

Vysoké učení technické v Brně

Fakulta informačních technologií



Mikroprocesorové a vestavěné systémy

Dokumentace k projektu

ESP8266: ovládání LED

16.12.2020

Daniel Pátek (xpatek08)

Obsah

1.	Úvod	4
2.	Příprava	4
3.	Implementace	5
3.1.	ESP8266.....	5
3.2.	Android aplikace.....	6
4.	Výsledné aplikace	7
5.	Odkaz na demonstrační video	7
6.	Autoevaluace výsledků projektu	7
7.	Závěr.....	8
8.	Zdroje	9

1. Úvod

Tato dokumentace si klade za cíl seznámit čtenáře o průběhu řešení školního projektu do předmětu IMP.

Cílem projektu je implementovat vestavný systém, který umožní pomocí wifi připojení ovládat LED diody připojené k modulu ESP8266. Součástí projektu je také implementace mobilní aplikace, pomocí které se budou LED diody zapínat a vypínat. Wifi modul zařízení ESP8266 bude fungovat v režimu AP, aby se k němu mohla připojit další zařízení. Z aplikace bude také možné spustit sekvenci blikání.

2. Příprava

Pro programování zařízení ESP8266 jsem zvolil vývojové prostředí Arduino. Pro správnou funkčnost je zapotřebí jednak doinstalovat model desky do tohoto vývojového prostředí [\[1\]](#) a jednak nainstalovat správný ovladač pro obsluhu portu ve Windows 10. Následně jsem zjistil, že ovladatelné diody jsou připojeny na piny D2 – D4, což odpovídá hodnotám 2 – 4 použitých v tomto IDE. Pro obsluhu sítě wifi a http serveru použiji knihovnu `ESP8266WiFi.h` [\[2\]](#).

Pro vývoj mobilní aplikace jsem zvolil framework `Flutter` s programovacím jazykem `Dart` [\[3\]](#) a prostředí Visual Studio Code. Abych byl schopný posílat http požadavky a přijímat odpovědi, využil jsem knihovnu `http` [\[4\]](#).

3. Implementace

3.1. ESP8266

Pro implementaci byl použit programovací jazyk `Wiring` [5], který je používán v rámci vývojového prostředí Arduino.

Základem programu jsou dvě funkce, `setup()` a `loop()`. První má na starosti nastavení zařízení a druhá funguje jako nekonečná smyčka a obsahuje funkčnost programu.

Program obsahuje globální proměnné, byly využity jednak pro výstupní piny připojených LED diod, jednak pro proměnné `server` a `request` (viz dále).

Ve funkci `setup()` je zapotřebí otevřít zmíněné piny, to zajišťuje funkce `pinMode()` [6], která piny nastaví na možný výstup. Dále je třeba zapnout wifi připojení v režimu AP. Knihovni funkce `softAP()` [2] tento požadavek plní a umožní také nastavení SSID a hesla. Po úspěšném startu wifi připojení zbývá jen zapnout jednoduchý HTTP server na portu 80, který je součástí knihovny Arduino [7] a v projektu je vytvořen jako globální proměnná.

Druhá část programu je obsažena ve funkci `loop()`. Ta začíná kontrolou, zda některý z klientů serveru má data k přečtení, jinými slovy - zda někdo odeslal požadavek na server. Pokud ne, pokus se opakuje. Pokud ano, do výše zmíněné globální proměnné `request` se uloží první řádek tohoto požadavku. Následně se kontroluje, zda požadavek obsahuje významná data pro tento program, například `LED1ON` značí zapnutí první LED diody, `LED3OFF` vypnutí třetí LED diody nebo slovo `SEQUENCE` značí zapnutí sekvence.

O ovládání LED diod se stará funkce `digitalWrite()` [8]. Po dokončení je nutné danému klientovi odpovědět. Tento program používá odpověď ve formátu `HTTP/1.1 204 No Content` [9]. Následně spojení s klientem program ukončí pomocí funkce `flush()` [10].

3.2. Android aplikace

Mobilní aplikace je napsána v jazyce `Dart`. Základem aplikace je soubor `main.dart`. Zde je aplikace vytvořena a pomocí funkce `runApp()` také spuštěna.

Soubor `screen.dart` obsahuje tzv. `widget-tree` [11], definici uživatelského rozhraní aplikace. Stránka obsahuje 4 různá tlačítka. Tři z nich se starají o zapnutí a vypnutí jednotlivých LED diod, přičemž po zapnutí LED diody tlačítko změní barvu na žlutou. Čtvrté tlačítko slouží k zapnutí blikající sekvence. Při provádění této sekvence je místo tlačítka zobrazen pouze symbol načítání.

Stav aplikace je určen pomocí čtyř proměnných typu `bool`. Tyto proměnné jsou nastavované dynamicky podle toho, jaké LED diody jsou zapnuté. Takové chování určují funkce `_toggleSequence()` a `_toggleLED()` volané na základě interakce uživatele s tlačítky.

Posledním souborem je `http_helper.dart`. Obsahuje taktéž dvě funkce, které se starají o komunikaci se serverem běžícím na zařízení ESP8266. První z nich, `espLed()`, umožňuje odeslání požadavku na zapnutí nebo vypnutí určité LED diody a zpracování odpovědi. Druhá funkce `espSequence()` odesláním požadavku spustí na zařízení blikající sekvenci.

4. Výsledné aplikace

Oba programy splňují svůj účel. Mobilní aplikace dokáže úspěšně odeslat http požadavek a zařízení ESP8266 ho zase dokáže zpracovat a adekvátně na něj reagovat.

Výsledkem projektu je jednoduchý vestavný systém schopný spuštění wifi připojení v režimu AP a webového serveru. Tento systém na základě požadavků připojených klientů dokáže ovládat LED diody, které jsou připojené na pinech D2 – D4 a také s nimi umožňuje spustit blikající sekvenci.

Dalším produktem tohoto řešení je moderní mobilní aplikace, ze které je umožněno LED diody připojené k zařízení ovládat. Podmínkou je připojení na wifi síť tohoto zařízení.

5. Odkaz na demonstrační video

<https://youtu.be/ty2ree1MS1k>

6. Autoevaluace výsledků projektu

- E – Samozřejmě se do projektu dá vždy něco přidat. Já jsem splnil zadání bez nějakého velkého rozšíření, dal bych si 0,75 bodu.
- F – Projekt funguje bezchybně, dal bych si 5 bodů.
- Q – Uživatelská přívětivost by také nějak šla vylepšit, co se týče dekompozice a kvality zdrojových kódů, nevidím chybu. Proto bych si dal 2,5 bodu.
- P – 1 bod
- D – Dokumentace je podle mě kompletní, 4 body.

7. Závěr

Cílem projektu bylo implementovat vestavný systém, který umožní pomocí wifi připojení ovládat LED diody připojené k modulu ESP8266. Tento cíl byl dodržen a implementace proběhla úspěšně.

Druhým cílem byla implementace mobilní aplikace, pomocí které se tyto LED diody budou moct ovládat a bude možné také spustit blikající sekvenci. I tento cíl byl úspěšně implementován.

Oba programy byly otestovány pomocí poskytnutého hardwaru a běžného mobilního telefonu a jsou plně funkční.

8. Zdroje

- [1] <https://github.com/esp8266/Arduino>
- [2] <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
- [3] <https://flutter.dev/>
- [4] <https://pub.dev/packages/http>
- [5] [https://cs.wikipedia.org/wiki/Wiring \(programovac%C3%AD jazyk\)](https://cs.wikipedia.org/wiki/Wiring_(programovac%C3%AD_ jazyk))
- [6] <https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>
- [7] <https://www.arduino.cc/en/Tutorial/LibraryExamples/WebServer>
- [8] <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>
- [9] <https://httpstatuses.com/204>
- [10] <https://www.arduino.cc/en/Reference/ClientFlush>
- [11] <https://flutter.dev/docs/development/ui/widgets-intro>