

RadioESP32

Úvod

Hned na úvod se sluší uvést, že tento projekt by asi nikdy nevznikl bez populárního projektu "ESP32Radio-V2" od Edzelf. Autorovi jsem za něj velice vděčný a děkuji mu za jeho práci. Původně jsem samozřejmě převzal originální projekt v nezměněné podobě, ale velice brzy mi to přestalo vyhovovat. Zejména mi vadilo, že displej nezobrazoval znaky s diakritikou.

Druhým největším problémem pro mne bylo fungování přehrávače souborů z SD karty. Jedním z mých požadavků na přístroj bylo, aby umožňoval poslech audioknih. V takovém případě je ale třeba, aby soubory byly přehrávány v definovaném (tedy abecedním) pořadí. Ale to s originálním SW nefungovalo. Po překopírování souborů z HDD na SD kartu se totiž soubory vždy seřadily v jakémsi náhodném pořadí a poslouchat jednotlivé díly ve správném pořadí tak bylo velmi komplikované (správný díl bylo nutné volit ručně).

Kromě uvedených dvou problémů tu pak byla ještě celá řada různých maličkostí, jejichž fungování jsem si představoval jinak. Jako příklad mohu uvést funkci, která zajišťovala, že po zapnutí se vždy přehrávala naposledy poslouchaná stanice (a navíc s naposledy použitou úrovní hlasitosti). To mně ale vůbec nevyhovovalo, já preferuji aby se po zapnutí přístroje přehrávala nějaká defaultní stanice (defaultní hlasitostí). Dále jsem plánoval přidat nějaké nové funkce, které originální konstrukce neměla. Zejména jsem chtěl, aby rádio bylo možné zapnout a vypnout stiskem jednoho (stejného) tlačítka a aby rádio mělo funkci usínání (tedy automatického vypnutí po nastaveném čase).

I přes uvedené skutečnosti jsem z originálního projektu "ESP32Radio-V2" převzal několik věcí téměř beze změny a moje práce tak byla usnadněna.

Design webových stránek, způsob komunikace mezi ESP32 a počítačem (prostřednictvím webového prohlížeče) a několika dalších řešení (například způsob provádění upgrade nebo práce s konfiguračním souborem) jsou převzaty z projektu **esp-rfid**. Použil jsem to již v několika projektech a to přesně vyhovuje mým potřebám a představám.

Vývoj s VS Code a PlatformIO

Jako vývojové prostředí používám VS Code/PlatformIO. Pomocí souboru `platformio.ini` je možné vytvořit několik variant software a to uvedením anebo neuvedením takzvaného „build-flagu“.

Důležité jsou zejména build-flagy `DATAWEB`, `SDCARD`, `BATTERY`, `AUTOSHUTDOWN` a `OTA`.

- S použitím build-flagu `DATAWEB` se zkompiluje SW ve verzi, která má soubory webového serveru uložené v oblasti souborového systému „LittleFS“, zatímco bez parametru `DATAWEB` je SW zkompilován ve verzi, ve které soubory webového serveru jsou uloženy v programové paměti (pomocí atributu `PROGMEM`) ve formě proměnných. První verzi říkám zkráceně „`data`“, druhé „`webh`“. Varianta `data` je určena především pro fázi vývoje (ale nic nebrání používat to i jako produkční), zatímco varianta `webh` je zamýšlena výhradně jako produkční a provádět s ní vývoj je víceméně nemožné.

- Použitím build-flagu `SDCARD` se zapíná podpora funkce přehrávače souborů z SD karty. Bez použití uvedeného build-flagu je přístroj schopen přehrávat pouze rozhlasové stanice.

- Použitím build-flagu `BATTERY` se zapíná podpora funkce měření napájecího napětí. Je to samozřejmě vhodné v případě napájení z baterií. Displej potom zobrazuje úroveň nabití.

- Použitím build-flagu `AUTOSHUTDOWN` se zapíná podpora funkce, umožňující zapínání a vypínání přístroje jedním tlačítkem a také funkce sleep (nastavitelná doba automatického vypnutí). Podmínkou je samozřejmě vybavení HW potřebným obvodem.

- Použitím build-flagu `OTA` se zapíná funkce, umožňující komfortní upgrade z prostředí webového rozhraní. Viz kapitola Upgrade.

V souboru `platformio.ini` jsou také připraveny 4 druhy „envs“. Pomocí nich lze vybrat jednak variantu vývojového kitu (4 nebo 8 MB) a jednak variantu SW s displejem anebo bez displeje. Celkem tedy 4 kombinace (`noota4mb`, `wota8mb`, `noota4mbnod` a `wota8mbnod`). Příslušná verze „env“ se volí zakomentováním/odkomentováním v části `[platformio]`.

Za použití build-flagu DATAWEB postupují při vývoji SW takto:

- 1) Vývoj probíhá výhradně ve variantě software **data** (v souboru `platformio.ini` je použit build-flag DATAWEB). Po úpravě zdrojových souborů se obvyklým způsobem provede kompilace a přístroj se upgraduje přes USB (sériový) port.
- 2) Pokud je pro opravu nebo přidání nějaké funkce nutné upravit i soubory *.html nebo *.js webového serveru, je třeba otevřít stránku <http://<IP adresa přístroje>/edit>. Aby to fungovalo, PC musí být připojen na internet. K editování je totiž využívána online knihovna „ace“ (Ajax.org Cloud9 Editor). Více viz <https://ace.c9.io/>.
- 3) Jakmile se pomocí kroků 1 a 2 dosáhne požadované funkcionality, lze vytvořit **webh** variantu software. To se provede spuštěním uživatelského skriptu **Download FS & Create WEBH**, který je dostupný prostřednictvím nabídky **PROJECT TASKS/<vybraná verze env>/Custom**. Tím dojde k transformaci souborů webového serveru (ze souborového systému LittleFS) do souborů **webh**. Ke stažení obrazu souborového systému a jeho extrakci je použita utilita „pio-esp32-esp8266-filessystem-downloader“ od **maxgerhardt**.
- 4) Přepnutí na variantu **webh** (v souboru `platformio.ini` je třeba odstranit [zakomentovat] build-flag DATAWEB) a provedení kompilace. Výsledný obraz „*firmware_WEBH_XXX.bin*“ (vytvořený během procesu kompilace) lze nalézt v adresáři `/bin/<vybraná verze env>`. Označení verze „XXX“ se nastavuje před kompilací v souboru `platformio.ini` rovněž jako „build-flag“.

Upgrade

Přístroj, ve kterém běží **webh** varianta firmware, může být upgradován velmi snadno prostřednictvím webového rozhraní. Tato funkce je bohužel možná pouze pro vývojové kity, které mají alespoň 8MB paměti flash. Postup:

- a) V nabídce (na levé straně obrazovky) vybereme položku **Update**.
- b) Objeví se okno, které v dolní části zobrazuje aktuální verzi.
- c) Pomocí tlačítka „Vybrat soubor“ lze v počítači vyhledat soubor s firmwarem (např. *firmware_WEBH_100.bin*) a tlačítkem **Update** proces spustit.

Pokud vše proběhne v pořádku, po restartu již přístroj obsahuje novou verzi software.
POZOR! V kitech s pamětí 4MB je možné SW upgradovat pouze přes USB/sériový kabel!

První spuštění

Jakmile je přístroj poprvé spuštěn, běží ve výchozím nastavení a v režimu AP. Na PC se připojíme k síti „RadioESP32“ a v prohlížeči na adresu 192.168.4.1. Měla by se objevit webová stránka prostřednictvím které je třeba provést nastavení přístroje.

Hlavní nabídka

V levé části obrazovky je vždy hlavní nabídka. Popis položek hlavní nabídky:

Radio Player

Po kliknutí na tuto položku se objeví stránka, pomocí které se dá komfortně ovládat rádio.

SD Card Player

Tato položka se zobrazuje jen v případě, že SW je zkompilován s možností SDCARD.
Po kliknutí na tuto položku se objeví stránka, pomocí které se dají komfortně ovládat funkce přehrávače souborů z SD karty.

Status

Po kliknutí na tuto položku se objeví stránka se základními údaji o přístroji.

Settings

Je-li třeba upravit nastavení přístroje (například při prvním spuštění), vybereme položku **Settings**. Po kliknutí na ni se nabídka rozbalí a uvidíme osm dílčích nabídek:

Radio Settings

Na této stránce se zejména vytváří nabídka rozhlasových stanic. Dále se zde dá nastavit výchozí stanice (ta se vždy automaticky vybere při zapnutí přístroje), výchozí hlasitost a hodnoty jednoduchého ekvalizéru.

SD Card Settings

Tato položka se zobrazuje jen v případě, že SW je zkompilován s možností SDCARD. Na této stránce se dá nastavit pouze jeden parametr a tím je "Seek step". To je délka skoku (vpřed i vzad) ve vteřinách. Nastavení výchozí hlasitosti a ekvalizéru, které je na stránce **Radio Settings**, platí i pro přehrávač souborů z SD karty.

Network Settings

Na této stránce je třeba nastavit zejména síť (případně více sítí), na kterou se má přístroj připojit. Stiskneme **Edit Networks** a následně **New Network**. V dialogovém okně, které se objeví nastavíme požadované hodnoty. Lze použít tlačítko **Scan** k vyhledání dostupných sítí. Po nastavení stiskneme **Save Changes**. Pod tabulkou sítí je možné ještě změnit SSID, IP adresu a masku (vše v AP režimu). Změny je třeba potvrdit stisknutím tlačítka **Save** a posléze také žlutého tlačítka s textem **You have uncommitted changes, please click here to commit and reboot ...**. Toto žluté tlačítko je však vhodné použít teprve až po projití všech ostatních nastavení, protože po jeho použití se přístroj restartuje.

Hardware Settings

Na této stránce je možné nastavit některé použité GPIO piny (některé jsou nastaveny před kompilací v **platformio.ini**) a několik dalších parametrů. Nastavené hodnoty je opět třeba potvrdit stisknutím tlačítka **Save**.

IR Remote Settings

Na této stránce se konfiguruje povely dálkového ovládání. Přidání nového povelu je velmi jednoduché a intuitivní. Nastavené povely je třeba opět třeba potvrdit stisknutím tlačítka **Save**.

Dialog box titled "Edit Command # Hyundai/Channel+" with a close button (X) in the top right corner.

Description: Hyundai/Channel+

The "IR Code" field is filled in automatically as soon as you press the desired button on the remote control. It is recommended to press the button several times and check whether the code detection is reliable.

Remote controls with unreliable code detection are better not to use !

IR Code: F906BF00

Command: Channel+ (dropdown menu)

Buttons: **Save changes** (blue), **Cancel** (grey)

Display Settings

Na této stránce je možné nastavit zejména úroveň podsvícení displeje a nastavení zobrazení hodin v klidu. Uživatel si také může pojmenovat dny v týdnu ve svém jazyce (ve výchozím stavu jsou jména anglická). Jako vždy se po všech změnách musí stisknout tlačítko **Save**.

General Settings

Na této stránce je možné nastavit "Jméno stroje" (Host Name) a také se zde kalibruje zobrazování úrovně nabití baterie. Potvrzuje se opět stisknutím tlačítka **Save**.

NTP (Time) Settings

Na této stránce je možné nastavit NTP server, interval synchronizace (v minutách) a časovou zónu. Změny se opět potvrzují stisknutím tlačítka **Save**.

Někdy se může stát, že v době zapnutí (nebo restartu) přístroje není dostupná žádná ze známých WiFi sítí. Potom samozřejmě není možné synchronizovat čas s NTP serverem. V takové krajní situaci pomůže tlačítko [Sync Browser Time to Device](#).

Backup & Restore

V hlavní nabídce dále nalezneme položku „Backup & Restore“. Význam je zřejmý z názvu. Je možné uložit si do PC soubor se zálohou nastavení a v případě potřeby se ke stejnému nastavení snadno vrátit. Pod uvedenou položkou je umístěna ještě další funkce – reboot přístroje.

Factory reset

Pod touto položkou hlavní nabídky se skrývá stejnojmenná funkce. Po její aktivaci se vymaže konfigurační soubor a po rebootu je přístroj ve stejném stavu jako při prvním spuštění.

Update

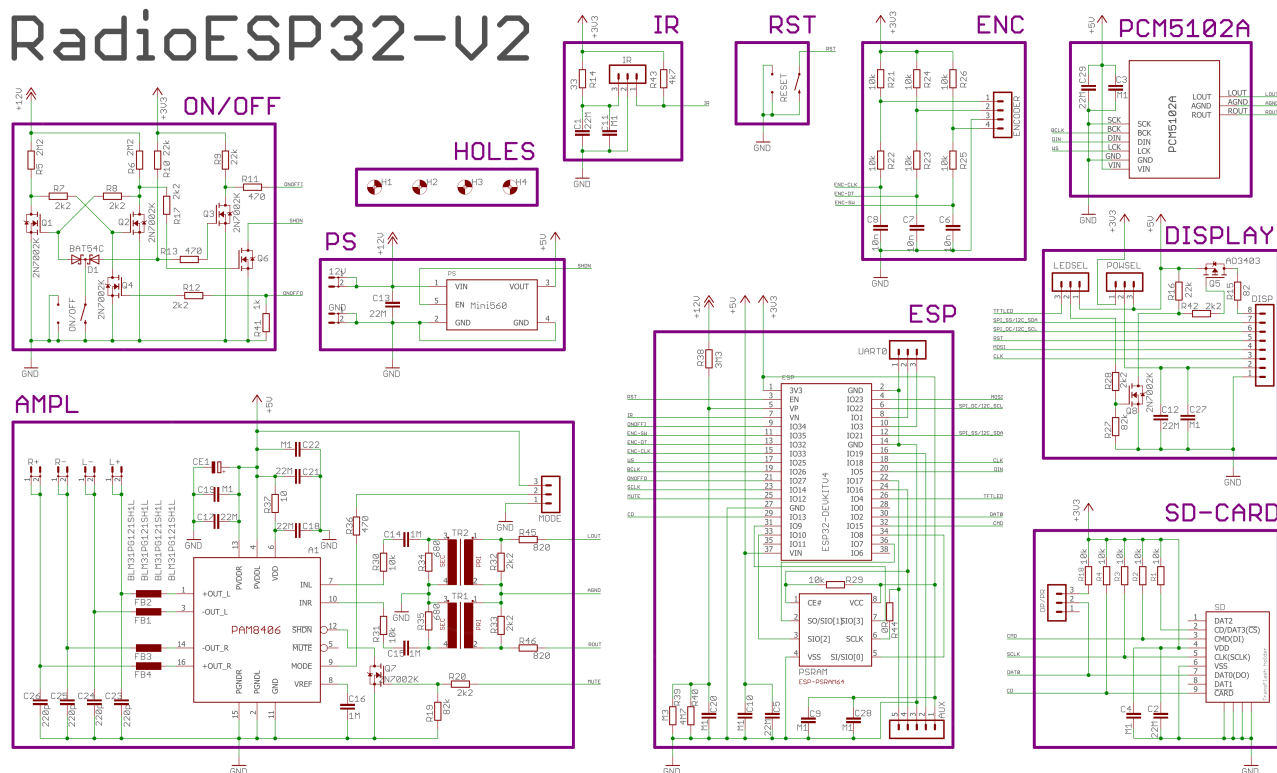
Tato položka se zobrazuje jen v případě, že SW byl zkompileován s build-flagem OTA, tedy v případě „envs“ wota8mb a wota8mbnod. Postup je popsán výše v odstavci Upgrade.

Konstrukce přístroje

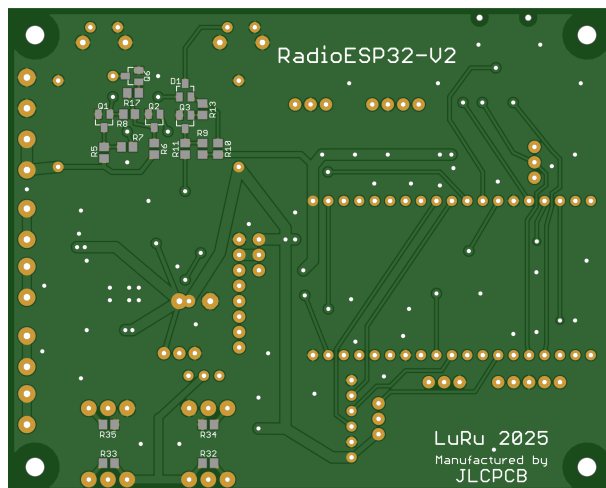
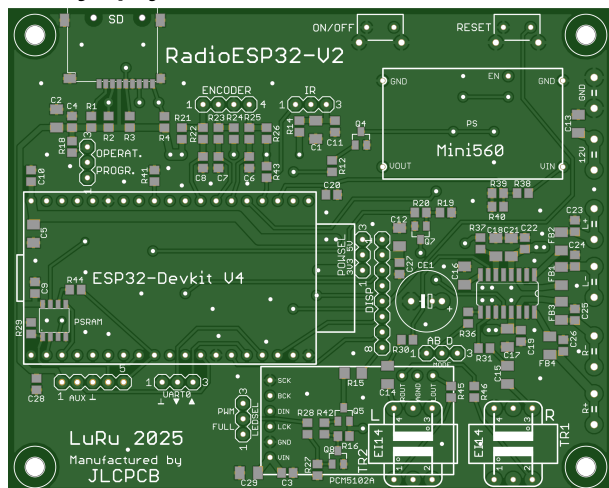
Schéma zapojení a plošný spoj byly vytvořeny pomocí návrhového systému Eagle.

Schéma zapojení

RadioESP32-V2

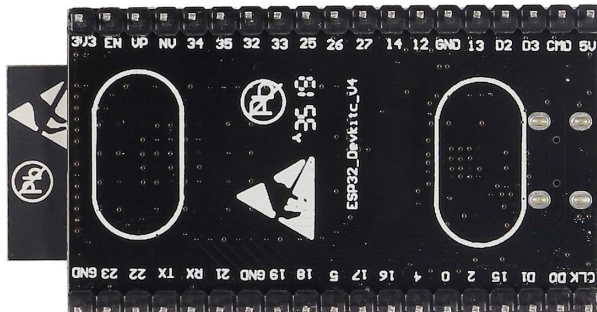
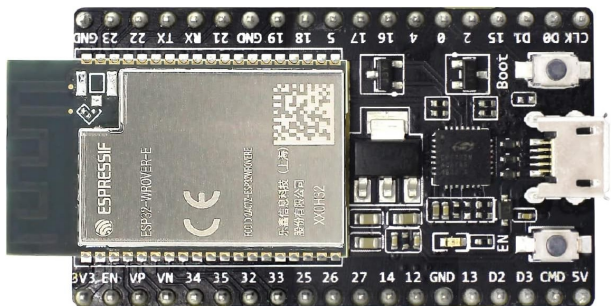


Plošný spoj

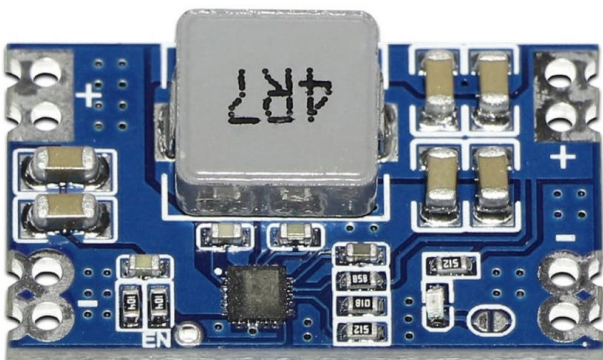


Volba součástek

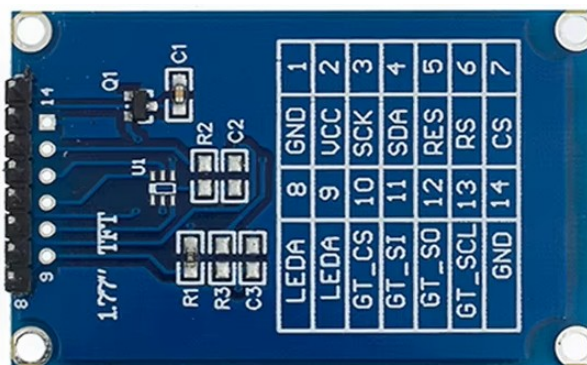
V první fázi vývoje jsem po vzoru Edzelf používal (jako DAC) modul VS1053. To sice fungovalo dobře, ale toto řešení mi připadalo zbytečně drahé a také dosti problematické (například jednou jsem omylem koupil VS1003 místo toho a to nefungovalo dobře). Proto jsem tuto cestu úplně opustil a přešel výhradně na řešení s I2S. Za základ SW jsem proto zvolil výbornou knihovnu "ESP32-audioI2S" od autora **shreibfaul1**. Z toho vyplynula potřeba použít takový typ vývojového kitu ESP32, který je vybaven pamětí PSRAM. Bez této paměti se totiž s uvedenou knihovnou nedá vytvořit žádný komplexnější projekt, brzy se narazí na nedostatek paměti RAM. Z toho důvodu jsem zvolil vývojový kit typu WROVER. Také je třeba hned na začátku upozornit, že revize čipu musí být minimálně "Rev 3". Zjistil jsem, že starší revize nefungují dobře. Dále doporučuji použít verzi modulu s pamětí 8MB. Běžné verze s pamětí 4MB lze sice použít také, ale ztrácí se tím možnost pohodlného upgrade z webového prostředí. Podrobněji to již bylo vysvětleno v předchozím textu. Poznámka: Na plošném spoji je počítáno i s možností použít externí paměť PSRAM, ta však se v případě použití kitu typu WROVER neosazuje, je přítomna již na uvedeném kitu.



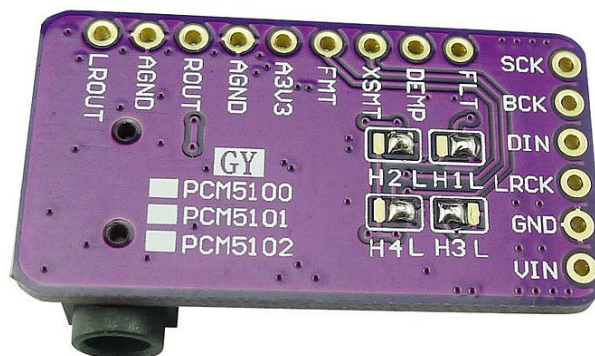
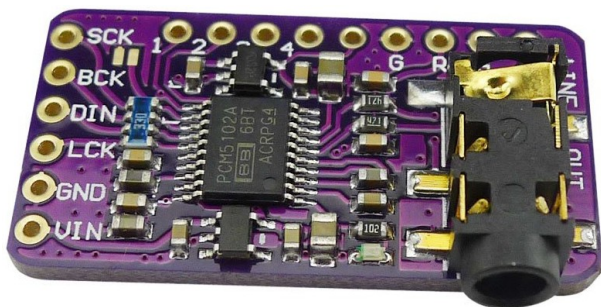
Rozhodl jsem se, že základní napájecí napětí přístroje bude 12V, protože může být velmi dobře realizováno třemi Lion články 18650. Vývojový kit ESP32 však potřebuje napětí 5V (stejně jako některé další komponenty) a proto je třeba použít vhodný step-down modul. Zvolil jsem modul Mini560, který je dostatečně dimenzovaný (napěťově i proudově) a je vybaven (pro mé potřeby nezbytným) vstupem EN.



Původně jsem (stejně jako Edzelf) zachovával možnost při kompilaci SW vybrat si mezi displejem OLED a displejem TFT. Nakonec jsem od toho ale ustoupil a OLED displeje již nejsou vůbec podporovány. Možnosti volby jsou tedy jen dvě - buď displej typu TFT anebo žádný. Já jsem použil displej s úhlopříčkou 1,77 palce a rozlišením 160x128. Neměl by však být problém použít jakýkoliv jiný, podporovaný knihovnou TFT_eSPI.



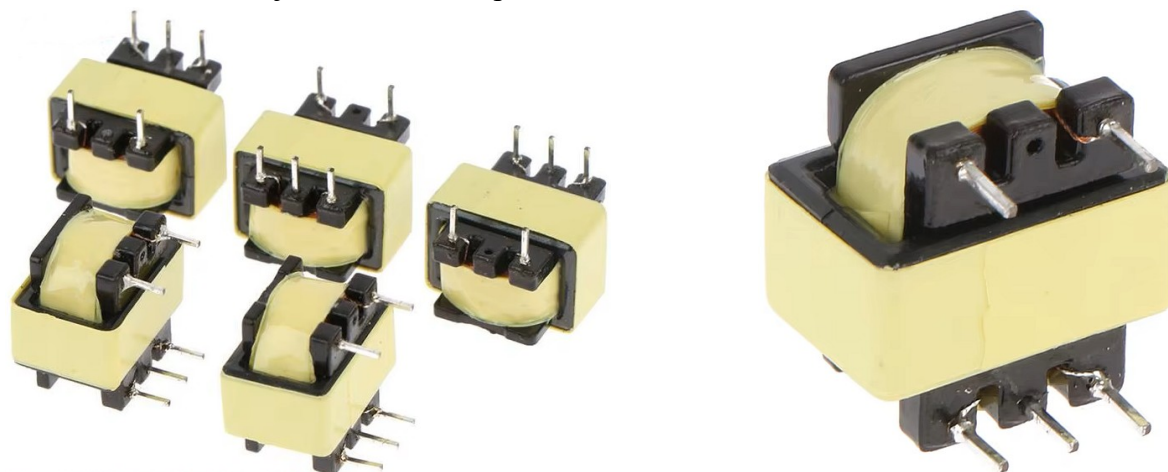
Podstatnou součástí přístroje je digitálně-analogový převodník. Zvolil jsem modul s čipem PCM5102A, který mi vyhovoval svými rozměry a také tím, že audio výstup je vyveden nejen na konektor typu jack, ale také na pinovou lištu. Tím se totiž velmi zjednodušuje propojení převodníku se základní deskou.



Jako zesilovač jsem vybral typ PAM8406, který je dostatečně výkonný, stačí mu napájecí napětí 5V a dá se dobře ručně pájet. Tento zesilovač má ještě jednu výhodu - může pracovat buď ve třídě D nebo AB. Já jsem zachoval možnost volby i na desce plošných spojů - režim se vybírá umístěním příslušného jumperu.



Mezi DAC a zesilovač je nutné zařadit oddělovací transformátory, aby se přerušila zemní smyčka a zabránilo se tak nepříjemnému rušení. Vybral jsem si transformátory, které mají výšku pouze 9.1mm a dobře se tak vejdou mezi desku převodníku a základní desku.



Důležitou součástí přístroje je rovněž rotační kodér. Použil jsem zcela běžný a levný typ EC11. Samozřejmým požadavkem na funkci přístroje byla také možnost ovládání pomocí infračerveného dálkového ovládání. Jako přijímač jsem vyzkoušel typ VS1838B a osvědčil se. Nakonec bylo třeba vyřešit ještě způsob připojení napájecího napětí a reproduktorů na základní desku. Chtěl jsem se vyhnout svorkovnicím se šroubky a proto jsem zvolil konektory typu faston šířky 4,8mm.

