

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií



IMP - PROJEKT

FM Radio

2023/2024

Xpejch08 - Štěpán Pejchar

<b>Úvod</b>	<b>1</b>
Zadání projektu	1
<b>Videa ukázky projektu</b>	<b>1</b>
<b>Zapojení HW</b>	<b>2</b>
Fm přijímač RDA5807M	2
Zesilovač PAM8407	2
Rotační enkodér KY-040	2
Oled display	2
Finální zapojení	3
<b>Implementace SW</b>	<b>4</b>
Fm přijímač RDA5807M	4
Rotační enkodér	4
Oled Display	5
<b>Závěr</b>	<b>5</b>
<b>Bibliografie</b>	<b>5</b>

# Úvod

## Zadání projektu

Cílem projektu bylo vytvořit FM radio za pomoci volitelných knihoven a těchto modulů:

- [Oled display 0.96](#)
- [FM přijímač RDA5807M](#)
- [Audi zesilovač PAM8407](#)
- [Rotační enkodér KY-040](#)
- [vývojová deska ESP32](#)

Radio by mělo přijímat stanice v pásmu 76-108MHz. Uživatel by měl být schopen radio přeladit pomocí rotačního enkodéru. Radio by mělo komunikovat s uživatelem pomocí oled displaye, případně sdělovat RDS informace posílané stanicí.

## Videa ukázky projektu

Po komunikaci s Vámi přes email a domluvení se na situaci přidávám 3 videa. Video číslo 1 je video původní funkční verze radia z Brna. Video 2 je ukázka, na které se mi podařilo zachytit šumivý signál stanice Evropa 2 i po převozu do Poličky. Toho jsem docílil po dlouhém upravování antény, tudíž je pravděpodobné, že problém je opravdu se signálem, jak jsme zhodnotili v emailu. Video číslo 3 ukazuje funkčnost rotačního enkodéru při přepínání stanic nahoru, dolů a lazení pomocí funkce seek() při zmáčknutí. Dále ukazuje funkčnost oled displaye a komunikaci s userem.

1. <https://youtube.com/shorts/W-rYSb7FjQA?feature=share>
2. <https://youtube.com/shorts/DUeixb9-pkA?feature=share>
3. <https://youtu.be/4IAkx14MbNQ>

## Zapojení HW

### Fm přijímač RDA5807M

Fm přijímač má 8 připojených pinů. Nejprve musíme připojit přijímač k esp32 desce, abychom s ním mohli komunikovat pomocí SW, toho docílíme pomocí SDA(Serial data line), a SCK(serial clock line). SDA připojíme k SDA na ESP32 desce a SCK k SCL pinu. Součástka také potřebuje napětí, konkrétně 3.3V z pinu 3v3 a zem. Se zemí jsem měl problémy, je možné, že vznikali ground loopy, proto jsem ve výsledném zapojení připojil přijímač k jiné zemi než ostatní součástky. K přijímači je také připojena anténa (drát). Dále máme dva výstupní piny RCH a LCH, které jsou připojeny k zesilovači, přes ně je posílán odchycený fm signál do reproduktorů.

### Zesilovač PAM8407

Do zesilovače jsou připojeny reproduktory, ty mají specifický konektor na vrchu součástky. Dále je do zesilovače přiveden signál LCH na pin 8 a RCH na pin číslo 1. Každý z reproduktorů je také připojen na napětí, konkrétně 5V a to sice na pin 16 pro LCH a pin 9 pro RCH. Reproduktory jsou také uzemněny a to přes piny 13 a 12. Ve finální verzi mám zpojen pouze reproduktor LCH, přijde mi, že při zapojení obou reproduktorů se hlasitost rádia výrazně sníží, pravděpodobně, protože oba čerpají napětí ze stejného zdroje.

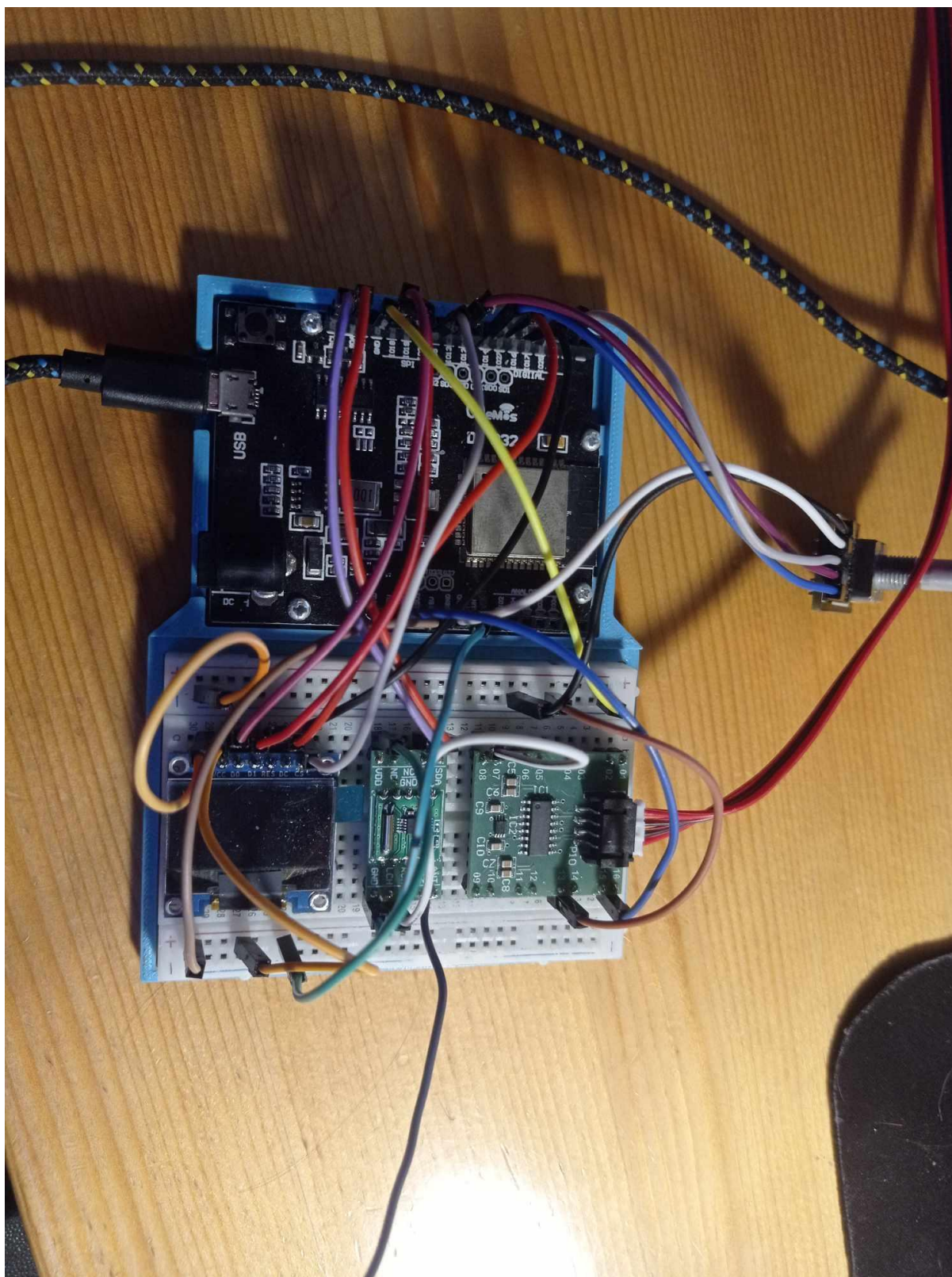
### Rotační enkodér KY-040

Rotační enkodér je připojen pomocí pěti pinů. Nejprve musíme rotační enkodér připojit do země, enkodér je připojen ke stejné zemi jako oled display a zesilovač. Dále musíme připojit napájení konkrétně 5V. Následující 3 piny enkodéru jsou použity pro vstup. Pin SW k pinu GPIO 25, pin DT k pinu GPIO 13 a pin CLK k pinu GPIO 14. Pin SW má za úkol obstarávat přerušení enkodéru při stlačení tlačítka, pin DT a CLK je použit pro určení směru rotace enkodéru.

### Oled display

Oled je připojen pomocí sedmi pinů. Nejprve musí být připojen k zemi pomocí pinu GND a k napájení na pinu VCC a to sice 3.3V. Ostatní piny jsou zapojeny podle návodu v zadání projektu [\[3\]](#)

## Finální zapojení



# Implementace SW

Pro implementaci byl využit nástroj platformio a framework arduino. Další specifické knihovny pro jednotlivé moduly jsou popsány níže u každého modulu. Projekt byl implementován v souboru main.cpp, s config filem platformio.ini. Vývojové prostředí Kubuntu 22.04. V souboru main.cpp nejprve inicializujeme všechny konstanty pro piny podle toho jak jsou připojené k desce ESP32, například CLK\_PIN = 14 značí, že CLK pin rotačního enkodéru je připojen na GPIO pin 14. Také definujeme všechny konstanty pro práci s displayem, jako jsou piny, na které je připojen a jeho velikost. Následně ve funkci setup inicializujeme všechny počáteční parametry pro práci s moduly. Nejdůležitější je funkce loop(), která je volána pořád dokola s delayem 0.01 sekundy. V ní implementuji veškerou logiku pro práci s moduly. Implementace také obsahuje funkci showFrequency(), která je volána ve funkci seek() pro debugging vypisování frekvence vyhledávané stanice.

## Fm přijímač RDA5807M

Pro implementaci projektu byla využita knihovna pu2clr/PU2CLR RDA5807@^1.1.9 [\[1\]](#). Nejprve musíme instanciovat radio pomocí třídy RDA5807. Následně radio setupujeme ve funkci setup(). Využívám funkce setRDS() pro nastavení možnosti rds informací od stanice, setAFC() [\[4\]](#) pro nastavení možnosti automatic frequency controll, setRdsFifo() pro nastavení fifo módu rds, setVolume() pro nastavení hlasitosti, implicitně na hodnotu 6, a setFrequency() pro nastavení implicitní frekvence pro přijímač - 99.3MHz. Dále pro práci s radiem voláme při určených přerušeních rotačního enkodéru (v sekci rotačního enkodéru) funkce pro manipulaci stanice. Při otočení rotačního enkodéru po směru, nebo proti směru, posouváme nastavenou frekvenci o +/- 10. Frekvence je v knihovně zadávána jako frekvence vynásobená hodnotou 100 (např. frekvence 103 je 10300). Přičítáním a odečítáním čísla 10 posouváme frekvenci o 0.1 MHz nahoru nebo dolů. Při zmáčknutí rotačního enkodéru je vyvoláno přerušení, které spustí funkci seek(). Ta vyhledá směrem dolů stanici, na kterou je možné se připojit, a následně se na ni naladí.

## Rotační enkodér

Rotační enkodér slouží pro přerušení použité k přelazování radia. Nejprve musíme inicializovat konstanty pro jeho HW zapojení a to sice:

- const int CLK\_PIN = 14; // CLK pin of KY-040 connected to GPIO 14
- const int DT\_PIN = 13; // DT pin of KY-040 connected to GPIO 13
- const int SW\_PIN = 25; // SW pin of KY-040 connected to GPIO 25

Následně všem pinům nastavíme input mode abychom mohli přijímat signál z enkodéru a povolíme jejich pullup rezistory. Pullup rezistory nastavujeme proto, aby měli piny implicitní high stav. Je to nastaveno proto, abychom mohli exaktně pozorovat přerušení, při zmáčknutí, nebo otočení enkodéru. Ve funkci loop kontrolujeme otočení nebo zmáčknutí enkodéru. Kontrolujeme, pomocí funkce digitalRead() na jakou hodnotu je nastavena vstupní signál enkodéru, pokud je signál CLK nastaven na HIGH a signál DT nastaven na LOW (otočení doprava) přičteme 0.1MHz ke stanici a radio je přelazeno, v opačném případě je enkodér otočen na druhou stranu, a 0.1MHz je odečteno. Ve funkci loop také oštrujeme stlačení enkodéru. Pokud bylo tlačítko enkodéru stlačeno, zavolá se funkce seek() a radio se přeladí na první stanici, kterou najde v pásmu směrem dolů.



## Oled Display

Oled display slouží ke komunikaci s uživatelem radia. Implementace oled displeje byla provedena za pomoci knihovny `adafruit/Adafruit_SSD1306@^2.5.9` [2]. V úvodní části souboru mapujeme všechny jeho připojené piny ke konstantám. Následně inicializujeme třídu `Adafruit_SSD1306`, která slouží pro sw instanci displeje. Display má pomocí konstant také nastavenou velikost, a to sice `SCREEN_WIDTH = 128` a `SCREEN_HEIGHT = 64`. Ve funkci `setup`, nastavujeme parametry displeje, zejména inicializujeme display pomocí funkce `begin()`. U každého volání funkce `loop()`, kontroluji, jestli přijímač přijal nějaké RDS informace. Pokud ano, jsou vypsány na display v pořadí: station name, station info, program info, rdsTime. Pokud RDS info není k dispozici display vypisuje pouze aktuální frekvenci na které chytá radio signál. Display také komunikuje s uživatelem při přelazování stanic. Pokud uživatel zmáčkne tlačítko enkodéru, display vypíše zprávu "Seeking!" aby bylo potvrzeno zmáčknutí tlačítka a po najetí a nalezení stanice display vypíše zprávu "Found!" a následně vypíše frekvenci konkrétní stanice. Při otáčení enkodéru display vypíše frekvenci naladěné stanice před vypsáním RDS informací. Frekvence je tudíž pro uživatele dostupná vždy.

## Závěr

Projekt fungoval tak jak má do chvíle převozu z Brna do Poličky. Převezen byl kvůli mému zdravotnímu stavu. Po emailové konzultaci s vámi jsme se shodli na tom, že by problém teoreticky mohl být v anténě a odchytávaném signálu, což se potvrdilo potom, co se mi podařilo zachytit velmi slabý signál stanice Evropa 2 jak je ukázáno na videu číslo 2. První video projektu, které není úplně ukázkové, dokazuje, že mé radio hrálo, a druhé video dokazuje problém se signálem v místě mého bydliště, a potvrzuje správné zapojení. Ve třetím videu představuji implementaci ostatních částí projektu, přesto že z reproduktorů vychází pouze šum. Finální verze radia ve výsledku splňuje všechny požadavky, a funkčnost, jako je přelazování stanic a komunikace s uživatelem s tím problémem, že nezvládá naladit stanici kvůli špatnému signálu.

## Bibliografie

[1] Caratti, Ricardo Lima. "pu2clr/RDA5807: Arduino Library for SINGLE-CHIP BROADCAST FM RADIO TUNER RDA5807." *GitHub*, <https://github.com/pu2clr/RDA5807>.

Accessed 13 December 2023.

[2] Fried, Limor. "adafruit/Adafruit\_SSD1306: Arduino library for SSD1306 monochrome 128x64 and 128x32 OLEDs." *GitHub*, [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306).

Accessed 14 December 2023.

[3] Mrázek, Vojtěch. 2 March 2023, <https://moodle.vut.cz/mod/page/view.php?id=338880>.

Accessed 14 December 2023.

[4] “What is Automated Frequency Coordination (AFC)?” *everything RF*, 5 November 2022, <https://www.everythingrf.com/community/what-is-automated-frequency-coordination-or-afc>. Accessed 14 December 2023.