



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DATA KONCENTRÁTOR SENZORŮ A PŘÍSTROJŮ 3D TISKÁRNY

DATA CONCENTRATOR OF 3D PRINTER'S SENSORS AND DEVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Samuel Vaculík

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jakub Arm, Ph.D.

BRNO 2024

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Samuel Vaculík

ID: 240462

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Data koncentrátor senzorů a přístrojů 3D tiskárny

POKYNY PRO VYPRACOVÁNÍ:

Práce je zaměřena na integraci IoT systému pro sběr dat z 3D tiskárny pomocí stávajících a přidaných senzorů. Dalším zdrojem dat je řídicí jednotka tiskárny a jiné přístroje. Měřená data z jednotlivých zařízení budou koncentrována pomocí edge device a efektivně posílána do cloudové služby.

1. Navrhněte systém pro přenos dat ze zařízení 3D tiskárny.
2. Realizujte hardware systému pro sběr dat.
3. Vypočítejte datové toky a navrhněte metody pro jejich redukci.
4. Relizujte software systému pro sběr dat.
5. Vyhodnoťte vlastnosti a parametry vytvořeného systému.

DOPORUČENÁ LITERATURA:

Kirianaki, N., Yurish, S., Shpak, N., Deynega, V. Data Acquisition and Signal Processing for Smart Sensors. Wiley; 1st Edition, 2002. ISBN: 978-0470843178.

Termín zadání: 5.2.2024

Termín odevzdání: 22.5.2024

Vedoucí práce: doc. Ing. Jakub Arm, Ph.D.

Ing. Miroslav Jirgl, 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

Bakalářská práce se skládá z návržení a implementace systému pro sběr dat souvisejících s 3D tiskem. Cílem je definovat veličiny vhodné k měření, vybrat senzory pro daná měření a implementovat je na tiskárnu. Dalším bodem je návrh a realizace sběru dat z tiskového firmwaru Klipper pro shromažďování tiskových dat. Následně je úkolem realizovat software pro zpracování dat z jednotlivých měřicích zdrojů, včetně analýzy datových toků v systému.

KLÍČOVÁ SLOVA

3D tisk, Zpracování dat, Klipper, Moonraker, Mainsail, Raspberry Pi, Arduino, Python, MQTT, Mosquitto

ABSTRACT

The bachelor thesis consists of designing and implementing a system for collecting data related to 3D printing. The aim is to define the variables suitable for measurement, select sensors for the measurements and implement them on the printer. The next point is the design and implementation of data acquisition from the printing firmware Klipper for collecting print data. Then the task is to implement the software for processing data from the individual measurement sources, including data flow analysis in the system.

KEYWORDS

3D printing, Data processing, Klipper, Moonraker, Mainsail, Raspberry Pi, Arduino, Python, MQTT, Mosquitto

VACULÍK, Samuel. *DATA KONCENTRÁTOR SENZORŮ A PŘÍSTROJŮ 3D TISKÁRNÝ*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2024. Vedoucí práce: doc. Ing. Jakub Arm, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Samuel Vaculík
VUT ID autora: 240462
Typ práce: Bakalářská práce
Akademický rok: 2023/24
Téma závěrečné práce: DATA KONCENTRÁTOR SENZORŮ A
PŘÍSTROJŮ 3D TISKÁRNY

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 20.5.2024.

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jakubovi Armovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	23
1 Stávající řešení	25
1.1 Tiskárna	25
1.2 Použitý hardware	26
1.3 Tiskový firmware	27
1.4 Uživatelské rozhraní	28
2 Návrh senzorické části	29
2.1 Teplota v okolí tiskárny	29
2.2 Vlhkost v okolí tiskárny	30
2.3 Teploty krokových motorů	31
2.4 Teploty procesorů	33
2.5 Detekce plynů	33
2.5.1 Měření koncentrace oxidu uhličitého	33
2.5.2 Měření koncentrace oxidu uhelnatého	34
2.6 Vibrace	34
2.6.1 Manuální kompenzace vibrací	35
2.6.2 Automatická kompenzace vibrací	35
2.7 Hmotnost filamentu	36
2.7.1 Snímač hmotnosti	36
2.7.2 Spoolman rozhraní	37
2.8 Průměr filamentu	37
2.9 Proudý v motorech	38
2.10 Zvuk	39
2.11 Obraz	40
3 Realizace senzorické části	41
3.1 Rozmístění senzorů	41
3.2 Zapojení senzorů	42
3.2.1 Zapojení do Arduino Nano	42
3.2.2 Zapojení do Raspberry Pi	43
3.3 Software a komunikace	44
4 Sběr dat z tiskového firmwaru Klipper	45
4.1 Protokol MQTT	45
4.2 Konfigurace Mosquitto brokeru	45
4.3 Konfigurace Moonrakeru	46

4.4	Konfigurace senzoru průměru filamentu	46
5	Zpracování dat v Raspberry Pi	47
5.1	Koncept	47
5.2	Hlavní program	47
5.3	Podprogramy	48
5.3.1	Příjem dat	49
5.3.2	Uložení dat do souboru	51
5.3.3	Komprimace souboru	52
5.3.4	Odeslání na server	53
6	Datové toky v systému	55
6.1	Mosquitto	55
6.2	Arduino Nano	55
6.2.1	Vzorkovací frekvence	55
6.2.2	Rozlišení dat	56
6.3	Audio sběrnice	56
6.4	Kamerový přenos	56
6.5	Přenos na server	57
7	Otestování systému	59
7.1	Vyhodnocení dat z externích senzorů	59
7.2	Vyhodnocení dat z tiskového firmwaru	61
7.3	Možná vylepšení	62
	Závěr	63
	Literatura	65
	Seznam symbolů a zkratk	69
	Seznam příloh	71
A	Struktura celého systému	73
B	Uživatelská prostředí	75
B.1	Mainsail	75
B.2	Fluidd	75
C	Bloková schémata jednotlivých programů	77
C.1	Arduino program pro čtení dat ze senzorů	77
C.2	Hlavní program pro správu podprogramů	78

C.3	Program pro zpracování dat z Klipperu	79
C.4	Program pro zpracování dat z Arduina	80
C.5	Program pro zpracování dat z mikrofónů	81
D	Naměřená data ze senzorů v průběhu tisku	83
D.1	Vibrace	83
D.2	Teploty krokových motorů	84
D.3	Okolní teploty	85
D.4	Okolní vlhkosti	86
D.5	Koncentrace plynů	87
E	Naměřená data z tiskového firmwaru v průběhu tisku	89
E.1	Teplota Raspberry Pi	89
E.2	Teplota desky Octopus	90
E.3	Teplota hotendu	91
E.4	Teplota tiskové podložky	92
E.5	Spotřeba filamentu	93
E.6	Průměr filamentu	94
E.7	Trajektorie tiskové hlavy	95
F	Naměřená zvuková data při tisku	97
F.1	Časová oblast	97
F.2	Frekvenční oblast	98
G	Konfigurace Klipperu	99
G.1	Konfigurace MQTT v souboru moonraker.conf	99
G.2	Konfigurace senzoru průměru filamentu v souboru printer.cfg	101
G.3	Konfigurace kamer v souboru crowsnest.conf	102
H	Konfigurace Mosquitto brokera v souboru mosquitto.conf	103
I	Fotodokumentace	105
I.1	Umístění první kamery	105
I.2	Umístění druhé kamery	106
I.3	Senzory v horní části tiskárny	107
I.4	Držák teplotního čidla na krytu motoru	108
I.5	Senzor průměru filamentu	109
J	Obsah elektronické přílohy	111

Seznam obrázků

1.1	CoreXY kinematika [3]	25
1.2	Podávání filamentu - Přímé podávání vlevo a Bowden systém vpravo [4]	26
1.3	BigTreeTech Octopus Pro V1.0 [5]	27
1.4	Driver krokových motorů TMC2209 [6]	27
2.1	Modul SHT-31 [9]	30
2.2	Digitální čidlo DS18B20 [10]	31
2.3	Parazitní zapojení vlevo, třívodičové zapojení vpravo [12]	32
2.4	Detektory plynů MQ-135 (vlevo) [13] a MQ-7 (vpravo) [14]	34
2.5	Senzor hmotnosti YZC-131 (vlevo) [15] s AD převodníkem HX711 (vpravo) [16]	36
2.6	Ukázka prostředí Spoolman [17]	37
2.7	Použitý mikrofon SPH-0645LM4H [19]	40
3.1	Rozmístění senzorů na 3D modelu tiskárny [1]	41
3.2	Zapojení senzorů do mikrokontroleru Arduino Nano [20]	42
3.3	Přehled pinů Raspberry Pi [21]	43
3.4	Zapojení mikrofónů SPH-0645LM4H do Raspberry Pi [19] [22]	44
7.1	Tiskové modely pro test funkčnosti	59
B.1	Prostředí Mainsail [7]	75
B.2	Prostředí Fluidd [8]	75

Seznam tabulek

2.1	Porovnání senzorů SHT-31 a DHT-22	30
2.2	Parametry teplotního senzoru DS18B20 [11]	32
2.3	Rozlišení a čas teplotní konverze [11]	32
2.4	Porovnání základních parametrů mikrofونů	39
7.1	Statistická data ze senzorů	60
7.2	Statistická data z Klipperu	61

Seznam výpisů

2.1	Konfigurace měření teplot CPU v souboru printer.cfg	33
4.1	Konfigurace zabezpečení brokera	46
5.1	Ukázka parsování (dělení na části) JSON zpráv	48
5.2	Spuštění paralelního vlákna	49
5.3	Ukázka nastavení a čtení dat ze sériové linky	49
5.4	Přihlášení klienta na broker	50
5.5	Přihlášení klienta k odběru zpráv	50
5.6	Konfigurace audio zařízení	51
5.7	Ukázka zápisu hlaviček do CSV souboru	51
5.8	Ukázka použití knihovny Queue	52
5.9	Ukázka komprimace dat	52
5.10	Ukázka odesílání dat	53
5.11	Nahrání dotenv souboru v programu	54
6.1	Konfigurace kamery	57
G.1	Konfigurace MQTT [26]	99
G.2	Konfigurace senzoru průměru filamentu [23]	101
G.3	Konfigurace kamer [30]	102
H.1	Konfigurace Mosquitto brokera [29]	103

Úvod

V dnešní době přináší IoT (Internet of Things - Internet věcí) revoluční přístup k propojení a monitorování různorodých fyzických zařízení. Tato bakalářská práce se věnuje integraci 3D tiskárny do konceptu IoT za účelem monitorování a sběru dat, což otevírá cestu k efektivnějšímu a chytřejšímu provozu 3D tiskáren.

Práce vzniká jako pokračování dvou předchozích projektů. První z nich, realizovaný Janem Čtvrtníčkem, se zaměřil na revitalizaci 3D tiskárny. Díky instalovanému výložnému systému pro hotový výtisk je tiskárna do budoucna připravena pro autonomní provoz, případně pro zapojení do konceptu Průmysl 4.0. Tímto bylo dosaženo robustního základu pro další inovace. Druhá práce, od Jana Fialy, se soustředila na sběr provozních dat z 3D tiskárny. Zkušenosti a poznatky z této práce slouží jako inspirace pro zadaný aktuální úkol.

V první části se práce věnuje popisu fyzikálních veličin ovlivňující proces 3D tisku s následným výběrem senzorů pro daná měření. Na kapitulu je dále navázáno návrhem komunikace pro sběr dat z použitého tiskového firmwaru Klipper.

Druhá část práce je zaměřena na realizaci softwaru pro příjem a účinné zpracování změřených dat s cílem snížit datový tok mezi řídicí jednotkou Raspberry Pi a serverem. Detailnějšímu rozboru datových toků je věnována předposlední kapitola této práce.

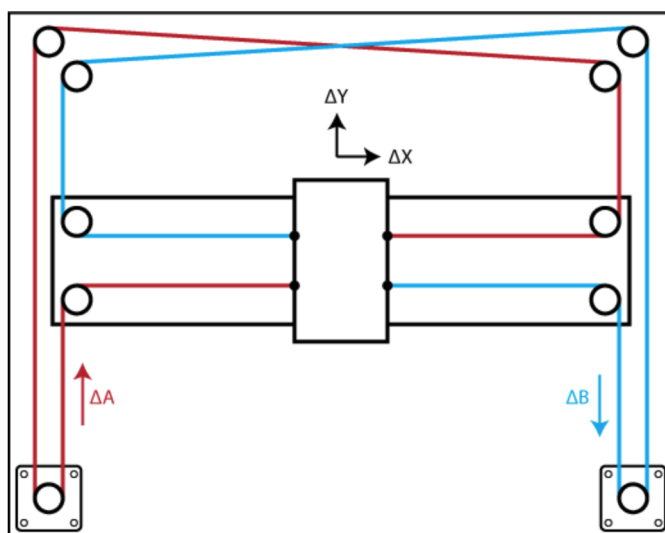
V závěrečné kapitole je poté komentována funkčnost celého měřicího systému, který byl otestován pomocí testovací sady tisků. Tato část dále rozebírá možnosti vylepšení celého systému.

1 Stávající řešení

Vzhledem k tomu, že práce bude provedena na již existující tiskárně, je vhodné pro další postup v práci popsat její základní parametry a vlastnosti, a to nejen z hlediska konstrukce tiskárny, ale také z hlediska použité elektroniky a softwaru.

1.1 Tiskárna

Tiskárna disponuje tiskovým prostorem 300x300x330 mm, řadí se tak spíše mezi větší 3D tiskárny. Kinematika tiskárny je založena na systému CoreXY (viz obr. 1.1). Ten se vyznačuje vysokou rychlostí pohybu tiskové hlavy, zároveň však bez ztráty kvality výtisku.



Obr. 1.1: CoreXY kinematika [3]

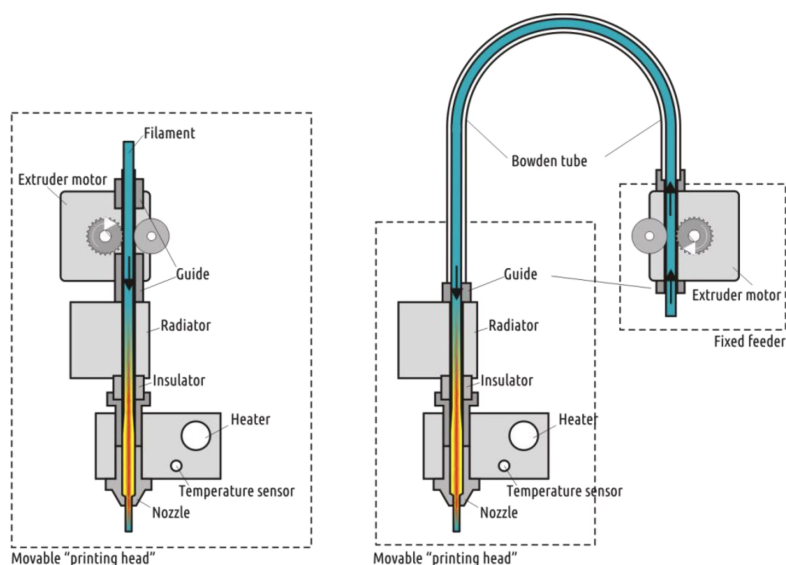
Hlavní výhoda tohoto systému oproti kartézské kinematice ¹ spočívá v tom, že tisková hlava se pohybuje v obou směrech X i Y a není tak potřeba, aby jeden motor pohyboval tiskovou plochou. Při větších rozměrech výtisku a tím pádem i větší hmotnosti je totiž motor více zatěžován a kvalita tisku by mohla klesat. S pohybem tiskové plochy vzniká u kartézské kinematiky další problém, neboť rozsah, ve kterém se tisková plocha pohybuje, je často větší než velikost samotné tiskárny v dané ose. Tento fakt poté zbytečně zvětšuje celkovou velikost tiskárny.

Pohyb tiskárny v ose Z je z důvodu velikosti tiskové plochy realizován čtyřmi krokovými motory, díky čemuž jsou jednotlivé motory méně zatěžovány. V dolní části tiskárny se nachází zmíněný výložník hotového výtisku. Vykládání je zajištěno

¹Kartézská kinematika využívá lineární pohyb ve všech třech osách.

čtyřmi ložisky, které jsou přes řemeny a hřídel ovládány jedním krokovým motorem. Samotný výložník však zatím nemůže být provozován, neboť k tiskárně není připojen žádný manipulátor, který by vysunutý tiskový plát spolu s výtiskem odebral.

Podávání tiskové struny (filamentu) do hotendu ² je realizováno Bowdenovou trubicí (viz obr. 1.2). Při tomto zapojení se extruder ³ nenachází na tiskové hlavě, a tudíž jsou motory os X a Y méně zatěžovány. Díky tomu lze dosáhnout lepší kvality tisku než u přímého podávání filamentu. Další nespornou výhodou tohoto systému jsou nižší vibrace na tiskové hlavě. Oproti přímému podávání filamentu však může mít Bowden systém problém při tisku výrobků s vyšším počtem retrakcí ⁴ kvůli pružnosti určitých materiálů.



Obr. 1.2: Podávání filamentu - Přímé podávání vlevo a Bowden systém vpravo [4]

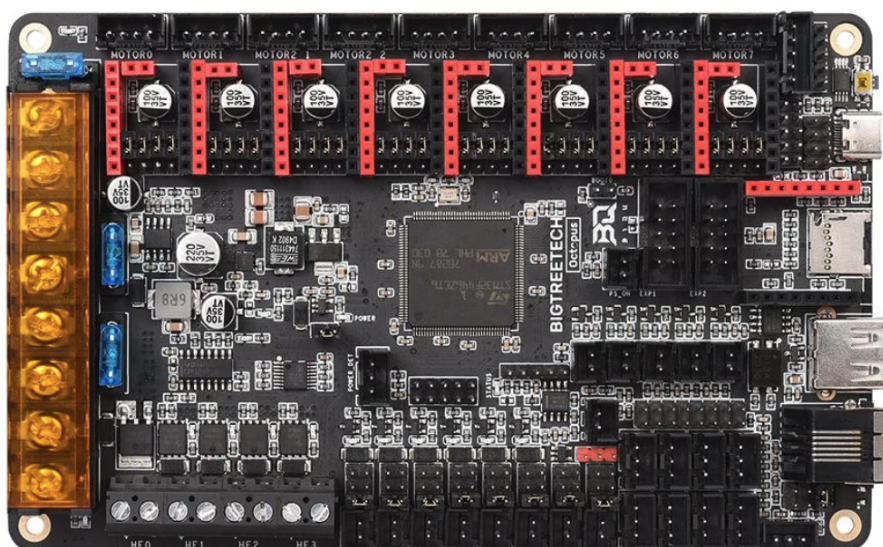
1.2 Použitý hardware

Pro řízení tiskárny je použita 32-bitová deska BigTreeTech Octopus Pro V1.0 (viz obr. 1.3). Jedná se o jednu z nejlepších desek, které výrobce aktuálně nabízí, a to díky podpoře až osmi motorů, čtyř hotendů či mnoha portů pro připojení periférií. Drivery krokových motorů jsou hojně používané TMC2208 a TMC2209 (viz obr. 1.4). Pro veškeré výpočty spojené s tiskem a pro běh tiskového firmwaru je použit jednodeskový počítač Raspberry Pi model 3B+. Deska obsahuje paměť RAM o velikosti 1 GB, což je dostačující velikost paměti pro běh tiskového firmwaru.

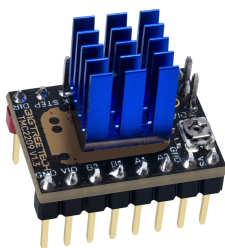
²část tiskárny, kde se materiál taví a poté pomocí trysky nanáší na tiskovou podložku

³motor zodpovědný za správný posun filamentu do hotendu

⁴proces zpětného pohybu filamentu



Obr. 1.3: BigTreeTech Octopus Pro V1.0 [5]



Obr. 1.4: Driver krokových motorů TMC2209 [6]

1.3 Tiskový firmware

Při výběru tiskového firmwaru má člověk v zásadě dvě možnosti - Klipper nebo Marlin. Jedná se o dva nejčastěji používané open-source firmwary. Klipper přistupuje k řízení 3D tiskárny zcela jiným způsobem než Marlin a ačkoliv je Marlin vyvíjen delší dobu, z hlediska výkonu poměrně zaostává. Při použití Marlin firmwaru se totiž veškeré výpočty a řízení odehrávají přímo na řídicí desce tiskárny, která je tak velmi zatěžována. Naopak při použití Klipper firmwaru se výpočty provádějí na externím procesoru (Raspberry Pi), který ve většině případů výrazně převyšuje výkon procesoru na řídicí desce. Řídicí deska tak pouze provádí operace bez nutnosti výpočtu, což má za následek rychlejší, ale především znatelně kvalitnější tisk. Tiskový firmware byl tedy ponechán na již nainstalovaný Klipper.

Další možností bylo použití RRF (RepRapFirmware). Ten je kromě 3D tiskáren určen například i pro laserové gravírování nebo CNC stroje. Na rozdíl od předchozích dvou jmenovaných je určen pouze pro 32-bitové procesory a nelze jej tak nahrát na většinu starších 3D tiskáren obsahující 8-bitové řídicí desky. [24]

1.4 Uživatelské rozhraní

Vzhledem k tomu, že je tiskárna díky výsuvnému systému připravena pro automatickou tiskovou farmu, je žádoucí, aby bylo možné k tiskárně přistupovat vzdáleně a nebylo tak potřeba při každém tisku nahrávat soubory do tiskárny pomocí SD karty. V předchozí práci bylo uživatelským rozhraním prostředí Octoprint, které je díky své jednoduchosti častým softwarem pro vzdálený přístup k tiskárnám. Speciálně pro Klipper byla však vytvořena další dvojice prostředí - Mainsail a Fluidt (viz obr. B.1 a B.2), které umožňují lepší interakci s tímto firmwarem. Z tohoto důvodu byl Octoprint zaměněn za Mainsail. Rozdíl oproti rozhraní Fluidt není nikterak zásadní. Důvodem použití prostředí Mainsail byla pouze lepší integrace historie tisku a celkové statistiky, které mohou hrát vyšší roli při použití tiskové farmy. Druhým důvodem by mohl být propracovanější G-code viewer, který v reálném čase na 3D modelu zobrazuje aktuální postup v tisku.

2 Návrh senzorické části

Pro kvalitní tisk je potřeba, aby byla tiskárna správně zkalibrována. Tento proces zahrnuje nastavení vzdálenosti mezi tryskou a tiskovou podložkou, nastavení hodnot PID regulátoru v hotendu, případně ve vyhřívané podložce nebo například kalibraci nerovnosti tiskové plochy pomocí Mesh Bed Leveling algoritmu. Dalšími důležitými parametry, které musí být pro precizní tisk nastavené jsou například *Pressure Advance*, který upravuje extruzi (posun filamentu do hotendu) v závislosti na rychlosti pohybu tiskové hlavy nebo *Input Shaping* kompenzující vibrace způsobené při pohybu tiskové hlavy.

Všechny tyto parametry jsou velmi důležité, avšak soustředí se pouze samotný tisk. V průběhu času se vlivem opotřebení některých součástí můžou parametry měnit a bez zpětné vazby informující o stavu samotné tiskárny není možné zjistit, zda tiskárna například nepotřebuje revizi či u ní nedochází k abnormálním jevům.

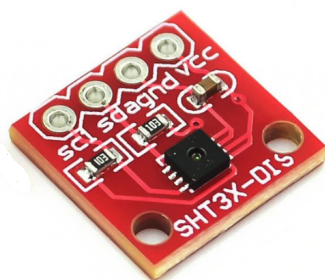
Tato kapitola se bude tedy dále zabývat popisem fyzikálních veličin, které je vhodné u 3D tisku měřit s následným výběrem senzorů.

2.1 Teplota v okolí tiskárny

Teplota je jednou z nejčastěji měřených veličin a je vhodné ji monitorovat i při 3D tisku. Hlavním důvodem je teplotní stabilita. Ta hraje výraznou roli při tisku materiálů typu ABS, případně ASA. Tyto materiály se vlivem náhlých změn teplot deformují a odlepují od tiskové podložky, tudíž je výtisk často nepoužitelný. Proto je vhodné teplotu měřit a při detekci teplotního výkyvu oznámit poruchu, případně výtisk zkontrolovat. Z tohoto důvodu je výhodné, aby byla tiskárna uzavřena v boxu, ve kterém se omezí proudění okolního vzduchu. Zároveň u těchto materiálů dochází při tisku k úniku nezdravých a zapáchajících výparů, které by se díky boxu nešířily dále do místnosti. Do boxu je však poté žádoucí nainstalovat filtr pro správnou filtraci výparů z filamentu.

Vzhledem ke značné velikosti tiskárny (50x50x100 cm) je vhodné měřit teplotu ve dvou místech. Sensory tak budou připevněny v dolní i horní části tiskárny. Pro snímání teploty byl zvolen modul SHT-31 s digitálním výstupem. Modul se připojuje na I^2C sběrnici a není tak potřeba dalších přírodních vodičů (viz obr. 2.1).

Původně měl být tento senzor použit jak v dolní, tak v horní části tiskárny, neboť senzor disponuje dvěma volitelnými adresami. Výrobce, od kterého byl modul zakoupen, však pin pro volení adresy uzemnil bez možnosti její změny, a proto nemohly být tyto senzory použity oba. Pro druhý senzor tak byla zvolena náhrada ve formě senzoru DHT-22.



Obr. 2.1: Modul SHT-31 [9]

Tab. 2.1: Porovnání senzorů SHT-31 a DHT-22

Parametr	SHT-31	DHT-22
Rozsah měření teploty	-40 až 125 °C	-40 až 80 °C
Přesnost měření teploty	$\pm 0,3$ °C	$\pm 0,5$ °C
Rozsah měření vlhkosti	0 až 100 %	0 až 100 %
Přesnost měření vlhkosti	± 2 %	± 2 %

Ačkoliv je senzor SHT-31 schopen měřit vyšší teploty, vnitřní teplota v uzavřeném boxu by neměla dosáhnout 80 °C, rozsah senzoru DHT-22 je tak dostačující. Nevýhodou oproti SHT-31 je nižší přesnost při snímání teploty a zároveň nutnost vedení samostatného datového vodiče do mikrokontroleru. Vzhledem k tomu, že tisk probíhá v horní části tiskárny ¹, bude i přesnější senzor SHT-31 umístěn právě do horní části.

2.2 Vlhkost v okolí tiskárny

Vlhkost nemá příliš velký dopad na samotnou tiskárnu (při uvážení normálních provozních podmínek) jako spíše na použité filamenty. Každý filament je do určité míry hygrokopický, čímž přirozeně absorbuje vlhkost ze vzduchu. U většiny filamentů se absorbovaná vlhkost projeví v rámci týdnů či měsíců. Některé filamenty však mohou být vlhkostí ovlivněny již v řádu hodin. [18] Výsledkem je poté nekonzistentní extruze, neboť absorbovaná vlhkost se v hotendu odpaří a tisková struna tak obsahuje množství dutin, které způsobí nerovnoměrný tisk.

Nejčastěji používané materiály typu PLA, PET-G, ASA či ABS trpí na vlhkost nejméně. Naopak materiál PVA je nutno uchovávat nepřetržitě v uzavřeném obalu,

¹Při CoreXY kinematice je posun v ose Z realizován pohybem tiskové plochy směrem dolů. Tisková hlava svou výšku v tomto případě nemění.

neboť tento materiál je rozpustný ve vodě a příliš velká vlhkost by jej tak začala nechtěně rozkládat.

Optimálním způsobem je používané filamenty ponechávat v externím uzavřeném vyhřívatelném boxu. Uvnitř by byla měřena vlhkost a při překročení daného limitu by se aktivovalo vyhřívání a odvlhčování.

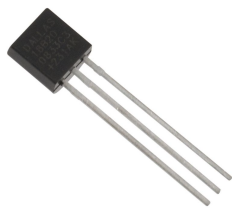
U stávající tiskárny je zatím předpoklad ponechání aktivního filamentu uvnitř boxu s tiskárnou. Vlhkost bude tedy měřena na dvou místech tiskárny. Pro měření budou použity již zmíněné senzory - SHT-31 a DHT-22, tudíž není potřeba dalších dvou senzorů pro daná měření.

2.3 Teploty krokových motorů

Krokové motory mohou při pohybu běžně dosahovat vysokých teplot. Měření aktuální teploty motoru má v tomto případě dva důvody. Prvním důvodem je detekce přehřátí motoru, neboť i přes vysoké pracovní teploty obvykle krokový motor výrazně nepřesáhne při tisku na běžné 3D tiskárně teplotu 100 °C.

Druhým důležitým důvodem je detekce rozdílných teplot motorů. Všechny čtyři motory v ose Z by při tisku měly dosahovat přibližně stejných teplot, neboť jsou totožně zatěžovány. Odlišné hodnoty by poté mohly znamenat poruchu na některém z motorů.

Měření teplot všech motorů bude realizováno digitálními čidly DS18B20 od firmy Dallas. Tento senzor je často používán z důvodu jeho spolehlivosti a jednoduchosti zapojení. Každé čidlo totiž obsahuje unikátní 64-bitovou adresu a lze tak provozovat více čidel na jedné sběrnici.



Obr. 2.2: Digitální čidlo DS18B20 [10]

Při teplotě 100 °C je chyba měření teploty přibližně 0,4 °C. Tato hodnota je naprosto dostačující k detekci výrazných teplotních rozdílů. Senzor má volitelné rozlišení 9 až 12 bitů, které umožňuje nastavit teplotní rozlišení 0,5 až 0,0625 °C. Nevýhoda většího rozlišení však spočívá v delším čase teplotní konverze, a tím pádem zpomaluje rychlost čtení dat (viz tab. 2.3).

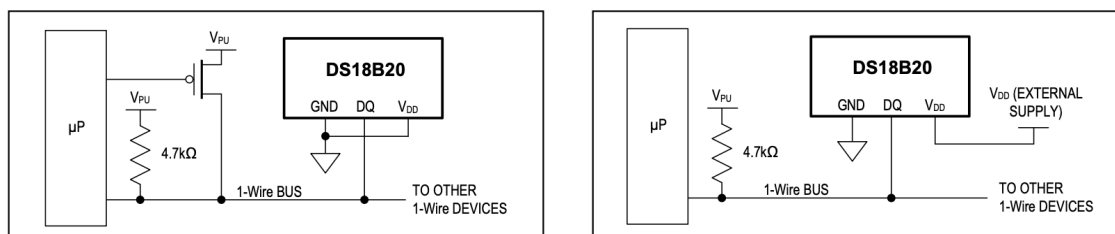
Tab. 2.2: Parametry teplotního senzoru DS18B20 [11]

Parametr	Hodnota
Napájecí napětí	3 až 5,5 V
Proudový odběr při měření	1 až 1,5 mA
Spotřeba při nečinnosti (do 70 °C)	750 až 1000 nA
Spotřeba při nečinnosti (při 125 °C)	3 μ A
Rozsah měření teploty	-55 až 125 °C
Přesnost měření teploty (-10 až 80 °C)	$\pm 0,5$ °C
Přesnost měření teploty (-55 až 125 °C)	± 2 °C

Tab. 2.3: Rozlišení a čas teplotní konverze [11]

Rozlišení	Teplotní rozlišení [°C]	Čas konverze [ms]
9 bitů	0,5	93,75
10 bitů	0,25	187,5
11 bitů	0,125	375,0
12 bitů	0,0625	750,0

Senzor dále disponuje funkcí alarmu při překročení daného limitu či parazitním zapojením (viz obr. 2.3). To využívá pouze dvou vodičů - DATA a GND, přičemž k napájení se využívá vodiče DATA. Do zapojení se však musí přidat další prvek (výrobce doporučuje použití unipolárního tranzistoru), který zajistí korektní napájení čidla. Výrobce však nedoporučuje používat toto zapojení při teplotách blízkých 100 °C, neboť teplotní konverze vyšších teplot vyžaduje delší čas a spolu s vyšším nastaveným rozlišením by vodič nemusel stíhat zařízení napájet a zároveň skrz něj komunikovat. Proto budou senzory zapojeny dle klasického třívodičového zapojení: V+, GND a DATA.



Obr. 2.3: Parazitní zapojení vlevo, třívodičové zapojení vpravo [12]

2.4 Teploty procesorů

Měření teplot procesorů na použitých deskách Raspberry Pi a Octopus Pro může přispět k optimalizaci provozních podmínek těchto zařízení. Detailní monitorování teploty umožňuje identifikovat potenciální problémy s přehříváním, což může vést k výkonnostním omezením. Získaná data také umožňují provádět úpravy chlazení, což bude mít za následek stabilnější provoz a delší životnost zařízení.

Vzhledem k tomu, že obě použité desky mají již zabudovaný termistor přímo uvnitř procesoru, není tak potřeba přidávat další teplotní senzory. Výhodou již zabudovaného snímače je také přesnější měření teploty, neboť externí senzor by musel být umístěn na čipu a mohl by tak být negativně ovlivňován okolním prostředím.

Pro měření teplot CPU tak stačí do konfiguračního souboru Klipperu dopsat pouze dvě sekce (viz výpis. 2.1) a teploty se již budou zobrazovat na hlavním panelu v rozhraní Mainsail.

Výpis 2.1: Konfigurace měření teplot CPU v souboru printer.cfg

```
[temperature_sensor raspberry_pi]
sensor_type: temperature_host
max_temp: 80

[temperature_sensor mcu_temp]
sensor_type: temperature_mcu
max_temp: 60
```

1
2
3
4
5
6
7

2.5 Detekce plynů

Při tavení většiny materiálů se uvolňují určité plyny. Stejně tomu tak je i u 3D tisku. Při procesu tavení tiskové struny se může uvolňovat oxid uhličitý, oxid uhelnatý, v případě PET-G materiálu také karcinogenní sloučeniny ethylbenzenu a styrenu [32]. Většina těchto plynů je nedýchatelných, popřípadě jedovatých a pokud by byla tiskárna uzavřena, mohly by se plyny při delším tisku akumulovat a při otevření boxu s tiskárnou na člověka negativně zapůsobit. Proto je vhodné koncentraci plynů měřit a případný nárůst koncentrace detekovat.

2.5.1 Měření koncentrace oxidu uhličitého

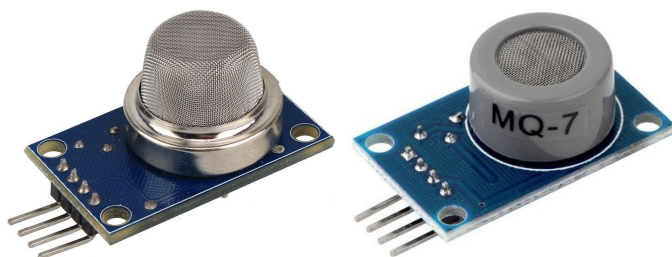
Monitorování oxidu uhličitého bude realizováno senzorem MQ-135. Snímač má maximální pracovní teplotu 65 °C, a tak jej lze provozovat i v uzavřeném boxu s tiskárnou. Modul je kromě CO_2 schopen detekovat i alkohol, benzen, či kouř. Detekci

kouře lze využít v případě, pokud by v hlavním zdroji tiskárny ($230 V_{AC}$ / $24 V_{DC}$) nebo na desce Octopus Pro došlo ke vzplanutí ohně.

Veškerá elektronika tiskárny je umístěna v její dolní části. Zároveň je plyn CO_2 těžší než vzduch a má tedy tendenci držet se u země. Z těchto dvou důvodů bude senzor MQ-135 umístěn do dolní části tiskárny pro správné monitorování obou zmíněných plynů.

2.5.2 Měření koncentrace oxidu uhelnatého

Ačkoliv by se při tisku nemělo uvolňovat nebezpečné množství oxidu uhelnatého, je vhodné jej v uzavřeném prostoru tiskárny měřit. Pro měření koncentrace CO byl zvolen senzor MQ-7. Tento modul oproti výše zmíněnému senzoru MQ-135 měří pouze oxid uhelnatý, což však splňuje potřeby měření. Výrobce uvádí maximální provozní teplotu $50\text{ }^{\circ}\text{C}$, tato hodnota by však při velikosti celkového boxu neměla být uvnitř dosažena, a tak je možné senzor používat. Pokud by vnitřní teplota při tisku přece jen dosahovala hodnot blízkých $50\text{ }^{\circ}\text{C}$, bude vzhledem k životnosti senzoru vhodné zvolit robustnější snímač.



Obr. 2.4: Detektory plynů MQ-135 (vlevo) [13] a MQ-7 (vpravo) [14]

2.6 Vibrace

Každý pohyb je více či méně doprovázen mechanickými vibracemi. Ty mohou být v určitých případech žádoucí, nýbrž u 3D tisku je snaha o jejich eliminaci. Při mechanickém vibrování totiž může docházet k posunutí tiskových vrstev, případně k nerovnoměrným stěnám na výtisku. Tiskárna by proto měla stát na podkladu, který bude vzniklé vibrace absorbovat a její konstrukce by měla být stabilní k co největšímu omezení vibrací.

Měření mechanického vibrování a následná detekce zvýšených vibrací by mohla predikovat povolení spojů v konstrukci, což by mohlo být následováno zhoršením stability.

Měření vibrací bude opět z důvodu velikosti konstrukce realizováno pomocí dvou modulů MPU-6050. První bude umístěn v dolní části a druhý v horní části tiskárny. Modul má v sobě zabudovaný 3-osý akcelerometr, 3-osý gyroskop a teplotní senzor. Komunikace probíhá přes rozhraní I^2C , přičemž senzor má možnost volby mezi dvěma adresami. Ačkoli má tento modul snímač teploty, z důvodu velké nepřesnosti jej není vhodné využívat.

Jak již bylo zmíněno, vibrace konstrukce se můžou negativně projevit na aktuálním tisku. Je tedy vhodné je nejen měřit, ale i zpětně kompenzovat (metoda Input Shaping).

Při použití tiskového firmwaru Klipper jsou dvě možnosti kompenzace.

2.6.1 Manuální kompenzace vibrací

Spočívá v tisku kalibračního modelu, při kterém se postupně zvyšuje akcelerace tiskové hlavy. Rychlé pohyby jsou totiž hlavním zdrojem vibrací. Při kalibračním tisku se tak zjistí, při jakých rychlostech mají již vibrace značný dopad na kvalitu tisku. Z hotového výtisku se poté dle návodu odečte frekvence s jakou se na tisku objevuje tzv. ringing (zvlnění materiálu). Z tohoto parametru se následně vybere vhodný Input Shaper algoritmus a zvolí se maximální přípustná akcelerace.

Pro tento postup není třeba dalšího modulu, což představuje výhodu této metody. Zásadní nevýhodou je však fakt, že při úpravě tiskové hlavy nebo změně tiskové podložky se změní celková kinematika tiskárny a kalibrační proces bude nejspíš nutné provést znova.

2.6.2 Automatická kompenzace vibrací

Jak již název napovídá, při této metodě tiskárna automaticky čte data z akcelerometru a podle nich upravuje hodnoty Input Shaperu. Není tedy nutné při každé úpravě provádět kalibraci.

Je však potřeba přidat další akcelerometr, který musí být namontován přímo na tiskové hlavě, jinak by kompenzace nebyla optimální. Z důvodu nedostatku času byl však tento bod vynechán.

2.7 Hmotnost filamentu

Jeden z obecných problémů při 3D tisku je docházející filament. Ne vždy je totiž obsluha poblíž tiskárny, aby množství zbývajících filamentu zkontrolovala, případně vyměnila za nový. Nejen u tiskových farem je proto důležité o docházejícím materiálu obsluhu včas informovat.

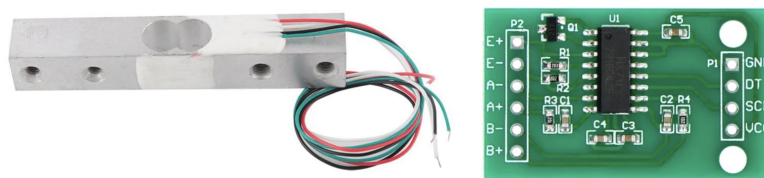
Zde se nabízí použití MMU (Multi Material Unit), se kterou je možno jednak tisknout více barvami při jednom tisku, ale také automaticky přepnout na druhou přichystanou cívku s navinutým filamentem při vypotřebování materiálu. Jednotka MMU je proto vhodná i pro jednobarevný tisk, neboť ulehčuje práci při výměně filamentu za jiný typ. Člověk tak není při každé změně materiálu nucen manuálně vysouvat a nasouvat materiály.

I při použití jednotky pro vícebarevný tisk však stále není možnost předčasně detekovat docházející materiál. Proto je vhodné využít váhu k tomu, aby byla hmotnost cívky s filamentem v reálném čase měřena. Poté je již jednoduché detekovat, že cívka bude brzy vypotřebována. S pomocí aktuální hmotnosti a znalosti potřebného materiálu pro daný tisk lze také před tiskem odhadnout, zda bude v průběhu potřeba cívku vyměnit.

2.7.1 Snímač hmotnosti

Pro měření hmotnosti byl vybrán vážící senzor YZC-131 s maximální zátěží 5 kg. Nejbližší další možnost byl rozsah 3 kg, případně 10 kg. Váha s maximální hmotností 10 kg je nicméně předimenzovaná, přičemž u váhy 3 kg bychom se při použití těžších cívek nacházeli na okraji měřicího rozsahu. Proto byla zvolena pětikilová varianta.

Senzor je dále doplněn 24-bitovým analogově-digitálním převodníkem HX-711, který aktuální hmotnost převede na digitální hodnotu. Ta bude poté v podobě pulzů čtena mikrokontrolerem.

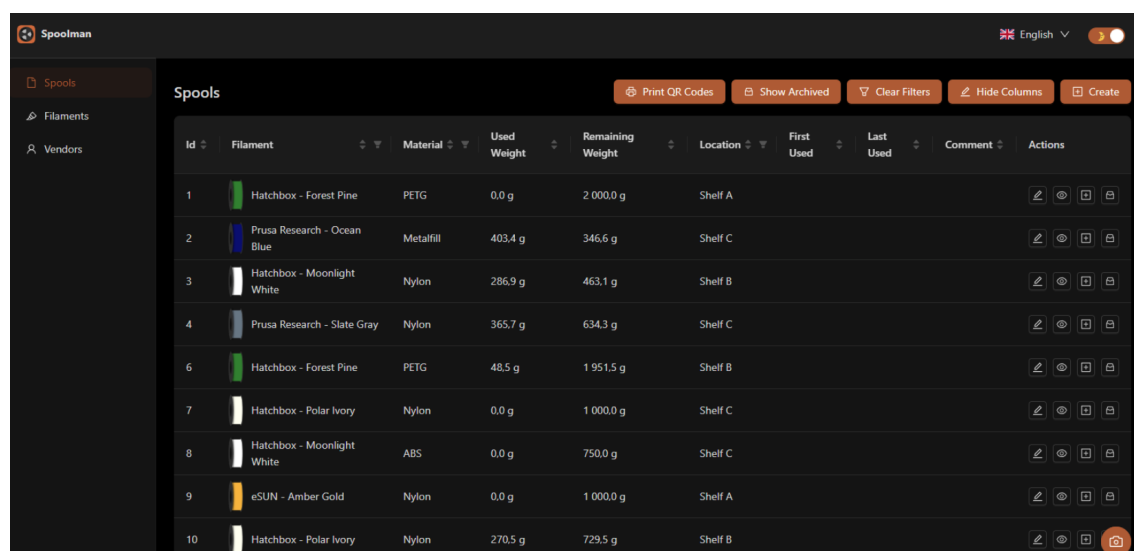


Obr. 2.5: Senzor hmotnosti YZC-131 (vlevo) [15] s AD převodníkem HX711 (vpravo) [16]

2.7.2 Spoolman rozhraní

I přes to, že vážení filamentu poskytuje dobrou zpětnou vazbu, není pravděpodobně optimálním řešením. Při použití více filamentů je totiž nutnost vážit každou cívku jednotlivě, protože prázdné cívky od různých výrobců mají jiné hmotnosti a naměřená data by tak neodpovídala skutečnosti. Druhou možností je softwarově měnit váhu prázdné cívky v závislosti na použitém materiálu. Obě dvě řešení však přináší více komplikací, než užitku.

Možným řešením je použití rozhraní Spoolman pro správu a přehled používaných filamentů. Hlavní výhodou je komunikace s prostředím Mainsail a zároveň automatický výpočet zbývajících filamentu v reálném čase. Veškeré filamenty a jejich délky jsou tak přehledně dostupné v této databázi (viz obr. 2.6). Před tiskem může být uživatel také upozorněn na nedostatek materiálu pro daný tisk. Bonusem je navíc i podpora *Home Assistant*, nástroje pro správu chytré domácnosti.



The screenshot shows the Spoolman web interface. On the left is a sidebar with navigation links: Spools, Filaments, and Vendors. The main area is titled 'Spools' and contains a table with 10 rows of filament data. Above the table are several action buttons: Print QR Codes, Show Archived, Clear Filters, Hide Columns, and Create. The table columns are: Id, Filament, Material, Used Weight, Remaining Weight, Location, First Used, Last Used, Comment, and Actions. Each row represents a different filament spool with its specific details.

Id	Filament	Material	Used Weight	Remaining Weight	Location	First Used	Last Used	Comment	Actions
1	Hatchbox - Forest Pine	PETG	0.0 g	2 000.0 g	Shelf A				[Edit] [View] [Delete] [Archive]
2	Prusa Research - Ocean Blue	Metalfill	403.4 g	346.6 g	Shelf C				[Edit] [View] [Delete] [Archive]
3	Hatchbox - Moonlight White	Nylon	286.9 g	463.1 g	Shelf B				[Edit] [View] [Delete] [Archive]
4	Prusa Research - Slate Gray	Nylon	365.7 g	634.3 g	Shelf C				[Edit] [View] [Delete] [Archive]
6	Hatchbox - Forest Pine	PETG	48.5 g	1 951.5 g	Shelf B				[Edit] [View] [Delete] [Archive]
7	Hatchbox - Polar Ivory	Nylon	0.0 g	1 000.0 g	Shelf C				[Edit] [View] [Delete] [Archive]
8	Hatchbox - Moonlight White	ABS	0.0 g	750.0 g	Shelf C				[Edit] [View] [Delete] [Archive]
9	eSUN - Amber Gold	Nylon	0.0 g	1 000.0 g	Shelf A				[Edit] [View] [Delete] [Archive]
10	Hatchbox - Polar Ivory	Nylon	270.5 g	729.5 g	Shelf B				[Edit] [View] [Delete] [Archive]

Obr. 2.6: Ukázka prostředí Spoolman [17]

2.8 Průměr filamentu

Nejpoužívanější průměr filamentu u 3D tisku je 1,75 mm. Tento konstantní průměr ovšem nikdy nelze dodržet po celé délce filamentu, a tak se uvádí obvyklá výrobní tolerance $\pm 0,05$ mm. Někteří výrobci mají sice odchylku průměru menší, u některých výrobců však může odchylka dosáhnout až $\pm 0,1$ mm. Extruder, respektive hotend, však počítá s průměrem 1,75 mm a má na tuto hodnotu nastavenou extruzi. Při výskytu ovality na tiskové struně může tak materiál z trysky vytékat nerovnoměrně, což ovlivní vzhled výtisku.

Proto je vhodné do cesty tiskové struny umístit senzor průměru filamentu, který bude ovalitu měřit a regulovat extruzi v závislosti na aktuální hodnotě průměru. Zároveň je tento senzor schopen detekovat přítomnost filamentu a přerušit tisk při docházejícím materiálu.

Senzor je složen ze dvou lineárních hallových sond SS49E, které jsou zapojeny v diferenciálním módu. Jsou tak generována dvě napětí, přičemž součet těchto napětí je přímo roven tloušťce filamentu. [23] Výhodou diferenciálního zapojení je stejný teplotní drift obou sond, který nebude negativně ovlivňovat měření.

2.9 Proudý v motorech

Měření proudu v motorových vinutích může spolu se znalostí teploty motoru vést k odhalení poruchy uvnitř motoru. Proudý ve vinutích jsou včetně veškerých dalších stavů a parametrů motoru počítány v motorových driverech obsahující velké množství registrů pro uložení daných proměnných. Není tak potřeba přidávat další měřicí systém pro měření proudu, ale stačí pouze vyčíst potřebné hodnoty z registrů.

Aktuální hodnoty proudů se ukládají do registru MSCURACT, který je rozdělen na dvě 9-bitové sekce (vinutí A a B), ve kterých jsou hodnoty uloženy ve formě 8-bitového čísla se znaménkem (označeno CUR_A a CUR_B). Pokud jsou tyto hodnoty z registru vyčteny, lze aktuální hodnoty proudů ve vinutích vypočíst pomocí vztahu:

$$I_{MOT}(A/B) = \frac{CUR_A/B}{248} \cdot \frac{CS + 1}{32} \cdot \frac{V_{FS}}{R_S + 20m\Omega} \quad (2.1)$$

Kde:

CS = měřítko, definované hodnotami $IRUN$ a $IHOLD$

V_{FS} = napětí v plném rozsahu definované kontrolním bitem $vsense$ (Pokud je $vsense$ bit roven nule ($V_{FS} = 0,32V$), driver dosahuje nejlepší a robustní proudové regulace, při nastavení $vsense$ bitu hodnotu 1 ($V_{FS} = 0,18V$) jsou naopak na snímacím rezistoru poloviční teplotní ztráty)

R_S = hodnota snímacího rezistoru při měření proudu

2.10 Zvuk

Závadu na mechanické úrovni lze detekovat pomocí senzorů vibrací. Ty však nemusí být schopny zachytit veškeré jevy a anomálie. Pro získání komplexnějšího obrazu stavu mechanických komponentů je proto vhodné měřicí řetězec doplnit o snímače zvuku. Aby mikrofony snímaly pouze děje spojené se samotnou tiskárnou a ne jejím okolím, je žádoucí tiskárnu provozovat v uzavřeném boxu nebo přinejmenším v tiché a méně frekventované místnosti.

Pro detekci zvuku byl zvolen mikrofon SPH-0645LM4H (viz obr. 2.7) od společnosti Adafruit disponující měřicím frekvenčním rozsahem 50 Hz až 15 kHz. Druhou možností bylo použít mikrofon INMP441, avšak z důvodu horších parametrů byl vybrán první jmenovaný (viz tab. 2.4).

Tab. 2.4: Porovnání základních parametrů mikrofonů

Parametr	SPH-0645LM4H	INMP441
f_R	50 - 15 000 Hz	60 - 15 000 Hz
SNR	65 dBA	61 dBA
PSR	-86 dBA	-75 dBFS
THD	1 % (SPL 110 dB)	3 % (SPL 105 dB)

Kde:

SNR (Signal to Noise Ratio) - Poměr výkonu signálu k výkonu šumu

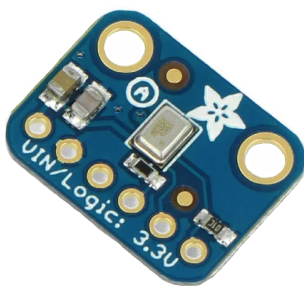
PSR (Power Supply Rejection) - Potlačení vlivu změny napájecího napětí

THD (Total Harmonic Distortion) - Míra zkreslení signálu

SPL (Sound Pressure Level) - Hladina akustického tlaku

Z tabulky lze vyčíst, že ve všech hlavních parametrech je model SPH-0645LM4H kvalitnější. Mechanické vibrace mají často frekvenci jen desítky Hz, volba mikrofonu SPH-0645LM4H s nižší měřitelnou frekvencí by tak mohla lépe zachytit dané signály. Další výhodou je nižší harmonické zkreslení i při vyšší hladině akustického tlaku.

Komunikace a čtení dat z mikrofonu probíhá přes třívodičovou sériovou I^2S sběrnici (Inter-IC Sound). Jedná se o sběrnici určenou primárně pro připojení zvukových zařízení, která vyžadují vysokou rychlost přenosu dat. Z tohoto důvodu maximální rychlost sběrnice přesahuje 6 MHz, zatímco například sběrnice I^2C dosahuje maximální rychlosti 1 MHz.



Obr. 2.7: Použitý mikrofón SPH-0645LM4H [19]

2.11 Obraz

Dříve byl přenos obrazu pouze nástrojem pro vzdálený dohled na 3D tiskárnu, případně jakékoli jiné zařízení. S rychle se rozvíjejícími algoritmy strojového učení lze však z obrazu automaticky detekovat vadný tisk, eventuálně jinou závadu. Použití kamery při 3D tisku je tak velice přínosné, a to hlavně z důvodu úspory materiálu při poškozeném tisku.

Výhodou prostředí Mainsail je konfigurace až čtyř kamer. Optimálním řešením je použití dvou kamer, přičemž první snímá detail správné extruze filamentu (především při první tiskové vrstvě je důležitá správná přilnavost k tiskové podložce) a druhá celkový pohled na tisknutý objekt.

Pro realizaci budou použity dvě webkamery s ohniskovou vzdáleností 3,85 mm a desetinásobným digitálním přiblížením. Obecně lze říci, že pro 3D tiskárny jsou vhodnější širokoúhlé kamery s nižší ohniskovou vzdáleností. Jedním z důvodů může být kvalitnější obraz i při nižším osvětlení v místnosti.

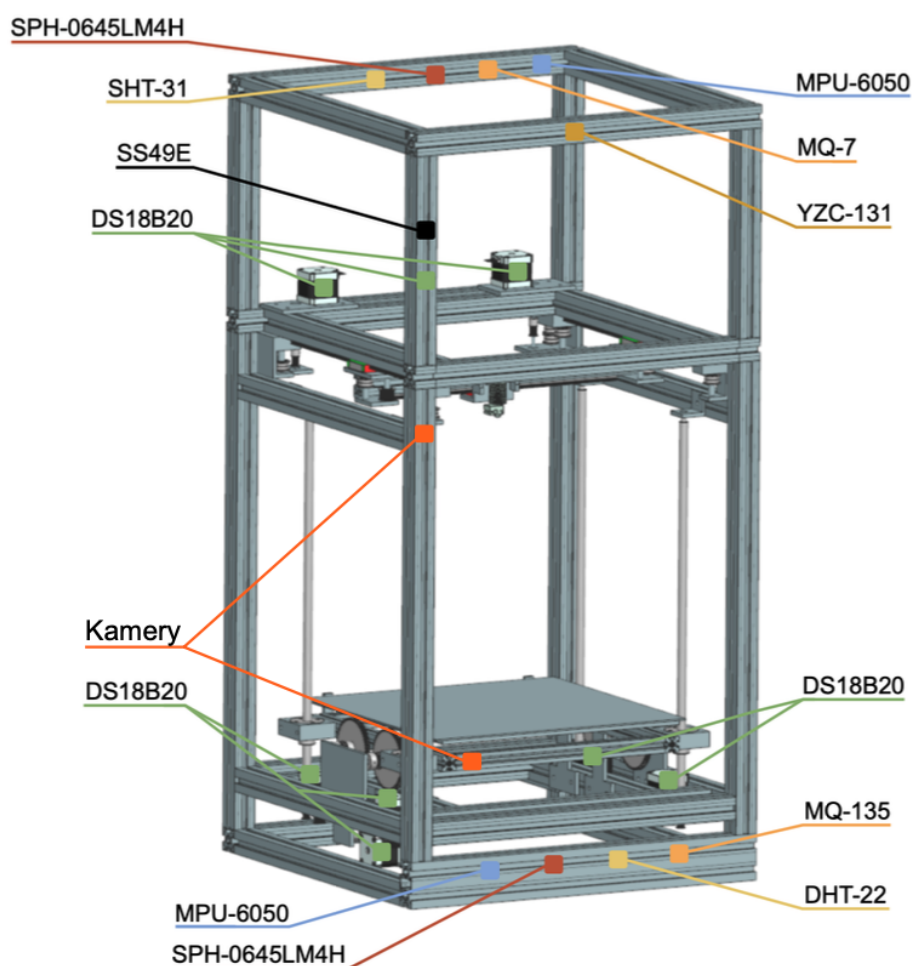
Jak již bylo zmíněno výše, špatné nastavení tiskárny může mít za následek nekvalitní tisk. Může však nastat i situace, kdy se vlivem složitého tvaru modelu výtisk zhroutl a tiskárna bude tak zbytečně tisknout do volného prostoru. V tuto chvíli je vzhledem k úspoře materiálu žádoucí tisk zastavit. Nabízí se tak použití softwarového doplňku *Spaghetti detective*. Ten je založen na modelu strojového učení a spolu s algoritmy pro rozpoznávání obrazu je tisk monitorován a v případě detekce havárie je buď obsluha informována, případně tisk zastaven.

3 Realizace senzorické části

V předchozí kapitole bylo popsáno měření jednotlivých fyzikálních veličin pomocí vybraných senzorů. Tato kapitola se bude zabývat použitím a zapojením jednotlivých snímačů k vytvoření celkového měřicího řetězce.

3.1 Rozmístění senzorů

Použité snímače byly na konstrukci tiskárny rozmístěny tak, aby dokázaly co nejlépe měřit jednotlivé fyzikální veličiny. Rozmístění senzorů je znázorněno na 3D modelu tiskárny.¹



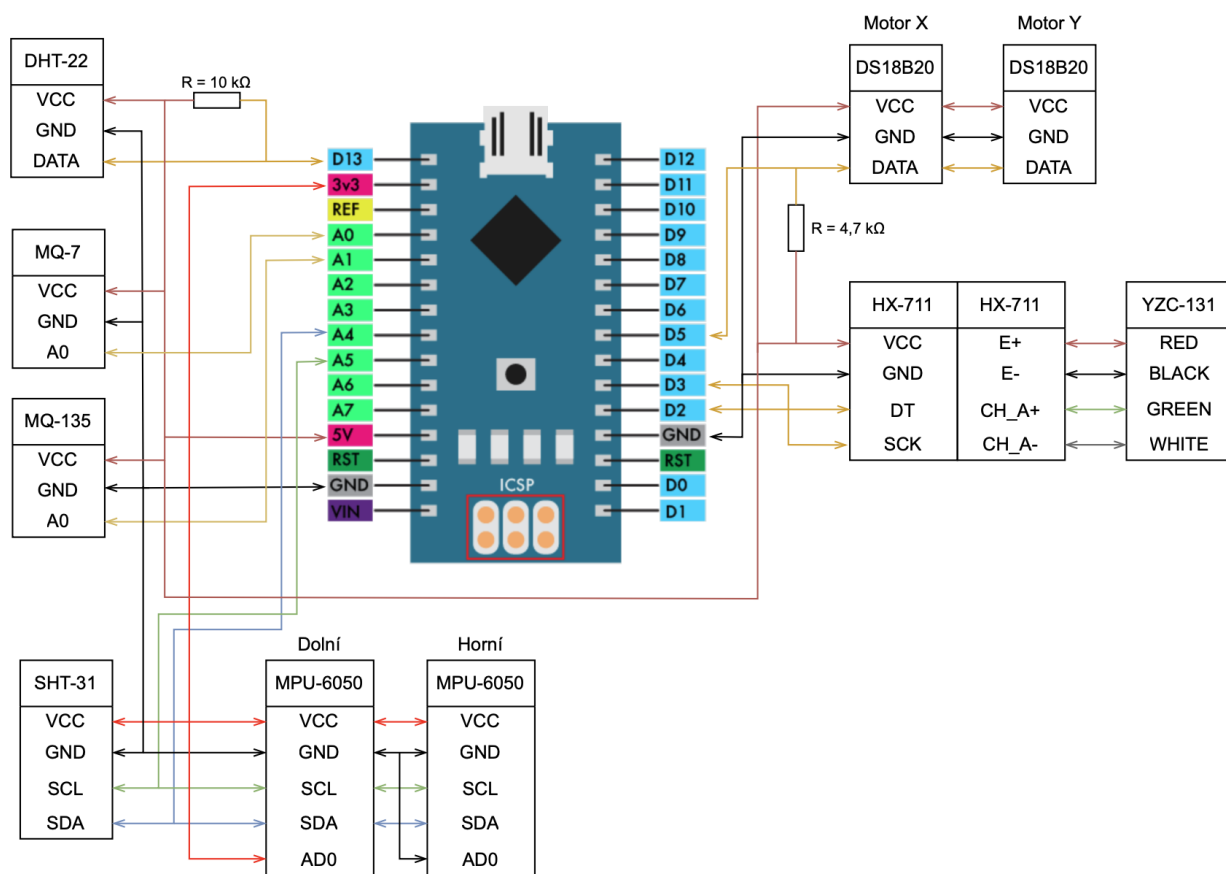
Obr. 3.1: Rozmístění senzorů na 3D modelu tiskárny [1]

¹Model vytvořil Jan Čtvrtníček při práci na revitalizaci tiskárny.

3.2 Zapojení senzorů

Většina senzorů bude komunikovat s jednodeskovým počítačem Arduino Nano.
² Ačkoliv tento modul disponuje na dnešní dobu poměrně pomalým 8-bitovým procesorem ATmega328 ³, pro účel sběru dat ze snímačů je však dostačující. Důvodem použití byla cena a také možnost programování v prostředí Arduino IDE.

3.2.1 Zapojení do Arduino Nano



Obr. 3.2: Zapojení senzorů do mikrokontroleru Arduino Nano [20]

²Arduino Nano nedisponuje sběrnici I^2S pro připojení audio zařízení, proto budou mikrofony SPH-0645LM4H připojeny do Raspberry Pi.

³Ve srovnání s rozměrově podobným modulem Raspberry Pi Pico W obsahující procesor RP2040 s maximální frekvencí procesoru 133 MHz, dosahuje ATmega328 maximální rychlosti procesoru pouze 20 MHz.

Veškeré senzory byly zapojeny dle doporučení výrobců. Snímače plynu MQ-135 a MQ-7 mají i digitální výstup, který přejde do aktivního stavu při překročení koncentrace daného plynu. Pro účely zpracování a vyhodnocení dat to však není vhodné, a proto byly snímače připojeny pomocí analogového výstupu. Na obrázku bylo kvůli úspoře místa vynecháno šest teplotních senzorů DS18B20, jejich připojení je však totožné. Senzory SHT-31 a MPU-6050 byly připojeny na zmíněnou sběrnici I^2C nacházející se na pinech A4 (Datový pin) a A5 (Hodinové impulzy).

3.2.2 Zapojení do Raspberry Pi

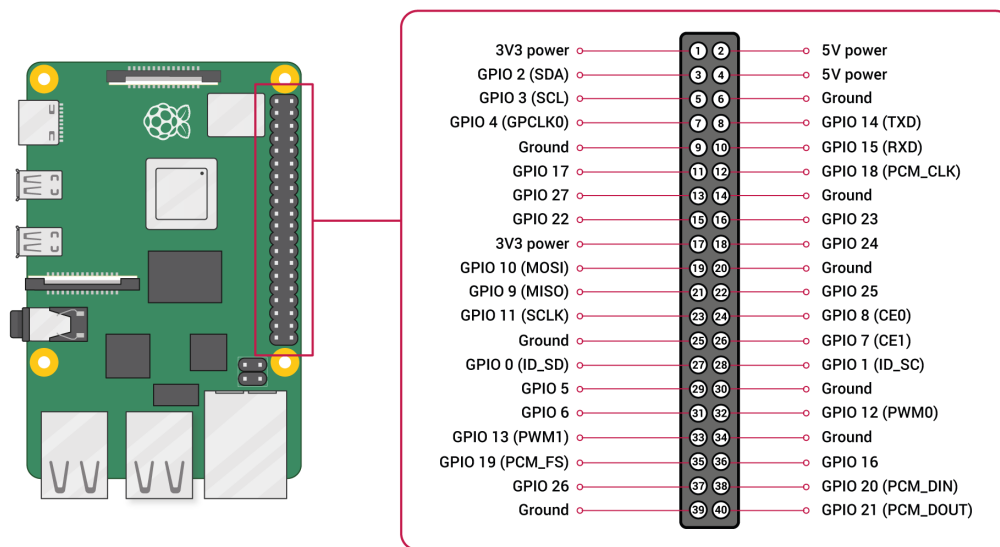
Popis jednotlivých pinů Raspberry Pi je na obr. 3.3. Zapojení mikrofونů bylo provedeno dle návodu pro stereo záznam zvuku (viz obr. 3.4).⁴

Pro připojení mikrofونů na zmíněnou sběrnici I^2S bylo potřeba tří vodičů:

BLCK - Hlavní hodinový signál pro přenos dat. Pro správnou komunikaci je doporučena minimální frekvence signálu o hodnotě 2 MHz.

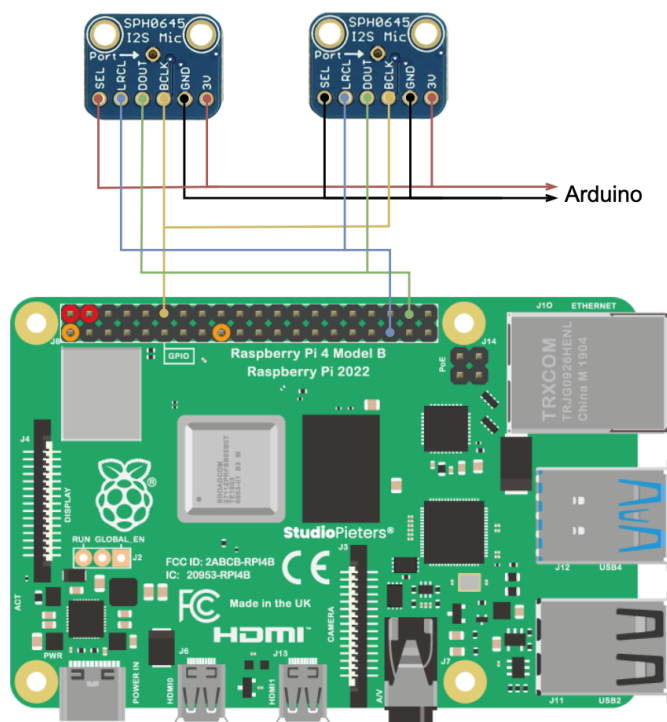
DOUT - Výstup dat z mikrofону

LRCL - Také označováno WS (Word Select). Definuje časy pro přenos dat do levého a pravého kanálu.



Obr. 3.3: Přehled pinů Raspberry Pi [21]

⁴Na obrázcích se nachází Raspberry Pi model 4, jednotlivé piny a jejich funkce jsou však stejné i pro 3. generaci.



Obr. 3.4: Zapojení mikrofونů SPH-0645LM4H do Raspberry Pi [19] [22]

3.3 Software a komunikace

Program pro čtení dat ze senzorů připojených do Arduino s následným odesláním do Raspberry Pi byl již z velké části realizován panem Fialou. Program byl doplněn o čtení dat z přidanych senzorů a upraven tak, aby vyhovoval požadavkům.

Struktura programu je na obr. C.1. Po spuštění programu se vytvoří instance jednotlivých senzorů a definují potřebné proměnné. Funkce *setup()* slouží k aktivaci senzorů a zahájení komunikace na sběrnici. Zároveň je v této funkci nastaveno rozlišení senzorů DS18B20 (popis v kapitole 2.3) a kalibrace senzoru oxidu uhelnatého. Poté jsou periodicky volány jednotlivé funkce, které do JSON ⁵ dokumentu vloží aktuální výstup daného senzoru. Největší nároky jsou kladeny na snímače vibrací, neboť rychlé pohyby tiskové hlavy způsobují rychlé změny hodnot akcelerometrů a při nižší frekvenci měření by se mohlo stát, že nebudou zachycena důležitá data. Čtení těchto hodnot proto probíhá s periodou 15 ms. Měření zvukových dat z mikrofونů včetně veškerého zpracování je popsáno v kapitole 5.

Kompletní program pro Arduino Nano lze nalézt v příloze.

⁵JSON (JavaScript Object Notation) je způsob zápisu dat určených pro přenos mezi programy, případně zařízeními. Důvodem jeho používání je jednoduchost zápisu a srozumitelnost pro člověka, zároveň jednoduchost generování a parsování pro počítače. [25]

4 Sběr dat z tiskového firmwaru Klipper

Předchozí kapitoly se zabývaly návrhem a realizací snímačů pro měření fyzikálních veličin, které mohou ovlivnit 3D tisk. Jedním z hlavních požadavků kladených na tuto práci bylo také měření, respektive sběr dat z tiskového firmwaru. To zahrnuje čtení veškerých dat spojených s aktivním tiskem. V následující kapitole bude tedy navrhnout a realizován systém pro sběr dat z firmwaru Klipper.

Ačkoli je cílem sbírat data z tiskového firmwaru, Klipper za tento přenos není zodpovědný. Pro správnou interakci mezi Klipperem a prostředím Mainsail se využívá komunikačního rozhraní Moonraker. Jedná se o API ¹ web server založený na jazyku Python 3. Přenos tiskových dat bude probíhat přes MQTT protokol, pro který má Moonraker připravenou implementaci.

4.1 Protokol MQTT

MQTT (Message Queuing Telemetry Transport) je standardizovaný protokol typu Publish–Subscribe vytvořený primárně pro IoT. Hlavními částmi protokolu jsou Klienti a Broker. Ten je spojením mezi odesílateli a příjemci, přičemž jeho cílem je přeposílat zprávy daným klientům. Každá zpráva obsahuje tzv. `topic`. Klienti obdrží odeslané zprávy pouze v případě, že daný `topic` odebírají (z tohoto důvodu je protokol typu Publish–Subscribe). Tímto způsobem je realizováno posílání dat mezi zařízeními.

Broker byl zvolen Mosquitto, což je široce používaný open-source broker pro MQTT protokol.

4.2 Konfigurace Mosquitto brokeru

Pro úspěšné zprovoznění MQTT brokeru bylo nejprve nutné brokera nainstalovat a poté nastavit, aby se automaticky spouštěl po zapnutí Raspberry Pi. Následně bylo třeba do konfiguračního souboru `mosquitto.conf` dopsat sekci umožňující komunikaci mezi jednotlivými zařízeními (viz výpis H.1).

¹API (Application Programming Interface) je programové rozhraní pro daný web umožňující vzájemnou komunikaci mezi různými aplikacemi na softwarové úrovni.

Jakýkoli přenos dat mezi zařízeními je vhodné z hlediska bezpečnosti zašifrovat, případně zabezpečit uživatelským jménem a heslem. Při komunikaci s brokerem jsou tři možnosti zabezpečení: [27]

1. Ověření pomocí jména a hesla - klasický způsob zabezpečení
2. Ověření pomocí pluginů - vyšší kontrola nad autorizací než u předchozího způsobu, přičemž způsoby ověřování lze měnit v závislosti na použitém pluginu
3. Bez ověření - bez zabezpečení, na broker se může připojit jakýkoli klient

V tomto případě byla zvolena první možnost zabezpečení dat.

Výpis 4.1: Konfigurace zabezpečení brokera

<code>listener 1883</code>	1
<code>allow_anonymous false</code>	2
<code>password_file /etc/mosquitto/passwd</code>	3

4.3 Konfigurace Moonrakeru

Pro zprovoznění komunikace přes MQTT bylo zapotřebí napsat do konfiguračního souboru `moonraker.conf` sekci `mqtt` (viz výpis G.1).

Ačkoli je v manuálu pro konfiguraci Klipperu sekce pro odesílání dat z TMC driverů, nelze přes MQTT odesílat hodnoty registrů. Jediné užitečné informace, které lze odeslat jsou parametry `run_current` a `hold_current`. Tyto parametry se však po většinu času nemění, tudíž jejich odesílání není nutné. Zatím tedy není vyřešeno, jakým způsobem se budou proudy v motorech zpracovávat. Jedinou možností jak hodnoty proudů v tuto chvíli zobrazit, je vypsání registrů do konzole v prostředí Mainsail příkazem `DUMP_TMC STEPPER=<name>` (Při dodatku `[REGISTER=MSCURACT]` bude vypsán pouze registr obsahující aktuální proudy v motoru.)

4.4 Konfigurace senzoru průměru filamentu

Senzor ovality filamentu byl zapojen přímo do desky Octopus Pro do dvou analogových pinů. Konfigurace senzoru probíhala v souboru `printer.cfg` dle Klipper manuálu [23]. Po zapojení senzoru bylo zapotřebí jej kalibrovat. Kalibrace spočívala ve vložení dvou dílů, u kterých byl přesně znám jejich průměr. Při vložených dílech se z Klipperu vyčetla analogová hodnota hallových sond. Vzhledem k tomu, že použité sondy mají lineární závislost výstupního napětí na magnetickém poli, byla vytvořena přímka, ze které poté Klipper při daném výstupním napětí určuje průměr filamentu.

5 Zpracování dat v Raspberry Pi

V minulých kapitolách byly popsány fyzikální veličiny vhodné k měření a hardware, jakým budou jednotlivá data sbírána. Následující kapitola se proto bude zabývat návrhem a realizací softwarové části celého systému. Veškeré zpracování dat bude soustředěno do Raspberry Pi, které tak bude sloužit jako edge device pro celý systém 3D tiskárny.

Pro částečnou tvorbu programů byla v této části práce využita generativní umělá inteligence ChatGPT.

5.1 Koncept

Před samotnou realizací je vhodné definovat požadavky na zpracování dat a vymyslet optimální řešení pro daný problém. Jednou z možností je vytvoření jednoduchého programu, který by nepřetržitě přijímal data a dále je zpracovával. Tento princip by však zbytečně zatěžoval Raspberry Pi (případně server), neboť by zpracovával data i pokud by byla tiskárna neaktivní.

Vzhledem k tomu, že je do budoucna předpoklad naměřená data vyhodnocovat pomocí algoritmů strojového učení, je zapotřebí realizovat systém, který bude data zpracovávat pouze při tisku. Pokud by algoritmy strojového učení určovaly stav tiskárny z dat, která nesouvisí se samotným průběhem tisku, mělo by to za následek mnohem menší efektivitu a méně přesnou předpověď stavu tiskárny.

V návaznosti na předchozí odstavec bude softwarová část rozdělena na čtyři programy. První a hlavní program bude zodpovědný za čtení tiskových dat, které obsahují i informace ohledně doby tisku. Jakmile program detekuje výskyt této informace ve zprávě přijaté přes MQTT protokol, znamená to, že je tisk spuštěn. V té chvíli hlavní program spustí tři podprogramy pro čtení dat z Arduina, mikrofónů a tiskových dat. Stejným způsobem pak budou tyto podprogramy ukončeny, jakmile se ve zprávách přestane vyskytovat informace o tiskové době.

Ačkoli hlavní program i program pro zpracování tiskových dat využívají MQTT protokol a mohly by tak být v jednom programu, pro přehlednost a hierarchickou strukturu budou tyto programy rozděleny.

5.2 Hlavní program

Příchozí zprávy z MQTT protokolu jsou v JSON formátu. Pro hledání potřebných informací v příchozích zprávách bude využit balíček `parse` z knihovny `jsonpath_ng`.

Příklad hledání daných frází je zobrazen níže:

Výpis 5.1: Ukázka parsování (dělení na části) JSON zpráv

<code>printDuration = parse("\$.status.print_stats.print_duration")</code>	1
<code>totalDuration = parse("\$.status.print_stats.total_duration")</code>	2

Pro správnou funkčnost programu musí být použity dvě proměnné. Proměnná `totalDuration` začne čítat jakmile se spustí tisk. Před samotným tiskem však vždy chvíli trvá, než se nahřeje tisková plocha (případně také hotend, který je ovšem ve většině případů nahřátý za mnohem kratší dobu). Použití této proměnné pro spuštění měření by tedy nebylo vhodné. Z toho důvodu je pro spuštění podprogramů hledán výskyt druhé proměnné `printDuration`, která se ve zprávě objeví, jakmile je tiskárna připravena k tisku a provede první pohyb.

Občas je potřeba tisk pozastavit, ať už kvůli doplnění filamentu, případně vložení jiného dílu do výtisku. Při pozastavení se však proměnná `printDuration` vynuluje a ve zprávách se přestane objevovat. Řízení podprogramů pomocí této proměnné by tak mělo za následek jejich ukončení. Místo toho se pro ukončení využije již zmíněná první proměnná.

Vzhledem k tomu, že se vyhledává pouze výskyt proměnných a ne jejich konkrétní hodnoty, je zajištěna bezpečnost při náhlém výpadku MQTT brokera a ztrátu případných zpráv, což by mohlo mít za následek neukončení podprogramů na konci tisku.

5.3 Podprogramy

Pro správu podprogramů a provádění paralelních operací se v programech využijí knihovny `subprocess` a `threading`. První zmíněná knihovna slouží k spouštění externích skriptů běžících mimo hlavní program. S pomocí této knihovny tak bude řízeno spouštění a ukončování podprogramů pro zpracování dat. Se spuštěnými podprogramy lze poté komunikovat pomocí standardního vstupu a výstupu, čehož bude využito při předávání jména CSV souborům (viz dále).

Druhá knihovna slouží k aktivaci paralelního vlákna, které však sdílí paměť a prostředky s hlavním programem. Tato knihovna bude tedy využita v podprogramech pro běh paralelní smyčky zajišťující pravidelné zpracování dat, které tak nebude zpomalovat hlavní běh programu pro čtení dat.

Výpis 5.2: Spuštění paralelního vlákna

```
threading.Thread(target=call_function, daemon=True).start()
```

1

Příklad výše zobrazuje spuštění vlákna v programu. První parametr `target` definuje, co se má v paralelní smyčce spustit. Druhý parametr `daemon` určuje, zda poběží vlákno i po ukončení programu, který jej spustil. V tomto případě je nastaveno ukončení vlákna ihned po ukončení programu.

Všechny tři programy pro zpracování dat využívají strukturu popsanou v následujících podkapitolách.

5.3.1 Příjem dat

Příjem dat je prvním článkem řetězce pro zpracování. Každý program využívá jiný druh komunikace, a tudíž budou použity tři způsoby přijímání dat.

Sériová linka

Pro práci se sériovou linkou bude využita knihovna `serial`. Následný princip čtení dat je zobrazen níže.

Výpis 5.3: Ukázka nastavení a čtení dat ze sériové linky

```
ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1)
ser.flushInput()

while True:
    if ser.is_open and ser.in_waiting > 0:
        try:
            line = ser.readline().decode('utf-8').rstrip()
```

1

2

3

4

5

6

7

Nejprve je v programu třeba vytvořit instanci sériové linky. Zde je nutné definovat port, na jakém je Arduino připojeno včetně rychlosti komunikace. Funkce na druhém řádku zajišťuje vyprázdnění vstupního bufferu, který by mohl obsahovat nežádoucí data. Poté je již v nekonečné smyčce realizováno čtení příchozích dat a jejich správné formátování pro následnou práci s daty.

Protokol MQTT

Příjem dat přes MQTT protokol je realizován knihovnou `paho.mqtt`. Aby příjem dat fungoval správně, je potřeba se nejprve přihlásit na adresu brokera.

Výpis 5.4: Přihlášení klienta na broker

```
def connect_mqtt() -> mqtt_client:
    def on_connect(client_mqtt, userdata, flags, rc):
        if rc == 0:
            print(f"Connected to MQTT broker!")
        else:
            print(f"Failed to connect! {rc}")

    client = mqtt_client.Client()
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client
```

Následně je nutné přihlásit klienta k odběru daného topic názvu. V případě přenosu dat z Moonrakeru je topic `octopus/klipper/status`.

Výpis 5.5: Přihlášení klienta k odběru zpráv

```
def subscribe(client: mqtt_client):
    def on_message(client_mqtt, userdata, msg):
        store_data(msg.payload.decode())

    client.subscribe(topic)
    client.on_message = on_message
```

Audio sběrnice

Posledním typem komunikace je přenos zvukových dat přes I^2S sběrnici. Pro tyto účely bude využita knihovna `pyaudio` obsahující základní příkazy pro práci se zvukovými soubory. V první řadě je nutné stáhnout a spustit skript `i2smic.py` který provede konfiguraci pro přenos dat po dané sběrnici. Následně je důležitá správná konfigurace audio zařízení, viz níže.

Výpis 5.6: Konfigurace audio zařízení

```
p = pyaudio.PyAudio()  
  
audio_stream = p.open(format=pyaudio.paInt16,  
                        channels=2,  
                        rate=44100,  
                        input=True,  
                        input_device_index=1,  
                        frames_per_buffer=32768)
```

1
2
3
4
5
6
7
8

5.3.2 Uložení dat do souboru

Data ze tří měřicích zdrojů přicházejí v pravidelných intervalech. Odesílat na server vždy pouze jednu přijatou zprávu by bylo nevýhodné a také časově náročné. Proto budou přicházející zprávy ukládány do souborů, které se poté komprimují a budou odeslány na danou adresu.

Jedním z nejpoužívanějších formátů pro ukládání číselných dat do tabulek je formát CSV. Pro práci s tímto formátem lze s výhodou použít integrovanou Python knihovnu `csv`. Data budou v souboru ukládána do řádků, přičemž každý sloupec bude odpovídat danému parametru dle hlavičky sloupce.

Výpis 5.7: Ukázka zápisu hlaviček do CSV souboru

```
with open(csv_file, 'w', newline='') as csvfile:  
    writer = csv.DictWriter(csvfile, fieldnames=headers)  
    writer.writeheader()
```

1
2
3

Aby bylo možné určit, který soubor odpovídá jakým datům a kterému tisku, musí mít soubory unikátní název. Při spuštění tisku je přes MQTT protokol odeslána zpráva obsahující přesný název aktuálního procesu, tak jak byl uživatelem pojmenován. Hlavní program tedy převezme tuto informaci, odstraní `.gcode` příponu stejnou pro jakýkoli soubor a přidá na konec speciální časové ID pro případ opakovaných tiskových úloh. Následně tento název předá jako argument spouštěným podprogramům.

Pokud by se do souboru zapisovalo okamžitě, jakmile jsou přijata nová data, musel by být soubor neustále otevřen pro zápis nebo by se musel otevírat při každém přijetí dat. Druhá možnost není časově efektivní a udržovat soubor otevřený po celou dobu běhu programu také není vhodné.

Z těchto důvodů bude pro pravidelné ukládání dat využita fronta. Jakmile se poté objeví požadavek na odeslání dat, otevře se CSV soubor, data se z fronty do souboru přemístí a soubor bude uzavřen. Druhým důvodem využití fronty je větší robustnost proti ztrátě dat, ke které by mohlo dojít při příchodu dat ve chvíli kdy je soubor komprimován a odeslán.

Výpis 5.8: Ukázka použití knihovny Queue

<code>data_queue = Queue()</code>	1
	2
<code>data_queue.put(row)</code>	3
	4
<code>while not data_queue.empty():</code>	5
<code> writer.writerow(data_queue.get())</code>	6

5.3.3 Komprimace souboru

Čím větší je soubor určený pro odeslání na server, tím déle trvá jeho přenos. Vzhledem k tomu, že při přenosu není podstatné, zda je soubor čitelný pro člověka, je vhodné soubory před jejich odesláním komprimovat. Pro komprimaci dat byla zvolena knihovna `gzip` společně s modulem `shutil` určeným pro manipulaci se soubory, který umožní jednoduché kopírování dat mezi soubory.

Výpis 5.9: Ukázka komprimace dat

<code>with open(csv_file, 'rb') as f_orig:</code>	1
<code> with gzip.open(compressed_csv_file, 'wb') as f_comp:</code>	2
<code> shutil.copyfileobj(f_orig, f_comp)</code>	3

Na zobrazeném kódu probíhá kopírování dat z originálního do komprimovaného souboru, přičemž kopírovaná data se zároveň komprimují. Knihovna `gzip` využívá bezztrátových komprimačních algoritmů pro kompresi dat, tudíž data lze zpětně dekomprimovat bez ztráty informace, což je v tomto případě žádoucí. Knihovna také využívá automatického nastavení kompresního poměru na základě obsahu dat.

Použitá knihovna komprimuje soubor do GZ formátu. Jedná se o jeden z nejběžnějších kompresních formátů v systémech Linux, který se od hojně používaného ZIP formátu liší způsobem komprimace. Ačkoli oba formáty využívají stejných kompresních principů, formát ZIP aplikuje kompresi na jednotlivé soubory, zatímco formát GZ komprimuje celý archiv jako jeden celek. [34]

5.3.4 Odeslání na server

Poslední částí řetězce pro zpracování dat je odesílání komprimovaných CSV souborů na server. Pro tyto účely bude využita knihovna `requests` poskytující základní příkazy pro komunikaci se serverem.

Server pro příjem souborů zatím není spuštěný, proto bude v tuto chvíli využita pouze testovací služba [33], díky které bude otestována funkčnost systému. Tato stránka poskytuje základní metody pro HTTP požadavky, včetně příkazu `POST`, kterým se vytvoří požadavek na odeslání dat.

Výpis 5.10: Ukázka odesílání dat

<pre>response = requests.post(url, files=csv_file, timeout=5)</pre>	1
<pre>response.raise_for_status()</pre>	2

Funkce na druhém řádku slouží ke kontrole stavového kódu ¹ odpovědi na daný požadavek a vyvolá výjimku, pokud bude stavový kód chybový. Tímto způsobem tak lze případné výjimky zpracovat.

Zabezpečení serveru

Pro správný přístup a odeslání požadavku na server je vždy potřeba provést autentizaci, která zabrání klientovi přístup na danou adresu bez znalosti uživatelského jména a hesla. Pro tento případ obsahuje použitá služba metody pro testování zabezpečení. Pro tyto účely bude zvolena první možnost ověření pomocí uživatelského jména a hesla.

Informace o uživatelských jménech a heslech není vhodné uchovávat v programu jako proměnné, neboť by k těmto datům měl kdokoli přístup. Proto bude využita knihovna `dotenv`, která skryje citlivé údaje do speciálního souboru. V programu se poté tento soubor načte a do proměnných se přiřadí dané informace.

Stejným způsobem jsou uschovány údaje pro přihlášení na Mosquitto broker.

¹Stavový kód je trojčíferné číslo oznamující, zda byl požadavek úspěšný či nikoli. První cifra udává kategorii odpovědi (2xx - Úspěch, 4xx - Error klienta, 5xx - Error serveru)

Výpis 5.11: Nahrání dotenv souboru v programu

<code>dotenv_path = '/home/pi/BP_Vaculik/.env'</code>	1
<code>load_dotenv(dotenv_path)</code>	2
	3
<code>server_username = os.getenv('SERVER_USERNAME')</code>	4
<code>server_password = os.getenv('SERVER_PASSWORD')</code>	5
<code>mosquitto_username = os.getenv('MOSQUITTO_USERNAME')</code>	6
<code>mosquitto_password = os.getenv('MOSQUITTO_PASSWORD')</code>	7

Bloková schémata podprogramů jsou zobrazeny ve výpisech C.3, C.4 a C.5.

Kompletní programy jsou k dispozici v příloze.

6 Datové toky v systému

Jakýkoli přenos dat je doprovázen datovým tokem. Ten určuje množství dat, které se přes komunikační rozhraní přenesou za časový úsek. Množství přenesených dat definuje Bit Rate, který udává počet bitů přenesených za jednotku času, případně Baud Rate určující počet symbolů přenesených za jednotku času.

Navržený systém obsahuje různé způsoby komunikace a je tedy vhodné určit jednotlivé datové toky mezi zařízeními. Cílem následující kapitoly je tedy definovat veškeré datové toky v systému a navrhnout způsoby pro jejich redukci.

6.1 Mosquitto

V datasheetu pro konfiguraci Moonrakeru přes MQTT je definována perioda odesílání dat 250 *ms*. Při testování přenosu byla zjištěna průměrná délka odesílaných zpráv 360 *bytů*, neboli 2880 *bitů*. Během jedné vteřiny se tedy odešlou čtyři zprávy a datový tok je tak 11,5 *kbps*.

Perioda odesílání dat nelze měnit, jedinou možností jak ovlivnit datový tok je odesílat pouze ta data, ze kterých lze poté vyhodnocovat stav tiskárny. Z tohoto důvodu jsou z tiskového firmwaru odesílány výhradně informace, které v čase neustále mění svou hodnotu (viz výpis G.1)

6.2 Arduino Nano

Perioda odesílání dat z akcelerometrů přes sériovou linku je nastavena na 15 *ms*. Ostatní senzory mají mnohem delší dynamiku měření, a proto jsou ostatní veličiny měřeny s periodou 5 *s*. Vypočtená průměrná délka JSON zprávy je 85 *bytů*. Perioda 15 *ms* znamená, že data jsou odeslána 67krát za vteřinu, což poté určuje datový tok 45,5 *kbps*.

Maximální délka JSON zprávy byla změřena 120 *bytů*, neboli 960 *bitů*. Tato hodnota odpovídá datovému toku 64,3 *kbps*. Rychlost sériové linky byla poté zvolena jako nejbližší vyšší hodnota, tedy 115,2 *kbps*.

Redukci datového toku lze v tomto případě ovlivnit dvěma způsoby.

6.2.1 Vzorkovací frekvence

Čím vyšší bude vzorkovací frekvence (dále jen f_V), tím více budou data odpovídat real-time měření, avšak za cenu zvýšení datového toku. Zároveň může dojít při zvyšování f_V k omezení vlivem parametrů snímače. Příkladem by mohl být snímač

teploty a vlhkosti DHT-22, který má předepsanou periodu měření 2 s. V případě čtení hodnot ze snímače s kratší periodou může dojít k nesprávným výsledkům vlivem zahřívání samotného snímače. Naopak zásadní snížení f_V by mohlo mít za následek ztrátu dat vlivem případných výkyvů měřených veličin.

6.2.2 Rozlišení dat

Přesnost veličiny se obvykle vyjadřuje desetinným číslem. Přenos delšího desetinného čísla ovšem v případě textového přenosu zvyšuje datový tok. Proto je vhodné optimalizovat číselné vyjádření jednotlivých veličin. Pro většinu veličin (teplota, vlhkost, koncentrace plynů) bylo pevně stanoveno vyjádření s jedním desetinným číslem. Vzhledem k ceně, a tudíž přesnosti použitých senzorů, je nastavené vyjádření dostačující. Největší důraz byl kladen na senzory vibrací, které tak mají nastaveno rozlišení na tři desetinná místa. Data o hmotnosti filamentu jsou přenášena v celých číslech, neboť vzhledem k neznámé váze prázdné cívky je vyšší přesnost zbytečná. Poslední veličinou jsou teploty krokových motorů, které jsou měřeny čidly DS18B20 přímo umožňující definovat rozlišení snímače (viz tab. 2.3). Zvoleno bylo 9bitové rozlišení s rozlišením teploty 0,5 °C, což je dostačující přesnost pro danou úlohu.

6.3 Audio sběrnice

Rychlost I^2S sběrnice je dána především počtem kanálů, frekvencí vzorkování a bitovým vyjádřením zvukových dat. Vzhledem k tomu, že jsou na sběrnici připojeny dva mikrofony, je třeba využít stereo záznam, přičemž je použit 16bitový formát pro jednotlivé vzorky. Frekvence vzorkování byla stanovena na 44,1 kHz, což zaručuje dostatečnou kvalitu zvukových dat. Součinem těchto čísel dostáváme Bit Rate 1,4 Mbps.

Datový tok by v tomto případě mohl být snížen nižší vzorkovací frekvencí, případně použitím 8bitového vyjádření. Obě tyto varianty by však zhoršily kvalitu záznamu, což není žádoucí.

6.4 Kamerový přenos

Dvě použité kamery jsou do Raspberry Pi připojeny pomocí USB rozhraní, přičemž obrazová data jsou pouze zobrazována v prostředí Mainsail. Níže je zobrazena definice kamery v konfiguračním souboru `crowsnest.conf`.

Výpis 6.1: Konfigurace kamery

```
[cam 2]
mode: mjpg
port: 8080
device: /dev/video2
resolution: 640x480
max_fps: 15
```

1
2
3
4
5
6

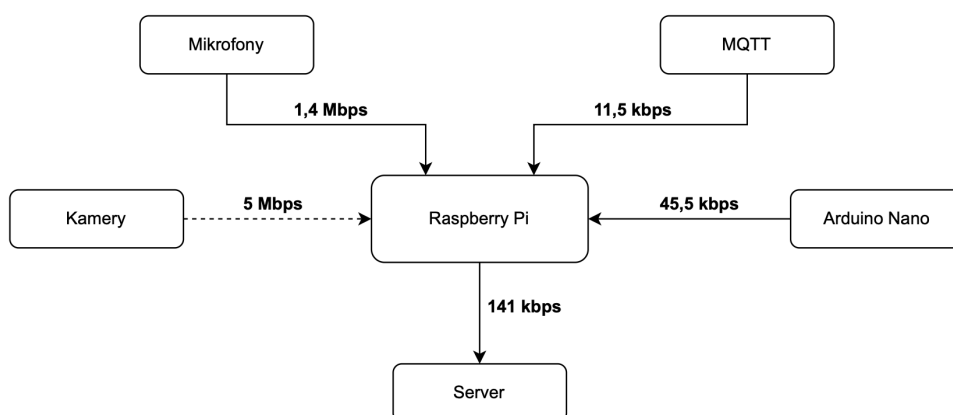
Celkový datový tok z kamer nebyl určen, neboť formát MJPG využívá různých kompresních poměrů v závislosti na obsahu dat. Bit Rate před kompresí by však se znalostí rozlišení videa, počtem FPS a předpokládaným vyjádřením 24 bitů na pixel mohl dosahovat hodnot 110 Mbps.

Datový tok již byl redukován, neboť původně byly kamery nastavené na 30 FPS, avšak z důvodu přehřívání Raspberry Pi byla tato hodnota snížena na 15 FPS. Ačkoli je zhoršená kvalita obrazového přenosu znát, pro potřeby monitorování tiskárny je dostačující.

6.5 Přenos na server

Jak již bylo zmíněno v minulých kapitolách, data se před odesláním na server komprimují. Díky tomu je datový tok mezi Raspberry Pi a serverem značně snížen. Celková velikost souborů před odesláním činí průměrně 216 kB, přičemž po komprimaci mají soubory dohromady velikost 88 kB. Datový tok na server je tedy 141 kbps¹.

Schéma níže přehledněji zobrazuje tok dat mezi zařízeními:²

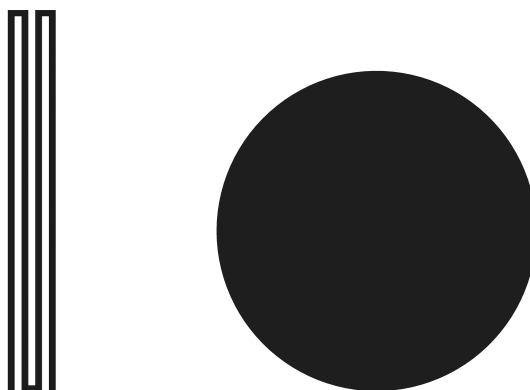


¹Jedná se o průměrnou hodnotu, která může kolísat.

²Datový tok z kamer se nepodílí na celkovém toku dat na server. Proto je značení odlišeno.

7 Otestování systému

Funkčnost celého systému byla otestována pomocí pěti tiskových úloh. Cílem testů bylo měřit data z tiskárny při pohybu v osách X a Y, kruhovém pohybu a pohybu do úhlopříček. Tiskové logy jsou k dispozici v příloze. Následující obrázek zobrazuje modely tiskových testů, přičemž při lineárním pohybu byl využit model vlevo, pouze byl rotován o 45° , 90° a 135° .



Obr. 7.1: Tiskové modely pro test funkčnosti

Naměřená data byla v průběhu měření ukládána do CSV souborů a následně zpracována v programu Matlab. Z důvodu množství a velikosti byly veškeré grafy umístěny do přílohy dokumentu.

Grafy byly sestaveny pro tiskový záznam z tisku kruhového objektu.

7.1 Vyhodnocení dat z externích senzorů

Na grafu zobrazujícím hodnoty akcelerometrů D.1 lze během tisku pozorovat periodické zvýšení vibrací. Tento jev je pravděpodobně způsoben pohybem tiskárny v ose Z. Kromě toho je v grafu také možno vidět aditivní chybu ovlivňující nulovou hodnotu snímačů. Osa X akcelerometru směřuje kolmo k zemi a je tak kromě požadovaných vibrací měřeno i gravitační zrychlení.

Graf změřených teplot motorů D.2 vykazuje nárůst teplot během tisku, což odpovídá skutečnosti zahřívání motorů při běhu. Motor výložného systému nebyl při tisku aktivní, jeho hodnota byla tedy neměnná. To odpovídá černému průběhu v grafu. Ačkoli byl předpoklad, že všechny čtyři motory v ose Z budou mít při tisku stejnou teplotu, jejich teploty jsou rozdílné. Důvodem je povolení držáku pro senzor teploty, který se vlivem ohřevu motoru roztáhl a senzor již přesně nedoléhá na motor.

Grafy okolních teplot D.3 a vlhkostí D.4 vykazují předpokládané chování veličin v průběhu tisku. Na vlhkostním grafu lze pozorovat odlehlou prvotní hodnotu naměřenou senzorem DHT-22. Při pohledu na následující hodnoty se však spíše jedná o chybu senzoru, než o skutečně naměřená data.

Změřené koncentrace plynů jsou vyneseny v grafu D.5. Koncentrace oxidu uhelnatého je za normálních podmínek značně nižší než koncentrace oxidu uhličitého.

Audio záznam včetně zvukového spektra z jednoho testovacího tisku je zobrazen v grafech F.1 a F.2. V časové oblasti lze opět pozorovat periodické kmity, stejně jako v grafu vibrací. Grafy byly totiž pro porovnání sestaveny ze stejného tiskového záznamu, a tudíž bylo zachyceno stejné chování tiskárny pomocí odlišných senzorů.

V tabulce níže jsou zobrazeny základní statistické údaje ze snímačů. Hodnoty byly vypočteny ze všech pěti tiskových záznamů. Největší naměřený výkyv je u senzoru oxidu uhelnatého, u kterého byla změřena maximální koncentrace 164,2 ppm. Tato hodnota by však znamenala, že v daný moment byla koncentrace oxidu uhelnatého vyšší než koncentrace oxidu uhličitého, což není s ohledem na možné výpary z filamentů reálné. Důvodem naměření této odlehlé hodnoty by mohlo být příliš časté čtení dat ze snímače, který se následně nestíhal stabilizovat.

Tab. 7.1: Statistická data ze senzorů

Veličina	Minimum	Maximum	Průměrná hodnota
Zrychlení X_1 [$m \cdot s^{-2}$]	7,372	13,641	10,347
Zrychlení Y_1 [$m \cdot s^{-2}$]	-1,378	1,010	0,182
Zrychlení Z_1 [$m \cdot s^{-2}$]	-3,070	1,158	-0,934
Zrychlení X_2 [$m \cdot s^{-2}$]	9,562	10,861	10,347
Zrychlení Y_2 [$m \cdot s^{-2}$]	-1,565	1,606	-0,129
Zrychlení Z_2 [$m \cdot s^{-2}$]	-2,175	1,275	-0,581
Teplota motorů [°C]	23,25	41,75	31,79
Okolní teplota [°C]	22,7	24,7	23,7
Okolní vlhkost [%]	36,5	41,0	39,3
Koncentrace CO_2 [ppm]	132,5	143,0	137,1
Koncentrace CO [ppm]	0,3	164,2	11,6

7.2 Vyhodnocení dat z tiskového firmwaru

Grafy teplot E.1, E.2, E.3 a E.4 zobrazují vývoj teplot v jednotlivých částech tiskárny. Při tisku byl hotend nastaven na hodnotu 210 °C. Počáteční zvýšená hodnota je způsobena překmitem akčního zásahu PID regulátoru, který je zodpovědný za řízení teploty hotendu. Teplotní překmit je obvykle okolo 5 °C, tiskový firmware však čeká na klesnutí teploty pod danou mez pro zahájení tisku. Z grafu E.3 lze vyvodit, že tato mez je přibližně 3 °C.

Z důvodu velké plochy tiskové podložky je její vyhřívání realizováno topným tělesem určeným na 230 V_{AC}. Řízení vyhřívání je tudíž realizováno pouze pomocí SSR¹ s hysterezí (viz graf E.4).

Grafy E.5 a E.6 zobrazují spotřebu a průměr filamentu během tisku. Vzhledem k jednoduchému tvaru modelů je závislost spotřeby filamentu v čase lineární. Počáteční menší sklon závislosti je způsoben nižší rychlostí tiskové hlavy při tisku první vrstvy. Graf průměru filamentu překvapivě není konstantní, ale jeho hodnota v průběhu tisku neustále kolísala. Nelze však přesně říci, zda byla chyba v uspořádání senzoru nebo filament opravdu obsahoval známky ovality.

Aby byla dále ukázána funkčnost systému pro sběr dat z firmwaru Klipper, sestavily se zpětně po tisku z naměřených polohových dat trajektorie tiskové hlavy. Ty jsou v grafech E.7. Všechny grafy obsahují posun tiskové hlavy na pozici [0; 200]. Na tyto souřadnice se tisková hlava pokaždé přemístila, jakmile dotiskla daný model. Nejedná se tedy o chybná data.

Obdobně jako v předchozím případě byla vypočtena statistická data z vybraných tiskových veličin. Vypočtené hodnoty jsou dle očekávání a nevykazují známky poruchy ve firmwaru.

Tab. 7.2: Statistická data z Klipperu

Veličina	Minimum	Maximum	Průměrná hodnota
Teplota hotendu [°C]	208,89	213,38	210,15
Výkon hotendu [-]	0	0,47	0,33
Teplota podložky [°C]	57,32	64,39	60,92
Teplota Raspberry [°C]	51,54	58,00	55,46
Teplota Octopus [°C]	32,15	39,04	36,46
Průměr filamentu [mm]	1,71	1,87	1,79

¹Solid State Relay

7.3 Možná vylepšení

Ačkoliv je systém po hardwarové i softwarové stránce funkční, stále by mohly být některé věci vylepšeny. Jedná se především o nevhodné použití plastových dílů v konstrukci tiskárny. Napínání řemenů je realizováno plastovými díly, které se však při působení síly napnutého řemenu deformují a při dlouhodobějším používání tiskárny by mohlo dojít k jejich zlomení. Během práce byla snaha o předělání těchto dílů na hliníkové, avšak z důvodu vysoké ceny za výrobu dílů nebyla tato část nakonec uskutečněna.

Dalším problémem jsou plastové díly, které přitlačí teplotní senzory DS18B20 na kryty krokových motorů. Tyto díly byly vyrobeny z PET-G filamentu, který není určen pro příliš velkou teplotní zátěž a již po pár tiscích lze pozorovat povolení držáků a teplotní čidla tak nedoléhají přímo na povrch krytu. Lepší variantou by do budoucna bylo použití ASA materiálu s teplotní odolností do 96 °C.² Druhou možností by mohlo být použití PA filamentu, který se díky své teplotní odolnosti řadí mezi nejlepší filamentey pro tyto aplikace.³ Tento filament je však pro běžné používání značně drahý a použití ASA materiálu by tak bylo dostačující.

Senzory plynů byly primárně vybrány dle dostupnosti knihoven pro jejich programování a také s ohledem na pořizovací cenu. Vzhledem k malému množství výparů, které při tisku vzniká by však pro věrné vyjádření koncentrace bylo nutné využít mnohem přesnější, citlivější, a tudíž značně dražší snímače. Instalované senzory tak spíše slouží k doplnění měřicího řetězce.

²Platí pro ASA filament od značky Devil Design [36]. Ostatní filamentey z daného materiálu mají teplotní odolnost podobnou.

³Filament Spektrum PA6 CF15 se deformuje při teplotě 200 °C [35].

Závěr

Cílem bakalářské práce bylo navrhnout a realizovat systém pro sběr dat z 3D tiskárny včetně jejich zpracování.

První kapitola se věnuje stávajícímu řešení tiskárny. Jsou zde popsány základní informace o tiskárně, použitý hardware a software. Zároveň je zde navrženo použití jiného prostředí pro vzdálenou obsluhu tiskárny.

Sběr dat byl rozdělen na dvě hlavní části - data z externích senzorů a data z tiskového firmwaru.

V druhé kapitole jsou tak popsány veličiny, které je vhodné ve spojitosti s 3D tiskem měřit s následným výběrem senzorů pro daná měření. V realizační části je poté navrženo rozmístění senzorů, zobrazeno jejich zapojení do použitého mikro-počítače Arduino Nano a také popsána struktura programu. Obdobně je popsáno zapojení mikrofónů do Raspberry Pi, které muselo být pro tyto účely využito z důvodu chybějící zvukové sběrnice I^2S na desce Arduino.

Další část se zaměřuje na tiskový firmware Klipper a jakým způsobem lze z něj shromažďovat data spojená se samotnou tiskárnou. Následně je navrhnout a realizován způsob komunikace přes MQTT protokol, který zahrnoval nastavení MQTT brokera a konfiguraci protokolu na straně tiskárny. Posledním bodem v této kapitole bylo zprovoznění senzoru ovality filamentu a jeho nastavení v hlavním konfiguračním souboru tiskárny. Bohužel nebylo možné realizovat odesílání hodnot proudů motorů z důvodu chybějícího příkazu pro posílání těchto dat.

Požadavky byly také kladeny na způsob zpracování dat, kterému je věnována kapitola 5. V první podkapitole je popsán základní koncept, jakým budou data ze tří různých zdrojů zpracovávána. Byla zvolena hierarchická struktura skládající se z jednoho hlavního programu a tří podprogramů. Hlavní program má za úkol číst data přicházející z Mosquitto brokera a spustit dané podprogramy ve chvíli, jakmile přijde informace o spuštění, případně ukončení tisku. Díky tomuto řešení se data zpracovávají pouze při spuštěném tisku, a tudíž lze v budoucnu využít algoritmy strojového učení na jejich vyhodnocení. Kapitola se dále zabývá komprimací dat s následným odesláním do cloudové služby. Vzhledem k tomu, že příjem dat ze strany serveru zatím není spuštěný, byla pro testování přenosu využita jednoduchá služba pro zpracování HTTP požadavků. Schéma celého systému lze nalézt v příloze A.

Obě realizované části pro měření dat byly následně otestovány na zkušebních tiscích. Výsledky včetně možných vylepšení jsou komentovány v kapitole 7.

Hlavním přínosem této práce pro mě bylo porozumění tiskovému firmwaru Klipper včetně jeho konfigurace. Dále jsem se naučil použít pokročilé knihovny jazyka Python k realizaci daného úkolu. Díky této práci je systém do budoucna připraven pro aplikaci pokročilého zpracování dat.

Literatura

- [1] ČTVRTNÍČEK, Jan. *Revitalizace 3D tiskárny*. Online, Bakalářská práce, vedoucí Ing. Jakub Arm Ph.D. Brno: Vysoké učení technické v Brně, 2023. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=254958. [cit. 2023-11-06].
- [2] FIALA, Jan. *Sběr provozních dat z 3D tiskárny*. Online, Bakalářská práce, vedoucí Ing. Jakub Arm, Ph.D. Brno: Vysoké učení technické v Brně, 2022. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=241428. [cit. 2023-11-06].
- [3] MOYER, Ilan. *CoreXY Principle of Operation*. Online. In: Core(X,Y). 2012. Dostupné z: <https://corexy.com/reference.png>. [cit. 2023-11-09].
- [4] KILLIOGLU, Alex. *Direct Drive vs Bowden Extruder: Ultimate Comparison Guide*. Online. In: How to 3D print net. c2024. Dostupné z: <https://howto3dprint.net/direct-drive-vs-bowden-extruder-comparison-guide/>. [cit. 2023-11-09].
- [5] *BIGTREETECH Octopus V1.1 Control Board For Voron 2.4*. Online. In: Biqu Equipment. [cca 2015]. Dostupné z: https://biqu.equipment/cdn/shop/products/A_2230f45e-703c-4a43-a60f-4b74757ce32d_1300x.jpg?v=1700265641. [cit. 2023-11-10].
- [6] *BIGTREETECH TMC2209 V1.3 Stepper Motor Driver*. Online. In: Biqu Equipment. [cca 2015]. Dostupné z: <https://biqu.equipment/collections/driver-board/products/bigtreetech-tmc2209-stepper-motor-driver-for-3d-printer-board-vs-tmc2208>. [cit. 2023-11-10].
- [7] *Mainsail*. Online. In: Mainsail. [cca 2021]. Dostupné z: <https://2822477169-files.gitbook.io/~files/v0/b/gitbook-x-prod.appspot.com/o/spaces%2FhMRQnt097x6GxmrCYaFy%2Fuploads%2FzlcYc0ZsuMmQ3sYWVEly%2Fscreen%2Fshot.png?alt=media&token=948f5430-1c26-41fa-a871-48b4f1734fd9>. [cit. 2023-11-13].
- [8] *Fluidd*. Online. In: Fluidd. [cca 2019]. Dostupné z: https://docs.fluidd.xyz/assets/images/preview_sliced.png. [cit. 2023-11-13].

- [9] *Senzor teploty a vlhkosti vzduchu SHT31.* Online. In: Láskakit. C2024. Dostupné z: <https://www.laskakit.cz/senzor-teploty-a-vlhkosti-vzduchu-sht31/>. [cit. 2023-11-19].
- [10] *Digitální teplotní senzor DALLAS - DS18B20.* Online. In: Drátek. C. Dostupné z: <https://dratek.cz/arduino/1187-teplotni-senzor-digitalni-dallas-ds18b20.html>. [cit. 2023-11-19].
- [11] *DS18B20 Datasheet (PDF) - Dallas Semiconductor.* Online. In: All Datasheet. C. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/227472/DALLAS/DS18B20.html>. [cit. 2023-11-19].
- [12] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer.* Online. In: Analog. C1995-2023. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf>. [cit. 2023-11-19].
- [13] *Senzor plynu MQ135 MQ - 135 pro Arduino.* Online. In: Drátek. C. Dostupné z: <https://dratek.cz/arduino/1298-senzor-plynu-mq135-mq-135-pro-arduino.html>. [cit. 2023-11-20].
- [14] *Senzor oxidu uhelnatého MQ-7.* Online. In: Láskakit. C2024. Dostupné z: <https://www.laskakit.cz/arduino-senzor-oxidu-uhelnateho-mq-7/>. [cit. 2023-11-20].
- [15] *Weighing Load Cell Sensor 3KG For Electronic Kitchen Scale YZC-131 With Wires.* Online. In: Robo Tools. C2024. Dostupné z: <https://robotools.in/wp-content/uploads/2022/10/DIYmall-2kg-font-b-Load-b-font-font-b-Cell-b-font-Digital-Portable-Electronic-for.jpg>. [cit. 2023-11-20].
- [16] *HX711 - zesilovač pro tenzometrický paprsek.* Online. In: Botland. C2024. Dostupné z: <https://botland.cz/senzory-tlaku/12479-hx711-zesilovac-pro-tenzometricky-paprsek-5904422319366.html>. [cit. 2023-11-20].
- [17] *Spoolman.* Online. In: Github. C2024. Dostupné z: <https://github.com/Donkie/Spoolman>. [cit. 2023-11-27].
- [18] *Jak poznat navlhlý filament a co s tím?* Online. In: Material pro 3D. C2024. Dostupné z: <https://www.materialpro3d.cz/blog/jak-poznat-navlhly-filament-a-co-s-tim/>. [cit. 2023-11-30].

- [19] *SPH0645LM4H - MEMS I2S mikrofon - Adafruit 3421*. Online. In: Botland. C2024. Dostupné z: <https://botland.cz/mikrofony-a-detektory-zvuku/9715-sph0645lm4h-mems-i2s-mikrofon-adafruit-3421-5904422307103.html>. [cit. 2023-12-03].
- [20] *Arduino Nano Pinout / A Complete Guide*. Online. In: Linuxhint. -. Dostupné z: <https://linuxhint.com/wp-content/uploads/2022/05/image3-64.png>. [cit. 2023-12-03].
- [21] *Raspberry Pi hardware*. Online. In: Raspberry Pi. C2012-2023. Dostupné z: <https://www.raspberrypi.com/documentation/computers/images/GPIO-Pinout-Diagram-2.png>. [cit. 2023-12-03].
- [22] *Raspberry Pi Hardware*. Online. In: Studio Pieters. C2024. Dostupné z: <https://www.studiopieters.nl/wp-content/uploads/2021/11/Raspberry-pin-out-Power.png>. [cit. 2023-12-03].
- [23] *Hall filament width sensor*. Online. In: Klipper3D. 2016. Dostupné z: https://www.klipper3d.org/Hall_Filament_Width_Sensor.html. [cit. 2023-12-17].
- [24] *RepRapFirmware overview*. Online. [cca 2013]. Dostupné z: https://docs.duet3d.com/User_manual/RepRapFirmware/RepRapFirmware_overview. [cit. 2022-12-28].
- [25] *Introducing JSON*. Online. 2001. Dostupné z: <https://www.json.org/json-en.html>. [cit. 2023-12-28].
- [26] *Moonraker [mqtt]*. Online. In: Moonraker. C2024. Dostupné z: <https://moonraker.readthedocs.io/en/latest/configuration/#mqtt>. [cit. 2023-12-29].
- [27] *Authentication methods*. Online. In: Eclipse Mosquitto. C. Dostupné z: <https://mosquitto.org/documentation/authentication-methods/>. [cit. 2023-12-30].
- [28] *Jsonpath-ng*. Online. In: Pypi. Dostupné z: <https://pypi.org/project/jsonpath-ng/>. [cit. 2023-12-31].
- [29] *Medium - How to Set Up a Mosquitto MQTT Broker Securely - Using Client Certificates*. Online. 2012. Dostupné z: <https://medium.com>. [cit. 2024-03-15].
- [30] *Crownsnest*. Online. Mainsail. [cca 2021]. Dostupné z: <https://crownsnest.mainsail.xyz/configuration/cam-section>. [cit. 2024-03-26].

- [31] VACULÍK, Samuel. *IoT systém pro zpracování dat z 3D tiskárny*. Online, Semestrální práce, vedoucí Ing. Jakub Arm, Ph. D. Brno: Vysoké učení technické v Brně, 2024. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor.php?file_id=262045&zp_id=155205. [cit. 2024-04-03].
- [32] *3D Printer Fumes: Everything You Need to Know*. Online. SelfCAD. C2020. Dostupné z: <https://www.selfcad.com/blog/3d-printer-fumes-what-you-need-to-know>. [cit. 2024-04-11].
- [33] *A simple HTTP Request & Response Service*. Online. [cca 2014]. Dostupné z: <https://httpbin.org>. [cit. 2024-05-01].
- [34] *Compression File Formats*. Online. In: FILEFORMAT. C2024. Dostupné z: <https://docs.fileformat.com/compression/gz/>. [cit. 2024-05-01].
- [35] *TECHNICAL DATA SHEET PA6 CF15*. Online. In: Spektrum FILAMENTS. C2024. Dostupné z: https://spectrumfilaments.com/wp-content/uploads/2022/06/pl_tds_spectrum_pa6_cf15.pdf. [cit. 2024-05-02].
- [36] *ASA Filament*. Online. Devil Design. [cca 2018]. Dostupné z: https://devildesign.com/download/ASA_-_karta_produkту.pdf. [cit. 2024-05-01].

Seznam symbolů a zkratek

ABS	Acrylonitrile butadiene styrene
API	Application Programming Interface
ASA	Acrylonitrile styrene acrylic
BLCK	Bit Clock Line
CPU	Central Processing Unit
CSV	Comma Separated Values
DOUT	Data Out
FPS	Frames Per Second
HTTP	Hypertext Transfer Protocol
<i>I²C</i>	Inter-Integrated Circuit
<i>I²S</i>	Inter-IC Sound
IoT	Internet of Things
JSON	JavaScript Object Notation
LRCL	Left Right Clock
MMU	Multi Material Unit
MQTT	Message Queuing Telemetry Transport
PA	Polyamide
PET-G	Glycol-modified Polyethylene Terephthalate
PID	Proportional Integrative Derivative
PLA	Polylactic acid
PPM	Parts Per Million
PSR	Power Supply Rejection
PVA	Polyvinylalcohol
RAM	Random Access Memory

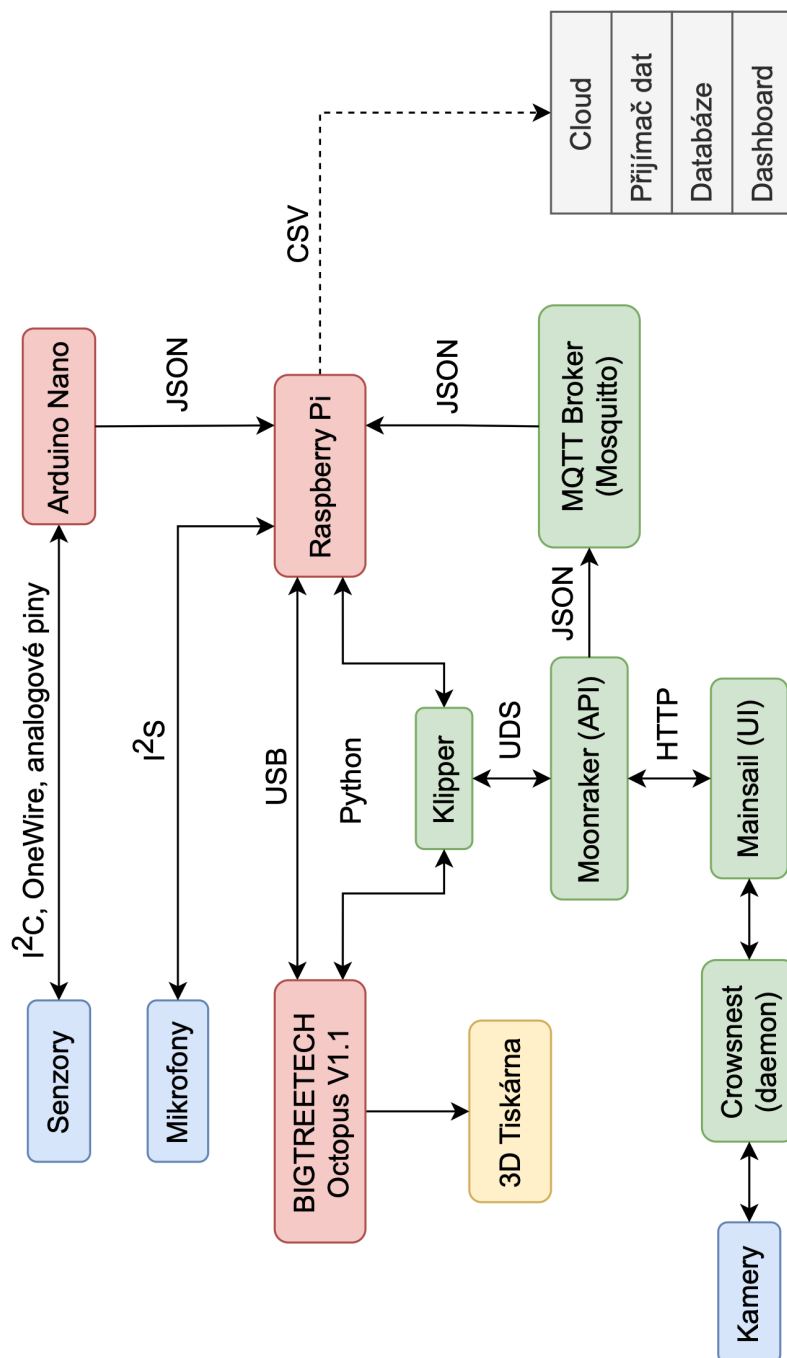
RRP	Rep Rap Firmware
SSR	Solid State Relay
SNR	Signal to Noise Ratio
SPL	Sound Pressure Leve
THD	Total Harmonic Distortion
UDS	Unix Domain Socket
USB	Universal Serial Bus
WAV	Waveform Audio File Format

Seznam příloh

A	Struktura celého systému	73
B	Uživatelská prostředí	75
B.1	Mainsail	75
B.2	Fluidd	75
C	Bloková schémata jednotlivých programů	77
C.1	Arduino program pro čtení dat ze senzorů	77
C.2	Hlavní program pro správu podprogramů	78
C.3	Program pro zpracování dat z Klipperu	79
C.4	Program pro zpracování dat z Arduina	80
C.5	Program pro zpracování dat z mikrofónů	81
D	Naměřená data ze senzorů v průběhu tisku	83
D.1	Vibrace	83
D.2	Teploty krokových motorů	84
D.3	Okolní teploty	85
D.4	Okolní vlhkosti	86
D.5	Koncentrace plynů	87
E	Naměřená data z tiskového firmwaru v průběhu tisku	89
E.1	Teplota Raspberry Pi	89
E.2	Teplota desky Octopus	90
E.3	Teplota hotendu	91
E.4	Teplota tiskové podložky	92
E.5	Spotřeba filamentu	93
E.6	Průměr filamentu	94
E.7	Trajektorie tiskové hlavy	95
F	Naměřená zvuková data při tisku	97
F.1	Časová oblast	97
F.2	Frekvenční oblast	98
G	Konfigurace Klipperu	99
G.1	Konfigurace MQTT v souboru moonraker.conf	99
G.2	Konfigurace senzoru průměru filamentu v souboru printer.cfg	101
G.3	Konfigurace kamer v souboru crowsnest.conf	102
H	Konfigurace Mosquitto brokera v souboru mosquitto.conf	103

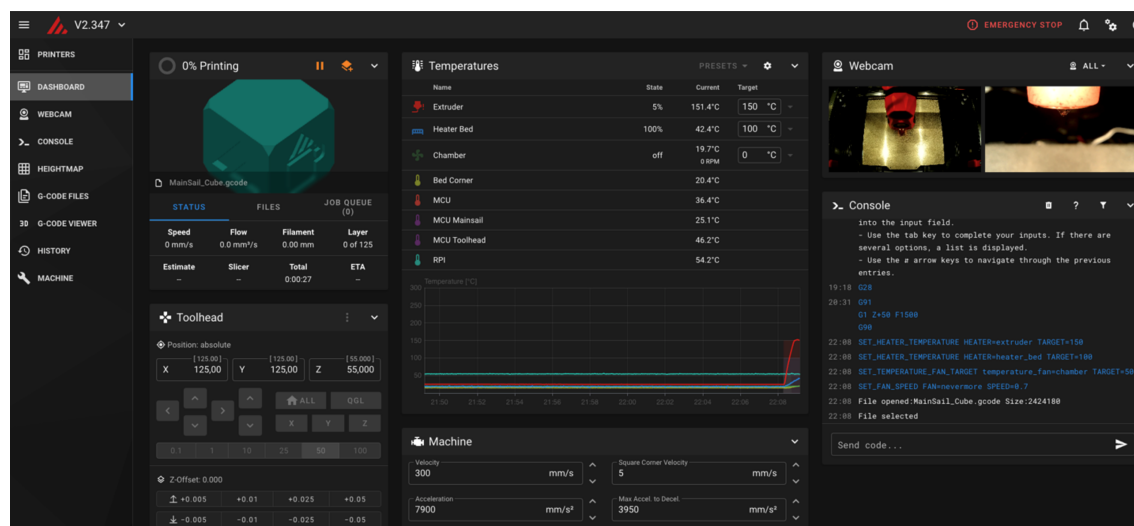
I	Fotodokumentace	105
I.1	Umístění první kamery	105
I.2	Umístění druhé kamery	106
I.3	Senzory v horní části tiskárny	107
I.4	Držák teplotního čidla na krytu motoru	108
I.5	Senzor průměru filamentu	109
J	Obsah elektronické přílohy	111

A Struktura celého systému



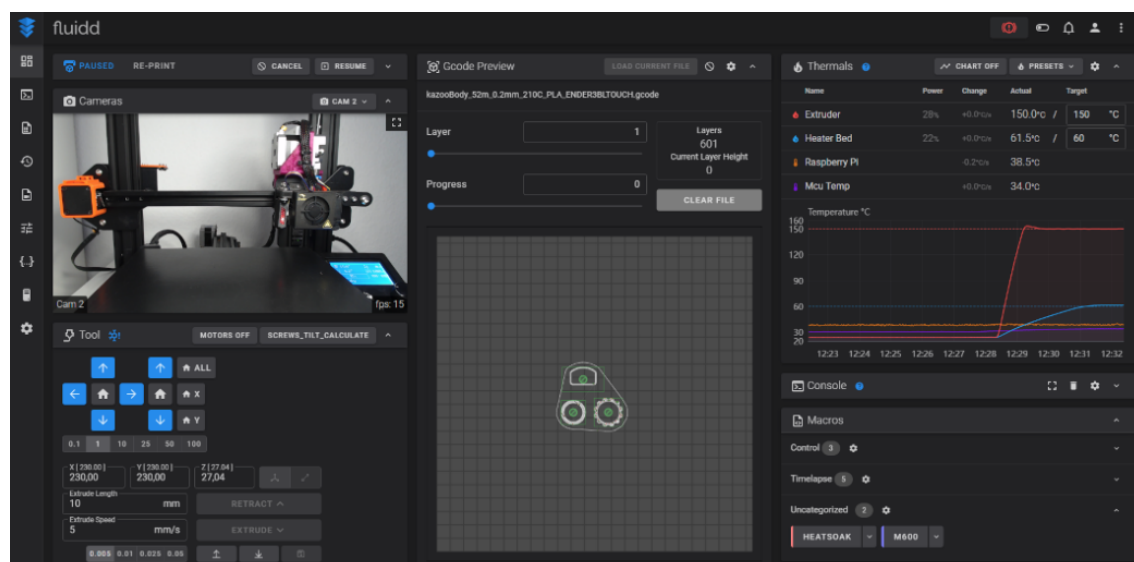
B Uživatelská prostředí

B.1 Mainsail



Obr. B.1: Prostředí Mainsail [7]

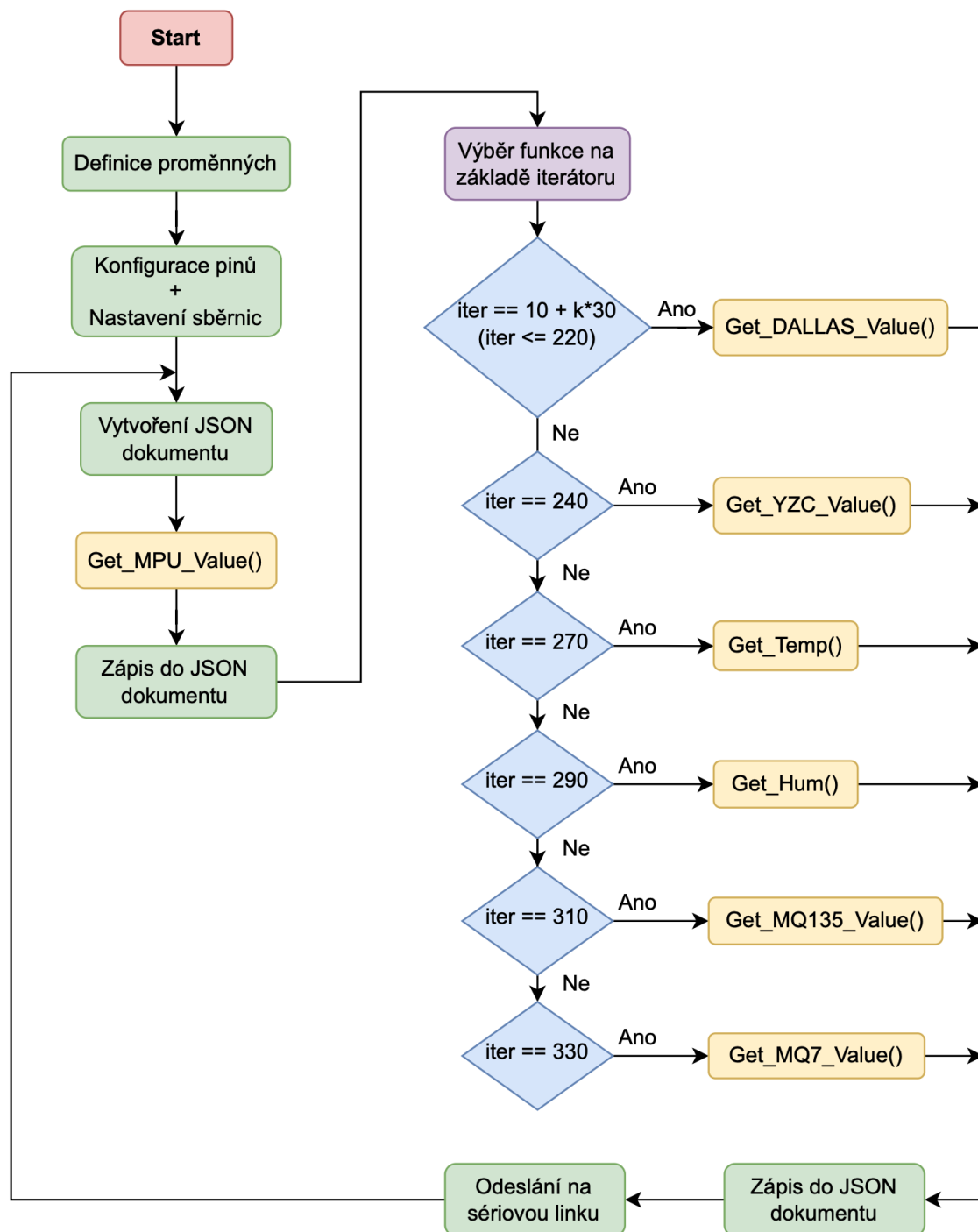
B.2 Fluidt



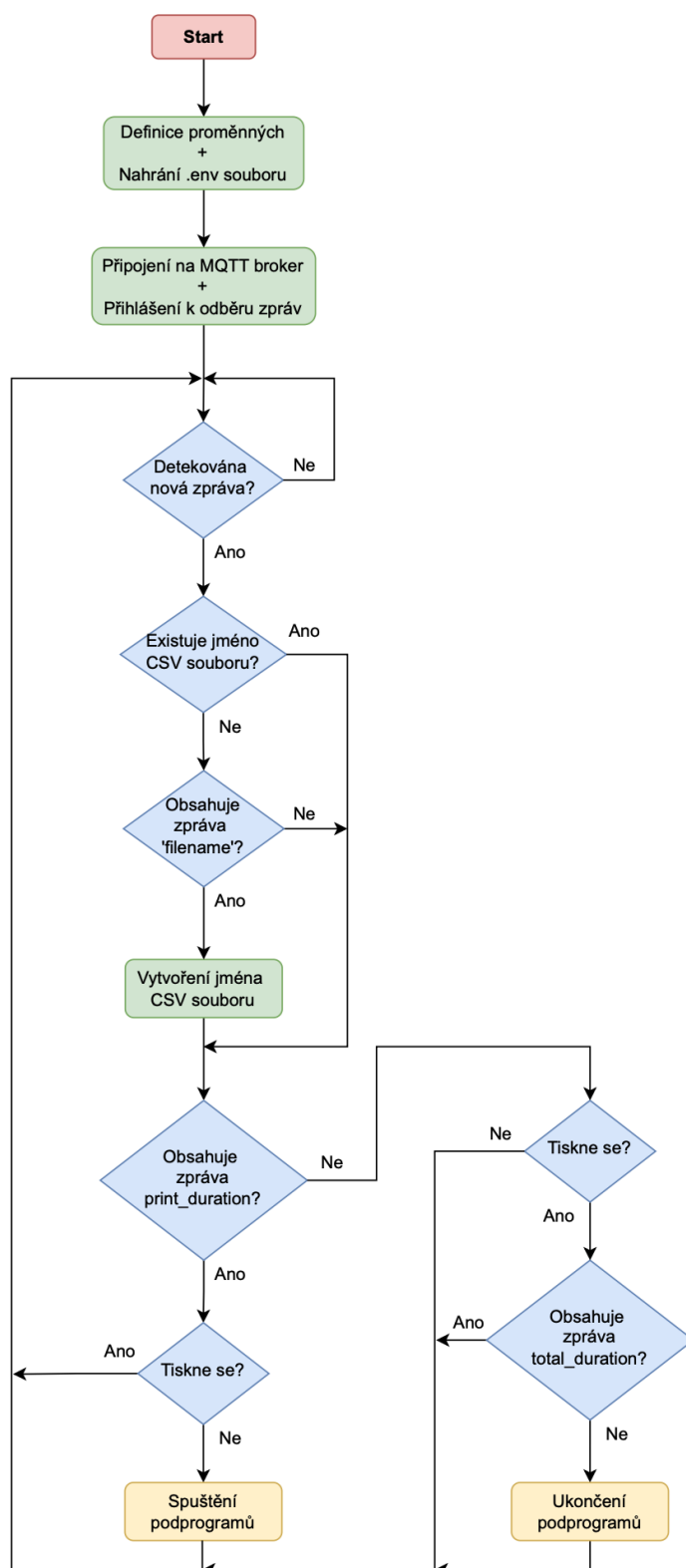
Obr. B.2: Prostředí Fluidt [8]

C Bloková schémata jednotlivých programů

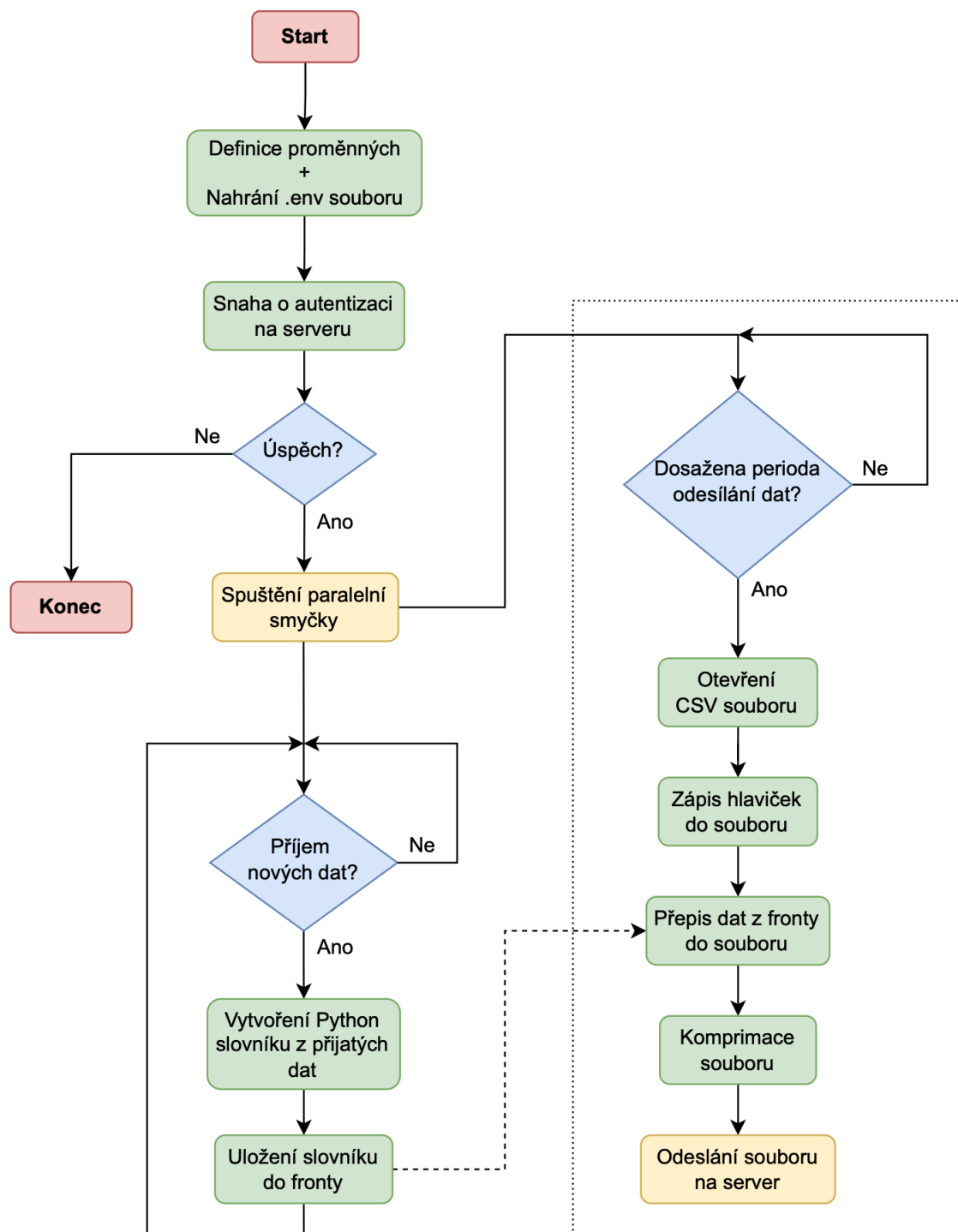
C.1 Arduino program pro čtení dat ze senzorů



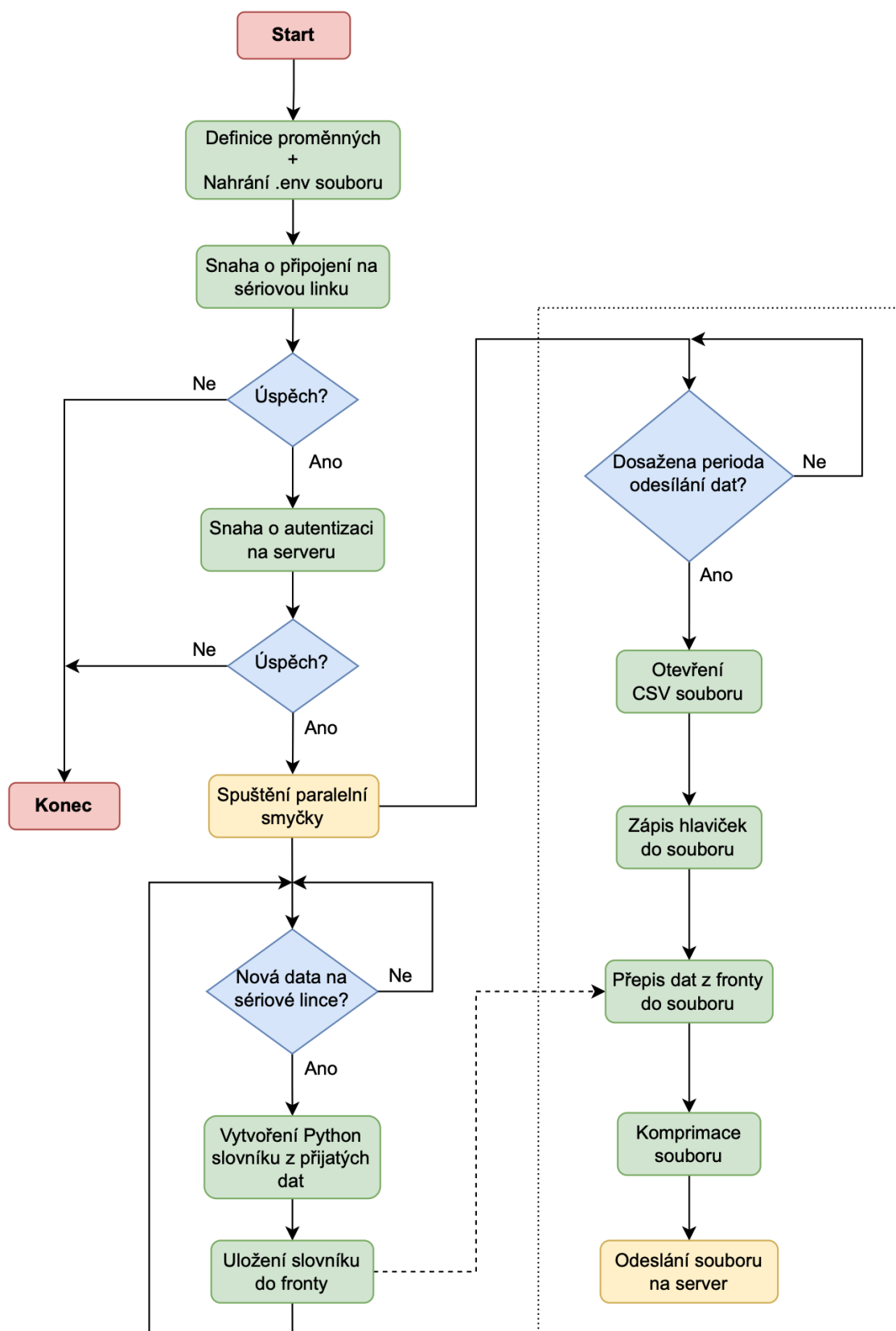
C.2 Hlavní program pro správu podprogramů



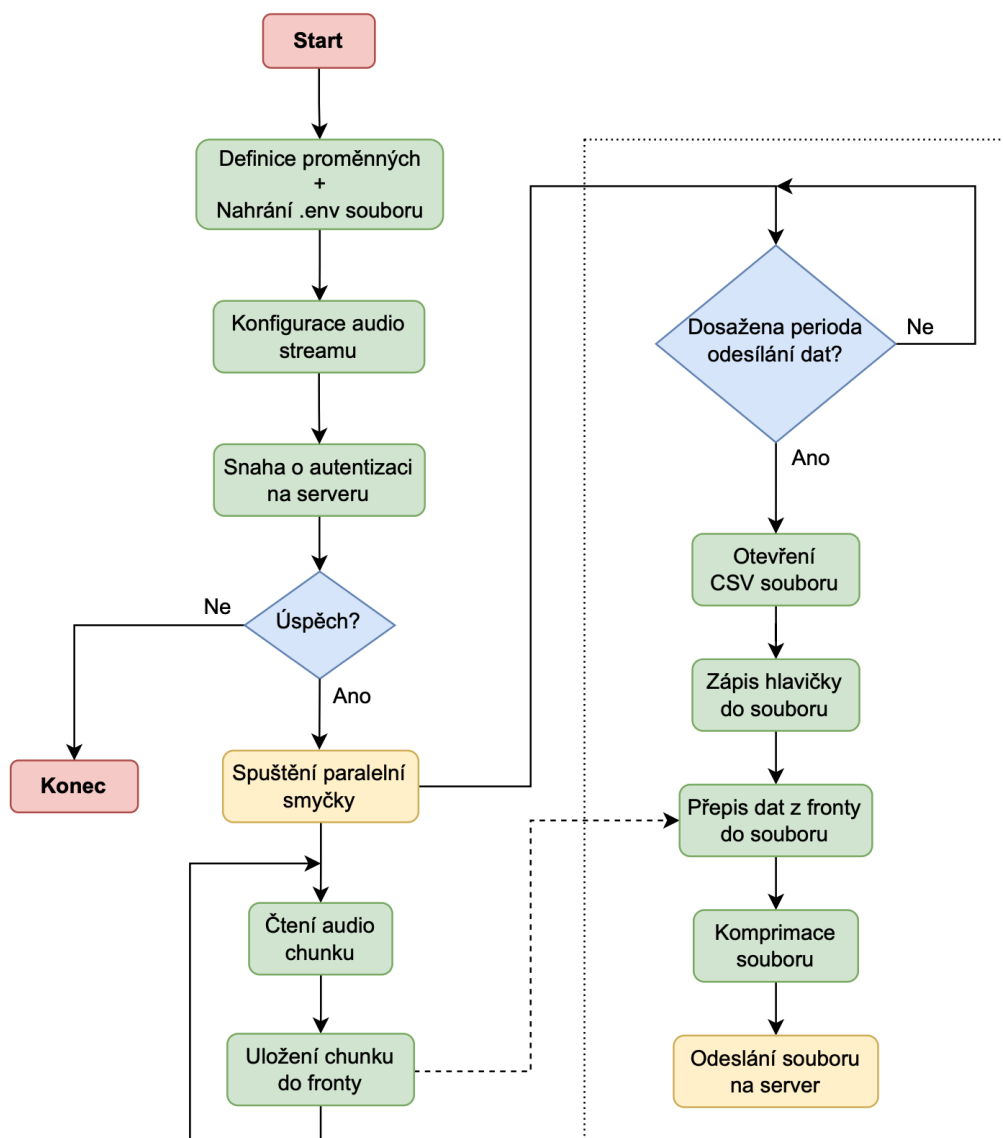
C.3 Program pro zpracování dat z Klipperu



C.4 Program pro zpracování dat z Arduina

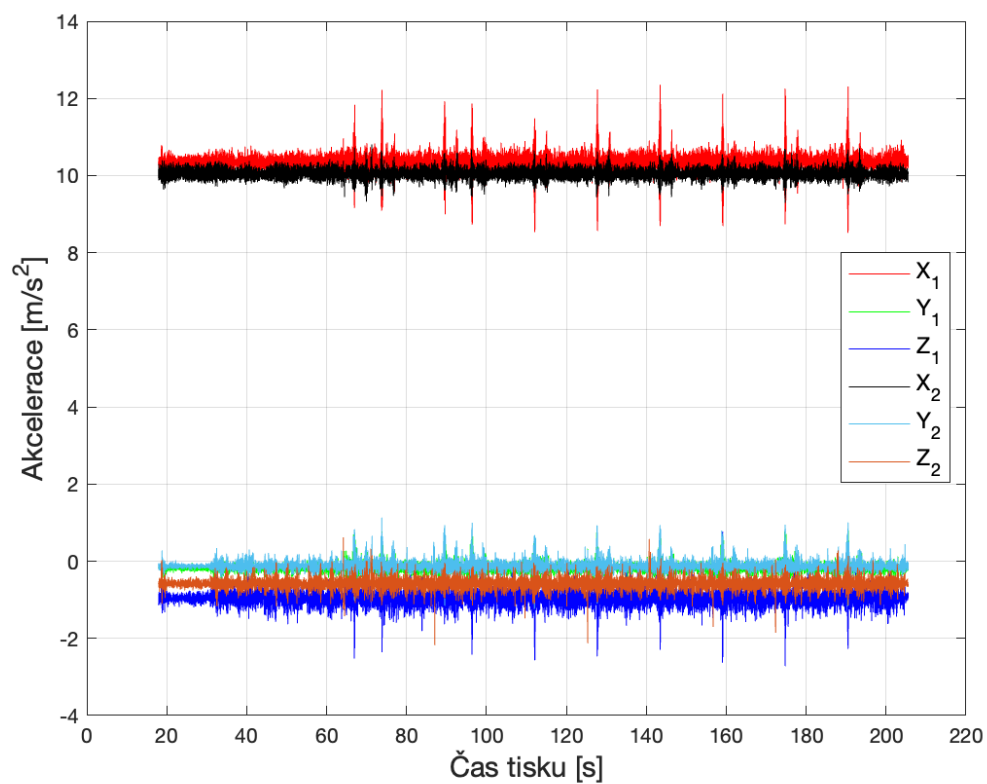


C.5 Program pro zpracování dat z mikrofónů

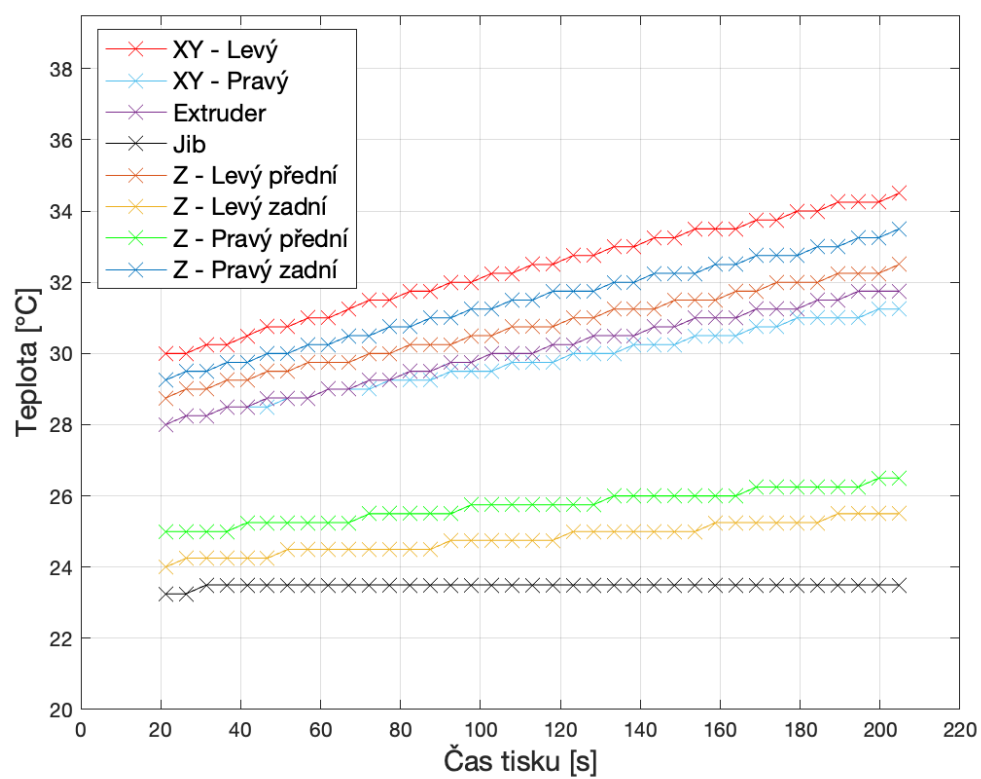


D Naměřená data ze senzorů v průběhu tisku

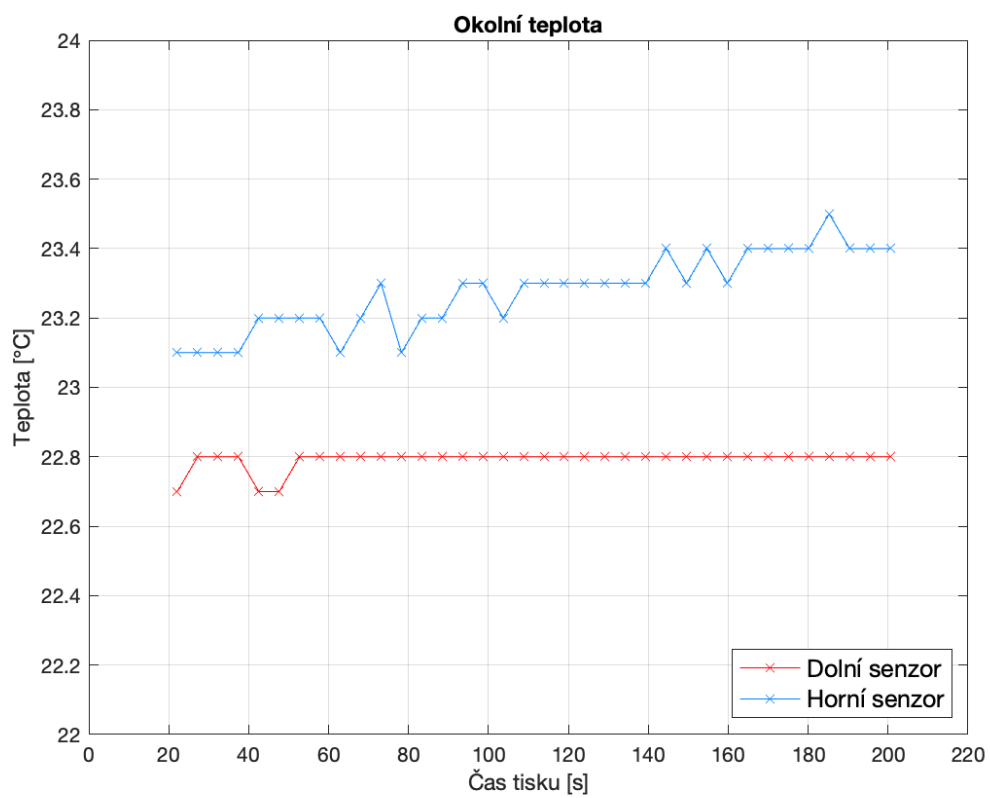
D.1 Vibrace



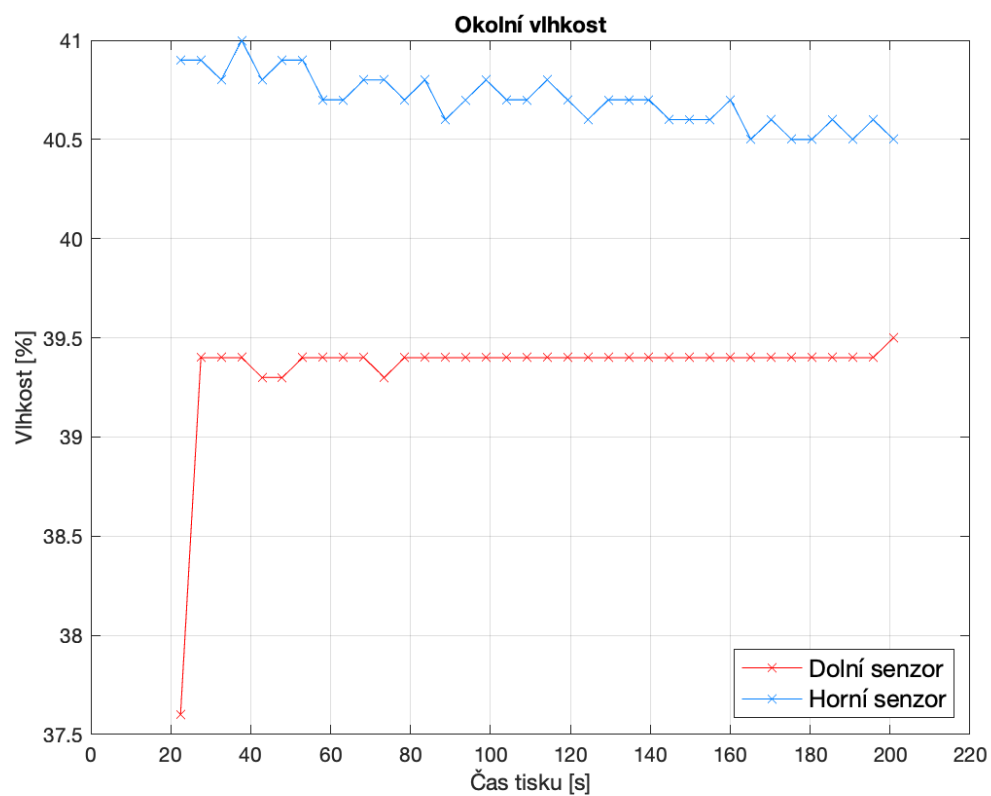
D.2 Teploty krokových motorů



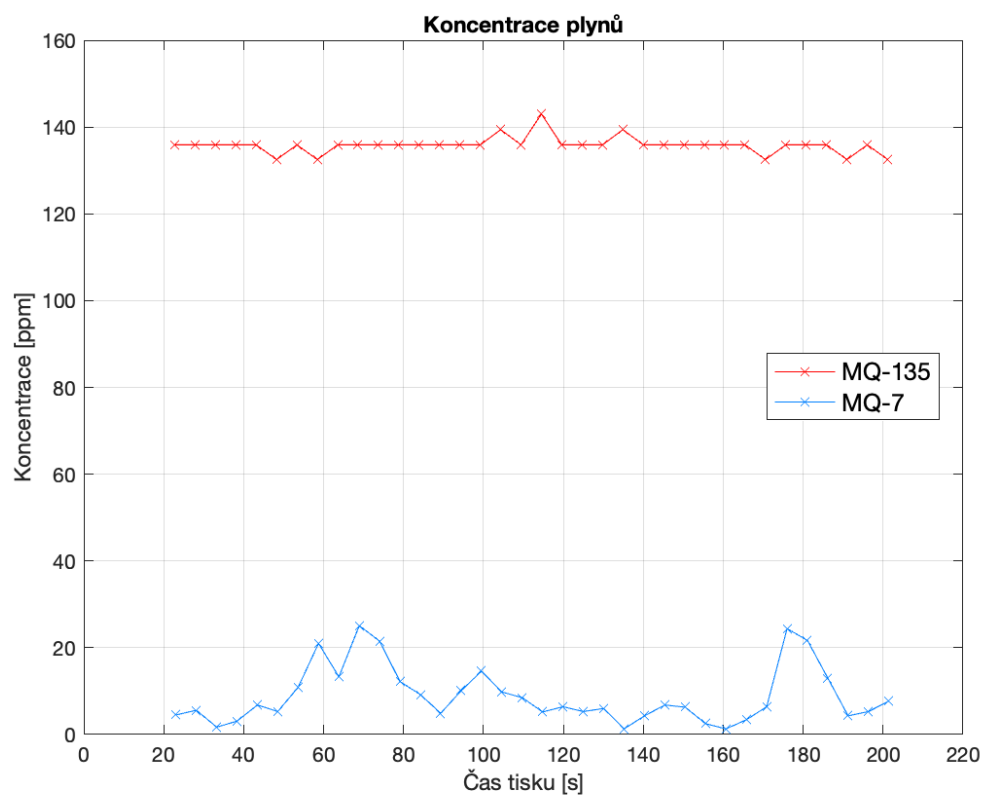
D.3 Okolní teploty



D.4 Okolní vlhkosti



D.5 Koncentrace plynů

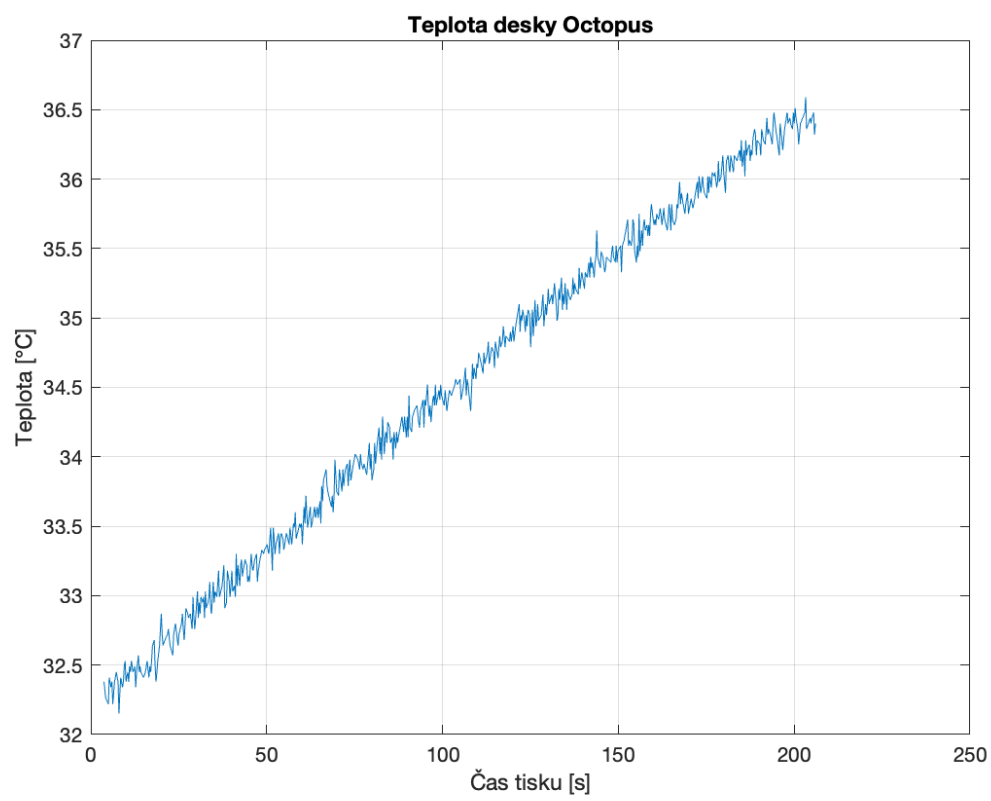


E Naměřená data z tiskového firmwaru v průběhu tisku

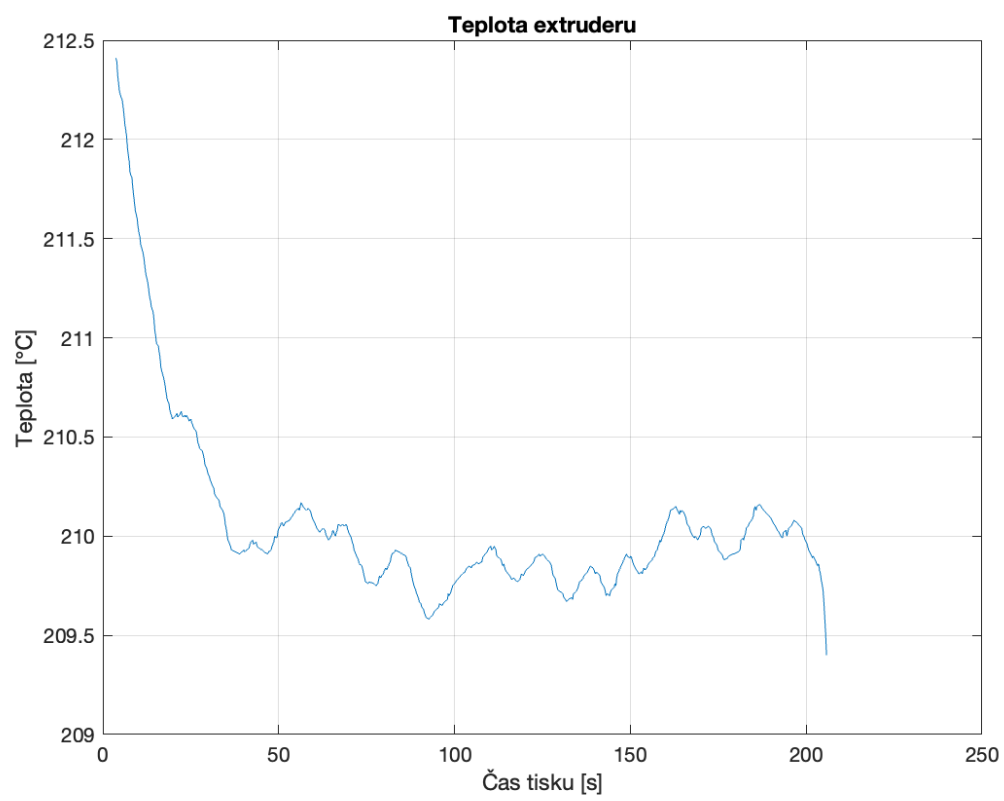
E.1 Teplota Raspberry Pi



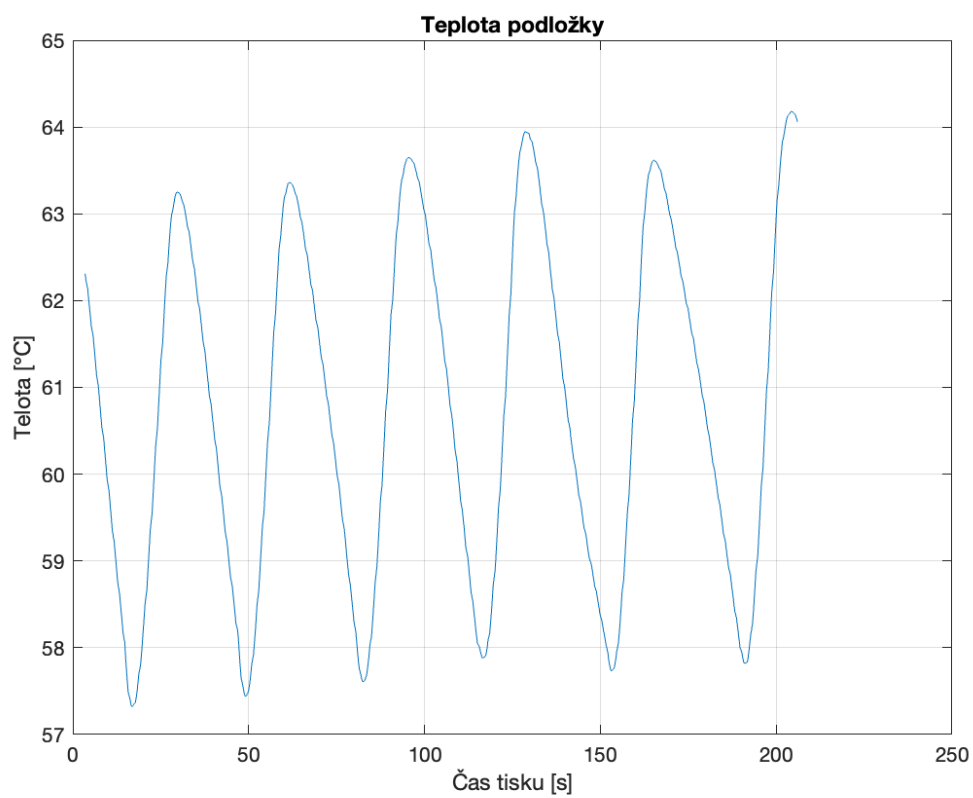
E.2 Teplota desky Octopus



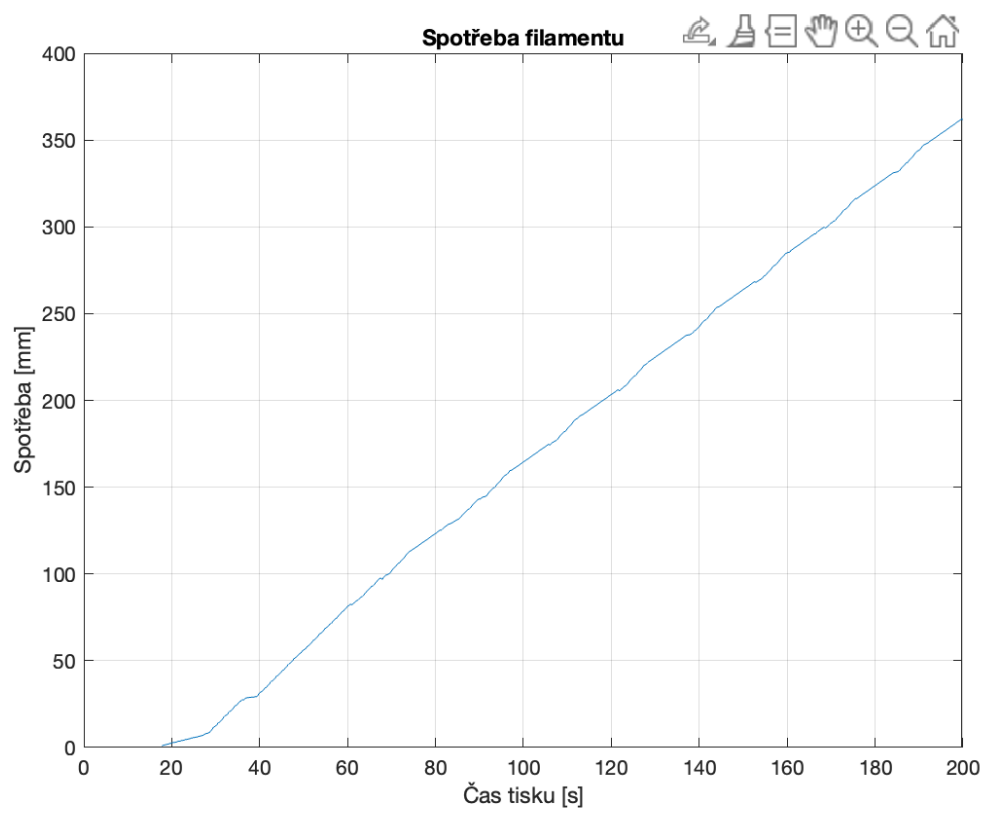
E.3 Teplota hotendu



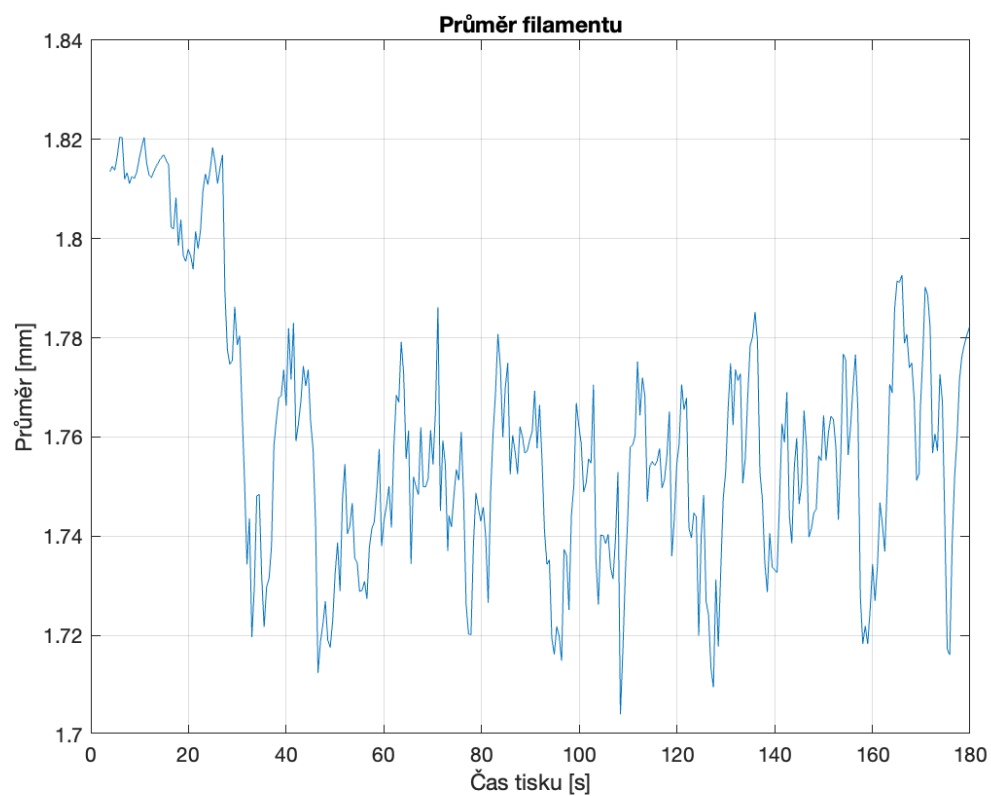
E.4 Teplota tiskové podložky



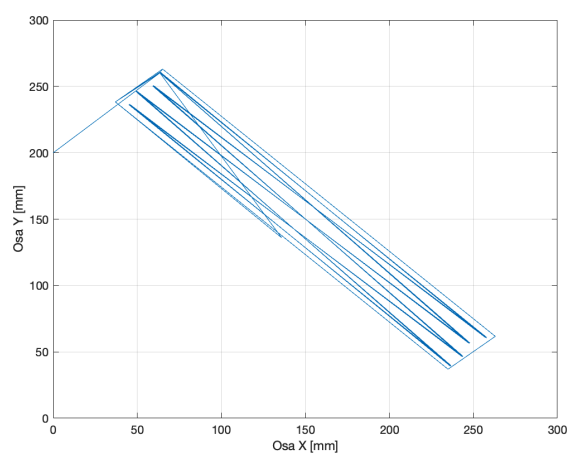
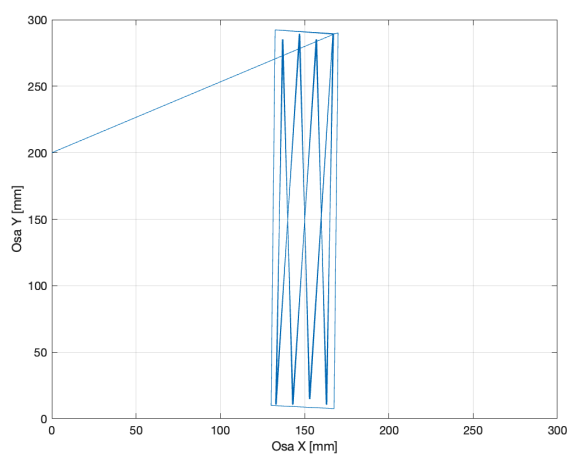
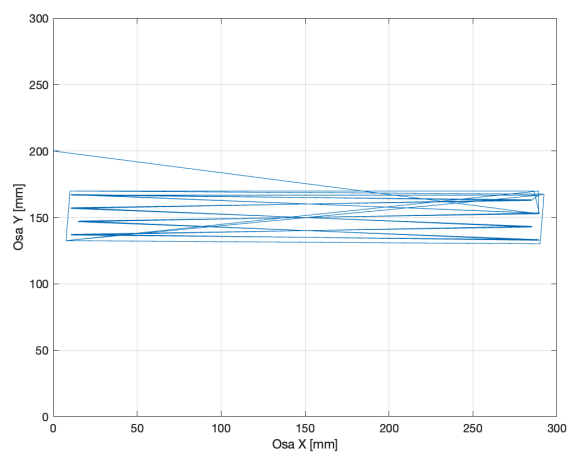
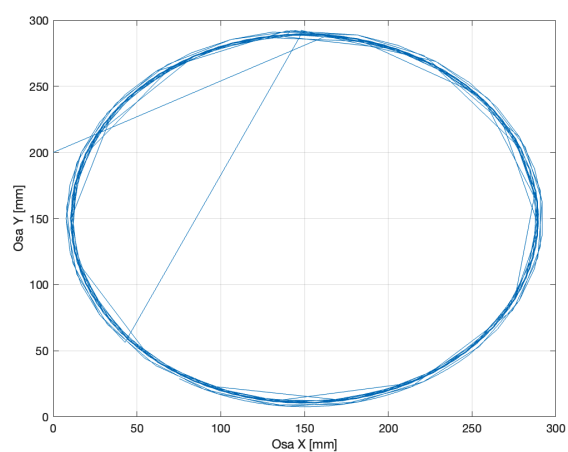
E.5 Spotřeba filamentu



E.6 Průměr filamentu

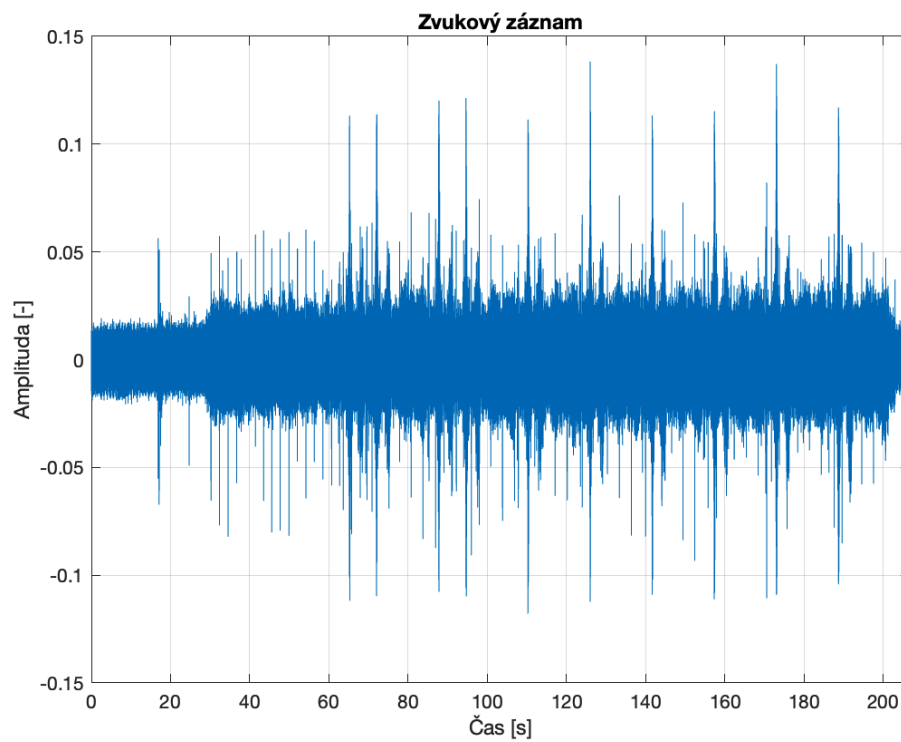


E.7 Trajektorie tiskové hlavy

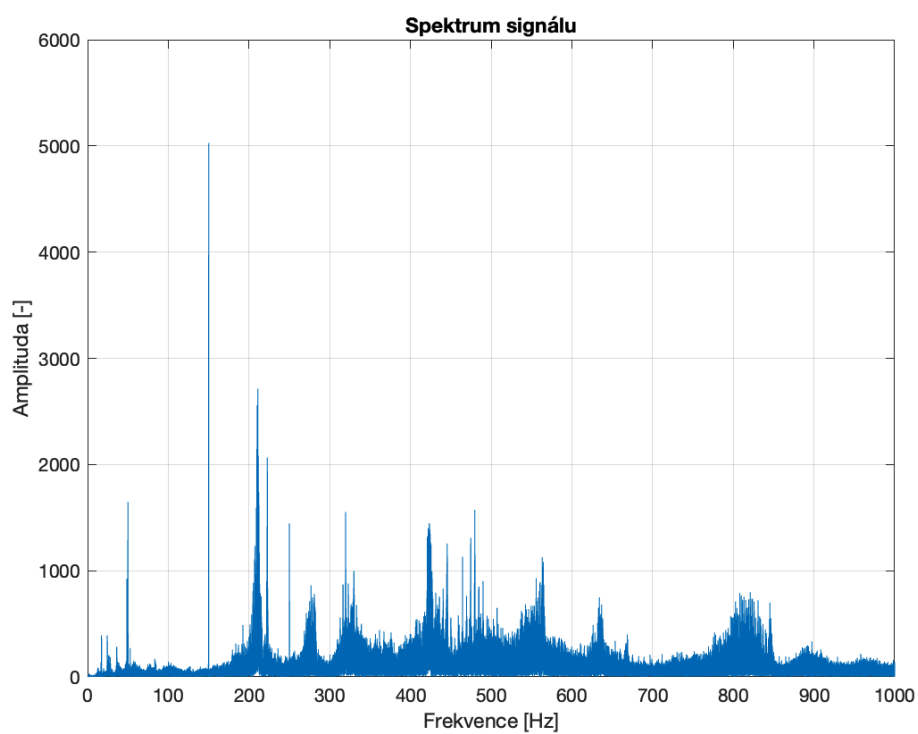


F Naměřená zvuková data při tisku

F.1 Časová oblast



F.2 Frekvenční oblast



G Konfigurace Klipperu

G.1 Konfigurace MQTT v souboru moonraker.conf

Výpis G.1: Konfigurace MQTT [26]

```
[mqtt]
address: octopus.local
    # IP adresa brokeru, případně hostname
port: 1883
    # parametr, určující na jakém portu bude broker
    # naslouchat
username: octopus
password: octopus123
    # přihlašovací jméno a heslo na mqtt broker
mqtt_protocol: v3.1.1
    # verze použitého protokolu
instance_name: octopus
    # identifikátor odlišení při více instancích
    # Moonrakeru na stejné síti
status_objects:
    # definuje, jaká data budou odesílána
    display_status = progress
    fan = rpm
    heater_fan Hotend_fan = rpm
    gcode_move = position
    hall_filament_width_sensor = Diameter
    extruder = temperature, power
    heater_bed = temperature, power
    idle_timeout = printing_time
    motion_report = live_velocity, live_extruder_velocity
    print_stats = total_duration, print_duration
    print_stats = filament_used
    system_stats = sysload, cputime, memavail
    temperature_sensor raspberry_pi = temperature
    temperature_sensor mcu_temp = temperature

publish_split_status: False
    # metoda, jakou budou data odesílána
default_qos: 0
    # typ potvrzování dat
```

Poznámky:

status_objects - Objekty budou odeslány pouze v případě, že se jejich hodnota změnila, přičemž odesílání probíhá přibližně každých 250 ms. Abychom byli schopni odesílat aktuální polohu tiskové hlavy s přesností na 0,1 mm, musela by být perioda odesílání dat zhruba 0,5 ms. Odeslaná poloha bude tedy spíše orientační.

publish_split_status - Při hodnotě *False* bude pro veškeré objekty použit jeden *topic* název. Při nastavení hodnoty *True* bude pro daný objekt nastaven vlastní *topic*.

default_qos - QoS 0 znamená, že odesílatel nevyžaduje potvrzení přijetí zprávy. QoS 1 vyžaduje od příjemce zpětnou vazbu, zda zpráva dorazila. Pokud v určitém čase nedorazí potvrzení zprávy, je zpráva znova poslána. QoS 2 je nejvyšší úroveň kvality přenosu, kdy odesílatel posílá potvrzovací zprávu s tím, že mu bylo doručeno potvrzení o přijetí původně poslané zprávy.

G.2 Konfigurace senzoru průměru filamentu v souboru printer.cfg

Výpis G.2: Konfigurace senzoru průměru filamentu [23]

[hall_filament_width_sensor]	1
adc1: PA5	2
adc2: PA4	3
# piny pro připojení senzoru	4
cal_dia1: 1.50	5
cal_dia2: 2.00	6
# kalibrační hodnoty průměru (v mm)	7
raw_dia1: 9639	8
raw_dia2: 10427	9
# kalibrační hodnoty výstupů senzoru	10
default_nominal_filament_diameter: 1.75	11
# použitý průměr filamentu	12
max_difference: 0.1	13
# maximální povolená ovalita (v mm)	14
# po překročení je ovlivněna extruze	15
measurement_delay: 310	16
# vzdálenost od senzoru k hotendu (v mm)	17
enable: True	18
# aktivace senzoru po spuštění	19
measurement_interval: 10	20
# vzdálenost mezi měřeními (v mm)	21
pause_on_runout: True	22
runout_gcode:	23
{action_respond_info("Filament_runout!")}	24
insert_gcode:	25
{action_respond_info("Filament_detected!")}	26

G.3 Konfigurace kamer v souboru crowsnest.conf

Výpis G.3: Konfigurace kamer [30]

[crowsnest]	1
log_path: /home/pi/printer_data/logs/crowsnest.log	2
log_level: verbose	3
delete_log: false	4
no_proxy: false	5
	6
[cam 1]	7
mode: mjpg	8
port: 8080	9
device: /dev/video0	10
resolution: 640x480	11
max_fps: 15	12
	13
[cam 2]	14
mode: mjpg	15
port: 8081	16
device: /dev/video2	17
resolution: 640x480	18
max_fps: 15	19

H Konfigurace Mosquitto brokera v souboru mosquitto.conf

Výpis H.1: Konfigurace Mosquitto brokera [29]

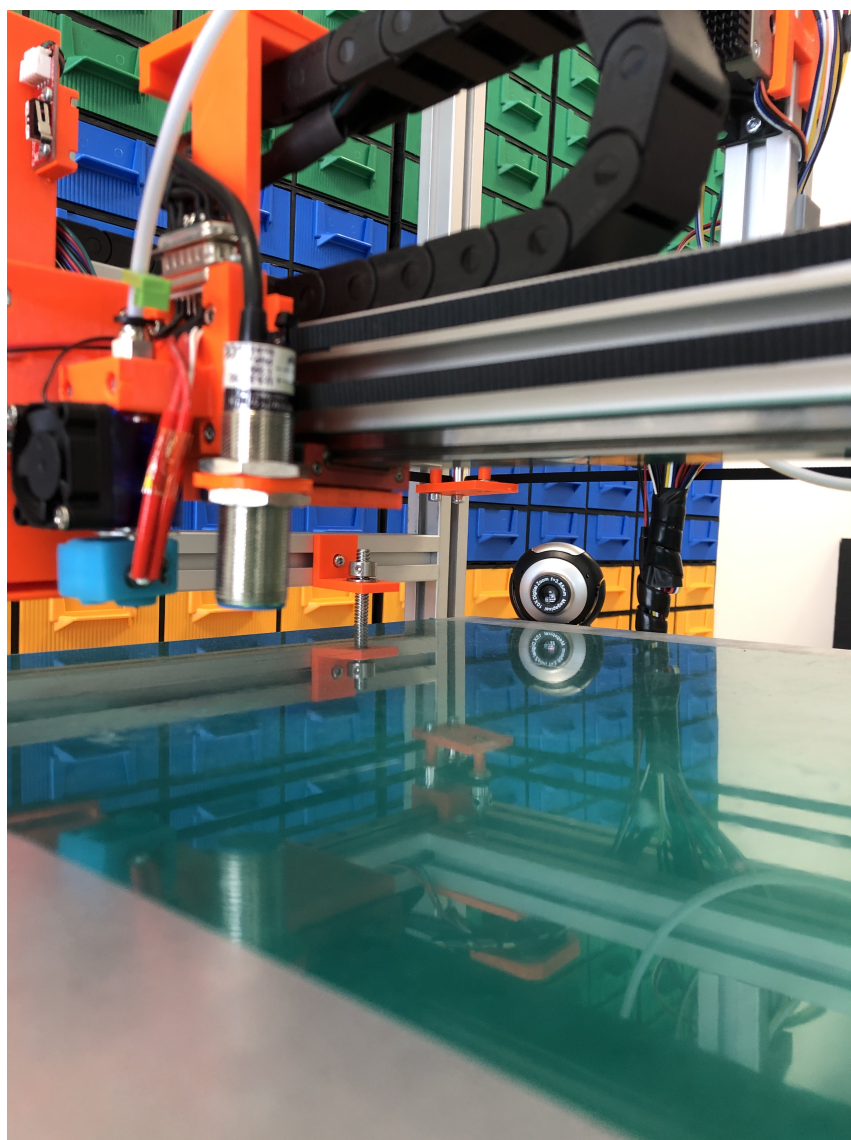
# Place your local configuration in /etc/mosquitto/conf.d/	1
#	2
# A full description of the configuration file is at	3
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example	4
	5
pid_file /run/mosquitto/mosquitto.pid	6
	7
persistence true	8
persistence_location /var/lib/mosquitto/	9
	10
log_dest file /var/log/mosquitto/mosquitto.log	11
	12
include_dir /etc/mosquitto/conf.d	13
	14
listener 1883	15
allow_anonymous false	16
password_file /etc/mosquitto/passwd	17

I Fotodokumentace

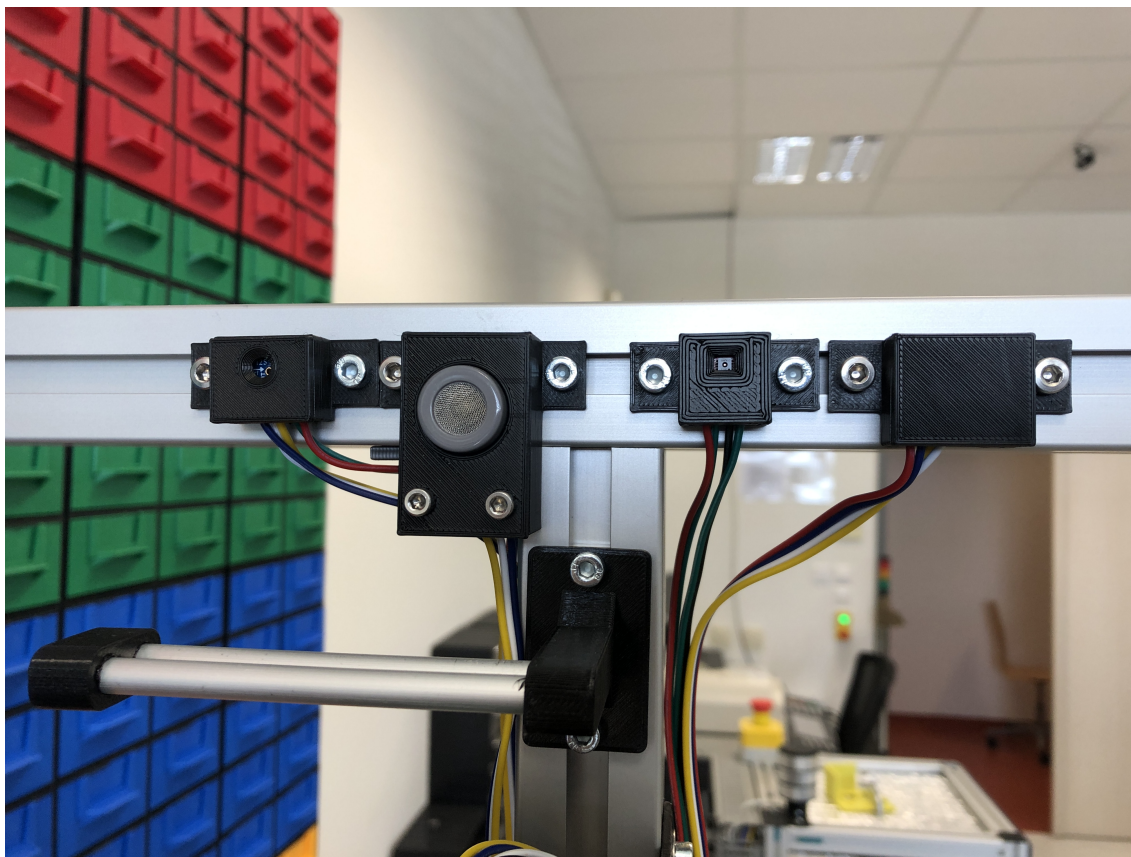
I.1 Umístění první kamery



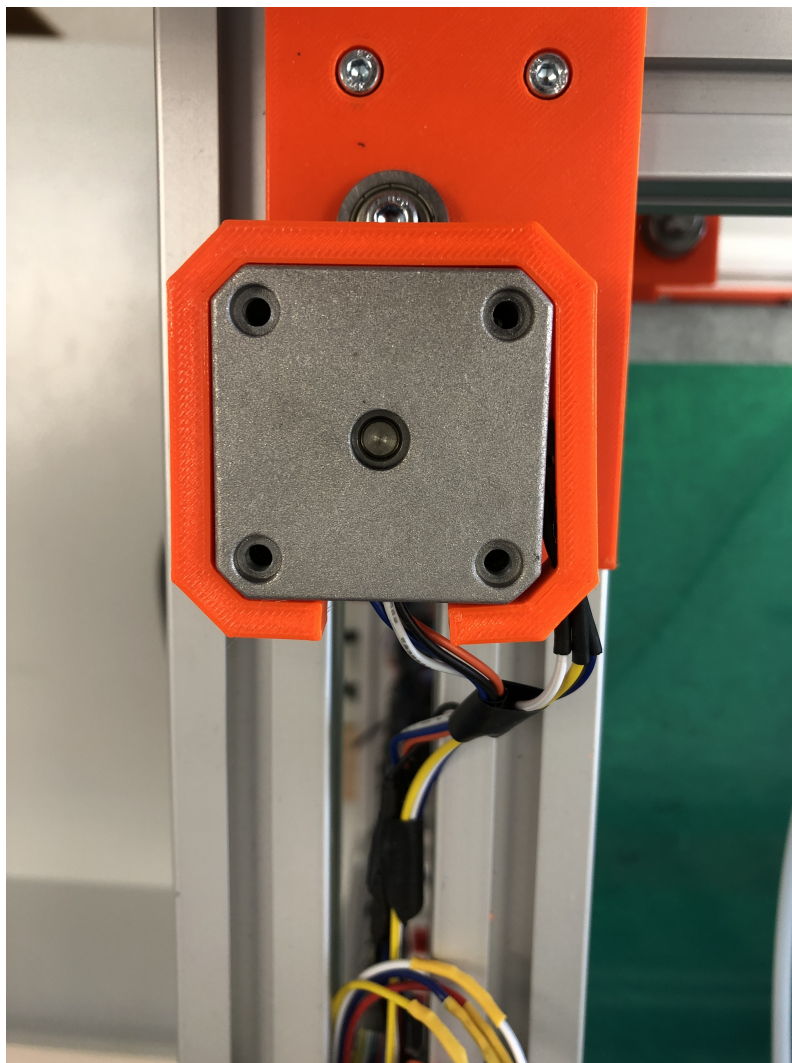
I.2 Umístění druhé kamery



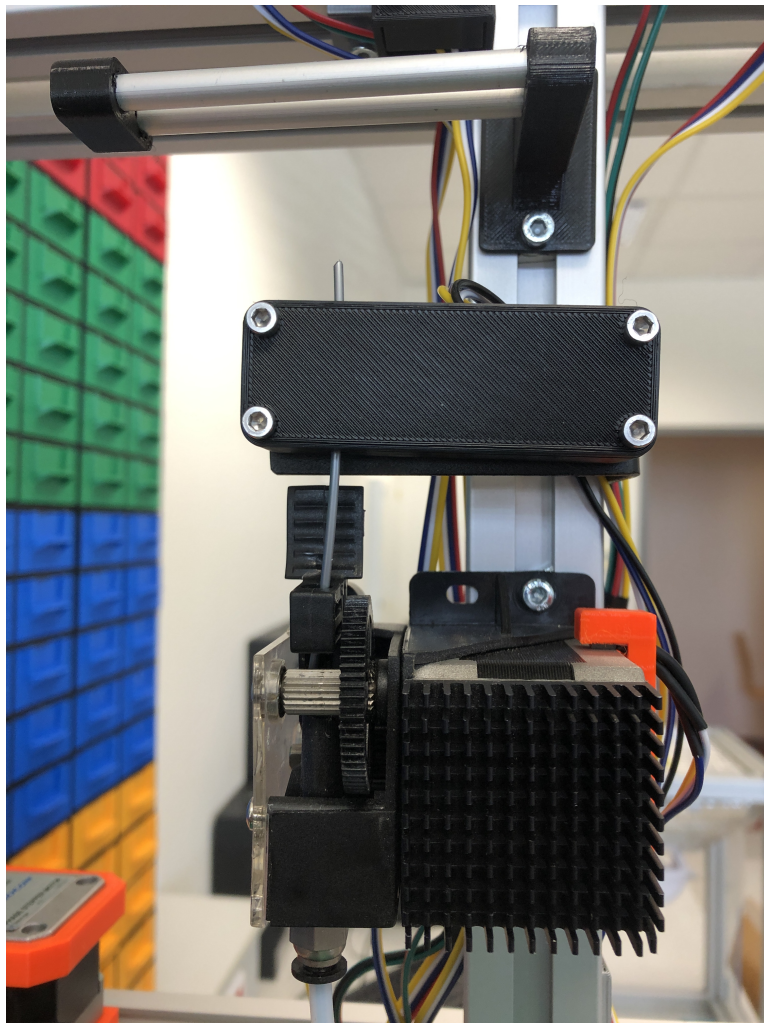
I.3 Senzory v horní části tiskárny



I.4 Držák teplotního čidla na krytu motoru



I.5 Senzor průměru filamentu



J Obsah elektronické přílohy

```
/ .....kořenový adresář přiloženého archivu
├── Text .....elektronická verze práce
│   └── BP_Vaculik_2024.pdf
├── Python .....zdrojové Python kódy
│   ├── main.py
│   ├── mqtt.py
│   ├── arduino.py
│   └── microphones.py
├── Arduino .....zdrojový kód pro Arduino Nano
│   ├── main
│   │   └── main.ino
├── Klipper .....konfigurační soubory Klipperu
│   ├── printer.cfg
│   ├── moonraker.conf
│   └── crowsnest.conf
├── Mosquitto .....konfigurační soubor Mosquitto brokera
│   └── mosquitto.conf
├── Logs .....změřená tisková data
│   ├── Arduino
│   │   ├── arduino-Circle.csv
│   │   ├── arduino-X_axis.csv
│   │   ├── arduino-Y_axis.csv
│   │   ├── arduino-XY_left.csv
│   │   └── arduino-XY_right.csv
│   └── MQTT
│       ├── mqtt-Circle.csv
│       ├── mqtt-X_axis.csv
│       ├── mqtt-Y_axis.csv
│       ├── mqtt-XY_left.csv
│       └── mqtt-XY_right.csv
```