



## **Bakalářská práce**

# **Využití průmyslového robota pro 3D aditivní tisk**

*Studijní program:*

B2612 Elektrotechnika a informatika

*Studijní obor:*

Elektronické informační a řídicí systémy

*Autor práce:*

**Aleš Smahel**

*Vedoucí práce:*

Ing. Tomáš Martinec, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



## Zadání bakalářské práce

# Využití průmyslového robota pro 3D aditivní tisk

<i>Jméno a příjmení:</i>	<b>Aleš Smahel</b>
<i>Osobní číslo:</i>	M18000064
<i>Studijní program:</i>	B2612 Elektrotechnika a informatika
<i>Studijní obor:</i>	Elektronické informační a řídicí systémy
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2021/2022

### Zásady pro vypracování:

1. Seznamte se s problematikou 3D tisku s využitím robotů, popište možnosti využití standardních průmyslových robotů pro realizaci aditivního 3Dtisku. Popište pravděpodobná slabá místa a limity takového řešení a najděte existující HW i SW komponenty, které by bylo možné k realizaci využít.
2. Popište hlavní uzly navrhovaného řešení (extruder, hotend, interpretace G-kódů atd.) a vytvořte ze zvolených komponent kompletní řešení tiskové hlavy - zapojte elektrickou část, nakonfigurujte a případně upravte firmware pro řízení tiskové hlavy, ověřte funkci extrudéru a hotendu.
3. Navrhněte mechanickou konstrukci pro připevnění tiskové hlavy a výrobku, realizujte potřebný software a hardware pro připojení a komunikaci tiskové hlavy a robota.
4. Připravte potřebný software pro robota a otestujte realizované řešení na vhodných modelech, popište jeho vlastnosti a limity, navrhněte možná další vylepšení systému.

*Rozsah grafických prací:* dle potřeby dokumentace  
*Rozsah pracovní zprávy:* 30-40 stran  
*Forma zpracování práce:* tištěná/elektronická  
*Jazyk práce:* Čeština

### **Seznam odborné literatury:**

- [1] KUKA System Software 8.3, Návod k obsluze a programování pro konečné uživatele, KUKA Roboter GmbH
- [2] Canessa, E.; Fonda, C.; Zennaro, M. Low - Cost 3D printing for science, education & sustainable development, 2013, [online] <http://sdu.ictp.it/3D/book.html>
- [3] Průša, J.; Průša, M. Základy 3D tisku, 2014, [online] <http://www.prusa3d.cz/kniha-zaklady-3d-tisku-josefa-prusi/>

*Vedoucí práce:* Ing. Tomáš Martinec, Ph.D.  
Ústav mechatroniky a technické informatiky

*Datum zadání práce:* 22. listopadu 2021  
*Předpokládaný termín odevzdání:* 22. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Josef Černohorský, Ph.D.  
vedoucí ústavu

## Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje domých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2023

Aleš Smahel

## **Poděkování**

Děkuji Ing. Tomáši Martincovi, Ph.D. za pomoc při vedení a konzultace bakalářské práce.  
Mé poděkování patří též Ing. Ivo Brixímu za další konzultace a Bc. Miroslavě Vejčíkové za pomoc při gramatické kontrole práce.

# Využití průmyslového robota pro 3D aditivní tisk

## Anotace

Práce je zaměřena na téma 3D tisku s využitím robota. První kapitola se zabývá návrhem a realizací hardwarového propojení řízení od firmy Arduino pro podávání materiálu a robota od firmy KUKA. V druhé kapitole je popsána úprava firmwaru pro dané řízení a to z původních *G* kódů na výměnu signálů mezi tímto řízením a samotným robotem. V předposlední kapitole byla provedena rešerše na téma použití vhodného překladače z *G* kódu do jazyka Kuka Robot Language (KRL) a následně došlo k výměru jednoho z navrhovaných řešení, které bylo dále upraveno pro potřeby této práce. Poslední kapitola obsahuje vlastní zkoušku 3D tisku pomocí robota.

## Klíčová slova

3D tisk, analogový převodník, Arduino, digitální převodník, *G* kód, KUKA, robot

# **The use of an industrial robot for 3D additive printing**

## **Annotation**

The work is focused on the topic of 3D printing using a robot. The first chapter deals with the design and implementation of a hardware control connection from Arduino for material feeding and a robot from KUKA. The second chapter describes the modification of the firmware for the given control, from the original G codes for the exchange of signals between this control and the robot itself. In the penultimate chapter, research was conducted on the topic of using a suitable translator from G code to KUKA Robot Language (KRL) and subsequently one of the proposed solutions was selected, which was further modified for the needs of this work. The final chapter contains a self-examination of 3D printing using a robot.

## **Keywords**

3D printing, analog converter, Arduino, digital converter, *G* code, KUKA, robot

# Obsah

Seznam obrázků .....	9
Úvod.....	11
1. Návrh HW .....	12
1.1 Digitální převodník 24 V/5 V.....	12
1.2 Digitální převodník 5 V/24 V.....	17
1.3 Analogový převodník 0-10 V/0-5 V (AI karta) .....	19
1.4 Napájení analogového převodníku .....	25
1.5 Návrh desky pro řídicí elektroniku.....	28
1.6 Cenová kalkulace.....	31
1.7 Vyhřívaná podložka .....	31
2. Firmware (FW) pro řízení 3D tisku.....	33
2.1 Vytvoření FW v konfigurátoru .....	33
2.2 Úprava FW.....	34
2.3 Nastavení rychlosti .....	36
2.4 Nastavení teploty extruderu .....	40
2.5 Čtení a zpracování analogových hodnot.....	41
3. Software (SW) pro robota.....	45
4. Vlastní zkouška 3D tisku .....	49
Závěr.....	50
Seznam použité literatury.....	52
Seznam příloh na CD .....	55



## Seznam obrázků

Obr. 1.1 Vstupní obvod procesoru, interní pull-up rezistor $R_{pu}$ je spínáný tranzistorem [1].....	14
Obr. 1.2 Schéma vnitřního připojení PC817 [2].....	14
Obr. 1.3 Doporučené zapojení PC817 od výrobce [2] .....	15
Obr. 1.4 Schéma zapojení převodníku 24 V/5 V pro 2 kanály a napájení karty .....	16
Obr. 1.5 DPS převodníku 24 V/5 V (vlevo), výsledný převodník (vpravo).....	16
Obr. 1.6 Příklad zapojení pull-down rezistoru .....	18
Obr. 1.7 Schéma zapojení převodníku 5 V/24 V pro 2 kanály a napájení karty .....	18
Obr. 1.8 DPS převodníku 5 V/24 V (vlevo), výsledný převodník (vpravo).....	19
Obr. 1.9 Schéma vnitřního připojení IL300 [5].....	20
Obr. 1.10 Příklad doporučeného zapojení optočlenu HCNR200 [6].....	21
Obr. 1.11 Zjednodušený vstupní obvod .....	23
Obr. 1.12 Schéma zapojení analogového převodníku 0-10 V/0-5 V a napájení karty .....	25
Obr. 1.13 DPS analogového převodníku 0-10 V/0-5 V (vlevo), výsledný převodník (vpravo).....	25
Obr. 1.14 Doporučené zapojení stabilizátoru 78XX [9].....	26
Obr. 1.15 Schéma zapojení stabilizátoru 7805 .....	27
Obr. 1.16 DPS stabilizátoru 5 V (vlevo), výsledný stabilizátor (vpravo).....	27
Obr. 1.17 Rozvržení jednotlivých komponent na desce.....	28
Obr. 1.18 Schéma zapojení napěťových úrovní .....	28
Obr. 1.19 Schéma zapojení pro výměnu signálů .....	29
Obr. 1.20 Osazená deska pro řídicí elektroniku .....	29
Obr. 1.21 Montáž osazené desky na robota.....	30
Obr. 1.22 Prodloužený držák extruderu .....	30
Obr. 1.23 Znázornění odporu na vodičích.....	32
Obr. 2.1 Hot end MK8.....	34
Obr. 2.2 DRV8825 pinout .....	35
Obr. 2.3 Mikrokrokování proudového průběhu [13].....	35
Obr. 2.4 Tabulka nastavení mikrokrokování podle zapojených pinů M0 – M2 [13] .....	36
Obr. 2.5 Průběh frekvence z původního řízení.....	38
Obr. 2.6 Registr TIMSK1 (Timer/Counter 1 Interrupt Mask Register) [1].....	39
Obr. 2.7 Stavový registr SREG [1].....	40
Obr. 2.8 Registr výběru multiplexoru pro ADC [1] .....	41
Obr. 2.9 Volby referenčního napětí pro ADC [1].....	42
Obr. 2.10 ADC kontrolní a stavový registr B [1] .....	42

Obr. 2.11 ADC kontrolní a stavový registr A [1].....	42
Obr. 2.12 Tabulka dělicího faktoru [1].....	42
Obr. 2.13 Tabulka hodnot analogového vstupu 1 - rychlost .....	43
Obr. 2.14 Graf rychlosti v závislosti na výstupním napětí analogového převodníku AI1 .....	44
Obr. 2.15 Tabulka hodnot analogového vstupu 2 – teplota.....	44
Obr. 2.16 Graf rychlosti v teploty na výstupním napětí analogového převodníku AI2 .....	44
Obr. 3.1 Vizualizace zvolené sestavy v programu RoboDK.....	46
Obr. 4.1 První test 3D tisku.....	49

## Úvod

Tato práce navazuje na ročníkový projekt 3D tisk pomocí robota, kde bylo cílem seznámit se s aditivní technologií pro 3D tisk. Byla provedena rešerše existujících 3D projektů s využitím robotů. Na základě získaných zkušeností bylo navrženo vlastní řešení pro 3D tisk s použitím robota. V návrhu byla aplikována stejná konstrukce jako u uvedených řešení a to taková, že robot nese tiskovou hlavu a tisk byl prováděn na pevnou podložku. Pro vlastní řešení byla vybrána standartní 3D tisková hlava MK8 (Obr. 2.1) používaná v běžných 3D tiskárnách. Dále byl vyroben držák pro montáž na robotické rameno. Použitý robot byl KR16-2 s nosností 16 kg od firmy Kuka s řídicím systémem KRC4, který byl součástí vybavení laboratoře. Byla navržena a vyrobena deska učená pro řídicí elektroniku a vybráno její umístění na druhém rameni robota. K volbě daného umístění došlo záměrně, z důvodů předpřipravených montážních otvorů od výrobce a vyvedených konektorů pro napájení a výměnu signálů. K řízení motoru extruderu byl vybrán modul Arduino s nástavbou RAMPS 1.6 a driverem krokového motoru DRV8825. Jako základ pro vývoj firmware (FW) byl zvolen FW pro řízení běžných 3D tiskáren od firmy Repetier.

Jedním z cílů této práce bylo upravit tento FW pro použitý hardware (HW) a dále jej přizpůsobit na jiný typ komunikace, než byl navržený pro původní 3D tiskárnu. Pro komunikaci mezi Arduinem a robotem bylo zvoleno použití digitálních a analogových vstupů/výstupů. Pro tento způsob komunikace bylo potřeba navrhnout a vyrobit převodníkové karty. Tyto karty, řídicí modul Arduino, napájecí zdroj a další komponenty bylo následně nutné umístit na desku pro řídicí elektroniku.

Předposlední kapitola pojednává o návrhu možného řešení překladu z *G* kódu do jazyka KRL. Popisuje jednotlivá existující řešení, vypisuje klady a zápory těchto řešení. Dle těchto parametrů byl zvolen jeden způsob řešení, který byl následně přizpůsoben pro danou aplikaci.

V poslední kapitole byl proveden samotný experiment tisku pomocí robota.

## 1. Návrh HW

Pro vlastní komunikaci mezi robotem a Arduinem, který je osazen procesorem Atmel ATmega 2560 byla navržena digitální a analogová výměna signálů. Z důvodu rozdílných napěťových úrovní pro digitální komunikaci - na straně robota to bylo 24 V a na straně procesoru to bylo 5 V. Bylo potřeba navrhnout a vyrobit převodní kartu těchto signálů jak pro komunikaci z robota do procesoru tak z procesoru do robota. U analogové komunikace to bylo obdobné - na straně robota bylo možné volit napěťový nebo proudový signál a na straně procesoru byl jen vstupní analogově-digitální převodník (ADC) převodník 0 - 5 V. Procesor nebyl vybaven vestavěným digitálně-analogovým převodníkem (DAC), ale byl by schopen pulzně šířkovou modulací (PWM) digitálního signálu dosáhnout některých funkcí analogového výstupu. Z důvodu absence analogového výstupu v této práci, nebylo již nutné se touto problematikou dále zabývat. Všechny návrhy obvodů a zpracování desek plošných spojů (DPS) byly vytvořeny v programu *Eagle*, v příloze byly vloženy podklady pro výrobu.

### 1.1 Digitální převodník 24 V/5 V

Z pohledu řízení Arduina – procesoru Atmel, se jednalo o vstupní převodník (DI karta). Pro danou změnu napětí by bylo možné použít více způsobů. Jeden z nich byl odporový dělič napětí, který představuje jednoduchý způsob, kdy dojde k rozdělení napětí dle úbytků napětí na rezistorech. Při návrhu děliče se počítalo s nezátíženým děličem, protože vstupní odpor na vstupu procesoru je vysoký a tudíž by byl jeho odběr zanedbatelný. Napětí děliče na výstupu by kopírovalo úměrně napětí na vstupu do děliče a pro případné přepětí na vstupu by musely být ještě ošetřeny výstupní svorky tohoto převodníku, aby se zabránilo poškození digitálních vstupů procesoru, jinak by došlo k přenosu tohoto přepětí i na výstup děliče. Přes celý dělič by protékal proud  $I$  a tímto proudem by byly zatíženy jednotlivé rezistory. Pro návrh děliče bylo nutné počítat se ztrátou.

$$P = U_R \times I \quad [W; V, A] \quad (1.1)$$

$P$  – ztrátový výkon,  $U_R$  – úbytek napětí na rezistoru,  $I$  – protékající proud  
Minimálně pro tento ztrátový výkon by měly být rezistory zvoleny.

Další variantou by bylo využití optočlenu, který by bylo možné zapojit obecně dvěma různými způsoby – jako plné galvanické oddělení nebo se společným uzemněním. První způsob by představoval úplně oddělené obvody a to i jejich uzemnění. Aby tato varianta splnila svůj účel, musela by být dodržena i izolační vzdálenost na desce plošných spojů mezi vodiči na vstupní a výstupní straně obvodu. U druhého způsobu zapojení optočlenu by byla jeho vstupní dioda i výstupní tranzistor připojený na společnou zem. Tímto způsobem by nebylo dosaženo plného galvanického oddělení, ale i tak by byla zajištěna dostatečná ochrana vstupu procesoru – v případě příliš velkého napětí na vstupu (například z důvodu chyby v připojení nebo vlivem elektromagnetického pulsu z vnějšího prostředí) by mohlo dojít ke zničení diody na vstupu optočlenu, ale vstup procesoru by byl chráněn. Samotný optočlen by bylo také dobré chránit před případným přepětím. Pro dosažení této ochrany by bylo použito paralelní zapojení k diodě v optočlenu opačně polarizovanou Zenerovou diodou. Při návrhu by bylo také nutné zvolit sériový odpor k diodě optočlenu dle vzorce.

$$R = \frac{U_{IN} - U_D}{I_D} \quad [\Omega; V, A] \quad (1.2)$$

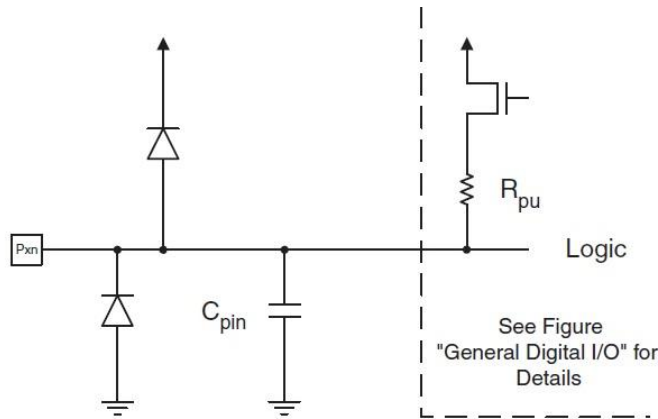
R – hodnota rezistoru na vstupu,  $U_{IN}$  – vstupní napětí,  $U_D$  – úbytek napětí na diodě dle katalogu,  $I_D$  – proud diodou – katalogová hodnota

V optočlenu bývá často zapojen NPN tranzistor, který spíná proti zemi, a proto by bylo potřeba použít pull-up rezistor (Obr. 1.1). Funkcí pull-up rezistor bývá rozuměno, že by byl mezi kladné napájecí napětí a vstupní pin do procesoru připojen rezistor. Nevýhodou tohoto řešení ale bývá, že je vstupní signál do procesoru negovaný. Velikost tohoto vnitřního rezistoru se liší a závisí na použitém typu procesoru, jeho hodnota nabývá velikosti od 20 do 150 k $\Omega$ .

$$R_{MAX} = \frac{U_{INarduinio}}{I_{TRmax}} \quad [\Omega; V, A] \quad (1.3)$$

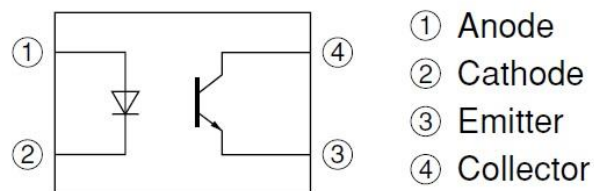
$R_{MAX}$  – maximální hodnota pull-up rezistoru,  $U_{INarduinio}$  – vstupní napětí do procesoru, obvykle 5V nebo 3,3V,  $I_{TRmax}$  – proud tranzistorem v optočlenu – katalogová hodnota.

Nejčastěji zvolenou hodnotou tohoto rezistoru bývá 10 k $\Omega$ . Tato součástka může být vynechána, pokud by byl použit vnitřní pull-up rezistor. K tomu slouží u procesoru příkaz `pinMode(pin, INPUT_PULLUP)`.



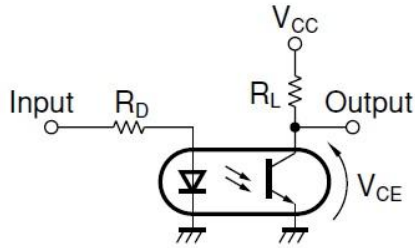
Obr. 1.1 Vstupní obvod procesoru, interní pull-up rezistor  $R_{pu}$  je spínaný tranzistorem [1]

Z dostupných informací byla nejlepší volba pomocí optočlenu, kde byly galvanicky oddělené obvody. Díky tomu by se nepřenesla případná porucha na výstup tohoto optočlenu a nebylo proto nutné jej doplňovat další ochranou. Samotný optočlen nebyl chráněn Zenerovou diodou. Výsledná deska byl prototyp a do finální podoby by bylo vhodné ji zahrnout. Jako optočlen byla použita řada PC8x7, přesněji model PC817 z důvodu dobré dostupnosti na trhu. Ostatní modely (PC827, PC837, PC847) bylo možné také použít, ale jejich zdánlivá výhoda (jsou zapouzdřeny od jednoho až po 4 optočlenu), by mohla být v případě poškození jednoho optočlenu spíše nevýhodou.



Obr. 1.2 Schéma vnitřního připojení PC817 [2]

Na vstupu obvodu, kde bylo vstupní napětí 24 V, bylo nutné dle doporučeného zapojení od výrobce doplnit odpor (Obr. 1.3). Ten byl vypočten dle vzorců (1.2) a (1.1). Úbytek napětí na diodě  $U_D = 1,2$  V a proud  $I_D$  byl zvolen 10 mA. [2]



Obr. 1.3 Doporučené zapojení PC817 od výrobce [2]

$$R = \frac{U_{IN} - U_D}{I_D} = \frac{24 - 1,2}{0,01} = 2280 \Omega$$

$$P = (U_{IN} - U_D) \times I_D = (24 - 1,2) \times 0,01 = 0,228 W$$

Nejbližší hodnota rezistoru ze standardní řady E24 byla 2k2, z toho důvodu byl vybrán rezistor s touto hodnotou. Dále byl obvod vybaven indikací vstupního napětí žlutou LED diodou, výpočet odporu pro tyto diody byl proveden podle stejných vzorců (1.2) a (1.1). Úbytek napětí na této diodě činil dle katalogu  $U_{LED} = 2,1 V$  [3]

$$R_{LED} = \frac{U_{IN} - U_{LED}}{I_{LED}} = \frac{24 - 2,1}{0,01} = 2190 \Omega$$

$$P = (U_{IN} - U_{LED}) \times I_{LED} = (24 - 2,1) \times 0,01 = 0,219 W$$

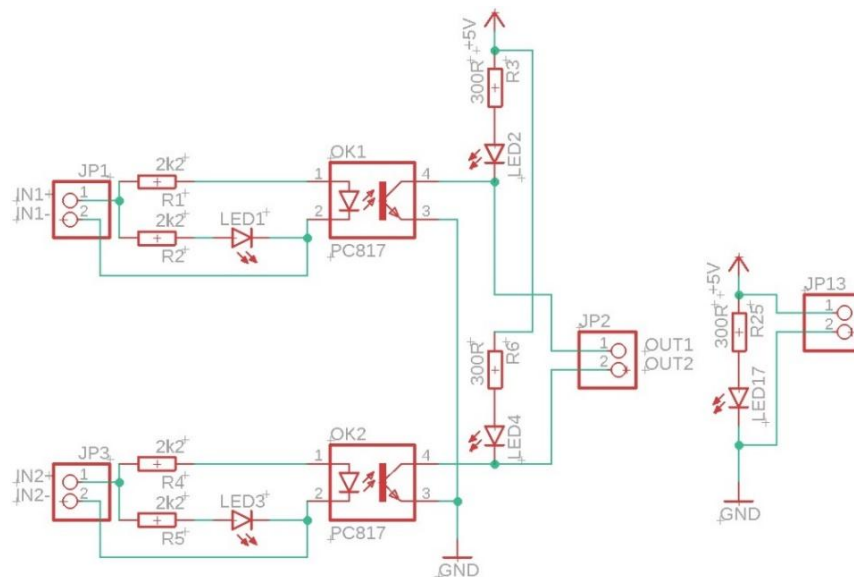
Na výstupu obvodu, kde bylo požadované napětí 5 V, byla také doplněna indikace žlutou LED diodou. Pro tuto diodu byl předřazen rezistor. Hodnota odporu tohoto rezistoru byla získána dle vzorců (1.2) a (1.1).

$$R = \frac{U_{OUT} - U_D}{I_D} = \frac{5 - 2,1}{0,01} = 290 \Omega$$

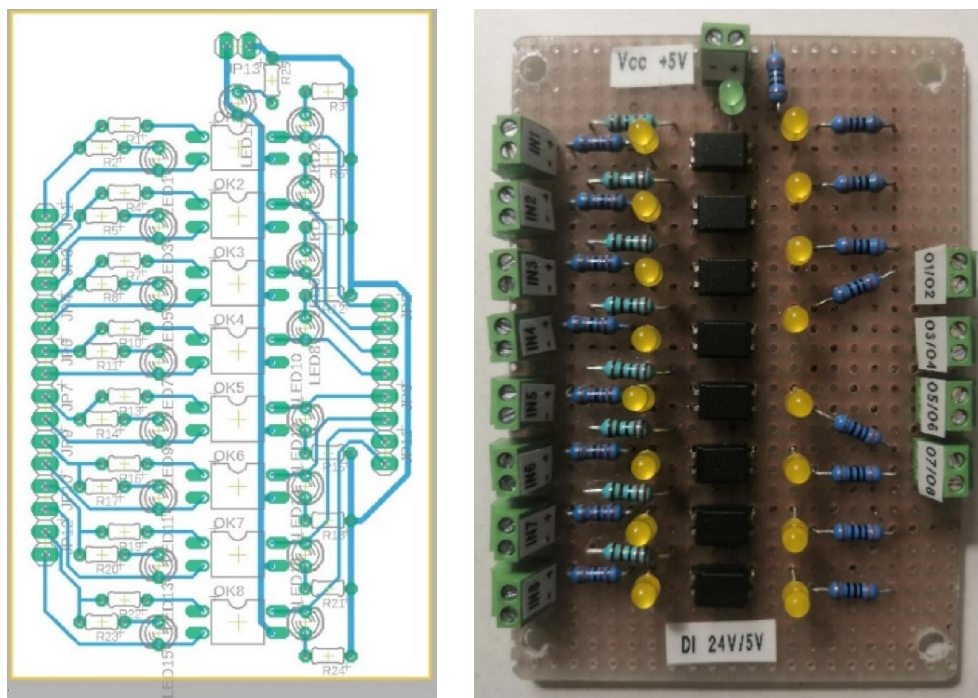
$$P = (U_{OUT} - U_D) \times I_D = (5 - 2,1) \times 0,01 = 0,029 W$$

Ve standardní řadě E24 byla nejbližší hodnota rezistoru 300R. Ke správné funkci obvodu bylo nutné jej doplnit napájením na výstupní straně optočlenu. Přítomnost napětí byla opět signalizována LED diodou, ale v tomto případě zelenou. Úbytek napětí na této diodě byl stejný jako u žluté a to  $U_{LED} = 2,1 V$  [4]. Předřazený rezistor byl opět vypočten pomocí známých vzorců (1.2) a (1.1). Zde byl použit rezistor s hodnotou odporu 300R, protože byl jeho výpočet stejný jako v předešlém případě. Podle vypočtených hodnot výkonového zatížení na jednotlivých rezistorech byla zvolena řada 0,6 W a hodnota tolerance odporu je 1%. Převodník

byl navržen jako osmi kanálový a pro tuto aplikaci byly využity tři kanály. Zbylé kanály byly ponechány jako rezerva pro případné další signály z robota.



Obr. 1.4 Schéma zapojení převodníku 24 V/5 V pro 2 kanály a napájení karty



Obr. 1.5 DPS převodníku 24 V/5 V (vlevo), výsledný převodník (vpravo)



## 1.2 Digitální převodník 5 V/24 V

Tento převodník byl naopak výstupní (DO karta) z pohledu Arduino modulu (procesoru Atmel). Pro tuto aplikaci byl použit obdobný návrh jako u převodníku 24 V/5 V, s tím rozdílem, že byla provedena změna hodnot rezistorů odpovídající napěťové úrovni. Byl použit opět optočlen PC817 a jeho doporučené zapojení (Obr. 1.3). Na vstupním obvodu byla pro rezistor, k indikaci přítomnosti signálu, zvolena hodnota odporu 300R a na výstupním obvodu hodnota odporu 2k2. Zbývalo tedy navrhnout hodnotu odporu rezistoru na vstupním obvodu, jako to uváděl doporučený návrh výrobce. Pro výpočet odporu byly použity vzorce (1.2) a (1.1). Katalogová hodnota pro úbytek napětí na diodě byla  $U_D = 1,2V$  a proud  $I_D$  byl zvolen 10 mA [2].

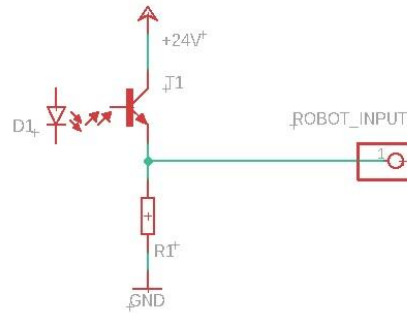
$$R = \frac{U_{IN} - U_D}{I_D} = \frac{5 - 1,2}{0,01} = 380 \Omega$$

$$P = (U_{IN} - U_{LED}) \times I_{LED} = (5 - 1,2) \times 0,01 = 0,038 W$$

Pro požadovanou hodnotu odporu 380  $\Omega$  byla v řadě E24 nejbližší hodnota 390R. Výkonová zátěž rezistorů byla taktéž zvolena stejná jako u převodníku DI karty, a to řada s hodnotou 0,6 W a hodnotou tolerance 1% odporu. Po odzkoušení na nepájivém poli bylo zjištěno, že po připojení do vstupu robota se výstup z karty choval jako odporový dělič. Problém byl v návrhu, který počítal s vnitřním pull-up odporem na straně robota a robot měl na vstupu vnitřní pull-down rezistor (Obr. 1.6). Tento pull-down rezistor byl připojen mezi vstup robota a potenciál GND. Jeho funkce spočívala v tom, že utlumila případné rušivé signály na potenciál země a při nezapojeném stavu byla na vstupu stále log. 0. Návrh byl proto přepracován, tak aby nedocházelo k aktivaci vstupu robota, který by nezáležel na stavu výstupu z převodníku. Pro tuto úpravu bylo i nutné přepočítat hodnotu rezistoru. Vzorec (1.2) byl přizpůsoben tomuto zapojení, saturační napětí  $U_{CE} = 0,1 V$ .

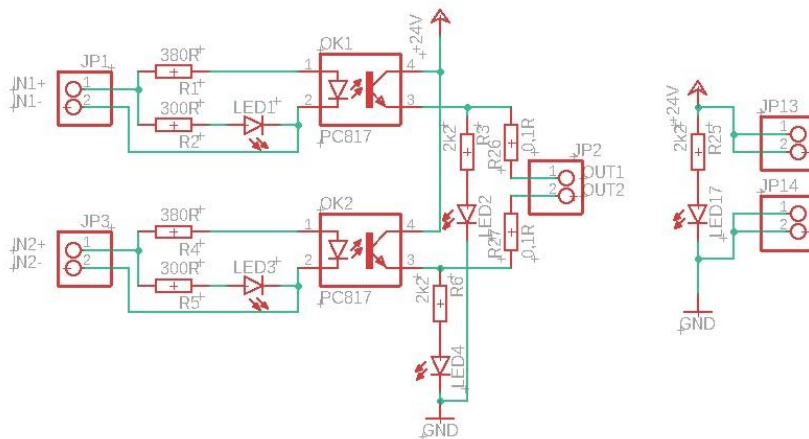
$$R_{LED} = \frac{U_{IN} - U_{CE} - U_D}{I_{LED}} = \frac{24 - 0,1 - 2,1}{0,01} = 2180 \Omega$$

$$P = (U_{IN} - U_{CE} - U_D) \times I_{LED} = (24 - 0,1 - 2,1) \times 0,01 = 0,218 W$$

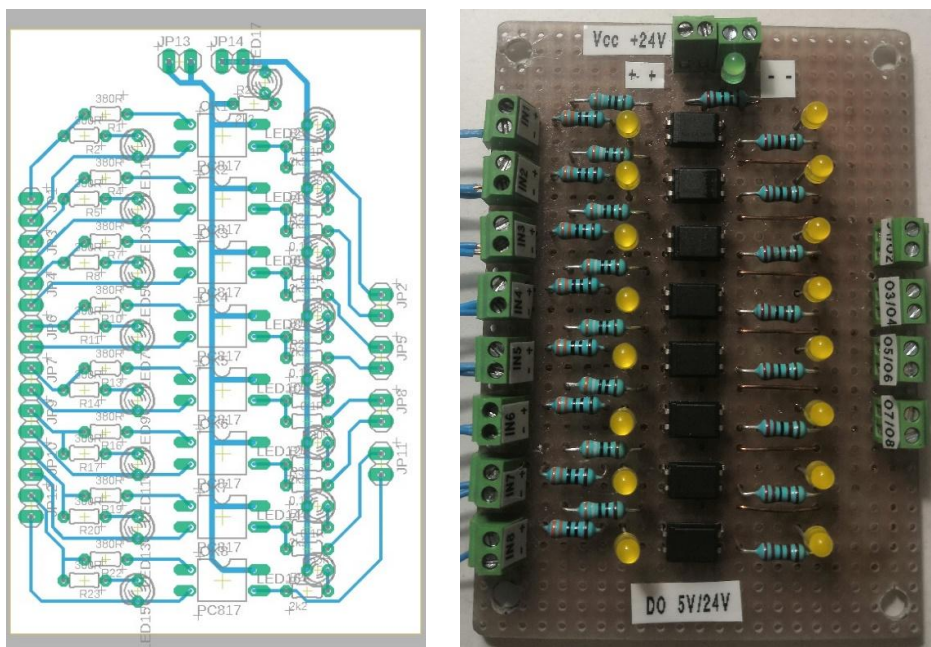


Obr. 1.6 Příklad zapojení pull-down rezistoru

Výsledný návrh po úpravě byl uvedený na následujícím obrázku (Obr. 1.7). Hodnota odporu výstupního rezistoru se po úpravě příliš nezměnila, a proto byl ponechán původní rezistor. Rezistory s hodnotou 0,1R byly jen náhrady drátových propojek pro zakreslení v programu *Eagle*. Převodník byl navržen jako osmi kanálový a pro tuto aplikaci byl využit jeden kanál. Zbylé kanály byly ponechány jako rezerva pro případné další signály z robota.



Obr. 1.7 Schéma zapojení převodníku 5 V/24 V pro 2 kanály a napájení karty



Obr. 1.8 DPS převodníku 5 V/24 V (vlevo), výsledný převodník (vpravo)

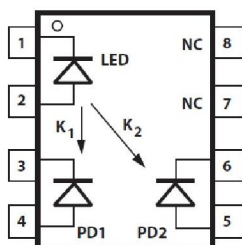
### 1.3 Analogový převodník 0-10 V/0-5 V (AI karta)

Pro analogový výstup měl robot k dispozici více nastavení rozsahů, a to jak proudové, tak napěťové. V případě volby proudového analogového výstupu byly k dispozici od různých výrobců standartní rozsahy 0-20 mA, 4-20 mA nebo -10 mA - +10 mA. V případě volby napěťového výstupu byly obvykle dvě možnosti, a to -10 V - +10 V nebo 0-10 V. Robot v laboratoři byl osazen modulem, který měl napěťový rozsah 0 - 10V. Na druhé straně měl procesor na vstupním analogovém pinu rozsah 0-5 V. Pro správnou funkci analogové komunikace bylo proto nutné navrhnout také převodník mezi těmito napěťovými úrovněmi. Požadavky byly následující: spojitý průběh přenosu signálu ze vstupu na výstup a co nejmenší ovlivnění výstupního obvodu při poruše vstupního obvodu.

Na základě stanovených požadavků by bylo možné navrhnout odporový dělič. Tento způsob převodu napětí by byl jednoduchý a splňoval první požadavek, ale v případě přepětí na vstupních svorkách by došlo k přenosu na výstupní svorky a k následnému poškození analogového vstupu procesoru. Dalším způsobem by bylo opětovné použití optočlenu, v tomto případě analogového. Mezi jeho vlastnosti by měla být lineární přenosová charakteristika v celém rozsahu požadovaného přenosu napětí, velká šířka pásma přenosu a vysoká stabilita

zisku přenosu. Z uvedených požadavků byla na trhu velká nabídka od různých výrobců např. optočlen HCNR200(HCNR201) od firmy Broadcom nebo optočlen IL300 od firmy Vishay semiconductors.

Znovu byl zvolen optočlen, protože splňuje oba požadavky pro správnou funkci převodníku. Po prostudování dokumentace od zmíněných optočlenů, bylo rozhodnuto pro optočlen IL300, vzhledem k jeho snadné dostupnosti na českém trhu a také k výši jeho pořizovací ceny. V tomto optočlenu byla výrobcem použita LED dioda, která osvětluje dvě fotodiody. První vstupní fotodioda (PD1) bývá nejčastěji zapojena do zpětné vazby a pro danou LED diodu generuje řídicí signál. Výstupní fotodioda (PD2) vytváří výstupní signál (proud), který je lineární a odpovídá světelnému toku LED diody.



Obr. 1.9 Schéma vnitřního připojení IL300 [5]

Hodnota  $K_1$  je poměr vstupního proudu fotodiody  $I_{PD1}$  k proudu LED diody  $I_F$  a jeho katalogová hodnota je  $K_1 = 0,007$ .

$$K_1 = \frac{I_{PD1}}{I_F} \quad [-; A, A] \quad (1.4)$$

Hodnota  $K_2$  je poměr výstupního proudu fotodiody  $I_{PD2}$  k proudu LED diody  $I_F$  a jeho katalogová hodnota je  $K_2 = 0,007$ .

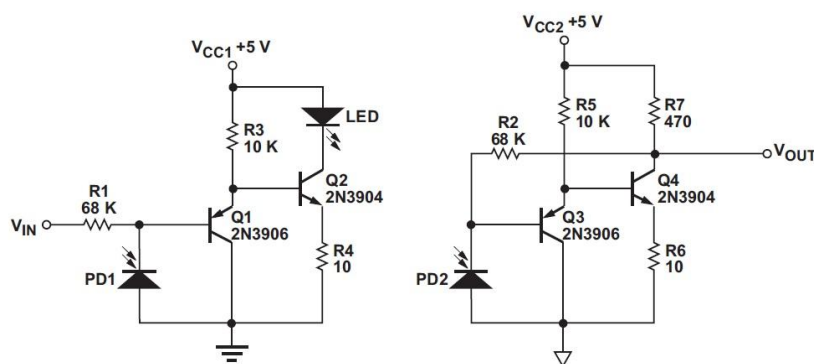
$$K_2 = \frac{I_{PD2}}{I_F} \quad [-; A, A] \quad (1.5)$$

Přenosový zisk  $K_3$  je poměr hodnot  $K_2$  a  $K_1$ . Jeho hodnota byla rovna 1.

$$K_3 = \frac{K_2}{K_1} \quad [-; -, -] \quad (1.6)$$

Základem návrhu pro daný převodník bylo zajistit, aby bylo kladné vstupní napětí neinvertované vůči výstupnímu napětí. Pro optimální funkci optočlenu bylo vhodné ke vstupní

diodě připojit externí zesilovač zpětné vazby tak, aby bylo dosaženo regulace světelného výkonu pro LED diodu. Na výstupní fotodiodu bylo také nutné zapojit externí zesilovač, aby byl výstupní proud převeden opět na napětí. Jako základní návrh bylo použito jedno z doporučených zapojení od výrobce (Obr. 1.10), které mělo ale na vstupním obvodu jen 5 V a následně bylo upraveno pro dané použití tj. na 10 V. Z ekonomického hlediska byly zesilovače nahrazeny tranzistory. Vstupní zesilovač byl tedy zkonstruován pomocí dvou tranzistorů Q<sub>1</sub> a Q<sub>2</sub> a dvou rezistorů R<sub>3</sub> a R<sub>4</sub>. Výstupní zesilovač se skládal také ze dvou tranzistorů Q<sub>3</sub> a Q<sub>4</sub> a dále ze tří rezistorů R<sub>5</sub>, R<sub>6</sub> a R<sub>7</sub>.



Obr. 1.10 Příklad doporučeného zapojení optočlenu HCNR200 [6]

Nová hodnota rezistoru R<sub>1</sub> pro upravený obvod byla navržena tak, aby byl splněn požadavek protékajícího proudu pro LED diodu tj. I<sub>F</sub> = 10 mA. Z rovnice

$$I_F = \frac{U_{IN}}{\frac{R_1}{K_1}} \quad [A; V, \Omega, -] \quad (1.7)$$

byl vypočten R<sub>1</sub> pro dané vstupní napětí U<sub>IN</sub> = 10 V

$$R_1 = \frac{U_{IN}}{I_F \times K_1} = \frac{10}{0,01 \times 0,007} = 142857 \Omega$$

Pro správnou volbu rezistoru bylo také nutné dopočítat jeho výkonovou ztrátu dle vzorce (1.1)

$$P = U_{IN} \times I_F = 10 \times 0,01 = 0,1 W$$

Požadovaný rezistor byl opět zvolen z řady E24, hodnota odporu byla 150 kΩ, maximální výkonová zátěž 0,6 W a hodnota tolerance 1% odporu.

Odpor  $R_2$  byl pak zvolen k dosažení požadovaného výstupního napětí  $U_{OUT} = 5 \text{ V}$  podle rovnice:

$$\frac{U_{OUT}}{U_{IN}} = \frac{R_2}{R_1} \quad [V, V; \Omega, \Omega] \quad (1.8)$$

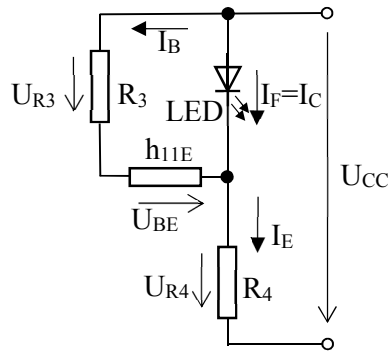
po úpravě pro výpočet  $R_2$ :

$$R_2 = R_1 \times \frac{U_{OUT}}{U_{IN}} = 150 \times 10^3 \times \frac{5}{10} = 75000 \Omega$$

V návrhu obvodu byl rezistor  $R_2$  vytvořen paralelní kombinací ze dvou rezistorů  $150 \text{ k}\Omega$ , z důvodu dostatečné zásoby této hodnoty. Výkonová ztráta byla spočítána také dle vzorce (1.1)

$$P = U_{OUT} \times I_F = 5 \times 0,01 = 0,05 \text{ W}$$

Rezistory  $R_4$  a  $R_6$  byly v obvodu použity pro zlepšení dynamické odezvy vstupních a výstupních obvodů, ale hodnota  $R_4$  byla upravena pro větší vstupní napětí než byl původní návrh. Návrh hodnoty tohoto rezistoru vycházel z doporučených katalogových hodnot pro daný tranzistor [7] a vstupní LED diodu optočlenu. Doporučený proud na vstupní LED diodě [5] se mohl pohybovat v rozmezí  $I_F = 1 \div 20 \text{ mA}$  a hodnota tomu odpovídajícího napětí  $U_{LED} = 1,32 \text{ V}$ . Dalšími parametry pro výpočet hodnoty rezistoru byly zesílení tranzistoru  $h_{21E} = 60$  a úbytek napětí  $U_{BE} = 0,95 \text{ V}$  z katalogu pro tranzistor Q2 [7]. Hodnoty z katalogu tranzistoru byly záměrně voleny o řád vyšší z důvodu neznámé přesné hodnoty protékajícího proudu přes kolektor  $I_C$ . Q2 byl zcela otevřen a Q1 zcela uzavřen, a proto bylo uvažováno jako by tranzistor Q1 v obvodu nebyl (Obr. 1.11). Celý návrh se tímto zjednodušil. Na jedné větvi tohoto vstupního obvodu byl rezistor  $R_3 = 10 \text{ k}\Omega$  a na druhé vstupní LED dioda optočlenu. Tranzistor Q2 byl ve zcela otevřeném stavu, a tak byl nahrazen uzlem.



Obr. 1.11 Zjednodušený vstupní obvod

$$I_E = I_C + I_B \quad [A; A, A] \quad (1.9)$$

$$I_C = I_B \times h_{21E} \quad [A; A, -] \quad (1.10)$$

Podle I. Kirchhoffova zákona [8] byla tedy sestavena rovnice pro výpočet proudu  $I_E$ , který protékal přes  $R_4$  a podle II. Kirchhoffova zákona [8] byla sestavena rovnice na výpočet  $R_4$

$$R_3 \times I_B + R_4(I_B \times h_{21E} + I_B) + U_{BE} = U_{CC} \quad [U; U, U, U] \quad (1.11)$$

Z původního obvodu pro napájecí napětí  $U_{CC} = 5 \text{ V}$  bylo zjištěno, že přes rezistor  $R_3$  procházel proud  $I_B$  vypočtený dle vzorce (1.11)

$$R_3 \times I_B + R_4 \times I_B \times h_{21E} + R_4 \times I_B = U_{CC} - U_{BE}$$

$$I_B(R_3 + R_4 \times h_{21E} + R_4) = U_{CC} - U_{BE}$$

$$I_B = \frac{U_{CC} - U_{BE}}{R_3 + R_4 \times h_{21E} + R_4} = \frac{5 - 0,95}{10000 + 10 \times 60 + 10} = 3,82 \times 10^{-4} \text{ A}$$

po dopočítání proudu  $I_E$  přes rezistor  $R_4$  dle vzorců (1.9) a (1.10)

$$I_E = I_C + I_B = I_B \times h_{21E} + I_B = I_B(h_{21E} + 1) = 3,82 \times 10^{-4}(60 + 1) = 2,33 \times 10^{-2} \text{ A}$$

Výsledný proud  $I_E$  přes  $R_4$  byl v původním obvodu cca 25 mA a tato hodnota byla požadována i na upraveném obvodu pro vyšší napětí potřebné pro danou aplikaci, aby bylo dosaženo stejných vlastností dynamické odezvy vstupního obvodu.

Pro výpočet nové hodnoty  $R_4$  byl opět použit vzorec (1.11)

$$R_4 = \frac{U_{CC} - U_{BE} - R_3 \times I_B}{I_B \times h_{21E} + I_B} = \frac{10 - 0,95 - 1000 \times 3,82 \times 10^{-4}}{0,025} = 209 \Omega$$

Nejbližší vyšší hodnota odporu v řadě E24 pro vypočtenou hodnotu  $209 \Omega$  byla hodnota rezistoru 220R. Nakonec bylo ještě nutné dopočítat výkonovou ztrátu pro daný rezistor dle vzorce (1.1), který byl upraven pro známý proud  $I_E$  a hodnotu rezistoru  $R_4$

$$P = R_4 * I_E^2 = 220 * 0,0025^2 = 0,1375 W$$

Vybraný rezistor splňoval parametry zvolené řady, jak odporové hodnoty, tak i výkonové ztráty s hodnotou tolerance 1% odporu.

Hodnoty rezistorů  $R_3$  a  $R_5$  byly ponechány z původního návrhu a jejich funkcí bylo zajišťovat dostatečný proud pro tranzistory  $Q_2$  a  $Q_4$ . Rezistor  $R_7$  byl ponechán z původního návrhu a jeho hodnota byla navržena tak, že tranzistor  $Q_4$  měl přibližně stejný kolektorový proud jako tranzistor  $Q_2$ .

Celý obvod byl ještě doplněn o signalizaci přítomnosti napájecího napětí, a to jak na vstupním obvodu, tak na výstupním obvodu. Tato signalizace byla indikována zelenou LED diodou, napětí na této diodě bylo dle katalogu  $U_{LED} = 2,1 V$  a proud  $I_{LED} = 10 mA$  [4], pro každý obvod zvlášť a bylo nutné dopočítat hodnoty ochranných rezistorů dle vzorců (1.2) a (1.1).

Výpočet hodnoty rezistoru pro napájení vstupního zesilovače:

$$R = \frac{U_{IN} - U_{LED}}{I_{LED}} = \frac{10 - 2,1}{0,01} = 790 \Omega$$

$$P = (U_{IN} - U_{LED}) \times I_{LED} = (10 - 2,1) \times 0,01 = 0,079 W$$

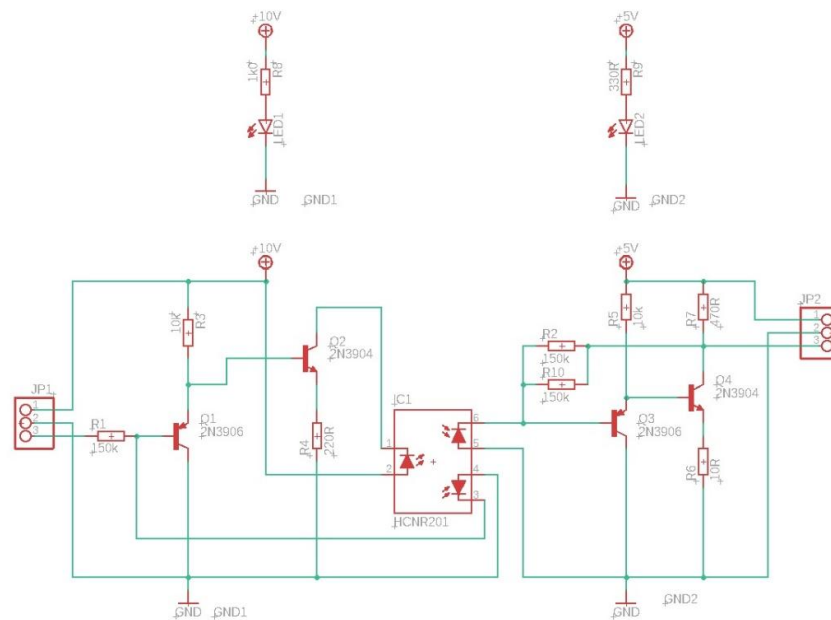
Výpočet hodnoty rezistoru pro napájení výstupního zesilovače:

$$R = \frac{U_{OUT} - U_{LED}}{I_{LED}} = \frac{5 - 2,1}{0,01} = 290 \Omega$$

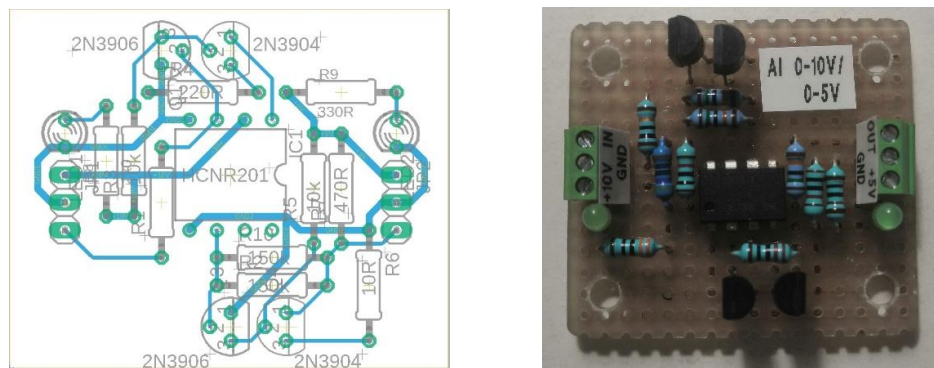
$$P = (U_{OUT} - U_{LED}) \times I_{LED} = (5 - 2,1) \times 0,01 = 0,029 W$$

Pro vstupní obvod byla vybrána hodnota rezistoru 1k0 z řady E24 a pro výstupní obvod byla vybrána hodnota 330R ze stejné řady. Výkonová ztráta splňovala mezní hodnoty 0,6 W pro tuto řadu a její hodnota tolerance odporu 1% také.





Obr. 1.12 Schéma zapojení analogového převodníku 0-10 V/0-5 V a napájení karty



Obr. 1.13 DPS analogového převodníku 0-10 V/0-5 V (vlevo), výsledný převodník (vpravo)

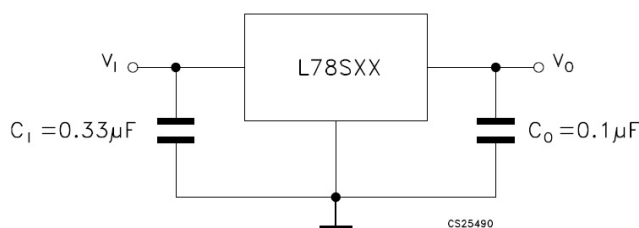
Analogový převodník byl jednokanálový a pro danou aplikaci bylo potřeba použít dva analogové převodníky, pro každý analogový signál zvlášť. Jeden analogový kanál byl použit pro přenos hodnoty rychlosti podávání materiálu z robota do extrudéru, a druhý pro přenos hodnoty teploty extrudéru.

#### 1.4 Napájení analogového převodníku

Analogová karta potřebovala dvě napájecí napětí, a to jak vstupního odvodu  $U_{CC1} = 10\text{ V}$ , tak výstupního obvodu  $U_{CC2} = 5\text{ V}$ . Pro napájení výstupního obvodu bylo možné použít napětí vyvedené z některých pinů procesoru. Na nástavbě RAMPS 1.6 to byl piny pro

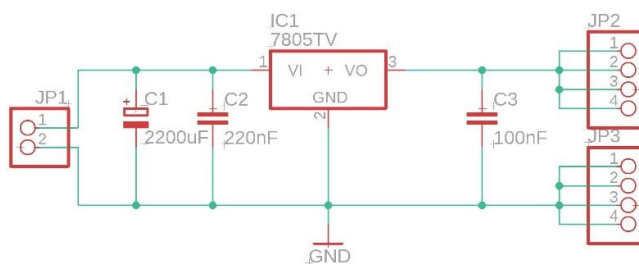
servomotory, piny pro možné připojení externích zařízení AUX 1 ÷ 4, piny pro komunikaci I2C a pak pin přímo pro napájení  $U_{CC}$ . Při testování bylo zjištěno, že tyto piny byly napájeny z jednoho zdroje, kde výrobce udával hodnotu napětí 5 V, ale po změření byla tato hodnota okolo 4,7 V. Po zapojení jedné karty tato hodnota klesla o 0,3 V a při zapojení další se tato hodnota ještě snížila. To bylo způsobeno tím, že integrovaný stabilizátor na modulu Arduino neposkytoval dostatečný proud. Pro správnou funkci AD převodníku (mimo jiné aby bylo možno využít jeho celý rozsah) bylo nutné mít na obou obvodech analogové karty stabilní napětí, na které byl převodník navržen. V případě nedodržení této podmínky by bylo výstupní napětí zatíženo chybou. K dispozici byl další způsob napájení a to z robota, který měl úroveň napájecího napětí 24 V. Toto napětí bylo možné s výhodou použít pro vstupní obvod analogové karty, jen bylo nutné ho adekvátně přizpůsobit napěťové úrovni vstupního obvodu analogové karty, tj. na požadovaných 10 V. Pro výstupní obvod analogové karty bylo možné využít napájecí zdroj 12 V, který byl již používán pro napájení Arduino modulu. Dalším způsobem napájení by bylo doplnit dva externí zdroje odpovídající napěťovým úrovním.

Zvolena byla varianta úpravy již používaných napětí, jak bylo popsáno ve druhém návrhu. Tímto způsobem bylo zajištěno požadované napájení zesilovačů pro využití jejich celých rozsahů. Nejjednodušší řešení by bylo opět použít děliče napětí a to jak pro napájení vstupního, tak pro napájení výstupního obvodu. Tato varianta měla ale stejný problém, jako to bylo popsáno u napájení z procesoru. Jednalo se o zdroj s velkým vnitřním odporem a v případě větší zátěže nebo navýšení počtu analogových převodníků nebylo možné je spolehlivě napájet. Další možnou variantou bylo použití stabilizátorů napětí a to pro vstupní obvod stabilizátor napětí 7805 a výstupní obvod stabilizátor napětí 7810. Oba stabilizátory byly vybrány ve variantě pro vyšší proudové zatížení až 2 A, aby byla zajištěna dostatečná rezerva pro případ přidání dalších analogových převodníků. Tento stabilizátor byl zapojen na základě doporučeného zapojení od výrobce (Obr. 1.14).

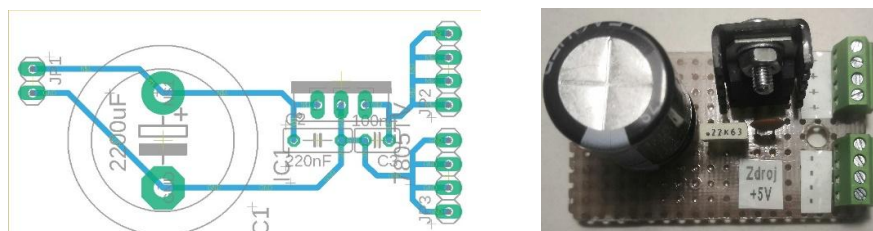


Obr. 1.14 Doporučené zapojení stabilizátoru 78XX [9]

Funkcí filtračních kondenzátorů  $C_1$  a  $C_3$  v obvodu bylo zajištění vyhlazení výstupního proudu při změně zátěže. Kapacita kondenzátoru před stabilizátorem by měla být 10x větší než kapacita za stabilizátorem. Pokud by nebyla splněna tato podmínka a kondenzátor na výstupu by měl větší kapacitu, mohl by tento kondenzátor přestat plnit funkci filtrace a místo toho by se výstupní proud mohl rozkmitat. Stabilizátor by na to zareagoval tím, že by nedokázal zvládnout tak vysoký proud a odpojil by výstup. Po opětovném připojení by se celý problém opakoval. Kondenzátor před stabilizátorem byl doporučen jako elektrolytický a dimenzovaný na minimální napětí 35 V, z důvodu možného dosažení 35 V na vstupu stabilizátoru. Funkcí kondenzátoru  $C_2$  bylo filtrovat šum a rušení z obvodu zapojení. Další podmínkou pro co nejlepší vlastnosti takového zapojení bylo, aby byly kondenzátory  $C_2$  a  $C_3$  umístěny co nejblíže stabilizátoru, nejlépe se společnou zemí. V tomto návrhu byly na oba stabilizátory namontovány chladiče z důvodů možných vyšších odběrných proudů.



Obr. 1.15 Schéma zapojení stabilizátoru 7805

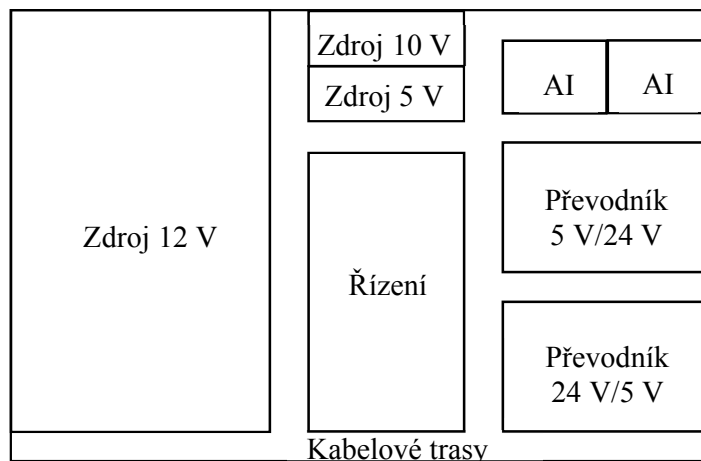


Obr. 1.16 DPS stabilizátoru 5 V (vlevo), výsledný stabilizátor (vpravo)

Pro stabilizátor napětí na 10 V bylo zapojení, schéma a DPS stejné jen byl změněn samotný stabilizátor na 7810.

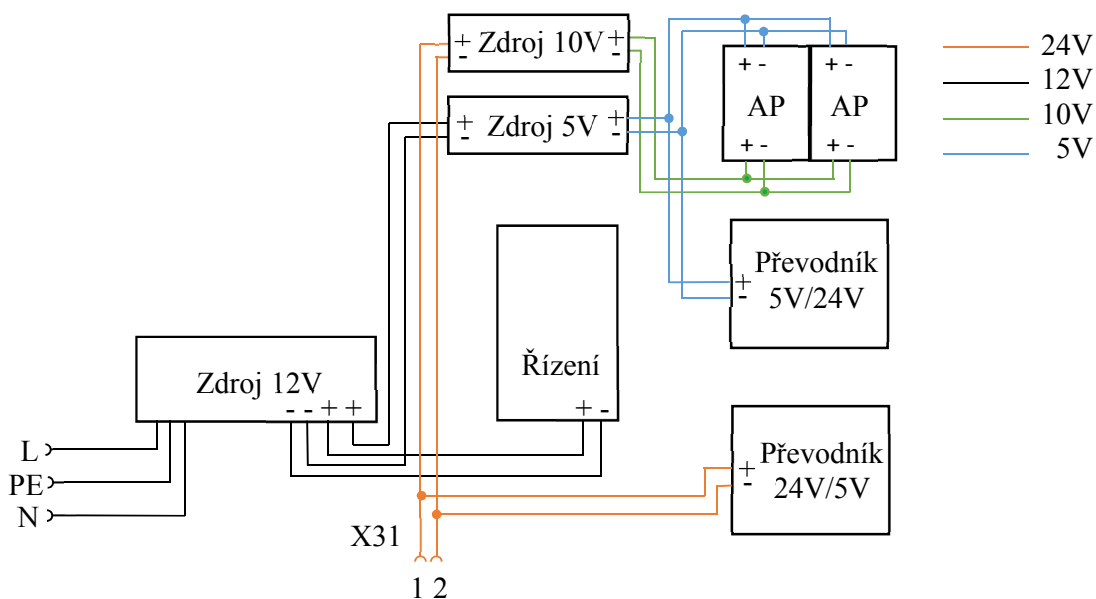
## 1.5 Návrh desky pro řídicí elektroniku

Samotná deska byla navržena v předešlé ročníkové práci. Následovala její výroba z hliníkového plechu o tloušťce 2 mm, rozvržení umístění jednotlivých komponent a kabelových tras, vyvrtání otvorů pro montáž a dále jejich osazení dle Obr. 1.17



Obr. 1.17 Rozvržení jednotlivých komponent na desce

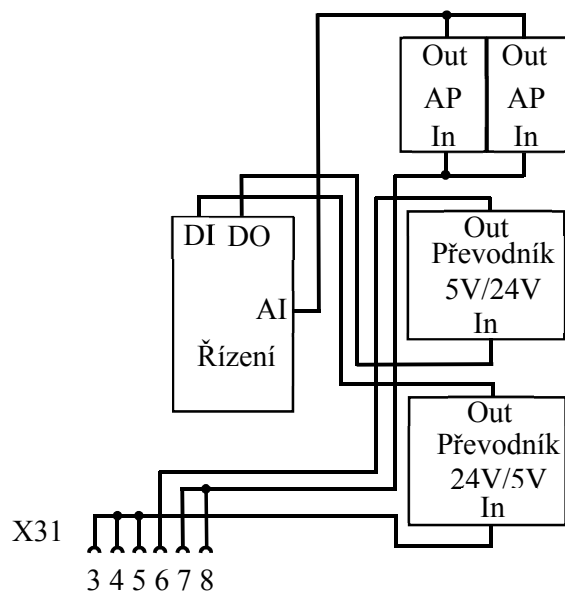
Jednotlivé komponenty byly napájeny různými úrovněmi napětí, jejich přesné zapojení je uvedeno na Obr. 1.18. Napájecí a komunikační konektory byly také součástí tohoto schématu a tyto konektory byly připojeny na rameno čtvrté osy robota.



Obr. 1.18 Schéma zapojení napěťových úrovní

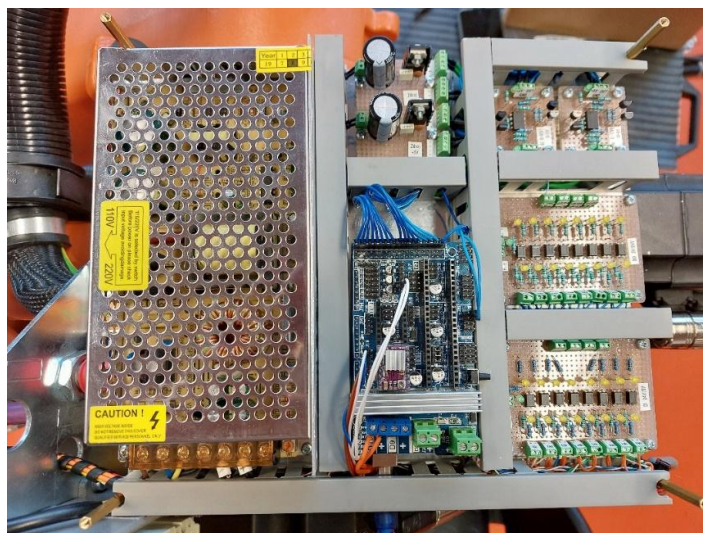
AP – analogový převodník

Na dalším schématu je rozkresleno zapojení pro výměnu signálů Obr. 1.19



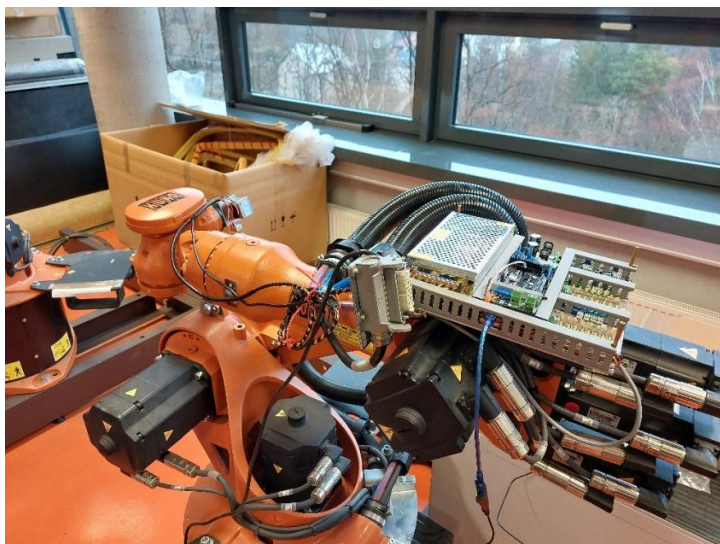
Obr. 1.19 Schéma zapojení pro výměnu signálů

Osazenou deska pro řídicí elektroniku je možné vidět obrázku Obr. 1.20. Přes celou desku bylo ještě vyřezáno průhledné plexisklo, které bylo připevněno na distančních sloupcích.



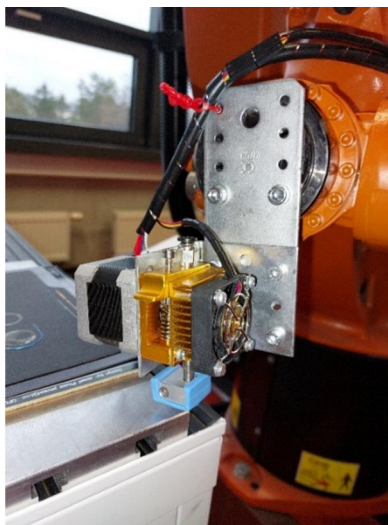
Obr. 1.20 Osazená deska pro řídicí elektroniku

Při montáži osazené desky na robota Obr. 1.21 bylo zjištěno, že stávající kabeláž mezi touto deskou a extruderem byla krátká, a proto bylo vyrobeno prodloužení o 0,7 m.



*Obr. 1.21 Montáž osazené desky na robota*

Extruder byl přichycen na navržený držák a upevněn na přírubu robota na šesté ose. Při učení nového TCP nástroje (trysky extruderu) se ukázalo, že v případě použití navrženého držáku extruderu zůstane mezi tělem ramene šesté osy a podložkou cca 1 cm prostoru. Držák bylo proto nutno prodloužit, aby nedošlo k případné kolizi kloubu robota s podložkou (viz Obr. 1.22).



*Obr. 1.22 Prodloužený držák extruderu*

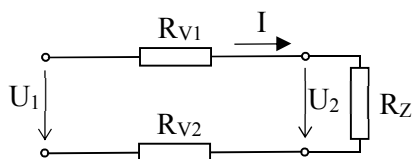
## 1.6 Cenová kalkulace

V této kapitole byly uvedeny přibližné náklady na stavbu této nástavby pro využití robota. V celkové částce nebyla zahrnuta práce autora, samotný robot ani SW vybavení. Vyhřívaná deska byla zakoupena pro účely případného rozšíření možnosti tisku z různých materiálů, ale pro základní projekt nebyla použita. Výhody jejího použití budou diskutovány dále.

hliníková deska	1100 Kč
elektronické součástky a montážní materiál	2472 Kč
extruder	699 Kč
sestava Arduino včetně driverů krokových motorů	1137 Kč
napájecí zdroj 12 V 20 A	799 Kč
vyhřívaná podložka	399 Kč
<hr/>	
celkem	6606 Kč

## 1.7 Vyhřívaná podložka

Pro další rozšíření použití této tiskárny pro tisk materiálů, která potřebující vyhřívání (např. ABS), by bylo vhodné doplnit vyhřívanou podložku, která byla také zakoupena společně s ostatními komponenty na 3D tiskárnu. Byla zvolena podložka od firmy Průša, pro tiskárny RepRap MK2b o velikosti 210 x 210 mm. Tuto desku by bylo možné napájet napětím o hodnotě 12 V nebo 24 V ze svorek D8, umístěných na nástavbě řízení. Podle katalogových hodnot [10] by byl při napájení 12 V proudový odběr podložky 10 A, to by odpovídalo výkonu 120 W. Z toho plyne, že by bylo nutné na tento odběr dimenzovat vodiče, protože deska by mohla být umístěna v dosahu robota, což by bylo 1,6 metru. Délka takto vedeného vodiče by mohla dosáhnout až 4 metrů a na tuto vzdálenost by bylo vhodné použití tohoto řízení pro ovládání další elektroniky pro spínání externího zdroje. Minimální průměr vodiče při započteném úbytku napětí na vedení 2 % (tj.  $U_V = 0,24 \text{ V}$ ) by bylo možno spočítat dle vztahu (1.13). Nejdříve byl vyjádřen odpor na vedení  $R_V$  a to na obou vodičích viz schéma na Obr. 1.23.



Obr. 1.23 Znázornění odporu na vodičích

$$U_V = R_v \times I = (R_{v1} + R_{v2}) \times I \quad (1.12)$$

$$R_v = \frac{U_v}{I} = \frac{0,24}{10} = 0,024 \Omega$$

Na základě vypočteného odporu vedení byl pomocí dalšího vzorce spočítán průřez vodiče. Tento vzorec byl upraven na dvojnásobnou délku vodiče (Obr. 1.23). Měrný odpor mědi při 20 °C je  $\rho = 0,0178 \Omega\text{m}/\text{mm}^2$  [11]

$$R = \rho \times \frac{2 \times l}{S} [\Omega; \Omega\text{m}/\text{mm}^2, m, \text{mm}^2] \quad (1.13)$$

$$S = \rho \times \frac{2 \times l}{R} = 0,0178 \times \frac{2 \times 4}{0,024} = 5,93 \text{ mm}^2$$

Z toho vyplývá, že průřez vodiče by měl být zvolen o průměru minimálně 6 mm<sup>2</sup>. Dále by bylo potřeba pro řízení teploty vyhřívání podložky, doplnit čidlo teploty, které by se umístilo na tuto desku, ale nebylo součástí nakoupených dílů. Použití takto velkého průměru vodiče by nebylo žádoucí vzhledem k jeho omezené ohebnosti. Byla by potřeba navrhnout jiné řešení, kde by byla podložka napájena z dalšího zdroje, který by byl spolu s odpovídajícím spínacím prvkem umístěn co nejbližší k podložce. Řídicí signál by pak bylo možné vést vnitřkem ramene robota a nebylo by potřeba využívat zbytečně vodiče s vypočteným průřezem.



## 2. Firmware (FW) pro řízení 3D tisku

V předešlém ročníkovém projektu byly popsány dva možné FW, první byl od firmy Marlin a druhý od firmy Repetier, dále byly popsány jejich výhody a nevýhody. V závěru této rešerše byl vybrán FW od firmy Repetier (firmware je šířen pod licencí GNU GPL V3, která umožňovala použití libovolných částí kódu a jejich libovolnou je modifikaci). Tento FW byl určený pro standartní 3D tiskárnu, ale na rozdíl od standartní 3D tiskárny byly pohyby v osách X, Y a Z nahrazeny robotem. Z daného FW bylo potřeba zachovat jen části, které se týkají podávání materiálu a ohřevu extruderu, případně ohřevu podložky. Naopak bylo potřeba doplnit komunikaci s robotem a synchronizaci rychlosti podávání materiálu vůči rychlosti robota. V online konfiguratoru byla zvolena minimální verze tohoto FW a byla provedena jen nejnnutnější nastavení v několika záložkách.

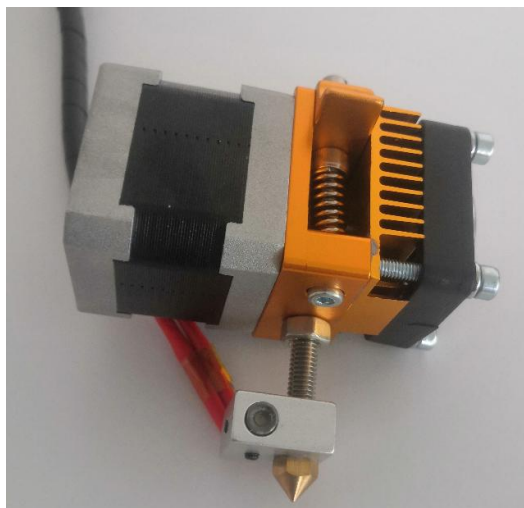
### 2.1 Vytvoření FW v konfiguratoru

Na záložce *Všeobecné* byl zvolen podle vybraného hardwaru (HW) procesor Atmel ATmega 2560 s nastavbou RAMPS 1.6, dále byl zvolen kartézský souřadný systém. I když na této tiskárně nedošlo k použití pohonů pro X, Y a Z, tak nebylo možné nastavit žádný souřadný systém nebo ho zcela vypnout. Základní komunikace byla zvolena po standartní sériové lince s rozšířením na USB port o rychlosti 115200 Baud. Došlo k vypnutí použití EEPROM, Bluetooth a jeho přenosové rychlosti a Z – korekce. Rozměry tiskárny ponechány v původním nastavení z výše uvedených důvodů.

Na záložce *Mechanika* bylo nastavení pro drivery krokových motorů a zde bylo nastaveno interní mikrokrokování na hodnotu 4, vypnuty pohony X, Y, Z a vypnuty jejich koncové spínače tzv. end stopy. Ostatní parametry zůstaly ve výchozím nastavení.

Na záložce *Nástroje* bylo ponecháno původní nastavení pro tiskový mód a to Fused Filament Fabrication (FFF). Možnost nastavení teplotních limitů a předehřátí trysky nebyla opět využita z důvodu vlastního nastavení teploty dle analogové hodnoty. V této záložce se nastavovalo použití vyhřívané podložky, ale opět této možnosti nebylo využito, podrobnosti viz kapitola 1.7 na straně 31. Tisková hlava byla také součástí této záložky a její nastavení bylo

ponecháno na výchozích hodnotách, až na nastavení rozlišení počtu kroků na mm, které bylo z původních 370 kroků/mm změněno na 740 kroků/mm. Teplotní čidlo umístěné na topné části tiskové hlavy bylo nastaveno dle zakoupené sestavy MK8 Obr. 2.1 a to 100 k $\Omega$  NTC 3950. Snímač vlákna filamentu nebyl použit, a proto bylo ponecháno výchozí nastavení.



*Obr. 2.1 Hot end MK8*

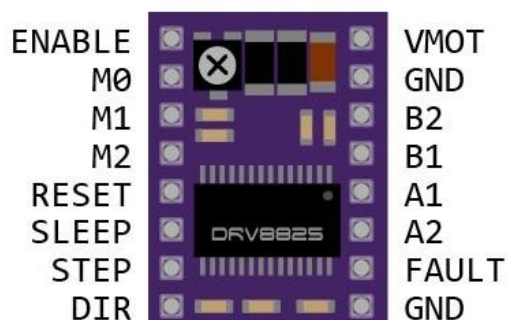
Na záložce *Vlastnosti* bylo ponecháno vše ve výchozím nastavení. Vygenerovaný FW, i v minimální konfiguraci, byl komplexní nástroj, který mohl ovládat laserovou řezačku, CNC obrábění a několik druhů tiskáren, mezi nimiž byla i delta tiskárna a různé její mutace, byl uložen a dále upraven dle požadavků.

## **2.2 Úprava FW**

Pro správné fungování 3D tisku pomocí robota bylo nejlepší zachovat, nebo se co nejvíce přiblížit, funkci klasické 3D tiskárny. Proto byl vygenerovaný FW upraven tak, aby byl tento předpoklad splněn.

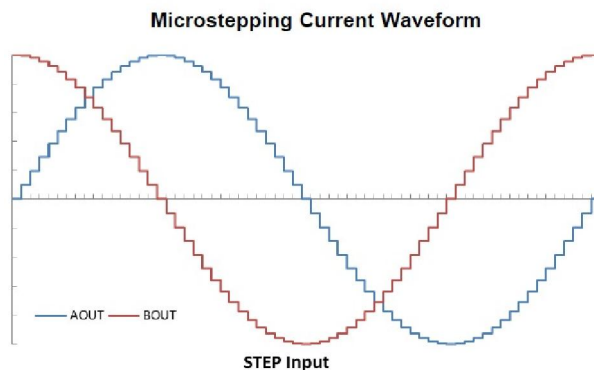
Na úpravy FW bylo použito vývojové prostředí *Arduino IDE*, které bylo k dispozici zdarma. Nejprve byly odstraněny všechny nepotřebné volby a jejich nastavení, dále byly odstraněny všechny funkce pro jiné typy tiskáren a následně byly odstraněny bloky pro jazykové mutace a použití menu přes displej. Byla také odstraněna podpora klávesnice, nastavení Z – sondy a polohování podložky. I když bylo v konfigurátoru nastaveno, že nejsou použity pohony, přesto se vygeneroval i kód pro jejich ovládání, který byl také odstraněn.

Na testování funkcí tiskárny byl použit SW *Pronterface*, ve kterém bylo možné testovat jednotlivé příkazy buď pomocí *G* kódu nebo *M* kódu, nebo pomocí tlačítek pro pohyb motorů a ohřev extruderu. Před samotným testováním pohybu motoru extruderu bylo nutné nastavit driver krokového motoru DRV8825. Tento driver neboli ovladač motoru, sloužil jako výkonový prvek pro spínání cívek krokového motoru a z procesoru byl ovládán signály krok (STEP) a směr otáčení (DIR).



*Obr. 2.2 DRV8825 pinout*

Použitý ovladač byl vhodný pro pohon malých krokových motorů, jako byl v této práci použitý bipolární krokový motor NEMA 17 v extruderu MK8, protože jeho maximální výstupní napětí bylo 45 V a výstupní proud  $\pm 2$  A. Krokový motor NEMA 17 měl velikost kroku dle katalogu výrobce [12]  $1,8^\circ$ , neboli 200 kroků za jednu otáčku. Ovladačem DRV8825 bylo možné toto nastavení změnit tak, že se pomocí pinů M0 – M2, které jsou vyvedeny jako propojky umístěné na nástavbě RAMPS 1.6 pod tímto ovladačem, nastavily větší rozlišení (ovladač mezi základní kroky vloží mezikroky buzením cívek středními úrovněmi proudu Obr. 2.3, takzvané mikrokrokování) podle použitých propojek viz Obr. 2.4.



*Obr. 2.3 Mikrokrokování proudového průběhu [13]*

MODE2	MODE1	MODE0	STEP MODE
0	0	0	Full step (2-phase excitation) with 71% current
0	0	1	1/2 step (1-2 phase excitation)
0	1	0	1/4 step (W1-2 phase excitation)
0	1	1	8 microsteps/step
1	0	0	16 microsteps/step
1	0	1	32 microsteps/step
1	1	0	32 microsteps/step
1	1	1	32 microsteps/step

Obr. 2.4 Tabulka nastavení mikrokrokování podle zapojených pinů M0 – M2 [13]

Podle této tabulky byly zapojeny všechny tři propojky, aby došlo k využití maximální hodnoty mikrokrokování, což byla hodnota 32. Dále bylo potřeba nastavit proudový limit odpovídající zapojenému motoru, v tomto případě právě motor NEMA 17, který měl podle katalogu výrobce [12] jmenovitý proud 1 A. V případě, že by nebyl tento proud nastaven, mohlo by dojít k poškození buď krokového motoru, nebo ovladače DRV8825. Pro správné nastavení proudového limitu, bylo potřeba změřit referenční napětí a podle toho nastavit odporový trimr umístění na tomto ovladači. Tento úkon byl prováděn bez zapojeného krokového motoru, aby nedošlo k ovlivnění tohoto napětí. Výpočet proudového limitu byl proveden podle následujícího vzorce:

$$I_{limit} = U_{ref} \times 2 \quad (2.1)$$

$$U_{ref} = \frac{I_{limit}}{2} = \frac{1}{2} = 0,5 V$$

Referenční napětí bylo přizpůsobeno dle vypočtené hodnoty přenastavením odporového trimru. Po nastavení těchto parametrů bylo započato testování dvou požadovaných parametrů, a to rychlosti podávání materiálu a ohřevu extruderu. Jak bylo popsáno v ročníkovém projektu, výchozí řízení bylo použito z klasické 3D tiskárny, která byla řízena *G* kódy pro programování dráhy nástroje v kartézském souřadném systému a *M* kódy pro ovládání funkcí.

### 2.3 Nastavení rychlosti

Pro základní pohyby řízení 3D tiskárny byly použity příkazy *G0* a *G1*. Příkaz *G0* byl obecný pohyb a jako takový se používal pro rychlý přesun nástroje např. *G0 X300 Y300 Z200*. Kladné hodnoty těchto čísel (souřadnic) by měly pohybovat motory doprava a záporné doleva. V této práci nebylo použito řízení os, a proto nebyl tento příkaz dále řešen. Příkaz *G1* byl lineární pohyb, se kterým se ovládala i osa *E*, což byl v tomto případě právě extruder. U tohoto

příkazu byl použit i parametr  $F$  pro rychlost posuvu os, ale ne pro rychlost podávání materiálu. Rychlost podávání materiálu se řídila jen pomocí číselné hodnoty udané vedle písmene  $E$ , tato hodnota musela být přepočtena na rychlost a následně na čas potřebný k nastavení krokového motoru. K tomu, aby byla zjištěna rychlost jedné otáčky, bylo nutné nejdříve znát dráhu této otáčky, která představovala obvod podávacího kolečka umístěného na hřídeli krokového motoru extruderu. Poloměr tohoto kolečka byl změřen  $r = 5,5$  mm a obvod následně vypočítán podle vzorce pro obvod kruhu.

$$o = 2 \times \pi \times r \quad (2.2)$$

$$o = 2 \times \pi \times 5,5 = 34,56 \text{ mm}$$

V testovacím software *Pronterface* byla tedy zadána tato hodnota k souřadnici  $E$ , testovací rychlost podávání byla nastavena na 100 mm/min (toto nastavení umožňoval zmíněný software), motor byl z extruderu vyjmut a jeho hřídel byla označena praporkem, aby byla zjištěna správnost zapojení motoru a tím i směr otáčení a zjištěn počet otáček. Byla provedena zkouška otočením motoru, jejímž výsledkem bylo, že se motor otočil správně, ale došlo k otočení čtyřikrát. Tato cesta se tedy ukázala jako slepá a výpočet vzdálenosti na jednu otáčku byl zpracován z informace o počtu kroků na otáčku a počtu kroků na 1 mm (740 kroků) jako to bylo uvedeno v kapitole 2.1 na straně 34. Výpočet byl proveden trojčlenkou.

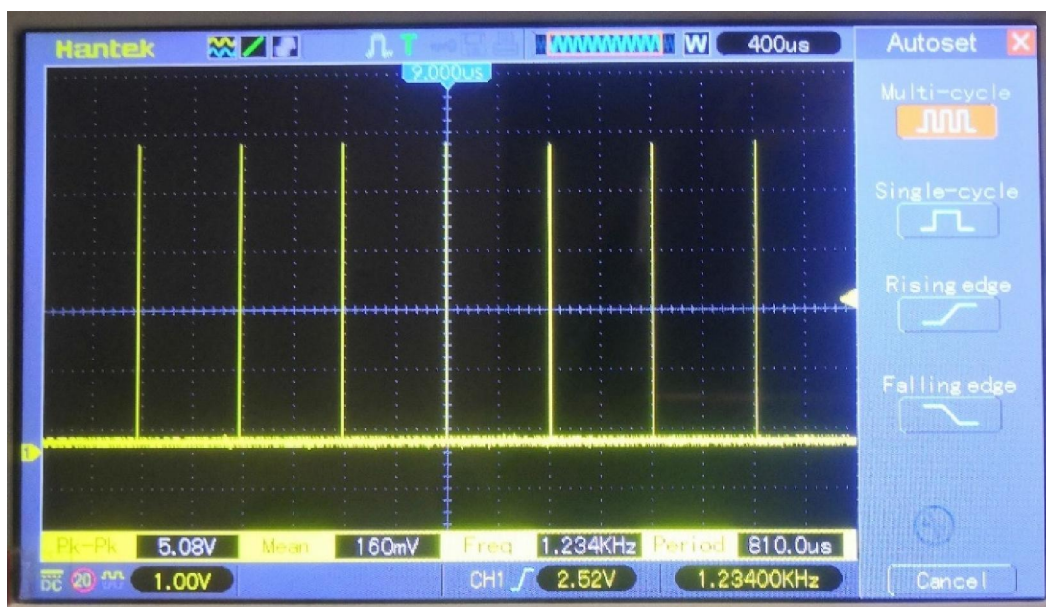
$$x \text{ mm} \dots \dots \dots 6400 \text{ kroků}$$

$$1 \text{ mm} \dots \dots \dots 740 \text{ kroků}$$

$$x = \frac{6400}{740} = 8,64 \text{ mm}$$

Provedení druhé zkoušky dopadlo dle předpokladů. Pro ověření správné krokovací frekvence byl změřen průběh signálů *STEP* osciloskopu.

Průběh měření je uveden na následujícím obrázku Obr. 2.5.



Obr. 2.5 Průběh frekvence z původního řízení

Dalším krokem byla úprava části FW, určená pro řízení krokového motoru s periodou  $T = 810 \mu\text{s}$ . Tento pokus nedopadl pozitivně, naměřená perioda  $T$  byla cca o +25% vůči nastavené hodnotě. Po dalších úpravách kódu FW byly zjištěny podobné odchylky, ale střední hodnota periody byla okolo požadovaných  $810 \mu\text{s}$ . Tento výsledek byl shledán jako neuspokojivý. Na záložce *Motion* ve FW Repetier byly nalezeny vzorce pro přepočítání velikosti vzdálenosti na frekvenci krokového motoru.

$$timeForMove = \frac{F\_CPU \times distance}{Printer::feedrate} \quad (2.3)$$

$$limitInterval = \frac{timeForMove}{stepRemaining} \quad (2.4)$$

$$f = \frac{F\_CPU}{limitInterval} \quad (2.5)$$

$F\_CPU$  představovala frekvenci použitého CPU s hodnotou 16 MHz, *distance* byla velikost vzdálenosti pro jednu otáčku (8,64 mm), *Printer::feedrate* byla přednastavená rychlost a její hodnota byla  $100 \text{ mm/min} = 1,67 \text{ mm/s}$ , *stepRemaining* byl původní počet kroků motoru

(200) vynásobený hodnotou mikrokrokování (32). Po dosazení do rovnic (2.3), (2.4) a (2.5) byly získány následující hodnoty:

$$timeForMove = \frac{F\_CPU \times distance}{Printer::feedrate} = \frac{16 \times 10^6 \times 8,64}{1,67} = 82938046$$

$$limitInterval = \frac{timeForMove}{stepRemaining} = \frac{82938046}{200 \times 32} = 12959$$

$$f = \frac{F\_CPU}{limitInterval} = \frac{16 \times 10^6}{12959} = 1234,66 \text{ Hz}$$

Velikost frekvence byla 1234 Hz, což odpovídalo naměřené frekvenci na osciloskopu a podle vzorce:

$$T = \frac{1}{f} \quad [s; -, Hz] \quad (2.6)$$

byla hledaná perioda:

$$T = \frac{1}{f} = \frac{1}{1234} = 8,10 \times 10^{-4} = 810 \mu s$$

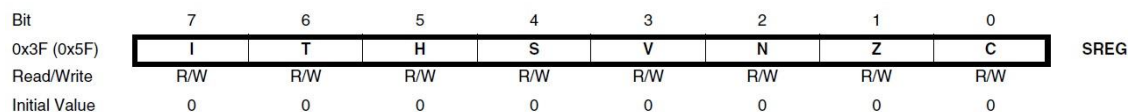
Tato hodnota periody byla úměrná nastavené rychlosti  $v = 100 \text{ mm/min}$ . Při dalším pokusu o dosazení vypočtené periody do vlastního kódu se toho příliš nezměnilo, protože velikost chyby byla dána cyklickým zpracováním programu, kde v každém dalším cyklu byla hodnota periody zatížena chybou odpovídající času na zpracování daného kódu. Se zvyšující se délkou kódu by se tato chyba dále zvětšovala, z toho bylo vyvozeno, že není použito softwarového nastavení časovače pro řízení krokového motoru, ale bylo použito hardwarového nastavení pomocí přerušení. Z toho důvodu byl opět prohledán původní FW, kde se nastavení přerušení nachází na záložce *HAL* (Hardware Abstraction Layer) pro funkci *setTimer*, která obsahovala výsledek ze vzorce (2.4), který odpovídá hodnotě 12959 a *Timer1*.

V dokumentaci procesoru Atmel bylo nalezeno, že pod označením *Timer1* je myšlen *Timer/Counter1 Compare Match A*, který byl označen jako *vektor 17* [1] a nastavoval se přes registr *TIMSK1* (Timer/Counter 1 Interrupt Mask Register) Obr. 2.6.

Bit	7	6	5	4	3	2	1	0	
(0x6F)	-	-	ICIE1	-	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obr. 2.6 Registr *TIMSK1* (Timer/Counter 1 Interrupt Mask Register) [1]

V tomto registru se pomocí první bitu *OCIE1A* (Timer/Counter1, Output Compare a Match Interrupt Enable) povolovalo nebo zakazovalo přerušení. Zároveň ale musel být ve stavovém registru *SREG* (Status register) Obr. 2.7 nastaven příznak *I* (globální povolení přerušení).



Obr. 2.7 Stavový registr *SREG* [1]

Následně bylo toto ovládání přerušení implementováno do vlastního kódu pro ovládání krokového motoru. Byla provedena úspěšná zkouška pomocí řídicích bitů vpřed/vzad (forward/backward) z robota na pinech D16/D17, ve kterém byl naměřen stejný průběh signálu jako na Obr. 2.5 Nastavení časovače pro požadované krokování bylo prováděno s pevnou hodnotou rychlosti 100 mm/min (z důvodů zkoušek), pro proměnou hodnotu řízení rychlosti z robota to bude uvedeno v kapitole 2.5 na straně 41.

## 2.4 Nastavení teploty extruderu

Pro ovládání teploty extruderu byl používán ve standardní 3D tiskárně příkaz *M104*, spolu s dalším parametrem *S*, kde byla udávaná teplota. Hodnota parametru *S0* znamená vypnout vyhřívání extruderu. Zkouška funkce pro vyhřívání extruderu v programu *Pronterface* pomocí příslušného příkazu dopadla úspěšně.

Po nalezení kódu pro ohřev bylo dále zjištěno, že se tento příkaz odkazuje na záložku *Extruder* ve FW Repetier, přesněji na proceduru (*setTemperatureForExtruder*), která přímo ovládala vyhřívání extruderu. Tato funkce disponovala čtyřmi parametry. První parametr představoval teplotu ve stupních Celsia, druhým parametrem bylo číslo extruderu (v FW bylo možné ovládat až šesti extruderů), třetí parametr ovládal pípnutí a čtvrtým bylo ovládání časové prodlevy. Poslední dva parametry byly typu *BOOL* a nebyly pro tuto práci použity, ale v původní funkci zůstaly zachovány a byly nastaveny na hodnotu “FALSE”. Ve vlastním programu byla tato funkce volána s parametry pro teplotu 100 °C a extruder “0” na základě řídicího signálu (*startHeatingExtruder*) z robota na pinu D23. Bylo úspěšně ověřeno nahřátí extruderu na 100 °C pomocí programu *Pronterface*, ve kterém bylo možné sledovat průběh



ohřevu. Do kódu byla ještě doplněna informace o zpětném vyčtení teploty zahřátého extruderu. Po nahřátí na požadovanou teplotu byl nastaven výstupní signál (*temperatureOk*) do robota (D35). Řízení teploty podle analogového signálu z robota bylo podrobně zpracováno v následující kapitole. Tento příkaz (*current- >tempControl.currentTemperatureC*) byl také nalezen v záložce *Extruder*.

Pro vypínání ohřevu extruderu byly nastaveny dvě podmínky. První podmínkou bylo vypnutí řídicího signálu (*startHeatingExtruder*) a druhá představovala nastavení teploty nižší než 100 °C, z důvodu lepšího rozsahu pro nastavení teplot a také z důvodu, že žádný z popisovaných materiálů (materiály byly popsány v ročníkovém projektu) nevyužíval teplotu nižší než je 100 °C.

## 2.5 Čtení a zpracování analogových hodnot

V procesoru Atmel ATmega 2560 byl výrobcem použit postupný deseti-bitový převodník ADC, který převádí analogové vstupní napětí na digitální hodnotu. ADC je připojený k šestnácti-kanálovému analogovému multiplexoru, který dokáže obsloužit 16 analogových vstupů. Každý analogový vstup je vztažen k nulovému napěťovému potenciálu, který představuje minimální hodnotu a maximální hodnota je zastoupena napětím na referenčním pinu  $A_{REF}$  ponížena o nejnižší váhový bit např. pro hodnotu 1024 je to 1023. Referenční napětí bylo možné použít interní s možností volit jeho hodnotu jako 1,1 V nebo 2,56 V, případně bylo možné použít externí napěťovou referenci, připojenou k příslušnému pinu procesoru. V tomto případě zaleží na uživateli, jaké napětí přivede na pin  $AV_{CC}$ , ale bylo zapotřebí dodržet podmínku, že se napětí na pinu  $AV_{CC}$  nesmí lišit o více než  $\pm 0,3$  V od  $V_{CC}$ . Použitá napěťová reference pro AD převodník se v procesoru volí nastavením registru *ADMUX* (ADC Multiplexer Selection Register) Obr. 2.8 na bitech *REFS1* a *REFS0* (Reference Selection Bits) podle Obr. 2.9.

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obr. 2.8 Registr výběru multiplexoru pro ADC [1]

REFS1	REFS0	Voltage Reference Selection <sup>(1)</sup>
0	0	AREF, Internal $V_{REF}$ turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Internal 1.1V Voltage Reference with external capacitor at AREF pin
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Obr. 2.9 Volby referenčního napětí pro ADC [1]

K výběru analogového vstupního kanálu docházelo zápisem do bitů  $MUX0 - MUX5$  v registru  $ADMUX$  Obr. 2.8 a v registru  $ADCSRB$  (ADC Control and Status Register B) Obr. 2.10.

Bit	7	6	5	4	3	2	1	0	
(0x7B)	-	ACME	-	-	MUX5	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obr. 2.10 ADC kontrolní a stavový registr B [1]

Převodník ADC byl povolen nastavením bitu  $ADEN$  (ADC Enable) a odstartoval se nastavením bitu  $ADCS$  (ADC Start Conversion), dále se nastavil dělicí faktor pomocí bitů  $ADPS0 - ADPS2$ , byl vypočítán jako poměr frekvence použitého procesoru (v tomto případě 16MHz) a vstupními hodinami AD převodníku podle Obr. 2.11 v registru  $ADCSRA$  (ADC Control and Status Register A). Dělicí faktor byl nutný pro dosažení potřebné přesnosti převodníku, např. pro deseti bitovou přesnost musely být nastaveny hodiny převodníku na frekvenci mezi 50 až 200 kHz. Z toho vyplývalo, že byl dělicí faktor nastaven na hodnotu 128, aby byla splněna požadovaná přesnost ( $16 \text{ MHz}/128 = 125 \text{ kHz}$ ). Výsledek byl ukládán do registrů  $ADCH$  a  $ADCL$ .

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obr. 2.11 ADC kontrolní a stavový registr A [1]

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

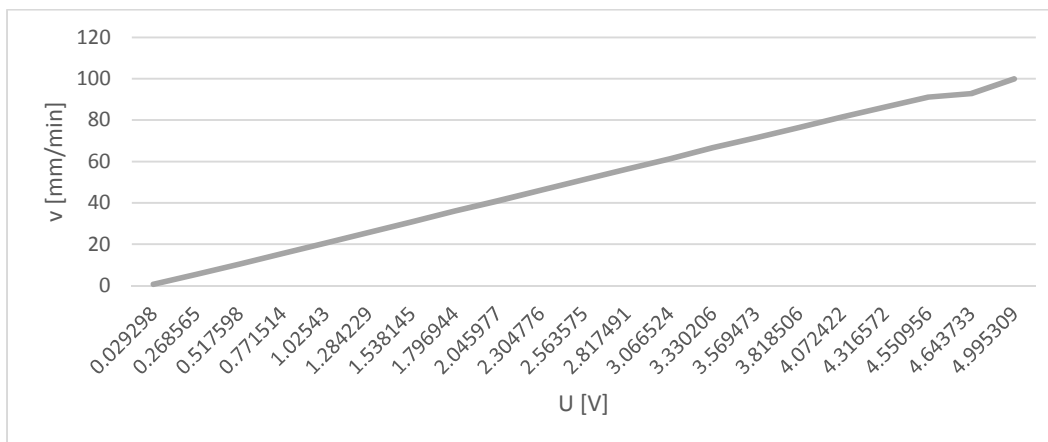
Obr. 2.12 Tabulka dělicího faktoru [1]

V programu *Arduino IDE* nebylo nutné přistupovat k těmto registrům přímo, protože tento program používal funkce *analogRead()* - do závorky se uvádělo číslo analogového pinu. Došlo k provedení zkoušky čtení jednoho analogového vstupu pro rychlost. Výsledek zkoušky nebyl příliš úspěšný, jelikož se objevily náhodné změny měřené teploty extruderu, což byla také analogová hodnota. Analýza této chyby ukázala, že ve FW už se jedna analogová hodnota vyčítala a při použití druhého vyčítání se tyto hodnoty přenášely do původního čtení. Opět byl prohledán FW, kde bylo původní vyčítání použito. Na záložce *HAL* byla tato funkce nalezena a rozšířena o další analogové vstupy z důvodu, že se jednalo o cyklickou funkci, která v každém cyklu přestavovala vstupní kanál na bitech *MUX0 - MUX5*. Následná zkouška vyčítání všech analogových vstupů dopadla úspěšně.

Pro správné nastavení obou převodníkových karet bylo potřeba zjistit, jak se chovají při jednotlivých vstupních napěťových úrovních, a proto bylo provedeno měření se zápisem do tabulek (rychlost AI1 Obr. 2.13 a teplota AI2 Obr. 2.15) a byly vyhodnoceny podle naměřených hodnot grafy Obr. 2.14 a Obr. 2.16. Na základě takto získaných měření byly ve FW upraveny hodnoty podle vzorce  $y = k \cdot x + q$ . Pro řízení teploty byla provedena druhá dodatečná korekce, která byla popsána v kapitole 2.4 na straně 41, následně byly tyto výpočty zahrnuty do funkcí pro ovládání rychlosti podávání materiálu a ohřevu teploty extruderu.

vstup [V]	AI1-OUT [V]	Hodnota ADC1	Korekce $y = x - 72$	AI1-OUT' [V]	v [mm/min]
0	0.4	78	6	0.03	0.59
0.5	0.63	127	55	0.27	5.38
1	0.88	178	106	0.52	10.36
1.5	1.13	230	158	0.78	15.43
2	1.38	282	210	1.03	20.51
2.5	1.63	335	263	1.29	25.69
3	1.88	387	315	1.54	30.77
3.5	2.13	440	368	1.8	35.94
4	2.4	491	419	2.05	40.92
4.5	2.63	544	472	2.31	46.1
5	2.89	597	525	2.57	51.27
5.5	3.14	649	577	2.82	56.35
6	3.39	700	628	3.07	61.33
6.5	3.64	754	682	3.34	66.61
7	3.88	803	731	3.57	71.39
7.5	4.13	854	782	3.82	76.37
8	4.37	906	834	4.08	81.45
8.5	4.61	956	884	4.32	86.33
9	4.85	1004	932	4.56	91.02
9.5	5.03	1023	951	4.65	92.88
10	5.05	1023	1023	5	99.91

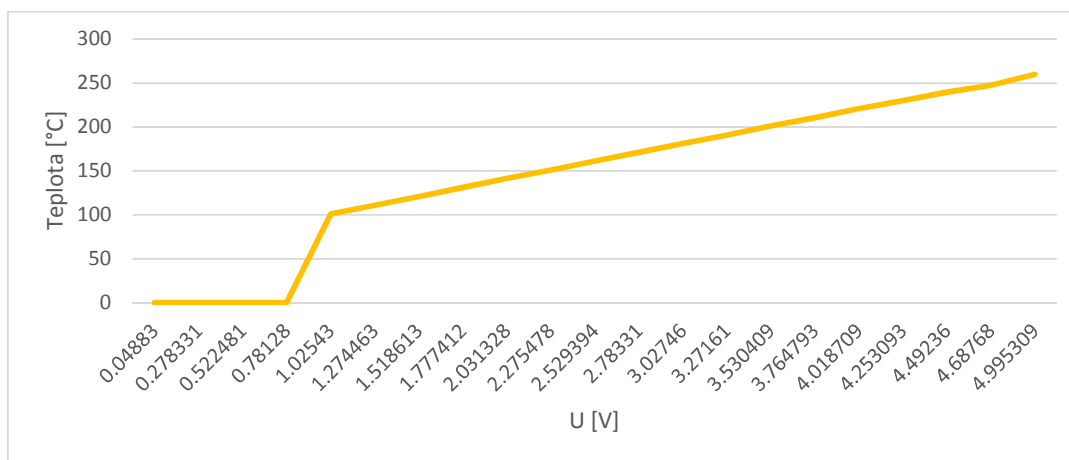
Obr. 2.13 Tabulka hodnot analogového vstupu 1 - rychlost



Obr. 2.14 Graf rychlosti v závislosti na výstupním napětí analogového převodníku AI1

vstup [V]	AI2-OUT [V]	Hodnota ADC2	Korekce y = x - 62	AI2-OUT [V]	Teplota [°C]	Korekce' y = 0.19536x + 59.9512
0	0.36	72	10	0.05	2.54	0
0.5	0.6	119	57	0.28	14.48	0
1	0.84	169	107	0.53	27.17	0
1.5	1.08	222	160	0.79	40.63	0
2	1.33	272	210	1.03	53.33	100.98
2.5	1.58	323	261	1.28	66.27	110.95
3	1.82	373	311	1.52	78.97	120.71
3.5	2.07	426	364	1.78	92.43	131.07
4	2.32	478	416	2.04	105.63	141.23
4.5	2.56	528	466	2.28	118.33	150.99
5	2.81	580	518	2.53	131.53	161.15
5.5	3.06	632	570	2.79	144.73	171.31
6	3.3	682	620	3.03	157.43	181.08
6.5	3.54	732	670	3.28	170.12	190.85
7	3.79	785	723	3.54	183.58	201.2
7.5	4.03	833	771	3.77	195.77	210.58
8	4.27	885	823	4.02	208.97	220.74
8.5	4.5	933	871	4.26	221.16	230.11
9	4.74	982	920	4.5	233.6	239.69
9.5	4.95	1022	960	4.69	243.75	247.5
10	5.05	1023	1023	5	259.75	259.81

Obr. 2.15 Tabulka hodnot analogového vstupu 2 – teplota



Obr. 2.16 Graf rychlosti v teploty na výstupním napětí analogového převodníku AI2

### 3. Software (SW) pro robota

Tato kapitola se zabývá vytvořením pohybového programu pro robota s možností synchronního ovládní extruderu. Jak již bylo zmíněno v úvodu, v této práci byl použit robot KR16-2 s nosností 16 kg, od firmy Kuka, s řídicím systémem KRC4. Tento systém nebyl schopen zpracovat *G* kódy a *M* kódy, které byly vygenerovány ze Sliceru a pomocí nichž se prováděl samotný 3D tisk. Bylo tedy nutné přeložit soubor ze sliceru, který obsahoval *G* kódy do jazyka KRL (Kuka Robot Language). Samotnými *G* kódy byly pohyby robota a *M* kódy sloužily v tomto případě k ovládní funkcí, jako jsou zapínání/vypínání a směr otáčení motoru extruderu, rychlost podávání materiálu závislá na rychlosti robota, zapínání/vypínání ohřevu extruderu a nastavení teploty extruderu podle použitého materiálu. Široká škála možností byla popsána v ročníkové práci. Jednou z nich bylo použití modulu od firmy KUKA (*KUKA CNC*) nebo od jiného výrobce (*Autodesk PowerMill*). Tato možnost je ale placená. Další možností bylo vytvoření části, nebo úplně celého modulu vlastními silami, nebo vyhledat nějaký podobný modul a ten následně upravit na danou aplikaci. Byla vybrána poslední jmenovaná možnost a následně byly zvoleny tyto programy *Gcode* [14], *Zuka* [15], *Translator G-code do jazyka robotů KAWASAKI* [16] a *RoboDK* [17].

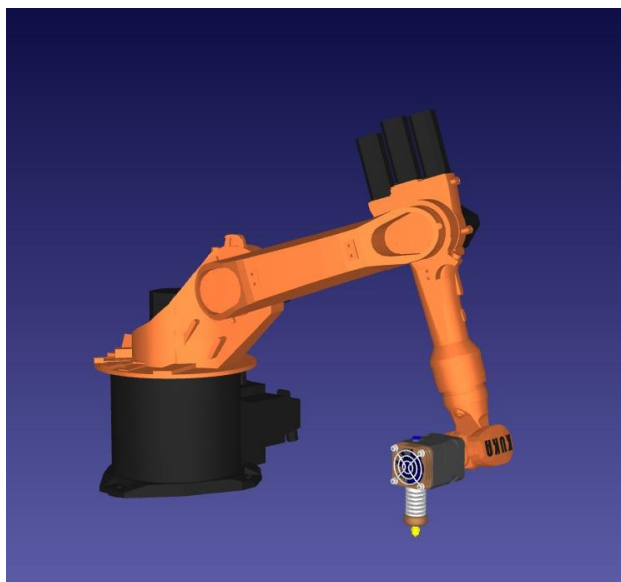
*Gcode* představuje jen dva skripty napsané v jazyce *Python*, jeden vytváří soubor s příponou DAT a druhý s příponou SRC. Testováním zjištěno, že přeložený soubor měl velikost jen 1 kB, když původní soubor byl tisícinásobně větší. Po prostudování daných skriptů byl vyvozen závěr, že se jednalo pouze o jakousi pracovní verzi, která není dopracována do funkčního programu.

*Zuka* – tato aplikace byla navržena na robota firmy KUKA KR6 R900 pro 3D frézování a 2D kreslení. Obsahovala i překladač *G* kódu do KRL. Pro použití k tomuto projektu by musela být aplikace upravena na jiný typ robota a bylo by nutné přepracování 3D frézování na 3D tisk.

*Translator G-code do jazyka robotů KAWASAKI* – jednalo se o překladač *G* kódů do jazyka AS, který byl podporován roboty Kawasaki, a byl otestován na robotu Kawasaki RS003N. Program napsán v jazyce *Python* a jeho použití bylo vytvořeno pro gravírování. Tento program není dostupný na internetu, musel by být zcela vytvořen podle této práce a dále upraven na systém KUKA KRL a pro 3D tisk.

*RoboDK* – tento program je dostupný na internetu v plné verzi, s omezením na třiceti denní zkušební období. Po uplynutí tohoto zkušební období dojde k přepnutí do omezené verze, přičemž všechny funkce potřebné pro tuto práci zůstanou přístupné. Program byl vyvinut jak na různé simulace s roboty, tak na celá pracoviště, a uživatel může sledovat reálný průběh jím navrženého programu, aniž by musel vlastnit fyzické komponenty. Obsahuje knihovny aktuálně nabízených robotů všech výrobců, části linek, uchopovacích systémů a pro tuto práci potřebný extruder. Součástí programu byly i překladače tzv. postprocesory napsané v jazyce *Python*, které sloužily mimo jiné i pro vytvoření programu robota z *G* kódu.

Z popsaných programů byl zvolen *RoboDK*, právě pro jeho kompletní knihovny robotů a jeho možnost překladu *G* kódu do jazyka KRL. Pro správnou funkčnost tohoto programu na 3D tisk bylo potřeba ještě nahrát do adresáře *Other slicer* (*Slic3r*), dle doporučení výrobce v návodu [18]. Ve virtuálním prostředí byl vybrán robot KUKA KR16-2 se systémem KRC4, určen souřadný systém nástroje a doplněn vlastní nástroj, a to extruder.



Obr. 3.1 Vizualizace zvolené sestavy v programu *RoboDK*

Byl vybrán soubor s návrhem pro zkušební tisk a pomocí sliceru nastaveny parametry pro 3D tisk, následně byl vygenerován program (dva soubory, jeden s příponou *DAT* a druhý s příponou *SRC*) pro robota pomocí postprocesoru *KUKA\_KRC4\_DAT*. Ve výsledném programu byly správně přeloženy všechny *G* kódy pro osy X, Y, Z a pro extruder došlo k vygenerování volání programu Extruder a *M* kódy nebylo možné přeložit. Program *RoboDK* umožňuje editovat nabízené postprocesory, a proto byl daný postprocesor upraven pro program Extruder a použité *M* kódy v tomto projektu, což byly:

- *M82* – nastavuje relativní souřadnice pro extruder – tento kód byl sice přeložen, ale nevykonává žádnou operaci.
- *M84* – vypnutí všech motorů – nastaví digitální výstupy DO1 (*forward*) a DO2 (*backward*) do logické 0.
- *M106* – zapnutí ventilátoru na chlazení extruderu – tato funkce nebyla v této práci použita, ale v postprocesoru byl již připraven digitální výstup DO4 do logické 1.
- *M107* – vypnutí ventilátoru na chlazení extruderu – tato funkce nebyla v této práci použita, ale v postprocesoru byl již připraven digitální výstup DO4 do logické 0.
- *M109* – čeká na nahřátí extruderu na požadovanou teplotu - čeká na digitální vstup DI1.
- *M140 Sxxx* – nastaví teplotu podložky – tato funkce nebyla v této práci použita, viz kapitola 2.7 na straně 31.
- *M104 Sxxx* – nastaví teplotu extruderu - zapne vyhřívání extruderu na digitálním výstupu DO3 (*startHeatingExtruder*) a nastaví teplotu extruderu podle parametru *S*, která byla převedena na výstupní napětí podle vzorce

$$u = \frac{U_{MAX}}{\tau_{MAX}} \times PRINT\_E\_temperatureExtruder \quad (3.1)$$

kde  $U_{MAX} = 10 \text{ V}$ ,  $\tau_{MAX} = 260 \text{ }^\circ\text{C}$ , *PRINT\_E\_temperatureExtruder* byla hodnota parametru *S*. Výsledná hodnota byla zapsána na analogový výstup AO2.

Další *M* kódy nebyly v této práci zahrnuty a překladač na ně reaguje ve výsledném programu větou: “*Tento kód neznám*“. Následovala další zkouška přeložení programu s úspěšným výsledkem.

Pro úpravu programu *Extruder* bylo možné využít funkce doporučené od výrobce podle návodu [14], která byla převzata jen částečně a to pro výpočet času (*calculate\_time*)

$$T_{tot} = \frac{\text{distance}}{V_{MAX}} \quad [\text{s}; \text{mm}, \text{mms}^{-1}] \quad (3.2)$$

kde byl  $T_{tot}$  celkový čas, *distance* hodnota velikosti souřadnice,  $V_{MAX}$  byla maximální rychlost robota v dané souřadnici. Vzorec (3.2) bylo možné použít za předpokladu, že nebylo

použito zrychlení (hodnota  $A_{MAX} = 1$ ). Další část kódu nebyla využita. Samotný program *Extruder* byl vytvořen na základě kladné, záporné nebo nulové hodnoty této souřadnice a podle toho bylo nastaveno zapnutí vpřed, vzad nebo vypnutí obou směrů v případě nulové souřadnice. Rychlost motoru pro podávání materiálu byla přizpůsobena rychlosti robota podle vzorce

$$v = speed\_mms \times PRINT\_FLOW\_MAX\_SIGNAL \quad (3.3)$$

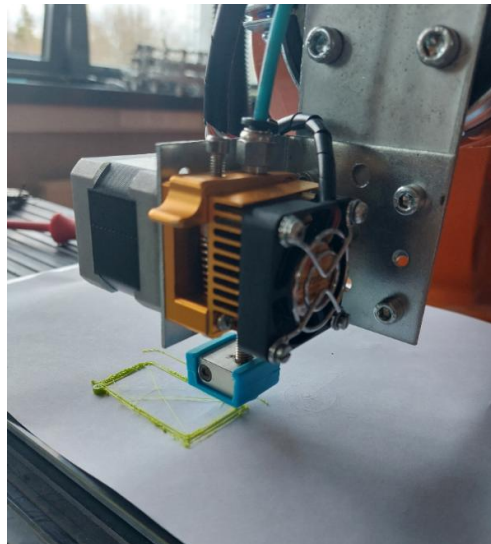
kde *speed\_mms* byl parametr odpovídající aktuální rychlosti robota v daném bodě,  $PRINT\_FLOW\_MAX\_SIGNAL = 10$  V. Výsledná hodnota byla zapsána na analogový výstup AO1.

Použitý nástroj byl nastaven jako  $DEFAULT\_TOOL\_ID = 1$  a použitý souřadnicový systém pro tisk  $DEFAULT\_BASE\_ID = 7$ . Po ukončení všech úprav postprocesoru, byla provedena další zkouška překladu programu a požadované funkce se přeložily správně.



## 4. Vlastní zkouška 3D tisku

V laboratoři byl na robotu nastaven nástroj *Extruder* (Tool[1]) přes TCP (Tool Center Point) pomocí čtyř bodové metody a základna pro tisk *3D\_Tisk\_podložka* (Base[7]) pomocí tří bodové metody. Tato informace byla převedena i do virtuálního prostředí RoboDK a následně byl znovu přeložen program pro robota. Při projetí programu robota bylo zjištěno, že se příruba robota nastavila kolmo k podložce, a přitom bylo požadováno, aby byla vůči podložce o 90° otočena. Byla provedena korekce v nastavení nástroje v robotu, a to taková, že do souřadnice B bylo doplněno natočení zápěstí robota o 90°. V následujícím pokusu projetí programu robotem, se již zápěstí robota nastavilo dle očekávání. Dalším problémem bylo, že byly vygenerované analogové hodnoty pro rychlost a teplotu desetkrát větší, než byl výpočet z postprocesoru. V manuálu robota [19] bylo zjištěno, že je požadované napětí vynásobeno číslem (faktorem), jehož velikost byla dána použitou analogovou kartou. V tomto případě napěťovou kartou 0-10 V a v nastavení HW měl tento faktor hodnotu 10. Na základě takto získaných informací byly vzorce (3.1) a (3.3) upraveny v postprocesoru, a to tak že byly vynásobeny konstantou 0,1. Po překladu programu bylo provedeno další projetí programu robotem, kde se vykonaly všechny požadované funkce dle očekávání. Všechny tyto testy probíhaly bez materiálu, a proto proběhl poslední test s vloženým materiálem Obr. 4.1.



Obr. 4.1 První test 3D tisku

Výsledkem tohoto testu bylo úspěšné provádění 3D tisku. Byly ovšem nalezeny chyby, jako např. nedokonalá rovnost podlahy a následkem toho i podložky, dle předpokladu byla zjištěna nepřesnost robota v prostoru daná jeho mechanikou a rychlost podávání materiálu při změně trajektorie robota.

## Závěr

Tato práce představuje rozšíření ročníkového projektu, v němž došlo ke shrnutí problematiky 3D tisku a následně výběru vlastního řešení, které bylo v této práci dále rozpracováno. Pro komunikaci mezi použitou řídicí elektronikou, od výrobce Arduino a robotem KR16-2 s nosností 16 kg od firmy KUKA s řídicím systémem KRC4, byla zvolena výměna řídicích a informačních signálů pomocí digitálních a analogových vstupů a výstupů. Elektronika od firmy Arduino je osazena procesorem Atmel ATmega 2560 a tento procesor disponuje napětím 5 V pro výměnu digitálních signálů a napětím 0-5 V pro vstupní analogový signál. Zatímco robot KR16-2 měl k dispozici pro digitální výměnu signálů napětí 24 V a pro analogovou výměnu signálů byl osazen napěťovou kartou 0-10 V. Z těchto důvodů byly navrženy a vyrobeny převodníky pro jednotlivé signály. Pro využití celého rozsahu analogového převodníku bylo nutné navrhnout a vyrobit napájení vstupní a výstupní části tohoto převodníku. Dále bylo navrženo další možné rozšíření pro využití více materiálů (vyhřívání podložky). Veškeré vyrobené komponenty byly osazeny a zapojeny na desku řídicí elektroniky spolu s napájecím zdrojem a tato deska byla následně umístěna na zvolené místo na robotu. Na přírubu robota byl umístěn i držák s extruderem, který bylo nutné při testech dodatečně upravit a následně byl tento extruder zapojen do řídicí elektroniky.

Pro řídicí elektroniku byl vygenerován firmware (FW) v online konfigurátoru a tento FW byl následně upraven pro danou aplikaci. Mezi nejdůležitější úpravy se řadí přepracování řízení 3D tiskárny z *G* kódu na pouhé ovládání motoru extruderu pomocí digitálních signálů a řízení rychlosti pomocí analogového signálu z robota. Ovládání ohřevu extruderu bylo přepracováno z *M* kódů také na digitální signály pro start ohřevu a pro nastavení teploty byl opět použit analogový signál z robota.

Překlad *G* kódů na pohyb robota se synchronním řízením motoru extruderu, podle rychlosti robota, byl popsán v předposlední kapitole. Na základě toho byla provedena rešerše a byl vybrán program RoboDK, který obsahuje překladač *G* kódu do jazyka KRL tzv. postprocessor. Tento program vytváří virtuální prostředí za použití robota (KR16-2) a jeho nástroje (extruder). Došlo k úpravě postprocesoru pro překlad souřadnice *E*, jako volaného podprogramu a pro překlad *M* kódů použitých v této práci.

Byl proveden pokusný 3D tisk, při něm bylo zjištěno, že by bylo vhodné zapracovat na rovnosti podložky nebo doplnit robota o snímač odměřování vzdálenosti a zapracovat jej do projektu. Dále došlo ke zjištění, že je nutné ještě více propracovat řízení rychlosti podávání materiálu při změně trajektorie robota a pokusit se eliminovat nepřesnost robota v prostoru pomocí softwaru.

## Seznam použité literatury

- [1] ATmega640-1280-1281-2560-2561-Datasheet. *Microchip.com* [online]. [cit. 2023-03-28]. Dostupné z:  
<https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf>
- [2] PC817 optocoupler: Datasheet, Circuit and Equivalent. *Https://www.utmel.com/* [online]. [cit. 2023-03-29]. Dostupné z:  
<https://www.utmel.com/components/pc817-optocoupler-datasheet-circuit-and-equivalents?id=2150>
- [3] Lite-On Inc. LTL-4251N. *Https://www.digikey.at* [online]. [cit. 2023-03-29].  
Dostupné z: <https://www.digikey.at/en/products/detail/lite-on-inc/LTL-4251N/149419>
- [4] Lite-On Inc. LTL-4231N. *Https://www.digikey.at* [online]. [cit. 2023-03-29].  
Dostupné z: <https://www.digikey.be/en/products/detail/lite-on-inc/LTL-4231N/214440>
- [5] IL300 Linear Optocoupler, High Gain Stability, Wide Bandwidth. *Https://www.vishay.com* [online]. [cit. 2023-03-29]. Dostupné z:  
<https://www.vishay.com/docs/83622/il300.pdf>
- [6] Broadcom HCNR200 and HCNR201 High-Linearity Analog Optocouplers. *Https://www.broadcom.com/* [online]. [cit. 2023-03-29]. Dostupné z:  
<https://docs.broadcom.com/doc/HCNR200-HCNR201-High-Linearity-Analog-Optocouplers-DS>
- [7] NPN Transistor 2N3903, 2N3904. *Https://www.onsemi.com/* [online]. [cit. 2023-03-29]. Dostupné z: <https://www.onsemi.com/pdf/datasheet/2n3903-d.pdf>
- [8] SVOBODA, Emanuel, a kol. *Přehled středoškolské fyziky. 2.*, přepracované vyd. Praha: Prometheus, 1996. ISBN 80-7196-006-3. S. 258–259.
- [9] L78S positive voltage regulator. *Https://www.st.com/* [online]. [cit. 2023-03-29].  
Dostupné z: <https://www.st.com/en/power-management/l78s.html>

- [10] Heated Bed PCB 200x200mm 12/24V MK2B. *Https://reprap.world/* [online]. [cit. 2023-03-29]. Dostupné z: <https://reprap.world/build-plate/heaters/heated-bed-pcb-200x200mm-12-24v-mk2b/#specifications>
- [11] Štěpán Berka; Elektrotechnická schémata a zapojení 2; BEN - technická literatura, Praha 2010, ISBN 978-80-7300-254-1, str. 228 (Měrný odpor kovů - Rezistivita)
- [12] Microcon Technické parametry krokového motoru SX17-1005LQCEF. *Http://www.krokovemotory.cz* [online]. [cit. 2023-03-29]. Dostupné z: <http://www.krokovemotory.cz/SX17-1005LQCEFds.pdf>
- [13] DRV8825 Stepper Motor Controller IC. *Https://www.ti.com/* [online]. [cit. 2023-03-29]. Dostupné z: [https://www.ti.com/product/DRV8825?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=asc-mdbu-null-prodfolderdynamic-cpc-pf-google-wwe\\_int&utm\\_content=prodfolddynamic&ds\\_k=DYNAMIC+SEARCH+ADS&DCM=yes&gclid=CjwKCAjw5pShBhB\\_EiwAvmnNV3hTW5CfMB\\_LN7nQBFa0KMUwS3QYqvsDi5W-VORkxeB8rtOMb03rHRoCUTIQA\\_vD\\_BwE&gclsrc=aw.ds](https://www.ti.com/product/DRV8825?utm_source=google&utm_medium=cpc&utm_campaign=asc-mdbu-null-prodfolderdynamic-cpc-pf-google-wwe_int&utm_content=prodfolddynamic&ds_k=DYNAMIC+SEARCH+ADS&DCM=yes&gclid=CjwKCAjw5pShBhB_EiwAvmnNV3hTW5CfMB_LN7nQBFa0KMUwS3QYqvsDi5W-VORkxeB8rtOMb03rHRoCUTIQA_vD_BwE&gclsrc=aw.ds)
- [14] GCode. *Https://www.robot-forum.com* [online]. [cit. 2023-03-29]. Dostupné z: <https://www.robot-forum.com/robotforum/thread/27535-how-can-i-integrate-the-g-code-into-krl/>
- [15] Zuka. *Https://github.com* [online]. [cit. 2023-03-29]. Dostupné z: <https://github.com/mnourgwad/zuka>
- [16] ŁYGAS, K. a W. DANILCZUK. Translator G-codu na jazyk robota Kawasaki. *Http://yadda.icm.edu.pl* [online]. [cit. 2023-03-29]. Dostupné z: <http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-4e47c661-25b5-4cf5-9456-a06b5dd62430>
- [17] RoboDK Simulate Robot Applications. *Https://robodk.com/* [online]. [cit. 2023-03-29]. Dostupné z: <https://robodk.com/download>
- [18] RoboDK Documentation. *Https://robodk.com* [online]. [cit. 2023-03-29]. Dostupné z: <https://robodk.com/doc/en/Basic-Guide.html>

- [19] KUKA System Software 8.3. *Http://www.wtech.com.tw* [online]. [cit. 2023-03-29].  
Dostupné z:  
<http://www.wtech.com.tw/public/download/manual/kuka/krc4/KUKA%20KSS-8.3-Programming-Manual-for-SI.pdf>

## Seznam příloh na CD

Schéma zapojení a desky plošného spoje (DPS) převodníku 24 V/5 V

Schéma zapojení a DPS převodníku 5 V/24 V,

Schéma zapojení a DPS analogového převodníku 0-10 V/0-5 V

Schéma zapojení a DPS stabilizátoru 5 V (10 V)

Zdrojový kód upraveného firmwaru

Upravený postprocesor *KUKA\_KRC4\_DAT* pro RoboDK

Projekt pro zkušební tisk (rectangle) z programu RoboDK

Program (rectangle) pro robot KUKA

Videozáznam z prvního testu tisku