

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STROJNÍCKA FAKULTA**

**MIKROPROCESOROVÁ TECHNIKA  
ZÁVEREČNÁ PRÁCA: OVLÁDAČ HLASITOSTI**

# Zoznam obrázkov

1.1	schéma zapojenia . . . . .	2
1.2	operačný zosilňovač MCP602.[2] . . . . .	3
1.3	elektrický kondenzátorový mikrofón.[1] . . . . .	3
1.4	Zapojenie na breadboarde. . . . .	5
1.5	Zapojenie na breadboarde. . . . .	5
1.6	model ovládača aj s obalom. . . . .	6
1.7	model ovládača zadná strana. . . . .	7
1.8	model ovládača predná strana. . . . .	7
1.9	zapojenie pomocou jednostrannej prototypovej dosky . . . . .	8
2.1	Výstup mikrofónu pri intervale merania 20 ms. . . . .	10
2.2	Výstup mikrofónu pri intervale merania 100 ms. . . . .	11

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Hardware</b>	<b>2</b>
1.1 Schéma zapojenia . . . . .	2
1.2 Prototyp zapojenia pomocou breadboardu . . . . .	5
1.3 Prototyp zapojenia pomocou jednostrannej prototypovej dosky . . . . .	6
<b>2 kód</b>	<b>9</b>
2.1 Vysvetlenie kódu . . . . .	9
2.2 Zero() . . . . .	9
2.3 Maxvolume() . . . . .	9
2.4 Volume() . . . . .	10
2.5 Volumelevel() . . . . .	12
2.6 Reactiontime() . . . . .	12
<b>3 Záver</b>	<b>13</b>
<b>Literatúra</b>	<b>14</b>

# Úvod

Problém veľa moderných filmov je, že hladiny zvuku nie sú konsistentné počas celého trvania filmu, najbežnejší prípad je omnoho hlasnejšia hudba ako zvuk konverzácií, čo núti ustavične upravovať zvuk pomocou ovládača. Tento problém nastáva, pretože zvuk moderných filmov je prispôsobený pozerať v kinách a nie na domáce pozeraanie.

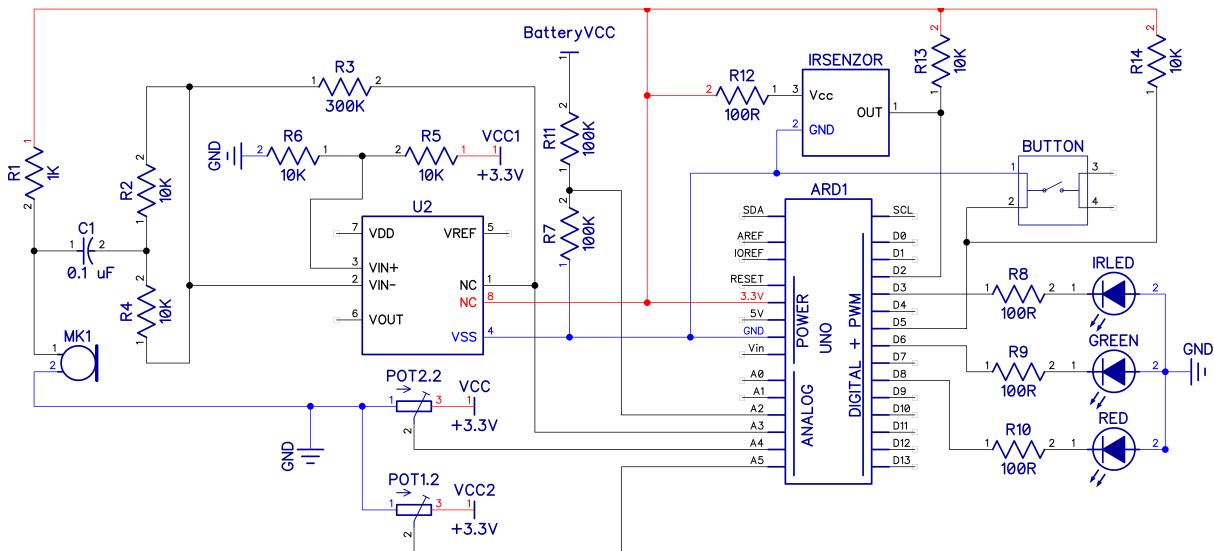
Jedno z riešení je zapnutie dynamickej kompresie zvuku v televíznom prijímači, avšak nie každý televízor má túto funkciu. Naše zariadenie ponúka riešenie pre televízory bez tejto funkcie.

Princíp fungovania nášho zariadenia je založený na prijímaní signálu z mikrofónu, ktorého výška určuje či budeme zvyšovať alebo znížovať hlasitosť televízora a tým meniť výstup infračervenej LED mierenej na televízor

# 1 Hardware

## 1.1 Schéma zapojenia

Schéma zapojenia je pomerne jednoduchá obr.1.1. Jej najdôležitejšiu časť tvorí operačný zosilňovač MCP602 obr.1.2. Operačný zosilňovač je jednosmerne viazaný elektronický zosilňovač napäťa s vysokým ziskom s diferenciálnym vstupom a zvyčajne s jednosmerným výstupom. V tejto konfigurácii optický zosilňovač vytvára výstupný potenciál (vzhľadom na zem obvodu), ktorý je zvyčajne 100 000-krát väčší ako rozdiel potenciálov medzi jeho vstupnými svorkami.



Obr. 1.1: schéma zapojenia.

Na tento zosilňovač máme pripojený elektrický kondenzátorový mikrofón obr.1.3. Skladá sa z dvoch platení, jednej pevnej a druhej pohyblivej. Vibracie vzduchu sa transformujú na posun pohyblivej membrány, ktorá vytvára zmenu elektrického potenciálu. Túto zmenu zachytáva snímač, ktorý následne vysiela elektrický signál. Elektrický signál sa zosilní a následne je spracovaný v arduine.

Ďalšou kritickou časťou systému je infračervená LED dióda, pomocou ktorej posielame signály potrebné na ovládanie televízora. Je dôležité podotknúť že nie všetky televízory resp. ovládače, fungujú na princípe infračervenej LED diódy, avšak v dnešnej dobe pracuje na tomto princípe väčšina zariadení. LED dióda vysiela špeciálne príkazy pomocou modulácie dĺžok medzier medzi vyslanými signálmi ktoré následne zachytáva a spracováva



Obr. 1.2: operačný zosilňovač MCP602.[2]



Obr. 1.3: elektrický kondenzátorový mikrofón.[1]

televízor.

Následne sa na doske nachádzajú aj 2 potenciometre ktoré slúžia na nastavovanie hodnôt potrebných pre správne fungovanie API a 2 LED diódy, jedna zelená, druhá červená, ktoré slúžia len na vizuálnu kontrolu fungovania systému keďže infračervené lúče vysielané IR LED diódou sú pre naše oči neviditeľné.

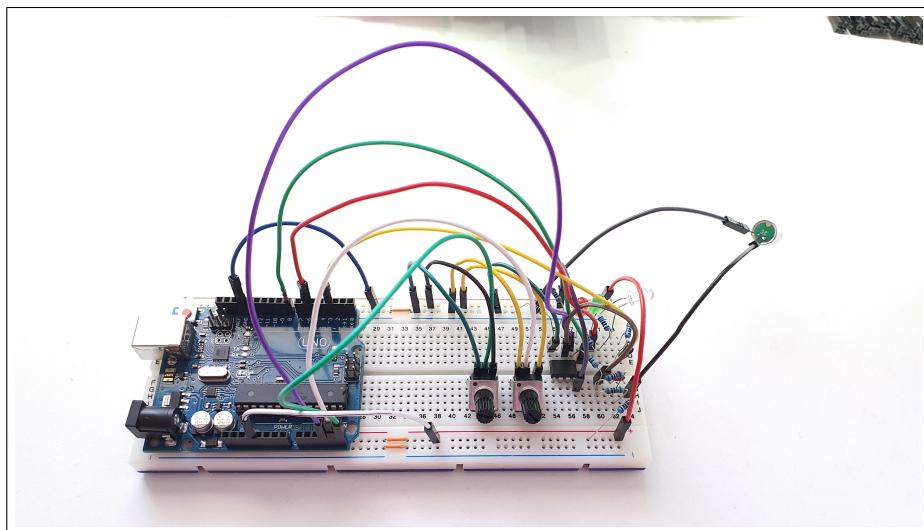
Na schéme môžeme takisto vidieť IR snímač VS1838B ktorý sme chceli využiť na zapisovanie signálov na zvyšovanie a znižovanie hlasitosti. Na spustenie tejto funkcie slúži tlačidlo s názvom BUTTON ktoré je tiež viditeľné na schéme. Žiaľ, túto funkciu sa nám nepodarilo úspešne implementovať do nášho kódu. Problémom pri spolupráci na projekte

bol lockdown a nemožnosť spoločnej práce v reálnom svete ale iba pomocou online komunikácie. Práca s hardwarom tak bola náročnejšia a testovanie kódu, ktorý sme napísali, bolo obmedzené.

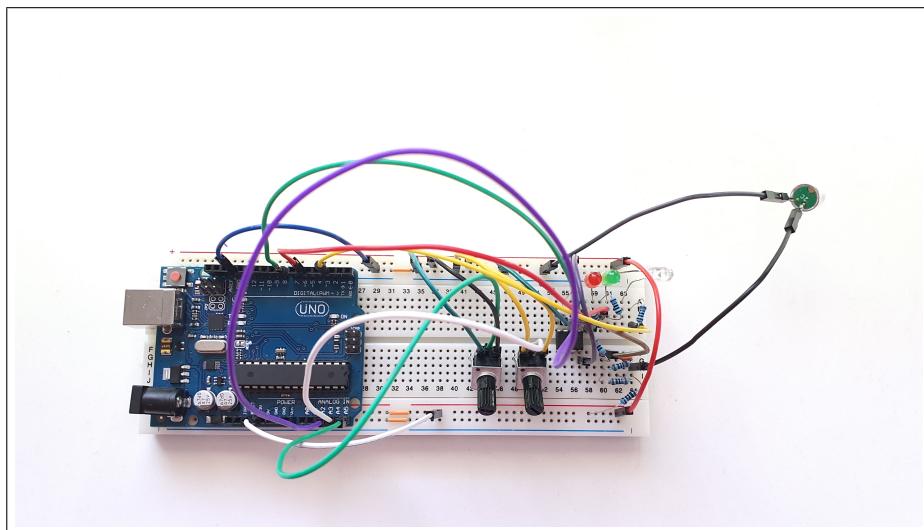
Hlavným nedostatkom tohto kódu bolo správne zapísanie získaných hodnôt do funkcie IRsenderu. Kód sme úspešne detegovali a vedeli ho uložiť do pamäte EEPROM, no následné zapísanie uloženého kódu a jeho poslanie nefungovalo ani po mnohých iteráciách. Kód bol teda na 99% funkčný, ale jeho najdôležitejšia časť, posielanie signálov, nám nechcela fungovať. V situácii kedy by bolo na projekt viacej času, túto funkciu by sme radi implementovali do ovládača hlasitosti.

## 1.2 Prototyp zapojenia pomocou breadboardu

Na otestovanie správnosti a funkčnosti kódu bol zostavený model systému na breadboarde obr.1.5. Systém po niekoľkých iteráciách softwaru a hardwaru fungoval s vysokou presnosťou a bolo možné ho vyladiť podľa potrieb užívateľa. Na ovládanie prvej verzie ovládača hlasitosti sme použili Arduino UNO ktorého sériový port a vykreslovanie meraných veličín nám veľmi uľahčili proces vylepšovania kódu.



Obr. 1.4: Zapojenie na breadboarde.



Obr. 1.5: Zapojenie na breadboarde.

### 1.3 Prototyp zapojenia pomocou jednostrannej prototypovej dosky

Ako ovládaci prvok v 2. generácii obr.1.6 systému sme si zvolili Arduino pro mini, ktorého rozmery a cena boli vyhovujúce pre naše potreby. Schéma zapojenia pre Arduino pro mini je totožná so zapojením pre Arduino UNO až na to, že Arduino pro mini sme sa rozhodli napájať pomocou 3,7V lítiovej batérie zapojenej v kontrolóri nabíjania batérie 03962A. Na správne fungovanie lítiovej batérie je kritické nedovoliť batérii úplne sa vybiť. Z tohto dôvodu budeme pomocou analógového pinu A2, merať napätie batérie a pri kritickom napätií sa spustí blikanie led diódy obr.1.8, ktorá indikuje nutnosť nabitia batérie pomocou mikro USB portu obr.1.7.



Obr. 1.6: model ovládača aj s obalom.

Prototyp číslo 2 fungoval počas testovania správne a všetky funkcie sme vyladili priamo pre tento systém. Avšak pred odovzdaním projektu sa vyskytol problém so zapojením obvodov a výstupy do Arduina prichádzali v zlom tvare obr.1.9. Problém sme odhadli na skrat medzi súčiastkami zapojenia kedy sa do výstupu mikrofónu pridával neznámy šum, ktorý znehodnotil výsledky merania mikrofónu. Chybu sme sa snažili lokalizovať a odstrániť, avšak aj napriek veľkej snahe sa nám chybu nepodarilo nájsť a opraviť.

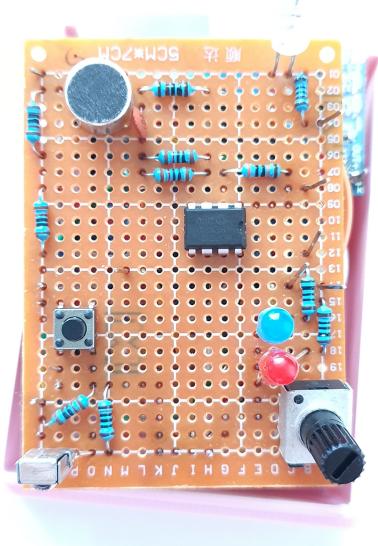
Kvôli zabezpečeniu všetkých komponentov sme sa rozhodli vyrobiť pre náš ovládač hlasitosti aj obal ktorý bol vytlačený na 3D tlačiarni. Obal má na vrchnej strane výrezy pre všetky potrebné komponenty ako aj popisy funkcií akčných členov systému (potenciometrov). Z bočnej strany obalu je výrez pre vypínač s ktorým sa dá zariadenie vypnúť a zapnúť. Z druhej strany je výrez pre nabíjanie batérie a piny pre pripojenie Arduina ku počítaču.



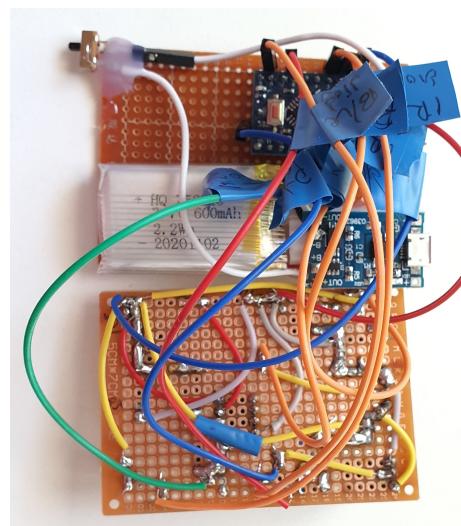
Obr. 1.7: model ovládača zadná strana.



Obr. 1.8: model ovládača predná strana.



(a) vrchná strana prototypovej dosky.



(b) spodná strana spojov

Obr. 1.9: zapojenie pomocou jednostrannej prototypovej dosky

# 2 kód

## 2.1 Vysvetlenie kódu

Pre správne fungovanie zariadenia sme potrebovali veľa funkcií na spracovanie signálu z mikrofónu do formy, s ktorou by sme mohli pracovať.

## 2.2 Zero()

Prvá funkcia potrebná na analýzu signálu je funkcia zero(). Jej úlohou je nájsť nulovú hladinu, číslo okolo ktorého oscilujú hodnoty signálu prijatého z mikrofónu. Princíp je nasledovný: do pomocnej premennej sum ukladá 5000 nameraných hodnôt takým spôsobom, že ich sčítava dohromady. Na konci tohto cyklu vydelí konečné číslo počtom nameraných hodnôt, čím získame priemer, ktorý sa zároveň rovná nulovej hladine. Táto funkcia sa vykoná v setupe().

```
void zero() {  
    float sum=0;  
    int x;  
    int i=0;  
    while ( i < 5000) {  
        x=analogRead(ReadPin);  
        sum=sum+x;  
        i++;  
    }  
    zerolevel=sum/5000.;  
}
```

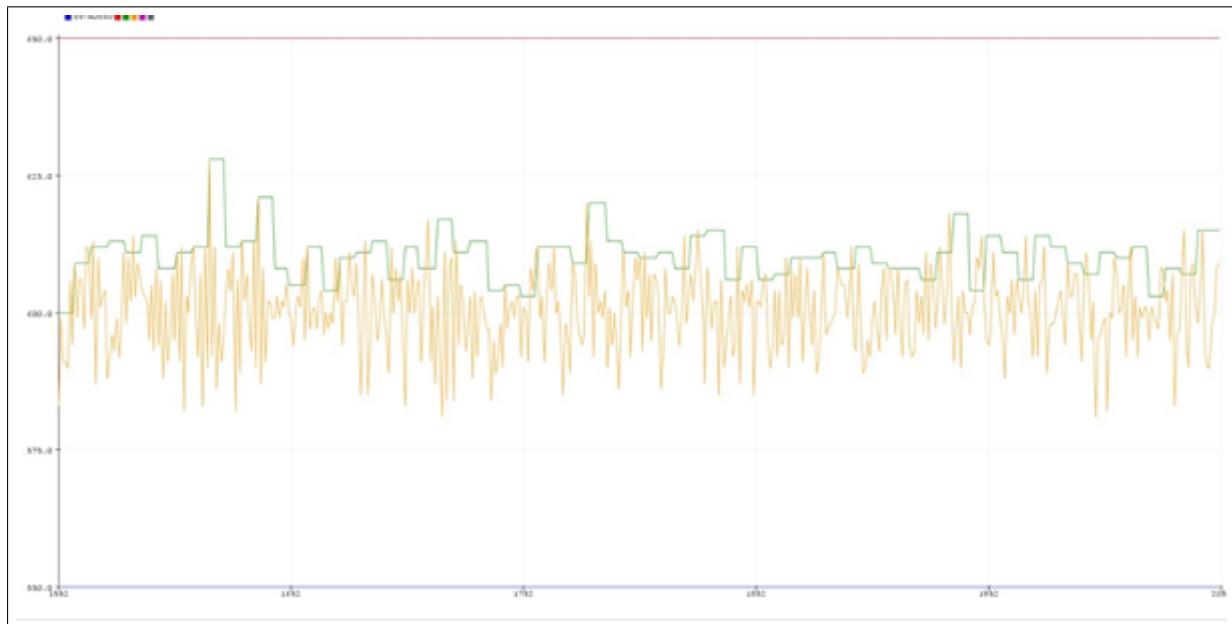
## 2.3 Maxvolume()

Ďalšou dôležitou funkciou je funkcia maxvolume(). Jej úlohou je na konkrétnych intervaloch zisťovať maximálnu hodnotu signálu. Túto hodnotu si drží uloženú v premennej u po dĺžku hľadania novej hodnoty na ďalšom intervale rovnakej dĺžky. Táto funkcia je pre analýzu signálu veľmi dôležitá lebo pomocou nej môžeme signál upraviť do formy, v ktorej budeme vedieť overovať či hlasitosť už prekročila nami určenú hladinu hlasitosti.

```

void maxvolume(int mic) {
    currentMillis = millis();
    Umaxval=zerolevel;
    if (mic>Umaxval) {Umaxval=mic;}
    if (currentMillis - startMillis >= period1)
    {
        u=Umaxval;
        startMillis = currentMillis;
        Umaxval=zerolevel;
    }
}

```



Obr. 2.1: Výstup mikrofónu pri intervale merania 20 ms.

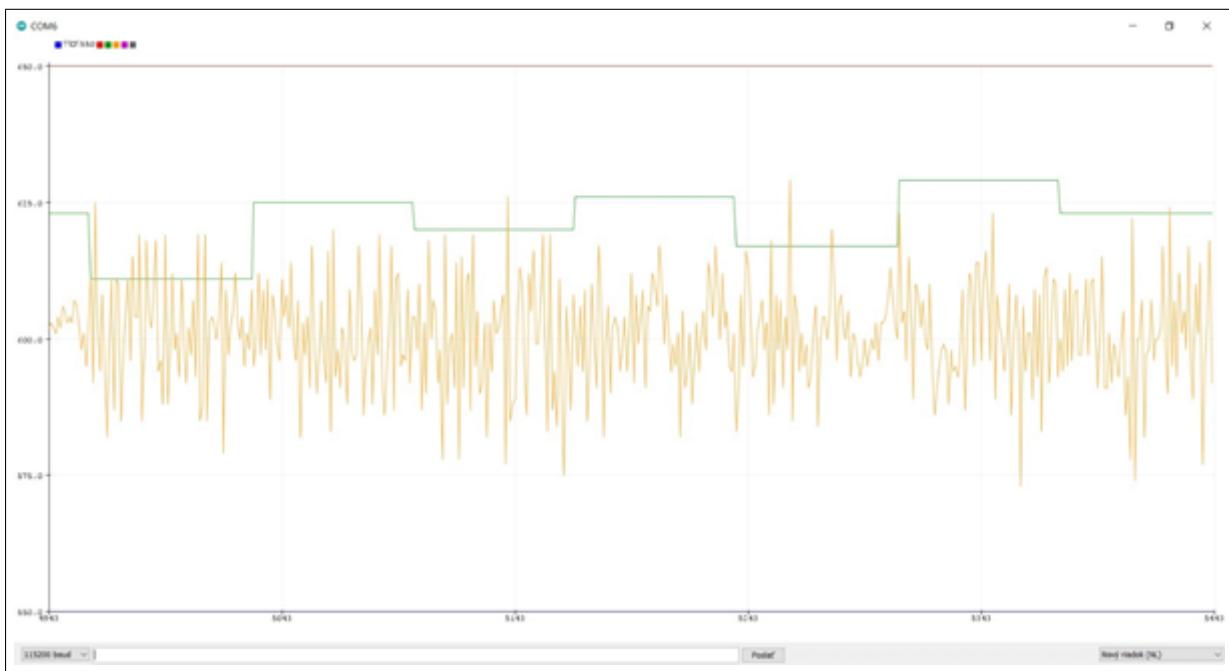
## 2.4 Volume()

Funkcia volume() slúži na zvyšovanie a znižovanie hlasitosti televízora. Kontroluje, či hodnota signálu (u) prekročila nami určenú maximálnu alebo minimálnu prípustnú hladinu. Ak táto situácia nastane počká určitý interval – period2 – a po jeho uplynutí znova overí podmienku. Toto slúži ako ochrana pred chybami, ktoré sa môžu vyskytnúť v priebehu signálu ale taktiež nám umožňuje nastavovať rýchlosť reakcie na zmenu hlasitosti. Ak podmienka platí aj po overení, zariadenie pomocou IR led pošle signál televízoru, ktorý hlasitosť buď zvýší alebo zníži.

```

void volume(int u){
    if (u < minval){
        currentmill = millis();
    }
}

```



Obr. 2.2: Výstup mikrofónu pri intervale merania 100 ms.

```

if (currentmill - startmill >= period2)
{
    digitalWrite(8, HIGH);
    IrSender.sendNEC(0x4, 0x2, 0);
    startmill = millis();
    YieldDelay(100);
    digitalWrite(8, LOW);
    startmill = currentmill;
}
}

if (u > maxval ){
    currentmill = millis();
    if (currentmill - startmill >= period2)
    {
        digitalWrite(6, HIGH);
        IrSender.sendNEC(0x4, 0x3, 0);
        YieldDelay(100);
        digitalWrite(6, LOW);
        startmill = currentmill;
    }
}
}

```

Nasledujúce funkcie slúžia na ovládanie a nastavovanie parametrov programu pomocou potenciometrov.

## 2.5 Volumelevel()

Pomocou tejto funkcie si cez potenciometer dokážeme nastaviť aktuálnu výšku maximálnej (maxval) a minimálnej (minval) prípustnej hladiny hlasitosti bez toho by sme prepisovali program. V programe si musíme len nastaviť hodnoty, v ktorých sa budú môcť dané hodnoty pohybovať a maximálnu a minimálnu hodnotu získanú z potenciometra.

```
void volumelevel() {  
    int MAXVAL_maxlevel = zerolevel + 60;  
    int MAXVAL_minlevel = zerolevel + 10;  
    int MINVAL_maxlevel = zerolevel + 30;  
    int MINVAL_minlevel = zerolevel + 5;  
  
    int minpot=0;  
    int maxpot=660;  
    float x=analogRead(LevelPin);  
  
    maxval=((x/(maxpot-minpot))*(MAXVAL_maxlevel -  
        MAXVAL_minlevel))+MAXVAL_minlevel;  
  
    minval=((x/(maxpot-minpot))*(MINVAL_maxlevel -  
        MINVAL_minlevel))+MINVAL_minlevel;  
}
```

## 2.6 Reactiontime()

Táto funkcia slúži na nastavovanie rýchlosťi reakcie zariadenia na zmenu signálu z mikrofónu pomocou potenciometra v reálnom čase. Tak isto ako vo funkcii volumelevel() si v programe musíme nastaviť hodnoty, v ktorých intervale sa budeme pri nastavovaní pohybovať – maxperiod a minperiod. Tiež si nastavíme maximálnu a minimálnu hodnotu získanú z potenciometra.

```
void reactiontime () {  
    int minpot=0;  
    int maxpot=660;  
    float x=analogRead(Period2Pin);  
  
    int maxperiod = 5000;  
    int minperiod = 500;  
  
    period2 = ((x/(maxpot-minpot))*(maxperiod-minperiod))+  
        minperiod;  
}
```

V loope programu teda len voláme dané funkcie až na zero(), ktorá je v setupe.

## 3 Záver

Cieľom projektu bolo navrhnúť a skonštruovať zariadenie na úpravu hlasitosti televízora, ktorý sme splnili, až na technický problém pri finálnom testovaní.

Pri postupe sme najprv navrhli po častiach jednotlivé funkcie kódu ktoré sme testovali na breadboarde. Jediná funkcia ktorá sa nám nepodarila realizovať bola predošle spomenutá funkcia na ukladanie kódov zmeny hlasitosti. Táto funkcia by iba vylepšila zariadenie a nebola kľúčová pre fungovanie, preto sme sa rozhodli ju vynechať. Keď sme mali pripravené všetky časti kódu, spojili sme ich do jedného programu a otestovali sme ho na prvom prototype – zapojenie na breadboarde. Po ďalších úpravách sme zostavili prototyp 2 aj s obalom vyrobeným v 3D tlačiarni.

Hlavný problém, s ktorým sme sa stretli pri tomto projekte bola obmedzená komunikácia a lockdown, kvôli ktorému bol znemožnený osobný kontakt a online komunikácia bola nedostačujúca.

Aj napriek týmto komplikáciám sme si overili schopnosť využitia získaných vedomostí počas tohto semestra, čo považujeme za významný osobný prínos.

# Literatúra

- [1] Electret condenser microphone. store. online., 2021. 2021, <https://www.pcboard.ca/electret-condenser-microphone>.
- [2] Mcp602. store. online., 2021. 2021, <https://www.microchip.com/en-us/product/MCP602>.