



IMP – PROJEKT

19. Demonstrace ovládání TFT displeje přes WiFi/Bluetooth

Jakub Kuzník
xkuzni04

8. prosince 2022

Obsah

1	Úvod	2
2	ESP32	2
3	Schéma zapojení	3
4	Implementace	3
5	SPI komunikace	5
6	WiFi	5
7	Závěr	5

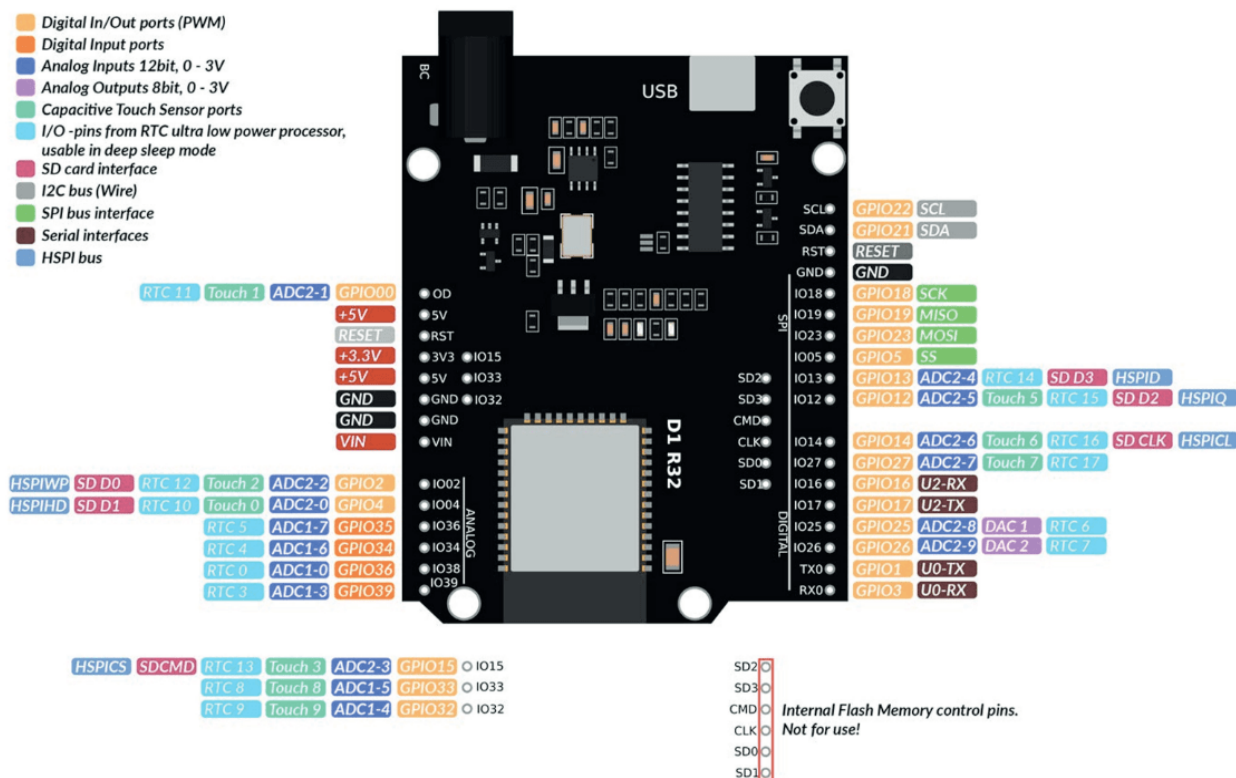
1 Úvod

Cílem projektu je zapojit a zprovoznit TFT (thin-film transistor) displej, který komunikuje přes rozhraní SPI (Serial Peripheral Interface) na zařízení ESP32 a demonstrovat jeho funkčnost pomocí Wifi či Bluetooth. V našem případě budeme demonstrovat funkčnost displeje pomocí Wifi a to tak, že na ESP32 poběží webový server s jednoduchou webovou stránkou přes, kterou můžeme změnit aktuální zobrazení displeje. Pro připojení na webový server zařízení ESP32 rovněž vystupuje v roli AP (Access Point).

2 ESP32

Pro náš projekt jsme využili mikrokontrolér ESP32 konkrétně model Wemos D1 R32. To jaké nám dané zařízení poskytuje piny lze vidět na obrázku číslo 1. Zařízení ESP32 nám poskytuje GPIO piny pro práci s SPI a dokáže vytvářet Wifi nebo bluetooth spojení, což potřebujeme pro tento projekt [1].

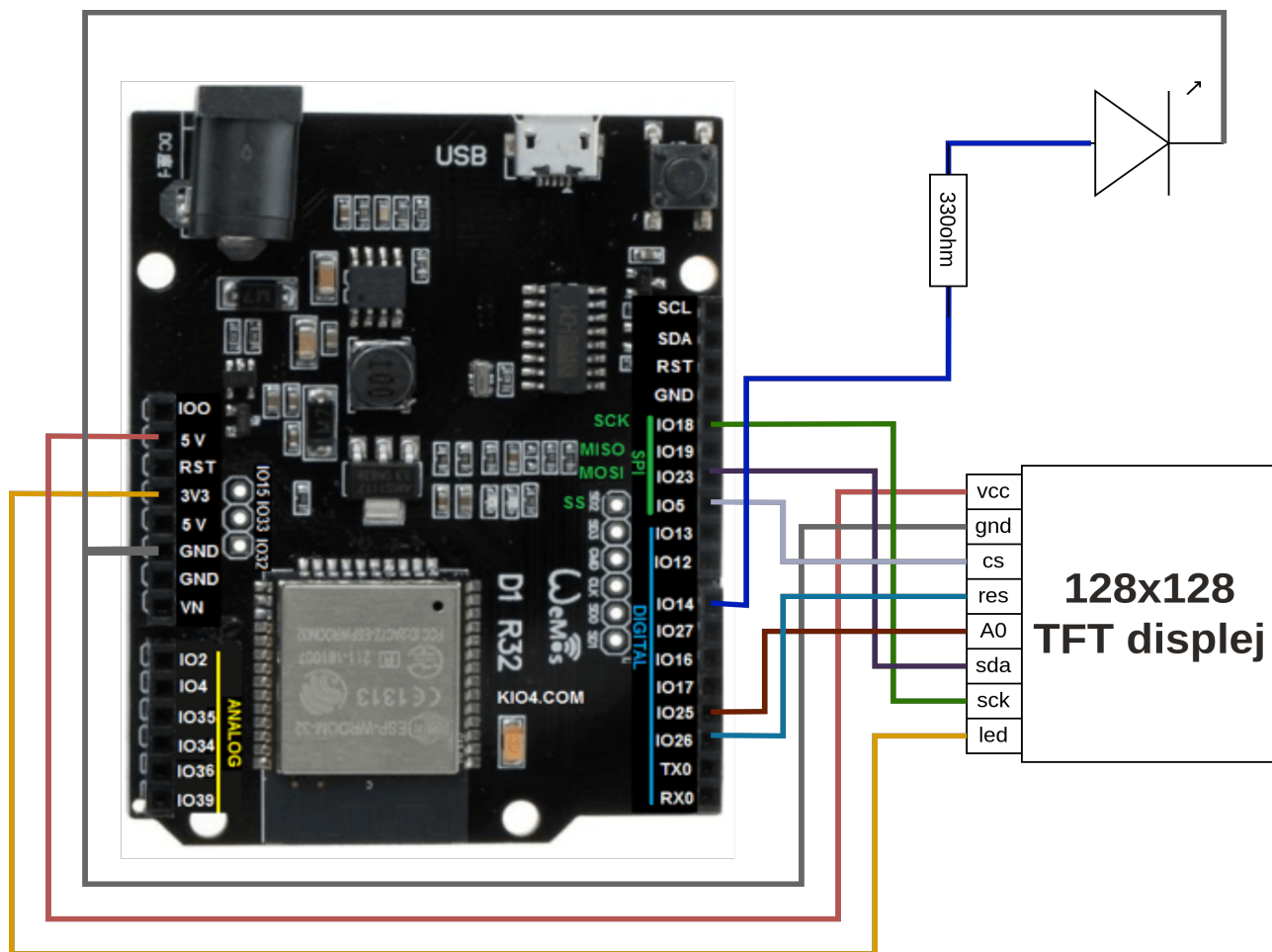
D1 R32 Board Pinout



Obrázek 1: ESP32

3 Schéma zapojení

Schéma našeho zapojení je na obrázku 3. Máme zde dvě komponenty a těmi jsou výše zmíněný TFT barevný display o rozlišení 128x128 pixelů a indikační led. Display má dva napájecí piny, těmi jsou pin led, který využívá 3.3V a pin vcc, ten potřebuje 5V. Dále jako dioda, tak i display potřebují připojenou zem (u displeje port gnd a led katoda). Pro propojení s SPI kanálem jsme využili zeleně zvýrazněné piny IO18, IO23 a IO5. Na pin IO18 se připojil hodinový signál sck. Pin IO23 slouží pro sda probíhající tam komunikace MOSI (Master output slave input). MISO (Master input slave output), pak vůbec nepotřebujeme, protože komunikace je jednostranná od mikrokontroléru k displeji. A na port IO5 jsme zapojili pin cs, což nám reprezentuje SS (slave select). Dále zde jsou piny A0 a Res připojeny na IO25 a IO26. Pro diodu jsme vyhradili pin IO14.



Obrázek 2: Schéma zapojení

4 Implementace

Náš projekt jsme implementovali v prostředí Arduino IDE pomocí jazyka C a zároveň jsme využili knihovny, které Arduino poskytuje. těmi jsou:

Adafruit_GFX.h - knihovna pro práci s grafikou displeje.

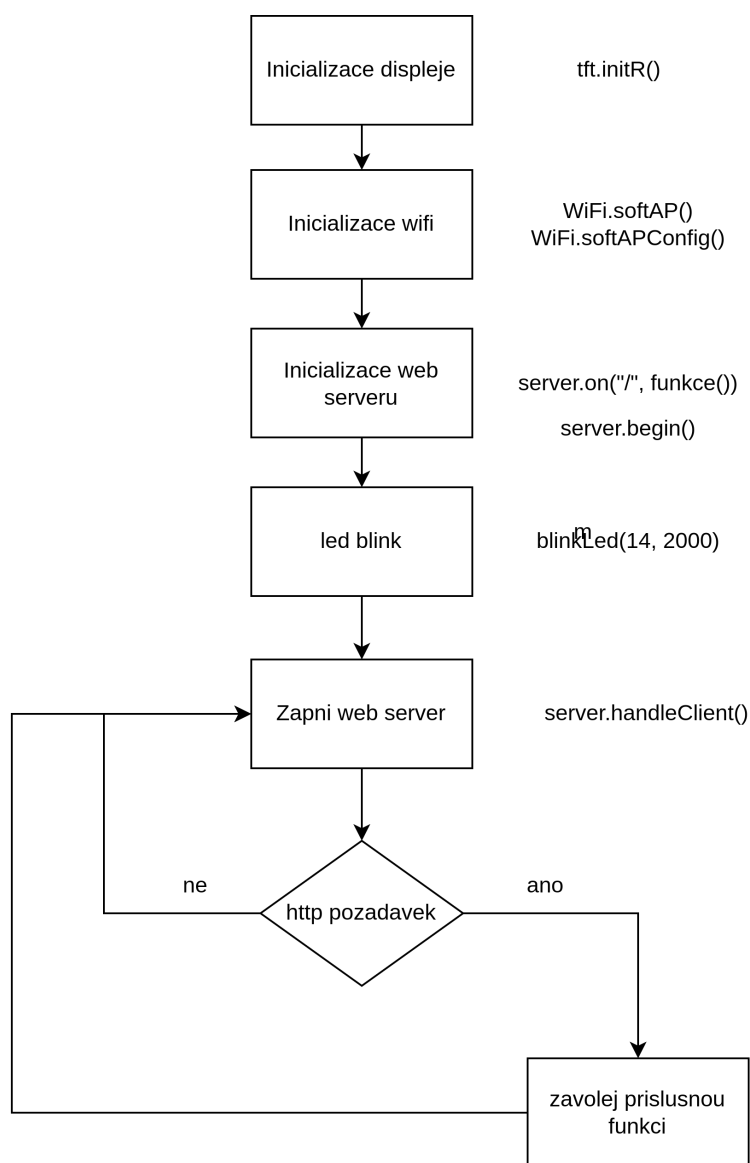
Adafruit_ST7735.h - knihovna pro práci s naším konkrétním displejem.

SPI.h - knihovna pro práci s SPI rozhraním.

WiFi.h - knihovna pro vytváření Wifi

WebServer.h - knihovna pro vytvoření webového serveru.

Logika programu je velmi jednoduchá. Na začátku inicializujeme Wifi, webový server, displej (především inicializace SPI kanálu) a led. Po inicializaci dostane uživatel signál a to takový, že bude led na 2 sekundy svítit, což znamená úspěšnou inicializaci. Program pak vytvoří Wifi, tedy zařízení bude vystupovat v roli AP (access point) na síti 192.168.1.0/24, kde se může libovolné jiné zařízení připojit. Adresa 192.168.1.1 je rezervovaná pro mikrokontrolér všechny ostatní adresy se přidělují přes DHCP, které je vytvořeno pomocí knihovny `WiFi.h`. Na ip adrese mikrokontroléru je pak dostupná webová stránka, kterou program vytvořil pomocí funkce `generateHTML()` a požadavky na tento web se zpracovávají pomocí dalších funkcí programu a hlavně díky knihovny `WebServer`. Uživatel si za předpokladu, že je připojen k Wifi tohoto zařízení, může měnit stav displeje pomocí tlačítek. Obrazce na displej jsou vykreslovány za pomoci knihovny `Adafruit_GFX.h` a data jsou pak přeneseny pomocí SPI MOSI kanálu. Tuto logiku můžeme vidět na následujícím algoritmickém schéma.



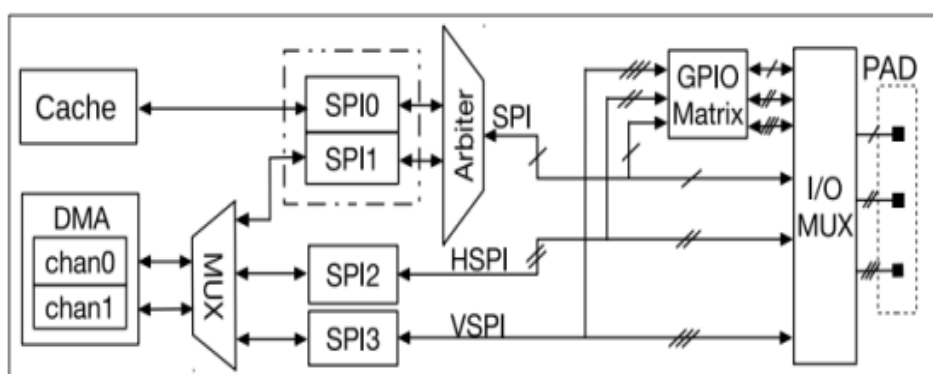
Obrázek 3: Algoritmické schéma

5 SPI komunikace

SPI je sériové periferní rozhraní, které umožňuje připojit více zařízení na jeden kanál. Vybrání zařízení se, kterým se bude komunikovat se dělá pomocí pinu SS (slave select). SPI je technologie na principu master slave. Master je v našem případě mikrokontrolér ESP32 a slave je TFT displej.

ESP se dá využít v obou rolích, jak slave tak master a jeho SPI schéma můžeme vidět na obrázku číslo 5. SPI0 je zde používán jako buffer pro přístup k externí paměti (cache). SPI1 je používán jako Master, SPI2 a SPI3 pak mohou fungovat jako master nebo slave. ESP může zpracovat více CS signálu, a tedy může mít připojených více slave najednou, například více displejů. SPI má signal BUS, který umí zpracovat signály D, Q, CS0-CS2, CLK, WP a HD.

Naše ESP je nastaveno v režimu master. Pro posílání dat na displej používáme signál 0x6, který zapíše data do GP-SPI bufferu a pak je pomocí odešle na MOSI pin [1].



6 WiFi

Pro to aby WiFi fungovala musí hodinový signál APB_CLK využívat signál PLL_CLK jako svůj zdrojový hodinový signál. To se zajistí nastavením příznaku DPORT_WIFI_CLK_EMAC_EN v registru 5.19. Mac adresa wifi je pak uložena v registru WIFI_MAC_Address. Wifi zapneme pomocí nastavení příznaku RTC_CNTL_WIFI_PD_EN v registru 31.28 [1].

7 Závěr

Táto práce poskytuje řešení pro TFT SPI displej ovládaný skrze zařízení ESP32. Ovládání je zprostředkováno pomocí technologie WiFi. Projekt poskytuje dobrou jednoduchou programovou šablonu v jazyce C, která může být dále rozšířena.

Subjektivní sebehodnocení: Práce má velmi přehledný a dobrý zdrojový kód s slušnou dokumentací. Pohlál jsem si zajímavým vykreslováním objektů na displej. Ale použil jsme knihovny, které mi celou práci trochu ulehčily.

Reference

- [1] ESP32 Technical Reference Manual Version 4.7 Espressif Systems [online]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf