravte `.env` soubor, aby obsahoval `AIMTEC_URL=` a zadejte správnou URL adresu. Poté spusťte `aimtec.py` skript, pro získání `TEAM_UUID`. Ten také zadejte do již zmíněného souboru. Nyní by měl `.env` soubor vypadat následovně:

### 2.4.2 Subscriber

Pro zajištění komunikace s brokerem je třeba opět upravit `.env` soubor, je třeba zadat následující položky:

```
BROKER_IP=IP
BROKER_PORT=PORT
BROKER_UNAME=uzivatelske_jmeno
BROKER_PASSWD =heslo
TOPIC=topic
```

Po provedení úprav v sekcích 2.3, 2.4.1 a 2.4.2 by tedy vaše `.env` složka měla vypadat následovně:

```
DB_USER=uzivatel
DB_PASSWORD=heslo
DB_HOST=host
DB_NAME=nazev_databaze
AIMTEC_URL=URL
TEAM_UUID=UUID
BROKER_IP=IP
BROKER_PORT=PORT
BROKER_UNAME=uzivatelske_jmeno
BROKER_PASSWD =heslo
TOPIC=topic
```

Pokud ano, nyní můžete spustit skript `subscriber_vm.py`.

### 2.4.3 Funkce samotného subscribera

Subscriber má několik funkcí:

1. Validace příchozích dat
2. Přijímání chyb z ESP8266
3. Ukládání dat do databáze
4. Posílání dat na Aimtec AWS

#### Validace dat

Momentálně subscriber validuje:

1. správný formát příchozího JSONu
2. správný název týmu
3. smysluplné hodnoty teploty, osvětlení a vlhkosti
4. zda má zpráva jiný `timestamp`, než ta poslední (kvůli použití QoS 1)
5. zda `timestamp` není více jak hodinu z budoucnosti

#### Přijímání chyb z ESP8266

U přijímání chyb je třeba zmínit, že je ukládá do složky `subscriber\err\`, kde názvy souborů jsou `timestampy`, které se v chybové zprávě vyskytují.

#### Ukládání do databáze

Do "ostré" databáze se ukládají pouze validní data. Pro validační a testovací účely se většina dat (ve správném formátu) uloží do testovací tabulky. Je to z důvodu možnosti zjištění případných chyb, ať už na straně ESPčka nebo subscribera. Také lze beztréstně manipulovat s daty, aniž by se změny projevíli na webové stránce.

### Posílání dat na Aimtec

Funkčnost se dá popsat následovně: Při spuštění skriptu se pokusí přihlásit na Aimtec AWS. Pokud server nespí, proběhne všechno v pořádku a data se budou posílat na Aimtec každou minutu. V případě, že AWS nefunguje, skript se bude o login pokoušet každých deset vteřin, dokud se mu to nepodaří. V pozadí vše ostatní bude probíhat tak jak má (především samozřejmě ukládání dat do databáze), ale data se na AWS nepošlou. Momentálně nefunguje žádné zpětné posílání dat, takže v případě výpadku/spánku AWS serveru, data jsou ztracena.

## 2.5 Backend

Backend webové stránky se jednoduše spustí pomocí skriptu `backend.py`. Musí se spustit na stroji, kde se vyskytuje databáze a zároveň běží skript `subscriber_vm.py`. K souběžnému pouštění skriptů doporučuji tmux. Pokud se Vám skript podařil spustit, stránky by nyní měly fungovat a pracovat zcela nezávisle na ostatních modulech. Pouze si vybírají potřebná data z databáze a zobrazují je. Na hlavní stránce jsou k vidění nejnovější poslaná data každého týmu. Je zde zajištěna asynchronní komunikace pomocí technologie WebSockets. Na stránce `/statistics` jsou k vidění celkové množství naměřených dat a základní statistiky žlutého týmu. Poté na stránkách `/graphs-one-day` a `/graphs-one-week` jsou k vidění grafy, zobrazující průměrnou hodnotu každé hodinu/každého den za poslední den/sedm dní. Grafy jsou částečně interaktivní lze, vyškrtnutím týmů z nabídky v grafu, zobrazit jen některé týmy.

## 3 Závěr