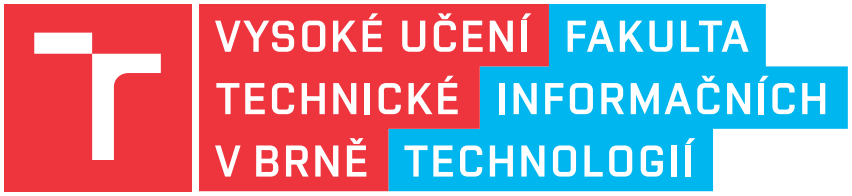


IMP – Mikroprocesorové a vstavané systémy

Demonštrácia ovládania TFT displeja cez WiFi/Bluetooth

14. decembra 2022

Patrik Sehnoutek (xsehno01)

Obsah

1 Úvod

Cieľom projektu je demonštrovať komunikáciu jadra vývojového kitu obsahujúceho SoC ESP32 s periferným zariadením v podobe farebného TFT displeja pripojeného cez SPI rozhranie. Pre tento projekt bola zvolená vývojová doska `Wemos D1 R32`[4]. Z dvoch dostupných možností komunikácie cez rohrania WiFi alebo Bluetooth, som si zvolil komunikáciu cez WiFi.

2 Návrh

2.1 Vývojové prostredie

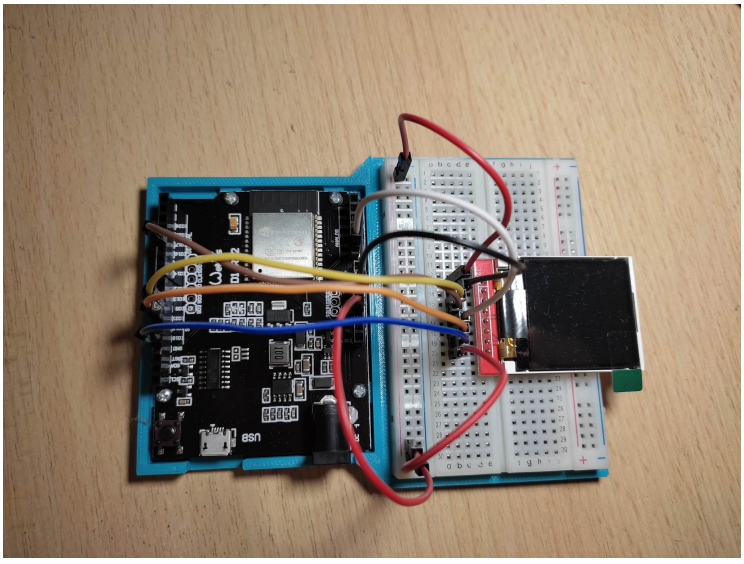
Ako vývojé prostredie som zvolil `VSCode` s rozšírením `PlatformIO`. Toto rozšírenie mi umožnilo nahrávanie kódu do vývojového kitu.

2.2 Použité knižnice

- **<Adafruit_GFX.h>** - základná grafická knižnica [2]
- **<Adafruit_ST7735.h>** - hardwarová špecifická knižnica pre ST7735 [1]
- **<SPI.h>** - rozhranie displeja
- **WiFi.h** - rozhranie WiFi
- **WebServer.h** - vytvorenie webového serveru

2.3 Zapojenie

Na obrázku 1 je možné vidieť zapojenie TFT displeja cez SPI rozhranie. Podrobné zapojenie portov je znázornené v tabuľke 2.3.



Obr. 1: Pripojenie TFT displeja cez SPI rozhranie

TFT Displej	vývojový kit s ESP32
LED	3v3
SCK	I018
SDA	I023
DC	I02
RESET	I017
CS	I05
GND	GND
VCC	3V3

Tabuľka 1: Zapojenie portov TFT displeja

3 Implementácia

Projekt som implementoval v jazyku C++ za pomoci knižníc viz 2.2. Každý kód musí obsahovať dve funkcie `setup` a `loop`.

Na úvod je potrebné správne namapovanie portov na základe zapojenia. Následne vytvorím dve globálne premenné. Jednou je inštancia triedy displeja a druhou je web server. Vo funkcii setup je na úvod potrebné nastaviť prenosovú rýchlosť, v mojom prípade 115200 Bd. Po nastavení prenosovej rýchlosti je potrebné inicializovať displej pomocou metódy `tft.initR`. Po inicializácii je potrebné pripojiť vývojovú dosku na WiFi. Pripojenie na WiFi prebieha vo funkcii `connectToWifi`. Vo funkcii sa nastaví režim WiFi na `WiFi_STA`, tj. režim stanice. Režim stanice znamená, že sa ESP pripojí na nejaký prístupový bod. Prístupový bod je definovaný pomocou identifikátora služby `ssid` a hesla `password`. Keď je vývojový kit pripojený, je potrebné nastaviť routingy. Nastavenie routingov sa nachádza vo funkcii `setUpRoutings`, kde sa vytvoria 3 routingy [3]. Jeden slúží ako úvodná stránka, ďalšie dva sú typu POST a používaju sa na spracovania požiadavkov od klienta. Posledným krokom vo funkcii `setup` je spustenie servera.

Vo funkcii `loop` sa odchytávajú požiadavky od klienta pomocou metódy servera `handleClient`. Na úvodnej webovej stránke sa nachádzajú dva formuláre, ktoré posielajú POST požiadavky, ktoré spracováva vyššie spomínaná funkcia `handleClient`. Prvý formulár slúži na zadanie textu a následného vypísania textu na TFT displeji. Druhý formulár umožňuje vykresliť pixelový obrázok a zvoliť si jeho farbu z predvoleného výberu farieb. Displej má rozlíšenie 128x128 pixelov. Pre užívateľské rozhranie by nebolo prívetivé zobraziť mriežku 128x128 a preto som sa rozhodol vytvoriť mriežku 16x16. V tomto prípade jedno políčko v mriežke reprezentuje plochu 8x8 pixelov na displeji. Na vykreslovanie slúži funkcia `drawAction` s pomocou funkciou `drawPixel`. Mriežka vo webovom rozhraní je zložená z html prvkov `checkbox`. Každý checkbox má vo svojom názve určenú pozíciu v 2D mriežke. Na základe tejto pozície sa vykresľujú pixely na správne miesto na TFT dipleji.

4 Použitie

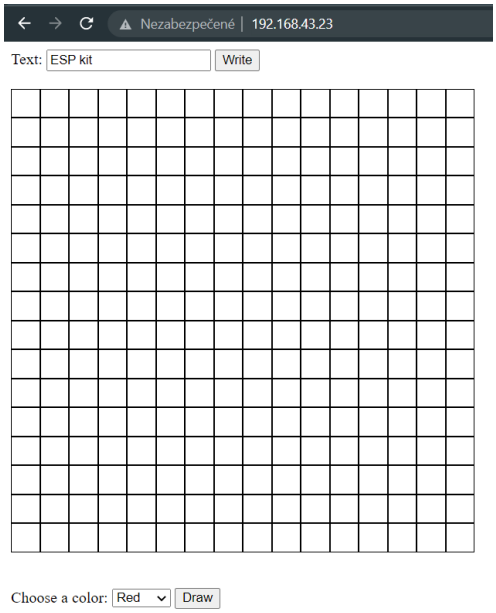
Podrobné použitie aplikácie je zhrnuté v komentovanom videu. Vo videu je ukázané ako pracovať s aplikáciou zo strany užívateľa. Užívateľ môže zadávať textové vstupy alebo pixovelový obrázok prostredníctvom jednoduchej webovej stránky.

5 Testovanie

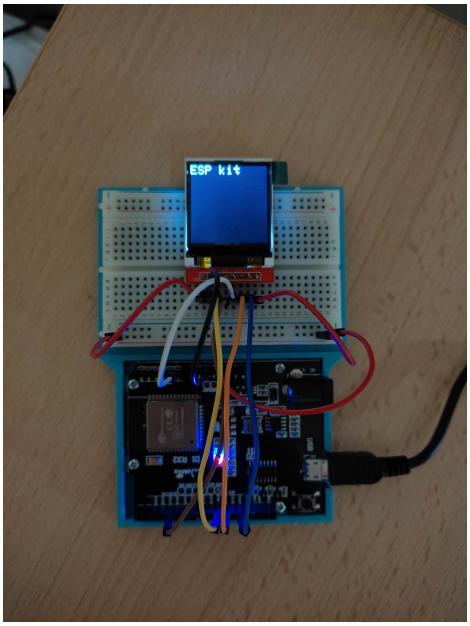
Na účely testovania som pripojil vývojový kit s ESP32 aj notebook na rovnaké WiFi rozhranie. Následne som v prehliadači otvoril webovú stránku s lokálnou IP adresou, na ktorej bežal server z vývojového kitu. Dáta som posielal pomocou formulárov prostredníctvom POST požiadavkov.

5.1 Test č.1 - výpis textu

Do prvého formulára na zadávanie textu som napísal mnou zvolený text a odoslaním formulára sa poslal požiadavok na sever. Na serveri sa požiadavok spracoval a následne sa na TFT displeji zobrazil zadaný text.



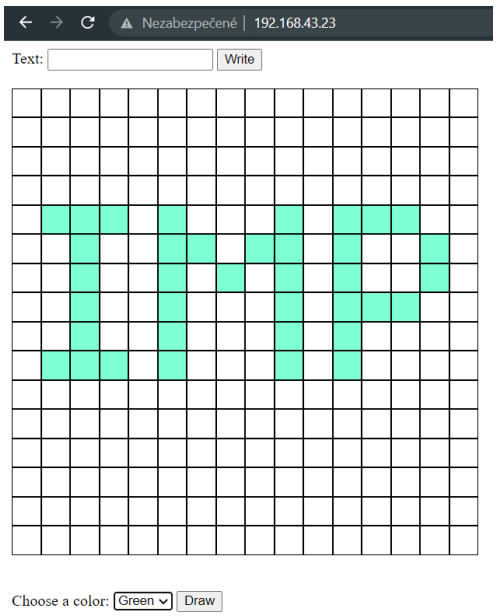
Obr. 2: Web stránka



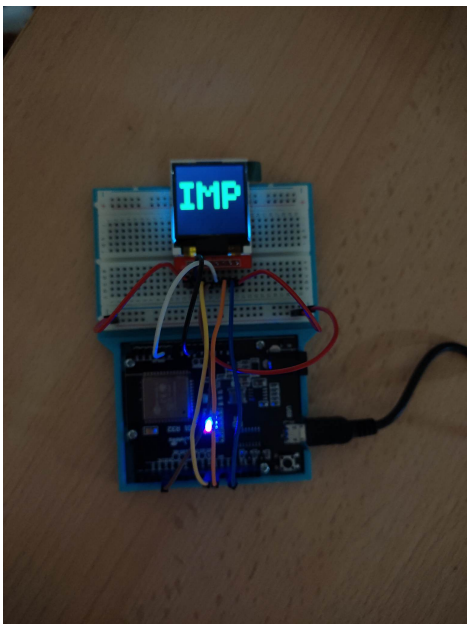
Obr. 3: TFT Displej

5.2 Test č.2 - vykreslenie obrázku z pixelov

V druhom formulári som vyklikal políčka, ktoré som chcel vykreliť na TFT displeji. V selecte som zvolil farbu, ktorou som chcel pixelový obrázok vykresliť a následne som poslal požiadavok na server. Server spracoval požiadavok a na základe dát, ktoré mu prišli zvolil farbu a umiestnenie pixelov.



Obr. 4: Web stránka



Obr. 5: TFT Displej

6 Záver

Počas testovania bola overaná funkčnosť aplikácie, pretože pri zadaní rôznych vstupov vo formulároch na webovej stránke a odoslaní formulára sa požadovaný obsah správne vykreslil na TFT displeji.

V rámci tohoto projektu vznikla aplikácia, ktorá slúži na komunikáciu cez WiFi medzi vývojovým kitom s ESP32 a webovým rozhraním prostredníctvom požiadavkov. Výsledky spracovaných požiadavkou sú vykreslované na TFT displeji.

7 Literatúra

[1] Limor, F.: Adafruit-ST7735-Library. [Source code], 2019, [vid. 2022-12-10].
Dostupné z: https://github.com/adafruit/Adafruit-ST7735-Library/blob/master/Adafruit_ST7735.h

[2] Limor, F.: Adafruit-GFX-Library. [Source code], 2020, [vid. 2022-12-10].
Dostupné z: https://github.com/adafruit/Adafruit-GFX-Library/blob/master/Adafruit_GFX.h

[3] Microcontrollerslab.com, T.: ESP32 Rest API Web Server GET and POST Examples with Postman API. [online], [vid. 2022-12-11].
Dostupné z: https://microcontrollerslab.com/esp32-rest-api-web-server-get-post-postman/

[4] Šimek, V.: IMP_projekt_board_ESP32_Wemos_D1_R32.pdf. [online], [vid. 2022-12-10].
Dostupné z: https://www.fit.vutbr.cz/~simekv/IMP_projekt_board_ESP32_Wemos_D1_R32.pdf