

Semestrální práce KKY/ITE

Green team: Jana Romová , Jan Šimek , Petr Železný

Cíle

Cílem semestrální práce bylo:

- Naprogramovat **teplotní čidlo** na čipu **ESP8266** tak, aby bylo schopné odesílat naměřené teploty do sítě, a zprovoznit jej (plus pouzdro jako bonus)
- Zprovoznit **virtuální stroj**, který bude hostovat **webový server** a uchovávat naměřená data
- Zprovoznit komunikaci mezi čidlem a **MQTT** brokerem, který poskytuje data všem zapojeným virtuálním strojům
- Odesílat data přes **HTTP** protokol přímo zákazníkovi (společnost Aimtec)
- Vytvořit **webové rozhraní**, které bude v přehledné formě zobrazovat naměřená data ze všech teplotních čidel

Teplotní čidlo

Teplotní čidlo měří a zpracovává teplotu, kterou následně odesílá pomocí **MQTT**. Tvoří ho čip **ESP8266** a teplotní senzor **DS18B20**.

Po spuštění čipu proběhne připojení na předem nastavenou wifi. Následně se aktualizuje čas na UTC a vytvoří client **MQTT**, který se připojí na broker. Poté začne čidlo měřit teplotu. To probíhá tím způsobem, že si načte a uloží teplotu každou sekundu, a jednou za minutu zprůměruje uložené hodnoty, odešle je na broker a smaže je. Častější měření a průměrováním omezuje čidlo odeslání chybné hodnoty způsobené senzorem nebo komunikací senzor-čip.

V případě, že nastane někde chyba, se čidlo restartuje, což znovu zahájí připojení na wifi a **MQTT** broker.

Virtuální stroj

Virtuální stroj přijímá naměřená data všech týmů z brokeru přes **MQTT**. Jednak je dává k dispozici pro zobrazení na webové stránce a jednak odesílá data našeho týmu přímo do společnosti Aimtec, obojí pomocí technologie **HTTP**.

Virtuální stroj je hostován na platformě OpenNebula a obsluhuje jej operační systém **Debian 11**. Je přístupný prostřednictvím známé IP adresy buď z univerzitní sítě (Eduroam) nebo skrz univerzitní VPN síť. K přihlašování ve Windows používáme PuTTY nebo Ubuntu ve [WSL](#)¹ (doporučuji vyzkoušet a zmínit i v rámci ITE, vyžaduje to víc místa na disku, ale funguje to hodně dobře).

¹ Windows Subsystem for Linux

Na virtuálním stroji běží pomocí **tmux** python script **websockets.py**. Ten řídí běh webového serveru pomocí knihovny **tornado**. Pro správný běh aplikace definované třídou **WebWSApp(TornadoApplication)**, bylo třeba upravit nastavení **Tornado aplikace** a upravit nebo importovat některé handlers.

Použité handlers:

- **BaseHandler** - spojuje všechny ostatní handlers,
- **LoginHandler** - načítá informace **cookie** z webového rozhraní,
- **MainHandler** - kontroluje informace o uživateli a při přihlášení přesměruje uživatele na stránky s grafy,
- **RecieveImageHandler** - při trénování rozpoznávání obličeje přijímá fotky z kamery a ukládá je na virtuální úložiště,
- **WSHandler** - obsluhuje klienty serveru,
- **StaticFileHandler** - umožňuje aplikaci přístup do složek na virtuálním stroji,
- **RecognizeImageHandler** - importován z externího skriptu **recognize_handler.py**, vyfotí uživatele a přiřadí mu konkrétní identitu s pravděpodobností správného určení.

Obě tyto úpravy se předávají aplikaci ve třídě **WebWSApp(TornadoApplication)** (**self.tornado_settings**, **self.tornado_handlers**), následně se aplikaci vytváří vlastní vlákno. Aplikace také obsahuje mechanismus pro přijímání zpráv přes **MQTT**. Pokud přijde nová zpráva, data rozdělena na teplotu a čas se přidají do svých polí přiřazených konkrétním týmům (barvám). Následně se pošlou obnovená data za uplynulý den na webové rozhraní, pokud byla zpráva od týmu green pošle se zpráva i na API **AIMTECHU**.

Každých 5 minut se všechna data uloží do zálohovacího souboru **data.txt** na virtuálním stroji. Při připojení nového klienta se data pošlou na jím zobrazenou webovou stránku, díky tomu nedochází k prodlevě při načtení dat. K tomu se využívá pomocná třída **Pripoj()**, která si drží informaci o posledním připojeném klientovi.

V hlavní metodě skriptu se pak pouze vytvářejí pole reprezentující jednotlivé týmy, načítají se do nich data uložená během předešlých spuštění. Jako poslední se inicializuje aplikace **tornado** třídou **WebWSApp(TornadoApplication)**, neboli spuštění **https serveru**. Připojení k serveru je zajištěno smyčkou **IOLoop**.

Webové rozhraní

Webové rozhraní je přístupné na webové adrese <https://sulis95.zcu.cz/>. Nejprve se zobrazí přihlašovací stránka, na které se musí uživatel autentifikovat prostřednictvím skenu obličeje pomocí kamery počítače, mobilního telefonu, tabletu nebo jiného zařízení.

Po přihlášení se zobrazí úvodní stránka, která se skládá z navigační lišty a přehledu grafů všech zapojených čidel. K vizualizaci grafů jsme použili Javascriptovou knihovnu [Chart.js](#), která je jednoduchá na použití a vizualizace jsou na úrovni.

Další design webové stránky byl tvořen pomocí **CSS** knihovny [Bootstrap](#) verze 5, která již funguje plně v **JavaScriptu** (předchozí verze využívaly ještě knihovnu jQuery). JavaScript rovněž obsluhuje všechny dynamické funkce webových stránek.

Navigační lišta obsahuje odkazy na stránky jednotlivých čidel. Každé čidlo má svou samostatnou stránku, jejíž první polovinu tvoří tabulka statistik a druhou polovinu širokoúhlý graf, ze kterého se lépe odečítají jednotlivé naměřené hodnoty.

Webové rozhraní by mělo bez větších problémů fungovat ve všech běžně používaných webových prohlížečích podporujících **HTML5**, a to na displejích a monitorech různých velikostí a rozlišení.

Webové stránky se skládají ze souborů:

- HTML
 - main.html - přihlašovací stránka
 - index.html - stránka s přehledem všech grafů
 - <color>.html - stránky jednotlivých teplotních čidel s grafy a statistikami
 - vyvojari.html - stránka se představením vývojářů a kontakty (spíš easter egg)
- JavaScript
 - capture.js - rozpoznávání obličeje
 - graf.js - vykreslování grafů
 - script.js - websockety
- CSS
 - main.css - kaskádové styly přihlašovací stránky
 - style.css - kaskádové styly všech ostatních stránek

Zabezpečení

Přihlašování na virtuální stroj

K přihlašování na virtuální stroj využíváme **SSH klíče**. Ty jsme si nechali vygenerovat a nakonfigurovat pomocí příkazů `ssh-keygen` a `ssh-copy-id` v prostředí Linuxu (podle návodu [zde](#)). Jako pojistku máme nastavená klasická hesla, ale pokud by se jednalo o nějaký prvek důležité infrastruktury (ne o školní úlohu), uvažovali bychom o jejich úplné deaktivaci.

Přihlašování na webové stránky

K přihlašování na webové stránky používáme knihovnu pro rozpoznávání tváří **OpenCV**. Uživatel se nejprve nechá rozpoznat kliknutím na tlačítko "Rozpoznat" a teprve potom, co zjistí, že byl rozpoznán s dostatečnou přesností (máme nastaveno 80% a daří se nám přesnosti během několika pokusů dosáhnout), může se přihlásit kliknutím na tlačítko "Přihlásit se".

Fungování systému ze strany virtuálního stroje zmíněno výše. Na straně webového rozhraní je problematika řešena formulářem ve skriptu **main.html**, který posílá data tornado aplikaci.