



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## HARDWAROVÝ MODUL PRO INTERAKTIVNÍ OVLÁDÁNÍ PROJEKTŮ V PUREDATA

HARDWARE MODULE FOR INTERACTIVE CONTROL OF PROJECTS IN PUREDATA

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Richard Ladislav

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Schimmel, Ph.D.

BRNO 2023



# Bakalářská práce

bakalářský studijní program **Audio inženýrství**  
specializace Zvuková technika  
Ústav telekomunikací

**Student:** Richard Ladislav

**ID:** 230106

**Ročník:** 3

**Akademický rok:** 2022/23

## NÁZEV TÉMATU:

### Hardwarový modul pro interaktivní ovládání projektů v PureData

#### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncept a základovou desku připojení periférií k Arduino UNO R3, které budou umožňovat interaktivní řízení projektů v PureData. Jednotlivé periférie budou studenti připojovat sami pomocí kabelů a není požadováno, aby bylo možné připojit všechny periférie současně. Dbejte ale na jednoduchost a přehlednost propojení a možnost připojit současně alespoň dvě vstupní a jedno výstupní zařízení. Deska bude obsahovat tahový potenciometr, rotační enkoder, dva dvoupólové přepínače, alespoň 8 tlačítek nebo tlačítková klávesnice, 8 LED diod různé barvy, mikrofon a piezoelektrický reproduktor. K desce bude také připojen modul gyroskopu s akcelerometrem, ultrazvukovým měřičem vzdálenosti a WiFi. Tyto moduly lze použít již hotové, dostupné na trhu. Deska bude obsahovat i další nezbytné obvody pro připojení těchto periférií, aby bylo možné je přímo propojit s vstupními a výstupními piny procesoru. Vytvořte ukázkové kódy pro využití všech periférií a vytvořte šablony, které umožní jejich snadné použití v prostředí Pure Data. Základovou desku realizujte, osadte ji a všechny periférie odzkoušejte.

#### DOPORUČENÁ LITERATURA:

- [1] SELECKÝ, Matúš. Arduino: uživatelská příručka. Brno: Computer Press, 2016. ISBN 978-802-5148-402.
- [2] KAVAN, Jan. Pure Data: platforma pro tvorbu interaktivního díla. Brno: Janáčkova akademie múzických umění v Brně, 2013. ISBN 8074600335

**Termín zadání:** 6.2.2023

**Termín odevzdání:** 26.5.2023

**Vedoucí práce:** doc. Ing. Jiří Schimmel, Ph.D.

**doc. Ing. Jiří Schimmel, Ph.D.**  
předseda rady studijního programu

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



## **ABSTRAKT**

Táto bakalárska práca sa zaoberá konceptom a návrhom základnej dosky slúžiacej na pripojenie rady periférií k mikrokontroléru Arduino UNO R3 využívanej na ovládanie interaktívnych projektov v programovacom prostredí Pure Data. Doska obsahuje ťahový potenciometer, rotačný enkodér, dva dvojpólové prepínače, 8 LED diód, 8 tlačidiel, piezoelektrický reproduktor a elektretový mikrofón. Periférie využité v realizácii tohto projektu boli doplnené o potrebné obvody a sú permanentne pripojené k vstupným a výstupným pinom mikrokontroléru. Súčasťou práce sú taktiež kódy v jazyku Wiring a základné patche vytvorené v prostredí Pure Data, slúžiace na testovanie jednotlivých periférií.

## **KLÚČOVÉ SLOVÁ**

Arduino, Pure Data, ATmega328P, CAC, CMG, základná doska

## **ABSTRACT**

This bachelor's thesis deals with the concept and design of a motherboard used to connect a range of peripherals to the Arduino UNO R3 microcontroller, which is used for controlling interactive projects in the Pure Data programming environment. The board includes a slide potentiometer, a rotary encoder, two double-pole switches, 8 LED diodes, 8 buttons, a piezoelectric speaker, and an electret microphone. The peripherals used in the implementation of this project have been supplemented with the necessary circuits and are permanently connected to the input and output pins of the microcontroller. The thesis also includes code written in the Wiring language and basic patches created in the Pure Data environment, which are used for testing the individual peripherals.

## **KEYWORDS**

Arduino, Pure Data, ATmega328P, CAC, CMG, motherboard



LADISLAV, Richard. *Hardwarový modul pro interaktivní ovládaní projektů v PureData*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 83 s. Bakalářská práce. Vedúci práce: doc. Ing. Jiří Schimmel, PhD.





## Vyhlásenie autora o pôvodnosti diela

**Meno a priezvisko autora:** Richard Ladislav  
**VUT ID autora:** 230106  
**Typ práce:** Bakalárska práca  
**Akademický rok:** 2022/23  
**Téma záverečnej práce:** Hardwarový modul pro interaktivní ovládaní projektů v PureData

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podpisuje iba v tlačenej verzii.



## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárske práce pánovi doc. Ing. Jiřímu Schimmelovi PhD. za odborné vedenie, konzultácie, podnecujúce návrhy k práci a hlavne za trpezlivosť. Taktiež by som rád poďakoval svojim rodičiom za nemalú podporu pri štúdiu. Ďalej by som rád poďakoval vyčujúcemu tímu z mikroeletrotechyného praktika za to, že ma naučili držať pájku tak, aby som si nespálil všetkých päť prstov. Nemalá vďaka patrí tiež internetovým fóram a komunitám, ktoré ma zachránili v nejednej zdanlivo neriešiteľnej situácii. Špeciálna vďaka patrí Jane Vaverkovej, bez ktorej by som to v oných zdanlivo neriešiteľných situáciach dávno vzdal.



# Obsah

Úvod	17
<b>1 Teoretický úvod</b>	<b>19</b>
1.1 Pure Data	19
1.1.1 História a motivácia vývoja Pure Data	19
1.1.2 Príchute Pure Data	20
1.1.3 Grafické programovacie jazyky	20
1.1.4 Základy programovacieho prostredia Pure Data	21
1.2 Arduino	23
1.2.1 História a motivácia vzniku Arduina	23
1.2.2 Arduino UNO R3	24
1.2.3 ATmega 328P	25
1.2.4 Zbernice	26
<b>2 Prvotné testovanie rozhraní</b>	<b>29</b>
2.1 Pure Data testovací patch pre vstupné periférie	29
2.2 Pure Data testovací patch pre výstupné periférie	31
2.2.1 Testovanie modulu PS2 Joy Stick pre Arduino	32
2.2.2 Testovanie modulu s akcelerometrom a gyroskopom MPU 6050	34
2.2.3 Testovanie Ultrazvukového senzoru HY-SRF 05	35
2.2.4 Testovanie tlačidiel	36
2.2.5 Testovanie rotačného enkodéra KY-040	38
2.2.6 Testovanie Wifi modulu nRFL01	40
2.2.7 Testovanie LED diód	41
2.2.8 Testovanie lineárneho ťahového potenciometra pre Arduino	42
2.2.9 Testovanie piezoelektrického reproduktora	43
2.2.10 Testovanie modulu mikrofónu so zosilňovačom MAX4466	44
2.2.11 Testovanie dvojpólových prepínačov	45
<b>3 Návrh Zariadenia</b>	<b>49</b>
3.1 Rozloženie periférií	49
3.2 Prvotný koncept zariadenia	50
3.3 Kultúra shieldu	52
3.4 Pinout	53
3.4.1 Redukcia potrebných pinov tlačidiel	53
3.4.2 Redukcia potrebných pinov LED diód	57
3.5 Finálny pinout zariadenia	59

<b>4 Konštrukcia zariadenia</b>	<b>61</b>
4.1 Realizácia enkodéru na DPS . . . . .	61
4.2 Realizácia GPIO expadnéru PCF8574 v zapojení s tlačidlami na DPS	61
4.3 Realizácia GPIO expadnéru PCF8574 v zapojení s LED diódami na DPS . . . . .	63
4.4 Doska plošných spojov . . . . .	63
4.4.1 TOP vrstva . . . . .	64
4.4.2 BOTTOM vrstva . . . . .	65
<b>Záver</b>	<b>67</b>
<b>Literatúra</b>	<b>69</b>
<b>Zoznam symbolov a skratiek</b>	<b>71</b>
<b>Zoznam príloh</b>	<b>73</b>
<b>A Dokumentácia DPS</b>	<b>76</b>
A.1 Schéma zapojenia . . . . .	76
A.2 TOP vrstva DPS . . . . .	77
A.3 BOTTOM vrstva DPS . . . . .	78
A.4 Osadenie súčiastok vrstvy TOP . . . . .	79
A.5 Osadenie súčiastok vrstvy BOTTOM . . . . .	80
A.6 Fotografia zariadenia TOP . . . . .	81
A.7 Fotografia zariadenia BOTTOM . . . . .	82
<b>B Obsah elektronickej prílohy</b>	<b>83</b>

# Zoznam obrázkov

1.1	Pinout dosky Arduino UNO Rev. 3., oficiálny datasheet Arduino UNO Rev. 3 . . . . .	24
1.2	Blokový diagrma architektúry AVR procesoru, oficiálny datasheet procesoru ATMEGA328P . . . . .	26
1.3	Časový diagram signálu zbernice I <sup>2</sup> C, vytvorený pomocou online nástroja wavedorm.com . . . . .	28
2.1	Testovací patch v prostredí Pure Data pre vstupné zariadenia. . . . .	30
2.2	Testovací patch v prostredí Pure Data pre vstupné zariadenia. . . . .	32
2.3	Pinout PS2 Joy Stick, vytvorené pomocou návrhového prostredia Eagle	34
2.4	Pinout MPU6050, Vytvorené pomocou návrhového prostredia Eagle .	35
2.5	Pinout HY-SRF05, vytvorené pomocou návrhového prostredia Eagle .	36
2.6	Úprava v Pure Data patchy pre testovanie vstupných periférií, pre účely testovania tlačidiel . . . . .	37
2.7	Pinou tlačidiel , schéma vytvorené pomocou návrhového prostredia Eagle . . . . .	38
2.8	Princíp fungovania rotačného enkodéru, <a href="https://howtomechatronics.com/tutorials/arduino/encoder-works-use-arduino/">https://howtomechatronics.com/tutorials/arduino/encoder-works-use-arduino/</a> . . . . .	39
2.9	Pinout enkodéru KY-040, vytvorené pomocou návrhového prostredia EAGLE . . . . .	40
2.10	Pinout Wifi modulu nRF101, vytvorené pomocou návrhového prostredia Eagle . . . . .	41
2.11	Pinot LED diód, vytvorené v návrhovom prostredí Eagle . . . . .	42
2.12	Pinout lineárneho ťahového potenciometra, vytvorené v programe Eagle . . . . .	43
2.13	Pinout piezoelektrického reproduktora, vytvorené v návrhovom prostredí Eagle . . . . .	44
2.14	Pinout modul mikrofónu so zosilňovačom MAX4466, vytvorené v návrhovom prostredí Eagle . . . . .	45
2.15	Úprava v Pure Data patchy pre testovanie vstupných periférií, pre účely dvojpólových prepínačov. Do tretieho number boxu sa na základe stavu prepínača vypisuje 1, alebo nula, čím vypíname a zapíname audio signál . . . . .	46
2.16	Pinout dvojpólových prepínačov, vytvorené pomocou návrhového prostredia Eagle . . . . .	47
3.1	Návrh fyzického rozloženia periférií v návrhovom prostredí Fritzing .	50
3.2	Prvotný koncept realizácie modulu pre interaktívne projekty v Pure Data navrhnutý v návrhovom prostredí Fritzing . . . . .	51

3.3	Prvotný koncept so zapojenými prepojuvacími káblami v návrhovom prostredí Fritzing. Zmyslom tejto ukážky je znázorniť problémy, ktoré so sebou koncept prinášal. . . . .	52
3.4	Princíp zapojenia tlačidiel ako odporového deliča, vytvorené v návrhovom prostredí Eagle . . . . .	54
3.5	Princíp fungovania maticovej klávesnice <a href="http://home.zcu.cz/dudacek/PZ/klavesnice.pdf">http://home.zcu.cz/dudacek/PZ/klavesnice.pdf</a> . . . . .	55
3.6	Zjednodušený blokový diagram čipu PCF8574 od firmy Texas Instruments. Oficiálny datasheet súčiastky . . . . .	55
3.7	Pinout tlačidiel pripojených skrz GPIO expandér PCF8574 . . . . .	56
3.8	Pinout LED diód pripojených skrz GPIO Posuvný register 74HC595, vytvorené pomocou návrhového prostredia Eagle . . . . .	57
3.9	Pinout pripojenia všetkých periférií využívajúcich zbernicu I <sup>2</sup> C naraz, vytvorené pomocou návrhového prostredia Eagle . . . . .	58
4.1	Finálne zapojenie enkodéru, vytvorené v návrhovom prostredí Eagle . . . . .	61
4.2	Finálne zapojenie tlačidiel na čip GPIO expandéra vytvorené v návrhovom prostredí Eagle . . . . .	62
4.3	Finálne zapojenie SMD jumpra vytvorené v návrhovom prostredí Eagle . . . . .	62
4.4	Finálne zapojenie LED diód na čip GPIO expandéra vytvorené v návrhovom prostredí Eagle . . . . .	63
4.5	TOP vrstva dosky plošných spojov, vztvorené pomocou návrhového prostredia Eagle . . . . .	64
4.6	BOTTOM vrstva dosky plošných spojov, vztvorené pomocou návrhového prostredia Eagle . . . . .	65



# Úvod

Cieľom tejto bakalárskej práce je vytvorenie hardware modulu pre interaktívne projekty vo vývojovom prostredí Pure Data, ktorý bude slúžiť na edukatívne účely. Pôvodným princípom, na ktorom mal byť modul založený, spočíval v možnosti prepájať jednotlivé periférie rovno ku doske s mikrokontrolérom. V priebehu práce však po konzultácii s vedúcim práce prišlo k zmene konceptu. Ukázalo sa, že modul realizovaný pôvodným konceptom, je náchylný na poškodenie hardware pri nesprávnej manipulácii zo strany študentov a riešenie pomocou rôznych typov ochrán poskytovalo nepraktické riešenie. Nový koncept je založený na permanentnom pripojení všetkých periférií ku doske s mikrokontrolérom a je koncipovaný vo forme Arduino shieldu. Modul obsahuje osem tlačidiel, osem LED diód, piezoelektrický reproduktor, dva dvojpólové prepínače, ťahový potenciometer, rotačný enkodér. Ďalej je možné k doske pripojiť modul s joy stickom, modul s ultrazvukovým meračom vzdialenosti, modul s akcelerometrom a gyroskopom, Wifi modul a modul s elekterovým mikrofónom. Komunikácia medzi hardware a software je sprostredkúvaná pomocou dosky s mikrokontrolérom Arduino Uno Rev. 3. Je nutné podotknúť, že voľba modulových rozhraní ako i doska s mikrokontrolérom Arduino Uno. Rev. 3 je daná už existujúcou výbavou laboratória a ich substitúcia za iné modely nie je žiadúca. Súčasťou práce sú i vzorové kódy a patche využité na testovanie jednotlivých periférií. V teoretickom úvode je rozobraná práve história, kontext a motivácia vzniku, ale i základné princípy fungovania oboch využitých platforiem. V nasledujúcej časti je opísaný proces testovaniu jednotlivých periférií. Práve v tejto časti prichádzalo k oboznámeniu s funkcionalitou jednotlivých rozhraní a uvedomenie si možností ich budúceho využitia. Obsahom nasledujúcej časti je proces návrhu dizajnu, pričom je bližšie opísaná motivácia zmeny pôvodného konceptu, ako i podrobnosti o realizácii finálneho konceptu. Bakalárska práca je ukončená časťou, ktorá je venovaná konkrétnej realizácii dosky plošných spojov.



# 1 Teoretický úvod

## 1.1 Pure Data

Pure Data, alebo často len skrátene „Pd“ je názov open source programovacieho jazyka vyvinutého Millerom Puckettom v roku 1996, ktorý slúži prevažne na vytváranie interaktívnych hudobných a multimediálnych projektov. Jedná sa o vizuálny programovací jazyk patriaci do rodiny tzv. „patcher“ jazykov, taktiež nazývaných i „Max“ rodinou. Do rovnakej rodiny môžeme zaradiť i jazyky ako Max/MSP, Max/FTS, či ISPW Max, ktoré boli taktiež navrhnuté Milerom Puckettom v tom čase pôsobiacim na francúzskom ICRAM (Institut de recherche et coordination acoustique/musique, teda Inštitút pre výskum a koordináciu akustiky/hudby). Pure Data za svoju obľúbenosť medzi hudobnými skladateľmi, vizuálnymi umelcami, performerami a zvyškom predstaviteľov umeleckého sveta vďačí celej rade svojich vlastností. Predovšetkým je to jeho všestrannosť, ktorá umožňuje syntézu postupov naprieč celým spektrom umeleckých disciplín. Ďalšou z výhod je intuitívnosť práce s Pure Data. Na rozdiel od väčšiny programovacích jazykov orientovaných na spracovanie zvuku akými sú SuperCollider, či CSound, prostredie PureData využíva princíp DFP (Data flow programming), teda programovanie na základe popisovania toku dát. Daný princíp je navyše realizovaný graficky prepájaním objektov na obrazovke namiesto písania príkazov. Toto je jeden z hlavných dôvodov, prečo je preferovaný medzi umelcami, ktorých znalosti programovania sú často veľmi laické až nulové. Open source charakter platformy je taktiež veľkým plus. Pure Data je distribuovaný pod SIBDS licenciou, čo zaručuje jeho bezplatné šírenie medzi užívateľov. Medzi jeho výhody taktiež možno zahrnúť kompatibilitu. Je možné ho spustiť na všetkých bežne rozšírených operačných systémoch pre osobné počítače, ako i na smartphone, či jednodoskovom počítači akým je Raspberry Pi. Medzi jeho výhody patrí i možnosť prepojenia s rozmanitou škálou externých senzorov, motorov a iných zariadení.

### 1.1.1 História a motivácia vývoja Pure Data

Pokusy využívať výpočtovú techniku na hudobnú kompozíciu a generovanie zvuku môžeme mapovať až k samotným začiatkom tohto vedného odvetvia. Za dva najvýznamnejšie mílniky pritom považujeme skladbu Illiac Suite z roku 1956, historicky prvú skladbu generovanú počítačovým algoritmom a radu hudobne orientovaných programovacích jazykov Music - N vyvinutých Maxom Mathewsom. Mnohé z princípov vyžívaných v Pure Data a Max/MSP nachádzajú svoj pôvod v rôznych verziách jazykov Music a je to práve Max Mathewson, na koho počesť dostal Max/MSP svoje meno. Vo verziji Music III sa napríklad stretáme s princípom jednotkových

generátorov (tzv. ugens). Ide o ideu rozdelenia programu do jednotlivých častí syntetizátora na funkčné bloky, akými sú oscilátory, filtre, či generátory obálok. Prvé experimenty s počítačovým spracovaním zvuku v reálnom čase Miller Smith Puckette vytvára pomocou počítača PDP-11 na MIT, no prvú verziu programovacieho jazyka vyvinie až po pozvaní na ICRAM v roku 1985. Táto verzia ešte neobsahovala grafické rozhranie, to sa objavuje prvý krát až u jazyka Patcher vyvinutom o tri roky neskôr. Patcher implementoval všetky zásadné prvky a princípy, ktoré dnes poznáme v Pure Data. Puckette sa po dlhšej práci na zdokonaľovaní Maxu presúva do San Diega, kde začína pracovať na vývoji Pure Data. Jeho ambíciou bolo zdokonaľiť nedostatky, ktoré u Maxu videl a to hlavne uvedenie grafických dátových štruktúr schopných realizovať grafické hudobné partitúry a následne vytvorenie základných DSP objektov, ktoré by boli schopné syntézy zvuku priamo v počítači, na ktorom software beží. Toto bolo do daného momentu realizované pomocou externej DSP jednotky ISPW. Jednou z najzásadnejších zmien v Pure Data bolo prechod na Open Source charakter programu, čo dramaticky zmenilo priebeh vývoja jazyka. Nastalo obrovské rozšírenie jeho možností vďaka vývinu a následnom implementovaní užívateľských knižníc. Medzi najznámejšie patrí knižnica GEM (Graphic Environment for Multimedia) autora Marka Danksa z roku 1995 umožňujúca prácu s obrazovou a video zložkou audiovizuálneho diela. Pure Data je dnes časom overenou platformou s robustnou užívateľskou komunitou.

### 1.1.2 Príchute Pure Data

Vzhľadom na open source charakter, ktorý Pure Data na akademickej pôde v San Diegu nadobudlo, užívateľské knižnice a vstupy do fungovania prostredia sa stávali čoraz markantnejšími. To viedlo až k dnešnému rozdeleniu verzií na Pd Vanilla a Pd Extended. Pod názvom Vanilla chápeme pôvodnú verziu Pure Data tak, ako ju v deväťdesiatych rokoch navrhol sám Miller Smith Puckette. Názov Extended naopak označuje verziu, ktorá je obohatená o celú radu užívateľských knižníc a funkcií. Toto viedlo k vytvoreniu rady rôznych distribúcií, nazývaných flavors („príchute“). Za zmienku stoja nepochybne flavors ako Pd-CEAMMC, kedy sa jedná o špeciálna distribúcia Pure Data vyvinutá za účelom používania v Centre Elektroakustickej hudby Moskovského konzervatória, či PlugData, čo je VST plugin verzia Pure Data. V dnešnej dobe je medzi užívateľmi rozšírená prevažne Extended verzia, vďaka obrovskej rade užívateľmi vytvorených možností, ktoré v pôvodnej verzii chýbajú.

### 1.1.3 Grafické programovacie jazyky

Grafickým programovacím jazykom nazývame také programovacie jazyky, ktoré na vytváranie programu využívajú grafické prostriedky vo väčšej miere ako tie textové.

Samotný Pure Data je vizuálny programovací jazyk, čo znamená, že návrh a samotná realizácia programov neprebíha pomocou klasického zapisovania kódu do konzoly v IDE, ale vytváraním špeciálnych boxov a ich vzájomným spájaním. Tieto boxy sú používateľom rozmiestňované a následne spájané pomocou „patchovacích káblov“ (patch cords) v prostredí nazývanom plátno (canvas). Jednotlivé boxy môžu predstavovať rôzne funkcie od základných matematických operácií, skrz audio procesy (Reverb, Echo, Lowpass filter, FFT) , až po implementáciu externých funkcií vytvorených v C/C++, či kompletných externých Pure Data patchov. Tento spôsob programovania môže byť pre mnohých používateľov jednak intuitívnejším, vzhľadom na to , že Pure Data je určený najmä pre hudobníkov a umelcov, ktorí nie vždy majú predošlé technické vzdelanie a taktiež môže pripomínať „patchovanie“ modulárneho syntetizátora.

### **1.1.4 Základy programovacieho prostredia Pure Data**

Po spustení programu sa nám zobrazí základné okno systému. Toto obsahuje menu pre prácu s nastaveniami, užívateľskou optimalizáciou, prácou so zložkami a všetko ostatné, čo pre prácu s patchom nie je primárne dôležité. Ďalšou významnou časťou základného okna je konzola, na ktorú prebieha výpis z programu a poprípade chybové hlásenia. Samotné patchovanie prebieha v samostatnom okne nazývanom canvas („plátno“), kde dochádza ako už bolo viackrát spomínané k spájaným grafických prvkov (boxov), čím sa vytvára logická štruktúra programu.

#### **Základné typy boxov v Pure Data**

V Pure Data rozoznávame niekoľko základných typov boxov, ktorými sú: objekty, správy, čísla, GUI a komentáre. Tieto sú vytvárané vpísaním textu do objektového boxu, alebo u niektorých často používaných boxov pomocou klávesovej skratky.

#### **Objekty**

Objekty sú vytvorené napísaním textu do objektového boxu. Od boxov iného typu sú odlišiteľné tým, že majú tvar prostého obdĺžnika. Text objektu je rozdelený na takzvané atómy oddelené medzerou. Prvý atóm definuje o akú triedu objektu sa jedná (creation argument). Napríklad napísaním „osc 440“ definujeme objekt, ako oscilátor o frekvencii 440Hz. Trieda objektu taktiež určuje počet jeho vstupov a výstupov (u niektorých je tento počet však nastaviteľný). U objektu oscilátora môžeme vidieť zapísanú vlnovku (tilda), čo značí, že sa jedná o objekt spracúvajúci audio signál. Ďalším typom objektu sú objekty ovládania (control objects) pričom objekty ovládania vykonávajú svoju funkciu sporadicky a sú väčšinou spúšťané na základe

istej udalosti, ktorou môže byť napríklad MIDI správa Objekty pracujúce s audio signálom naopak pracujú kontinuálne za určenej vzorkovacej frekvencie.

## **Správy**

Základnou funkciou správy je odoslanie informácie v rámci patchu. Text napísaný do boxu tohto typu je interpretovaný ako správa, ktorá je zasielaná kedykoľvek je box aktivovaný (či už kliknutím myši v manipulačnom móde , alebo aktivovaním iného boxu). Na rozdiel od objektov, správy nie sú posielané rovno Pure Data, ale sú posielané na ich výstup a ďalej spracúvané v rámci objektov. Graficky ich možno odlišiť na základne lichobežníkového výrezu na pravej strane obdĺžnika.

## **Číslo**

Objekt čísla, ako názov naznačuje slúži na vkladanie a oznamovanie číselných informácií. Číselnú hodnotu je možné nastaviť v manipulačnom móde pomocou ťahu myši (ak držíme klávesu Shift, nastavujeme v desatinných číslach inak je hodnota nastavovaná v celých číslach). Graficky je číselný box znázornený ako obdĺžnik s useknutým pravým horným okrajom. V prostredí Pure Data dnes existuje i forma čísla s rozšírenými možnosťami nastavenia (number2), u ktorého je možnosť nastavenia logaritmického nárastu hodnôt pri ťahaní myši a podobne.

## **Symbol**

Symbol má prakticky totožnú funkciu ako číslo, no namiesto vkladania a oznamovania čísel pracuje so symbolmi a textovými reťazcami. Grafické znázornenie boxu symbol je totožné s tým u čísla.

## **GUI**

*Graphic User Interface*, teda grafické používateľské rozhranie. Tieto boxy slúžia na bezprostrednú komunikáciu medzi používateľom a Pure Data. Najčastejšie využívanými rozhraniami sú bang, kedy sa jedná o najjednoduchšiu formu tlačidla, ktorou spúšťame rôzne procesy, ďalej toggle prepínač, vhodný na zapínanie a vypínanie častí patchu, vslider/hslider reprezentujúci ťahový potenciometer potenciometrov, či klasický VU meter. GUI boxy nadobúdajú väčšinou svoj osobitnú grafickú podobu podľa ich praktického využitia.

## **Komentáre**

Ako v každom inom programovacom jazyku, i v Pure Data vzniká potreba komentovať program pre prehľadnosť, či už pri zdieľaní s inými používateľmi, alebo pri

navráteniu sa k vlastnému patchu po istom čase. Presne pre tieto prípady slúžia v Pure Data komentáre

## 1.2 Arduino

Jedná sa o jednu z najpopulárnejších open source elektronických platforiem, pochádzajúcej z dielne talianskych inžinierov. Arduino dosky sú založené na mikrokontroléroch ATmega spoločnosti Atmel. Samotný mikrokontrolér môžeme definovať ako monolitický integrovaný obvod obsahujúci všetky základné časti mikropočítača na jednom čipe (CPU, pamäte RAM a ROM, vstupné a výstupné zariadenia a iné rozhrania). Základným princípom fungovania Arduino mikrokontrolérov je prijímanie dát z vstupných periférií (senzory tepla, tlaku, enkodéry etc.) a ich následné spracovanie a prevedenie na výstupné periférie (servomotory, LED diódy, LCD obrazovky a podobne). Zoznam inštrukcií, ktoré má doska s danými dátami vykonať je programovaný vo vývojovom prostredí Arduino IDE založenom na jazyku Processing. Je nutné podotknúť, že platforma Arduino nevďačí za svoj úspech žiadnej revolučnej technickej inovácii, no revolúcii na poli marketingu a schopnosti priblížiť svet mikroprocesorov a návrhu elektronických systémov širokej verejnosti. To Arduino však ako vývojová platforma dokázalo i vďaka veľkej variabilite modulov a rozhraní, navrhnutých či už samotnými výrobcami a či členmi komunity.. Pomocou pár senzorov, ktorých cena je k tomu väčšinou triviálne nízka, je možné vytvoriť nespočetné množstvo interaktívnych projektov v rôznych odvetviach, ako je v našom prípade napríklad hudobná kompozícia.

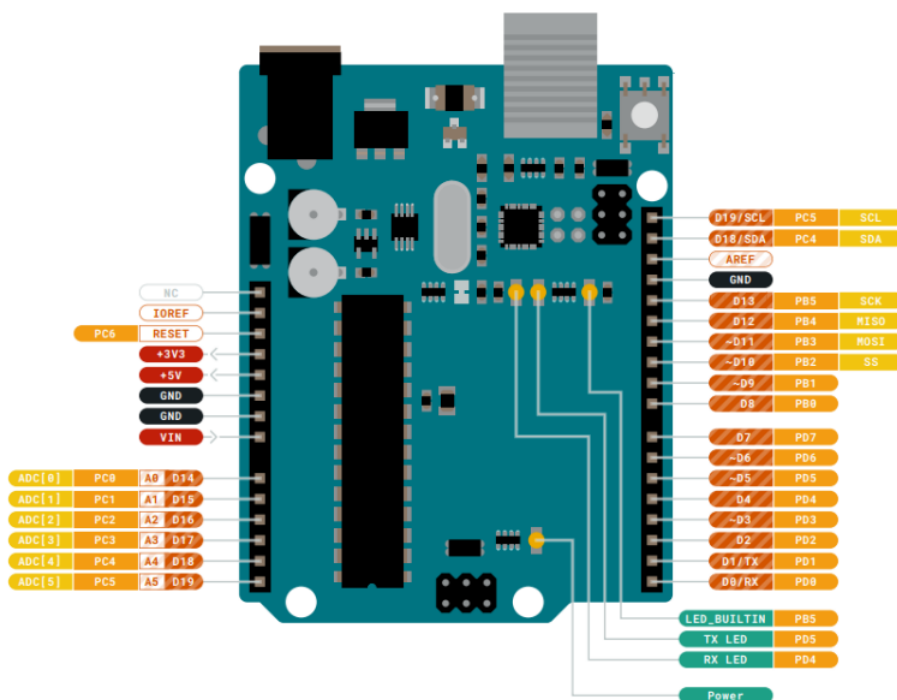
### 1.2.1 História a motivácia vzniku Arduina

Príbeh platformy Arduino má svoj počiatok v Taliansku na fakulte Interaction Design Institute Ivrea v diplomovej práci študenta Hernanda Barragána. Cieľom jeho práce bolo vytvoriť platformu, ktorá by uľahčila prácu s elektronikou umelcom, ktorý ju vo svojej tvorbe využívajú. Chcel tak minimalizovať potrebu technického vzdelania pre manipuláciu a prototypovanie systémov pomocou mikrokontroléru. Základné požiadavky pomocou, ktorých toto chcel dosiahnuť boli vytvorenie zrozumiteľného a kompatibilného IDE využívajúceho jednoduchý jazyk, kompletná integrácia nástrojov, bootloader pre jednoduché nahrávanie programov, serial-monitor na kontrolu odosielania dát mikroprocesoru a v neposlednej rade open source prístup k projektu s širokou radou užívateľských tutoriálov. Výsledkom tejto diplomovej práce bolo napokon zariadenie s názvom Wiring. Neskôr bola komercializovaná verzia platformy Wiring predávaná cez internet, no zlom nastal až v momente, kedy sa v roku 2005

z tímu odpojili Massimo Banzi a David Mellis a spoločne založili spoločnosť Arduino (pomenovaná podľa baru, kde sa zakladatelia stretávali). Platforma Arduino sa napokon dočkala celosvetového komerčného úspechu. V rokoch 2008 až 2014 prišlo kvôli interným sporom k rozdeleniu spoločnosti na dve odnože Arduino SRL a Arduino LLC.

## 1.2.2 Arduino UNO R3

Verzia UNO je takpovediac vlajkovou loďou spoločnosti Arduino a prvá verzia dosky s mikrokontrolérom bola firmou navrhnutá asi pred desiatimi rokmi. Dosku UNO R3 môžeme dnes považovať za jeden z najpoužívanejších a najlepšie zdokumentovaných mikrokontrolérov vôbec. Osadená procesorom ATmega328P, 14 vstupnými/výstupnými digitálnymi pinmi, 6 analógovými výstupmi, 16MHz keramickým rezonátorom, USB pripojením, ICSP headrom a resetovacím tlačidlom poskytuje univerzálny nástroj pre širokú radu projektov rôzneho charakteru.



Obr. 1.1: Pinout dosky Arduino UNO Rev. 3., oficiálny datasheet Arduino UNO Rev. 3

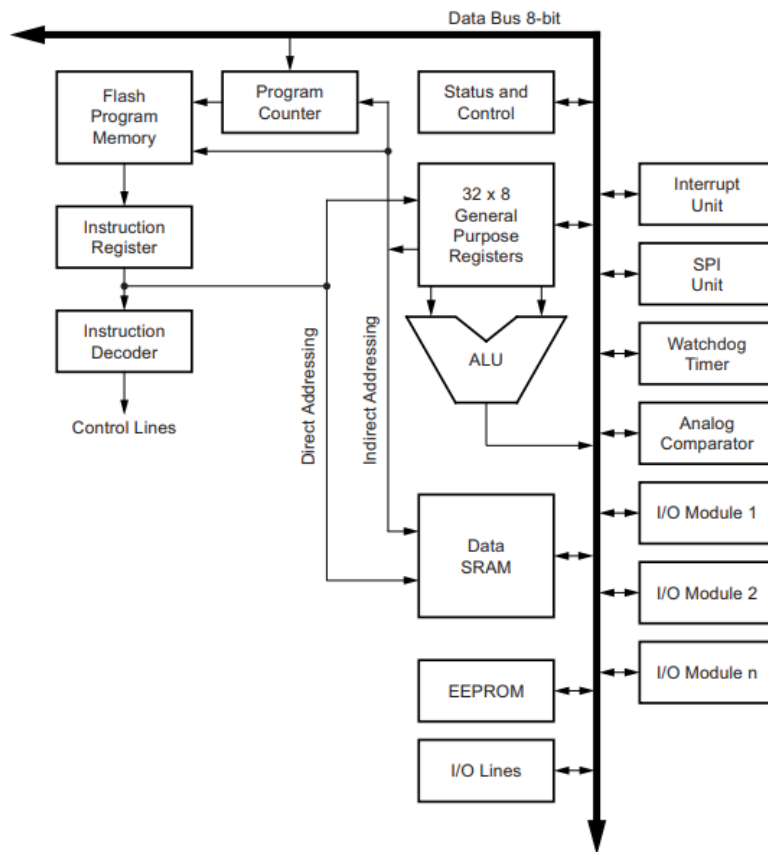


### 1.2.3 ATmega 328P

ATmega 328P patrí do rodiny procesorov AVR pôvodne vyrábaných firmou Atmel®(neskôr odkúpená firmou Microchip Technology). Jedná sa teda o 8-bitový nízko-odberový CMOS mikrokontrolér založený na vylepšenej AVR® RISC architektúre. Kombinuje 32 registrov so všeobecným účelom, pričom všetky sú priamo pripojené na aritmetickú logickú jednotku (ALU), čo umožňuje prístup ku dvom nezávislým registrom v rámci jedného hodinového cyklu.

#### Harvardská architektúra

Svoje meno získala vďaka tomu, že bola navrhnutá na Harvardsej univerzite a prvýkrát bola použitá pre elektrónkový počítač ENIAC (Electronic Numerical Integrater and Calculator). Za účelom optimalizovania výkonu procesory AVR využívajú práve túto architektúru, čo znamená oddelenie pamätí a zberníc pre program od tých, v ktorých sú uložené dáta. Toto umožňuje vykonanie väčšieho množstva inštrukcií za menší počet hodinových cyklov a to vďaka takzvanému paralelizmu vykonávania inštrukcií, kedy nová inštrukcia môže byť pridelená už v cykle, v ktorom je predošlá inštrukcia vykonávaná a nemusí čakať na jej dokončenie.



Obr. 1.2: Blokový diagrama architektúry AVR procesoru, oficiálny datasheet procesoru ATMEGA328P

## RISC

Z anglického *Reduce Instructions Set Computer*, je typ architektúry mikroprocesorov s redukovaným súborom inštrukcií. Mikroprocesory implementujúce túto architektúru využívajú jednoduchšie inštrukcie, pričom väčšina inštrukcií má rovnakú dĺžku a sú vyvolávané v jednom inštrukčnom cykle. Toto má za dôsledok celkové zrýchlenie chodu mikroprocesora spolu so zmenšením jeho rozmerov a znížením nákladov na jeho výrobu na rozdiel od pôvodnej CISC (Complex Instructions Set Computer) architektúry.

### 1.2.4 Zbernice

Embedded zariadenia pre svoju komunikáciu využívajú vo veľkej miere zbernice. Mikrokontroléry spoločnosti Atmel využívajú zbernice SPI a I<sup>2</sup>C, ktoré patria k najrozšírenejším typom zberníc v aplikáciách mimo priemysel. V realizácii hardware modulu pre Pure Data, ktorý je výstupom tejto práce využívame obe zbernice a

preto sa v nasledujúcej kapitole budeme venovať ich vzájomným rozdielom a charakteristikám

## Zbernica SPI

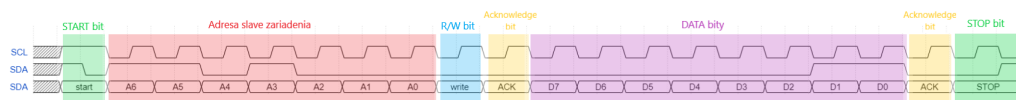
Z anglického serial peripheral interface je zbernica vhodná na vysokorýchlostnú synchrónnu komunikáciu na krátku vzdialenosť. Komunikácia prebieha pomocou štyroch vodičov. Vodič SCLK (serial clock) vysiela master zariadenie hodinový signál všetkým ostatným slave zariadeniam. MOSI (Master-out/Slave-in) vysiela dáta z master zariadenia do slave zariadenia. MISO (Master-in/Slave-out) slúži naopak na vysielanie dát z slave zariadenia do master zariadenia. SS (slave select) slúži na výber konkrétneho slave zariadenia (zariadenie je aktívne pri nízkej úrovni signálu). Zbernica SPI podporuje Full duplex a to až do rýchlosti 10 MHz. Na rozdiel od I<sup>2</sup>C má nižšie napäťové nároky a slave zariadenia nevyžadujú svoje unikátne adresy. Slave zariadenia taktiež môžu komunikovať iba s master zariadením a ich vzájomná komunikácia nie je možná. Výraznou nevýhodou zbernice SPI je fakt, že u jednotlivých embedded zariadenia môže byť implementovaná rozdielne.

## Zbernica I<sup>2</sup>C

*Inter-itergrated Circuit*, v preklade vnútorná integrovaná zbernica, je sériový protokol navrhnutý spoločnosťou Philips Semiconductor (dnešné NXP semiconductors) v roku 1982. Využívajúci prepojenie dvoch obojsmerných dátových vodičov. Vodič SCL (serial clock) sprostredkúva hodinový signál naprieč zariadeniami a vodič SDA (serial data) súži na prenášanie dát medzi nimi. Používaná je predovšetkým na prepojenie nízko-rýchlostných zariadení akými sú mikrokontroléry, EEPROM pamäte, D/A a A/D prevodníky etc. Na rozdiel od protokolu SPI má každé slave zariadenie svoju jedinečnú adresu nastavovanú pomocou siedmich bitov, čo umožňuje zapojenie až 128 slave zariadení. Okrem možnosti množstva pridelených adries je obmedzením v počte pripojených zariadení taktiež maximálna kapacita na zbernici 400pF. Komunikačná rýchlosť zbernice sa pohybuje až do hodnoty 5MHz (bežne používaná je však rýchlosť 100kHz, teda vidíme, že komunikácia pomocou I<sup>2</sup>C zbernice je výrazne pomalšia, ako komunikácia skrz SPI). Výrazným rozdielom od zbernice SPI je možnosť zapojenia viacerých master zariadení na zbernicu. Kanály I<sup>2</sup>C protokolu sú z elektrického hľadiska zapojené ako open drain (vodiče s otvoreným kolektorom), čo znamená že každý vodič musí mať na kladné napätie pripojený pull-up rezistor ktorý zabezpečuje vysokú úroveň signálu v stave pokoja. Typickým využitím I<sup>2</sup>C zbernice je ovládanie LED obrazoviek pomocou Arduina, či ako v našom prípade slúži na komunikáciu s gyroskopickou jednotkou a rozšírenie GPIO pinov pomocou čipu s posuvným registrom

## Časový diagram signálu I<sup>2</sup>C zbernice

Komunikáciu medzi zariadeniami iniciuje vždy master zariadenie vyslaním START bitu a prenos dát končí v momente vyslania STOP bitu. V časovom rozmedzí týchto dvoch bitov je zbernica považovaná za zaneprázdnenú a žiadne zo zvyšných zariadení na nej pripojených sa nepokúša komunikáciu prerušiť. Komunikácia je zabezpečená pomocou dvoch signálov SCL a SDA a je pravidlom, že signál SDA sa môže meniť iba ak je signál SCL na nízkej logickej úrovni. Výnimka nastáva práve pri vysielaní START a STOP bitov. Po spustení komunikácie pomocou START podmienky nasleduje adresný paket obsahujúci sedem adresných bitov (od LSB po MSB), za ktorými ihneď nasleduje READ/WRITE bit, ktorý rozhoduje, či sa jedná o čítanie, alebo zapisovanie dát a ACK/NACK bit, ktorý odosiela master zariadeniu informáciu o úspešnosti operácia. Druhým vysielaným paketom je paket dátový, ktorý obsahuje osem dátových bitov a ACK/NACK bit. Komunikácia je ukončená STOP bitom.



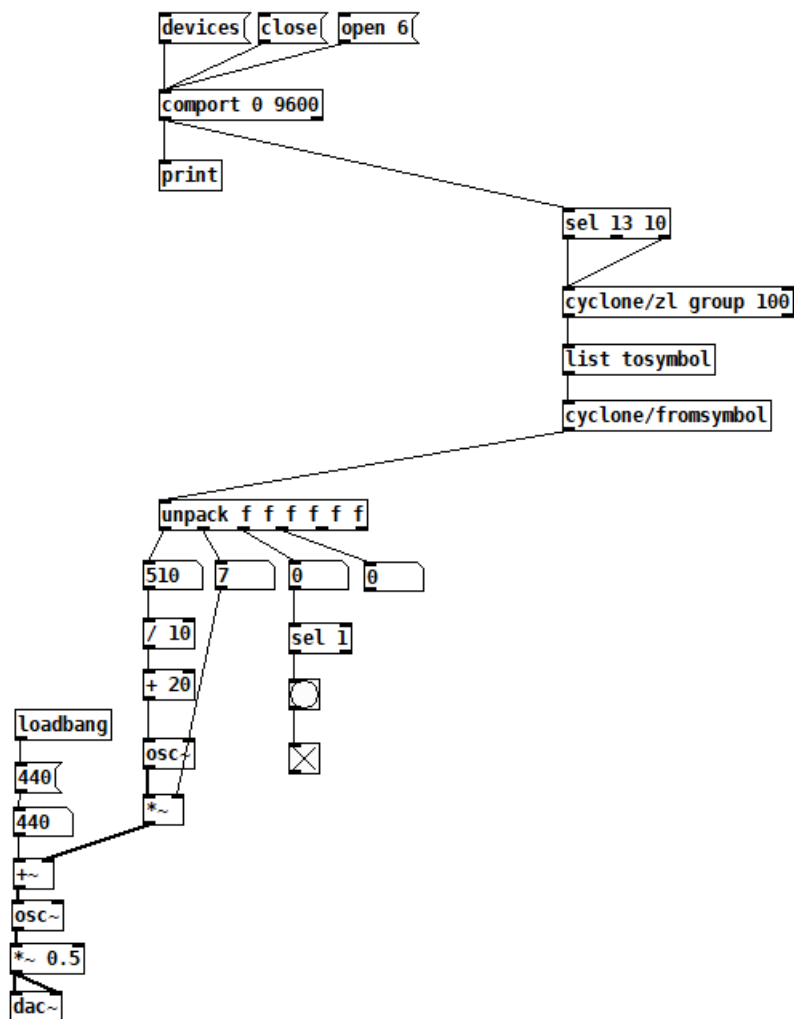
Obr. 1.3: Časový diagram signálu zbernice I<sup>2</sup>C, vytvorený pomocou online nástroja [wavedorm.com](http://wavedorm.com)

## 2 Prvotné testovanie rozhraní

Tento proces prebiehal v prvej fáze návrhu, kedy boli rozličné periférie testované pomocou testovacieho poľa a externe prepájané s doskou s mikrokontrolérom. Platforma ponúka celú radu externých modulov osadených všetkými potrebnými obvodmi, ktoré stačí pripojiť k zariadeniu a sú plne funkčné. Dôležitou časťou tejto fáze je objavovanie možností jednotlivých periférií v ich neskoršom konečnom využití. Toto objavovanie možností naberá na dôležitosti hlavne u periférií nie bežne využívaných v hudobnej elektronike, akými sú napríklad gyroskop s akcelerometrom, či ultrazvukový merač vzdialenosti. Vďaka rozsiahlej komunite zaoberajúcej sa platformou Arduino nevzniká takmer žiaden problém s riešením testovacieho pripojenia periférií. Na internete je k nájdeniu široká rada doporučení a návodov obsahujúcich schémy zapojení a testovacie kódy spolu s užívateľskými knižnicami. Poslednou nie menej významnou časťou prvotného testovania bolo vytvorenie jednoduchých testovacích patchov v prostredí Pure Data. Účelom týchto patchov nie je vytvoriť funkčný celok dávajúci akýkoľvek zmysel v CAC a CMG (computer aided composition teda počítačom podporovaná hudobná kompozícia a computer generated music teda počítačom generovaná hudba), no otestovať schopnosť pripojenia periférií k prostrediu Pure Data a otestovať schopnosť ovládania jednoduchých objektov. Súčasťou tejto fáze bolo taktiež fyzické meranie rozmerov niektorých súčiastok za účelom vytvorenia knižnice v návrhovom prostredí Eagle.. Z tohto dôvodu sú u niektorých z rozhraní uvedené hodnoty rozmerov.

### 2.1 Pure Data testovací patch pre vstupné periférie

Pre fázu prvotného testovania bolo nutné vytvoriť dva testovacie patche (resp. dva druhy testovacích patchov s drobnými obmenami, pre konkrétne potreby testovania) a to patch pre vstupné periférie a patch pre výstupné periférie. Funkcionalita vytvoreného testovacieho patchu je jednoduchá, a spočíva v podstate na prijímaní dát zo sériovej linky mikrokontroléru a ich jednoduchého vypisovania do číselných boxov v Pure Data. Daný patch je uvedený na obrázku nižšie spolu s podrobnosťami o princípoch, na ktorých je založený.



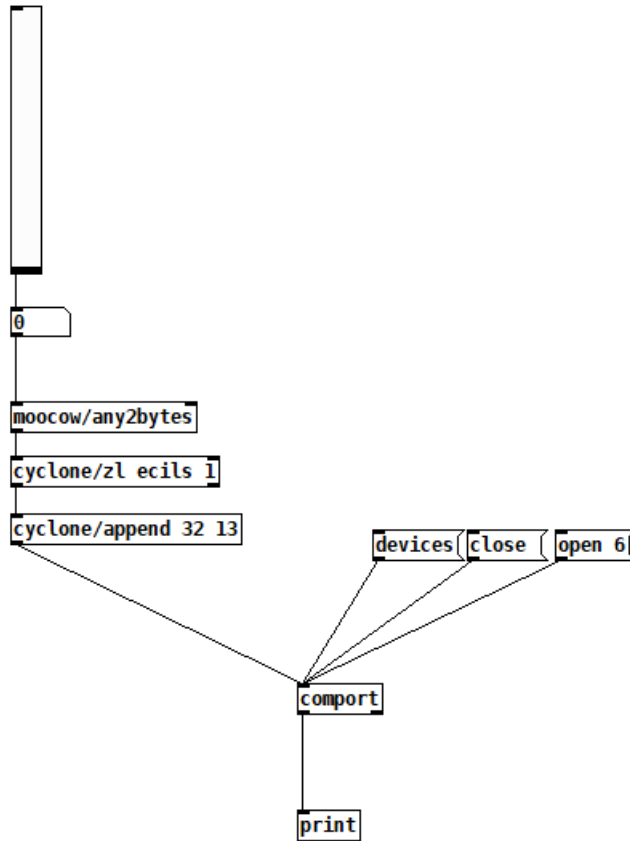
Obr. 2.1: Testovací patch v prostředí Pure Data pro vstupné zariadenia.

Ako možno vidieť na obrázku, prvým vstupným objektom patchu je objekt `comport`, pomocou ktorého je zahajovaná a sprostredkovaná komunikácia po sériovej linke mikrokontroléru. Na tento objekt sú posielané tri správy a to `devices`, po ktorého stlačení je vypísaná do konzoly správa o zariadeniach pripojených k sériovej linke, `open 6`, ktorý otvára šiesty kanál sériovej linky, na ktorej je náš konkrétny mikrokontrolér pripojený a správa `close`, ktorá vysiela pokyn na ukončenie komunikácie. Posledné dva príkazy sa stali v rámci testovania praktickou pomôckou, keďže neexistuje možnosť prijímať dáta zo sériovej linky do Pure Data a zároveň mať

spustený výpis na monitor sériového portu. Objekt `comport` odosiela dáta na objekt `print`, ktorý sa stará informačný výpis na konzolu a na objekt `sel 10 13`. `Sel` je skratkou pre objekt `select`, ktorý porovnáva hodnoty na svojom vstupe s hodnotami v jeho argumente. Ak sa hodnota čísla v jeho argumente nerovná vstupnej hodnote, kopíruje ju na svoj pravý výstup, no ak sa hodnota vstupnej veličiny rovná argumentu vysiela signál `bang` (signál slúžiaci ako trigger udalostí). Význam tohto objektu v našom patchy spočíva v tom, že objekt `comport` posielala dáta v ASCII formáte, kde sú správy *line feed* (koniec riadku) a *carrige return* (návrat na začiatok riadku) reprezentované práve číslami 10 a 13. To znamená, že všetky užitočné dáta sú v nezmenenej forme posielané ďalej pravým výstupom a v prípade ukončenia riadku je nasledujúcemu objektu vyslaná správa *bang*. Týmto je zabezpečená možnosť pripojenia viacerých vstupných periférií bez konfliktu na výstupe. Skupina troch nasledujúcich objektov zabezpečuje pretypovanie ASCII formátu na číselnú informáciu. Pomocou objektu `unpack` je nakoniec séria dát rozdelená na určitý počet výstupov (tento počet závisí na argumente, písmeno `f` v našom prípade značí dátový typ *floating point*) smerujúcich na výpis do *umber boxov*. Na konci patchu môžeme vidieť dve číselné hodnoty používané na modulovanie jednoduchého oscilátora o frekvencii 440Hz a tretí číselný box, pomocou ktorého je *triggrovaný* signál *bang* pripojený na *toggle switch* (prepínač) slúžiaci na testovanie tlačidiel a prepínačov.

## 2.2 Pure Data testovací patch pre výstupné periférie

Druhý z *patchov* nutných v prvotnej testovacej fáze je patch pre výstupné periférie. Jeho základnou funkcionalitou je vysielať dáta na vstup sériovej linky, ktorými sú následne dané periférie ovládané.



Obr. 2.2: Testovací patch v prostredí Pure Data pre vstupné zariadenia.

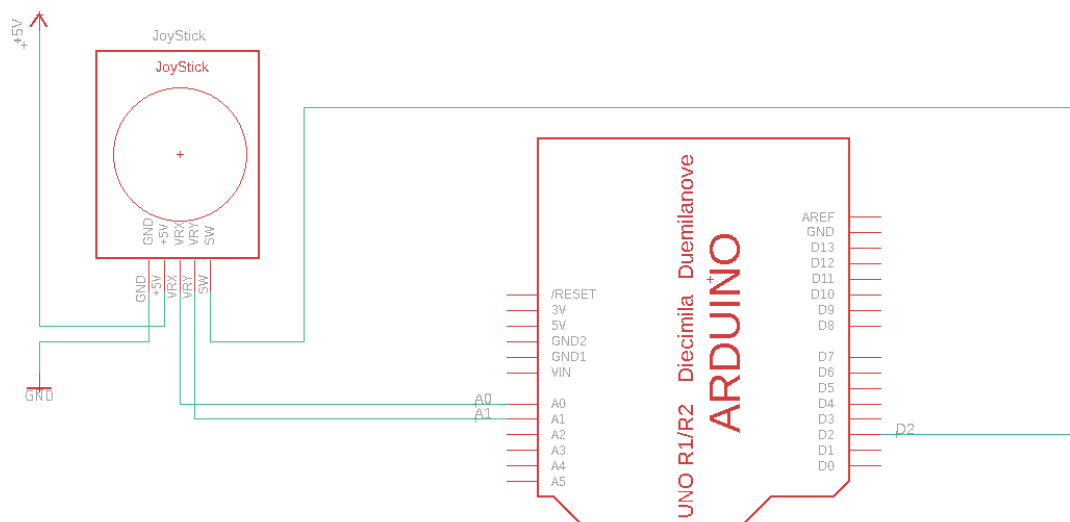
Prvým objektom nášho patchu je *vslider* (skratka pre *vertical slider*, teda vertikálny posuvný ovládač), s pomocou ktorého nastavujeme hodnoty na vstupe sériovej linky. Výstup *vslideru* smeruje do objektu *any2bytes* z užívateľskej knižnice *moocow* slúžiacim na pretypovanie ľubovoľných vstupných dát do *byte* formátu. Toto pretypovanie je nutné pre neskoršie spracovanie dát mikrokontrolérom. Nasledujúci objekt s názvom *zl ecils 1* rozdeľuje list obsahujúci dáta na pozícii číslo jedna. Posledný objekt využívaný na posielanie dát do objektu *comport* je objekt *append 32 13*, ktorý vkladá medzeru a príkaz *line feed* v ASCII formáte. Samotný objekt *comport*, funguje na totožných princípoch ako v patchy pre vstupné zariadenia, no v tomto prípade slúži na prijímanie dát z Pure Data a ich následné posielanie na vstup sériovej linky mikrokontroléru.

### 2.2.1 Testovanie modulu PS2 Joy Stick pre Arduino

Joy stick v audio a hudobnej elektronike môžeme často nájsť napríklad na MIDI kontroléroch, alebo syntetizátoroch ako ovládanie hĺbky modulácie v jednej ose a



ovládanie ohýbania tónu (pitchbend) na ose druhej. Joy stick týmto spôsobom využívajú napríklad MIDI klávesy AKAI – MPK mini MK3 MIDI, či ich variant s rozšíreným množstvom kláves AKAI – MPK mini PLUS. PS2 Joy Stick je prstom ovládaný modul pozostávajúci z dvoch potenciometrov o veľkosti 10 k $\Omega$  a jedného tlačidla. Potenciometre indikujú súradnice smeru do ktorého je páčka vychýlená a tlačidlo vysiela low signál, ak je stlačené. V stave pokoja, keď nie je s Joy Stickom manipulované, nadobúda výstupné napätie X i Y hodnotu 2,5V a následným pohybom meníme rozsah odporu potenciometrov v rozsahu 0  $\Omega$  až 10 k $\Omega$ , čo mení výstupné napätie modulu v rozmedzí 0 V až 5 V podľa smeru výchylky. Toto napätie je následne Arduinom interpretované ako celočíselná hodnota pohybujúca sa medzi hodnotami 0 až 1023. Výstupom modulu je päť prepojovacích pinov +5V pripájaný na napájacie napätie, GND pripájaný na zem, VRx a VRy výstupné piny potenciometrov pripájané na analógové vstupy a SW výstup tlačidla pripojený na digitálny vstup Arduino dosky. Modul bol testovaný pripojením na testovacie pole v zapojení, ktoré je možno vidieť na obrázku nižšie. Pretože modul PS2 Joy Stick nie je súčasťou žiadnej z dostupných knižníc návrhového prostredia Eagle, bolo nutné odmerať jednotlivé rozmery modulu a vytvoriť vlastnú súčiastku v našej užívateľskej knižnici. Na otestovanie bol použitý kód prístupný v návode k súčiastke, ktorý sme však museli pre potrebu ovládania testovacieho patchu v Pure Data jemne pozmeniť. Vo svojej podstate sa jedná o jednoduchý kód, vypisujúci na monitor sériového portu informácie z dvoch analógových vstupov a jedného tlačidla. Problém však nastáva u tlačidla, ktoré je active low, čo znamená že ak nie je stlačené vysiela do Pure Data konštantne logickú 1a vytvára tak permanentný rýchly sled *bang* signálov v patchy, čo je nežiadúce. Tento problém sme vyriešili jednoduchou podmienkou *if* v kóde ovládajúcim mikrokontrolér, ktorý zabezpečuje vypísanie logickej jednotky na monitor sériového portu v prípade, že je vysielať low signál, inak v stave pokoja konštantne vypisuje logickú nulu. V Pure Data *patchy* sme problém s nestálymi sériami signálov *bang* vyriešili pomocou objektu sel 1 (možno vidieť v testovacom *patchy* pre vstupné periférie). V každom ďalšom prípade využívania tlačidiel ošetríme túto situáciu obdobne.

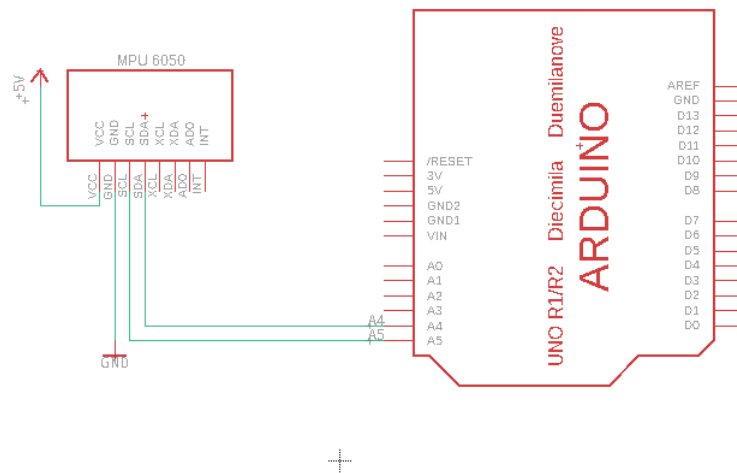


Obr. 2.3: Pinout PS2 Joy Stick, vytvorené pomocou návrhového prostredia Eagle

## 2.2.2 Testovanie modulu s akcelerometrom a gyroskopom MPU 6050

Akcelerometer a gyroskop sú senzory, o ktorých môžeme takmer s istotou tvrdiť, že sú v audiotechnike využívané minimálne, no ich potenciál je obrovský. Poloha v priestore a rýchlosť sú v čase pomerne rýchlo meniace sa veličiny a preto sú vhodné na ovládanie parametrov ako je výška tónu, tempo, medzný kmitočet kmitočtového filtra, či hĺbka modulácie. Ako prvý prístroj snažiaci sa zhudobniť práve tieto veličiny môžeme považovať teremín, no z podstaty veci veľa klasických hudobných nástrojov ich nepriamo využíva tiež. Vezmime si napríklad polohu prstov na hmatníku strunných nástrojov, či pohyby znižca pri hre na trombón. Z modulov kompatibilných s Arduino doskou obsahujúcich kombináciu gyroskopu a akcelerometru, bol vybraný práve model MPU 6050. Ostatné dostupné moduly obsahujú na doske i iné senzory, akými sú napríklad magnetometre, čo niekoľkonásobne navyšuje ich cenu. V tomto module je kombinovaný trojosý gyroskop a akcelerometer na rovnakom čipe spolu s DMP (Digital Motion Processor) kombinujúci dáta z obidvoch senzorov, čo značne zjednodušuje ich ďalšie spracovanie. Modul na komunikáciu s Arduino doskou využíva I<sup>2</sup>C multi-masterovú zbernicu. Výstup zariadenia je realizovaný pomocou ôsmich výstupných pinov pričom v našej aplikácii sú využívané len štyri. Pin VCC je pripájaný na napájacie napätie o veľkosti 5V, GND na zem, SCL na analógový vstup A5, SDA na analógový vstup A4. Pomocou pinov SCL a SDA je realizovaná komunikácia po I<sup>2</sup>C zbernici. Modul bol testovaný pripojením na testovacie pole v

zapojení, ktoré je možno vidieť na obrázku nižšie. Modul MPU6050 taktiež nie je obsiahnutý v žiadnej z dostupných knižníc pre návrhové prostredie Eagle, takže bolo opäť nutné zmerať jeho rozmery a navrhnuť vlastnú súčiastku do užívateľskej knižnice. Na otestovanie bol použitý kód prístupný v návode k súčiastke, ktorý bol iba nepatrne upravený. V pôvodnom kóde bola vypisovaná na monitor sériového portu taktiež informácia o teplote, ktorú bolo nutné odstrániť. V *patchy* na testovanie vstupných periférií bolo nutné zmeniť hodnotu baud rate (modulačná rýchlosť) na 115200 Bd, ktorou modul komunikuje. Toto je dosiahnuté prostým prepísaním hodnoty argumentu v objekte `comport`. Týmto však vzniká problém v budúcom používaní zariadenia, kedy nebude schopné používať modul MPU6050 v rovnaký moment so zvyšnými zariadeniami, ktoré komunikujú modulačnou rýchlosťou 9600Bd.

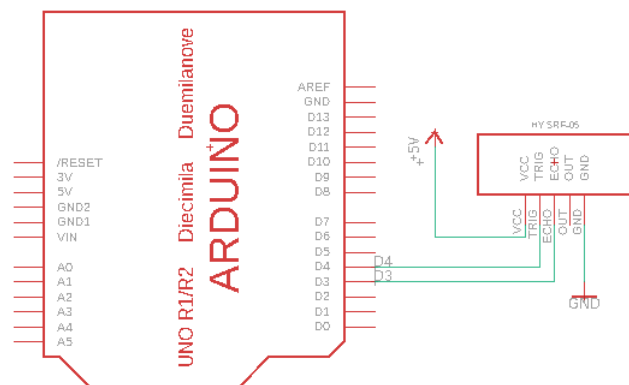


Obr. 2.4: Pinout MPU6050, Vytvorené pomocou návrhového prostredia Eagle

### 2.2.3 Testovanie Ultrazvukového senzoru HY-SRF 05

Ultrazvukový senzor je z kategórie modulov bežne nevyžívaných v audiotechnike určite ten najnezvyčajnejší, no môže priniesť do procesu CAC/CGM zaujímavý experimentálny prístup. Rovnako ako akcelerometer, môže byť tento modul využitý podobným spôsobom, ako modul MPU 6050 na simuláciu nástrojov podobných termínu. Pri voľbe konkrétneho modulu využívaného v module padla voľba na modul HY-SRF 05, ktorý na rozdiel svojho predchodcu modelu HY-SRF 04 disponuje väčším meracím rozsahom a väčšou presnosťou merania. Samotný modul funguje na princípe vysielania ultrazvukových pulzov o frekvencii 40 kHz a ich následnej detekcii. Vzdialenosť objektu je ďalej vypočítaná z rozdielu času medzi vysielanou a odrazenou vlnou. Senzor je schopný detegovať objekty v rozmedzí vzdialeností 2 cm až 4 m pod uhlom 15 stupňov a to s presnosťou až 3 mm (pri detekcii objektov vo

vzdialenosti väčšej ako 3m presnosť spravidla klesá). Napájacie napätie modulu je 4,5 V až 5,5 V a prúdový odber v stave pokoja 2 mA, pričom v špičkových hodnotách pri fungovaní modulu môže dosahovať až 40 mA. Výstup modulu je tvorený piatimi pinmi, no je možné ho použiť v štvorpinovom zapojení. Čo sa týka zapojenia Vcc pin je už tradične pripojovaný na napájacie napätie 5 V, GND na zem a ECHO a TRIG sú zapojené na ľubovoľné digitálne vstupy. V štvorpinovom spôsobe zapojenia pin OUT zostáva nevyužitým. Modul bol testovaný pripojením na testovacie pole v zapojení, ktoré je možno vidieť na obrázku nižšie. Modul ultrazvukového senzoru HY-SRF 05 taktiež nie je súčasťou žiadnej z knižníc dostupných pre návrhové prostredie Eagle, takže bolo nutné vytvoriť vlastnú súčiastku v užívateľskej knižnici. Na otestovanie bol použitý kód prístupný v návode k súčiastke, ktorý jednoducho vypisuje na Serial monitor údaje rozdielu času medzi vysielanou a dopadajúcou vlnou. Pomocou jednoduchej zmeny v kóde je možné však tento čas prepočítať na vzdialenosť v cm. Pure Data *patch* na testovanie vstupných periférií sme pre tento prípad upravovať nemuseli. Pri opakovanom testovaní si je však možné všimnúť, že modul ultrazvukového senzoru HY-SRF 05 neponúka stále a presné informácie o vzdialenosti, preto je jasné, že pri jeho praktick

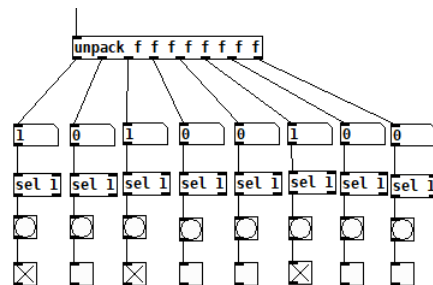


Obr. 2.5: Pinout HY-SRF05, vytvorené pomocou návrhového prostredia Eagle

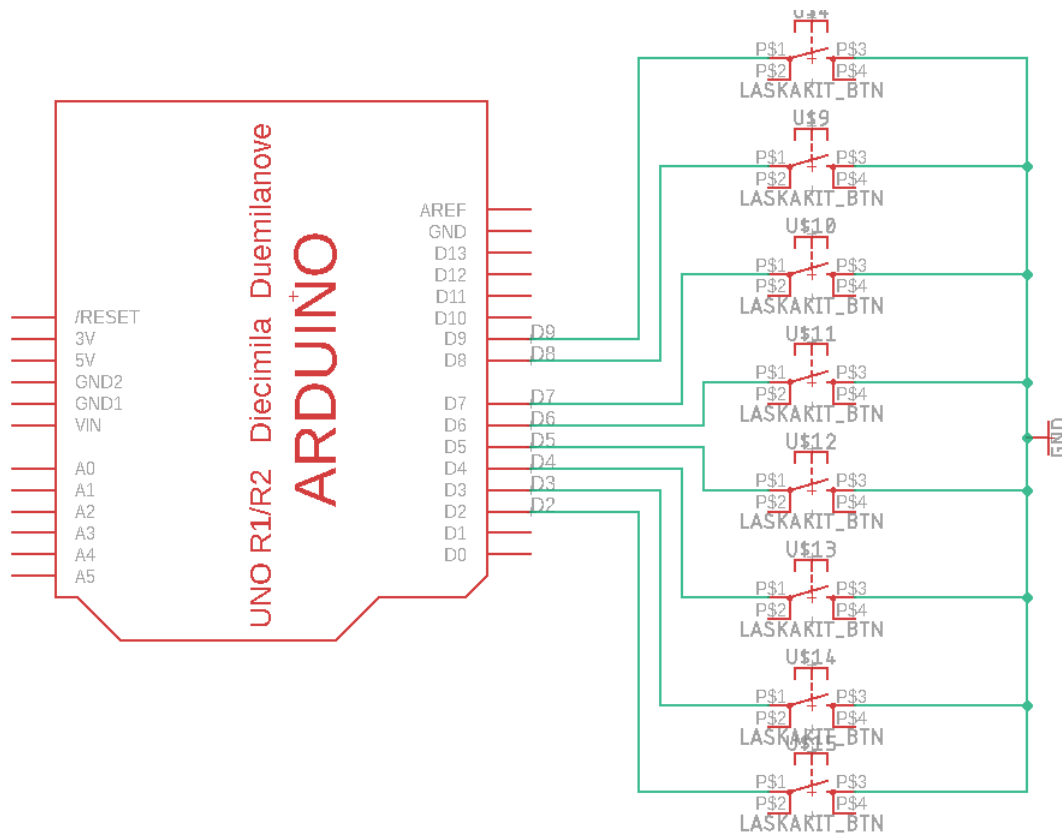
## 2.2.4 Testovanie tlačidiel

Tlačidlá sú klasicky využívaným rozhraním nie len v audioteknike, ale prakticky v každom elektronickom zariadení. Možnosti ich využitia sú skoro neobmedzené, môžeme nimi prepínať parametre, zapínať a vypínať časti *patchu* v Pure Data a mnoho iného. Na testovanie funkcionality tlačidiel bolo využitých osem jednotlivých tlačidiel separátne pripojených k digitálnym pinom dosky s mikrokontrolérom. Na testovanie tlačidiel bol použitý jednoduchý kód, detegujúci nízku logickú úroveň na

vstupe digitálnych pinov. Mikrokontrolér ATmega328P obsahuje vbudované pull-up rezistory, takže nevznikla potreba pripojenia externých rezistorov. V Pure Data patchy pre testovanie vstupných zariadení bola potreba úpravy za účelom testovania ôsmich tlačidiel naraz. Táto úprava spočíva v pripojení ôsmich objektov *sel 1* na objekt *bang* a *toggle*, pre prevenciu opakovaného odosielania série signálov *bang* (rovnaká štruktúra, ako na štvrtom, výstupe objektu *unpack*). Je zrejmé, že počet pinov potrebných na pripojenie ôsmich tlačidiel je privysoký vzhľadom na obmedzené možnosti využitia tlačidiel a eliminuje tak počet externých zariadení s digitálnym rozhraním, ktoré možno používať s tlačidlami naraz. Spôsob akým je výhodné tento počet eliminovať v našej realizácii musíme teda neskôr vyriešiť.



Obr. 2.6: Úprava v Pure Data patchy pre testovanie vstupných periférií, pre účely testovania tlačidiel

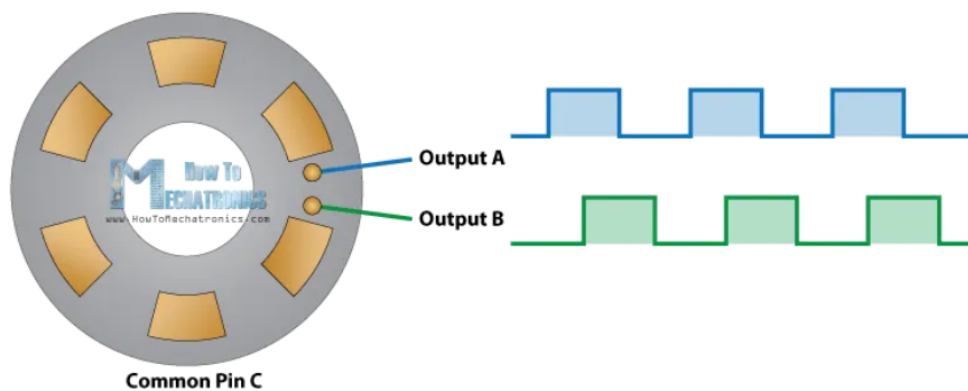


Obr. 2.7: Pinou tlačidiel , schéma vytvorené pomocou návrhového prostredia Eagle

## 2.2.5 Testovanie rotačného enkodéra KY-040

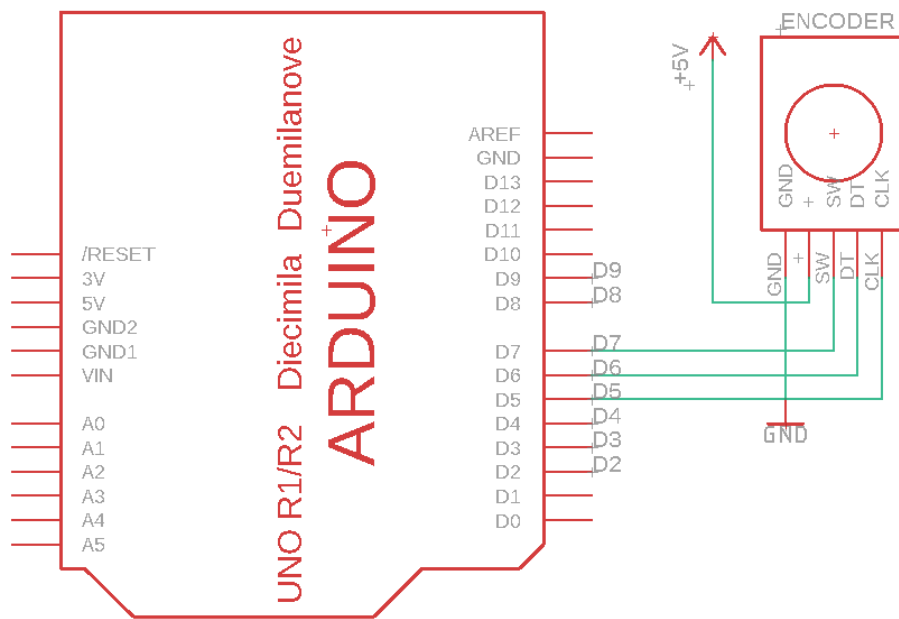
Rotačný enkodér je typ pozičného senzoru, používaný na detekciu pohybu na základe zmeny uhlu pri jeho otáčaní. Jeho veľkou výhodou oproti bežnému rotačnému potenciometru založenom na princípe zmeny odporu je fakt, že jeho rozsah nie je obmedzený na 360 stupňov, ale je možné ho otáčať v podstate neobmedzene. Toto z neho robí rozsiahlo používané rozhranie nie len v audiotechne. Typickým využitím rotačného enkodéra je jeho využitie na orientáciu v menu zariadenia. Samotný princíp fungovania rotačného enkodéra je realizovaný vďaka dômyselnému systému založenom na rotačnom disku obsahujúcom rovnomerne rozložené kontaktné plochy pripojené na spoločný pin C a dva separátne piny A a B. Enkodér generuje dva obdĺžnikové signály nazývané clock (pin CLK) a data (pin DT), ktoré sú generované vždy, keď je uzavretý kontakt s kontaktnou plochou. To znamená, že na samotnú detekciu rotácie enkodéra stačí, jeden zo signálov, no v praktickom využití skoro vždy potrebujeme určiť i smer rotácie. Princíp na ktorom, je toto určovanie založené vychádza z faktu, že pri otáčaní senzoru v smere hodinových ručičiek predbieha clock

signál data signál o fázu 90 stupňov a naopak. Čip ATmega328P, ktorým je doska Arduino UNO osadená následne túto zmenu fáze deteguje a podľa toho rozhoduje o inkrementácii, či dekrementácii premennej.



Obr. 2.8: Princíp fungovania rotačného enkodéru, <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

Modul bol otestovaný pripojením na testovacie pole v zapojení, ktoré je možno vidieť na obrázku nižšie. Na otestovanie bol použitý kód prístupný v návode k súčiastke, ktorý jednoducho vypisuje na Serial monitor údaje o inkrementácii, či dekrementácii hodnoty, pričom nebolo potrebné v ňom nič meniť. V testovacom Pure Data *patchy* pre vstupné zariadenia taktiež nebolo potreba vykonávať žiadnu zmenu.



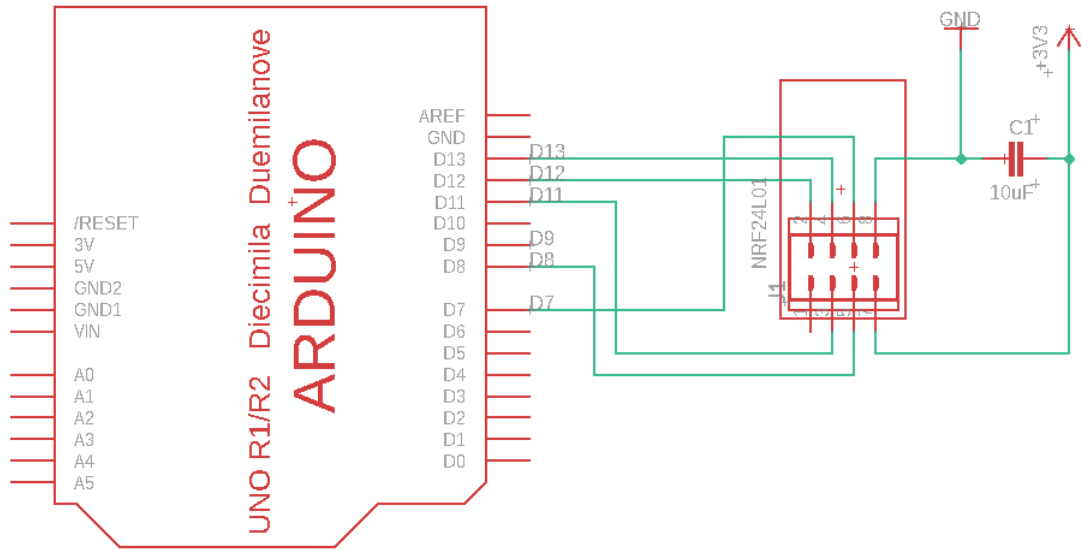
Obr. 2.9: Pinout enkodéru KY-040, vytvorené pomocou návrhového prostredia EAGLE

## 2.2.6 Testovanie Wifi modulu nRFL01

Jedná sa o modul slúžiaci na bezdrôtovú komunikáciu medzi mikrokontrolérmi. Slovo wifi v jeho názve naznačuje pracovnú frekvenciu 2,4GHz, ktorá je typicky využívaná na prepájanie počítačov a mobilných zariadení s routermi, no pripojenie Arduino dosky k internetovej sieti pomocou neho nie je možné. Veľmi podstatným faktom je, že modul nRFL01 je napájaný napätím 3,3 V a napätie 5 V používané na napájanie väčšiny bežne využívaných modulov by ho mohlo závažne poškodiť. Z hľadiska prúdového odberu modul v powerdown režime odoberá iba 1 mA, no v špičkách pri vysielaní a prijímaní dát jeho odber môže dosiahnuť stovky mA a preto je potrebné medzi napätie 3,3V a zem pripojiť kondenzátor 10 µF. Wifi modul nRFL01 je jediným modulom využívajúcim na svoje pripojenie k mikrokontroléru zbernicu SPI, v realizácii modulu pre interaktívne projekty v Pure Data a je jeho pinout pevne daný s čím treba počítať pri jeho využívaní s ostatnými perifériami. Výraznou zmenou v porovnaní so zvyškom modulov je ten, že na jeho testovanie bolo nutné požitie dvoch dosiek s mikrokontrolérom Arduino a to jednej, ktorá využíva Wifi modul nRFL01 vo funkcii vysieláča a druhej, ktorá ho používa vo funkcii prijímača. Kód obsiahnutý v návode k modulu nestačil na požiadavky testovania, keďže jeho funkcionálnosť spočíva vo vypísaní potvrdzovacej správy na monitore sériového portu oboch



dosiek Arduino. Aby bolo možné vysielat užitočné dáta na prácu s patchom v Pure Data, boli nutné skombinovať daný kód s kódom iného modulu a pokúsiť sa tak bezdrôtovo prenášať z vysielacieho mikrokontroléru na ten prijímací. Na tento účel bol využitý modul s PS2 Joy Stickom, keďže v sebe obsahuje rovno dva analógové senzory a tlačidlo zároveň. V samotnom testovacom *patchy* pre vstupné periférie nebolo potrebné meniť nič a využívaný bol totožnej forme, v akej bol využívaný na testovanie modul s PS2 Joy Stickom.

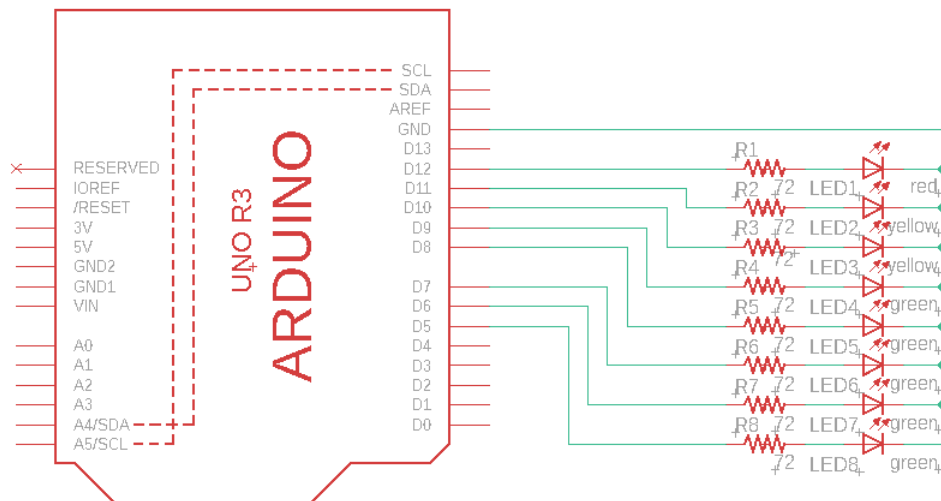


Obr. 2.10: Pinout Wifi modulu nRLF01, vytvorené pomocou návrhového prostredia Eagle

## 2.2.7 Testovanie LED diód

V návrhu modelu bolo využitých taktiež osem LED diód. LED diódy sú praktickým indikátorom pomocou, ktorého je možné užívateľovi poskytnúť rýchlo a zrozumiteľne informácie o stave zariadenia. Môžeme nimi indikovať zapnutie zariadenia, či hlásenie chyby vo funkčnosti. V audiatechnike môžeme často nájsť stĺpce LED diód zapojené ako indikátory hladiny signálu (VU metre, indikátory špičkovej hodnoty a podobne). Práve preto je v návrhu využitých 5 zelených LED diód, 2 žlté LED diódy indikujúce headroom a jedna červená, ktorá odpovedá oblasti limitácie signálu (od čoho bolo neskôr v samotnej realizácii zariadenia upustené). LED diódy boli prakticky testované dvomi spôsobmi a to najskôr manipuláciou jasou jednej samostatnej LED diódy a následne testovaním ovládania počtu rozsvietených LED diód pri ich

spoločnom zapojení. V prvom prípade vznikla nutnosť pripojenia LED diódy na pin s označením PWM (modulácia šírky pulzu). Pri tomto testovaní bol využitý nezmenený Pure Data patch pre testovanie výstupných periférií, kde boli pomocou objektu *vslider* posielané dáta na výstupný pin mikrokontroléru. V prípade testovania pripojenia všetkých ôsmich LED diód naraz nebolo možné spoliehať na kontrolu jasú LED diód, keďže mikrokontrolér ATmega328P obsahuje iba šesť výstupov s PWM. Zvolený bol preto rozličný prístup, ktorý spočíval vo vytvorení rady podmienok *if*, pomocou ktorých sa na základe veľkosti hodnoty premennej v argumente rozhoduje o počte svietiacich LED diód. Táto premenná je pritom nastavovaná ručne v Pure Data pomocou objektu *vslider* (v prípade pripojenia tohto objektu na audio signál môžeme jednoduchým spôsobom vytvoriť indikátor intenzity signálu). Je zrejmé, že v prípade LED diód vzniká totožný problém ako u tlačidiel, kedy využívame osem digitálnych pinov v pomerne obmedzenom využití.

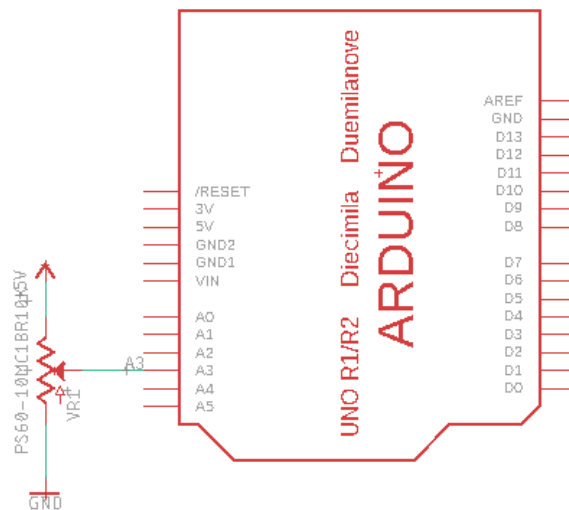


Obr. 2.11: Pinout LED diód, vytvorené v návrhovom prostredí Eagle

## 2.2.8 Testovanie lineárneho ťahového potenciometra pre Arduino

Ťahové potenciometre, či už s lineárnou, alebo logaritmickou charakteristikou, sú neoddeliteľnou súčasťou audiotechniky. Takmer v každom zariadení, kde je potrebné nastavovať hladinu signálu, sú ťahové potenciometre využívané ako intuitívny ovládací prvok. Pre účely testovania bol využitý hotový modul s lineárnym potenciomer-

tom s maximálnou hodnotou odporu 10 k $\Omega$  a dvomi analógovými výstupmi OTA a OTB. Na dosiahnutie vysokej presnosti zariadenia, je potrebné pripojiť obidva analógové výstupy na vstupné piny Arduino dosky a ich hodnoty spriemerovať, no je jasné, že v rámci úspory pinov vo finálnom využívaní zariadenia bude možné využívať iba jeden z daných pinov. Piny Vcc a GND sú už tradične pripájané na napájacie napätie Arduino dosky a zem. Modul bol otestovaný pripojením na testovacie pole v zapojení, ktoré je možno vidieť na obrázku nižšie. Na otestovanie bol použitý kód prístupný v návode k súčiastke, ktorý jednoducho vypisuje na monitor sériového portu údaje o napätí na pine OTA a nebolo ho potrebné nijakým spôsobom meniť. Testovací *patch* pre vstupné periférie v Pure Data takisto nebolo treba za účelom testovania meniť.

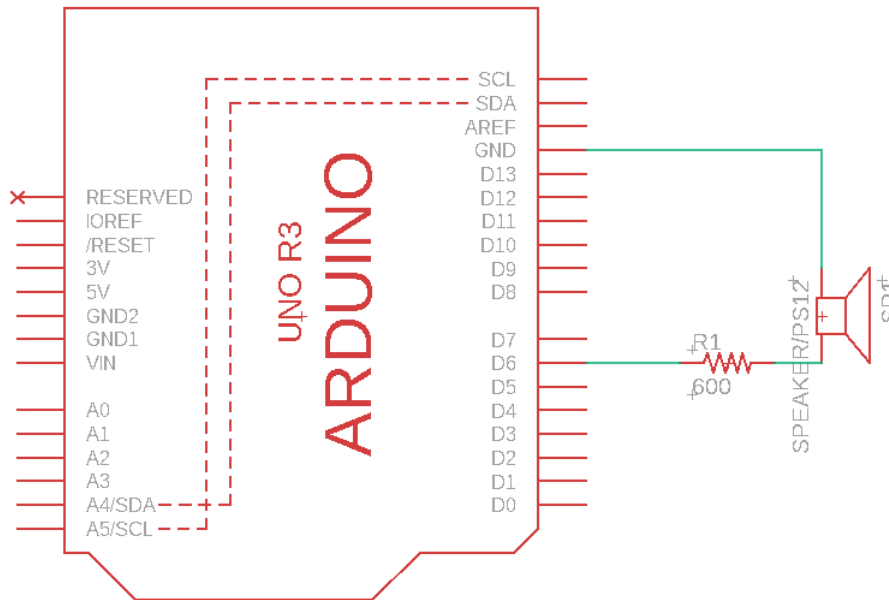


Obr. 2.12: Pinout lineárneho ťahového potenciometra, vytvorené v programe Eagle

## 2.2.9 Testovanie piezoelektrického reproduktora

Piezoelektrický reproduktor je praktickým rozhraním na aplikáciu v audiotechne, ktorý svoju nízku kvalitu zvuku vyrovnáva svojou cenou a jednoduchým používaním. Princíp fungovania piezoelektrického reproduktora je založený na nepriamom piezoelektrickom jave, podľa ktorého dostal i svoj názov, ten vo svojej podstate funguje na premieňaní elektrického napätia na deformácie v křišťále, ktorý na základe svojej oscilácie vytvára zvukové vlny. Pri testovaní bol reproduktor pripojený k Arduino doske na digitálny pin s PWM výstupom, pre ovládanie frekvencie reproduktora. Testovací kód uvedený v návode k súčiastke slúži na postupné prehrávanie jednotlivých tónov chromatickej stupnice a preto bolo nutné vytvoriť vlastný kód. Prvým pokusom na vytvorenie testovacieho kódu bol kód obdobný tomu, ktorým

bol použitý na testovanie intenzity jasu jednej LED diódy, teda jednalo sa o pokús riadiť výšku tónu priamo pomocou PWM skrz objekt vslikeru. Tento pokus však skončil generovaním náhodných nehudobných zvukov a bolo nutné zmeniť prístup. Problém bol vyriešený pridaním podmienok *if else*, kedy boli pomocou premennej v argumente nastavovanej objektom *vslider* prehrávané jednotlivé tóny akordu C-dur.

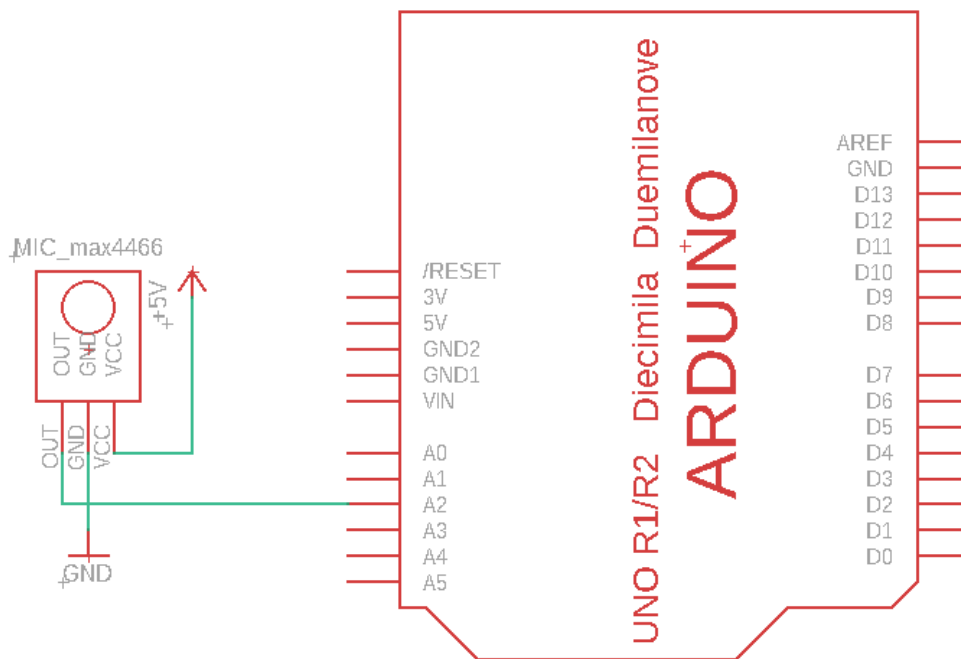


Obr. 2.13: Pinout piezoelektrického reproduktora, vytvorené v návrhovom prostredí Eagle

## 2.2.10 Testovanie modulu mikrofónu so zosilňovačom MAX4466

Mikrofóny sú neoddeliteľnou súčasťou zariadení v audiatechnike a hudobnej elektronike. Bez ich existencie by celé odvetvie audio inžinierstva vo svojej podstate nemohlo existovať. Pri výbere mikrofónu do návrhu modulu boli ako kritérium výberu zohľadňované najmä pomer ceny, kvality a jednoduchosti použitia. Konečné rozhodnutie padlo na modul mikrofónu GY 4466 so zosilňovačom MAX4466. Jedná sa o modul obsahujúci elektretový kondenzátorový mikrofón a nízko výkonový mikrofónový zosilňovač. Samotný modul ako výstup využíva jeden analógový pin Arduino dosky a piny Vcc a GND slúžiace na napájanie modulu. Pri testovaní modulu na testovacej doske bolo použité zapojenie, ktoré je možno vidieť na obrázku uvedenom nižšie. Mikrokontrolér ATmega328P disponuje A/D prevodníkom s rozlíšením desiatich bitov o konverznnej frekvencii až 77 kHz, čo vo svojej podstate stačí na

solídne spracovanie zvukového signálu, no neponúka vyššiu kvalitu ako bežný vbudovaný mikrofón u laptopu. Prvým pokusom na testovanie modulu s mikrofónom bola snaha prenášať zvukový signál v reálnom čase priamo do patchu v Pure Data, no tento pokus skončil neúspechom. Najbližšie ako bola možné sa dostať k spracovaniu audio signálu skrz mikrofón pripojený k mikrokontroléru, bolo nahranie krátkeho audio záznamu vo formáte wav. Toto však neodpovedá požiadavke na ovládanie interaktívnych projektov v reálnom čase a bolo nutné hľadať iný spôsob jeho využitia. V konečnom dôsledku, bol využitý mikrofón ako bežný analógový senzor, vypisujúci svoj výstup na monitor sériového portu. V Pure Data *patchy* pre vstupné periférie bol následne využitý vo funkcii detektora špičkových hodnôt, pomocou ktorého je *triggrovaný* signál *bang*.

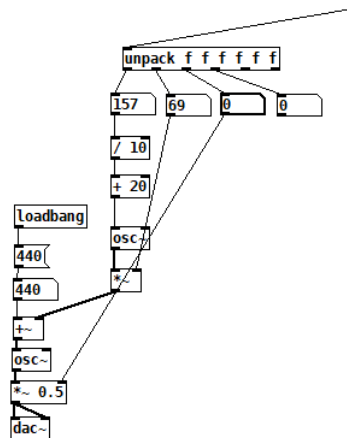


Obr. 2.14: Pinout modul mikrofónu so zosilňovačom MAX4466, vytvorené v návrhovom prostredí Eagle

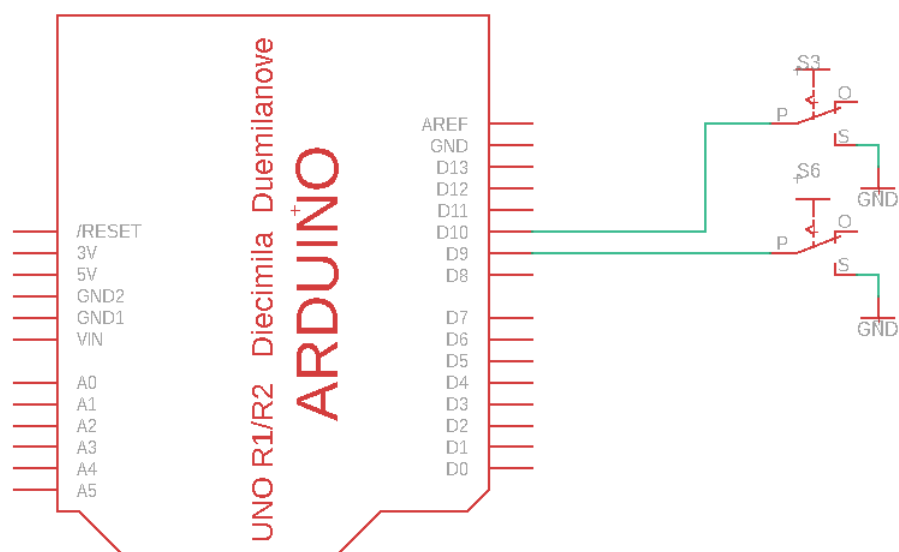
### 2.2.11 Testovanie dvojpólových prepínačov

Posledným z využívaných rozhraní sú dvojpólové prepínače. Ich funkcia je ako u všetkých základných platforiem jednoduchá, no disponujú širokým využitím. V prípade modulu pre interaktívne projekty v Pure Data môžu dvojpólové prepínače slúžiť na vypínanie a zapínanie častí patchu, či zmene módu funkčnosti. Pri testovaní bolo

využitie zapojenie s dvomi pinmi, pričom jeden pin je pripájaný na digitálny výstup Arduino dosky a druhý na zem. Schému zapojenia môžeme vidieť na obrázku uvedenom nižšie. Pôvodnou ideou bolo využiť na testovanie kód využitý na testovanie tlačidiel, no vznikol tu funkčný problém. Tento kód deteguje iba moment prepnutia dvojpólového prepínača a nevypisuje konštantnú správu o jeho stave. Pre vyriešenie tohto problému bolo nutné siahnuť po externej užívateľskej knižnici v jazyku Wiring nazvanej ezButton.h. Táto knižnica obsahuje funkciu *toggle.loop()*, ktorá v zvolenom intervale (v tomto prípade 50 ms) kontroluje stav prepínača a pomocou podmienky *if* vypisuje na monitor sériového portu hodnotu 1, alebo 0. V testovacom patchy v Pure Data pre vstupné zariadenia bolo takisto nutné spraviť drobnú zmenu. Pôvodným zámerom bolo pomocou prepínača vypínať a zapínať toggle switch v Pure Data, no to nebolo možné tým, že prepínač každých 50 milisekúnd vysielal signál *bang*. Spôsob akým bol konečnom dôsledku využívaný prepínač *switch*, je vypisovanie hodnoty 1, alebo 0 do *number boxu* a následné násobenie audio signálu. Týmto spôsobom sme schopný zapínať a vypínať časti audio signálu.



Obr. 2.15: Úprava v Pure Data patchy pre testovanie vstupných periférií, pre účely dvojpólových prepínačov. Do tretieho number boxu sa na základe stavu prepínača vypisuje 1, alebo nula, čím vypíname a zapíname audio signál



Obr. 2.16: Pinout dvojpólových prepínačov, vytvorené pomocou návrhového prostredia Eagle





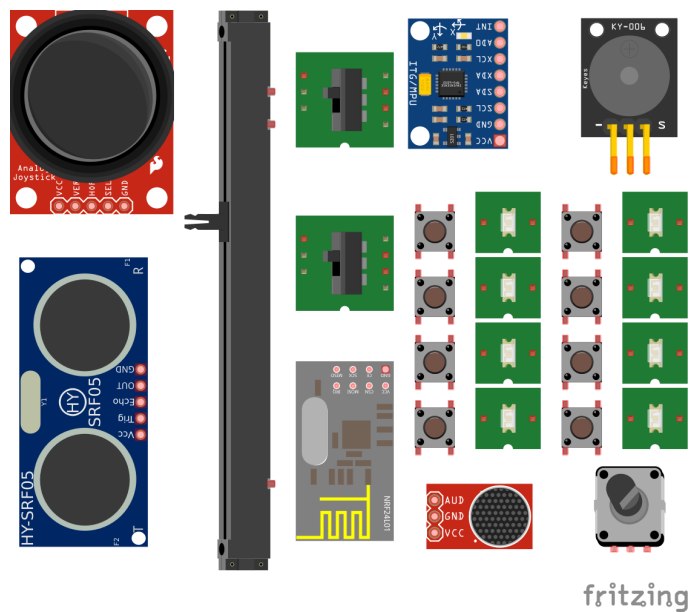
## 3 Návrh Zariadenia

Cielom tejto bakalárskej práce je návrh a konštrukcia plne funkčného hardware modulu pre interaktívne projekty v Pure Data. Pred začiatkom samotného procesu navrhovania je dôležité si uvedomiť požiadavky vyplývajúce zo zadania, účel za ktorým je modul navrhovaný, či spôsob akým bude využívaný v praxi a zistiť možnosti aké nám ponúkajú dostupné periférie a technológie. Zo zadaných požiadaviek na návrh vyplýva nutnosť použiť nasledujúce rozhrania:

- ťahový potenciometer
- dva bipolárne prepínače
- 8 tlačidiel
- piezoelektrický reproduktor
- mikrofón
- modul gyroskopu s akcelerometrom
- modul ultrazvukového snímača vzdialenosti
- modul s Joy Stickom
- modul na bezdrôtovú komunikáciu

### 3.1 Rozloženie periférií

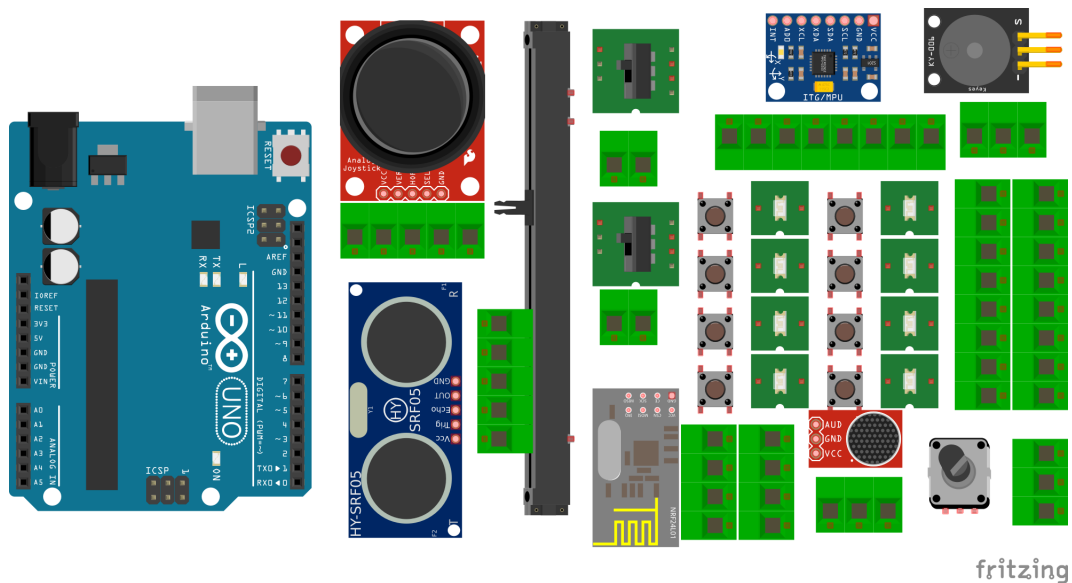
Konkrétnej realizácii modulu pre interaktívne projekty v Pure Data predchádzala fáza návrhu fyzického rozloženia jednotlivých periférií. V úvahu bolo nutné brať niekoľko kritérií. Prvým z nich a dalo by sa povedať najvýznamnejším je efektívne využívanie periférií v praxi, tak aby manipulácia s žiadnou z periférií nezabraňovala, alebo neobmedzovala manipulácii s inou. Je napríklad dôležité, aby modul s ultrazvukovým meradlom vzdialenosti HY-SRF 05 bol, čo možno najviac fyzicky izolovaný od zvyšku periférií, čím sa zabráni neúmyselnej manipulácii. Ďalej je praktické, umiestniť mikrofón a piezoelektrický reproduktor do čo najväčšej vzdialenosti od seba, aby nedochádzalo nechcenému ovplyvňovaniu mikrofónu reproduktorom. Ďalším kritériom je snaha, čo najviac zredukovať rozmer dosky, vzhľadom na výrobnú cenu. Posledným kritériom je rozdelenie periférií do logických celkov. Z tohto dôvodu sme sa napríklad rozhodli umiestniť, bežnejšie využívané periférie do jedného funkčného celku a tie obskúrne svojou polohou zvýrazniť. Taktiež padlo rozhodnutie vzdať sa využívania LED diód vo funkcii indikátora hladiny signálu a prikloniť sa k viac univerzálnemu riešeniu ôsmich jednofarebných LED diód (červených) každej priradenej k jednému tlačidlu. Toto umožňuje využiť LED diódy ako indikátor zapnutých krokov krokového sekvencéru (no nie je to však nutné a využitie LED diód ostáva voľné). Po niekoľkých revíziách bolo dospieť k rozloženiu periférií, ktoré možno vidieť na obrázku uvedenom nižšie.



Obr. 3.1: Návrh fyzického rozloženia periférií v návrhovom prostredí Fritzing

## 3.2 Prvotný koncept zariadenia

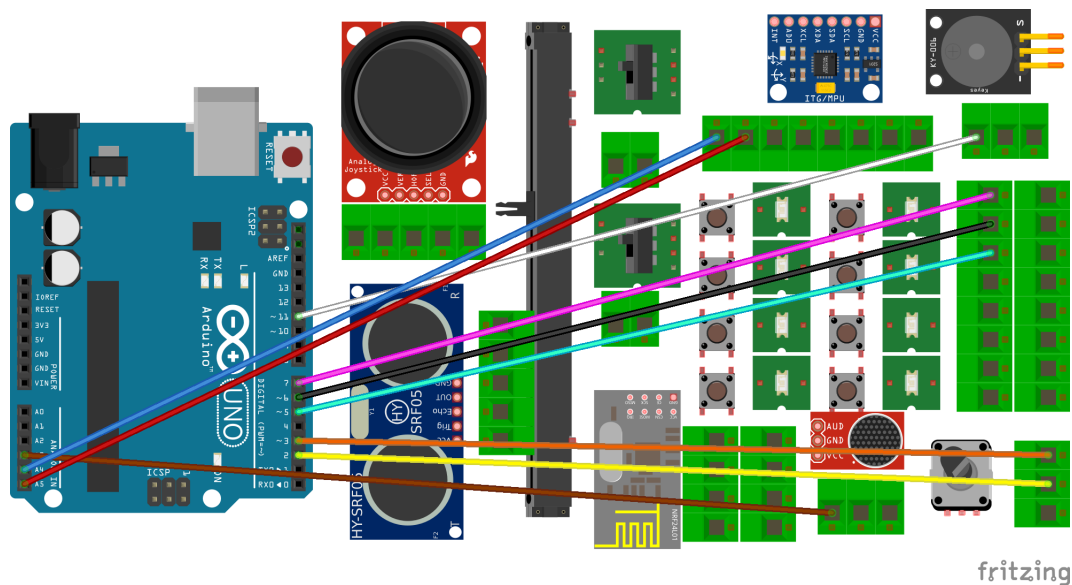
Ideou prvotného konceptu modulu pre interaktívne projekty v Pure Data bolo prikloniť sa k variante, poskytnutia čo najväčšej slobody študentovi vzhľadom na edukatívne využitie zariadenia. Toto malo byť dosiahnuté realizáciou, v ktorej má každá periféria u seba umiestnené dutinkové lišty pomocou ktorých je pripájaná priamo na dosku s mikrokontrolérom Arduino Uno Rev 3. Tento proces čo najvernejšie simuluje reálny proces prototypovania pri práci s platformou Arduino a teda dáva najväčší zmysel z hľadiska vyučovania. Navyše sa jedná o technicky jednoduché a riešenie. Neskôr sme však boli nútený od tohto konceptu realizácie ustúpiť z dôvodov uvedených v nasledujúcej kapitole.



Obr. 3.2: Prvotný koncept realizácie modulu pre interaktívne projekty v Pure Data navrhnutý v návrhovom prostredí Fritzing

Po bližšom zameraní sa na problematiku a prvých pokusoch o realizáciu tohto konceptu sa začalo vynárať čoraz viac problémov súvisiacich s daným spôsobom realizácie modulu pre interaktívne projekty v Pure Data. Prvým a najväčším problémom je, že spolu so slobodou manipulácie so zariadením a prepájania jednotlivých periférií, ktorá je študentovi poskytnutá sa priamoúmerne zvyšuje riziko poškodenia zariadenia. Stačí napríklad, aby nepozorný študent pripojil napájanie modulu nRF24L01 na +5 V namiesto na +3,3 V a celá periféria je zničená. A i v prípade, že študenti sú dobre oboznámení so špecifikami jednotlivých rozhraní (i keď nikdy to nie je stopercentné), tak s konštantným manipulovaním a prepájaním sa zvyšuje riziko poškodenia, ktoré nie je zapríčinené študentmi. Ďalším faktorom, ktorý hovorí v neprospech realizácie modulu týmto konkrétnym spôsobom je jeho prekvapivo nepraktické využívanie v praxi. Ako je možno vidieť na obrázku nižšie, pri zapojení niektorých z periférií, prepojujacie káble úplne zabraňujú používaniu zvyšných periférií. Takisto v zapojení panuje istá neprehľadnosť a chaos (a okrem iného je potreba ku každému jednotlivému modulu zaobstarat dostatočný počet funkčných prepojujacích káblov, čo prináša nečakané finančné náklady a v neposlednom rade ďalšiu starosť pre vyučujúceho). Posledným dôvodom zamietnutia tohto konceptu je úvaha o jeho samotnom zmysle a pridanej funkcionalite. Vo svojej podstate sa jedná o dosku plošných spojov, na ktorú sú pripojené periférie, ktoré sú opäť z dosky plošných spojov prepájané k doske Arduino Uno Rev. 3. V tomto bode sa z modulu stáva v podstate nepotrebný medzi komponent a jeho úlohu by pokojne dokázalo nahradiť obyčajné testovacie pole. Preto sme sa rozhodli hľadať iné riešenie, ktoré

by dovoľovalo periférie využívať praktickejším spôsobom a ponúkalo by dostatočnú pridanú hodnotu k testovaciemu poľu.



Obr. 3.3: Prvotný koncept so zapojenými prepojavacími káblami v návrhovom prostredí Fritzing. Zmyslom tejto ukážky je znázorniť problémy, ktoré so sebou koncept prinášal.

### 3.3 Kultúra shieldu

Platforma Arduino ponúka celú radu oficiálnych i užívateľných tzv. *shieldov*. Jedná o plne funkčné hotové moduly, do ktorých je zo spodnej strany Arduino doska zasunutá. Ponúkajú spravidla novú rozšírenú funkcionalitu k pôvodnej doske. Môžeme tak nájsť Arduino *shieldy*, slúžiace na jednoduchú prácu so servomotormi, pre jednoduché pripojenie k ethernetu, či rôzne výukové *shieldy*. Vzhľadom na túto rozšírenú kultúru shieldu v Arduino komunite padlo rozhodnutie modul koncipovať v tomto duchu, teda ako *shield* s permanentne pripojenými perifériami bez možnosti prepájania. Je nutné podotknúť, že pred týmto rozhodnutím stála ešte úvaha nevyužívať celú dosku s mikrokontrolérom Arduino Uno Rev. 3, no iba samotný mikrokontrolér ATmega328p, čo by výrazne zjednodušilo nároky na rozmer modulu. Po konzultácii však vysvitlo, že zo zadania je požiadavka využiť konkrétne celú dosku a substitúcia za samotný čip s mikrokontrolérom nie je odporúčaná. Taktiež z konkrétneho vybavenia laboratória vysvitla nutnosť využiť konkrétne hotové moduly pre PS2 Joy Stick, akcelerometer s gyroskopom MPU6050, WiFi modul nRF24L01 a ultrazvukový merač vzdialenosti HY-SRF 05 a ich substitúcia takisto nie je možná. Toto sú fakty, ktoré stáli pri rozhodovaní koncipovania modulu vo forme Arduino *shieldu*.

Spolu s týmto rozhodnutím však do procesu návrhu vstúpilo niekoľko významných problémov, z ktorých najzávažnejším bol nedostatočný počet GPIO pinov využívanej Arduino dosky.

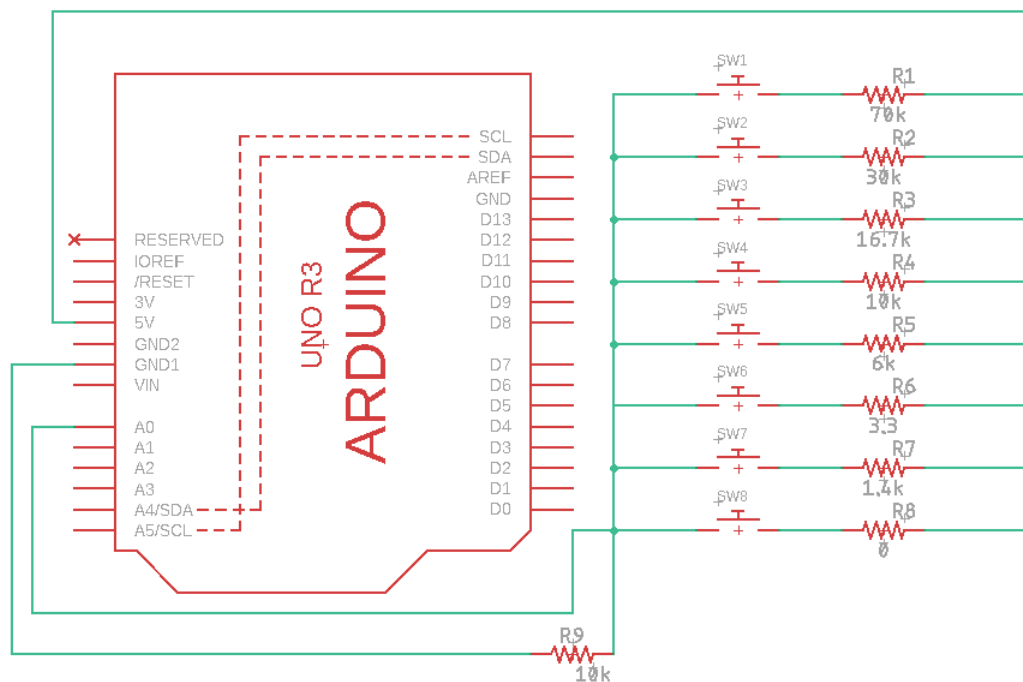
## 3.4 Pinout

S konceptom modulu počítajúcim s pripojením všetkých periférií súčasne vzniká výrazný problém s finálnym pinoutom (konkrétny spôsob pripojenia periférií na piny mikrokontroléru). Všetky požítvané vstupné a výstupné rozhrania na svoje plné fungovanie vyžadujú súhrne 30 digitálnych a 6 analógových pinov, pričom niektoré z pinov periférií je nutné pripojiť na dedikované piny mikrokontroléru (hlavne rozhrania využívajúce SPI a I<sup>2</sup>C zbernice). Tento počet výrazne prevyšuje počet dostupných GPIO pinov dosky Arduino Uno Rev.3, ktorý činí 14 digitálnych a 6 analógových pinov. Ďalším faktorom ovplyvňujúci finálny pinout je fakt, že pre potreby komunikácie mikrokontroléru s Pure Data je nutné neustále pripojenie Arduino dosky k počítaču pomocou zbernice UART. Táto zbernica má na svoje fungovanie dedikované dva piny mikrokontroléru D0 a D1 a nie je možné ich používať na pripájanie periférií počas jej používania, čím sa nám celkový počet digitálnych pinov znížil na 12.

### 3.4.1 Redukcia potrebných pinov tlačidiel

#### Využitie princípu odporového deliča

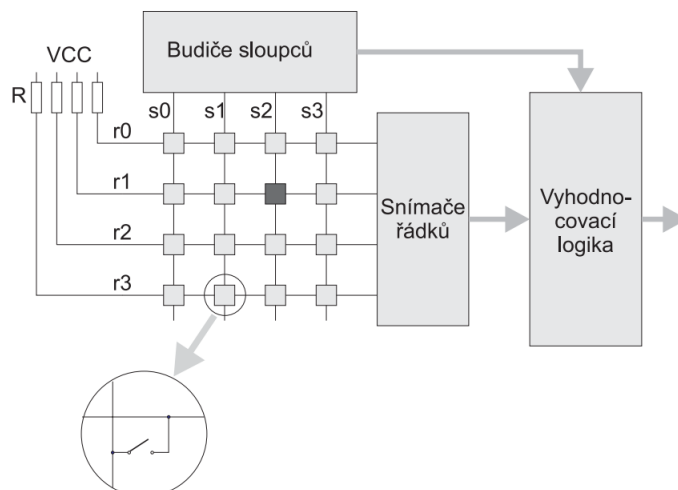
Ako riešenie sa pôvodne zdalo zapojenie tlačidiel s využitím odporového deliča, čo by počet potrebných pinov zredukovalo na jeden analógový pin. Celá idea je založená na pripojení jednotlivých tlačidiel na rezistory rôznych hodnôt paralelne s jedným rezistorom, čo v konečnom dôsledku pri stláčaní tlačidiel vytvára odporové deliče rôznych pomerov a čip ATmega328P potom deteguje napätia rôznych hodnôt, ako dané tlačidlá. Toto riešenie sa v konečnom dôsledku ukázalo, ako zbytočne konštrukčne i programovo náročné.



Obr. 3.4: Princíp zapojenia tlačidiel ako odporového deliča, vytvorené v návrhovom prostredí Eagle

### Princíp maticovej klávesnice

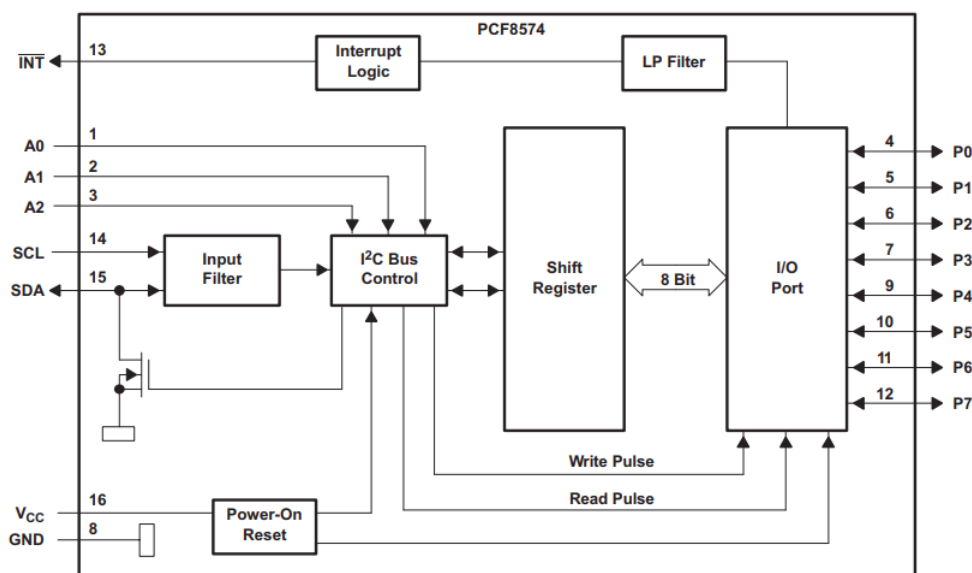
Ďalšou možnosťou ako redukovať počet pinov potrebných na zapojenie je využitie maticovej klávesnice. Princípom jej fungovania je detekcia stĺpcovej a riadkovej súradnice tlačidla. Každý riadok a stĺpec je pomocou pull up rezistora udržiavaný na úrovni logickej jednotky, no v momente, kedy je stlačené niektoré z tlačidiel vzniká skrat a signál na riadku a stĺpci je stiahnutý na úroveň logickej nuly. Softvérovo je potom na základe týchto súradníc vyhodnotené konkrétne stlačené tlačidlo. Týmto spôsobom je možné násobne výrazne znížiť počet potrebných pinov. Problémom však je, že táto úspora sa prejavuje až u väčšieho počtu tlačidiel a u počtu ôsmich tlačidiel, ktoré sú zariadení využívané nenastáva žiadna úspora.



Obr. 3.5: Princíp fungovania maticovej klávesnice <http://home.zcu.cz/~dudacek/P-Z/klavesnice.pdf>

### Pripojenie tlačidiel na GPIO Expandér PCF8574

Ak sa jedná o rozširovanie potrebných GPIO pinov mikrokontrolérov je často využívanou a praktickou variantou využiť GPIO expandéry pripájané na I<sup>2</sup>C zbernicu. Jedným z takýchto expandérov je i čip PCF8574 od firmy Texas Instruments.



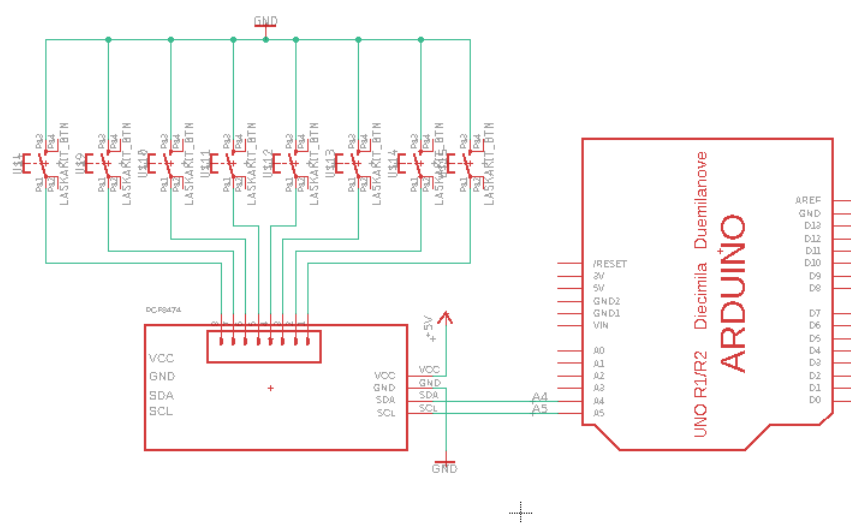
Obr. 3.6: Zjednodušený blokový diagram čipu PCF8574 od firmy Texas Instruments. Oficiálny datasheet súčiastky

Jeho hlavnou výhodou je, že pripojením na zbernicu nevzniká nárok na žiadne

digitálne piny a je možné sériovo za seba zapojiť niekoľko expandérov. Týmto spôsobom je možné zredukovať osem pinov potrebných na pripojenie jednotlivých tlačidiel.

### Testovanie pripojenia tlačidiel na GPIO expandér PCF8574

Za zámerom využitia tlačidiel v zapojení skrz GPIO expandér PCF8574 bolo nutné opäť vykonať fázu testovania. Pre tieto účely bol využitý hotový modul, ktorý umožňuje pripojenie čipu na testovacie pole. Spôsob pripojenia možno vidieť na obrázku uvedenom nižšie, kde je modul expandéru pripojený na napájanie a jeho piny SCL a SDA sú pripojené na analógové piny A5 a A4 dosky s mikrokontrolérom. Pred samotným testovaním je nutné jumperom nastaviť modulu s čipom adresu, ktorá bude v nasledujúcom testovacom kóde modulu pridelená. Túto adresu je možné vypočítať pomocou vzťahu uvedenom v katalógovom liste k súčiastke, no pre spoľahlivosť je lepšie ju skontrolovať pomocou skenovacieho kódu I<sup>2</sup>C zariadení, ktorý je súčasťou Arduino IDE. Pre komunikáciu s čipom PCF8574 je nutné implementovať používateľskú knižnicu *PCF8574.h*, ktorá umožňuje využívať piny expanderu P0 až P7 ako bežné digitálne piny Arduino dosky. Logika samotného testovacieho kódu je potom jednoduchá, kedy mikrokontrolér deteguje nízku úroveň signálu na vstupe a vypisuje na monitor sériového portu 1. Čo sa týka úprav testovacieho *patchu* pre vstupné periférie v Pure Data, bola využitá varianta pre testovanie ôsmich tlačidiel s tým rozdielom, že u objektu *comport* bola zmenená rýchlosť komunikácie na 112500 (rýchlosť komunikácie všetkých zariadení pripojených na zbernicu I<sup>2</sup>C).



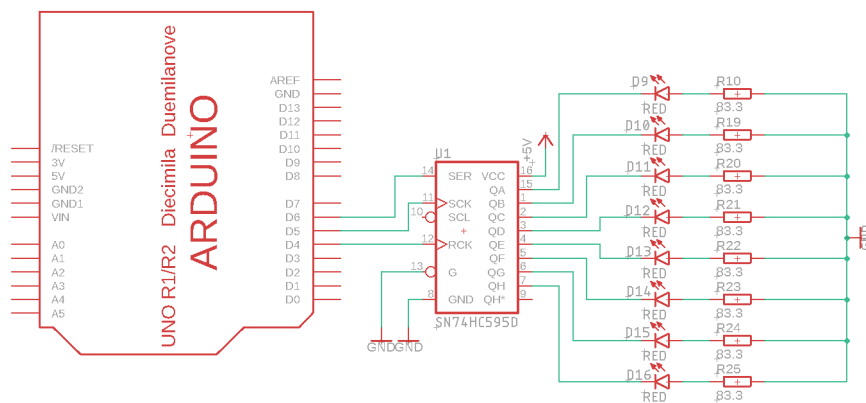
Obr. 3.7: Pinout tlačidiel pripojených skrz GPIO expandér PCF8574



## 3.4.2 Redukcia potrebných pinov LED diód

### Posuvný register 74HC595

Pôvodnou ideou na redukciiu potrebných pinov pri zapájaní LED diód bolo využitie posuvného registra 74HC595 typu serial-in paralel-out od firmy Texas Instruments. Toto riešenie sa však ukázalo ako zbytočne programovo náročné a nevyhovujúce z hľadiska realizácie. Navyše pre zapojenie posuvného registra sú potreba 3 digitálne piny mikrokontroléru, čo nepredstavuje dostatočnú úsporu.



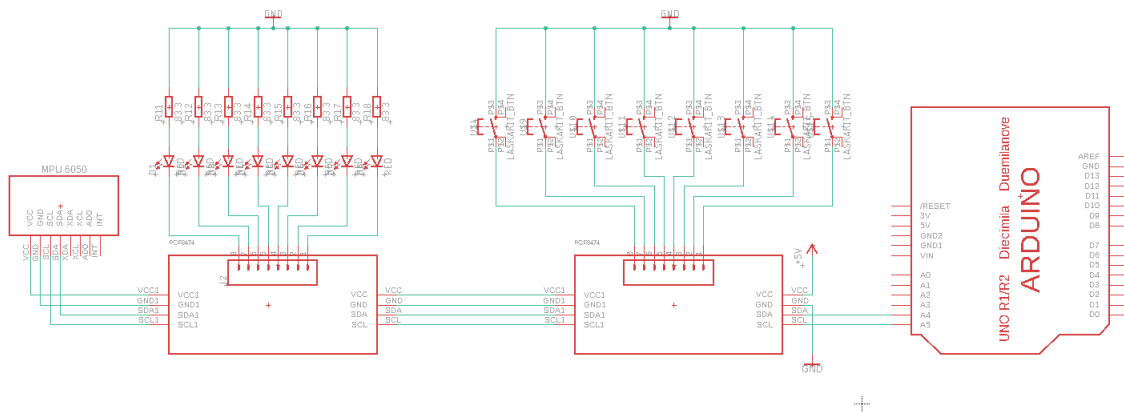
Obr. 3.8: Pinout LED diód pripojených skrz GPIO Posuvný register 74HC595, vytvorené pomocou návrhového prostredia Eagle

### Pripojenie LED diód na GPIO Expandér PCF8574

Ako najvýhodnejšie riešenie problému sa taktiež ukázalo využitie GPIO expandéru PCF8574, no tentokrát v režime rozšírenia výstupných pinov. To znamená, že vo finálnej realizácii modulu pre interaktívne projekty v Pure Data budeme mať na I<sup>2</sup>C zbernicu zapojené celkovo tri periférie.

### Testovanie pripojenia LED diód na GPIO expandér PCF8574

V rámci testovania ovládania ôsmich LED diód na GPIO expandér PCF8574 bol využitý priestor na otestovanie správania I<sup>2</sup>C zbernice v prípade zapojenia viacerých periférií naraz. Bolo nutné dbať na rôzne nastavenie adresových jumprov jednotlivých periférií, aby neprišlo ku konfliktu adres pri komunikácii (jednotlivé adresy boli opäť overené pomocou skenovacieho kódu). Čo sa týka software časti testovania, využívame opäť užívateľskú knižnicu *PCF8574.h*. Samotná logika kódu a prepojenia s testovacím *patchom* pre výstupné zariadenia je totožná, ako u testovania LED diód bez využitia GPIO expandéru.



Obr. 3.9: Pinout pripojenia všetkých periférií využívajúcich zbernicu I<sup>2</sup>C naraz, vytvorené pomocou návrhového prostredia Eagle

### 3.5 Finálny pinout zariadenia

V tabuľke nižšie je možno vidieť finálny pinout zariadenia po všetkých opatreniach na redukcii potrebných pinov. Okrem využitia GPIO expandérov PCF8574 bolo nutné tatiež nepripájať tlačíla u modulu s Joy Stickom a rotačného enkodéru. Tento kompromis bol nutný pre uvoľnenie pinov D0 a D1 pre komunikáciu UART.

Arduino pin	pin periférií
A0	Joy Stick (X)
A1	Joy Stick (Y)
A2	Modul s mikrofónom
A3	Lineárny ťahový potenciometer
A4(SDA)	GPIO EXP(8xBTN)→ GPIO EXP(8xLED)→GYRO
A5(SCL)	GPIO EXP(8xBTN)→ GPIO EXP(8xLED)→GYRO
D0	Rezervované pre UART
D1	Rezervované pre UART
D2	Enkodér (A)
D3	Enkodér (C)
D4	HY-SRF 05 (Echo)
D5	HY-SRF 05 (Trig)
D6	Piezo reproduktor
D7	nRF24L01 (CE)
D8	nRF24L01 (CSN)
D9	Bipolárny posuvný prepínač 1
D10	Bipolárny posuvný prepínač 2
D11(MOSI)	nRF24L01 (MOSI)
D12(MISO)	nRF24L01 (MISO)
D13	nRF24L01 (SCK)

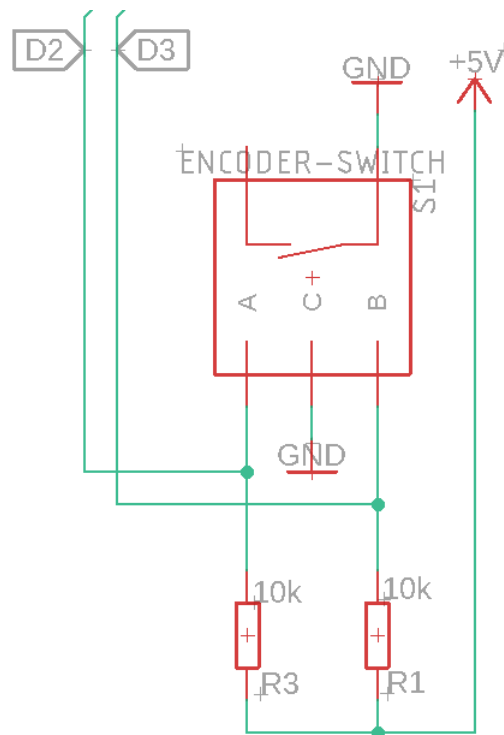


## 4 Konštrukcia zariadenia

U periférií, pri ktorých nebola explicitná požiadavka využiť hotové moduly a zároveň toto rozhodnutie poskytovalo konštrukčnú výhodu boli použité SMD/THT varianty týchto súčiastok spolu so všetkými potrebnými obvodmi pre ich fungovanie.

### 4.1 Realizácia enkodéru na DPS

Vo finálnej konštrukcii modulu pre interaktívne projekty v Pure Data bol využitý 12 milimetrový rotačný enkodér RE130F-41-175F-12P od firmy Alpha (Taiwan). Medzi päť voltovým napájaním a vstupmi A a B enkodéra sú osadené 10k $\Omega$  tenkovrstevné rezistory v puzdre 0603 za účelom potlačenia zákmitov (*debouncing*).

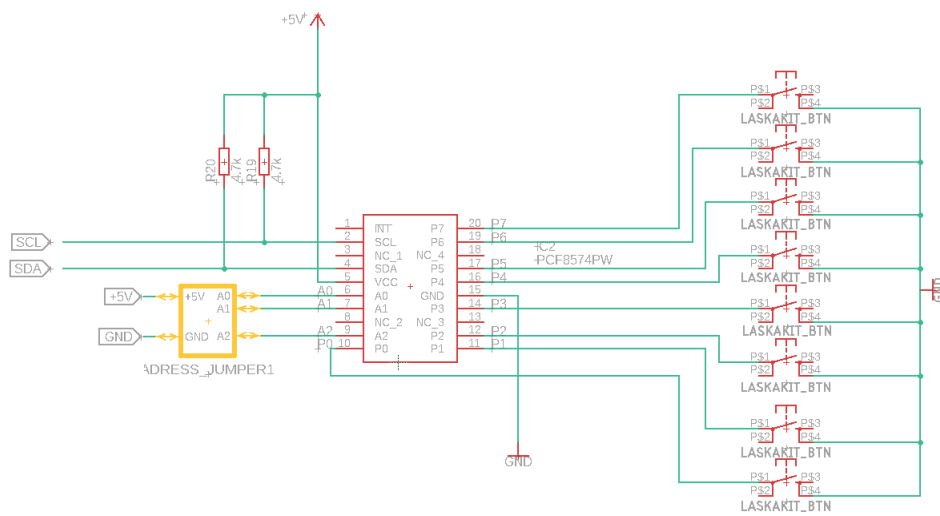


Obr. 4.1: Finálne zapojenie enkodéru, vytvorené v návrhovom prostredí Eagle

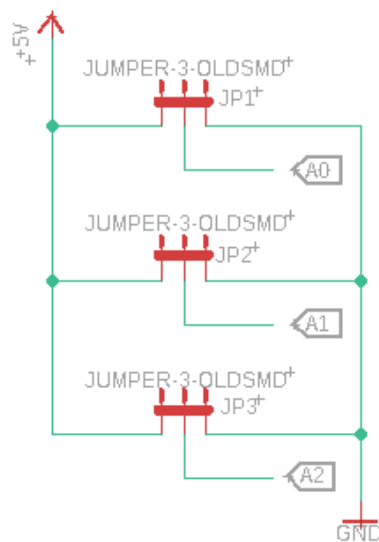
### 4.2 Realizácia GPIO expandéru PCF8574 v zapojení s tlačidlami na DPS

Vo finálnej konštrukcii bol využitý GPIO expandér PCF8574 v puzdre TSSOP od firmy Texas Instruments. Medzi vodičmi SDA, SCL a zdrojom napätia boli umiestnené tenkovrstvé pull-up rezistory s odporom 4,7k $\Omega$  v puzdre 0603. Obvod bol tak-

tiež doplnený o SMD adresový jumper. Na vstup čipu je pripojených osem šesť milimetrových THT tlačidiel v active low zapojení.



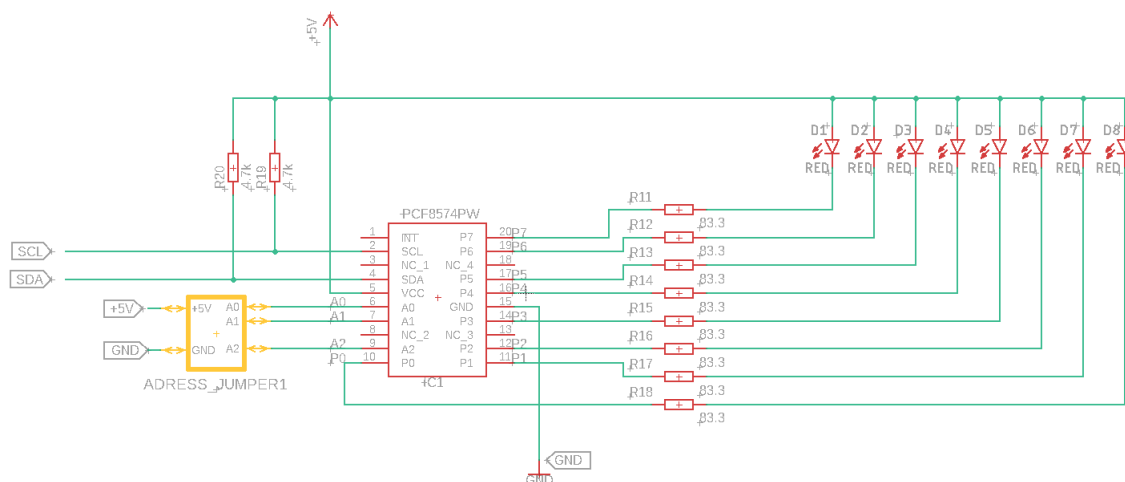
Obr. 4.2: Finálne zapojenie tlačidiel na čip GPIO expandéra vytvorené v návrhovom prostredí Eagle



Obr. 4.3: Finálne zapojenie SMD jumpra vytvorené v návrhovom prostredí Eagle

## 4.3 Realizácia GPIO expandnéru PCF8574 v zapojení s LED diódami na DPS

Zapojenie čipu odpovedá tomu v zapojení s tlačidlami. Na GPIO pinoch čipu P0 až P7 je pripojených osem červených LED diód v zapojení so spoločnou katódou v SMD prevedení v puzdre 0603 a osem rezistorov o veľkosti  $100\ \Omega$  takisto v SMD prevedení v puzdre 0603.



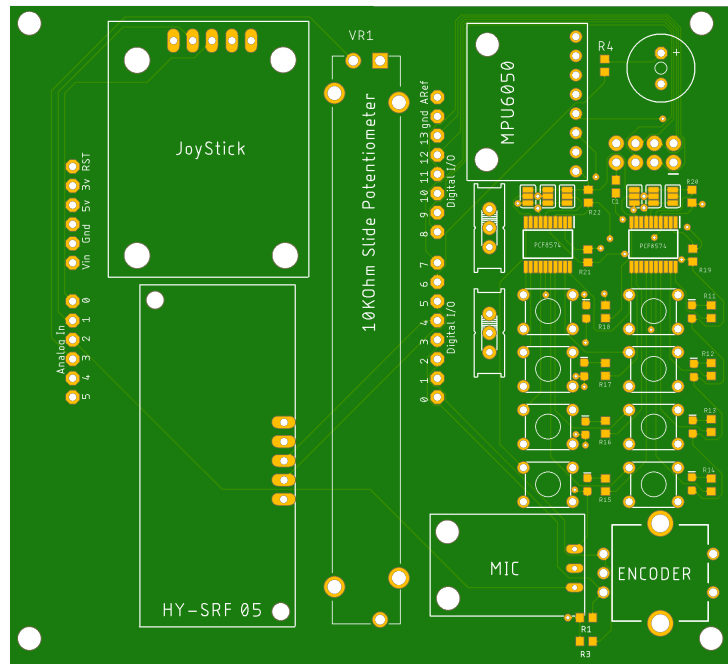
Obr. 4.4: Finálne zapojenie LED diód na čip GPIO expandéra vytvorené v návrhovom prostredí Eagle

## 4.4 Doska plošných spojov

Výsledkom práce je doska plošných spojov s týmito charakteristikami:

- Rozmer dosky je 96,17 mm x 87,42 mm
- Štyri vrstvy - vnútorné vrstvy slúžia ako zemná a napájacia plocha - lepšie vysokofrekvenčné vlastnosti
- Materiál FR-4

#### 4.4.1 TOP vrstva



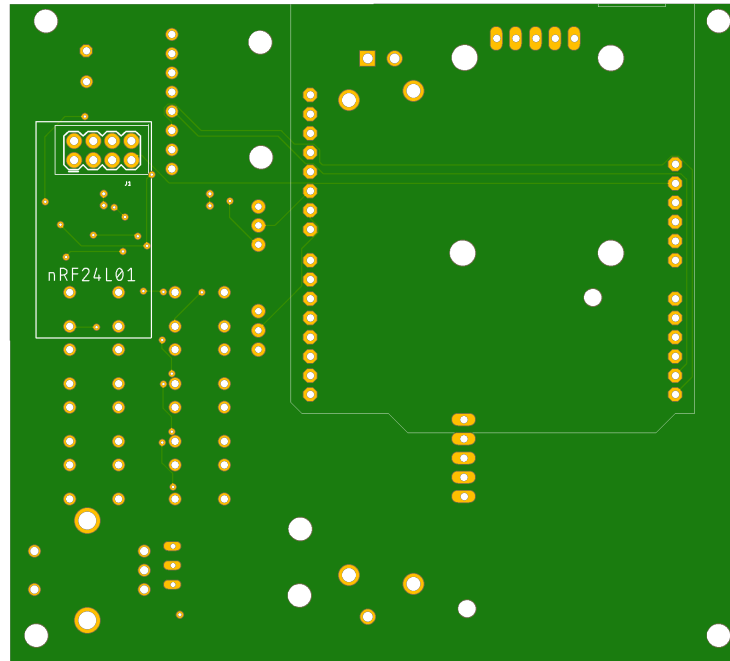
Obr. 4.5: TOP vrstva dosky plošných spojov, vztvorené pomocou návrhového prostredia Eagle

TOP vrstva dosky plošných spojov obsahuje:

- Modul s PS2 Joy Stickom
- Modul s akcelerometrom a gyroskopom MPU 6050
- Enkodér
- Piezoelektrický reproduktor
- Modul s ultrazvukovým meračom vzdialenosti HY-SRF 05
- Ťahový potenciometer
- Modul mikrofónu so zosilňovačom MAX4466
- GPIO expandér PCF8574 s ôsmimi tlačidlami
- GPIO expandér PCF8574 s ôsmimi LED diódami
- Dva bipolárne prepínače



#### 4.4.2 BOTTOM vrstva



Obr. 4.6: BOTTOM vrstva dosky plošných spojov, vztvorené pomocou návrhového prostredia Eagle

- Arduino Uno Rev. 3
- Wifi modul nRF24L01



## Záver

V rámci bakalárskej práce bola rozobratá problematika grafických programovacích jazykov slúžiacich na vytváranie hudobných a multimediálnych projektov, konkrétne opensource programovacie prostredie Pure Data. V úvode práce boli okolnosti a motiváciu jeho vzniku, jeho miesto medzi ostatnými jazykmi patriacimi do rodiny „patcher“ jazykov, kompabilitu, ako i informácie o základných princípoch jeho funkčnosti a orientácii v užívateľskom rozhraní. Ďalšou z tém práce, rozobranej v teoretickom úvode, bol využívaný mikrokontrolér Arduino UNO Rev. 3, spolu s čipom ATmega328P a priblíženie špecifik, na ktorých základoch tento hardware pracuje. Arduino UNO je výbornou pomôckou v práci s digitálnou elektronikou, keďže zabezpečuje zjednodušenie obsluhy čipu ATmega328P, ktorý parí k čipom využívajúcim Harvardskú architektúru s redukovaným súborom inštrukcií (RISC). Obsahom ďalšej časti práce bolo priblíženie procesu testovania jednotlivých periférií využívaných v návrhu modulu pre interaktívne projekty v Pure Data. Podstatnou súčasťou tohto procesu bolo oboznámenie sa s princípmi funkčnosti periférií a preskúmanie možností, ktoré ponúkajú v rámci edukatívnej práce s Pure Data. Funkčnosť jednotlivých modulov sme preto najskôr otestovali na testovacom poli. Za účelom testovania boli využité kódy v jazyku Wiring, inšpirované kódmi dostupnými k jednotlivým rozhraniam. Takisto boli navrhnuté vzorové patche v Pure Data separátne pre prácu s vstupnými a výstupnými perifériami. V tretej časti práce bol rozobraný proces návrhu a koncepcie zariadenia. Boli vysvetlené dôvody opustenia pôvodnej koncepcie, ktorá spočívala v prepájaní periférií s doskou študentmi a boli vysvetlené procesy, ktoré viedli k novej koncepcii. Tá spočíva v permanentnom pripojení všetkých periférií na dosku súčasne v duchu Arduino shield. Spolu s novým konceptom dizajnu zariadenia však nastal i problém s nedostatočným počtom vstupne výstupných pinov dosky s mikrokontrolérom. Tento problém bol nakoniec vyriešený pomocou čipov s expandérmí pripájanými na zbernicu I2C. V poslednej časti práce sa venujeme finálnemu návrhu dosky plošných spojov. Výsledkom práce je doska o rozmere 96,17 mm x 84,72 mm a štyroch vrstvách pre zlepšenie vysokofrekvenčných vlastností zariadenia.



# Literatúra

- [1] KAVAN, Jan. *Pure Data: platforma pro tvorbu interaktivního díla*. Brno: Janáčkova akademie múzických umění v Brně, 2013. ISBN 8074600335
- [2] CÁB, Michal. *Pure Data: Rukověť postdigitálního umělce, ver. 0.1a* Praha: Akademie výtvarných umění v Praze, 2013. ISBN 8074600335
- [3] SELECKÝ, Matúš: *Arduino: uživatelská příručka*. Brno: Computer press, 2016. ISBN 978-802-5148-402.
- [4] EVANS, Brian. *Beginning Arduino Programming* New York, NY, USA.: Apress, 2011. ISBN 977-4302-377-8
- [5] What is RISC (reduced instruction set computer)? | Definition from TechTarget. Purchase Intent Data for Enterprise Tech Sales and Marketing  
Dostupné z URL:<<https://www.techtarget.com/whatis/definition/RISC-reduced-instruction-set-computer>>
- [6] DATASHEET, Arduino® UNO R3 [cit.2022-12-12] [online]  
Dostupné z URL:<<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>>
- [7] Datasheet, Atmel Corporation. *San Jose, CA, 2008* [online][cit.2022-12-12]  
Dostupné z URL:<[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)>
- [8] How Rotary Encoder Works and How To Use It with Arduino - How To Mechatronics. How To Mechatronics [online]. Copyright © 2022 HowToMechatronics.com. All rights reserved. [cit. 12.12.2022]  
Dostupné z URL:<<https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>>
- [9] Arduino Joystick PS2 | Návody Drátek *Návody Drátek. Webový magazín o ARDUINU / Návody Drátek* [online]  
Dostupné z URL:<<https://navody.drateg.cz/navody-k-produktum/arduino-joystick-ps2.html>>
- [10] Ultrazvukový měřič vzdálenosti HY-SRF05 | Návody Drátek *Návody Drátek. Webový magazín o ARDUINU / Návody Drátek* [online]  
Dostupné z URL:<<https://navody.drateg.cz/navody-k-produktum/ultrazvukovy-meric-vzdalenosti-hy-srf05.html>>

- [11] Rotační enkodér KY-040 | Návodý Drátek | *Návodý Drátek Webový magazín o ARDUINU / Návodý Drátek* [online]  
Dostupné z URL:<[https://navody.drateg.cz/?gclid=CjwKCAiAv9ucBhBXEiwA6N8nYMeBEWsklYrn00k2b71xTrfFB8GqcnMrZ-XKdu\\_jXuJax6Xk9qpRCBoCqrAQAvD\\_BwE](https://navody.drateg.cz/?gclid=CjwKCAiAv9ucBhBXEiwA6N8nYMeBEWsklYrn00k2b71xTrfFB8GqcnMrZ-XKdu_jXuJax6Xk9qpRCBoCqrAQAvD_BwE)>
- [12] How Rotary Encoder Works and How To Use It with Arduino - How To Mechatronics. How To Mechatronics [online]. Copyright © 2022 HowToMechatronics.com. All rights reserved. [cit. 12.12.2022]  
Dostupné z URL:<<https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>>
- [13] Arduino WiFi modul NRF24L01 | *Návodý Drátek. Webový magazín o ARDUINU / Návodý Drátek* [online]  
Dostupné z URL:<<https://navody.drateg.cz/navody-k-produktum/arduino-wifi-modul-nrf24l01.html>>
- [14] Datasheet, Atmel Corporation. *San Jose, CA,2008* [online][cit.2022-12-12]  
Dostupné z URL:<[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)>
- [15] Datasheet, Atmel Corporation. *San Jose, CA,2008* [online][cit.2022-12-12]  
Dostupné z URL:<[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)>
- [16] *PCF8574 Remote 8-Bit I/O Expander for I2C Bus*[online]. Texas Instruments [cit.2023-6-20]  
Dostupné z URL:<<https://www.ti.com/lit/ds/symlink/pcf8574.pdf>>
- [17] *SN74HC595 8-Bit Shift Registers With 3-State Output Registers*[online]. Texas Instruments [cit.2023-6-20]  
Dostupné z URL:<<https://www.ti.com/lit/ds/symlink/sn74hc595.pdf>>

## Zoznam symbolov a skratiek

<b>DPS</b>	Doska plošných spojov
<b>CAC</b>	Počítačom podporovaná kompozícia – Computer aided composition
<b>CGM</b>	Počítačom generovaná hudba – Computer generated music
<b>PWM</b>	Pulzne šírková modulácia – Pulse width modulation
<b>GPIO</b>	Univerzálny výstupný/vstupný pin – General-purpose input/output pin





# Zoznam príloh

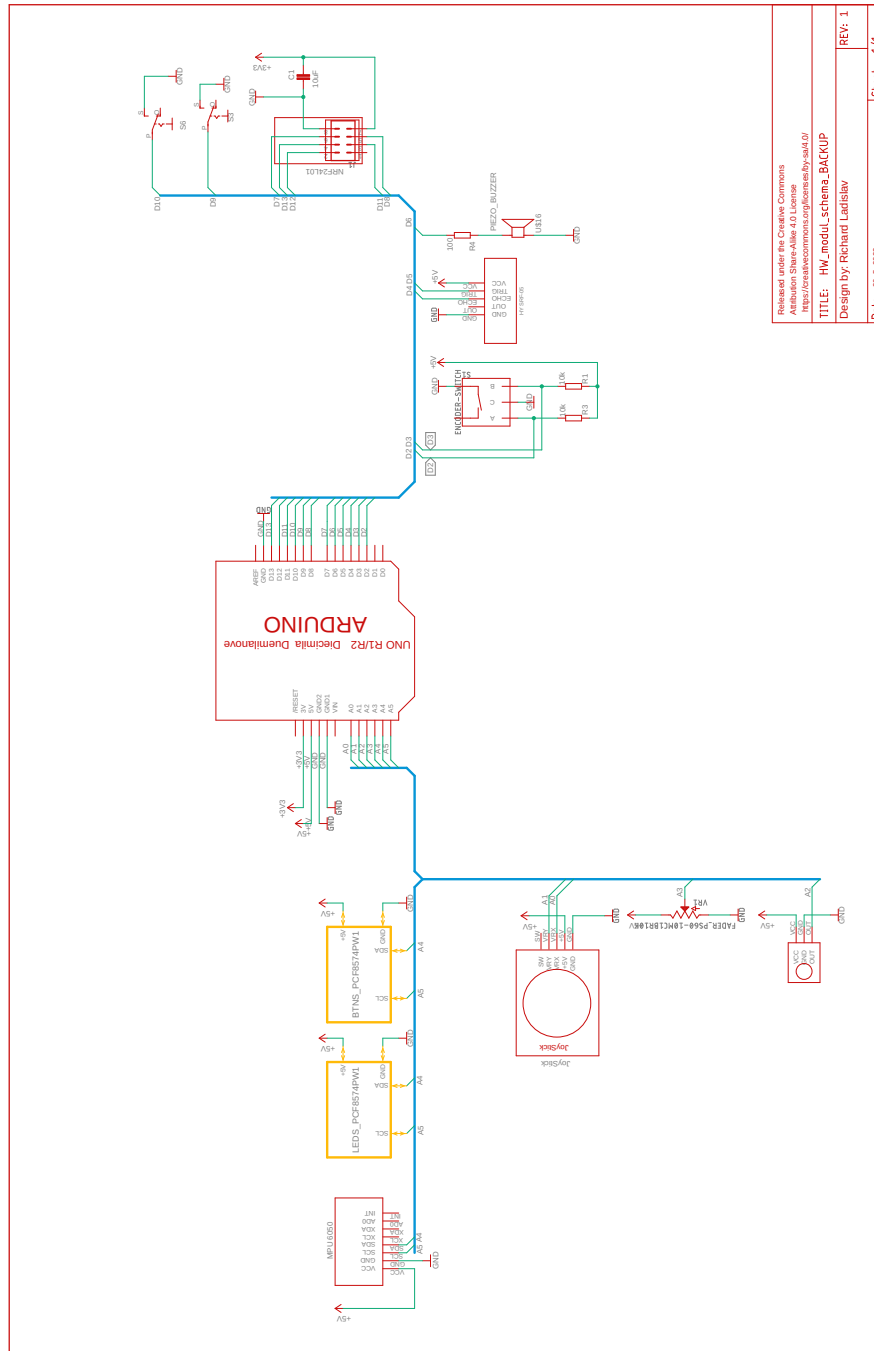
<b>A Dokumentácia DPS</b>	<b>76</b>
A.1 Schéma zapojenia . . . . .	76
A.2 TOP vrstva DPS . . . . .	77
A.3 BOTTOM vrstva DPS . . . . .	78
A.4 Osadenie súčiastok vrstvy TOP . . . . .	79
A.5 Osadenie súčiastok vrstvy BOTTOM . . . . .	80
A.6 Fotografia zariadenia TOP . . . . .	81
A.7 Fotografia zariadenia BOTTOM . . . . .	82
<b>B Obsah elektronickej prílohy</b>	<b>83</b>





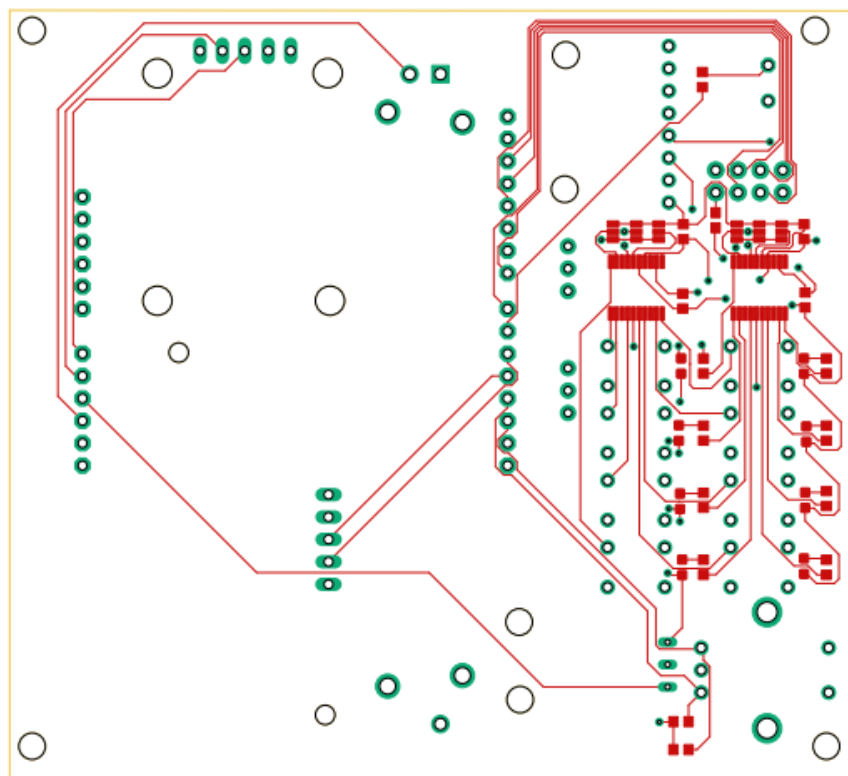
# A Dokumentácia DPS

## A.1 Schéma zapojenia

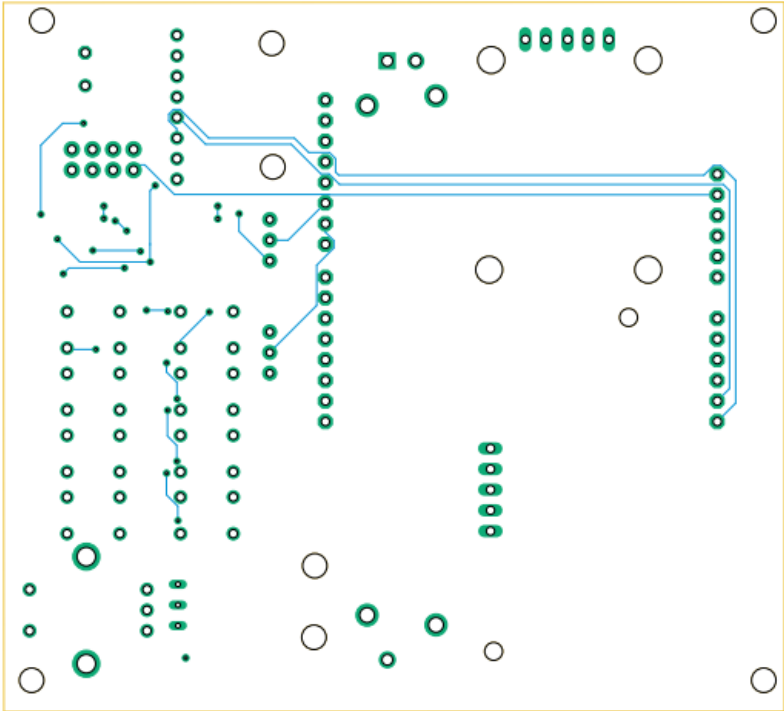


## A.2 TOP vrstva DPs

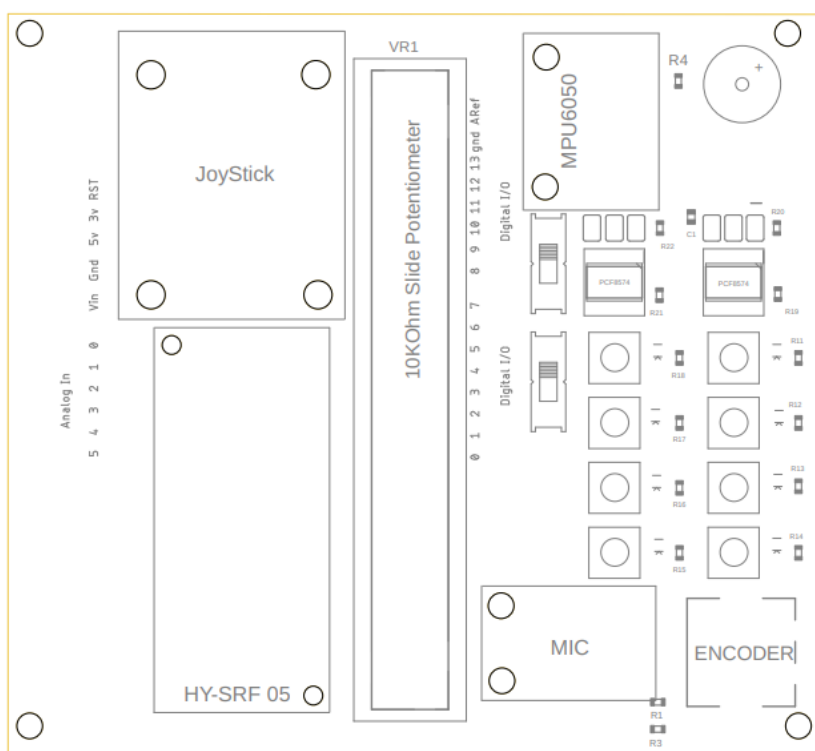
---



### A.3 BOTTOM vrstva DPs

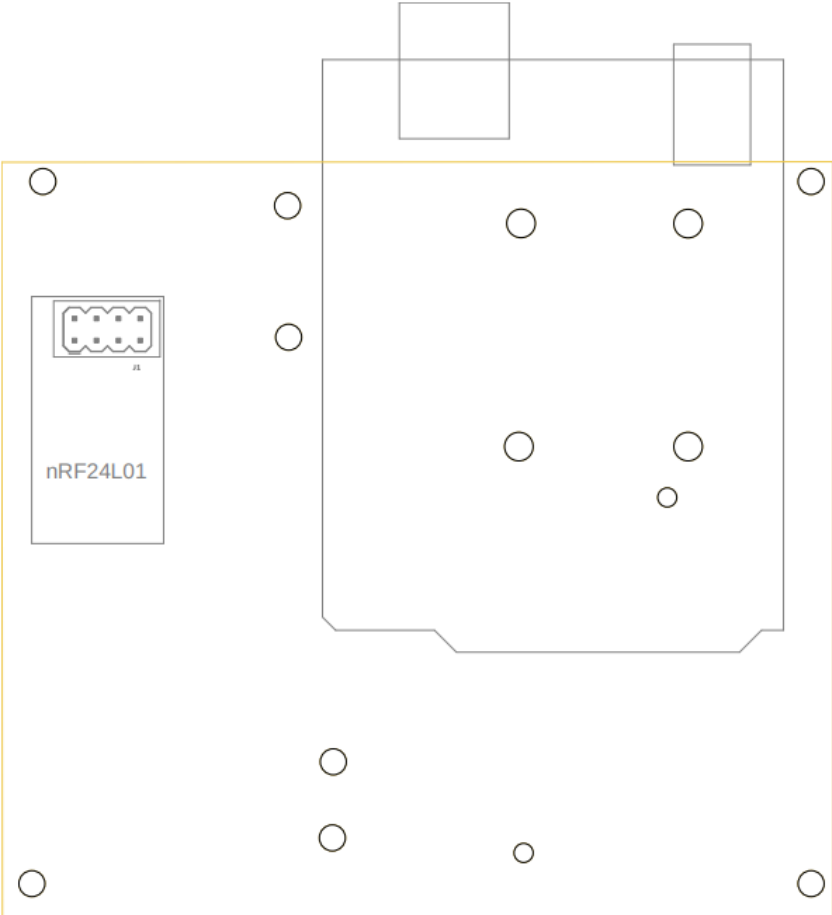


## A.4 Osadenie súčiastok vrstvy TOP



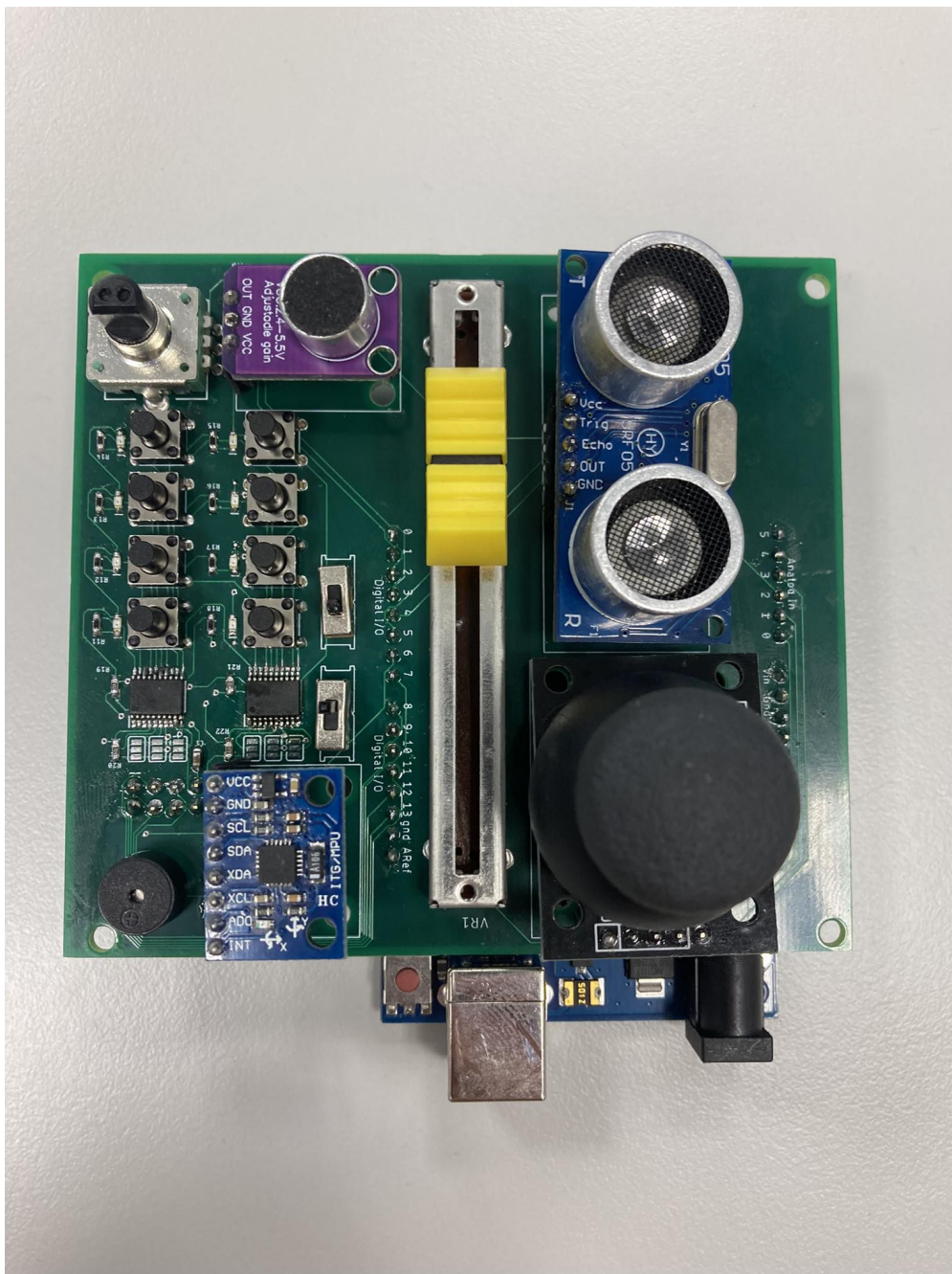
Released under the Creative Commons Attribution Share-Alike 4.0 License  
<https://creativecommons.org/licenses/by-sa/4.0/>  
Designed by:

# A.5 Osadenie súčiastok vrstvy BOTTOM

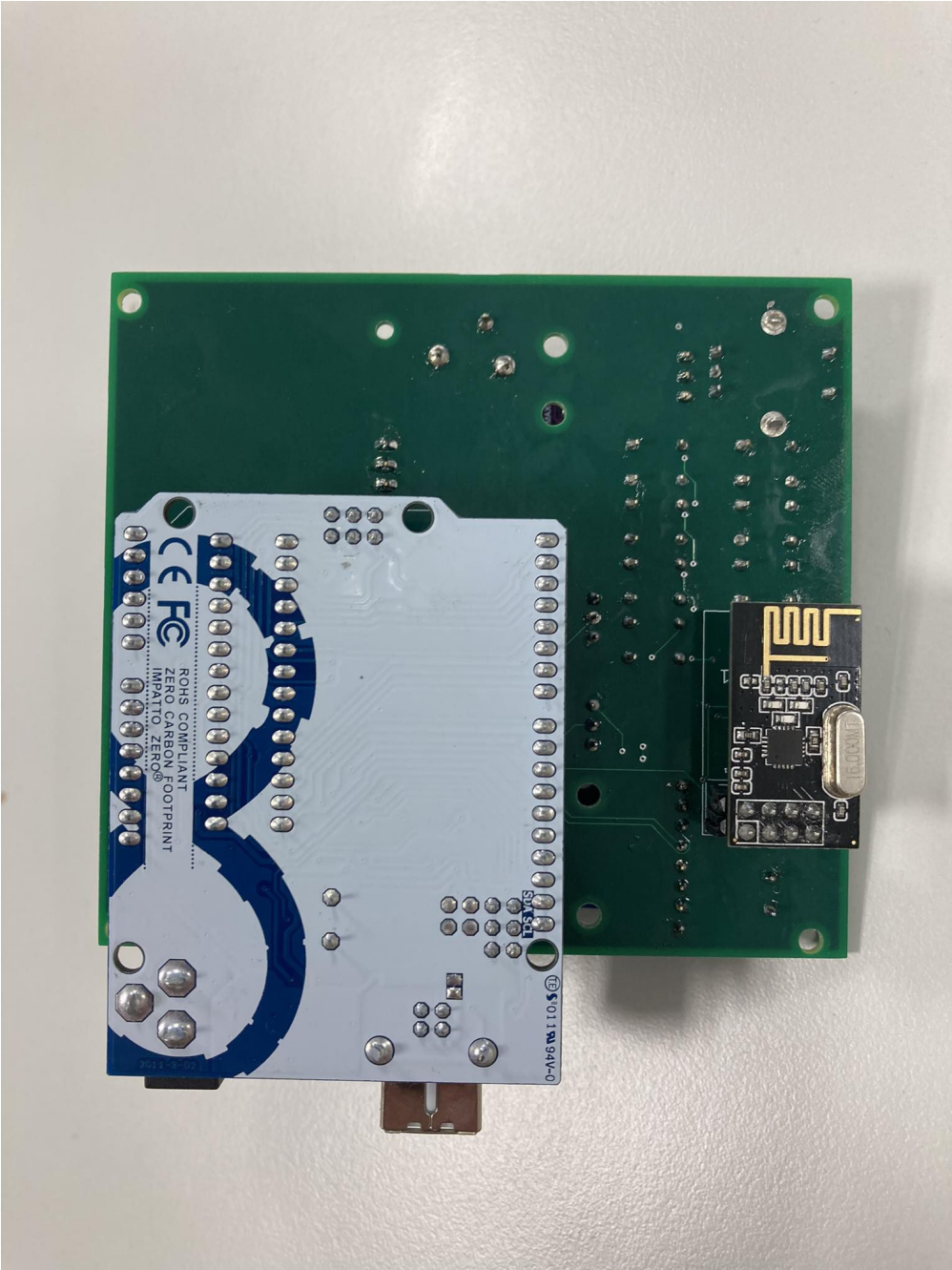




## A.6 Fotografia zariadenia TOP



# A.7 Fotografia zariadenia BOTTOM



## B Obsah elektronickej prílohy

```
Prilohy ..... koreňový dreskár priloženého archívu
├── DPS ..... zložka s projektom v Eagle
│   ├── HW_modul_schema.sch
│   ├── HW_modul_schema.brd
│   ├── breakout_PDmodul_FINAL.lbr
│   ├── CAMOutputpusHWmodul ..... zložka obsahujúca Gerber files
│   └── ...
├── PD_test ..... zložka obsahujúca testovacie patche v Pure Data
│   ├── PD_test_input.pd
│   └── PD_test_output.pd
└── Arduino_test ..... zložka obsahujúca testovacie patche v ArduinoIDE
    ├── Joystick.ino
    ├── HY-SRF05.ino
    ├── toggle.ino
    ├── GPIO_BTNs.ino
    ├── GPIO_LEDs.ino
    ├── MPU6050.ino
    ├── Electret_mic.ino
    ├── Potentiometer.ino
    ├── Encoder.ino
    ├── Piezo.ino
    ├── Wifi_trans.ino
    └── Wifi_rec.ino
```