

SIN – Inteligentní systémy

Implementace meteostanice s autonomním řízením

Bc. Petr Stehlík <xstehl14@stud.fit.vutbr.cz>

Bc. Matej Vido <xvidom00@stud.fit.vutbr.cz>

1 Úvod

Cílem projektu je navrhnout a implementovat meteostanici využívající MQTT protokol pro zasílání naměřených hodnot serveru. Server implementuje databázi a webové grafické uživatelské rozhraní (GUI), analyzuje historické hodnoty a na jejich základě řídí akce aktuátorů.

Meteostanice měří teplotu, vlhkost a tlak vzduchu; vlhkost v květináči a světelnou intenzitu. Navržené aktuátory jsou ovládání žaluzií, klimatizace, topení a zavlažování rostlin. Vše je autonomně ovládáno na základě předem stanovených pravidel.

GUI zobrazuje aktuální a historické naměřené hodnoty jednotlivých senzorů, trendy a naposledy vykonané akce aktuátorů společně s krátkodobou předpovědí počasí získanou z volně dostupných zdrojů.

2 Nástroje pro monitoring a řízení

Na trhu je velké množství dostupných nástrojů pro monitoring a řízení a to i na poli chytrých domácností. Je zde mnoho komerčních a uzavřených systémů pro automatizaci domácnosti. V současnosti nejznámější a pravděpodobně i nejrozšířenější je Apple Home¹, který využívá protokolu Homekit také od společnosti Apple. Na trhu existuje pro Apple Home velké množství produktů a neustále se jejich počet zvyšuje. Existuje ale i mnoho open-source nástrojů pro řízení domácnosti. Jejich nejrozšířenější zastupitele jsou zde v krátkosti popsány.

2.1 Grafana

Grafana² je vizualizační a analytický nástroj pro data zachycená v čase. Grafana samotná není primárně určena pro monitoring a řízení chytrých domácností, ale je natolik upravitelná, že existují konfigurace, které toto užití zpřístupňují. Je dostupná s mnoha rozšířeními a zdroji dat. Uživatel si vytvoří dashboard, který si následně nakonfiguruje a seskládá z dostupných modulů. Tyto moduly mimo dalších funkcionalit umožňují zobrazit čárový graf, jednotlivé hodnoty, trendy hodnot a s pomocí modulů i různé přepínače.

2.2 Domoticz

Domoticz³ je kompletním open-source systémem pro domácí automatizaci. Tento nástroj dovoluje monitorovat, řídit a konfigurovat mnoho různých zařízení od různých výrobců. Disponuje i automatickým učením senzorů a aktuátorů. Rozhraní pro uživatele je vytvořeno jako webová stránka dostupná na stroji s lokální instalací Domoticz.

2.3 OpenHAB

OpenHAB⁴ je dalším velmi známým zástupcem open-source nástrojem pro domácí automatizaci. Podporuje velké množství platforem, výrobců a zařízení. Dokáže integrovat mnoho systémů do jednoho

¹<https://www.apple.com/lae/ios/home/>

²<https://grafana.com/>

³<https://domoticz.com/>

⁴<https://www.openhab.org/>

centrálního řešení, které lze ovládat mobilní aplikací, webových rozhraním nebo nativní desktopovou aplikací. Pro automatizaci disponuje rozsáhlým systémem pro tvorbu komplexních pravidel.

2.4 Home Assistant

Home Assistant⁵ je primárně navrhován pro užívání na Raspberry Pi. Jako předchozí nástroje také podporuje široké množství výrobců a zařízení. Také disponuje konfigurovatelným webovým rozhraním s přehledy zařízení a kontrolou aktuátorů.

2.5 BeeeOn

BeeeOn⁶ je systém pro domácí automatizaci vyvíjený na FIT VUT v Brně. Tento systém je primárně vyvíjen jako bezpečná domácí brána pro různé IoT zařízení. Oproti předchozím systémům je nutno pro plnou funkcionalitu systému vlastnit domácí bránu, kterou lze propojit s několika výrobci a jejich zařízeními. BeeeOn disponuje Android mobilní aplikací pro ovládání domácnosti a prototypem webového rozhraní.

2.6 Vlastní řešení

Okrem uvedených kompletných riešení pre monitoring a analýzu riadiacich systémov je možné poskladať vlastné riešenie z dostupných open-source knižníc, frameworkov a nástrojov.

Rozšířeným frameworkom pre tvorbu klientskej časti webových aplikácií je Angular⁷. Angular napísaný v jazyku TypeScript umožňuje jednoduchú integráciu rôznych knižníc pre zobrazovanie grafov a tvorbu dashboardov.

K implementácii serverovej časti webovej aplikácie je možné použiť framework Flask⁸ v jazyku Python. Flask umožňuje rýchlu a jednoduchú tvorbu REST API na prepojenie serverovej a klientskej časti webovej aplikácie. Výhodou použitia jazyku Python pre tvorbu serverovej časti aplikácie je, že obsahuje moduly pre obsluhu databázového systému a zabezpečenie sieťovej komunikácie s ďalšími prvkami celého riadiaceho systému.

Mosquitto⁹ je voľne dostupný broker pre MQTT protokol používaný v prostredí systémov IoT podporujúci rôzne platformy.

3 Architektura a implementace

V tejto sekcii je popísaná architektúra a implementácia vytvorenej meteostanice s autonómnym riadením. Meteostanica obsahuje reálne senzory, z ktorých sú dáta prenášané na server. Server dáta ukladá do databázy, riadi činnosť aktuátorov podľa pravidiel, ktoré sú taktiež uložené v databáze, a obsluhuje webovú aplikáciu, ktorá poskytuje grafické užívateľské rozhranie k celému systému. Činnosť aktuátorov je simulovaná. Pre účely testovania bol taktiež zostrojený modul pre simulovanie činnosti senzorov. V časti 3.1 je popísaný hardware a spôsob jeho zapojenia. V nasledujúcej časti 3.2 je popísaná implementácia softwareových modulov systému.

3.1 Hardware

Základom pro hardware byly zvoleny Raspberry Pi 2 jako server a Raspberry Pi Zero W jako samotná meteostanice. Server realizován jako Raspberry Pi 2 zpřístupňuje uživateli minimalistické webové rozhraní a slouží jako MQTT broker pro meteostanici. Také jsou zde ukládány všechny naměřené hodnoty do SQLite3 databáze. Na obrázku ?? je znázorněná schéma zapojení senzorů a Raspberry Pi.

K Raspberry Pi Zero W jsou připojeny následující senzory:

⁵<https://home-assistant.io/>

⁶https://beeeon.org/wiki/Main_Page

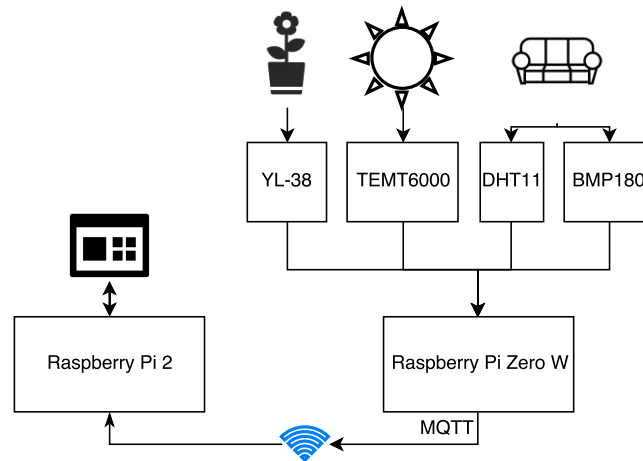
⁷<https://angular.io/>

⁸<http://flask.pocoo.org/>

⁹<https://mosquitto.org/>

- DHT11 - digitálny senzor pro měření teploty a vlhkosti vzduchu
- BMP180 - digitálny senzor pro měření barometrického tlaku, teploty a nadmořské výšky
- TEMT6000 - analogový senzor pro měření intenzity okolního světla
- YL-69 společně s YL-38 - analogový senzor pro měření vlhkosti půdy

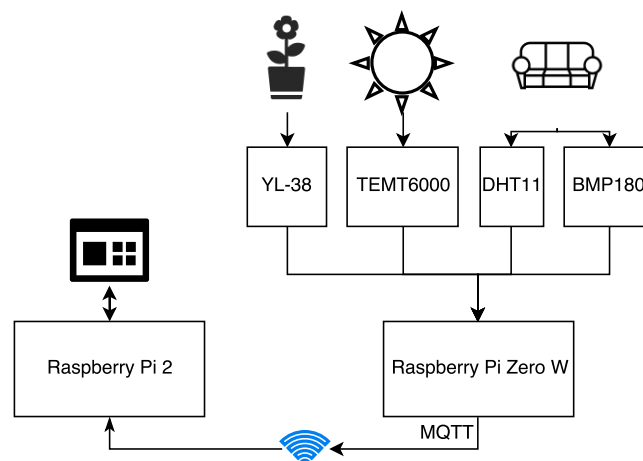
Pro konverzi analogových senzorů je použit AD převodník MCP3008.



Obrázek 1: Schéma zapojenia hardware.

3.2 Software

V tejto časti sú popísané naimplementované časti software a nástroje použité k činnosti meteostanice s autonómnym riadením. Na obrázku 3.2 je znázornené prepojenie jednotlivých modulov a spôsob ich komunikácie.



Obrázek 2: Schéma prepojenia softwareových modulov.

Interakciu s užívateľom sprostredkúva webová aplikácia, ktorá sa skladá z grafického užívateľského rozhrania (GUI) zobrazovaného vo webovom prehliadači a serverovej časti.

Webové grafické užívateľské rozhranie je naimplementované vo frameworku Angular verzie 4.

krátky popis angularu, knižnic použitých na zobrazenie grafov

GUI zobrazuje dáta prenášané zo serveru prostredníctvom REST API.

Serverová časť webovej aplikácie (backend) prijíma dáta zo senzorov a ukladá ich do databázy. Na základe dát zo senzorov a nastavených pravidiel určuje činnosť aktuátorov, a komunikuje s aktuátormi. Backend je naimplementovaný v jazyku Python. K implementácii REST API je použitý framework Flask.

krátky popis flasku

REST API umožňuje:

- získať poslednú vzorku dát zo senzorov vrátane časovej známky spolu s trendom vypočítaným pre každú meranú veličinu
- získať informácie o aktuálnom stave aktuátorov a pravidlách, ktorými sú riadené

Výpočet trendov prebieha na základe lineárnej aproximácie z určitého počtu posledných vzoriek danej veličiny. Pre každú veličinu je tak určená funkcia v tvare:

$$y = ax + b$$

Koeficient a udáva ako prudko funkcia klesá alebo stúpa. Koeficient b udáva posunutie funkcie po osi x . Toto posunutie nemá na zmenu veličiny vplyv a preto môže byť z ďalších výpočtov vypustené. Do vzťahu:

$$y = ax$$

je potom za x dosadený počet vzoriek, z ktorých sa počíta trend. Hodnota y udáva o koľko sa daná veličina za posledných x vzoriek zmenila. Pre každú veličinu boli empiricky určené hodnoty prahov. Ak je absolútna hodnota zmeny veličiny väčšia ako prah, podľa znamienka zmeny veličiny je určený stúpajúci alebo klesajúci trend, inak je určený trend rovnomerného priebehu.

Pre ukladanie dát je použitý databázový systém SQLite3, ktorého výhodou je jeho jednoduchosť a štandardná podpora v jazyku Python. Databázové dáta sú uložené v jedinom súbore. V databáze sa nachádzajú 3 tabuľky:

- **weather_data** - obsahuje históriu dát zo senzorov
- **actuators** - obsahuje informácie o stave aktuátorov
- **thresholds** - obsahuje informácie o pravidlách riadiacich činnosť aktuátorov

Dáta zo senzorov získava, agreguje a následne odosiela backendu modul **sensors**. Pre testovacie účely bol pripravený aj modul **fake-sensors**, ktorý simuluje činnosť modulu **sensors** a odosiela umelé dáta v rovnakom formáte.

Modul **fake-actuators** simuluje činnosť aktuátorov. Z backendu prijíma príkazy na zmenu stavu aktuátorov, ktoré simuluje. Po prevedení činnosti aktuátoru je odoslaná informácia o stave, v ktorom sa aktuátor nachádza backendu. Nasleduje zoznam simulovaných aktuátorov:

- klimatizácia - riadená na základe údajov o teplote
- kúrenie - riadené na základe údajov o teplote
- žalúzie - riadené na základe údajov o intenzite svetla
- zavlažovanie rastlín - riadené na základe údajov o vlhkosti pôdy

Komunikácia medzi backendom, senzormi a aktuátormi prebieha cez protokol MQTT.

krátky popis MQTT

Ako broker je použité Mosquitto.

4 Výsledky

5 Závěr