

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Internetové rádio

Lukáš Jureček



Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2020/2021

Poděkování

Chtěl bych poděkovat všem, kteří mi pomohli při výrobě a vývoji mého projektu, zvláště pak panu učiteli Ing. Petru Grussmanovi za rady, jak vyřešit některé problémy, mému kamarádovi Filipu Duškovi za poznámky při programování softwaru, a nakonec mému otci za pomoc při výrobě obalu pro můj projekt.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2020

podpis autora práce

ANOTACE

Projekt je návrh a realizace internetového rádia s integrovaným zdrojem energie schopným dobíjení prostřednictvím mikro USB portu s displejem zobrazující aktuální stanici a regulaci hlasitosti. Navíc je zde možnost ovládání přes HTML stránku s tlačítky pro změnu stanice a posuvníkem pro úpravu hlasitosti. Internetové rádio je zařízení, které prostřednictvím Wi-Fi/LAN připojení, umožňuje poslech rozhlasových stanic šířených v digitální formě přes internet.

OBSAH

Poděkování.....	2
ÚVOD.....	5
1 VYUŽITÉ TECHNOLOGIE	6
1.1 HARDWARE	6
1.1.1 Procesor.....	6
1.1.2 DAC audio převodník	6
1.1.3 LCD 1602 s I2C převodníkem	6
1.1.4 Zesilovač a reproduktory.....	7
1.1.5 Nabíjecí modul a napájení.....	7
1.2 SOFTWARE	7
1.2.1 Arduino IDE ver. 1.8.14.....	7
1.2.2 Knihovny.....	7
1.2.3 Nástroje pro tvorbu webu.....	8
1.2.4 KiCad ver. 5.1.7	8
2 ŘEŠENÍ HARDWAROVÉ ČÁSTI.....	9
2.1 SEZNAM POUŽITÝCH SOUČÁSTEK	9
2.2 SCHÉMATICKÉ ZAPOJENÍ	10
2.3 DAC MODUL	11
2.4 NAPÁJENÍ	12
3 ZPŮSOBY ŘEŠENÍ SOFTWARE A POUŽITÉ POSTUPY.....	13
3.1 ZÁKLADNÍ RÁDIO	13
3.2 ZMĚNA HLASITOSTI	14
3.3 ZMĚNA STANICE	14
3.4 DISPLEJ	15
3.5 WEB SERVER	16
ZÁVĚR	18
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ.....	19
SEZNAM PŘÍLOH.....	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.

ÚVOD

Cílem této práce bylo vytvořit funkční Internetové rádio, které by se dalo jednoduše použít v domácnosti. Dalšími prioritami byla jednoduchá ovladatelnost, tj. změna hlasitosti, změna stanice, ovládání přes telefon atd., nižší cena oproti trhu a pěkný lehce přenosný obal.

Ze začátku bylo hlavním cílem vytvořit Bluetooth reproduktor s vlastním zdrojem energie, který by se dal lehce nabíjet, a šel by použít kdekoliv a kdykoliv, ale po zjištění nekompatibility nové Bluetooth verze, které používá ESP32 a starší verze na mém telefonu jsem přešel na Internetové rádio, které hardwarově funguje podobně, liší se pouze v softwaru.

V této dokumentaci se budu podrobně věnovat jednotlivým částem které jsem řešil, jako bylo vymyšlení schematického zapojení všech součástek, zprovoznění DAC audio převodníku, napsání programu na změnu stanic a hlasitosti, zprovoznění web serveru pro ovládání z jiného zařízení, řešení napájení aj.

1 VYUŽITÉ TECHNOLOGIE

1.1 Hardware

1.1.1 Procesor

Celý chod programu je řízen pomocí 32bitového čipu **ESP32** od firmy Espressif, který má implementovaný Wi-Fi modul, a také Bluetooth ve verzi 4.2 s podporou BLE (Bluetooth Low Energy), který ale nespolupracuje se staršími zařízeními používající starší verzi Bluetooth.

Parametry:

- Napájení 3.3 V nebo 5 V
- 36 pinů
- Flash paměť 16 MB
- SRAM 512 KB
- Frekvence 160 MHz

1.1.2 DAC audio převodník

Použil jsem stereo DAC převodník od firmy Adafruit s názvem **UDA1334A** tento převodník využívá I2S rozhraní využívající pro přenos zvuku 3 sběrnice (Bit Clock, Left Right Clock, DIN), zařízení lze napájet od 3.3 V do 5 V. Z tohoto převodníku lze získat zvuk buď pomocí JACK výstupu nebo pomocí pinů.

1.1.3 LCD 1602 s I2C převodníkem

Jako displej jsem si vybral **LCD 1602**. Jedná se o displej, který má 16 znaků ve 2 řádcích. Na napájení potřebuje 5 V, má nastavitelné podsvícení trimrem, které se dá i vypnout vyjmutím jumperu. K výpisu na displej používám I2C sběrnici, díky které mi na ovládání displeje stačí pouze 2 piny SDA (datový kanál) a SCL (hodinový signál) a 2 piny na napájení. I2C sběrnice byla vyvi-

nuta firmou Philips a umožňuje propojení až 128 různých zařízení pomocí 2 obousměrných vodičů.

1.1.4 Zesilovač a reproduktory

Mám dva 3 W reproduktory s impedancí $4\ \Omega$ připojené na stereo zesilovač **PAM8403**. Výhodou tohoto zesilovače je že dokáže být napájen od 2,5 V do 5 V.

1.1.5 Nabíjecí modul a napájení

Pro zdroj energie jsem si vybral nabíjecí Li-Ion 3,7 V/2900 mAh **baterii 18650**, baterie je pak připojena na nabíjecí modul **TC4056**. Tento modul jsem si vybral, protože pro nabíjení baterie jej stačí připojit přes mikro USB port s klasickou 5 V nabíječkou. Z tohoto modulu pak přes vypínač jde proud do Step-up měniče **ME2108** který dává 5 V pro displej a z něj do Step-down měniče **LM2596** který se dá pomocí trimru regulovat od 1,5 V do 34 V.

1.2 Software

1.2.1 Arduino IDE ver. 1.8.14

Je to multiplatformní aplikace pro psaní programů nejčastěji v jazycích C a C++. Používá se k psaní a nahrávání programů nejen pro desky od firmy Arduino ale po stáhnutí pluginů i pro ostatní typy čipů. Používal jsem jej jelikož přes VSCode s rozšířením Platformio jsem měl problém nahrávání kódu (často jsem dostával upload error) a zároveň moc nefungovalo přidávání knihoven.

1.2.2 Knihovny

Jsou to složky se soubory ve kterých je napsán nějaký zdrojový kód, tudíž nám umožňují mít přehlednější a jednodušší vlastní program.

Ve svém kódu jsem použil tyto knihovny :

- WiFi.h - Knihovna pro připojení k internetu
- LiquidCrystal_I2C - Knihovna k I2C sběrnici kterou ovládám displej.
- SPIFFS.h - Knihovna pro nahrání HTML souboru apod. do flash paměti ESP32.
- AsyncTCP.h + ESPAsyncWebServer.h - Pro vytvoření web serveru na ESP32.

1.2.3 Nástroje pro tvorbu webu

HTML je značkovací jazyk pro tvorbu webových stránek. **CSS** je zkratka pro kaskádové styly. Popis způsobu zobrazení elementů napsaných v jazycích HTML, XHTML nebo XML. **JavaScript** je multiplatformní objektově orientovaný skriptovací jazyk.

1.2.4 KiCad ver. 5.1.7

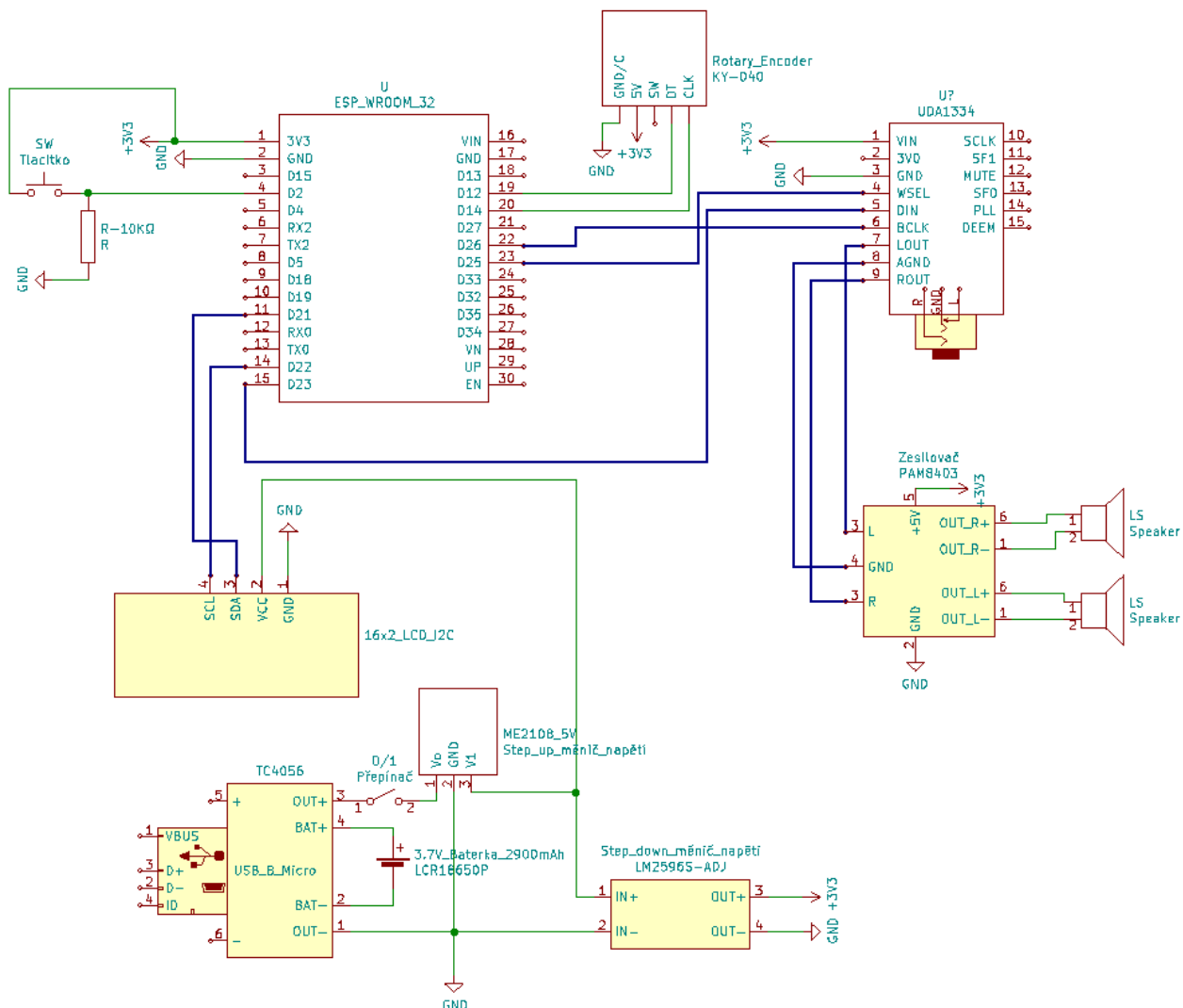
Svobodný software pro projektování plošných spojů, tedy pro jednu z úloh při automatizaci projektování elektroniky, podporovaného projektování zaměřeného na projektování elektronických výrobků a jejich součástí.

2 ŘEŠENÍ HARDWAROVÉ ČÁSTI

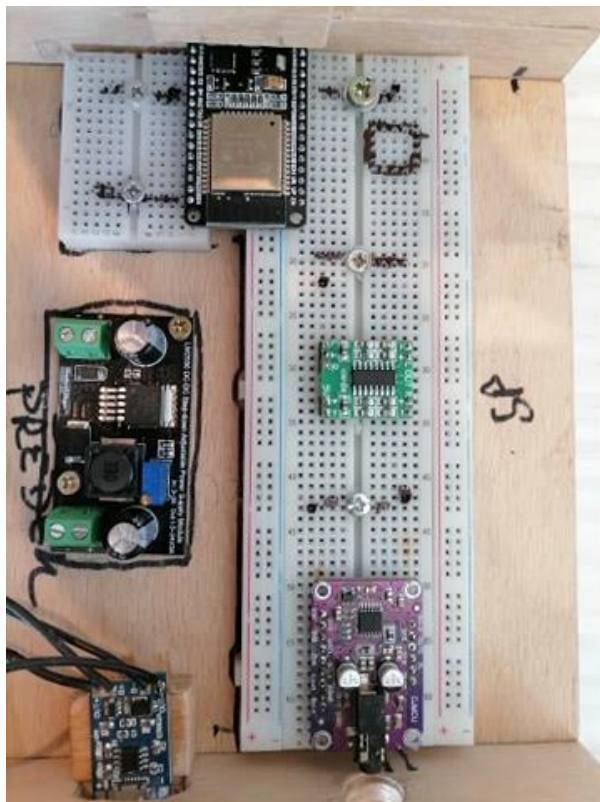
2.1 Seznam použitých součástek

- ESP32
- DAC UDA1334
- Zesilovač PAM8403
- LCD 1602 + I2C sběrnice
- KY - 040 enkodér
- Nabíjecí modul TC4056
- Step-down měnič LM2596
- Step-up měnič ME2108
- Baterie LCR 18650
- Držák pro baterii
- Dva 3 W reproduktory
- Tlačítko
- Přepínač
- Odpor 10 K Ω

2.2 Schématické zapojení



Obr1. schéma zapojení



Obr2. ukázka vnitřku rádia

2.3 DAC modul

Důležitým prvkem rádia je převodník na převod digitálního signálu na analogový. Jako první DAC jsem používal **PCM5102** z důvodu velkého počtu tutoriálů a lehkého zapojení. S tímto DAC modulem jsem měl samé problémy a nikdy se mi nepodařilo dostat fungující produkt. Poté co jsem ho vyzkratoval (viz. 2.4 Napájení), jsem si koupil další. Výsledek byl opět stejný. Tak jsem přešel k převodníku **UDA1334a** od firmy Adafruit. Tento DAC modul měl výborné recenze a skvělou a přehlednou technickou dokumentaci, díky které jsem jej lehce implementoval do svého projektu. Po zapojení a otestování vše najednou jelo tak jak má.

2.4 Napájení

Jelikož jsem pro napájení všech komponentů potřeboval dvě různá napětí, osadil jsem do rádia dva konvertory napětí. Objednal jsem dva konvertory, jeden pro usměrnění na 3,3 V a druhý na 5 V. Ovšem jeden z konvertorů byl špatný. Když jsem ho připojil ke zdroji energie, byl ve zkratu. Na tuto závadu jsem přišel pozdě a spálilo mi to větší polovinu součástek, včetně druhého konvertoru. Zároveň baterie nedokázala dávat potřebné napětí, takže jsem zvolil sériové zapojení s novými konvertory, s tím že druhý v pořadí má trimmer na úpravu výchozího napětí. Jenže toto zapojení znovu nebylo to, co jsem potřeboval a po spoustě hodinách zkoušení a měření jsem došel k závěru, že ona baterie nezvládne napájet ESP32, ani když je zapojená pouze k tomuto mikrokontroleru.

3 ZPŮSOBY ŘEŠENÍ SOFTWARE A POUŽITÉ POSTUPY

3.1 Základní rádio

Pro zajištění bezproblémové funkce rádia, jsem zvolil poměrně jednoduchý kód. Díky němu se jednotka ESP32 připojila k síti Wi-Fi, poté se připojila pomocí URL adresy na určený stream. Jednotka ESP32 začala posílat data na DAC převodník UDA1334 a na sluchátkovém výstupu jsem prostřednictvím aktivních reprosoustav ověřil funkčnost rádia. Všechny proměnné (SSID, heslo, hlasitost, ...) zde jsou řešeny pevným způsobem a nelze je změnit jinak než přepsáním kódu.

```
#include „Arduino.h“
#include „WiFi.h“
#include „Audio.h“

#define I2S_DOUT      22  // DIN connection
#define I2S_BCLK      26  // Bit clock
#define I2S_LRC       25  // Left Right Clock

Audio audio;

String ssid =      „SSID“;
String password = „Heslo“;

void setup() {
    Serial.begin(115200);
    WiFi.disconnect();
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid.c_str(), password.c_str());
    while (WiFi.status() != WL_CONNECTED) delay(1500);
    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(15); // 0...21
    audio.connecttohost(„http://www.wdr.de/wdrlive/media/einslive.m3u“);
}

void loop(){
    audio.loop();
}
```

3.2 Změna hlasitosti

První úpravou, kterou jsem implementoval byla změna hlasitosti. Našel jsem si, jak funguje můj rotační enkodér KY-040 a zjistil jsem že celkem jednoduše. V nekonečné smyčce `loop` se nachází jako první načtení stavu pinu CLK a poté jeho kontrola, jestli se neliší od předchozího stavu. V případě, že je tato podmínka splněna, následuje kontrola stavu pinu DT. V případě, že pin DT je v jiném stavu než pin CLK, tak se jedná o směr rotace ve směru hodin, tedy vpravo a tato informace je vytištěna po sériové lince společně s přičtením hodnoty 1 do proměnné udávající pozici enkodéru. V opačném případě, tedy že pin DT byl změněn dříve než pin CLK, se jedná o rotaci proti směru hodin, tedy vlevo. Zároveň jsem musel mít hodnoty pouze mezi 0 (min) a 21 (max) to jsem zajistil pomocí dalších funkcí `if`.

3.3 Změna stanice

Změnu stanice jsem vyřešil pomocí tlačítka. Při jeho stisku se k proměnné přičte jednička a v jednoduché funkci `switch` přepne na další `case`. Když je `switch` v posledním stavu, tak se ona proměnná (v mém případě `i`) vrátí zpět do hodnoty 0 a `switch` jede od začátku.

```

Void zmenaStanice(){
  TlacitkoStav = digitalRead(TlacitkoPin);
  Serial.println(TlacitkoStav);
  if (TlacitkoStav == HIGH) {
    i++;  }
  switch(i){
    case 0:
      audio.connectttohost(„http://21283.live.streamtheworld.com/EVROPA2.mp3“);
      delay(500);
      break;
    case 1:
      audio.connectttohost(„http://vis.media-ice.musicradio.com/CapitalMP3“);
      delay(500);
      break;  }
  }
}

```

3.4 Displej

Podle dokumentace a video návodu k displeji LCD1602, jsem zjistil, že má svou adresu. Našel jsem si kód pro zjištění dané adresy a použil jej. Poté jsem si vyzkoušel jednoduchý kód s výpisem mnou zadaných znaků. S displejem nebyl žádný problém, tak jsem jej přidal do mé funkce switch a následně po doplnění kódu o web server k tlačítkům.

```

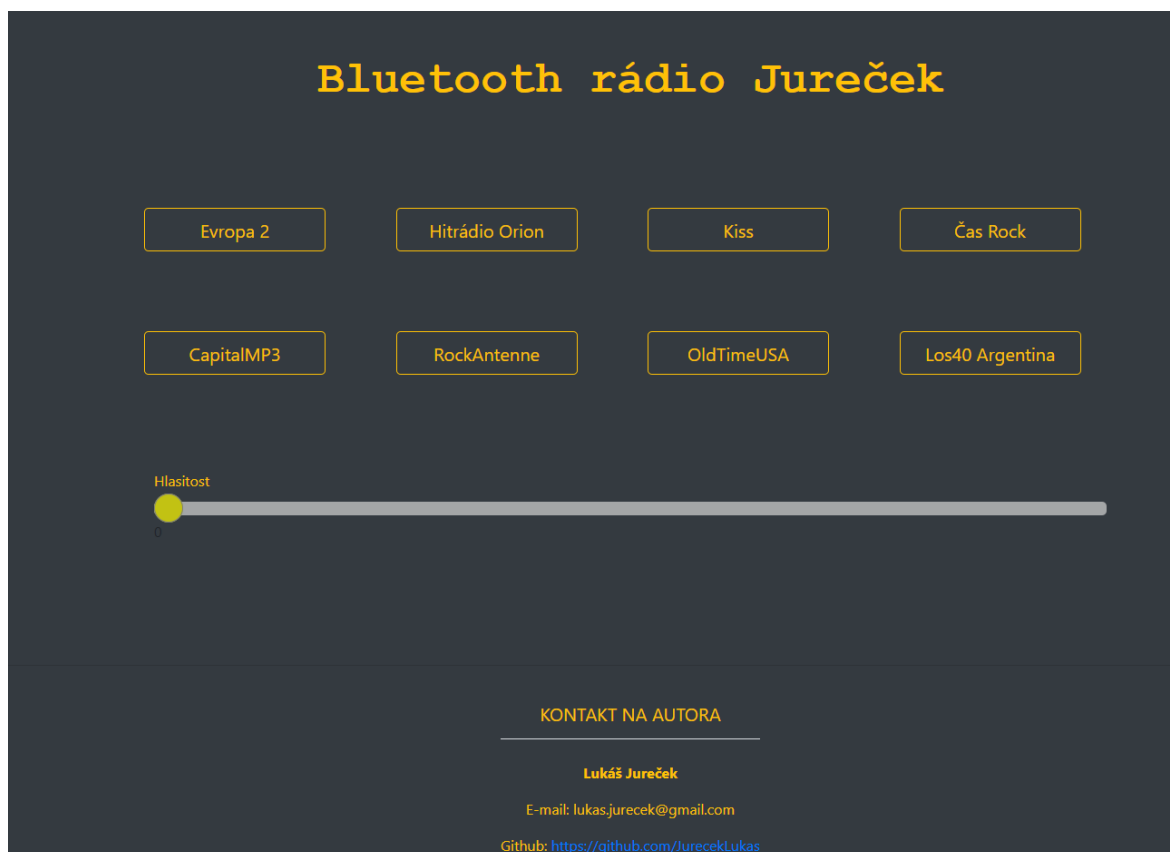
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  lcd.begin();
  lcd.backlight();
  lcd.print(„Wifi radio“);
  lcd.setCursor(0,1);
  lcd.print(„Evropa2“);
}

```

3.5 Web server

Poslední věc, co jsem řešil, bylo přidání web serveru s tlačítky pro změnu stanice a posuvník pro změnu hlasitosti, kterým se dá ovládat internetové rádio i z jiného zařízení. To se připojí skrz IP adresu, kterou nám vytvoří jednotka ESP32, k mnou vytvořenému web serveru, na kterém běží HTML stránka.

Vytvořil jsem si poměrně jednoduchou a přehlednou HTML stránku s několika tlačítky pro změnu stanice a posuvníkem pro regulaci hlasitosti. Za pomoci Bootstrapu 4 (volně dostupná sada nástrojů s CSS šablonami) a CSS stylů, jsem vytvořil základní vzhled. Nahrál jsem do ESP32 pomocí SPIFFSu složku data (HTML, CSS, JS). Pomocí pár řádků jsem zapnul web server. Tlačítka jsem zprovoznil napsáním funkcí GET a SEND. Nakonec v Javascriptu a mainu udělal další funkci pro posuvník na změnu hlasitosti.



Obr3. ukázka webu


```
server.on("/slider", HTTP_GET, [] (AsyncWebServerRequest *request) {  
    String message;  
    if (request->hasParam("value")) {  
        message = request->getParam("value")->value();  
        sliderValue = message;  
        ledcWrite(ledChannel, sliderValue.toInt());  
        posuvnik = sliderValue.toInt();  
    } else {  
        message = "No message sent";  
    }  
    request->send(200, "text/plain", "Hello, GET: " + message);  
});
```

Závěr

Úkolem bylo vyrobit internetové rádio, které by bylo levnější než komerční výrobky a přidat funkci ovládání z jiného zařízení na dálku v rámci stejného připojení k internetu. Finální verze je funkční (až na interní zdroj energie) a na domácí výrobu dobře vypadající. Konstrukce celého zařízení je navržena tak, že se dá jednoduše jakýkoliv komponent zaměnit za jiný, jak z důvodu opravy, nebo vylepšení (např. reproduktory, zesilovač, ...). Bohužel při vývoji a realizaci několik komponentů shořelo (3x ESP32, 2x DAC PCM5102, ...). Kdyby k tomu nedošlo, je výsledný produkt o polovinu levnější oproti komerčním výrobkům.

Samozřejmě je zde pár věcí, které by se daly vylepšit. Jedním z nich by bylo přidání Wifi Manageru na onom web serveru pro snadnější přihlašování k internetu. Momentálně je to řešeno staticky a pro přihlášení k jiné síti je nutno znovu nahrávat kód. Poté by chtěl vyřešit dostatečně silný zdroj energie pro napájení celého rádia. Problém by se dal nejspíše vyřešit koupením o něco silnější baterie. Baterie napsaná v dokumentaci je dostatečně silná pro napájení všech součástek, krom ESP32. Další prostor pro vylepšení je kvalita zvuku, která díky levným reproduktorům a zesilovači není nijak ohromná. Ovšem za tuto cenu je výsledný zvuk obstojný. A navíc je zde možnost připojit vlastní aktivní reproduktory, nebo sluchátka díky Jack portu. Díky promyšlené konstrukci zařízení, lze komponenty ovlivňující kvalitu zvuku (reproduktory, zesilovač, DAC převodník), velice jednoduše nahradit za kvalitnější varianty. Poslední úpravou, co mě napadá, by byl formulář pro přidávání nebo odebírání stanic na onom web serveru, popřípadě rozšíření stávajícího výběru rozhlasových stanic. Momentálně je zde předvoleno osm různých rádiových stanic, což jak pro testování, tak pro mou potřebu je dostačující.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] ESP32 I2S Internet Radio. Full Tutorial & explanation of I2S. For PCM5102 & MAX98357A I2S Decoders. - YouTube. *YouTube* [online]. Copyright © 2020 Google LLC [cit. 31.12.2020]. Dostupné z:
<https://www.youtube.com/watch?v=3ao87AgZ7Bc>
- [2] ESP32 Web Server using SPIFFS (SPI Flash File System) | Random Nerd Tutorials. *Random Nerd Tutorials / Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 31.12.2020]. Dostupné z:
<https://randomnerdtutorials.com/esp32-web-server-spiiffs-spi-flash-file-system/>
- [3] ESP32 Web Server - Arduino IDE | Random Nerd Tutorials. *Random Nerd Tutorials / Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 31.12.2020]. Dostupné z: <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>
- [4] ESP32-based internet radio and bluetooth speaker | Hackaday.io. *Hackaday.io / The world's largest collaborative hardware development community.* [online]. Copyright © 2020 Hackaday [cit. 31.12.2020]. Dostupné z:
<https://hackaday.io/project/168354-esp32-based-internet-radio-and-bluetooth-speaker>
- [5] Arduino - Rotary encoder Simple Example KY-040 - Arduino Project Hub. *Arduino Create* [online]. Copyright © [cit. 31.12.2020]. Dostupné z:
<https://create.arduino.cc/projecthub/vandenbrande/arduino-rotary-encoder-simple-example-ky-040-b78752>
- [6] Using LCD1602 or LCD2004 with ESP32 - YouTube. *YouTube* [online]. Copyright © 2020 Google LLC [cit. 31.12.2020]. Dostupné z:
<https://www.youtube.com/watch?v=COssWn4Pcm4>