

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy

Demonstrace ovládání TFT displeje přes WiFi/Bluetooth

1 Úvod

Cieľom projektu bolo demonštrovať komunikáciu ESP32 s TFT displejom, ktorý by sa mal dať ovládať cez bezdrôtové rozhranie Wifi alebo Bluetooth.

Projekt bol implementovaný s použitím nasledujúcich prostriedkov:

Hardware:

- Doska WeMos D1 R32 založená na čipe ESP32
- Displej SSD1306 SPI 128*64
- Prepájacie periférie (Breadbord a káblíky)

Bolo použité odporúčané zapojenie:

VCC - 3.3V; GND - GND; D0 (SCLK) - IO18, D1 (MOSI) - IO23, CS - IO5, DC (DATA/control) - IO27, RESET - IO17

Software:

- Platforma PlatformIO vstavaná do editoru VScode

Komunikácia:

Komunikácia medzi doskou a displejom prebieha cez periférne rozhranie SPI (Serial Peripheral Interface). Na úrovni kódu som priamo cez toto rozhranie nekomunikoval, namiesto toho som používal funkcie z knižnice, ktorá implementovala základné prostriedky priamo pre prácu s konkrétnym použitým displejom¹. Táto knižnica implementuje funkcie ako napríklad zobrazenie textu alebo obrázku na displej, a tým nás úplne odprúšuje od práce s SPI rozhraním.

Pre bezdrôtovú komunikáciu medzi doskou a vonkajším zariadením som zvolil technológiu Bluetooth Low Energy. Jedná sa o technológiu fungujúcu na základe bluetooth, ktorá je dizajnovaná pre veľmi nízku spotrebu energie so zachovaním pôvodného komunikačného rozsahu Bluetooth, a teda je vhodná na dlhodobé používanie. S pôvodnou technológiou Bluetooth siete nieje kompatibilná, ale takmer všetky moderné zariadenia, ako sú mobilné telefóny natívne podporujú aj BLE. V mojom prípade som na komunikáciu používal svoj mobilný telefón s OS Android a aplikáciu nRF Connect, ktorá slúži na vyhľadávanie, a komunikáciu s BLE zariadeniami.

Prostriedky pre implementáciu:

Dosku ESP 32 som prepojil USB-A/USB-micro káblom s mojím počítačom. V prostredí PlatformIO som založil projekt používajúci framework ESP-IDF. Kód napísaný v tomto prostredí testoval priamo na doske nahraním kódu na dosku a sledovaním sériového výstupu, prípadne pripojeného displeja.

2 Implementácia

Implementácia by sa dala rozdeliť do troch častí: Práca s displejom, Práca s bluetooth, a logika medzi týmito dvoma celkami (teda ako displej reaguje na jednotlivé správy prijaté cez Bluetooth)

2.1 Práca s displejom

Ako už bolo spomenuté, obrovskú časť práce s displejom už za mňa spravila knižnica esp-idf-ssd1306. Pre konfiguráciu jednotlivých pinov dosky som použil Menuconfig vstavaný do prostredia PlatformIO. Po spustení programu sa volajú funkcie inicializácie displeja a čipu ako mastera, ktoré používajú túto konfiguráciu a definovanú veľkosť displeja. Po inicializácii je možné s displejom pracovať. V programe sa používajú najmä

¹<https://github.com/nopnop2002/esp-idf-ssd1306/tree/master>

funkcie pre zobrazenie textu, bitmapového obrázku, a vyčistenie displeja. Funkcie sa starajú iba o vykreslenie pixelov na správne miesto displeja, ale napríklad to, či pixely budú stále v rámci dimenzií displeja, alebo či vykreslený objekt neprekryje iný už neriešia.

2.2 Bluetooth low energy

Aplikácia vytvorí BLE server, na ktorý je možné sa pripojiť a posilať naňho správy, pomocou ktorých sa ovláda TFT displej. Inicializácia a práca s BLE sa skladá z niekoľkých funkcií:

- **ble_app_advertise** - nastaví potrebné parametre a začne "inzerovať" okoliu svoju prítomnosť. To znamená, že sa server stane viditeľným pre ostatné BLE zariadenia.
- **ble_gap_event** - spustí sa zakaždým keď nastane nejaká udalosť. Udalosť je napríklad pripojenie alebo odpojenie BLE zariadenia. Podľa typu udalosti sa potom rozhodne, čo sa má stať.
- **host_task** - funkcia na ktorej program zostáva po celú dobu behu
- Štruktúra **gatt_svcs** - slúži na definíciu služieb, ktoré server ponúka. Služby môžu byť napríklad prijímanie alebo odosielanie dát. Pripojené zariadenia potom vidia dostupné služby a vedia ich používať. V tomto prípade je jedinou službou zápis dát (teda posielanie dát tomuto zariadeniu). Táto služba obsahuje odkaz na funkciu `device_write`, ktorá sa spustí pri prijatí správy. Popis implementácie tejto funkcie je popísaný nižšie.
- Niekoľko vstavaných inicializačných BLE funkcií

2.3 Prepojenie medzi rozhraním BLE a TFT displejom - funkcia `device_write`

Táto funkcia, ako už bolo spomenuté, sa spustí vždy keď pripojené zariadenie pošle správu aplikácii. Obsah tejto správy je potom dostupný v štruktúre `ctx->om->om_data`. V tejto funkcii sa podľa tohto obsahu pomocou spomenutých funkcií pre prácu s displejom zobrazí na displej príslušný text alebo grafika. Na základe prijatej správy sa na displej zobrazí nasledovné:

Pri prijatí správy CLEAR Sa vymaže všetko z displeja.

Pri prijatí správy VUT alebo BATMAN Sa ukáže obrázok. Obrázky sú definované vo forme bitmapy priamo v programe.

Pri prijatí akéhokoľvek iného textu Sa text vypíše na obrazovku. Jednotlivé správy sa vypisujú po riadkoch, pričom keď sa naplní obrazovka, nasledujúca správa všetky riadky vymaže a začne od prvého.

Pri prijatí bitového poľa vo forme 0x01AABB Sa prejde do stavu prijímania obrázku. Podrobnosti sú opísané nižšie.

2.4 Jednoduchý protokol pre prijímanie a zobrazovanie obrázkov

Chcel som aby bola aplikácia schopná prijímať aj obrázky, a následne ich zobrazovať, avšak problémom bolo, že do BLE správy sa dá umiestniť najviac 20 bytov, čo je na väčšinu obrázkov málo. Preto som vytvoril veľmi jednoduchý protokol umožňujúci obrázky posilať.

Prenos obrázku sa začne poslaním aplikácii správy 0x01AABB (3 byte), kde AA značí šírku posielaného obrázku a BB značí výšku (v pixeloch). Napríklad správa 0x01200D značí, že bude posielaný obrázok s šírkou 0x20, teda 32 pixelov, a výškou 0x0D, teda 13 pixelov. Ak by boli dimenzie obrázka väčšie ako dimenzie displeja, program vypíše príslušnú správu a prenos sa nezačne. Ak sú dimenzie v poriadku, prejde sa do stavu prenosu.

V stave prenosu aplikácia prijíma správy dĺžky 20 byte, kde jednotlivé byte sú samotnými bitmapovými dátami

obrázku (bit hodnoty 1 = vyfarbený pixel, bit hodnoty 0 = nevyfarbený pixel). Každá prijatá správa je interpretovaná ako 20 byte výsledného obrázku (teda nie je možné v tomto stave posielat' nič iné ako samotné bitmapové dáta).

Akonáhle príjme všetky dáta obrázku - veľkosť obrázku sa dá jednoducho vypočítať z šírky a výšky zadanej pri inicializácii prenosu - obrázok sa zobrazí a stav prenosu sa tým končí. Aplikácia je späť vo východiskovom stave a na zariadenie je potom možné ďalej posielat' všetky typy správ.

Poznámka - v poslednej prijatej správe sa samozrejme pozerá iba na prvých n byte, pričom $n \leq 20$ je počet byte potrebných pre dokončenie obrázku. Ostatné byty môžu mať akúkoľvek hodnotu

2.5 Demonštračné video

Okaz na video demonštrujúce funkčnosť aplikácie: https://www.youtube.com/watch?v=_HSHZMiTE6k (Vo videu sa nachádza drobná chyba kedy som v textovom popise nesprávne prepočítal hexadecimálne 0x0d na 12)

3 Záver

Výsledná implementácia bola overená s mojím mobilným telefónom zasielaním príslušných správ na vytvorený BLE server a sledovaním TFT displeja. Aj keď overovanie posielania obrázkov bolo z telefónu trochu nepríjemné (tento protokol je jednoznačne určený skôr pre automatizované posielanie dát), pri menších obrázkoch dopadlo dobre. Pre testovanie obrázkov som použil túto stránku, ktorá dokáže obrázky meniť na bitmapové dáta.

Myslím že projekt dopadol dobre, a dokážem si vďaka nemu jasne predstaviť, ako môžu fungovať bezdrôtové periférne zariadenia a dokonca by som vedel ich komunikáciu s použitím BLE naprogramovať, keďže bolo jasne definované, ako sa správy posielajú alebo prijímajú a bol som schopný prijaté dáta spracovať. Myslím že výsledná aplikácia robí to čo má: Demonštruje ovládanie TFT displeja cez bezdrôtové rozhranie Bluetooth. To je ukázané zobrazovaním textu, alebo obrázku, z čoho je, myslím zrejme, že by som pomocou rozhrania Bluetooth vedel ovládať v podstate čokoľvek. Displej zobrazuje prijaté dáta v reálnom čase, a pre zaujímavosť je dokonca pridaná stavová premenná, ktorá slúži pre posielanie väčšieho objemu dát, ktoré sa potom zobrazia v podobe obrázku na displej.

S riešením som nemal väčšie problémy, samozrejme z veľkej časti vďaka skvelým existujúcim knižniciam, najmä pre použitý TFT displej.

Autoevaluácia

- Dokumentácia - 4/4 - Myslím že som splnil, čo sa od dokumentácie očakávalo, a držal som sa predpísaných sekcií a ich obsahu. Aj keď priznávam, že som úplne nedodrжал stanovený rozsah.
- Prístup - 1/1 - Projekt som si nenechal na poslednú chvíľu, venoval som sa mu dlhšiu dobu a spravil som, myslím, viac ako absolútne minimum (pre doslovné splnenie zadania by som mohol napríklad iba vypisovať text prijatý cez bluetooth).
- Funkčnosť - 5/5 - Funkčnosť požadovanú v zadaní som splnil.
- Kvalita - 2/3 - Priznávam, že môj kód možno nie je najkrajší a najčítelnejší, a ťažko povedať, či som problém správne dokoňoval. Čo sa však týka jednoduchosti ovládania a overenia jeho činnosti, za to si myslím bod zaslužím - Ovládanie a komunikáciu si neviem predstaviť jednoduchšie.