**Semestrální práce**

FIT ČVUT

BI-ARD

**Zařízení pro ovládání zvuku v počítači**

Matěj Sobotka

LS 2023/2024

Obsah

[1. Specifikace zadání 3](#_Toc164680072)

[2. Teoretický rozbor 4](#_Toc164680073)

[2.1. Seznam součástek 4](#_Toc164680074)

[2.2. Popis zařízení 4](#_Toc164680075)

[2.3. Schéma zapojení 5](#_Toc164680076)

[2.4. Popis práce programu na Arduinu 6](#_Toc164680077)

[2.5. Popis práce programu na počítači 6](#_Toc164680078)

[2.5.1. Třída CSerial 7](#_Toc164680079)

[2.5.2. Třída CVolume 8](#_Toc164680080)

[3. Obal zařízení 9](#_Toc164680081)

[4. Dokumentace a uživatelská příručka 10](#_Toc164680082)

[5. Závěr 11](#_Toc164680083)

[6. Seznam obrázků a tabulek 12](#_Toc164680084)

[7. Seznam použité literatury 13](#_Toc164680085)

1. Specifikace zadání

Cílem projektu je tvorba univerzálního ovladače zvuku v počítači s OS Windows. Modul je opatřen několika ovládacími prvky, pomocí kterých lze ovládat libovolné zvukové kategorie (hry, prohlížeče, VoIP, …). Ovládání zvuku na hostitelském počítači zpracovává aplikace, která přijímá data od modulu a zároveň je v ní možné vytvořit vlastní kategorie a přiřadit je k jednotlivým ovládacím prvkům.

1. Teoretický rozbor
   1. Seznam součástek

|  |  |  |
| --- | --- | --- |
| **Název** | **Typ** | **Počet (ks)** |
| Arduino Nano | Arduino Nano ATmega328 CH340G | 1 |
| Rotační enkodér | KY-040 | 3 |
| Neopixel led ring | WS2812B | 3 |

Tabulka 1: Seznam použitých součástek

* 1. Popis zařízení

Zařízení je vybaveno třemi rotačními enkodéry (dále jen enkodér). Při otáčení ve směru hodinových ručiček dochází ke zvyšování hlasitosti, při otáčení v opačném směru pak k snížení hlasitosti. Tyto rotační enkodéry jsou zároveň vybaveny tlačítkem, které zprostředkovává funkci okamžitého ztlumení. Kolem každého enkodéru je kruh led diod (dále jen led kruh), který signalizuje aktuální úroveň hlasitosti, případně červeně signalizuje, že je aktivováno ztlumení.

O komunikaci s hostitelským počítačem (dále jen počítač) se stará deska Arduino Nano (dále jen Arduino), ke které je připojeno 6 součástek (3x enkodér a 3x led kruh). Komunikace mezi počítačem a Arduinem probíhá přes integrovaný USB port na Arduinu a využívá sériovou linku.

Celé zařízení je uložené v plastové schránce a tisknutelné modely budou volně dostupné se stažení.

* 1. A diagram of a circuit board

     Description automatically generatedSchéma zapojení

Obrázek 1: Schéma zapojení

Piny A a B na všech enkodérech vyžadují pull-up rezistory, které budou řešeny interně v Arduinu, a proto nejsou v schématu zakresleny.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Arduino PIN** | **Součástka PIN** | **Součástka** | **Arduino PIN** | **Součástka PIN** | **Součástka** |
| D7 | 4 (BTN) | Enkodér 0 | D12 | 4 (BTN) | Enkodér 2 |
| D2 | A | Enkodér 0 | D4 | A | Enkodér 2 |
| D5 | B | Enkodér 0 | D13 | B | Enkodér 2 |
| D8 | 4 (BTN) | Enkodér 1 | D9 | IN | Led kruh 0 |
| D3 | A | Enkodér 1 | D10 | IN | Led kruh 1 |
| D6 | B | Enkodér 1 | D11 | IN | Led kruh 2 |

Tabulka 2: Zapojení pinů

* 1. Popis práce programu na Arduinu

Celý program je napsaný v jazyce C++ pomocí Arduino IDE. Důležitou součástí je knihovna *Adafruit Neopixel[[1]](#footnote-1)*, která se umožňuje ovládat led kruhy pomocí jednoho drátu. Kód bude dostupný na mém osobním GitHubu[[2]](#footnote-2).

Program pracuje s informacemi získanými z enkodérů. Pokud dojde ke změně, tj. uživatel otočí enkodérem nebo zmáčkne odpovídající tlačítko program vyšle přes sériovou linku příkaz ve formátu *ID:CH:BTN*, kde *ID* značí identifikační číslo enkodéru (0, 1, 2), *CH* značí změnu, neboli pokyn na zvýšení, či snížení hlasitosti (+/-) a *BTN* značí, zda-li došlo ke stisku tlačítka enkodéru, které vyvolá funkci mute. Otočení enkodérem je detekováno přes změnu stavu z HIGH na LOW na pinu A, směr otáčení je pak detekováno pomocí hodnoty na pinu B (pokud je HIGH, hlasitost se zvýší, v opačném případě se sníží). Stisk tlačítka je pak detekován změnou stavu z HIGH na LOW na pinu 4 enkodéru.

Zároveň program zobrazuje informace o aktuální úrovni hlasitosti pomocí led kruhů kolem jednotlivých enkodérů (prvotní informace při zapnutí/připojení získá od počítače). Pokud je aktivována funkce mute, všech 8 led diod svítí červeně. Led diody se rozsvěcují postupně, ale skokově po 6 bodech hlasitosti a to následujícím způsobem: 1-6 = 50% jas, 7-12 = 100% jas. Pokud hlasitost přesáhne násobek 12, rozsvítí se další led dioda.

Pokud dojde k odpojení zařízení, led kruhy blikají červeně.

* 1. Popis práce programu na počítači

Program je psaný v jazyce Python a rozdělen do několika souborů pro možnou modularitu. Důležitou součástí jsou knihovny *comtypes[[3]](#footnote-3)*, *pycaw[[4]](#footnote-4)*, *serial[[5]](#footnote-5)*.

Jelikož je program členěn do několika součástí, je možné v budoucnu přidat třídu CVolume pro linux a mac se stejným rozhraním jako CVolume pro Windows.

Zároveň je možné nahradit třídu CSerial jinou, která bude například zprostředkovávat bezdrátovou komunikaci nebo koumunikaci přes jiný protokol. Celý kód bude také dostupný na mém osobním GitHubu[[6]](#footnote-6).

Uživatel nastavuje program v příkazové řádce, příkazy budou uvedeny v uživatelské dokumentaci.

V případě OS Windows program modifikuje vestavěný volume mixer, pomocí kterého je možné ovládat jednotlivé aplikace, které přehrávají zvuk. V programu se konfiguruje, jaký enkodér přísluší jaké aplikace nebo aplikacím a je ošetřeno, že pokud nakonfigurovaná aplikace zrovna neběží, program nespadne. Zároveň je umožněno měnit hlasitost hlavního zvukového výstupu.

Pokud dojde k odpojení zařízení, program tuto skutečnost pouze konstatuje a pokusí se o znovunavázaní spojení, případně má uživatel možnost spojení inicializovat znova. Při úspěšném navázaní spojen í dochází k synchronizaci s Arduinem pomocí příkazů ve formátu *ID:VOL*, kde *ID* značí číslo enkodéru a *VOL* aktuálně nastavenou úrověň zvuku v počítači.

Hlavní část programu pouze rozděluje příchozí příkazy a o samotnou konfiguraci se starají jednotlivé třídy. V třídě CVolume jsou předdefinované seznamy aplikací, ale uživatel si přirozeně může vytvořit svoje, a to opět přes příkazovou řádku. V budoucnu je plánované udělat grafickou nadstavbu, která bude uživatelsky přívětivější.

* + 1. Třída CSerial

Třída s jednoduchým rozhraním, při inicializaci se specifikuje sériový port, na který je zařízení připojeno (opět pomocí výběru skrze příkazovou řádku).

Třída dále obsahuje metody *readData()* a *writeData(data)*. Při čtení dat jsou návratovou hodnotou samotná data uložená jako string, v případě posílání dat jsou data konvertována na string a následně odeslána.

* + 1. Třída CVolume

Tato třída má již složitější rozhraní a stará se o samotnou úpravu nativního volume mixeru v OS Windows. Při inicializaci se pouze načte aktuální výstupní zařízení a jsou funkční metody *getMasterState()*, *toggleMasterState()*, *setMasterState(int)*, *getMasterVolume()*, *setMasterVolume(int)*, *setMasterVolumeIncNative(int)*, *setMasterVolumeDecNative(int)*. State zde reprezentuje funkci mute. Pokud je state nastaven do úrovně 1, zařízení nebo aplikace jsou ztlumeny.

Poslední dvě metody jsou obzvláště důležité, jelikož v základu je úroveň hlasitosti reprezentována jako hodnota z intervalu a škála není lineární. Tyto metody pak zaručují, že dochází ke správnému zvýšení nebo snížení hlasitosti v souladu s otáčením enkodéru.

**Popis metod pro práci s jednotlivými seznamy aplikací bude doplněn později. V zásadě se jedná o analogické metody metodám master, jen jako další parametr berou zmíněný seznam a poté dochází k iteraci přes prvky seznamu a na každý se zavolá odpovídající metoda (set, toggle, get).**

1. Obal zařízení
2. Dokumentace a uživatelská příručka
3. Závěr
4. Seznam obrázků a tabulek

[Obrázek 1: Schéma zapojení 5](#_Toc164680086)

[Tabulka 1: Seznam použitých součástek 4](#_Toc164680088)

[Tabulka 2: Zapojení pinů 5](#_Toc164680089)

1. Seznam použité literatury

1. <https://github.com/adafruit/Adafruit_NeoPixel> [↑](#footnote-ref-1)
2. <https://github.com/CZMates00/volumeControlArd> [↑](#footnote-ref-2)
3. <https://github.com/enthought/comtypes> [↑](#footnote-ref-3)
4. <https://github.com/AndreMiras/pycaw> [↑](#footnote-ref-4)
5. <https://github.com/pyserial/pyserial> [↑](#footnote-ref-5)
6. <https://github.com/CZMates00/volumeControlArd> [↑](#footnote-ref-6)