IMP – Mikroprocesorové a vestavěné systémy

ESP8266 – Ovládání LED

Martin Bartoš (xbarto96)

21.12.2019

Obsah

Úvod.....3
Použití.....3
  Zapojení HW.....3
  Stručný popis implementace HW.....4
    Obsluha serveru.....4
    Obsluha LED.....5
  Stručný popis implementace SW.....6
    Vzhled připojené aplikace.....6
Závěr.....7

Úvod

Podle zadání bylo cílem této práce navrhnout vestavěný systém umožňující ovládat připojené LED. Pro vývoj byl použit modul NodeMCU s čipem ESP8266 a knihovnou Arduino. Zařízení bude využívat WiFi v režimu AP, na které bude možné se připojit pomocí mobilního telefonu. Aplikace byla vytvořena pomocí React-native v JavaScript a hardwarová část projektu byla naimplementována pomocí jazyka C++.

Systém by měl umožňovat:

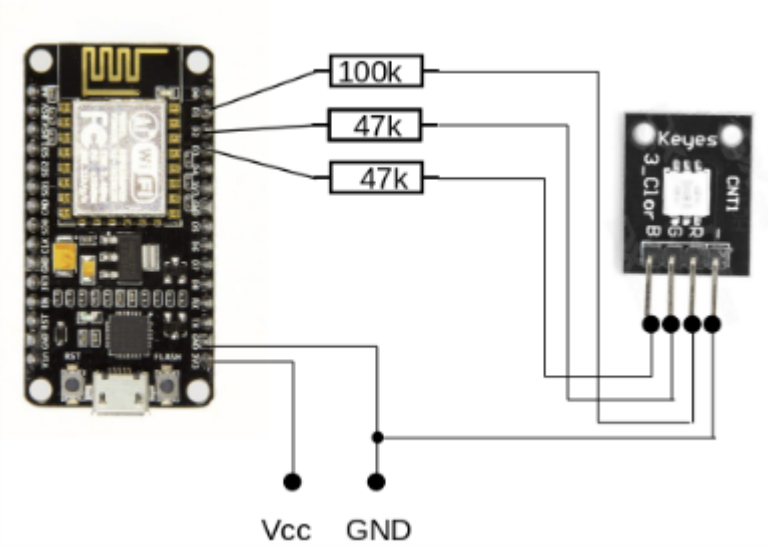
- Za všech okolností by se měl chovat přirozeně a rozumným způsobem ošetřeny vstupy.
- Ovládat jednotlivé světla.
- Spouštět sekvence.
- Při spuštění sekvence musí procesor na závěr informovat uživatele o dokončení sekvence.

Použití

Pro spuštění výsledných součástí tohoto projektu je potřeba vytvořena aplikace, která je přenositelná na platformách Android a iOS. Poté už jenom stačí připojit napájení 3.3V k naprogramovanému modulu NodeMCU, nebo 5V přes USB sběrnici, kde je poté také usměrněno na daných 3.3V. Poté už stačí připojit k daným pinum LED osvětlení. Pro můj projekt jsem využil RGB modul s 5050 LED.

Zapojení HW

Jak již bylo zmíněno v úvodu, byla využita RGB LED 5050. Ke každé složce z těchto 3 složek RGB diody byl přidán předřadný odpor. Pro výpočet odporu byl použit základní Ohmův zákon.Pro červenou složku byl vypočítán odpor 65Ω a pro zelenou a modrou 15Ω. Pro praktické zapojení byly využity odpory, které se svou hodnotou blížily vypočítané hodnotě. Tudíž pro červenou složku byl využit odpor 100Ω a pro zelenou a modrou 47Ω.



Stručný popis implementace HW

Implementace HW je řazena do dvou částí. Jedna z nich je obsluha server a ta druhá obsluha připojených LED. Je využita knihovna Arduino, která umožňuje snazší programování HW modulů. V *setup()* funkci je inicializována obsluha serveru a světel. Dále *loop()* funkce obsahuje implicitní nekonečnou smyčku, kde se v každé iteraci testují requesty přicházející na server.

Obsluha serveru

Nejdůležitější třídou použitou pro obsluhu serveru je ***ESP8266WebServer***, kde je vytvořena instance této třídy s názvem *server*. Defaultně je nastavena síť s IP 192.168.66.0/24 a adresa našeho serveru je **192.168.66.66** s defaultně nastaveným portem 8282, kde naslouchá. SSID našeho AP je ***BartosLED.*** Instance našeho serveru obsahuje metodu *on()*, která obsahuje 2 parametry.

První z nich je určení endpointu, a ten druhý je callback na funkci, která je vyvolána, když request URL odpovídá určenému endpointu. Na konci funkce *setUpServer()* je vyvolána metoda serveru *begin()*, která určuje, že konfigurace serveru je kompletní. V každém callbacku server posílá response zpět klientovi.

Např. *server.on(„/connect“,handleConnect)*

Obsluha LED

Obsluha serveru obsahuje připojené LED na jednotlivých portech modulu NodeMCU, kde první vyvolaná funkce nastaví tyto porty na výstupní a explicitní hodnota bude logická 0. Tato moje knihovna s názvem „*barLights*“ obsahuje už jednotlivé funkce, které mění stav na jednotlivých portech pomocí PWM.

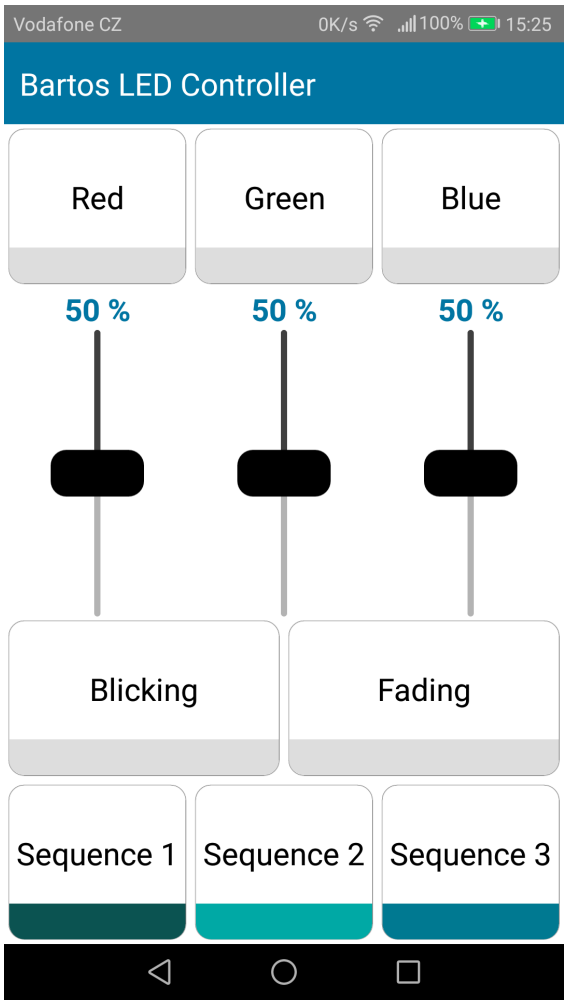
Pro použití PWM na daných portech se využívá funkce, která je zahrnuta v knihovně Arduino a to je ***analogWrite(uint8_t pin, int val)***. Tato funkce pracuje s hodnotami od 0 do 1024 a nastavuje střídu PWM signálu.

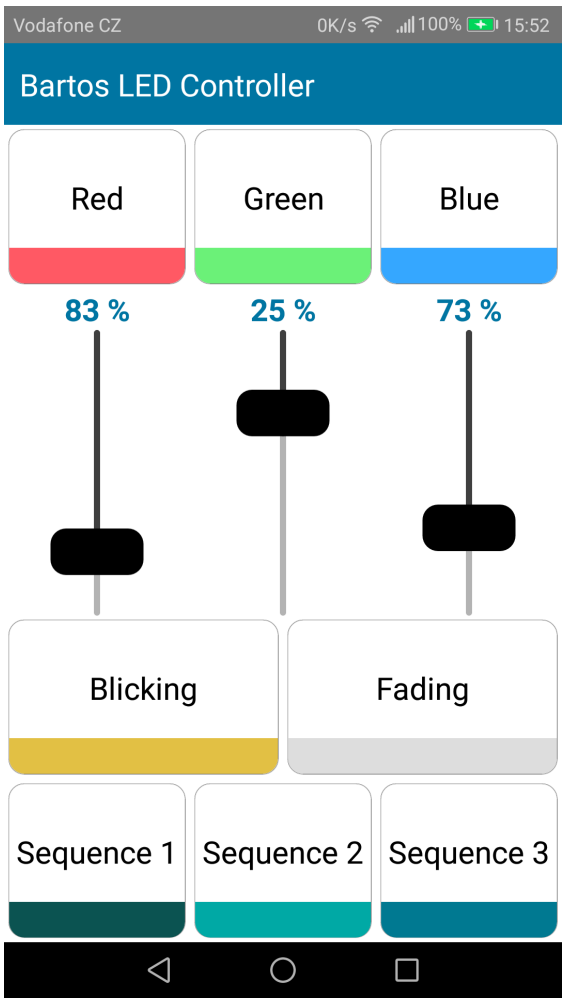
Stručný popis implementace SW

Pro tento projekt jsem nevolil žádné předchystané řešení pro mobilní aplikaci a vytvořil jsem svoji vlastní pomocí react-native. Tato jednoduchá aplikace obsahuje sadu komponent, které mění stav requestu, který bude poslán na server. Hlavní částí aplikace je komponenta App.js, která obsahuje a renderuje ostatní komponenty ve stromové struktuře. Také obsahuje funkce, které se starají o poslání requestu a následnému přijmu response ze serveru.

Při stisku tlačítek se vygeneruje rozdílný request, který je zaslán na server. Zda se jedná o sekvenci, tak aplikace zobrazí modální okno, které se zavře při přijmutí response ze serveru. Server zasílá response vždy po dokončení sekvence. Pro zajímavost, aplikace obsahuje 2 tlačítka s názvem *Blicking* a *Fading*. To určuje mód zapnutí jedné ze 3 složek RGB LED diody. Může být zapnutá vždy jedna z těchto možností a když není vybraný žádný mód, tak je světlo klasicky rozsvíceno.

Vzhled připojené aplikace





Závěr

Tato aplikace obsahuje vše, co bylo popsáno v zadání a je 100% funkční. Jelikož jsem pracoval na vývoji mobilní aplikace poprvé, tak jsou známé nějaké nedostatky, co se týče UX a malé chyby, co se týče UI. Ale podle všeho, myslím si, že cílem tohoto projektu je vyzkoušet si naprogramovat svoje embedded zařízení, kde je hlavní práce s HW. Jako rozšíření jsem implementoval módy pro rozdílné spuštění těchto LED diod. Tento projekt pro mě byl velice přínosný, co se týče HW i SW stránky projektu.