



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**BEZDRÁTOVÝ KOMUNIKAČNÍ MODUL PRO VÝ-  
UKOVÝ KIT MINERVA**

WIRELESS COMMUNICATION MODULE FOR MINERVA EDUCATIONAL KIT

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**RICHARD FILO**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÁCLAV ŠIMEK**

BRNO 2022

## Zadání bakalářské práce



Student: **Fil'o Richard**  
Program: Informační technologie  
Název: **Bezdrátový komunikační modul pro výukový kit Minerva**  
**Wireless Communication Module for Educational Kit Minerva**  
Kategorie: Vestavěné systémy

### Zadání:

1. Seznamte se s bezdrátovými komunikačními standardy (např. WiFi, Bluetooth, ZigBee) a komponentami s podporou těchto technologií dostupnými na trhu.
2. Seznamte se s výukovou platformou Minerva, kde se zaměřte především na mikrokontrolér Kinetis K60 a pro něj dostupné vývojové nástroje.
3. Zvolte vhodné komponenty a navrhnete koncepci rozšiřujícího komunikačního modulu, který bude implementovat zvolená bezdrátová rozhraní (minimálně 2 různá) pro platformu Minerva.
4. V návrhovém prostředí dle pokynů vedoucího vytvořte schéma zapojení a motiv desky plošných spojů. Proveďte osazení desky součástkami a následně její oživení.
5. K vytvořenému rozšiřujícímu modulu pro bezdrátovou komunikaci implementujte v jazyce C strukturovanou knihovnu s obslužným firmwarem pro mikrokontrolér Kinetis K60 na platformě Minerva.
6. Vhodným způsobem demonstруйте funkčnost navrženého řešení (např. upravte některý z existujících projektů v repozitáři výukové platformy FITkit).
7. Zhodnoťte dosažené výsledky a navrhnete možnosti dalšího rozšíření.

### Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**  
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.  
Datum zadání: 1. listopadu 2021  
Datum odevzdání: 11. května 2022  
Datum schválení: 29. října 2021

## Abstrakt

V práci sa zaoberáme návrhom rozširujúceho modulu, ktorý poskytne platforme Minerva možnosť bezdrôtovej komunikácie. Výsledkom našej práce je adaptér, ktorý sa pripevní na platformu Minerva a následne sa do neho za pomoci konektora M.2 vloží náš rozširujúci modul. Modul podporuje štandardy Wi-Fi 4.2, BLE 5 a LoRa. Pre vybraný modul sme vytvorili komunikačné rozhranie určené na jednoduchú prácu s daným modulom. V práci ďalej uvádzame vypracované príklady a demo aplikáciu využívajúcu tento modul. Všetky komponenty sú otestované a fungujú podľa našich očakávaní.

## Abstract

In this thesis, we are concerned with the design of an extension module that will provide wireless communication capability to the Minerva platform. The result of our work is an adapter that is attached to the Minerva platform and then our expansion module is inserted into it using an M.2 connector. The module supports Wi-Fi 4.2, BLE 5 and LoRa standards. For the selected module, we have created a communication interface designed to work easily with the module. In the following, we present worked examples and a demo application using this module. All components are tested and work as we expect.

## Klíčové slová

Minerva, FITkit, bezdrôtový modul, Wi-Fi, Bluetooth, LoRa, DPS, jazyk C, mikrokontrolér, MCU

## Keywords

Minerva, FITkit, wireless module, Wi-Fi, Bluetooth, LoRa, PCB, C language, microcontroller, MCU

## Citácia

FIEO, Richard. *Bezdrátový komunikační modul pro výukový kit Minerva*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

# Bezdrátový komunikační modul pro výukový kit Minerva

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Šimka. V práci som uviedol všetky literárne pramene, publikácie a iné zdroje, z ktorých som čerpal.

.....

Richard Fiľo

9. mája 2022

## Podakovanie

Chcel by som poďakovať pánovi inžinierovi Šimkovi za jeho odbornú pomoc a podnetné nápady pri riešení tejto práce.

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>   | <b>2</b>  |
| <b>2</b> | <b>Teoretický základ</b>                                    | <b>3</b>  |
| 2.1      | Vzdelávacia platforma Minerva . . . . .                     | 3         |
| 2.2      | Bezdrôtové štandardy . . . . .                              | 5         |
| <b>3</b> | <b>Návrh a realizácia bezdrôtového rozširujúceho modulu</b> | <b>10</b> |
| 3.1      | Výber vhodných komponentov . . . . .                        | 10        |
| 3.2      | Návrh schémy zapojenia . . . . .                            | 12        |
| 3.3      | Návrh dosky plošného spoja . . . . .                        | 16        |
| 3.4      | Výroba a kompletizácia dosky plošného spoja . . . . .       | 17        |
| <b>4</b> | <b>Tvorba programového vybavenia</b>                        | <b>20</b> |
| 4.1      | SPI komunikácia medzi MCU a modulom . . . . .               | 21        |
| 4.2      | Návrh štruktúrovanej knižnice . . . . .                     | 23        |
| 4.2.1    | Všeobecné definície a funkcie . . . . .                     | 24        |
| 4.2.2    | Definície a funkcie pre prácu s Wi-Fi . . . . .             | 25        |
| 4.2.3    | Transportná vrstva . . . . .                                | 26        |
| 4.2.4    | Príklady použitia . . . . .                                 | 27        |
| 4.3      | Demonštračná aplikácia . . . . .                            | 29        |
| <b>5</b> | <b>Záver</b>  | <b>36</b> |
|          | <b>Literatúra</b>   | <b>37</b> |
| <b>A</b> | <b>Foto-dokumentácia</b>                                    | <b>41</b> |

# Kapitola 1

## Úvod

Vzdelávacia platforma Minerva slúži predovšetkým na zoznámenie študentov s hardwarovými prvkami a tvorbou vstavaných systémov. Pri tvorbe vstavaných systémov sa stretávame s pojmom internet veci (angl. skratka IoT), ktorý je v dnešnej dobe populárny a frekventovane využívaný. Pri tomto riešení jednotlivé vstavané systémy komunikujú medzi sebou pomocou bezdrôtovej komunikácie, či už je to Wi-Fi alebo Bluetooth a preposielajú si potrebné dáta. Vzdelávacia platforma Minerva neposkytuje žiadnu podporu bezdrôtovej komunikácie. Rozhodli sme sa vytvoriť rozširujúci modul, ktorý by podporoval komunikačné štandardy ako napríklad Wi-Fi, Bluetooth alebo LoRa, čo bolo zároveň cieľom našej práce. Podpora týchto štandardov by poskytla študentom možnosť tvorby aplikácií, ktoré by neboli obmedzené výhradne drôtovou komunikáciou. Kroky k dosiahnutiu tohto cieľa sú uvedené nižšie v texte.

Kapitola 2 sa venuje zoznámeniu s platformou Minerva, predovšetkým s mikrokontrolérom Kinetis K60 a jeho dostupnými komunikačnými rozhraniami, pomocou ktorých by mohol mikrokontrolér komunikovať s rozširujúcim bezdrôtovým modulom. Vybraná kapitola sa taktiež venuje bezdrôtovým komunikačným štandardom, ktoré sme si zvolili pre našu prácu a dostupnosti komponentov na trhu, ktoré tieto štandardy podporujú.

V kapitole 3 sme vybrali vhodné komponenty a navrhli koncepciu rozširujúceho komunikačného modulu. Vytvorili sme schému zapojenia jednotlivých komponentov a návrh dosky plošného spoja. Následne sme dosky mohli vyrobiť a osadiť.

Doska bez logiky by pre nás znamenala len kus hardwaru, preto bolo nutné vytvoriť štruktúrovanú knižnicu s obslužným firmwarom. V kapitole 4 sa zaoberáme návrhom tejto knižnice a implementácii potrebného softwaru. Ďalej, sme v nej vytvorili príklady použitia, kde je znázornené ako by sa s modulom malo správne pracovať. V neposlednom rade sme zhotovili demo aplikáciu, ktorá demonštruje funkčnosť rozširujúceho modulu.

V záverečnej kapitole 5 sa venujeme zhrnutiu dosiahnutých výsledkov a tiež možnostiam ďalšieho rozšírenia.

## Kapitola 2

# Teoretický základ

Pred tvorbou bezdrôtového rozširujúceho modulu sa potrebujeme zoznámiť so vzdelávacou platformou Minerva, predovšetkým s mikrokontrolérom Kinetis K60, ktorý bude komunikovať s vytvoreným modulom. V dnešnej dobe disponujeme s rôznou škálou bezdrôtových komunikačných štandardov. V tejto časti práce sa zoznámime s troma z nich. Špecifikujeme dôvod, prečo sme sa zamerali práve na dané štandardy a zoznámime sa s komponentmi dostupnými na trhu podporujúcimi tieto štandardy.

### 2.1 Vzdelávacia platforma Minerva

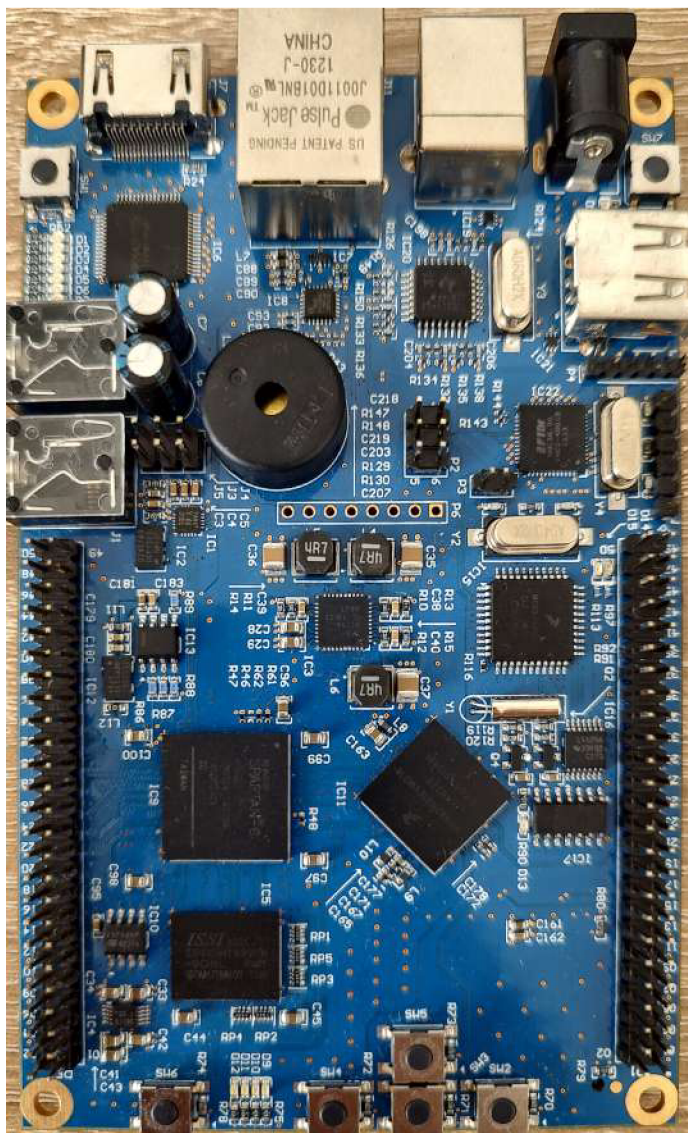
Platforma je známa tiež ako FITkit3 a zobrazujeme ju na obrázku 2.1. Poskytuje nám množstvo periférií a integrovaných obvodov, ktoré môžu študentovi pomôcť pri získavaní vedomostí a zručností s tvorbou vstavaných systémov. Na vzdelávacej platforme Minerva[29] sa nachádzajú dva čipy, ktoré sú primárne určené na nahrávanie projektov. Jedným z nich je programovateľné hradlové pole Xilinx Spartan-6 na získavanie znalostí s FPGA čipom a jazykom VHDL, ktorým môžeme popísať štruktúru zapojenia hradiel. Druhým čipom je mikrokontrolér Kinetis MK60 určený na programovanie univerzálnych vstavaných systémov v jazyku C/C++. V našej práci sa predovšetkým zameriavame na tento mikrokontrolér. Pomocou neho vieme komunikovať s vytvoreným rozširujúcim modulom a tvoriť aplikácie s podporou bezdrôtovej komunikácie.

#### Kinetis K60

Mikrokontrolér Kinetis K60 je založený na jadre ARM Cortex-M4 a ponúka nám zástup periférií aby sa hlavný procesor nemusel zaoberať frekventovanými úlohami. K týmto perifériám môžeme uviesť komunikačné rozhrania, ako sú: UART, I2C, I2S, SPI, SDHC a GPIO pre prácu so štandardným vstupom a výstupom. K systémovým perifériám jednotka ochrany pamäte alebo Watchdog. Podrobnejší popis mikrokontroléru je uvedený v referenčnom manuály[8].

Dostupné periférie potom môžeme riadiť nastavením hodnôt do príslušných registrov. Vývoj akejkoľvek aplikácie by bol časovo náročný z dôvodu neustáleho sledovania dokumentácie a hodnoty ktorú je potrebné zapísať do registra. Vývojári ku kontrolérom dodávajú software zvaný Software development kit (SDK), ten prácu s daným mikrokontrolérom zjednoduší a zrýchli. SDK obsahuje ovládače pre prácu s perifériami a taktiež definície rôznych konštánt aby sme znížili potrebu nahliadať do referenčného manuálu. Práve z týchto dôvodov sme sa rozhodli využiť pre vývoj štruktúrovanej knižnice a demo aplikácií vývo-

iový nástroj dodávaný priamo spoločnosťou NXP MCUXpresso IDE[25] a k nemu SDK[26] určený pre náš mikrokontrolér.



Obr. 2.1: Vzdelávacia platforma Minerva

## Dostupné rozhrania

Pre výber komponentov podporujúcich bezdrôtové štandardy potrebujeme zistiť aké komunikačné rozhrania máme dostupné na danej vývojovej doske. Zo schémy zapojenia[29] sme zistili, že výstupy mikrokontroléra sú napojené na konektor s označením P1. To je 2x25 vývodový konektor umiestnený na pravej dolnej strane dosky, vid. obrázok 2.1. Na tomto konektore sú uvedené komunikačné rozhrania SPI a I2C, ktoré by sa dali použiť pre komunikáciu s vybranými komponentmi. Ďalej máme dostupných zopár vstupno-výstupných vývodov. Tie by mohli byť využité napríklad na zapnutie a vypnutie komponentov rozširujúceho modulu alebo identifikáciu toho, v akom stave sa vybraný komponent nachádza.



## 2.2 Bezdrôtové štandardy

V dnešnej dobe je dostupné množstvo bezdrôtových komunikačných štandardov. Pre rozširujúci modul sme vybrali 3 z nich, ktoré sú podľa nášho názoru najznámejšie a najpoužívanejšie.

### Wi-Fi

Zariadenia používajúce Wi-Fi môžeme aktuálne nájsť v každej domácnosti. Pomocou tejto technológie sa implementuje bezdrôtový prístup k internetu. To znamená, že smerovač ktorý má prístup k internetovej sieti je nastavený v role prístupového bodu a ostatné zariadenia ako napríklad mobil, notebook alebo inteligentný televízor sú v role stanice, ktorá sa k tomuto bodu pripája.

Práve z tohto dôvodu podpora danej technológie na našom rozširujúcom module nesmie chýbať.

Wi-Fi technicky nie je označenie pre štandard, ale pre zariadenie ktoré je schopné komunikovať s ostatnými Wi-Fi zariadeniami. Toto označenie je po príslušnom testovaní pridelené združením spoločnosti Wi-Fi Alliance[1].

Zariadenia Wi-Fi využívajú štandardy, ktoré sú členmi rodiny štandardov IEEE 802.11. Je podstatné dávať si pozor, pretože nie všetky štandardy 802.11 sú testované Wi-Fi Alliance a Wi-Fi certifikované.

Z nášho pohľadu je potreba vedieť, že jednotlivé štandardy sa líšia predovšetkým používanou frekvenciou a maximálnou rýchlosťou prenosu. Jednotlivé hodnoty môžeme vidieť v tabuľke 2.1.

Tabuľka 2.1: Štandardy IEEE 802.11

| Štandard      | Označenie | Rok vydania | Pásmo[GHz] | Rýchlosť[Mbit/s] |
|---------------|-----------|-------------|------------|------------------|
| IEEE 802.11a  | Wi-Fi 1   | 1999        | 5          | 54               |
| IEEE 802.11b  | Wi-Fi 2   | 1999        | 2.4        | 11               |
| IEEE 802.11g  | Wi-Fi 3   | 2003        | 2.4        | 54               |
| IEEE 802.11n  | Wi-Fi 4   | 2009        | 2.4/5      | 600              |
| IEEE 802.11ac | Wi-Fi 5   | 2013        | 5          | 3466.8           |
| IEEE 802.11ax | Wi-Fi 6   | 2019        | 2.4/5/6    | 10530            |

Výrazná väčšina Wi-Fi zariadení v dnešnej dobe implementuje štandardy IEEE 802.11 b,g,n. Pokúsili sme sa preto nájsť komponenty kompatibilné s týmito štandardmi.

**Prieskum trhu** je pre výber vhodných komponentov poskytujúcej Wi-Fi významný. Pri nájdených zariadeniach sme sledovali aké štandardy sú podporované, rozhranie za pomoci ktorého budeme komunikovať s mikrokontrolérom Kinetis K60, rozmery, cenu a prítomnosť integrovanej antény. Pre študijné účely nám bude stačiť integrovaná anténa. Enormné množstvo takýchto komponentov nám ponúka taktiež podporu Bluetooth. Danej problematike sa podrobnejšie budeme venovať v ďalšej časti práce. Výsledkom nášho prieskumu je tabuľka 2.2, ktorú sme použili pri výbere vhodných komponentov v časti 3.1.

Tabulka 2.2: Wi-Fi moduly

| Označenie         | Výrobca   | Wi-Fi    | Rozhranie            | Ant. | Rozmer [mm] | Cena [\$/kus] | Ref. |
|-------------------|-----------|----------|----------------------|------|-------------|---------------|------|
| 88W8801           | NXP       | 802.11n  | SDIO, USB            | Nie  | 6x6         | 2.72          | [15] |
| 88W8977           | NXP       | 802.11n  | SDIO                 | Nie  | 8x8         | 5.2           | [18] |
| IW416             | NXP       | 802.11n  | SDIO                 | Nie  | 8x8         | 3.8           | [22] |
| 88W8887           | NXP       | 802.11ac | SDIO                 | Nie  | N/A         | 6.65          | [16] |
| 88W8897           | NXP       | 802.11ac | PCIe                 | Nie  | N/A         | 16.04         | [17] |
| 88W8987           | NXP       | 802.11ac | SDIO                 | Nie  | 8x8         | 9.48          | [19] |
| 88W8997           | NXP       | 802.11ac | PCIe, SDIO, USB      | Nie  | N/A         | 7.39          | [20] |
| IW620             | NXP       | 802.11ax | PCIe                 | Nie  | N/A         | N/A           | [24] |
| 88W9098           | NXP       | 802.11ax | PCIe                 | Nie  | N/A         | 15.2          | [21] |
| IW612             | NXP       | 802.11ax | SDIO                 | Nie  | N/A         | N/A           | [23] |
| MAYA-W1           | u-blox    | 802.11n  | SDIO                 | Nie  | 10.4x14.4   | N/A           | [40] |
| JODY-W3           | u-blox    | 802.11ax | PCIe, SDIO           | Nie  | 13.8x19.8   | N/A           | [39] |
| JODY-W2           | u-blox    | 802.11ac | SDIO                 | Nie  | 13.8x19.8   | 22.23         | [38] |
| NINA-W15          | u-blox    | 802.11n  | SPI, UART            | Áno  | 10x14       | 13.58         | [42] |
| NORA-W2           | u-blox    | 802.11n  | UART                 | Áno  | 10.4x14.3   | N/A           | [43] |
| NINA-W10          | u-blox    | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 10x14       | 7.55          | [41] |
| Sterling-LWB+     | Laird     | 802.11n  | SDIO                 | Nie  | 12x12       | 19.72         | [7]  |
| Sterling 70       | Laird     | 802.11ax | PCIe, SDIO           | Nie  | 18x20       | N/A           | [4]  |
| Sterling-LWB5+    | Laird     | 802.11ac | SDIO, USB            | Áno  | 12x17       | 15.59         | [6]  |
| Sterling-EWB      | Laird     | 802.11n  | SPI, QSPI, UART, PCM | Áno  | 16x21       | 21.75         | [5]  |
| ESP32-C3-MINI-1   | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 16.6x13.2   | 1.88          | [31] |
| ESP32-C3-WROOM-02 | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 20x18       | 1.95          | [32] |
| ESP32-S3-MINI-1   | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 20.5x15.4   | 3.08          | [34] |
| ESP32-S3-WROOM-1  | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 25.5x18     | 2.85          | [36] |
| ESP32-MINI-1      | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 19x13.2     | 2.55          | [33] |
| ESP32-WROOM-32E   | Espressif | 802.11n  | SPI, I2C, I2S, UART  | Áno  | 25.5x18     | 3.93          | [37] |

## Bluetooth

Ďalším známym a zároveň najpoužívanejším štandardom je Bluetooth[3]. Môžeme ho nájsť v širokej škále zariadení ako sú napríklad: mobil, slúchadlá, či inteligentné hodinky. Bluetooth sa predovšetkým využíval na prenos zvuku v reproduktoroch alebo slúchadlách. Od verzie 4 sa Bluetooth štandard rozdelil prakticky na dva rozdielne štandardy. Jedným z nich je Bluetooth Classic a druhým Bluetooth Low Energy (BLE). V poslednej dobe sa stretávame často s pojmom internet veci (IoT), v týchto riešeniach sa nevyužíva veľký prenos dát ako to bolo pri preposielaní vysoko kvalitnej hudby alebo videa. Bol vytvorený nový protokol BLE, ktorý má na rozdiel od klasického Bluetooth omnoho nižšiu spotrebu ale na úkor toho nedosahuje tak vysoké prenosové rýchlosti. Aj napriek tomu má BLE vysokú škálu využitia a dnes nemôže chýbať v mobilných zariadeniach na komunikáciu s inteligentnými hodinami, či náramkom. To sa stalo hlavným dôvodom prečo sme sa rozhodli daný štandard implementovať aj v našom rozširujúcom module.

Ako iste viete, Bluetooth funguje na rovnakej frekvencii ako Wi-Fi a to je 2.4 GHz. Dosiaľ, čo BLE dosahuje maximálnu prenosovú rýchlosť až 2 Mb/s klasický Bluetooth dosahuje rýchlosť 3 Mb/s. Využitie BLE sa v dnešnom svete rozširuje čoraz viac, prináša nám aj väčšiu škálu možností jeho využitia. Na rozdiel od klasického point to point spojenia, ktoré nám ponúka aj klasický Bluetooth, predstavuje broadcast vysielanie či takzvané Bluetooth mesh komunikáciu viacerých zariadení naraz. BLE využíva rovnako vlastnú množinu protokolov určenú na rozpoznávanie vysielania alebo rozposielanie dát. GAP z anglického Generic Access Profile je jedným z nich. Je určený predovšetkým na naviazanie komunikácie medzi zariadeniami. Ďalším veľmi dôležitým protokolom je GATT z anglického Generic Attribute Profile. Ten sa stará o uschovávanie a spôsob prenosu dát medzi zariadeniami.

Keďže BLE je podporované od verzie Bluetooth 4 pokúšali sme sa nájsť také zariadenie, ktoré by spĺňalo túto špecifikáciu. Rovnako tak, bolo by plusom nájsť zariadenie podporujúce Bluetooth 5, v závislosti od toho, že vo verzii máme možnosť pri BLE zvýšiť maximálny dosah zariadení z pôvodných päťdesiat metrov až na osemsto metrov.

**Prieskum trhu** je pre výber vhodných komponentov poskytujúcich BLE rovnako veľmi dôležitý. Pri nájdených zariadeniach sme sledovali aká verzia Bluetooth je podporovaná a všetky ostatné parametre, ktoré sme sledovali pri výbere Wi-Fi komponentov. Výsledkom nášho prieskumu je tabuľka 2.3, ktorú taktiež použijeme pri výbere vhodných komponentov v časti práce 3.1.

Tabuľka 2.3: Bluetooth moduly

| Označenie         | Výrobca   | Bluetooth | Rozhranie           | Ant. | Rozmer [mm] | Cena [\$/kus] | Ref. |
|-------------------|-----------|-----------|---------------------|------|-------------|---------------|------|
| 88W8977           | NXP       | 5.2       | UART                | Nie  | 8x8         | 5.2           | [18] |
| IW416             | NXP       | 5.2       | UART                | Nie  | 8x8         | 3.8           | [22] |
| 88W8887           | NXP       | 5.2       | UART                | Nie  | N/A         | 6.65          | [16] |
| 88W8897           | NXP       | 5.0       | UART                | Nie  | N/A         | 16.04         | [17] |
| 88W8987           | NXP       | 5.2       | UART, SDIO          | Nie  | 8x8         | 9.48          | [19] |
| 88W8997           | NXP       | 5.3       | UART                | Nie  | N/A         | 7.39          | [20] |
| IW620             | NXP       | 5.1       | UART                | Nie  | N/A         | N/A           | [24] |
| 88W9098           | NXP       | 5.3       | UART                | Nie  | N/A         | 15.2          | [21] |
| IW612             | NXP       | 5.2       | UART                | Nie  | N/A         | N/A           | [23] |
| MAYA-W1           | u-blox    | 5.0       | UART                | Nie  | 10.4x14.4   | N/A           | [40] |
| JODY-W3           | u-blox    | 5.3       | UART                | Nie  | 13.8x19.8   | N/A           | [39] |
| JODY-W2           | u-blox    | 5.2       | UART, SDIO          | Nie  | 13.8x19.8   | 22.23         | [38] |
| NINA-W15          | u-blox    | 4.2       | SPI, UART           | Áno  | 10x14       | 13.58         | [42] |
| NORA-W2           | u-blox    | 5.0       | UART                | Áno  | 10.4x14.3   | N/A           | [43] |
| NINA-W10          | u-blox    | 4.2       | SPI, I2C, I2S, UART | Áno  | 10x14       | 7.55          | [41] |
| Sterling-LWB+     | Laird     | 5.2       | HCI HS-UART, PCM    | Nie  | 12x12       | 19.72         | [7]  |
| Sterling 70       | Laird     | 5.1       | UART                | Nie  | 18x20       | N/A           | [4]  |
| Sterling-LWB5+    | Laird     | 5.0       | SDIO, USB           | Áno  | 12x17       | 15.59         | [6]  |
| Sterling-EWB      | Laird     | 5.1       | HCI HS-UART         | Áno  | 16x21       | 21.75         | [5]  |
| ESP32-C3-MINI-1   | Espressif | 5.0       | SPI, I2C, I2S, UART | Áno  | 16.6x13.2   | 1.88          | [31] |
| ESP32-C3-WROOM-02 | Espressif | 5.0       | SPI, I2C, I2S, UART | Áno  | 20x18       | 1.95          | [32] |
| ESP32-S3-MINI-1   | Espressif | 5.0       | SPI, I2C, I2S, UART | Áno  | 20.5x15.4   | 3.08          | [34] |
| ESP32-S3-WROOM-1  | Espressif | 5.0       | SPI, I2C, I2S, UART | Áno  | 25.5x18     | 2.85          | [36] |
| ESP32-MINI-1      | Espressif | 4.2       | SPI, I2C, I2S, UART | Áno  | 19x13.2     | 2.55          | [33] |
| ESP32-WROOM-32E   | Espressif | 4.2       | SPI, I2C, I2S, UART | Áno  | 25.5x18     | 3.93          | [37] |

## LoRa

Wi-Fi a Bluetooth majú štandardne dosah desiatky metrov. Táto vzdialenosť je pre našu domácu sieť dostačujúca ale čo sa stane, ak chceme komunikovať so zariadením, ktoré je od nás vzdialené stovky metrov? Práve z tohto dôvodu sme sa rozhodli k nášmu rozširujúcemu modulu pridať taký komponent, ktorý bude implementovať štandard LoRa.

Skratka LoRa pochádza z anglického Long Range v preklade to znamená dlhý dosah. LoRa je štandard, ktorý využíva nízke frekvencie a tým pádom dosahuje omnoho vyšší dosah prenosu dát. Pretože pracujeme na nižších frekvenciách nemôžeme očakávať vysokú rýchlosť prenosu. Každá krajina má zadefinované frekvenčné pásmo, ktoré môže využívať. V našom prípade je to 868 MHz. LoRa je technológia pracujúca na fyzickej vrstve. To znamená, že sa zameriava len na odosielanie a prijímanie dát. O prepojenie a komunikáciu medzi takýmito zariadeniami sa stará LoRaWAN. Podľa tejto stránky[28] štandard LoRa dosahuje enormných pätnásť kilometrov. Škála využitia tejto technológie je určite vysoká. Napríklad, máme záhradu na ktorej potrebujeme odmerať vlhkosť a poslať ju na server ale v blízkosti záhrady nemáme dosah pre použitie Wi-Fi siete. V takomto prípade môžeme jednoducho využiť vybraný štandard.

**Prieskum trhu** pri tomto štandarde bude jednoduchší z dôvodu, že na trhu nemáme veľa výrobcov týchto modulov. Najdôležitejším faktorom na ktorý je potrebné zamerať našu pozornosť je prenosové pásmo. Ďalšie vlastnosti na ktoré sa môžeme pozeráť je cena a rozmery modulu tak, aby ho bolo možné pripevniť na výsledný rozširujúci modul. Výsledkom prieskumu je tabuľka 2.4. Použijeme ju pri výbere vhodných komponentov v časti 3.1.

Tabuľka 2.4: LoRa moduly

| Označenie | Výrobca | Pásmo [MHz] | Rozhranie | Rozmer [mm] | Cena [\$/kus] | Ref. |
|-----------|---------|-------------|-----------|-------------|---------------|------|
| RFM95W    | HOPERF  | 868/915     | SPI       | 16x16       | 20.6          | [10] |
| RFM95PW   | HOPERF  | 868/915     | SPI       | 35.4x18     | 14.38         | [9]  |
| RFM96W    | HOPERF  | 433/470     | SPI       | 16x16       | 14.95         | [11] |
| RFM98W    | HOPERF  | 169         | SPI       | 16x16       | 8.78          | [13] |
| RFM98PW   | HOPERF  | 169/433/470 | SPI       | 35.4x18     | 13.84         | [12] |

## Kapitola 3

# Návrh a realizácia bezdrôtového rozširujúceho modulu

V tejto kapitole sa zameriavame predovšetkým na hardwarovú časť práce. Od výberu vhodných komponentov až po návrh ich zapojenia a osadenia dosky plošného spoja.

### 3.1 Výber vhodných komponentov

Bez výberu vhodných komponentov by náš rozširujúci modul nedokázal správne komunikovať s mikrokontrolérom Kinetis K60. V práci spomíname, že väčšia časť skúmaných komponentov sú moduly poskytujúce Bluetooth aj Wi-Fi. Tento variant bol vybraný z viacerých dôvodov. Jeden z nich bol, ušetrenie miesta na výslednej doske plošného spoja, pretože následne nám dostačoval iba jeden modul poskytujúci oba tieto štandardy. Ďalším z dôvodov, bolo zredukovanie počtu využitých vývodov, ktorými bolo nutné vybranú dosku prepojiť. Neuvádzame vývody poskytujúce sériovú komunikáciu, pretože týmito vývodmi môžeme prepojiť väčšie množstvo zariadení. Hovoríme skôr o vstupno-výstupných vývodoch, ktoré využijeme na zapnutie a vypnutie jednotlivých komponentov alebo získavanie informácií o ich stave. Poslednou výhodou využitia tohto variantu môže byť nižšia cena.

Následne začneme s výberom komponentu poskytujúceho Bluetooth a Wi-Fi. Ak by sme sa najprv nazerali výhradne na verzie štandardov, z určitou vieme povedať, že by nás zaujímali moduly 88W9098, 88W8997 alebo JODY-W3. Všetky tieto moduly nám poskytujú podporu Wi-Fi 6 a Bluetooth 5.3. Hlavným dôvodom prečo sme dané moduly nemohli použiť je to, že v čase nášho prieskumu bola väčšina týchto modulov v procese vývoja, čiže neboli dostupné. V časti 2.1 sme zistili, že musíme hľadať komponenty poskytujúce rozhrania SPI alebo I2C. Uvedené rozhrania nám ponúkajú všetky komponenty z rodiny ESP32, čiže aj moduly NINA ktorých srdcom je rovnaký komponent. Všimli sme si, že tieto komponenty nám poskytujú Wi-Fi 4. Vyššie sme zdôraznili, že táto verzia je pre nás dostačujúca. Bluetoothom vo verzii 5 disponujú vybrané komponenty ESP32-C3 a ESP32-S3. ESP32-C3 je odľahčená verzia, určená pre nie tak zložité úlohy. Na základe toho sme si zvolili pre komponent poskytujúci štandardy Wi-Fi a Bluetooth zostávajúci ESP32-S3, konkrétne verziu WROOM-1, ktorá sa nám jednoduchšie osadzovala na dosku plošného spoja.

Pri výbere komponenty poskytujúcej LoRaWAN sme si museli dať pozor na jednu z dôležitých vecí. Je ňou prenosové pásmo. Všetky skúmané moduly používajú rozhranie SPI, pomocou ktorého môžeme tento komponent prepojiť s mikrokontrolérom Kinetis K60. V našej krajine využívame prenosové pásmo 868 MHz. To znamená, že na výber nám zostali

moduly RFM95W a RFM95PW. Následne sme sa zamerali na to, aby bol rozširujúci modul čo najmenší. Vybrali sme modul RFM95W, ktorý spĺňal naše kritéria aj keď bola jeho suma približne o päť dolárov vyššia ako u modulu RFM95PW.

### ESP32-S3

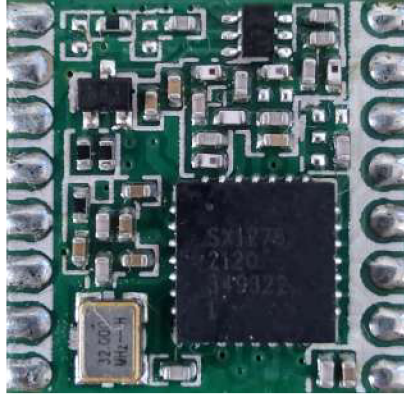
Srdcom tohto mikrokontroléra je dvojjadrový, tridsaťdva bitový mikroprocesor Xtensa LX7 s obnovovacou frekvenciou až do 240 MHz. Jednou z vlastností a zároveň hlavným z dôvodov prečo sme si tento komponent vybrali je podpora štandardu 802.11 b,g,n kde 802.11n nám ponúka prenosovú rýchlosť až do 150 Mbit/s a podpora Bluetooth 5. Pri podpore Bluetooth 5 je potrebné si dať pozor, keďže táto verzia poskytuje len BLE a nie klasický Bluetooth, ktorý je podporovaný napríklad v staršej verzii ESP32. Nám tento fakt neprekáča z hľadiska toho, že pri tvorbe aplikácií IoT, nejde vôbec o veľké prenosy dát ako sú napríklad rozposielanie vysoko kvalitnej hudby alebo videa. Na obrázku 3.1 môžeme vidieť práve tento mikrokontrolér, kde sme vybrali variant s integrovanou anténou. Tá je pre nás na študijné účely dostačujúca. K perifériám môžeme spomenúť GPIO, SPI, UART, I2C, I2S, časovače a Watchdog. Naša varianta rovnako tak poskytuje SPI pamäť s veľkosťou 8 MB.



Obr. 3.1: Mikrokontrolér ESP32-S3-WROOM-1

### RFM95W

Vysielač RFM95W zobrazený na obrázku 3.2, je vybavený LoRa modemom. Ten poskytuje dlhý dosah komunikácie v rozvinutom spektre, nízku úroveň rušenia a tiež minimalizuje spotrebu prúdu. Podporuje programovateľnú prenosovú rýchlosť až do 300 Kb/s. Ďalej, poskytuje pakety s veľkosťou až do 256 bajtov. Nevýhodou, je neprítomnosť integrovanej antény. Tento problém sme vyriešili dokúpením antény, ktorá dokáže vysielať na frekvencii 868 MHz. Komunikáciu s týmto modulom zabezpečuje SPI rozhranie. Za pomoci SPI rozhrania sa bežne nastavujú jednotlivé hodnoty registrov a tým sa môže riadiť správanie modulu.



Obr. 3.2: Modul RFM95W

## 3.2 Návrh schémy zapojenia

V tejto podkapitole venujeme pozornosť návrhu schémy zapojenia. Ešte predtým než sme určili aké vývody budeme potrebovať, museli sme si definovať k čomu budú slúžiť. Vybrali sme komponenty, ktoré sú napájané napätím 3.3V. Potrebovali sme jeden vývod, za pomoci ktorého sme rozviedli napájanie(3V3) a jeden na uzemnenie(GND). Oba komponenty využívajú SPI komunikáciu. Potrebovali sme preto tri vývody pre signály MISO, MOSI, CLK a ďalšie dva pre výber zariadenia (CS LORA, CS ESP). Následne sme potrebovali vývody pomocou ktorých, je možné zariadenie resetovať (RESET LORA, RESET ESP) a vývody prostredníctvom ktorých zisťujeme v akom stave sa dané zariadenie nachádza (IRQ LORA, IRQ ESP).

Na základe zistenia potreby spôsobu, ako bude rozširujúci modul pripojený na vzdelávaciu platformu Minerva, potrebujeme využiť vývody ktoré sú dostupné z konektora P1. Ak by sme navrhli schému tak, že rozširujúci modul je priamo napojený na vzdelávaciu platformu, modul by sa dal využiť len s touto platformou. Na univerzite disponujeme ďalšími dvoma verziami vzdelávacieho Kitu: FITkit1 a FITkit2. Rozhodli sme sa náš modul spraviť rozšírenejším a to tak, že sme pri návrhu schémy použili konektor M.2. Náš projekt sa skladá z dvoch častí. Prvou z nich je adaptér, ten sa nasadí na vzdelávaciu platformu a bude poskytovať M.2 rozhranie. Druhou časťou je modul, ktorý sa za pomoci M.2 konektora vloží do príslušného adaptéra. Týmto spôsobom, sme docielili rozšíriteľnosť nášho modulu na viacero zariadení. Závislosť modulu na rozložení vývodov danej platformy je bezpredmetná.

### Konektor M.2

Konektor M.2[27] má rôzne druhy klúčov, ktoré sa líšia definíciou rozhraní, ktoré daný klúč poskytuje. Zistili sme, že pre naše riešenie je vhodné využiť M.2 konektor s E klúčom. Ten poskytuje rozhranie USB 2.0, 2X PCIe, I2C, SDIO, UART, PCM. Pre účely našej práce sme využili I2C rozhranie, ktoré je možné využiť aj v budúcnosti pri iných projektoch a SDIO rozhranie v móde SPI. Výsledne prepojenie a výstupy jednotlivých vývodov môžeme viesť v tabuľke 3.1. Je potrebné držať sa tejto tabuľky a to v prípade, pokiaľ chcete vytvoriť adaptér pre akúkoľvek platformu kompatibilnú s našim rozširujúcim modulom.



Tabuľka 3.1: Definícia vývodov pre M.2 E kľúč

| Číslo M.2 vývodu                         | Označenie M.2 vývodu | Označenie vývodu modulu |
|--|----------------------|-------------------------|
| 1, 7, 18, 33, 39, 45, 51, 57, 63, 69, 75 | GND                  | GND                     |
| 2, 4, 72, 74                             | 3.3V                 | 3V3                     |
| 9  | SDIO CLK(CLK)        | CLK                     |
| 11                                       | SDIO CMD(MOSI)       | MOSI                    |
| 13                                       | SDIO DATA0(MISO)     | MISO                    |
| 15                                       | SDIO DATA1(IRQ)      | IRQ ESP                 |
| 17                                       | SDIO DATA2           | IRQ LORA                |
| 19                                       | SDIO DATA3(CS)       | CS ESP                  |
| 21                                       | SDIO WAKE            | CS LORA                 |
| 23                                       | SDIO RESET           | RESET ESP               |
| 58                                       | I2C DATA             | SDA                     |
| 60                                       | I2C CLK              | SCL                     |
| 62                                       | I2C ALERT            | RESET LORA              |

## Adaptér

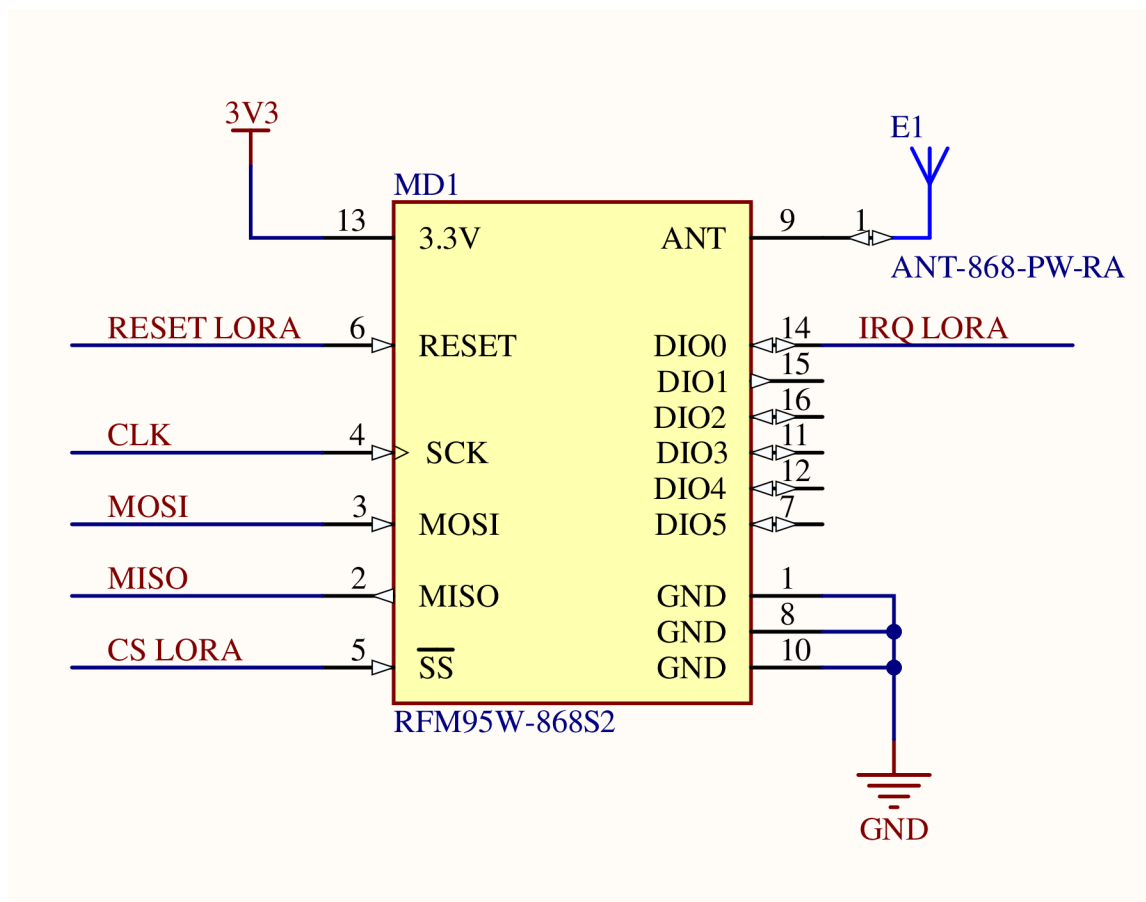
Pri návrhu schémy zapojenia časti adaptéra sa riadime podľa tabuľky 3.1. Pre MISO, MOSI, CLK a CS ESP použijeme vývody z rozhrania SPI2. Rovnako tak SDA a SCL vývody vyvedieme z rozhrania I2C0. Na základe toho vzniklo tretie zariadenie pripojené na toto rozhranie, ošetrili sme tieto vývody pull up rezistormi. Na všetky ostatné signály sme využili vstupno – výstupné vývody portu A. Prepojenie jednotlivých vývodov konektoru P1 na platforme Minerva a vývodov rozširujúceho modulu uvádzame v tabuľke 3.2. Výsledná schéma zapojenia sa nachádza v prílohe na obrázku A.1.

Tabuľka 3.2: Prepojenie vývodov adaptéra

| Číslo vývodu konektora P1 | Označenie MCU vývodu | Označenie vývodu modulu |
|---------------------------|----------------------|-------------------------|
| 1,3                       | V3.3                 | 3V3                     |
| 17                        | MCU I2C0 SCL         | SCL                     |
| 18                        | MCU I2C0 SDA         | SDA                     |
| 19                        | MCU SPI2 CLK         | CLK                     |
| 20                        | MCU SPI2 MOSI        | MOSI                    |
| 21                        | MCU SPI2 CS1         | CS ESP                  |
| 22                        | MCU SPI2 MISO        | MISO                    |
| 23                        | PTA8                 | IRQ ESP                 |
| 24                        | PTA10                | RESET ESP               |
| 25                        | PTA6                 | IRQ LORA                |
| 27                        | PTA7                 | CS LORA                 |
| 28                        | PTA9                 | RESET LORA              |
| 49,50                     | GND                  | GND                     |

## Rozširujúci modul

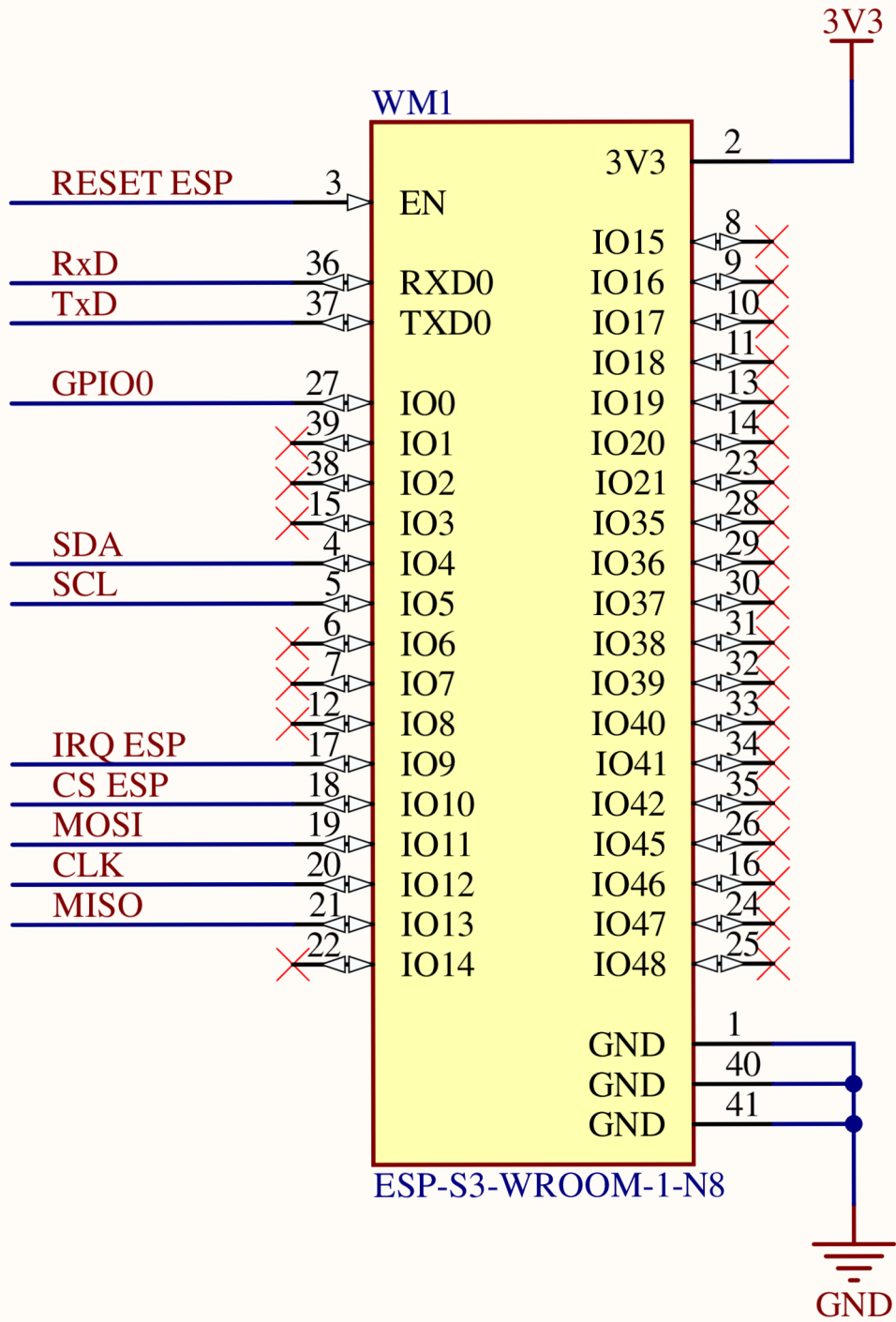
Návrh schémy zapojenia rozširujúceho modulu bol priamočiary. Zapojenie modulu RFM95W sme vytvorili podľa referenčného manuálu, kde sme venovali zvýšenú pozornosť tomu, aby všetky vývody boli pripojené na správne miesto. Výsledne zapojenie signálov uvádzame na obrázku 3.3. Celkové zapojenie je zobrazené v prílohe na obrázku A.4.



Obr. 3.3: RFM95W – Zapojenie signálov

Zapojenie mikrokontroléra ESP32-S3 bolo o niečo zložitejšie z dôvodu, že mikrokontrolér neobsahuje žiaden firmware. Bolo potrebné vyvieť vývody od rozhrania UART, aby sme mikrokontrolér mohli správne naprogramovať. Daný mikrokontrolér nie je určený pre časté programovanie, keďže nami vytvorený firmware nahrajeme len raz. Rozhodli sme sa do nášho zapojenia neimplementovať prevodník z USB na UART. Pri programovaní tohto mikrokontroléra použijeme externý prevodník, ktorý sme zakúpili. K automatickému programovaniu bez nutnosti stláčať tlačidlo reset a boot sme vyviedli na výstupný konektor ešte dva signály a to RST a DTR. Pri zapojení sme sa inšpirovali štandardným zapojením ESP32-S3 vývojovej dosky, ktoré môžeme nájsť tu[30]. Použili sme zapojenie dvoch tlačidiel určených pre reset a boot, kondenzátorov pripojených k napájaniu a k resetovaciemu vstupu, spoločne s pull up rezistorom a NPN tranzistorov pre dosiahnutie správnych hodnôt na resetovacom a bootovacom vývode pri automatickom programovaní. Na záver sme do nášho návrhu pridali červenú identifikačnú LED diódu, ktorá identifikuje, či je k rozširujúceho modulu privedené napájanie. Všetky ostatné signály sme napojili k príslušným

vývodom podľa referenčného manuálu. Zapojenie týchto signálov znázorňujeme na obrázku 3.4. Výslednú schému zapojenia uvádzame v prílohe na obrázku A.3.



Obr. 3.4: ESP32-S3 – Zapojenie signálov

### 3.3 Návrh dosky plošného spoja

Na návrh dosky plošného spoja sme si vybrali nástroj Altium Designer[2]. Keďže naša univerzita poskytuje licenciu k tomuto nástroju, bola by škola nevyužit ju. Altium Designer nám ponúka množstvo funkcií. Od návrhu štruktúrovanej schémy, výberu súčiastok z certifikovaných obchodov, až po návrh dosky plošného spoja. K tomu, vygenerovanie potrebných výstupných súborov. Zaujímavá vlastnosť je, že si pre navrhnutú dosku vieme vygenerovať 3D model a tým zistiť, ako by mohla vyzeráť výsledná doska plošného spoja po osadení. Predtým, ako sme začali návrh dosky plošného spoja, bolo potrebné naučiť sa ako daný nástroj správne používať. Vybranú skúsenosť sme si osvojili pomocou návodov na platforme YouTube.

Pri tvorbe plošného spoja sme v úvode potrebovali vybrať správne súčiastky, ktoré budeme používať. Väčšina z nich bola dostupná prostredníctvom internetu a nebolo k nim potrebné vytvárať schematickú značku alebo odtlačok. Pre súčiastky ako konektor M.2 alebo mikrokontrolér ESP32-S3 bolo potrebné tieto úkony zhotoviť ručne. Rozmery potrebné pre odtlačok plošného spoja sme zistili z referenčného manuálu daných súčiastok.

Potom, čo sme mali súčiastky zhromaždené a pripravené, mohli sme ich prepojiť podľa nami navrhnutých schém zapojenia.

#### Adaptér

Pri návrhu tejto časti, bolo potrebné brať do úvahy faktor, ktorým je vzdialenosť medzi 2x25 vývodovými konektormi. Pomocou týchto konektorov je adaptér pripevnený na vzdelávaciu platformu Minerva. Ako uvádzame na obrázku 2.1, využívame aj druhý 2x25 vývodový konektor. Vývody tohto konektora nie sú nikde pripojené. Slúži len na zvýšenie stability celej rozširujúcej dosky. Dosku sme sa snažili navrhnuť tak, aby mal užívateľ po pripojení rozširujúceho modulu k dispozícii dostupné všetky vývody, ktoré vzdelávacia platforma ponúka. Zároveň, sme chceli vytvoriť priestor aj tlačidlám ktoré sú umiestnené na spodnej časti vzdelávacej platformy Minerva. Výsledný návrh zobrazujeme v prílohe na obrázkoch A.6 a A.7.

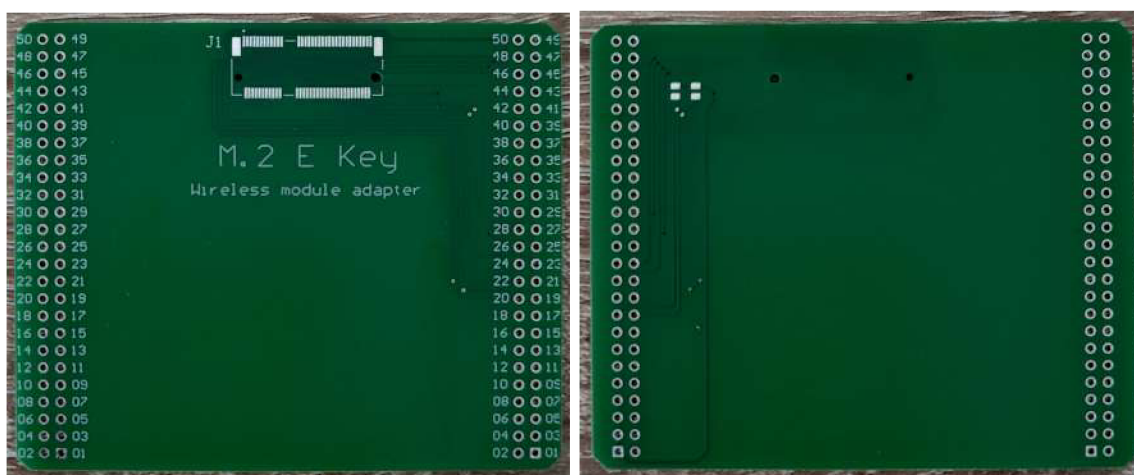
#### Rozširujúci modul

Pri návrhu tejto časti, bolo pre nás najťažšie vytvoriť správny odtlačok pre M.2 konektor. Po jeho úspešnom vytvorení, sme postupovali ku kresleniu približnej veľkosti dosky. Do dosky sme následne vsadili všetky použité súčiastky tak, aby sa neprekrývali. Pri rozložení súčiastok sme museli upriamiť pozornosť na to, aby antény vybraných modulov smerovali von z dosky plošného spoja. Inak by nefungovali správne. V ďalšom kroku sme jednotlivé súčiastky správne prepojili. Na napájacie cesty sme zvolili väčšiu hrúbku, aby boli schopné preniesť väčší prúd. Nevyužitá miesta sme zaliali zemou. Uľahčilo nám to celý návrh z dôvodu, že nebolo potrebné rozťahovať zem ku každej súčiastke a tiež to znížilo rušenie medzi jednotlivými modulmi. Zároveň, v procese výroby nie je potrebné zbaviť sa veľkého množstva medenej vrstvy. Je potrebné brať ohľad na to, aby táto zem nebola v blízkosti vodiča pripojeného k anténe. K výslednému návrhu sme taktiež pridali otvor pre upevňovaciu skrutku, ktorý danú dosku drží na nami navrhnutom adaptéri. Výsledný návrh predstavujeme v prílohe na obrázkoch A.8 a A.9.

### 3.4 Výroba a kompletizácia dosky plošného spoja

Po vyhotovení návrhu bolo potrebné dosky plošného spoja poslať do výroby. Pre výrobu je nutné vygenerovať takzvané gerber súbory. Gerber súbory vymedzujú ako jednotlivé vrstvy plošného spoja vyzerať. Okrem jednotlivých vrstiev je neodmysliteľnou súčasťou vygenerovať súbor, ktorý popisuje kde sa majú vyvrtáť diery nachádzajúce sa na doske plošného spoja a zároveň, akú by mali mať veľkosť. Zvýšenú pozornosť je potrebné venovať rozmerom medzi jednotlivými objektmi plošného spoja. Dôvodom je, že CNC stroje, ktoré vyrábajú dosky nemajú nekonečnú presnosť. Tieto minimálne a maximálne rozmery sme našli na stránkach výrobcu, u ktorého sme realizovali výrobu našich dosiek. Návrhy boli upravené tak, aby spĺňali tieto podmienky. Následne bolo možné tieto dosky zadať do výroby.

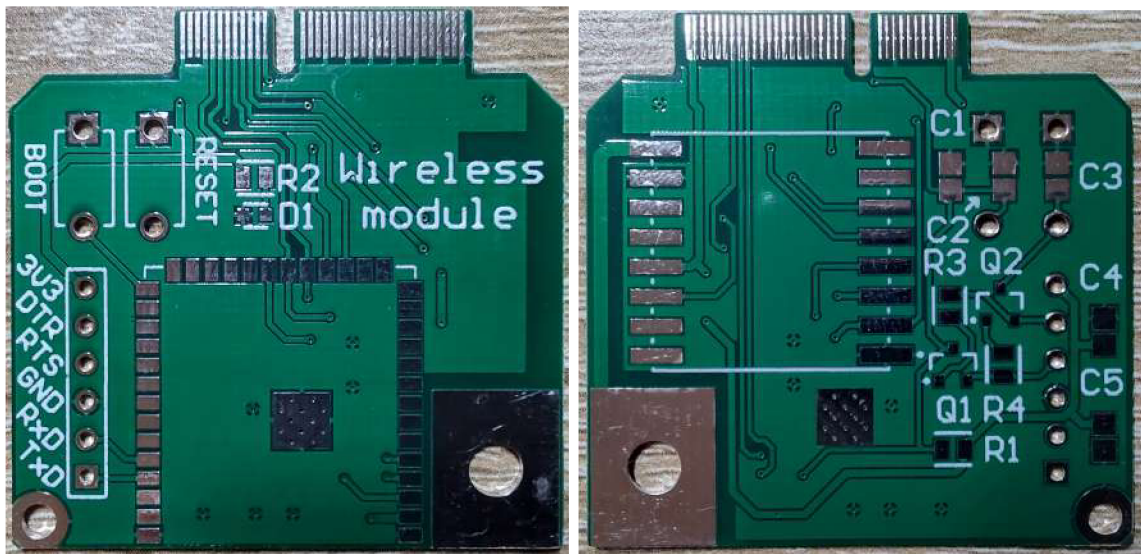
Výsledok výroby je zobrazený na obrázkoch 3.5 a 3.6. Doska určená pre adaptér bola v poriadku a pripravená na osadenie. Nanešťastie, druh výroby v ktorej sme dosky plošného spoja nechávali vyrobiť nám ponúkal len tie dosky, ktoré mali hrúbku 1.5 mm. To pre adaptér bolo dostačujúce ale doska rozširujúceho modulu, ktorá sa vsúva do konektora M.2 mohla prijať maximálnu hrúbku 0.92 mm. Z tohto dôvodu sme dosky nechali vyrobiť duplicitne s hrúbkou 0.8 mm a pre istotu aj s hrúbkou 0.6 mm. Posledný návrh dosiek bezproblémovo sadol do konektora M.2 a boli pripravené na osadenie.



Obr. 3.5: Adaptér — Doska plošného spoja

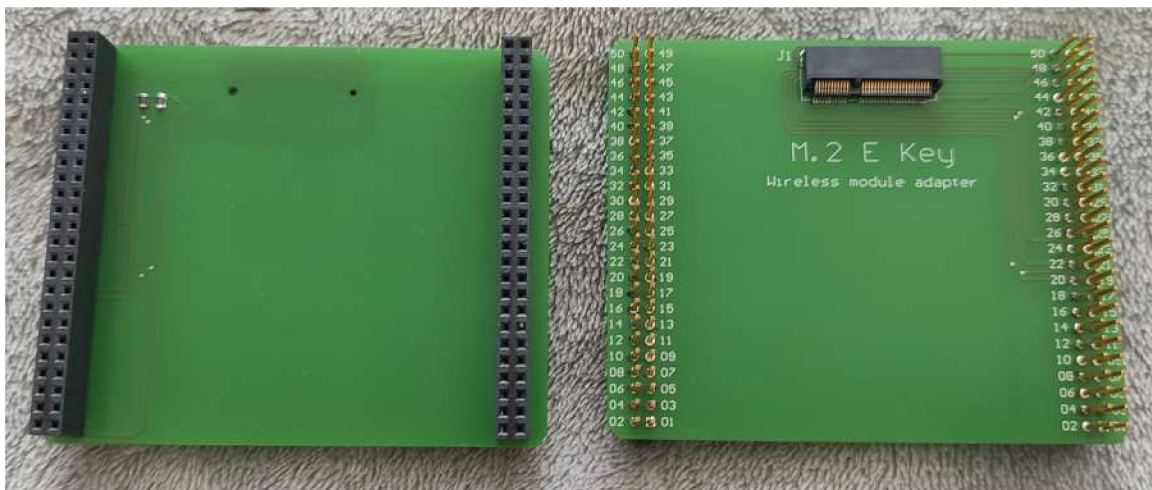
#### Osadenie súčiastok

Na osadenie súčiastok sme potrebovali mikros pájku, cín, medený pásik a spájkovaciu kvapalinu. Keďže väčšia časť súčiastok je typu SMD, použili sme pinzetu a lupu pre efektívnejšiu prácu s týmito súčiastkami. Spájkovať začíname od najnižšej súčiastky po najvyššiu aby nám v priebehu spájkovania väčšie súčiastky nerobili problém. Pred osadením jednotlivých súčiastok sme v prípade pasívnych súčiastok skontrolovali hodnoty odporu alebo kapacity a pri aktívnych súčiastkach skontrolovali PN priedchody. Najväčšou výzvou bolo osadenie mikrokontroléra ESP32-S3 a konektorov M.2. Pri týchto súčiastkach sme si plošky najprv natreli spájkovacou kvapalinou a po ich spájkovaní odstránili prebytočný cín pomocou medeného pásika.

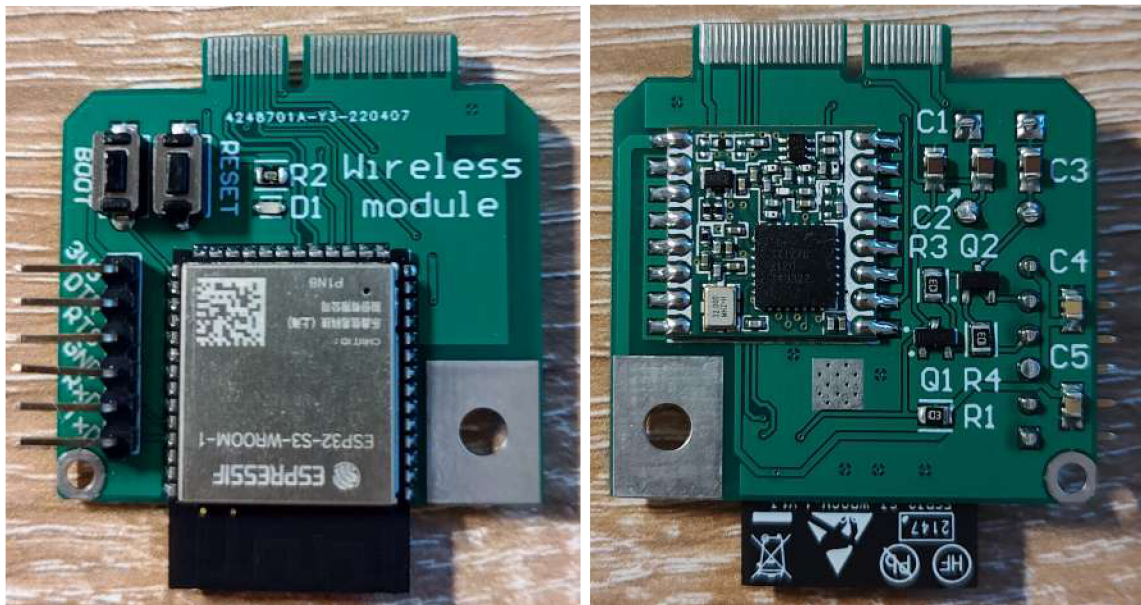


Obr. 3.6: Rozširujúci modul — Doska plošného spoja

Potom, čo boli plošné spoje osadené, bolo potrebné realizovať kontrolu, či jednotlivé spoje nie sú niekde omylom skratované. Na to, sme použili multimeter a prekontrolovali sme ním všetky spoje. Nakoniec, sme rozširujúce moduly vyskúšali naprogramovať a tým sme otestovali, či všetky súčiastky fungujú podľa našich očakávaní. Osadili sme celkovo dva adaptéry a štyri rozširujúce moduly. Dva s hrúbkou 0.8 mm a dva s hrúbkou 0.6 mm. Výsledné skompletizované dosky sú zobrazené na obrázkoch 3.7 a 3.8. Rozširujúci modul pripevnený na vzdelávaciu platformu Minerva uvádzame v prílohe na obrázku A.10.



Obr. 3.7: Adaptér — Osadená doska plošného spoja



Obr. 3.8: Rozšiřující modul — Osadená doska plošného spoja

## Kapitola 4

# Tvorba programového vybavenia

V tejto kapitole sa budeme primárne venovať softwarovej časti našej práce. Rozširujúci modul je po hardwarovej stránke skompletizovaný, jednoducho povedané hotový. Nasleduje potreba navrhnuť štruktúrovanú knižnicu, pomocou ktorej budeme modul ovládať. Keďže mikrokontrolér ESP32-S3 rovnako neobsahuje žiadny sprievodný software, mali sme potrebu navrhnuť a implementovať obslužný firmware tiež pre daný mikrokontrolér.

V časti 2.1 sme sa zamerali na mikrokontrolér Kinetis K60, ktorý sa nachádza na vzdelávacej platforme Minerva, a rozhodli sme sa ho programovať pomocou MCUXpresso IDE a k nemu dostupnému SDK. Mikrokontrolér ESP32-S3 má takisto viac možností akým spôsobom je pre neho vyvíjať software. Jedným z nich je napríklad podpora využitia Arduino frameworku<sup>1</sup>, ktorá nám umožňuje jednoduchú prácu z vstupom a výstupom. Zároveň, hromadu obslužných knižníc, či už pre Wi-Fi, Bluetooth alebo LoRa. Nevýhodou je, že z pohľadu vývojára nemáme tak rozšírené možnosti vývoja a sme obmedzení iba na dostupné funkcie, ktoré nám Arduino framework poskytuje. Ďalšou možnosťou bolo nahráť do mikrokontroléra MicroPython<sup>2</sup>, čo je odľahčená verzia jazyka Python určená pre vývoj vstavaných aplikácií. My sme sa vydali cestou použitia ESP-IDF frameworku.

### ESP-IDF

ESP-IDF[35] je framework vyvíjaný priamo spoločnosťou Espressif Systems a je určený na vývoj softwaru pre nimi ponúkané mikrokontroléri. Medzi ktoré spadá aj nami vybraný mikrokontrolér ESP32-S3. ESP-IDF je framework založený na jazyku C ale je možné nastaviť si aj podporu jazyka C++. Je dostupný v dvoch variantoch a to ako nástroj v príkazovom riadku alebo ako rozšírenie do textového editora VSCode, či Eclipse. Na písanie programov preferujeme textový editor VSCode. Na základe toho sme sa rozhodli využiť rozšírenie pre tento nástroj<sup>3</sup>.

Rozšírenie nám ponúka možnosť výberu typu dosky, ktorú ideme programovať. Ďalej výber rozhrania pomocou ktorého ju budeme programovať, nastavenie konfigurácie, zostavenie programu, nahranie programu do mikrokontroléra a taktiež pripojenie sa k sériovému monitoru, na ktorom môžeme sledovať ladiace výstupy. Poskytuje nám mnoho štruktúrovaných knižníc a príkladov, ako dané knižnice využívať. Pre účely našej práce sme využili knižnicu pre obsluhu Wi-Fi a ovládač pre prácu s SPI rozhraním.

<sup>1</sup><https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>

<sup>2</sup><https://docs.micropython.org/en/latest/esp32/tutorial/intro.html>

<sup>3</sup><https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/get-started/vscode-setup.html>



## ESP32-S3 firmware

Projekt, ktorý sme vytvorili s použitím ESP-IDF rozšírenia a ktorým realizujeme firmware určený pre mikrokontrolér ESP32-S3 sa nazýva `wireless_modul`. Štruktúra tohto projektu sa skladá zo zdrojového súboru `main.c`, v ktorom sa nachádza hlavná funkcia `main` a z dvoch modulov. Prvý modul je určený na prácu s SPI transakciami a skladá sa zo zdrojového súboru `module_spi.c` a hlavičkového súboru `module_spi.h`. V druhom module sa nachádzajú kompletné definície príkazov a definície potrebné pre prácu s nimi. Tento modul je zložený rovnako z dvoch súborov a to zdrojový súbor `module_commands.c` a hlavičkový súbor `module_commands.h`.

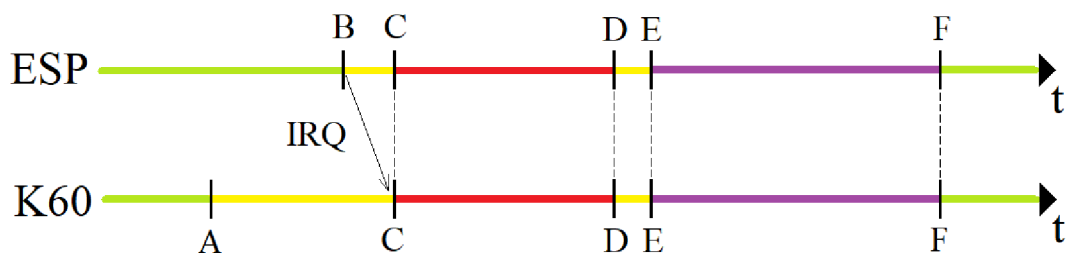
### 4.1 SPI komunikácia medzi MCU a modulom

Pre správne ovládanie modulu bolo potrebné aby sme správne navrhli spôsob akým sa bude mikrokontrolér Kinetis K60 dorozumievať s mikrokontrolérom ESP-S3 rozširujúcim modulom.

Oba vývojové prostredia ponúkajú ovládač, určený pre prácu s SPI rozhraním. Pri SPI komunikácii je mikrokontrolér K60 v role Master, ktorý inicializuje komunikáciu. Komponenty rozširujúceho modulu sú v role Slave, ktoré na inicializovanú komunikáciu reagujú. V prípade modulu RFM95W zasielame adresu registra a hodnotu, ktorú do neho chceme zapísať a tým ovládame celé správanie modulu. V prípade ESP modulu sme daný komunikačný protokol navrhli mierne odlišne.

#### Komunikácia medzi ESP32-S3 a Kinetis K60

SPI komunikácia medzi týmito zariadeniami funguje na princípe transakcií, kde každá transakcia obsahuje reťazec dát, ktorý chceme odoslať, ďalej reťazec dát, ktorý chceme prijať a ich dĺžku. Nevýhodou je, že pred inicializáciou transakcie od Master-a musí mať Slave už predom pripravenú transakciu. Inak Master prijme iba prázdny nezmyselný reťazec dát. Je potrebné dať Master-ovi nejakým spôsobom vedieť, že je transakcia pripravená a je možné zahájiť komunikáciu. Pre tento účel využívame signál ESP IRQ, ktorého zmena stavu z logickej nuly na jednotku vyvolá na strane Master-a prerušenie. Tým mu dá vedieť, že môže zahájiť komunikáciu. Tento signál používame aj na identifikáciu toho, že došlo k prerušeniu zo strany ESP a je potrebná obsluha tohto prerušenia. Na rozlíšenie týchto dvoch typov komunikácií použijeme transakciu skladajúcu sa z dvoch fáz. V prvej fáze zašleme typ transakcie, ktorý vyvolá prerušenie. V druhej fáze hodnoty, ktoré chceme preposlať. V poradí druhá transakcia už nevyvolá prerušenie z dôvodu, že transakciu obsluhujeme z riadiaceho programu. Pre lepšie pochopenie uvádzame obrázok 4.1.



Obr. 4.1: Časový priebeh SPI komunikácie

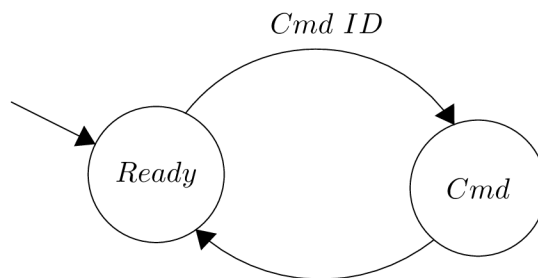
V čase A sa Master rozhodne, že chce začať transakciu. To však nemôže urobiť pokiaľ nedostane povolenie, že je transakcia pripravená na strane Slave-a. Medzitým si Slave v čase B pripraví transakciu a pomocou signálu ESP IRQ dá vedieť Master-ovi, že je pripravený. Daným signálom sa na strane Master-a v čase C vyvolá obsluha prerušenia. Tá zahájí prvú fázu a zistí o aký typ transakcie ide. Po ukončení prvej fázy môžu nastať dva prípady. Buď sa zistí, že ide o normálnu transakciu a tým pádom Master dostane povolenie zahájiť druhú fázu transakcie, ktorá nie je súčasťou obsluhy prerušenia a v čase E začne posielat potrebné dáta. Druhý prípad sa týka prerušenia zo strany Slave-a. Tým pádom v čase E je druhá fáza stále súčasťou obsluhy prerušenia, kde sa pomocou ďalšej transakcie zistí typ prerušenia a k nemu sa spustí príslušná obsluha, ktorá môže obsahovať ďalšie transakcie. Tie by mali byť zo strany Slave pripravené. V čase F končí druhá fáza a celý cyklus sa môže začať odznova.

## Spúšťanie jednotlivých príkazov

Každý z príkazov disponuje vlastným identifikátorom. Pred začatím príkazu je teda potrebné zaslať Slave-ovi správu o aký druh príkazu ide. Ďalej sa komunikácia líši v závislosti od druhu príkazu. Niektoré príkazy potrebujú previesť viacero transakcií, kde zasielajú napríklad veľkosť prijatých, či odoslaných dát. Následne, samotné dáta a v závere hodnotu návratového kódu. Ako je možné vidieť, aby sme boli schopní preposlať identifikátor príkazu potrebujeme mať istotu, že transakcia ktorá prijíma identifikátor príkazu je na strane Slave-a pripravená. To docielime takým spôsobom, že hneď po štarte Slave-a si pripravíme transakciu na prijatie identifikátora príkazu a tým sa dostaneme do stavu Ready. Po prijatí identifikátora sa spustí obsluha daného príkazu a po jeho dokončení si Slave opäť pripraví transakciu, ktorá prijíma identifikátor príkazu. Tým sa znova dostane do stavu Ready. Pre lepšie pochopenie uvádzame stanovy diagram na obrázku 4.2.

## Obsluha prerušení

Sme schopní spúšťať príkazy inicializované z Master-ovej strany ale čo ak budeme chcieť inicializovať komunikáciu zo Slave-ovej strany. Jednou z takýchto situácií môže byť napríklad prijatie dát, či už pomocou Wi-Fi alebo Bluetooth. V týchto situáciách budeme chcieť, aby v čase keď budú dáta dostupné, Slave informoval Master-a o tom, že nastala situácia na ktorú by mal reagovať. Pri tejto situácii využijeme transakciu typu prerušenie. V časti 4.1 sme čiastočne popísali ako obsluha takéhoto typu prerušenia prebieha z Master-ovej strany. Kládli sme si otázku, ako docielime to, že budeme schopní v danom čase spustiť obsluhu



Obr. 4.2: Stanový diagram mikrokontroléra ESP32-S3

prerušená bez toho, aby sme narušili celistvosť ostatných príkazov? Spôsob akým sme to docielili je, že sme dané prerušenie zahájili len vtedy, ak sa nachádzame v stave Ready. To znamená, že Slave bude čakať na to aby prijal príkaz. Zahájenie tohto prerušenia spravíme pomocou prepnutia signálu ESP IRQ z logickej nuly na jednotku. Následne sa Master za pomoci obslužnej rutiny postará o to, aby Slave dostal príkaz na obsluhu prerušenia. V takomto príkaze sa najprv prepošle o aký typ prerušenia sa jedná a na základe toho, sú pripravené transakcie na poslanie potrebných dát.

## 4.2 Návrh štruktúrovanej knižnice

Jednou z najdôležitejších častí tejto práce, je vytvorenie štruktúrovanej knižnice, pomocou ktorej budeme náš rozširujúci modul riadiť. V časti 4.1 sme si vysvetlili ako medzi sebou budú tieto zariadenia komunikovať. Aktuálne je potrebné navrhnúť to, ako budú vyzerat jednotlivé príkazy na základe ktorých budeme rozširujúci modul riadiť.

Prvý krok, ktorý sme realizovali bolo vytvorenie potrebných súborov. Jedná sa o zdrojový súbor `wireless_module.c` v ktorom sa nachádzajú všetky definície použitých funkcií, a k nemu najdôležitejší hlavičkový súbor `wireless_module.h`, v ktorom sa nachádzajú definície použitých makier, deklarácie štruktúr a funkcií. Zároveň sa tam nachádza potrebná dokumentácia k tomu.

V ďalšom kroku sme si náš hlavičkový súbor rozdelili do dvoch častí. Definície, kde sa nachádzajú všetky definície makier a dátových štruktúr, ktoré potrebujeme. Druhá časť je aplikačné rozhranie, v ktorom sa nachádzajú kompletne deklarácie funkcií, ktoré naša štruktúrovaná knižnica poskytuje. Obe časti sme si vo výsledku rozdelili do štyroch skupín. Prvá skupina sú všeobecné definície a funkcie. Druhá skupina, sú definície a funkcie určené pre prácu s Wi-Fi. Tretia skupina sú definície a funkcie pre prácu s Bluetooth a posledná, štvrtá skupina je určená pre definície a funkcie pre prácu s LoRa modulom.

V našej práci sa venujeme výhradne všeobecnej skupine a skupine pre prácu s Wi-Fi. Možné rozšírenie knižnice o ostatné skupiny je popísané v časti 4.2.

### Možnosti rozšírenia

Naša práca venuje pozornosť len Wi-Fi komunikácii. Je preto potrebné definovať ako je možné, nami vytvorenú štruktúrovanú knižnicu rozšíriť o ďalšie vlastnosti, ktoré by mohol náš rozširujúci modul ponúkať. Vyššie v práci sme sa venovali tomu, že naša knižnica je rozdelená do niekoľkých skupín.

V prípade, ak budeme chcieť pridať možnosť práce s LoRa modulom. V štruktúrovanej knižnici je pripravené miesto, do ktorého sa budú jednotlivé definície a deklarácie funkcií

vpisovať. V našom prípade už modul RFM95W obsahuje sprievodný firmware, čiže nastáva potreba nastaviť správne SPI komunikáciu podľa referenčného manuálu a vytvoriť toto rozhranie.

Zložitejšie to je s rozšírením o Bluetooth a ďalšie funkcie Wi-Fi rozhrania. O to sa stará mikrokontrolér ESP32-S3 a funkcie tohto rozhrania sme realizovali za pomoci vytvorených príkazov. Pri tvorbe nového príkazu je potrebné identifikátor tohto príkazu pridať do definície kódov príkazov pod názvom `WLM_command_t` v súbore `wireless_module.c`. Ďalej pridáme deklaráciu a definíciu funkcie, kde na začiatku danej funkcie pomocou SPI komunikácie zašleme identifikátor príkazu. Prijatá hodnota by sa po tejto transakcii mala rovnať `CMD_CHECK` konštante. Z dôvodu aby sme skontrolovali, že SPI komunikácia dopadla dobre a modul sa nachádza v správnom stave. Následne posielame dáta podľa potreby. Je individuálne ako si túto komunikáciu navrhne. Zo strany obslužného firmwaru pre ESP32-S3 je potreba znova pridať identifikátor tohto príkazu do definície príkazov a hlavnej funkcie pridať nový prípad do konštrukcie `switch()`. Ten by mal vyzeráť takto. Ak sa prijatý identifikátor rovná nášmu novému príkazu, spustí obslužnú funkciu, ktorá realizuje tento príkaz. Definíciu a deklaráciu danej obslužnej funkcie vložíme do modulu `module_commands`. Pri SPI komunikácii používame funkcie z modulu `module_spi`. Pre lepšie pochopenie môžete nazrieť na to, ako sú realizované existujúce príkazy.

#### 4.2.1 Všeobecné definície a funkcie

V prvom kroku sme zdefinovali, ktoré vývody sú použité pri práci s daným rozširujúcim modulom. Za týmto účelom boli vytvorené makrá `WLM_$(označenie_signálu)_pinNum`. V prípade využitia tohto rozširujúceho modulu na inej platforme, je možné čísla vývodov zmeniť podľa potreby.

#### Návratové kódy

Ďalšou podstatnou časťou je definícia návratových kódov. V tejto časti špecifikujeme o aký druh návratového kódu sa jedná. Je potrebné venovať pozornosť niektorým návratovým kódom, pretože môžu disponovať vlastným významom pri rozličnej funkcii. Návratové kódy, ktoré vám môže daná funkcia vrátiť a ich presný význam, je možné nájsť v dokumentácii nad deklaráciou vybranej funkcie. Jediný návratový kód, ktorý by sa mal pri všetkých funkciách zhodovať je `WLMStatus_Success` pri úspešnom vykonaní danej funkcie. Ak si nie ste istí, že daná funkcia dopadne úspešne je dobrým zvykom skontrolovať, či funkcia skončila s týmto návratovým kódom.

#### Inicializácia a de-inicializácia

Väčšina periférií potrebuje funkciu určenú pre inicializáciu. Nie je tomu inak ani v tomto prípade. Pomocou funkcie `WLM_InitModule()` inicializujeme všetky použité vývody a komunikačné rozhrania. V našom prípade ide o rozhranie SPI2. V závere zresetujeme oba využité komponenty, aby sme dosiahli očakávaného správania. Táto funkcia by mala byť použitá pred volaním všetkých ostatných funkcií, ktoré ponúka naša štruktúrovaná knižnica.

Späťou s touto funkciou je funkcia `WLM_DeinitModule()`, ktorá sa stará o presný opak. Táto funkcia by mala byť použitá až pri ukončení práce s týmto modulom.

## Kontrola prítomnosti modulu

Poslednou funkciou, ktorá patrí do tejto skupiny je `WLM_IsPresent()` a s ňou spätá definícia návratového typu `WLM_connection_t`. Pri potrebe zistiť, či je náš rozširujúci modul správne pripnutý k doske a SPI komunikácia funguje ako má, využije sa presne daná funkcia. Pokiaľ je modul prítomný a SPI komunikácia funguje, návratový kód je `WLM_Present`. V opačnom prípade je to `WLM_NotPresent`. Túto funkciu je rovnako doporučené používať pred začatím práce s rozširujúcim modulom.

### 4.2.2 Definície a funkcie pre prácu s Wi-Fi

V tejto časti našu pozornosť venujeme definíciám a funkciám určeným pre prácu s Wi-Fi. Pri návrhu tejto časti sme sa zamýšľali nad tým, ako by mohlo vyzeráť rozhranie. Potrebovali sme aby poskytovalo dostatočne veľkú škálu možností a pritom bolo priamočiare pre jednoduchú prácu s ním samotným. Rozhodli sme sa, že rozhranie bude poskytovať Wi-Fi zariadenie v role stanice alebo v role prístupového bodu. Prístupový bod podporujeme z dôvodu, aby sme nemuseli byť závislí na prítomnosti inej Wi-Fi siete, ak by sme chceli aby vybrané dva moduly komunikovali medzi sebou. Táto časť obsahuje funkcie na inicializáciu a de-inicializáciu týchto zariadení. Zvyšné funkcie tejto časti sú rozdelené pre prácu v role prístupového bodu, alebo v role stanice. Pokiaľ je funkcia určená pre inú rolu ako tú v ktorej je dané Wi-Fi zariadenie inicializované, funkcia vráti návratový kód `WLMStatus_BadMode`.

### Inicializácia a de-inicializácia

Pred začiatkom práce s Wi-Fi, potrebujeme správne inicializovať Wi-Fi zariadenie. Vyššie v texte uvádzame, že inicializovať ho môžeme do dvoch rolí. Na inicializáciu využijeme funkciu `WLM_WIFI_Init(WLM_WIFI_mode_t mode)`. Ako parametre zadáme možnosť `WIFI_MODE_STA` pre stanicu alebo `WIFI_MODE_AP` pre prístupový bod. Po tejto inicializácii môžeme využívať ďalšie funkcie, ktoré sú určené pre vybranú rolu.

Pokiaľ potrebujeme zistiť, v akej role sa práve nachádzame alebo, či je zariadenie inicializované, môžeme využiť funkciu `WLM_WIFI_GetMode(WLM_WIFI_mode_t* mode)`. Tá, nám vráti `WLMStatus_NotInitialized`, ak Wi-Fi zariadenie nebolo inicializované. Ak skončí úspešne, tak do premennej ktorú sme jej predali ako parameter, bude zapísaná aktuálna rola v ktorej sa Wi-Fi zariadenie nachádza.

Funkciu `WLM_WIFI_Deinit()` je potreba zavolať vždy, pokiaľ chceme ukončiť prácu s daným Wi-Fi zariadením.

### Vytvorenie prístupového bodu a práca s ním

Pokiaľ máme záujem vytvoriť prístupový bod, je potrebné najprv inicializovať zariadenie v role prístupového bodu. Ďalším krokom je spustenie tohto prístupového bodu. Použijeme na to funkciu `WLM_WIFI_AP_Start(WLM_WIFI_AP_config_t config)`, ktorej ako parameter dodáme konfiguráciu daného prístupového bodu. V konfigurácii sa nachádza zvolený názov prístupového bodu, heslo, zabezpečenie, maximálny počet pripojených zariadení, použitý kanál a je možné dokonca špecifikovať, či názov prístupového bodu bude skrytý alebo nie. Po úspešnom návrate z tejto funkcie, by mal byť náš prístupový bod dostupný a mali by sme disponovať možnosťou pripojenia sa k nemu.

Pokiaľ budeme chcieť zistiť aké množstvo zariadení je na náš prístupový bod pripojené a informácie o daných zariadeniach, môžeme využiť funkciu `WLM_WIFI_AP_GetListSTA()`. Tá nám pomocou parametru, ktorý jej predáme vráti informáciu o počte pripojených zariadení ale zároveň, aj o ich MAC adrese alebo sile signálu. Túto funkciu môžeme použiť aj vtedy, ak chceme zistiť, ktoré z pripojených zariadení je k nám najbližšie.

Funkciu `WLM_WIFI_AP_Stop()` použijeme pre ukončenie prístupového bodu s danou konfiguráciou. Príklad vytvorenia prístupového bodu je opísaný v časti [4.2.4](#).

## Sken dostupných prístupových bodov

Pokiaľ chceme zistiť, aké prístupové body sa nachádzajú v okolí, môžeme na to využiť funkciu `WLM_WIFI_Scan(WLM_WIFI_AP_record_t* records, uint8_t* size)`. Je potrebné venovať pozornosť tomu, že Wi-Fi zariadenie musí byť inicializované v role stanice. Ako parameter potom zadávame koľko zariadení je možné maximálne zoskenovať. Po prevedení funkcie sa daná hodnota zmení podľa počtu zoskenovaných prístupových bodov. Informácie o týchto prístupových bodoch sú názov, MAC adresa, použitý kanál, zabezpečenie, či sila signálu. Príklad takéhoto skenu prístupových bodov je opísaný v časti [4.2.4](#).

## Pripojenie stanice k prístupovému bodu

V úvode pokusu o pripojenie stanice k prístupovému bodu je potreba mať inicializované zariadenie v role stanice. `WLM_WIFI_STA_Connect(WLM_WIFI_STA_config_t config)` je funkcia, ktorú môžeme využiť za týmto účelom. Ako parameter tejto funkcie je konfigurácia, ktorá hovorí k akému prístupovému bodu sa má stanica pripojiť a ak je prístupový bod zabezpečený, tak je potrebné heslo. Pokiaľ všetko dobre dopadlo a funkcia sa ukončí s úspešným návratovým kódom, daná stanica by mala byť pripojená k špecifikovanému prístupovému bodu. Tiež by mala mať obdržanú IP adresu od DHCP servera. Následne po akcii môžeme začať pracovať s funkciami transportnej vrstvy, ktoré sú popísané v časti [4.2.3](#). Je potrebné venovať pozornosť pri zvolení prístupového bodu, keďže nemusí existovať alebo nie je možné sa k nemu v danom čase pripojiť. V danej situácii nám funkcia vráti `WLMStatus_NotConnected`.

V prípade toho, že mienime zistiť informácie o prístupovom bode ku ktorému sme pripojení, môžeme využiť funkciu `WLM_WIFI_STA_GetAP(WLM_WIFI_AP_record_t* record)`. Pokiaľ je zariadenie pripojené, funkcia nám vráti pomocou premennej, ktorú sme zadali ako parameter, informácie o danom prístupovom bode. Ak zariadenie nie je pripojené k žiadnemu prístupovému bodu, funkcia sa vráti z návratovým kódom `WLMStatus_NotConnected`. Funkciu môžeme využiť v prípade, že chceme zistiť, či sme stále pripojení k prístupovému bodu a naše spojenie nezlyhalo.

Funkciu `WLM_WIFI_STA_Disconnect()` využijeme vždy, pokiaľ chceme zrušiť spojenie s daným prístupovým bodom, alebo ukončiť prácu s Wi-Fi zariadením v role stanice s danou konfiguráciou. Príklad vytvorenia stanice a následného pokusu o pripojenie k prístupovému bodu je opísaný v časti [4.2.4](#).

### 4.2.3 Transportná vrstva

V predchádzajúcej časti sme si uviedli, ako je možné nadviazať spojenie s inými Wi-Fi zariadeniami. Na to aby celá bezdrôtová komunikácia dávala zmysel, je potrebné mať možnosť posielat dáta k jednotlivým zariadeniam s ktorými sme v spojení. S tým nám pomôže práca z TCP alebo UDP paketmi, ktorá je súčasťou transportnej vrstvy TCP/IP sieťového

modulu. V krátkosti popíšeme ako funguje. Funguje tak, že každé zariadenie v sieti má pridelenú IP adresu a jednotlivé procesy pracujú s paketmi naviazanými na určitý port. Port teda používame na rozlíšenie jednotlivých procesov prebiehajúcich na danom procesore. Pre tvorbu nášho softwaru sme si vybrali takzvané bare metal riešenie. Znamená to, že na danom mikrokontroléri bude možné spustiť iba jeden proces. To je dôvodom, prečo sme pri návrhu aplikačného rozhrania, našej transportnej vrstvy nedali možnosť špecifikovať port na aký sa má daný paket naviazať. Pre možnosť komunikácie medzi zariadeniami sme vytvorili UDP paket, ktorý sa viaže na lokálny port. Pomocou neho sme schopní prijímať a odosielať dáta na špecifikovanú IP adresu.

## Vytvorenie UDP Paketu a prijímanie dát

Pokiaľ chceme začať komunikovať s ostatnými zariadeniami v Wi-Fi sieti, budeme si musieť vytvoriť UDP paket, ktorý nám túto možnosť poskytne. Funkcia, ktorá vytvorí UDP paket počívajúci na lokálnom porte sa nazýva `WLM_WIFI_UDP_Start(callback)`. Parameter tejto funkcie je nami vytvorená funkcia, ktorá sa zavolá po tom, čo UDP paket prijme nejaké dáta. Pokiaľ chceme ignorovať prijaté dáta, môžeme namiesto tohto parametru dosadiť hodnotu `NULL`. Nami vytvorená funkcia, ktorá bude reagovať na prijaté dáta, má ako parameter veľkosť prijatých dát, IP adresu odosielateľa a samotné dáta. Zároveň, pozornosť je potrebné venovať tomu, že táto funkcia je stále súčasťou obsluhy prerušenia. Tým pádom ostatné funkcie naše štruktúrovanej knižnice v nej nejde použiť. To ale vôbec neprekáža, keďže je to možné vyriešiť napríklad prepínačom, ktorý v tejto funkcii budeme nastavovať a na základe neho budú z hlavnej funkcie volané príslušné príkazy nášho rozhrania.

## Odosielanie dát

Pre možnosť byť schopní dáta taktiež posielat, budeme musieť najprv vytvoriť UDP paket. Pomocou funkcie `WLM_WIFI_UDP_SendTo(uint8_t* data, uint32_t len, uint8_t dst_ip[4])`, je následne možné posielat špecifikovaný počet dát na určitú adresu. Opäť zvýšime pozornosť, maximálna posielaná veľkosť dát je 1024 bajtov. Pokiaľ chceme nejaké dáta niekam odoslať potrebujeme vedieť IP adresu zariadenia kam budú následne smerovať. Aby sme boli pomocou danej štruktúrovanej knižnice schopní určiť akú má naše zariadenie IP adresu, bola vytvorená funkcia `WLM_WIFI_IP_GetInfo(WLM_WIFI_IP_info_t* ip_info)`. Táto funkcia nám vráti informácie o pridelení IP adrese, IP adrese siete v ktorej sa nachádzame, či sieťovej maske. Využitie vybranej funkcie popisujeme v príklade, ktorý je v časti [4.2.4](#).

## Ukončenie UDP paketu

Funkciu `WLM_WIFI_UDP_Stop()` je potrebné zavolať pri ukončení práce s UDP paketom. V časti [4.2.4](#) môžete nájsť taktiež opísaný príklad využitia UDP paketu v role prijímateľa. Príklad využitia UDP paketu v role odosielateľa je opísaný v časti [4.2.4](#), ako sme spomínali vyššie v práci.

### 4.2.4 Príklady použitia

V každej dobrej štruktúrovanej knižnici, by sa mali nachádzať príklady, ako správne používať rozhranie ktoré ponúka. Nie je tomu inak ani v našom prípade. Rozhodli sme sa preto

vytvoriť zopár príkladov, ako napríklad zoskenovať dostupné prístupové body, tvorba prístupového bodu alebo stanice. Nakoniec prácu s UDP paketom. Všetky zmienené príklady popisujeme nižšie v texte.

## Sken dostupných prístupových bodov

Tento príklad sa nachádza v priečinku s názvom `WIFI_Scan`. V zdrojovom súbore `source/WIFI_Scan.c` nájdeme jeden zo spôsobov akým si dokážeme vypísať informácie o dostupných prístupových bodoch.

Najprv skontrolujeme, či je rozširujúci modul prítomný. Ak nie je, tak program skončí. Potom je potrebné inicializovať si Wi-Fi zariadenie v role stanice. To zrealizujeme za pomoci funkcie `WLM_WIFI_Init(WIFI_MODE_STA)`. Ak inicializácia dopadla dobre, môžeme si vytvoriť premenné, ktoré budú uchovávať informácie o zoskenovaných prístupových bodoch. Ďalej prevedieme sken pomocou funkcie `WLM_WIFI_Scan(records, &size)`. Pokiaľ sa funkcia ukončí úspešne, môžeme tieto informácie vypísať na výstup za pomoci sériového monitoru. Nakoniec všetko správne de-inicializujeme.

V hornej časti zdrojového súboru sa nachádza makro `MAX_RECORDS`, pomocou ktorého špecifikujeme maximálny počet zoskenovaných zariadení.

## Vytvorenie Wi-Fi v role stanice

Daný príklad sa nachádza v priečinku v názvom `WIFI_Station`. V zdrojovom súbore `source/WIFI_Station.c` nájdeme jeden zo spôsobov akým je možné nadviazať spojenie s prístupovým bodom.

Najprv si skontrolujeme prítomnosť rozširujúceho modulu. Pokiaľ nie je prítomný, program sa ukončí. Následne je potrebné inicializovať Wi-Fi zariadenie v role stanice. Správime to za pomoci funkcie `WLM_WIFI_Init(WIFI_MODE_STA)`. Ak nedošlo k chybe pokúsime sa pomocou funkcie `WLM_WIFI_STA_Connect(sta_config)` pripojiť k špecifikovanému prístupovému bodu. Pokiaľ spojenie dopadlo dobre, vypíšeme si informácie o prístupovom bode, ktoré dostaneme pomocou funkcie `WLM_WIFI_STA_GetAP(&record)`. V závere, sa z prístupového bodu odpojíme a de-inicializujeme Wi-Fi zariadenie.

V hornej časti zdrojového súboru sa nachádza makro `RETRY_NUM`. Za pomoci neho špecifikujeme počet pokusov o pripojenie. Pomocou makra `SSID` a `PASSWORD` zadáme heslo a názov prístupového bodu, ku ktorému sa chceme pripojiť.

## Vytvorenie Wi-Fi v role prístupového bodu

Tento príklad sa nachádza v priečinku s názvom `WIFI_AccessPoint`. V zdrojovom súbore `source/WIFI_AccessPoint.c` sa nachádza jeden zo spôsobov, ako môžeme vytvoriť prístupový bod.

V prvom kroku si skontrolujeme, či je rozširujúci modul prítomný. Ak nie, program sa ukončí. Potom je potrebné inicializovať si Wi-Fi zariadenie v role prístupového bodu. To spravíme za pomoci funkcie `WLM_WIFI_Init(WIFI_MODE_AP)`. Pokiaľ nenastala chyba, vytvoríme konfiguráciu pre náš prístupový bod. Prístupový bod spustíme za pomoci funkcie `WLM_WIFI_AP_Start(config)`. Ak je prístupový bod spustený, je možné sa k nemu pripojiť. Následne, periodicky vypisujeme informácie o pripojených zariadeniach. K získaniu týchto informácií využívame funkciu `WLM_WIFI_AP_GetListSTA(&list)`. Ak by niekde nastala chyba, prístupový bod sa ukončí a zariadenie Wi-Fi sa de-inicializuje.



V hornej časti zdrojového súboru sa nachádzajú makrá SSID, PASSWORD, CHANNEL a MAX\_CONN za pomociu ktorých, môžeme špecifikovať vlastnosti nášho prístupového bodu.

### Využitie UDP paketu v role odosielateľa

Uvedený príklad sa nachádza v priečinku s názvom WIFI\_UDP\_Sender. V zdrojovom súbore source/WIFI\_UDP\_Sender.c nájdeme využitie UDP paketu v role odosielateľa.

Wi-Fi zariadenie pre daný príklad, bude nakonfigurované v role stanice. Po skontrolovaní, či je modul prítomný a pripojený k prístupovému bodu, môžeme začať pracovať s UDP paketom. Vybraný paket budeme využívať iba v roli odosielateľa. Z toho dôvodu nás nebudú zaujímať prijaté dáta. Docielime to použitím funkcie WLM\_WIFI\_UDP\_Start(NULL), kde ako parameter zadáme hodnotu NULL. Po tom, čo je UDP paket správne vytvorený začíname periodicky zasielať správu zariadeniu so špecifikovanou IP adresou. Využívame na to funkciu WLM\_WIFI\_UDP\_SendTo(data, strlen((char\*)data)+1, dst\_ip). Ak nastane strata spojenia, UDP paket sa ukončí a dôjde k de-inicializácii Wi-Fi zariadenia a ukončeniu celého programu.

V hornej časti zdrojového súboru sa nachádzajú makrá SSID, PASSWORD pomocou ktorých, špecifikujeme na aký prístupový bod sa chceme pripojiť. Za pomoci makra DEST\_IP\_ADDR špecifikujeme na akú IP adresu sa majú dáta zasielať.

### Využitie UDP paketu v role prijímateľa

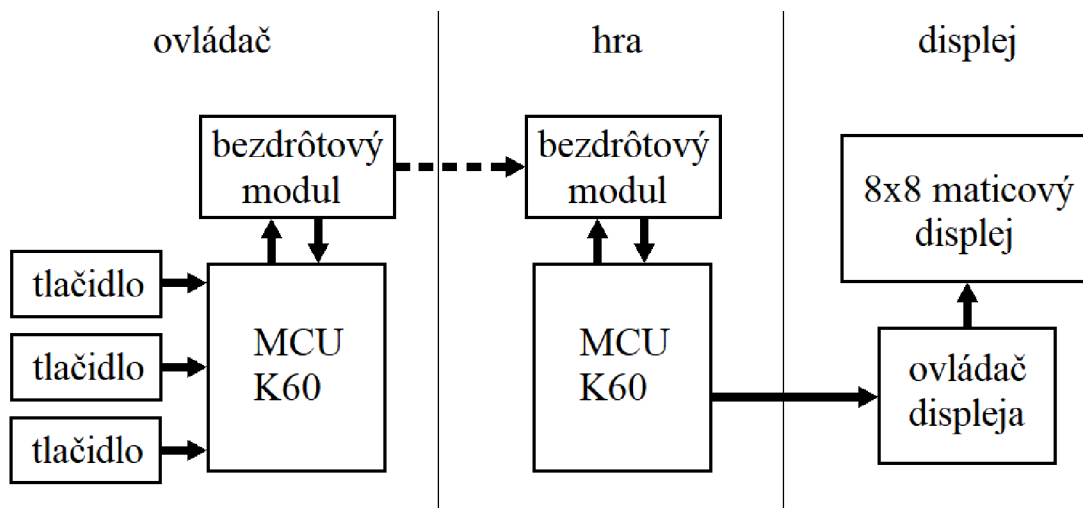
Tento príklad má miesto v priečinku s názvom WIFI\_UDP\_Receiver. V zdrojovom súbore source/WIFI\_UDP\_Receiver.c nájdeme využitie UDP paketu v role prijímateľa.

Wi-Fi zariadenie pre tento príklad bude nakonfigurované v role prístupového bodu. Po skontrolovaní, či je modul prítomný, spustíme náš prístupový bod. Predtým než vytvoríme potrebný UDP paket, zdefinujeme si obslužnú funkciu, ktorá sa zavolá pri prijatí nových dát. V našom prípade sa táto funkcia nazýva RXcallback(uint8\_t\* data, uint32\_t len, uint8\_t src\_ip[4]) a jediné čo robí je, že vypíše prijaté dáta a adresu odosielateľa na výstup. To všetko za pomoci sériovej komunikácie. Následne je možné vytvoriť UDP paket pomocou funkcie WLM\_WIFI\_UDP\_Start(RXcallback), kde parametrom je nami definovaná obslužná funkcia. V závere program stojí v nekonečnom cykle a čaká pokiaľ nebude spustená obslužná funkcia. Zvyšok kódu je len pre ukážku toho, ako by mal vyzeráť koniec práce s UDP paketom a Wi-Fi zariadením.

V hornej časti zdrojového súboru sa nachádzajú makrá SSID, PASSWORD, CHANNEL a MAX\_CONN za pomociu ktorých, môžeme špecifikovať vlastnosti nášho prístupového bodu.

## 4.3 Demonštračná aplikácia

Za účelom demonštrácie funkčnosti sme sa zamerali na známu hru had. Na jej zhotovenie budeme potrebovať zobrazovacie médium, ktoré nám bude hru zobrazovať. Ďalej budeme potrebovať zariadenie, na ktorom bude naša hra prebiehať. Aby sme využili bezdrôtovú komunikáciu a tým demonštrovali funkčnosť našich modulov, využijeme v našej aplikácii ďalšie zariadenie, ktoré bude v role bezdrôtového ovládača. Na obrázku 4.3 je vidieť celú štruktúru zapojenia jednotlivých zariadení a tok informácií medzi nimi.



Obr. 4.3: Bloková schéma zapojenia

## Ovládač

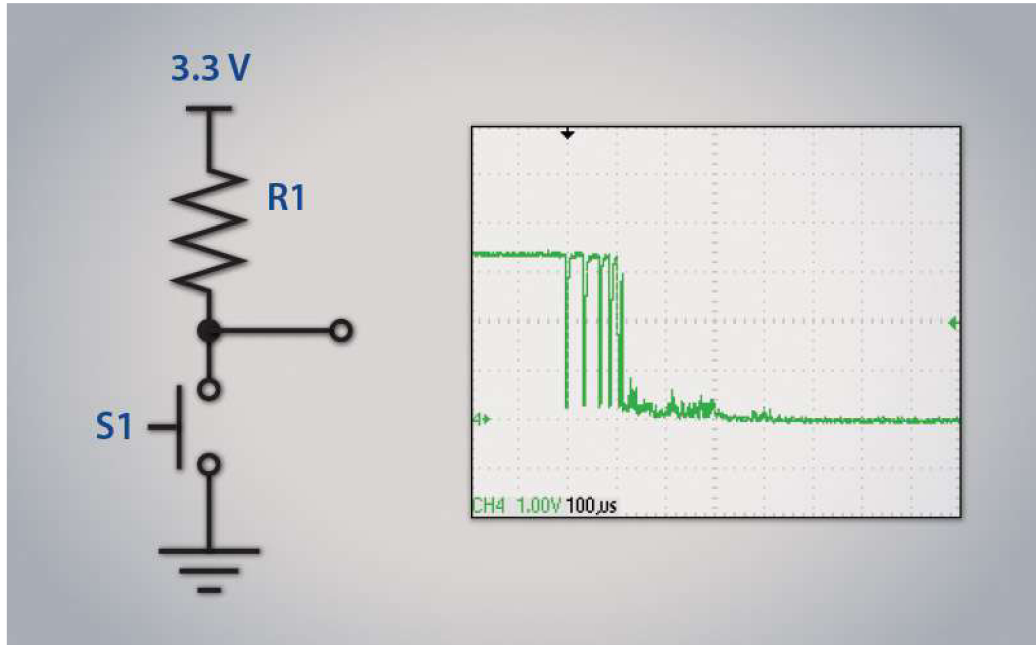
Ovládacia časť by mala byť určená na riadenie smeru hada a spúšťanie novej hry. Ako môžeme na obrázku 4.3 vidieť, v ľavej časti označenej ako ovládač, k ovládaniu využijeme tri tlačidlá. Prvé bude zamerané na zmenu smeru hada o 90 stupňov vľavo, druhé na zmenu smeru o 90 stupňov vpravo. Posledné, tretie na spustenie novej hry, či už na úplnom začiatku alebo po ukončení hry. Projekt, ktorý túto časť realizuje sa nazýva SNAKE\_RemoteControl. Nami implementovaný zdrojový súbor, ktorý obsahuje hlavnú logiku je pomenovaný source/SNAKE\_RemoveContorl.c. Ďalej v projekte nájdete nami vytvorenú štruktúrovanú knižnicu na obsluhu rozširujúceho bezdrôtového modulu v priečinku drivers/, skladajúcu sa z hlavičkového súboru wireless\_module.h a zdrojového súboru wireless\_module.c.

Na vzdelávacej platforme Minerva máme dostupných zopár tlačidiel, využijeme práve tieto a nebudeme vytvárať externé vstupné zariadenie, ktoré by obsahovalo zapojenie týchto tlačidiel. Z softwarovej stránky sa o snímanie stlačenia jednotlivých tlačidiel postaráme pomocou obsluhy prerušenia. Je to z dôvodu, že nechceme zbytočne zatažovať procesor tým, aby musel periodicky kontrolovať hodnoty na jednotlivých vstupoch a podľa toho zistiť, či došlo k stlačeniu. Na prácu z prerušením sme museli najprv inicializovať jednotlivé vývody, ktoré sú napojené k tlačidlám na doske. Vývody boli inicializované, ako vstupné a bolo potrebné povoliť im prerušenie na zostupnú hranu, keďže v nestlačenej stave je na vstup privedená logická jednotka. Po stlačení sa tento vstup uzemní a tým pádom, dôjde k zostupnej hrane. Ku ktorým vývodom sú jednotlivé tlačidlá zapojené sme zistili zo schémy zapojenia vzdelávacej platformy Minerva[29]. Inicializáciu týchto vývodov môžeme nájsť vo funkcii `initButtons()`.

Keďže máme prerušenie nastavené a povolené, je potrebné k tomu vytvoriť obslužnú rutinu. Táto rutina sa implicitne nazýva `PORTE_IRQHandler()`, keďže naše tlačidlá sú pripojené k portu E. V tejto rutine rozlišujeme, aké tlačidlo dané prerušenie vyvolalo a označíme si, že dané tlačidlo bolo stlačené.

V hlavnej funkcii `main()` následne inicializujeme Wi-Fi v role stanice a pripojíme sa k zvolenému prístupovému bodu. Pokiaľ sa spojenie naviazalo, kontrolujeme či nedošlo k stlačeniu tlačidiel. Ak áno, zasielame správu zariadeniu kde prebieha hra o tom, aké tlačidlo bolo stlačené. V prípade stlačenia tlačidla môže dôjsť k zvlneniu signálu, ako môžeme

vidieť na obrázku 4.4. To je možné vyhodnotiť ako viacnásobné stlačenie tlačidla. Potrebovali sme po stlačení nejakého z tlačidiel chvíľu počkať, aby nenastala daná chyba. Ak sa spojenie nenaviaže alebo v priebehu programu stratí, ovládač sa znova snaží naviazať toto spojenie.



Obr. 4.4: Zavlnenie signálu pri stlačení tlačidla[14]

### Zariadenie, v ktorom prebieha hra

Ako už názov napovedá, v tomto zariadení bude prebiehať hra. Zariadenie sa bude starať o priebeh hry, zmeny stavu na základe prijatých informácií od ovládača a posielanie informácií, ktoré budeme chcieť vykresliť na displej. Ako ukazujeme na obrázku 4.3, táto časť sa skladá z dvoch komponentov. Zo vzdelávacej platformy Minerva a rozširujúceho bezdrôtového modulu. Projekt, ktorý realizuje túto časť po programovej stránke sa nazýva SNAKE\_game. Hlavný nami implementovaný zdrojový súbor je možné nájsť pod názvom source/SNAKE\_game.c. Ďalej v priečinku drivers/ môžete nájsť nami vytvorenú štruktúrovanú knižnicu pre obsluhu rozširujúceho modulu ale rovnako aj knižnicu pre prácu s ovládačom displeja. Využívame ju vtedy, ak chceme niečo vykresliť na displej. Skladá sa z hlavičkového súboru 8x8\_display.h a zdrojového súboru 8x\_display.c.

K tomu aby sme mohli na pripojený displej niečo vykresliť, potrebujeme ho najprv inicializovať. S ovládačom tohto displeja komunikujeme pomocou I2C rozhrania. Pri inicializácii si potrebujeme povoliť vývody rozhrania I2C0 a inicializovať dané rozhranie. Výpis na displeji spravíme pomocou poslania ôsmich bajtov, v ktorých je zakódovaný stav displeja. Následne už len v rámci inicializácie displej vyčistíme.

V hlavnej funkcii potrebujeme znova inicializovať Wi-Fi, aby sme dokázali komunikovať s ovládačom. Vybraný projekt sme navrhli takým spôsobom, že pred nahraním projektu si môžeme zvoliť v akej role bude náš Wi-Fi modul operovať. Ak by sme v blízkosti nemali žiadnu použiteľnú Wi-Fi sieť, môžeme si definovaním makra USE\_AP vybrať možnosť, že Wi-Fi modul bude operovať v role prístupového bodu. Iná možnosť je tá, že využijeme

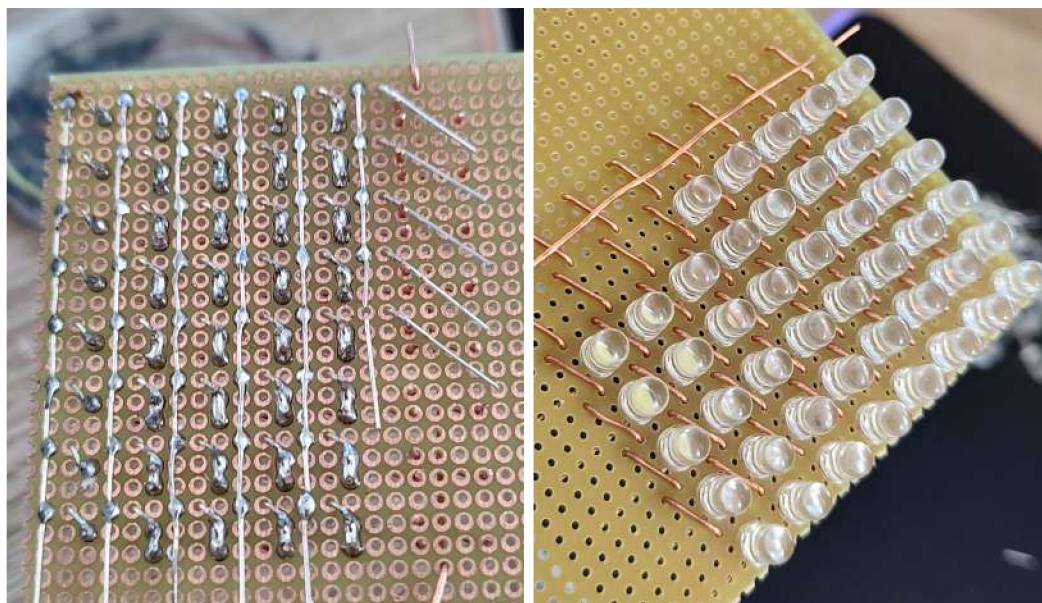
pripojenie k existujúcemu prístupovému bodu. Po inicializácii Wi-Fi si vypíšeme na sériový monitor to, akú IP adresu má dané zariadenie. Následne sme schopní povedať, ktorému zariadeniu má ovládač posielat dáta o stlačených tlačidlách. Potom je potrebné vytvoriť UDP paket, pomocou ktorého budeme prijímať dáta a zaregistrovať funkciu, ktorá sa spustí pri prijatí určitých dát.

Dostaneme sa na základe toho do stavu, kedy čakáme na spustenie samotnej hry. V tomto čase sa na displeji vykreslí štartovací obraz s písmenom S. Ako sme vyššie uviedli, hru spustíme stlačením stredného tlačidla na strane ovládača. Ak bola hra spustená, vygeneruje sa had s dĺžkou jedna a tiež ovocie. Had sa pohybuje na základe nastaveného smeru. Tento smer meníme vo funkcii, ktorá sa zavolá vždy keď UDP paket prijme určité dáta. Podľa týchto dát vieme povedať, aké tlačidlo bolo stlačené a ako sa má zmeniť stav hry. Po zjedení ovocia sa vygeneruje nové ovocie, na náhodnom mieste. Pre generovanie náhodných čísel sme potrebovali vytvoriť počítadlo systémových hodín, ktoré nám počíta hodnotu milisekúnd od spustenia programu. Počítadlo využívame aj na krokovanie nášho hada. Potom, čo had narazí do svojho vlastného tela, celá hra sa ukončí. Na displeji sa zobrazí krížik. Potom sa stlačením stredného tlačidla opäť vrátíme do stavu, kedy čakáme na spustenie novej hry a celý priebeh začína odznova. Daný stav môžeme vidieť aj na obrázku 4.8.

## Zobrazovač

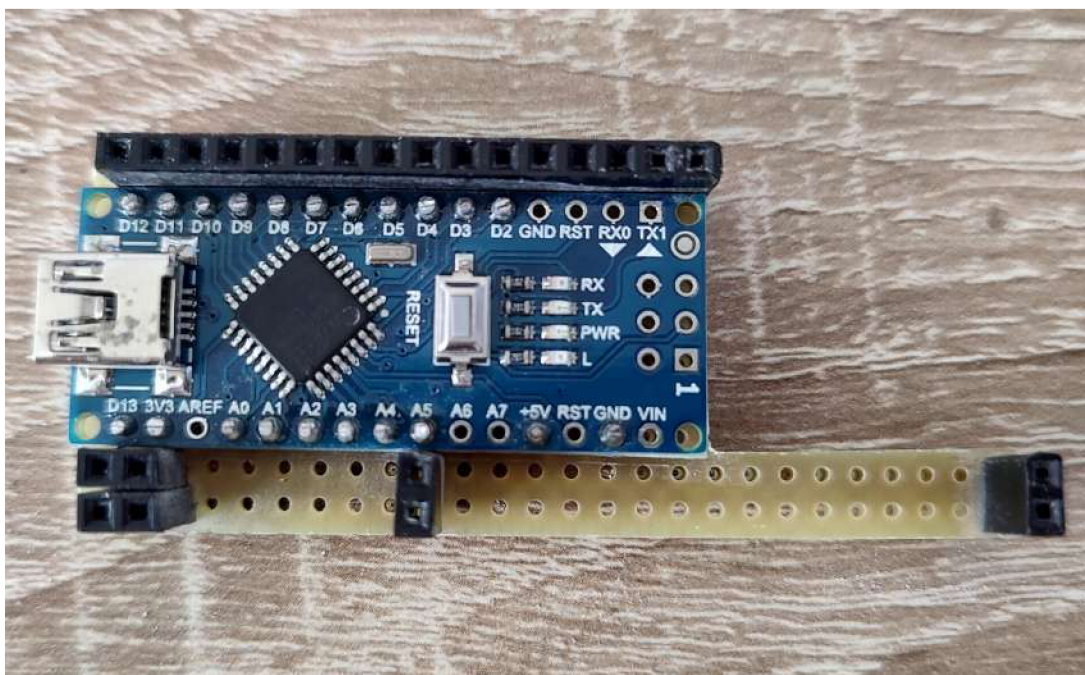
Posledná časť tejto demo aplikácie sa stará o zobrazenie výstupu. Za týmto účelom, sme navrhli a zhotovili 8x8 maticový displej, ku ktorému bolo potrebné vytvoriť ovládač. Vytvárali sme ho z dôvodu, že na výstupných konektoroch vzdelávacej platformy Minerva, nebol dostatočný počet použiteľných vstupno-výstupných vývodov vedúcich od mikrokontroléra Kinetis K60.

Na zhotovenie 8x8 maticového displeja sme použili šesťdesiatštyri LED diód, jeden šesťnásť vývodový konektor a osem ochranných rezistorov. Skupinu týchto súčiastok sme osadili a prepojili za pomoci predom pripravenej dosky plošného spoja, s pripravenými plôškami a otvormi. Zhotovenie tohto displeja je znázornené na obrázku 4.5.



Obr. 4.5: Zhotovenie maticového displeja

Na zhotovenie ovládača sme mohli použiť programovateľné hradlové pole, dostupné na vzdelávacej platforme Minerva. Vybrali sme si možnosť vytvoriť ovládač za pomoci platformy Arduino. Vybraná platforma je známa hlavne kvôli ľahkému spôsobu vývoja aplikácii, bez nutnosti poznania vnútornej štruktúry každého komponentu. Na základe toho, nám zaisťuje dostupné vývojové prostredie ArduinoIDE<sup>4</sup>, kde mikrokontrolér programujeme v jazyku C/C++ a k nemu dostupnému Arduino frameworku. Dané riešenie sme si vybrali z dôvodu, že vďaka nemu, bude náš ovládač viac rozširiteľný a tento displej budeme môcť použiť pri ktoromkoľvek zariadení poskytujúcom I2C rozhranie.



Obr. 4.6: Zhotovenie ovládača pre displej

Na zhotovenie ovládača pre displej sme použili Arduino Nano. Bolo to z dôvodu, jeho malého rozmeru. Konektory a Arduino Nano sme prepojili pomocou predom pripravenej dosky plošného spoja. Zhotovenie daného ovládača pre maticový displej, môžete vidieť na obrázku 4.6.

Pri tvorbe softwaru pre ovládač sme si najskôr inicializovali všetky výstupné vývody a keďže s ovládačom komunikujeme za pomoci I2C rozhrania, aj toto rozhranie ako perifériu s príslušnou adresou. Ďalej sme si definovali funkciu, ktorá sa zavolá ak zariadenie prijme nejaké dáta. Prijaté dáta by mali byť osem bajtov, v ktorých je zakódovaný stav displeja, ktorý chceme rozsvietiť. Nakoniec, vo funkcii loop() rozsvietime jednotlivé riadky podľa prijatých dát. Poslednou vecou, ktorú sme implementovali v tomto riešení je impulzová šírková modulácia výstupného signálu. Bolo to z dôvodu, že keď sme LED diódy rozsvietili na plný výkon, displej bol oslepujúci.

Výsledné dosky plošných spojov sme z estetických dôvodov zaliali taviacim lepidlom a nafarbili čiernou farbou. Výsledok je znázornený na obrázku 4.7.

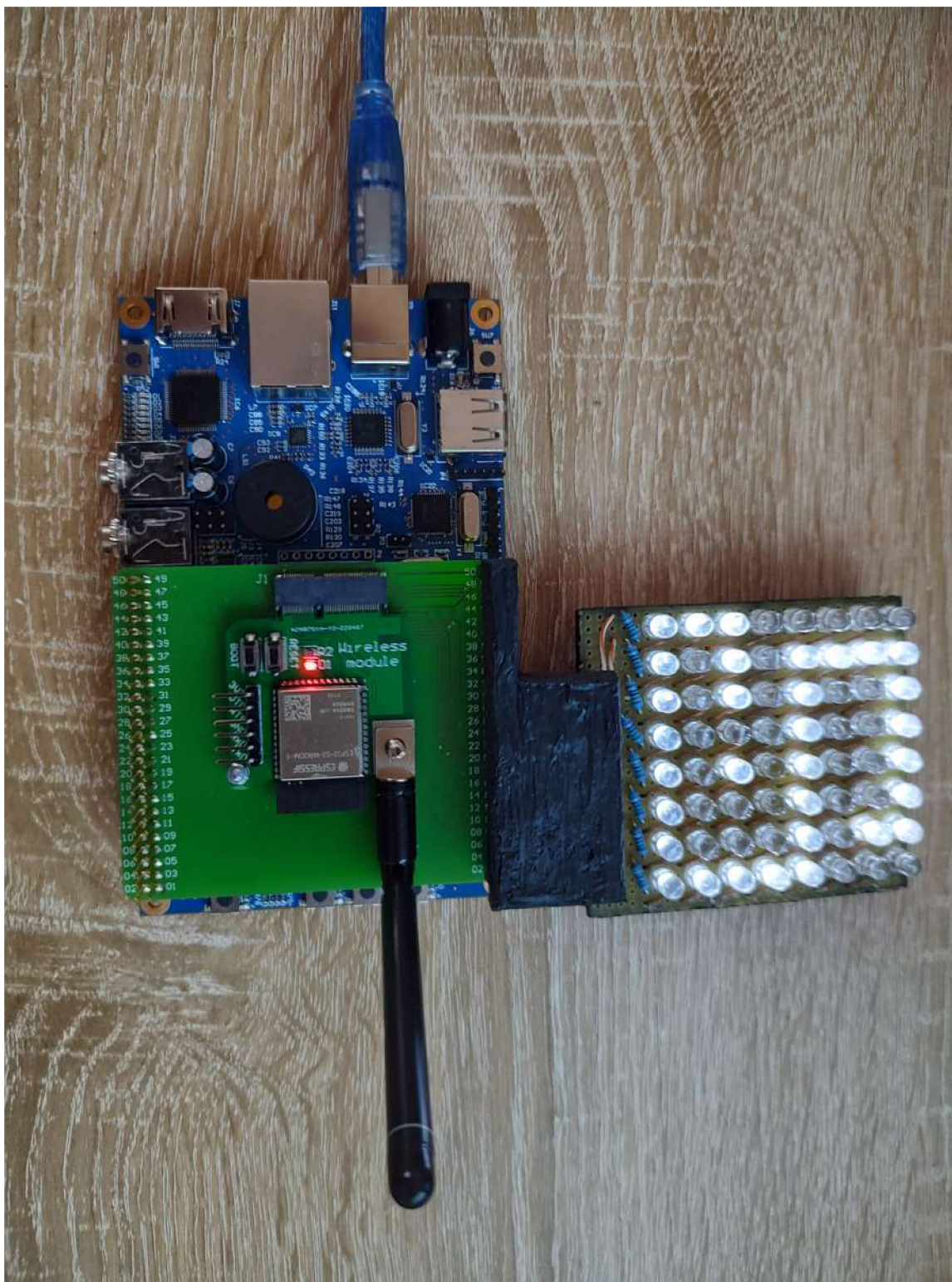
<sup>4</sup><https://www.arduino.cc/en/software>



Obr. 4.7: Maticový displej s ovládačom

## Výsledok

Nakoniec sme všetky komponenty prepojili podľa blokovej schémy na obrázku 4.3 a otestovali, či všetko funguje. Zariadenie, v ktorom prebieha hra sme otestovali v role stanice aj v role prístupového bodu. Hra sa podľa očakávania spustila a čakala dovedy, pokiaľ nebude stlačené stredné tlačidlo. Po pripojení ovládača a podržaní stredného tlačidla sme mohli začať hrať. Na tejto aplikácii bolo účelne demonštrované, že bezdrôtová komunikácia medzi jednotlivými zariadeniami fungovala bezchybne. Všetky naše očakávania sa naplnili.



Obr. 4.8: Stav pred spustením hry

# Kapitola 5

## Záver

Cieľom našej práce bolo vytvoriť rozširujúci modul, ktorý vzdelávacej platforme Minerva poskytne možnosť bezdrôtovej komunikácie. Štandardy, ktoré tento modul poskytuje sú Wi-Fi, Bluetooth a LoRa. Mali sme za potrebu vybrať správne komponenty, ktoré disponujú týmito zvolenými štandardmi. Ďalej sme potrebovali navrhnuť schému zapojenia a dosku plošného spoja. Po zhotovení dosky ju bolo treba osadiť vybranými súčiastkami. Pre jednoduché ovládanie modulu, za pomoci vzdelávacej platformy Minerva, bolo potrebné vytvoriť štruktúrovanú knižnicu. K danej knižnici následne vytvoriť vhodnú aplikáciu, ktorá demonštruje funkčnosť celej práce.

Podarilo sa nám vytvoriť funkčný rozširujúci modul, ktorý je predovšetkým určený pre vzdelávaciu platformu Minerva. Z dôvodu, že bol pri návrhu použitý konektor M.2, tento modul sa dokáže použiť aj pri práci s inými platformami. Pre iné platformy je potreba vytvoriť adaptér, na ktorý by sa modul nasadil. K vzdelávacej platforme Minerva sme vytvorili už spomínaný adaptér, na ktorý sa rozširujúci modul nasadil pomocou konektora M.2. Tento adaptér spoločne s modulom boli navrhnuté tak, aby všetky ostatné konektory vzdelávacej platformy Minerva boli neustále, aj po nasadení dostupné. Komponenty, boli vybrané na základe prieskumu trhu a dostupných komunikačných rozhraní. Návrh schémy zapojenia a dosiek plošných spojov funguje podľa nášho očakávania. Po osadení jednotlivých komponentov, boli výsledné dosky plošných spojov otestované a všetky zhotovené rozširujúce moduly sa dali naprogramovať. Ich funkčnosť naplnila naše očakávania. Boli zhotovené dva kusy adaptéra a štyri kusy rozširujúceho modulu. Pre tento modul bola vytvorená štruktúrovaná knižnica podporujúca prácu s Wi-Fi štandardom. Ku knižnici boli vytvorené príklady použitia, ako správne pracovať s vytvoreným rozhraním. Nakoniec, bola zhotovená demonštračná aplikácia had, ktorá definuje funkčnosť rozširujúceho bezdrôtového modulu. Táto aplikácia fungovala bez problémov.

V našej práci je tiež popísané, ako je možné vytvorenú, štruktúrovanú knižnicu rozšíriť o ďalšie vlastnosti. Keďže sme sa v práci venovali len všeobecným funkciám a funkciám Wi-Fi štandardu, možné rozšírenie je podpora LoRa štandardu a tiež Bluetooth. Štruktúrovanú knižnicu je možno podobne rozšíriť o podporu ďalších protokolov transportnej vrstvy akým je napríklad TCP protokol, alebo protokolov aplikačnej vrstvy, ako napríklad HTTP protokol.

V súčasnom stave je náš rozširujúci modul schopný komunikovať s pomocou Wi-Fi štandardu a tým poskytnúť študentom možnosť vývoja aplikácií s využitím bezdrôtovej komunikácie.



# Literatúra

- [1] ALLIANCE, W. *Wi-Fi Alliance: Certification* [online]. 2020 [cit. 2022-05-02]. Dostupné z: <https://www.wi-fi.org/certification>.
- [2] ALTIUM. *Altium designer* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.altium.com/altium-designer>.
- [3] BLUETOOTH. *Bluetooth Wireless Technology* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>.
- [4] CONNECTIVITY, L. *Sterling 70 Series: Dual-Band 2x2 Wi-Fi 6 + Bluetooth* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.lairdconnect.com/wireless-modules/wifi-modules-bluetooth/sterling-70-series-dual-band-2x2-wi-fi-6-bluetooth>.
- [5] CONNECTIVITY, L. *Sterling-EWB: IoT Module* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.lairdconnect.com/wireless-modules/wifi-modules-bluetooth/sterling-ewb-iot-module>.
- [6] CONNECTIVITY, L. *Sterling-LWB5+: WiFi 5 + Bluetooth 5.2 Module / USB Adapter* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.lairdconnect.com/wireless-modules/wifi-modules-bluetooth/sterling-lwb5-plus-wifi-5-bluetooth-5-module>.
- [7] CONNECTIVITY, L. *Sterling<sup>TM</sup>-LWB+: WiFi 4 and Bluetooth 5.2 Modules* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.lairdconnect.com/wireless-modules/wifi-modules-bluetooth/sterling-lwb-wifi-4-and-bluetooth-52-modules>.
- [8] FREESCALE. *Referencny manual pre Kinetis K60* [online]. 2. vyd. Mnichov: Freescale Semiconductor, jún 2012, rev. 2 Jun 2012 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/webapp/Download?colCode=K60P144M100SF2V2RM>.
- [9] MICROELECTRONICS, H. *RFM95PW: RF LoRa Module* [online]. 2018 [cit. 2022-05-02]. Dostupné z: <https://www.hoperf.com/modules/lora/RFM95P.html>.
- [10] MICROELECTRONICS, H. *RFM95W: LoRa Module* [online]. 2018 [cit. 2022-05-02]. Dostupné z: <https://www.hoperf.com/modules/lora/RFM95.html>.
- [11] MICROELECTRONICS, H. *RFM96W: RF Transceiver* [online]. 2018 [cit. 2022-05-02]. Dostupné z: <https://www.hoperf.com/modules/lora/RFM96.html>.
- [12] MICROELECTRONICS, H. *RFM98PW: LoRa Module* [online]. 2018 [cit. 2022-05-02]. Dostupné z: <https://www.hoperf.com/modules/lora/RFM98P.html>.
- [13] MICROELECTRONICS, H. *RFM98W: RF Transceiver* [online]. 2018 [cit. 2022-05-02]. Dostupné z: <https://www.hoperf.com/modules/lora/RFM98.html>.

- [14] NUVATION. *Switch Debouncing for Electronic Product Designs* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.nuvation.com/resources/article/switch-debouncing-electronic-product-designs>.
- [15] NXP. *88W8801: 2.4 GHz Single-Band 1x1 Wi-Fi® 4 (802.11n) Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-ghz-single-band-1x1-wi-fi-4-802-11n-solution:88W8801>.
- [16] NXP. *88W8887: 1x1 Dual-band Wi-Fi® 5 (802.11ac) and Bluetooth®5.2 Combo Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/1x1-dual-band-wi-fi-5-802-11ac-and-bluetooth5-2-combo-solution:88W8887>.
- [17] NXP. *88W8897: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 5 (802.11ac) + Bluetooth® 5.0 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-2x2-wi-fi-5-802-11ac-plus-bluetooth-5-0-solution:88W8897>.
- [18] NXP. *88W8977: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 4 (802.11n) + Bluetooth® 5.2 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-1x1-wi-fi-4-802-11n-plus-bluetooth-5-2-solution:88W8977>.
- [19] NXP. *88W8987: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 5 (802.11ac) + Bluetooth® 5.2 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-1x1-wi-fi-5-802-11ac-plus-bluetooth-5-2-solution:88W8987>.
- [20] NXP. *88W8997: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 5 (802.11ac) + Bluetooth® 5.3 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-2x2-wi-fi-5-802-11ac-plus-bluetooth-5-3-solution:88W8997>.
- [21] NXP. *88W9098: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 6 (802.11ax) + Bluetooth® 5.3* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-2x2-wi-fi-6-802-11ax-plus-bluetooth-5-3:88W9098>.
- [22] NXP. *IW416: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 4 (802.11n) + Bluetooth® 5.2 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/iw416-2-4-5-ghz-dual-band-1x1-wi-fi-4-802-11n-plus-bluetooth-5-2-solution:IW416>.
- [23] NXP. *IW612: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® 5.2 + 802.15.4 Tri-Radio Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-1x1-wi-fi-6-802-11ax-plus-bluetooth-5-2-plus-802-15-4-tri-radio-solution:IW612>.
- [24] NXP. *IW620: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 6 (802.11ax) + Bluetooth® 5.1 Solution* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/products/wireless/wi-fi-plus-bluetooth/2-4-5-ghz-dual-band-2x2-wi-fi-6-802-11ax-plus-bluetooth-5-1-solution:IW620>.

- [25] NXP. *MCUXpresso: Integrated Development Environment (IDE)* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE>.
- [26] NXP. *MCUXpresso: Software Development Kit (SDK)* [online]. 2006 [cit. 2022-05-02]. Dostupné z: <https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-software-development-kit-sdk:MCUXpresso-SDK>.
- [27] SDA. *M.2 Pinout Descriptions and Reference Designs* [online]. 1. vyd. San Diego: Congatec, január 2020, rev. 28 Jan 2020 [cit. 2022-05-02]. Dostupné z: [https://www.congatec.com/fileadmin/user\\_upload/Documents/Application\\_Notes/AN43\\_M.2\\_Pinout\\_Descriptions\\_and\\_Reference\\_Designs.pdf](https://www.congatec.com/fileadmin/user_upload/Documents/Application_Notes/AN43_M.2_Pinout_Descriptions_and_Reference_Designs.pdf).
- [28] SEMTECH. *What are LoRa® and LoRa WAN®?* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>.
- [29] SIMEK, V. *Schema zapojenia MINERVA kitu*. 0.92. Brno: Fakulta informačních technologií VUT v Brně, júl 2013.
- [30] SYSTEMS, E. *ESP32-S3-DevKitC-1* [online]. 1. vyd. Shanghai: Espressif Systems, marec 2021, rev. 12 Mar 2021 [cit. 2022-05-02]. Dostupné z: [https://dl.espressif.com/dl/SCH\\_ESP32-S3-DEVKITC-1\\_V1\\_20210312C.pdf](https://dl.espressif.com/dl/SCH_ESP32-S3-DEVKITC-1_V1_20210312C.pdf).
- [31] SYSTEMS, E. *Referencny manual pre ESP32-C3-MINI-1* [online]. 1. vyd. Shanghai: Espressif Systems, jún 2021, rev. 21 Jun 2021 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf).
- [32] SYSTEMS, E. *Referencny manual pre ESP32-C3-WROOM-02* [online]. 1. vyd. Shanghai: Espressif Systems, august 2021, rev. 20 Aug 2021 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-c3-wroom-02\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3-wroom-02_datasheet_en.pdf).
- [33] SYSTEMS, E. *Referencny manual pre ESP32-MINI-1* [online]. 1.1. Shanghai: Espressif Systems, november 2021, rev. 9 Nov 2021 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-mini-1\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-mini-1_datasheet_en.pdf).
- [34] SYSTEMS, E. *Referencny manual pre ESP32-S3-MINI-1* [online]. 0.6. Shanghai: Espressif Systems, november 2021, rev. 16 Nov 2021 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-s3-mini-1\\_mini-1u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-s3-mini-1_mini-1u_datasheet_en.pdf).
- [35] SYSTEMS, E. *ESP-IDF Programming Guide* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/index.html>.
- [36] SYSTEMS, E. *Referencny manual pre ESP32-S3-WROOM-1* [online]. 1. vyd. Shanghai: Espressif Systems, apríl 2022, rev. 21 Apr 2022 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1\\_wroom-1u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf).

- [37] SYSTEMS, E. *Referencny manual pre ESP32-WROOM-32E* [online]. 1.4. Shanghai: Espressif Systems, február 2022, rev. 22 Feb 2022 [cit. 2022-05-02]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf).
- [38] UBLOX. *JODY-W2 series: Host-based multiradio modules with Wi-Fi and Bluetooth* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/jody-w2-series>.
- [39] UBLOX. *JODY-W3 series: Host-based automotive modules with Wi-Fi 6 and Bluetooth 5.3* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/jody-w3-series>.
- [40] UBLOX. *MAYA-W1 series: Host-based multiradio modules with Wi-Fi 4 and Bluetooth 5* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/maya-w1-series>.
- [41] UBLOX. *NINA-W10 series (open CPU): Stand-alone multiradio modules* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/nina-w10-series-open-cpu>.
- [42] UBLOX. *NINA-W15 series (u-connect): Stand-alone multiradio modules* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/nina-w15-series-u-connect>.
- [43] UBLOX. *NORA-W2 series: AWS IoT ExpressLink stand-alone multiradio modules* [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.u-blox.com/en/product/nora-w2-series>.

# Príloha A

## Foto-dokumentácia

V tejto prílohe môžeme nájsť hotové schémy zapojenia, návrhy dosiek plošných spojov a fotky výsledných rozširujúcich modulov nasadených na vzdelávacej platforme Minerva.

### Schémy zapojenia

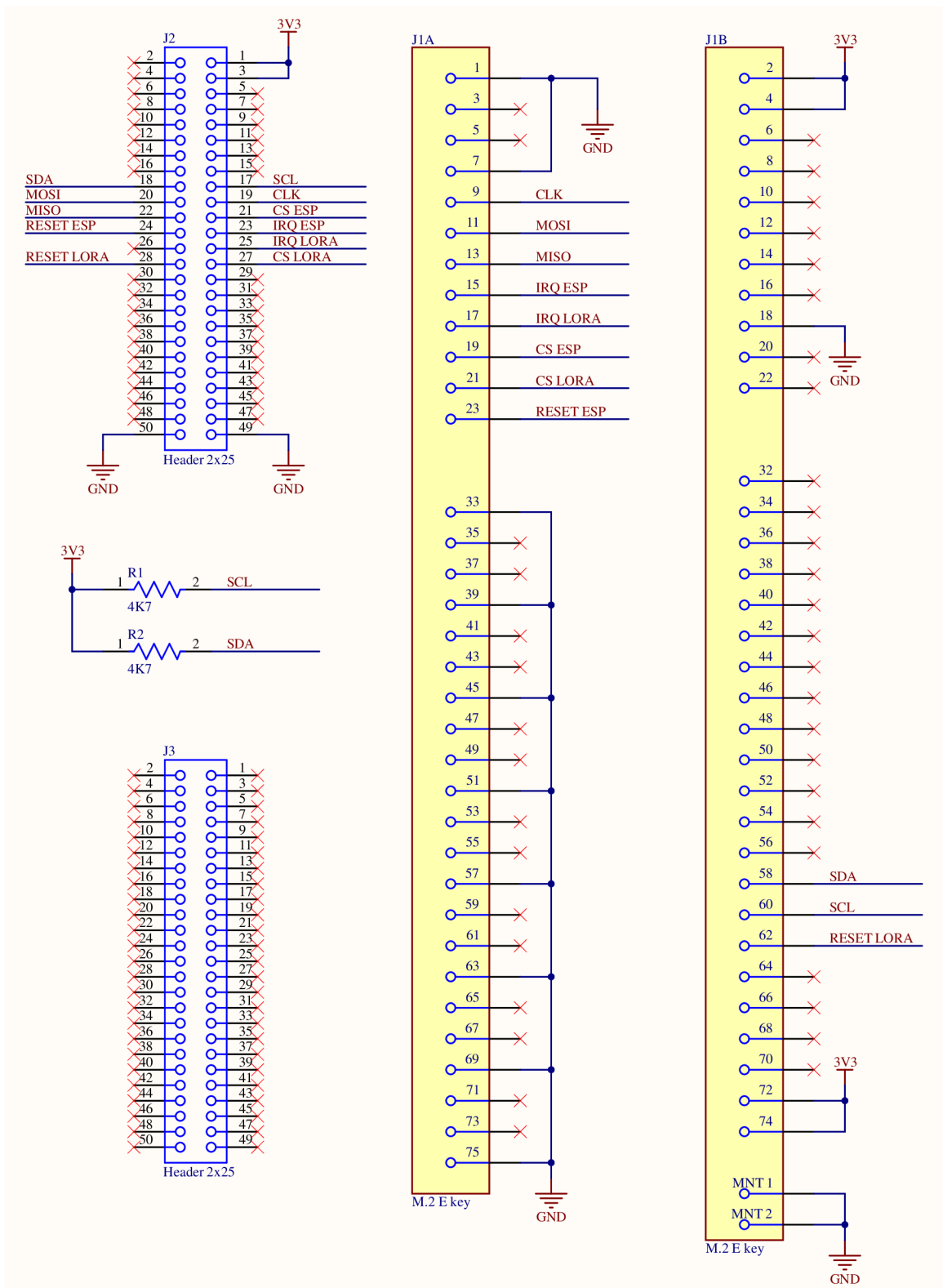
Schéma adaptéra sa skladá z jednej časti, ktorú môžeme vidieť na obrázku [A.1](#). Schéma rozširujúceho modulu je zložitejšia. Preto je rozdelená do viacerých častí. Štruktúra prepojení je ukázaná na obrázku [A.2](#). Schému zapojenia mikrokontroléra ESP32-S3 môžeme nájsť na obrázku [A.3](#). Schému zapojenia modulu RFM95W môžeme nájsť na obrázku [A.4](#). A poslednú schému zapojenia výstupných konektorov môžeme nájsť na obrázku [A.5](#).

### Návrhy dosiek plošných spojov

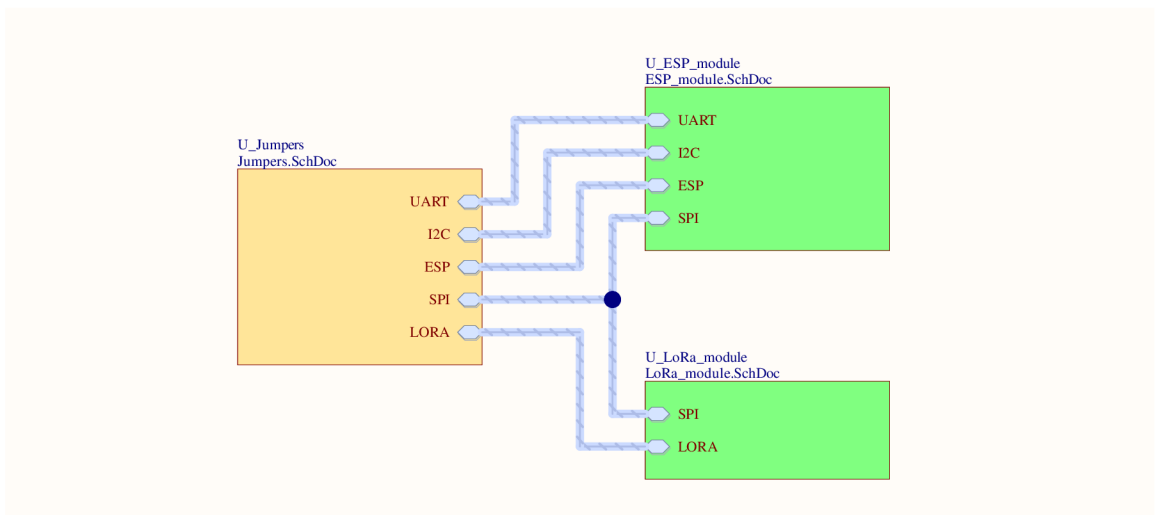
Návrh adaptéra môžeme vidieť na obrázkoch [A.6](#) a [A.7](#).  
Návrh rozširujúceho modulu môžeme vidieť na obrázkoch [A.8](#) a [A.9](#).

### Výsledný rozširujúci modul

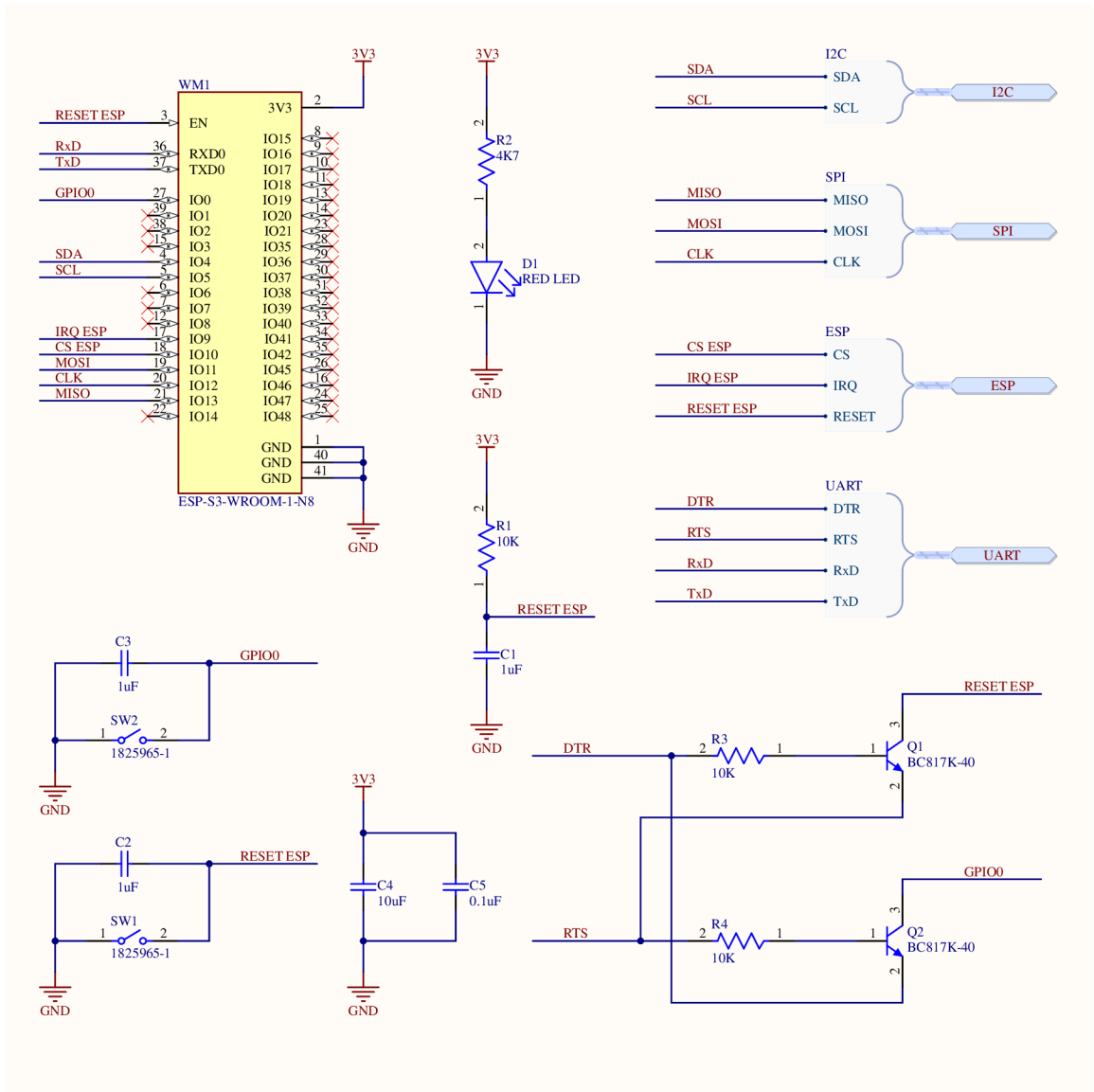
Na obrázku [A.10](#), môžeme vidieť skompletizované a pripevnené rozširujúce moduly k vzdelávacím platformám Minerva.



Obr. A.1: Schéma zapojenia adaptéra

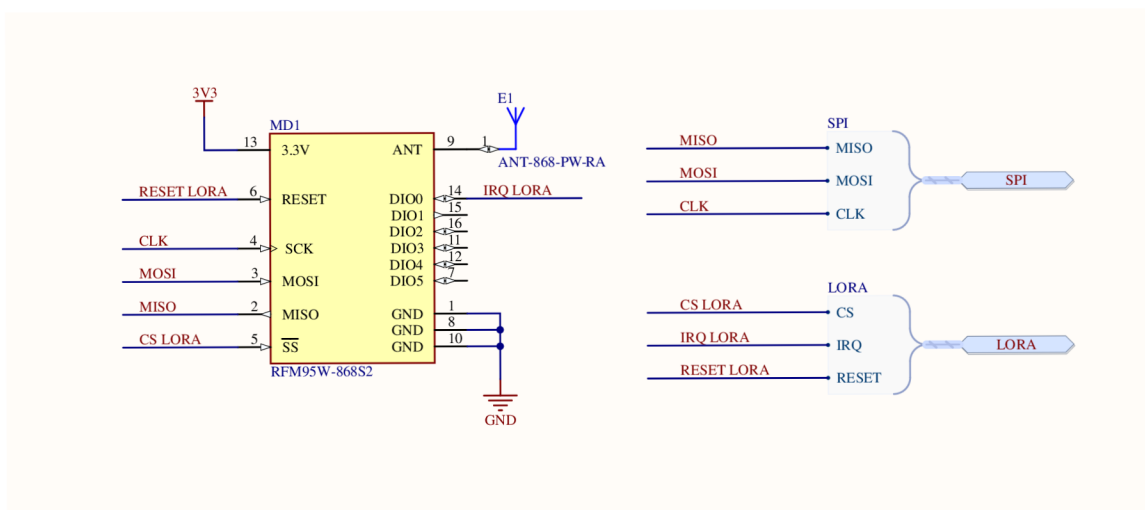


Obr. A.2: Štruktúra schémy zapojenia bezdrôtového modulu

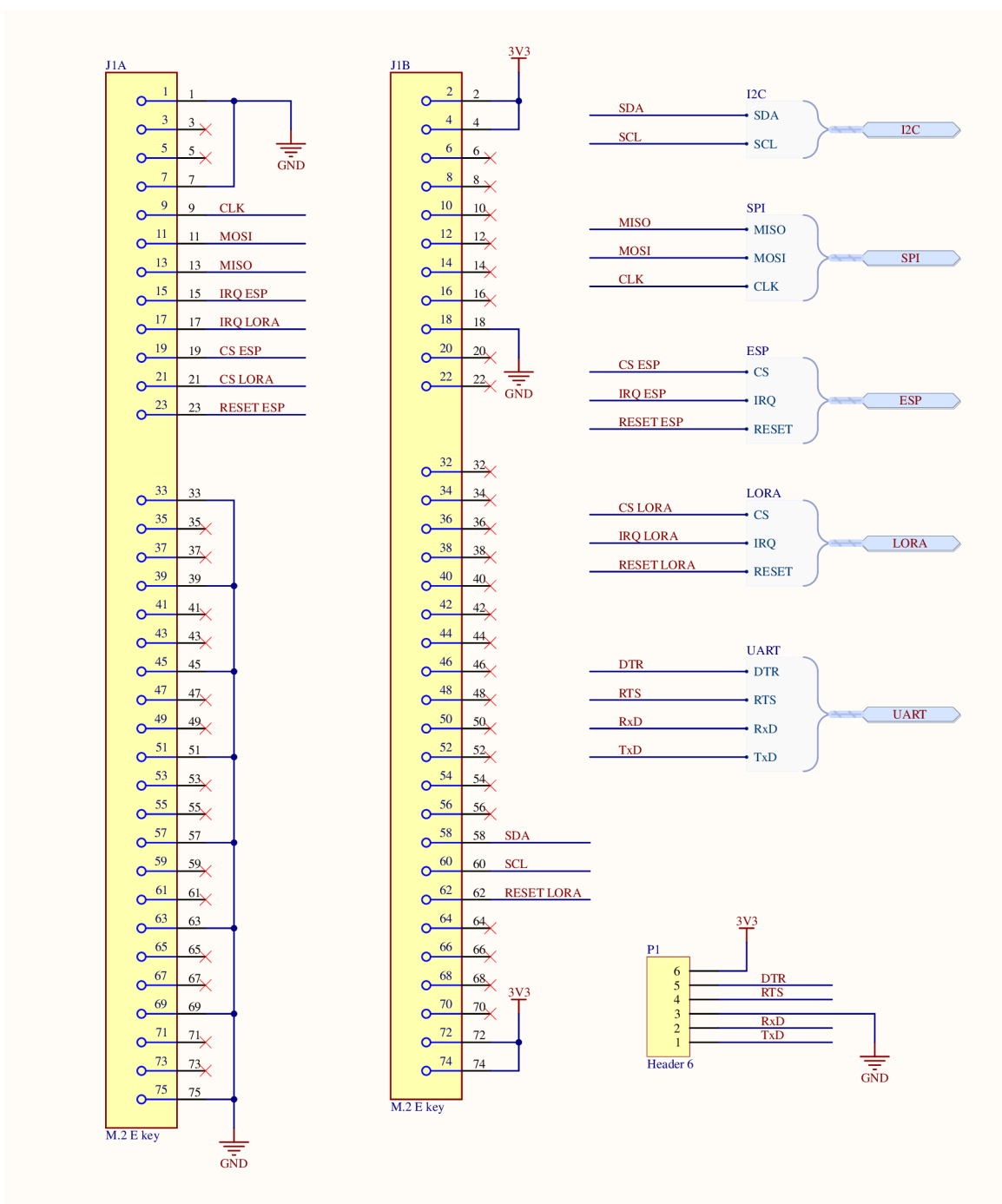


Obr. A.3: Schéma zapojenia mikrokontroléra ESP32-S3

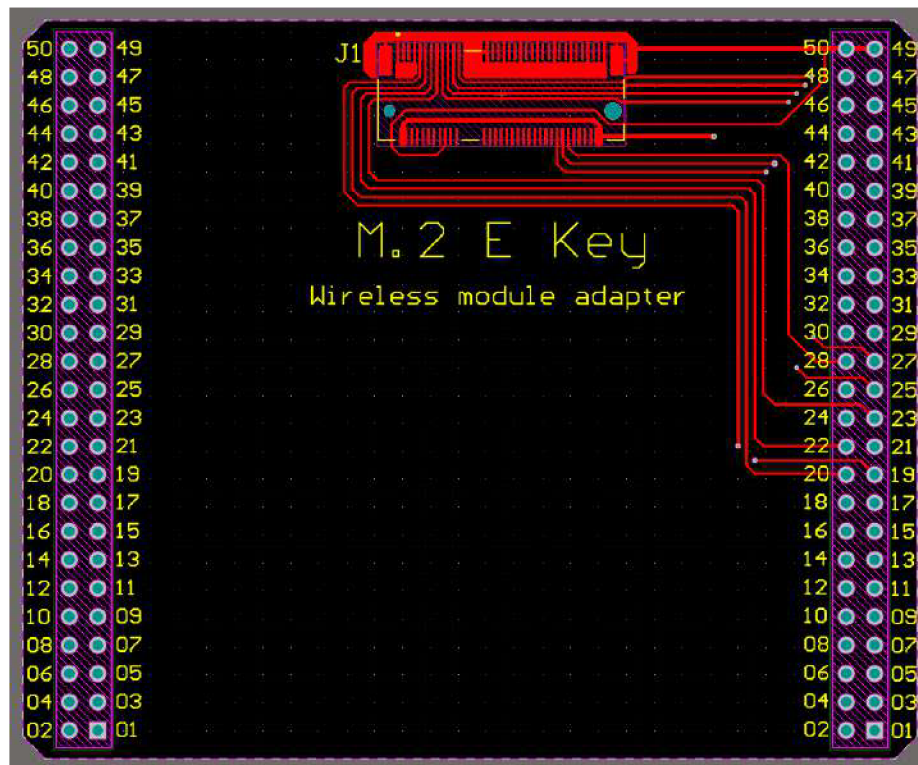




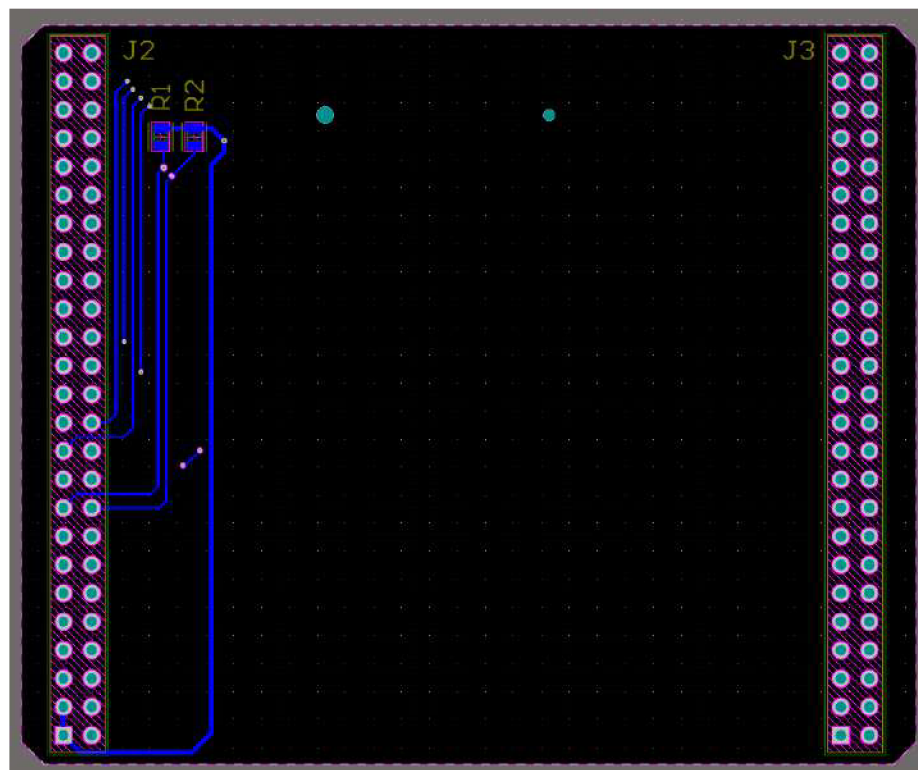
Obr. A.4: Schéma zapojenia modulu RFM95W



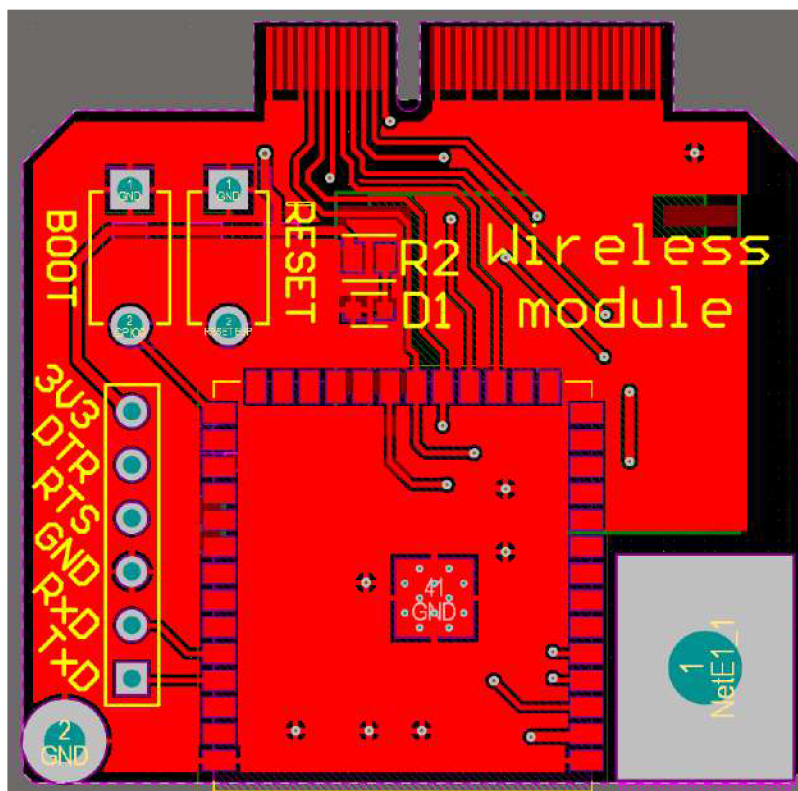
Obr. A.5: Schéma zapojenia výstupných konektorov



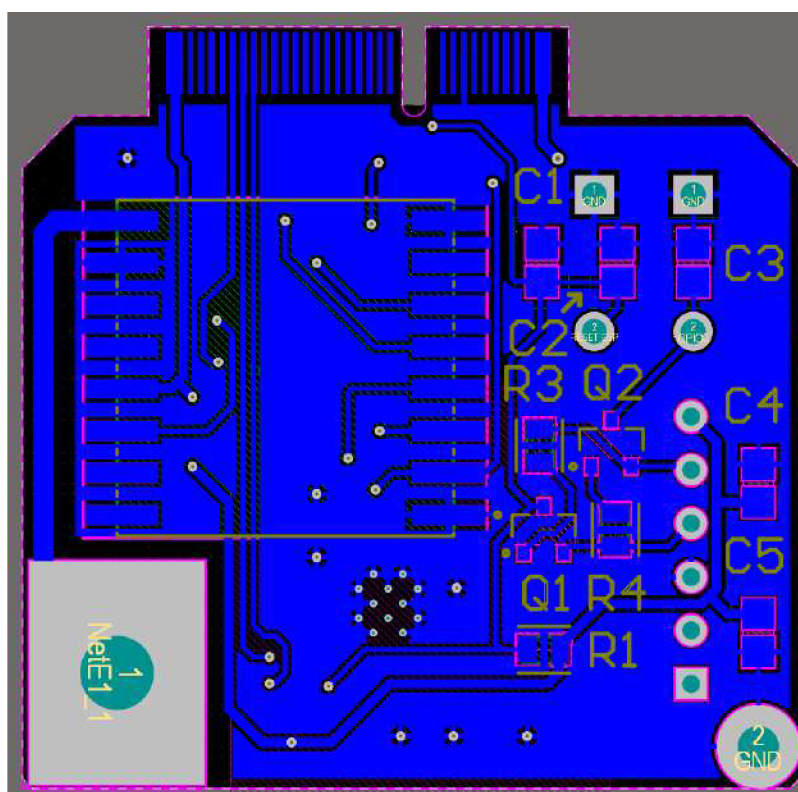
Obr. A.6: Návrh dosky plošného spoja pre adaptér – Pohľad z vrchu



Obr. A.7: Návrh dosky plošného spoja pre adaptér – Pohľad z doľa



Obr. A.8: Návrh dosky plošného spoja pre rozširujúci modul – Pohľad z vrchu



Obr. A.9: Návrh dosky plošného spoja pre rozširujúci modul – Pohľad z dola



Obr. A.10: Minerva s nasadeným rozširujúcim modulom