ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: N0715A270012 Průmyslové inženýrství a management
Studijní obor: N0715A270012S00-0 Průmyslové inženýrství a management

DIPLOMOVÁ PRÁCE

Propojení Arduino a stavebnice průmyslového modelu Fishertechnik

Autor: **Bc. Petr Švrčula**

Vedoucí práce: **Doc. Ing. Petr Hořejší, Ph.D.**

Akademický rok 2020/2021

ZÁPADOČESKÁ UNIVERZITA V PLZNI

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Petr ŠVRČULA**
Osobní číslo:	**S19N0076K**
Studijní program:	**N0715A270012 Průmyslové inženýrství a management**
Studijní obor:	**Průmyslové inženýrství a management**
Téma práce:	**Propojení Arduino a stavebnice průmyslového modelu Fishertechnik**
Zadávající katedra:	**Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

1. Úvod
2. Možnosti využití Arduino v průmyslové praxi
3. Zhodnocení možností Arduino v průmyslové praxi
4. Návrh vlastní případové studie – propojení Arduino a stavebnice FisherTechnik
5. Realizace vlastní případové studie
6. Závěr a zhodnocení

Rozsah diplomové práce: **50 – 70 stran**
Rozsah grafických prací: **0 výkresů**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

1. VODA, Zbyšek. *Arduino – Průvodce světem Arduina,* 2.vydání. Bučovice: Nakladatelství Martin Stříž, 2017. ISBN 978-80-87106-93-8.
2. MONK, Simon. *Programming Arduino: Getting Started with Sketches,* Second Edition (Tab) 2nd Edition. McGraw-Hill Education, 2016. ISBN 978-1259641633.
3. BLUM, Jeremy. *Exploring Arduino: Tools and Techniques for Engineering Wizardry,* Second Edition. John Wiley & Sons, 2020. ISBN 978-1119405375.

Vedoucí diplomové práce: **Doc. Ing. Petr Hořejší, Ph.D.**
Katedra průmyslového inženýrství a managementu

Konzultant diplomové práce: **Ing. Bc. Miroslav Malaga**
Katedra průmyslového inženýrství a managementu

Datum zadání diplomové práce: **21. září 2020**
Termín odevzdání diplomové práce: **28. května 2021**

L.S.

Doc. Ing. Milan Edl, Ph.D.
děkan

Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

V Plzni dne 21. září 2020

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této diplomové práce.

V Plzni dne: …….. . . . . .

podpis autora

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce panu Doc. Ing. Petru Hořejšímu Ph.D. a konzultantovi mé diplomové práce panu Ing. Miroslavovi Malagovi za poskytnuté podnětné návrhy, odborné vedení práce, vstřícný přístup a za čas, který mi věnovali při konzultacích.

ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

AUTOR	**Příjmení** Švrčula	**Jméno** Petr	
STUDIJNÍ OBOR	Průmyslové inženýrství a management		
VEDOUCÍ PRÁCE	**Příjmení (včetně titulů)** Doc. Ing. Hořejší, Ph.D..	**Jméno** Petr	
PRACOVIŠTĚ	ZČU - FST - KPV		
DRUH PRÁCE	**DIPLOMOVÁ**	~~**BAKALÁŘSKÁ**~~	**Nehodící se škrtněte**
NÁZEV PRÁCE	Propojení Arduino a stavebnice průmyslového modelu Fishertechnik		

FAKULTA	strojní		**KATEDRA**	KPV		**ROK ODEVZD.**	2021

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	46		**TEXTOVÁ ČÁST**	37		**GRAFICKÁ ČÁST**	9

STRUČNÝ POPIS (MAX 10 ŘÁDEK) **ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY**	Diplomová práce se zabývá rozborem možností platformy Arduino v průmyslu a řeší vzájemné propojení Arduina a průmyslového modelu Fishertechnik. Cílem práce je navrhnout funkční řešení pro ovládání průmyslového modelu Fishertechnik z osobního počítače pomocí mikrokontroleru Arduino. K tomu využívá komunikační sběrnice USB a I^2C. Přínosy práce jsou vytvoření vzdělávacího nástroje, demonstrace možností Arduina a Fishertechnik v oblasti průmyslových podniků směrem k optimalizaci procesů v rámci Průmyslu 4.0. Práce také může být podkladem pro další vývoj v oblasti návrhu a prezentace digitálních podniků.
KLÍČOVÁ SLOVA **ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE**	Arduino, Fishertechnik, IoT, Průmysl 4.0, RoboPro, STEM

SUMMARY OF DIPLOMA SHEET

AUTHOR	**Surname** Švrčula	**Name** Petr	
FIELD OF STUDY	Industrial engineering and management		
SUPERVISOR	**Surname (Inclusive of Degrees)** Doc. Ing. Hořejší, Ph.D..	**Name** Petr	
INSTITUTION	ZČU – FST – KPV		
TYPE OF WORK	**DIPLOMA**	~~**BACHELOR**~~	**Nehodící se škrtněte**
TITLE OF THE WORK	Connection of Arduino and Fischertechnik industrial model kit		

FACULTY	Mechanical Engineering		**DEPARTMENT**	KPV		**SUBMITTED IN**	2021

POČET STRAN (A4 a ekvivalentů A4)

TOTTALLY	**46**		**TEXT PART**	37		**GRAPHICAL PART**	9

BRIEF DESCRIPTION **TOPIC, GOAL, RESULTS AND CONTRIBUTIONS**	The diploma thesis deals with the analysis of the possibilities of the Arduino platform in the industry and solves the connection of Arduino and the industrial model kit Fishertechnik. This work aims to design a functional solution for controlling the industrial model Fishertechnik from a personal computer using an Arduino microcontroller. It uses the USB and I^2C communication buses for this. The benefits of the work are the creation of an educational tool, a demonstration of the possibilities of Arduino and Fishertechnik in the field of industrial enterprises towards the optimization of processes within Industry 4.0. The work can also be a basis for further development in the design and presentation of digital businesses.
KEY WORDS	Arduino, Fishertechnik, Industry 4.0, IoT, RoboPro, STEM

Obsah

Seznam zkratek

AVR	označení 8 a 32bitových procesorů typu RISC od společnosti Atmel
Baud	jednotka modulační rychlosti
C#	vysokoúrovňový objektově orientovaný programovací jazyk od společnosti Microsoft
CAN	Controller Area Network (komunikační sběrnice)
COM	komunikační rozhraní
CPU	centrální procesorová jednotka
DAC	digitálně analogový převodník
DC	stejnosměrný proud
EDBG	vestavěný debugger některých procesorů Atmel
GPU	grafická procesorová jednotka
HMI	rozhraní člověk – stroj
HTTP	internetový protokol
I/O	vstupně / výstupní
I^2C	Inter-Integrated Circuit – komunikační sběrnice
ICSP	způsob programování mikrokontrolerů
IDE	vývojové prostředí
IoT	internet věcí
IT	informační technologie
JPEG	datový formát obrázku
JSON	způsob zápisu dat nezávislý na počítačové platformě
JTAG	standard definovaný normou IEEE 1149.1
LED	světlo emitující dioda
Li-Po	lithium polymerový akumulátor
MQTT	odlehčený síťový protokol pro zasílání zpráv mezi různými zařízeními
OTG	normativní specifikace dovolující vzájemnou USB komunikaci mezi přístroji
PC	osobní počítač
PID	regulátor složený z proporcionální, integrační a derivační části
PLC	programovatelný logický automat
PWM	pulsně šířková modulace
RGB	barevný model
RS-232	standard komunikačního rozhraní
RS-485	standard komunikačního rozhraní
SPI	sériová periferní rozhraní
STEM	věda, technologie, strojírenství a matematika
UART	počítačová asynchronní sériová sběrnice
USB	univerzální sériová sběrnice
USB-B	standard konektoru USB
USB-C	standard konektoru USB
WPF	windows presentation foundation
X.509	kryptografický standard
XAML	značkovací jazyk využívaný k popisu grafického rozhraní

1 Úvod

V teoretické části, tato práce obsahuje souhrn vybraných aplikací různých variant platformy Arduino v průmyslovém oboru. Arduino je téměř univerzální nástroj pro vytváření různých projektů v mnoha oborech. Ačkoliv tato platforma nedisponuje průmyslovými standardy, nachází využití ve stále větším počtu průmyslových aplikací. Arduino je využíváno od jednoduchých prvků, kdy slouží jako překladač mezi ostatními komponenty přístroje, po komplexní zařízení, kde Arduino vykonává úlohu řídící jednotky. Využívání Arduina a podobných mikrokontrolerů se jeví jako přirozený krok v důsledku zvyšující se automatizace a robotizace.

Cílem této práce je vytvoření komplexního propojení průmyslového modelu Fishertechnik STEM ENGINEERING, mikrokontroleru Arduino a osobního počítače. Arduino v tomto případě slouží jako obousměrná komunikační jednotka mezi rozhraním USB na straně PC a sběrnicí I^2C na straně průmyslového modelu. Součástí řešení je vytvoření obslužné aplikace pro operační systém Windows. Stavebnice Fishertechnik je využívána pro modelování simulací automatizovaných provozů. Fishertechnik sice umí komunikovat s PC přímo, ale je nutné používat software ROBO Pro, nebo aplikaci vyvíjet v programovacím jazyku C nebo python. Použitím můstku Arduino se celé řešení zjednoduší a umožní využívat vlastní aplikace postavené v C# nebo jiné možnosti platformy Arduino.[17]

Pro realizaci tohoto projektu je zvolen průmyslový model Fishertechnik Sorter. Jedná se o model robotického ramena, které automaticky třídí barevné puky na základě jejich barvy do tří boxů. Z aplikačního rozhraní počítače bude možné procesy monitorovat, zjišťovat barvu aktuálního uchopeného puku a také vyvolat manuální spuštění celého procesu pomocí ovládacího tlačítka. V rámci vzorového modelu bude vytvořena aplikace pro počítač, program pro Arduino a bude provedena implementace již hotových prvků. Těmi jsou program Fishertechnik a dílčí komunikační rozhraní USB – Arduino a Arduino – Fishertechnik. Program pro Arduino bude vytvořen pomocí programovacího jazyka Wiring (vycházející z C/C++), obslužná aplikace bude vytvořená pomocí programovacího jazyka C#.

Průmyslové inženýrství předpokládá znalost z různých technických oborů jako je matematika, konstruování, IT a programování, mechanika, fyzika, elektrotechnika a jiné. Tyto znalosti se uplatňují při zlepšování výrobních i nevýrobních procesů v průmyslových podnicích. Právě koncept STEM spojuje tyto znalosti a představuje filozofii polytechnického vzdělání. V konceptu STEM se spojují znalosti z vědy (**S**cience), techniky (**T**echnology), technologie (**E**ngineering) a matematiky (**M**atematics). Stavebnice Fishertechnik, která do tohoto konceptu spadá, tak demonstruje možnosti využití těchto znalostí pro účely vzdělávání, simulací provozů nebo demonstraci navrhnutých řešení. Výstupy této práce jsou kombinací těchto znalostí a ukazují široké možnosti jejich uplatnění. [17]

Přínosem této práce je vytvoření základního stavebního kamene pro další stavbu komplexnějších systémů jako je průmyslová továrna ve virtuální realitě, kdy změny učiněné ve světě virtuální reality budou promítnuty do modelu v reálném světe. Tyto změny budou probíhat v reálném čase a v obou směrech. Interakce uživatele bude možná jak z fyzické, tak virtuální části stroje. Práci bude možné také využívat jako studijní pomůcku pro pochopení základních vztahů mezi počítačem a stavebnicí Fishertechnik.

Toto téma jsem si zvolil s ohledem na probíhající průmyslovou revoluci Průmysl 4.0. Průmysl 4.0 je orientován na maximální možnou automatizaci výrobních procesů a obsahuje mnoho potencionálních příležitostí pro využití platformy Arduino a stavebnice Fishertechnik.

2 Možnosti využití Arduino v průmyslové praxi

2.1 Popis platformy Arduino

Arduino vzniklo v roce 2005 v Ivrea Interaction Design Institute v Itálii. Jedná se o otevřenou platformu mikrokontrolerů, většinou osazených procesory Atmel. S pomocí Arduina je možné vyčítat úrovňové stavy digitálních a analogových vstupů. Výstupní piny mohou nabývat diskrétních hodnot (LOW, HIGH) nebo mohou být řízeny pomocí PWM (pulsně šířkové modulace). Základní možnost komunikace Arduina s okolním světem představuje sériová linka. Připojení je většinou realizováno přes port USB, kdy se o překlad stará čip FTDI, většinou přímo integrovaný na desce Arduino. Některé typy desek Arduino mají možnost připojení další sériové linky přes UART patice, většina typů desek také umožňuje komunikaci pomocí sběrnice I^2C. Desky jsou pro komunikaci dále schopné využívat technologii Bluetooth nebo wifi. Prvky pro komunikaci těmito technologiemi je možné připojit k desce jako přídavný modul a některé typy Arduin mají tyto prvky přímo integrované na základní desce. Arduino je řízeno pomocí takzvaných Sketchů. Tyto programy jsou psány v jazyce Wiring, který vychází z jazyka C/C++. K napsání a nahrání programu do mikrokontroleru slouží vývojové prostředí IDE Arduino. Jedná se o jednoduché vývojové prostředí, které je snadno použitelné pro začátečníky, ale dostatečně flexibilní pro pokročilé uživatele. [1, 2, 13, 15]

Arduino bylo stvořeno jako snadný nástroj pro rychlé prototypování, zaměřený na studenty bez znalostí elektroniky a programování. Jakmile se deska Arduino dostala do širší komunity, začala se přizpůsobovat novým potřebám a výzvám. Výsledkem je široké spektrum nabízených variant od jednoduchých 8bitových desek po produkty pro aplikace IoT, nositelná zařízení, 3D tisk a vestavěná prostředí. Všechny desky Arduino jsou open-source, to umožňuje uživatelské základně vytvářet vlastní návrhy desek nebo je přizpůsobit pro své konkrétní potřeby.

Za dobu své existence bylo Arduino mozkem tisíců projektů, od každodenních předmětů, až po složité vědecké přístroje. Kolem platformy se shromáždila celosvětová komunita uživatelů z různých profesí. Učitelé a studenti jej používají k výrobě levných vědeckých přístrojů, k prokázání chemických a fyzikálních principů nebo k učení programování a robotiky. Návrháři a architekti vytvářejí interaktivní prototypy, hudebníci a umělci jej používají pro instalace a experimentování s novými hudebními nástroji. Tvůrci jej využívají například k výstavbě svých projektů vystavovaných na veletrhu Maker Faire. Arduino je klíčový nástroj k učení se novým věcem. [1]

Na trhu je samozřejmě možné nalézt i jiné platformy mikrokontrolerů s podobnými funkcemi. Arduino však ve srovnání s nimi nabízí několik výhod:

- **Nízká cena** – desky Arduino jsou relativně levné ve srovnání s jinými platformami mikrokontrolerů.
- **Použití napříč platformami** – vývojové prostředí Arduino IDE je možné spustit na systémech Windows, Macintosh OSX a Linux.
- **Jednoduché a přehledné programovací prostředí.**
- **Open source a rozšiřitelný software** – Arduino IDE je publikováno jako open source nástroj, který mohou rozšířit zkušení programátoři. Programovací jazyk Wiring je pak možné rozšířit prostřednictvím volně dostupných knihoven C++ a zkušení programátoři mohou přímo využívat programovací jazyk AVR C, na kterém jsou založené mikrokontrolery Atmel. [1]

- **Open source a rozšiřitelný hardware** – plány desek Arduino jsou publikovány pod licencí Creative Commons, zkušení designéři obvodů tak mohou vytvořit vlastní verzi modulu, rozšířit ji a vylepšit. [1]

2.2 Rozdělení desek Arduino

Jelikož je platforma Arduino open source, umožňuje vytváření různých variant základních desek. Tato kapitola popisuje nejrozšířenější desky postavené na platformě Arduino.

2.2.1 Arduino Nano

Arduino Nano je malá deska založená na procesoru ATmega328. Postrádá DC napájecí konektor a pracuje s kabelem Mini-B USB místo standardního USB-B. [2]

2.2.2 Arduino Micro

Arduino Micro je deska mikrokontroleru založená na procesoru ATmega32U4, vyvinutá ve spolupráci se společností Adafruit. Má 20 digitálních vstupních / výstupních pinů (7 z nich lze použít jako PWM výstupy a 12 jako analogové vstupy), deska obsahuje 16 MHz oscilátor, připojení micro USB, port ICSP (In-Circuit Serial Programming) a resetovací tlačítko. [2]

Díky použití procesoru ATmega32U4 má deska vestavěnou USB komunikaci, což eliminuje potřebu sekundárního procesoru. Tato vlastnost dovoluje kontroléru emulovat myš a klávesnici během připojení k PC. [3]

2.2.3 Arduino Leonardo

Tato deska má identické vlastnosti jako Arduino Micro, liší se tím, že má jiné rozměry a je osazena externím DC napájecím konektorem. [2]

2.2.4 Arduino MEGA 2560

Deska Arduino MEGA 2560 viz obr. 1, je založená na procesoru ATmega2560. Má 54 digitálních vstupních / výstupních pinů (15 z nich lze použít jako PWM výstupy), 16 analogových vstupů, 4 UART (hardwarové sériové porty), 16 MHz oscilátor, připojení USB, DC napájecí konektor, port ICSP a resetovací tlačítko. [2, 4, 5]

Obrázek 1 Arduino MEGA 2560 [2]

2.2.5 Arduino Due

Arduino Due je deska mikrokontroleru postavená na procesoru Atmel SAM3X8E ARM Cortex-M3. Jedná se o první desku Arduino založenou na 32bitovém mikrokontroleru ARM. Má 54 digitálních vstupních / výstupních pinů (12 z nich lze použít jako výstupy PWM), 12 analogových vstupů, 4 UART porty, 48 MHz hodiny, připojení USB OTG (On-The-Go), 2 DAC piny, napájecí konektor, port SPI (Serial Peripheral Interface), port JTAG (Joint Test Action Group JTAG), resetovací a mazací tlačítko. Na rozdíl od většiny desek Arduino pracuje deska Arduino Due na napěťové hladině 3,3 V. [2, 3, 4]

2.2.6 Arduino Uno

Tato deska patří k nejpoužívanějším verzím Arduina. Deska je založená na čipu ATmega328. Má 14 digitálních vstupních / výstupních pinů (6 z nich lze použít jako PWM), 6 analogových vstupů, 16 MHz oscilátor, připojení USB, DC napájecí konektor, konektor ICSP a resetovací tlačítko. Uno byla první verzí desky Arduino pro USB připojení, která byla vydaná společně se softwarem Arduino IDE 1.0. [2, 3, 5]

2.2.7 Arduino Zero

Arduino Zero je výkonná deska využívající procesor Atmel SAMD21 MCU, který obsahuje 32bitová jádro ARM Cortex. Jednou z jeho nejdůležitějších funkcí je Atmel's Embedded Debugger (EDBG), který poskytuje úplné ladicí rozhraní bez nutnosti dalšího hardwaru. Tato funkce výrazně ulehčuje ladění vyvíjeného softwaru. Mikrokontroler poskytuje vyšší výkon, umožňuje širší uplatnění a zároveň slouží jako edukační pomůcka pro vývoj 32bitových aplikací. Deska obsahuje 20 digitálních vstupních / výstupních pinů (10 lze použít jako výstupy PWM), 6 analogových vstupů, připojení USB OTG, 2 UART porty, 48 MHz hodiny, 1 DAC pin, napájecí konektor, port SPI, port JTAG a resetovací tlačítko. Stejně jako Arduino Due pracuje tato deska na napěťové hladině 3,3 V. [2]

2.2.8 Arduino MKR WiFi 1010

Model MKR WiFi 1010 je primárně určený pro nasazení v IoT, hlavním procesorem desky je 32bitový SAMD21 Arm® Cortex®-M0 s nízkou spotřebou. Připojení wifi a Bluetooth® je realizováno pomocí modulu NINA-W10, nízkoenergetické čipsetu pracujícího v rozsahu 2,4 GHz. Zabezpečená komunikace je zajištěna prostřednictvím krypto čipu Microchip® ECC508. Deska je dále vybavena 8 digitálně vstupních / výstupních pinů (všechny lze použít jako PWM výstupy), 7 analogovými vstupy a 1 DAC výstupem, porty UART, SPI a připojením pro článek Li-Po 3,7V. [2, 6]

Další desky rodiny MKR jsou:

- **Arduino Uno WiFi rev2**: výuková verze MKR WiFi 1010 s konektorem USB-B a integrovaným akcelerometrem. [2]
- **Nano 33 IoT**: menší provedení, bez připojení článku Li-Po, základní funkce stejné. [2]
- **MKR WiFi 1000**: bez Bluetooth, pouze s wifi, obsahuje jinou čipovou sadu než MKR WiFi 1010. [2]

2.2.9 Arduino Portenta H7

Nejnovější generace desky Arduino, postavená na technologii Arm® Pelion ™, přináší uživatelům jednoduchost integrace a škálovatelnou, bezpečnou a profesionálně podporovanou službu. Portenta H7 je navržená pro průmyslové použití a nabízí výkonnou desku pro řízení automatizace, robotiku a AI (Artifical Inteligence). Arduino Portenta H7 viz obr. 2, s dvoujádrovými Arm® Cortex®-M7 a M4 na 400 MHz je schopna provozovat jazyky Wiring, Javascript a Python. Portenta H7 je první deska z rodiny Arduino, která je osazená dvěma procesory a současně spouští kód na vysoké úrovni společně s vykonáváním strojových instrukcí v reálném čase. Oba procesory mohou spouštět úlohy paralelně. Například je možné spustit kompilovaný kód Wiring spolu s kódem MicroPython, kdy spolu oba procesory navzájem komunikují. Portenta může fungovat buď jako kterákoli jiná integrovaná deska mikrokontroleru, nebo jako hlavní procesor integrovaného počítače. V kombinaci s Portenta Vision Shield může mikrokontroler fungovat jako průmyslová kamera, která je schopná provádět algoritmy strojového učení v reálném čase na živých video kanálech. [2, 3]

Obrázek 2 Arduino Portenta H7 [2]

Portenta je osazena hlavním procesorem STM32H747XI dual Cortex-M7, běžícím na frekvenci 480 MHz a druhým procesorem M4 32bit low power Arm® MCU, běžícím na frekvenci 280 MHz. Dvě jádra komunikují prostřednictvím vzdáleného volání procedur, to umožňuje bezproblémové volání funkcí na druhém procesoru. Oba procesory sdílejí všechny periferie v čipu. Deska je dále vybavena konektivitou USB, wifi, Bluetooth, ethernet a čtečkou SD karet, deska je vyráběna v rozměrovém standardu MKR a obsahuje 15 digitálních vstupních / výstupních pinů (7 lze použít jako výstupy PWM), 7 analogových vstupů, připojení pro článek Li-Po 3,7V včetně dobíjecího okruhu, 4 porty UART a 2 DAC piny. [2, 3]

Mikrokontroler dokáže spouštět tyto procesy [2, 3]:

- Wiring sketch v aplikační vrstvě OS Arm® Mbed ™,
- nativní aplikace Mbed ™,
- MicroPython / JavaScript ,
- TensorFlow Lite.

Portenta je první deska Arduino určená pro průmyslové nasazení. Deska má velký potenciál pro aplikace v oblastech [2]:

- špičkové průmyslové stroje,
- laboratorní vybavení,
- počítačové vidění,
- PLC,
- průmyslová uživatelská rozhraní,
- robotika,
- samostatný integrovaný počítač.

Jednou z nejzajímavějších funkcí Portenta H7 je možnost připojení externího monitoru k vytvoření vlastního integrovaného počítače s uživatelským rozhraním. Tuto vlastnost umožňuje GPU (Graphics Processing Unit) procesoru STM32H747 na čipu Chrom-ART Accelerator ™. Kromě GPU obsahuje čip dedikovaný kodér a dekodér formátu JPEG. [2, 3]

Rodina Portenta využívá dva 80 pinové konektory s vysokou hustotou pinů ve spodní části desky. Tato vlastnost zajišťuje škálovatelnost zařízení pro širokou oblast aplikací jednoduchým upgradem desky Portenta. Programovacím konektorem desky je port USB-C, který lze použít k napájení desky, k připojení monitoru DisplayPort, nebo k napájení zařízení připojených k OTG. [2, 3]

2.2.10 Ostatní desky

Výše uvedené desky reprezentují výběr nejvíce využívaných typů desek Arduino. Na trhu lze najít mnoho dalších typů a různých výrobců. Někteří uživatelé si dokonce vytvářejí vlastní typy nebo si upravují ty již vytvořené. Desky uvedené v této práci jsou originální desky Arduino vyráběné společností Arduino AG v Itálii. Neoriginální verze, označované jako klony, bývají levnější variantou, avšak často na úkor kvality. Nicméně je možné nalézt i produkty zavedených značek, které bývají dodávané ve srovnatelné kvalitě jako originální verze.

2.3 Arduino v porovnání s průmyslovým PLC

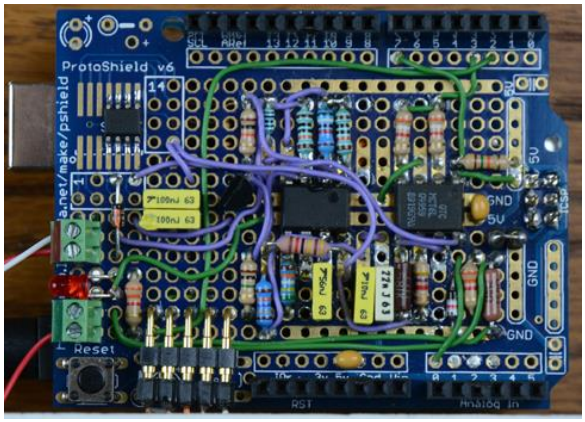
Na první pohled je Arduino zajímavá alternativa k běžnému PLC (Programmable Logic Controller). Mikrokontrolery platformy Arduino nabízejí podobné vlastnosti jako PLC za výrazně nižší náklady. Pro účely této práce bylo vybráno několik projektů s otestováním Arduina v průmyslovém prostředí.

2.3.1 PID regulace průtoku

Jako praktickou ukázku je možné zmínit projekt na regulaci průtoku. Hlavní úlohou projektu bylo porovnat možnosti platformy Arduina oproti průmyslovému PLC. Automatizační úkol byl vytvoření regulačního členu pro regulaci průtoku. Senzor měří průtok a odesílá data do mikrokontroleru, který upravuje aktuátor ovládacího ventilu pro dosažení požadované hodnoty průtoku. Arduino využívá PI řízení ke čtení signálu z průtokoměru a regulaci ventilu. Koncept je dostatečně jednoduchý, ale při práci s průmyslovými zařízeními v reálném světě se komplikuje. [7]

Arduino má digitální i analogové vstupy a výstupy, ty ale mají svá omezení, analogové vstupy fungují v rozsahu 0-5 V a analogové výstupy jsou řízeny pomocí PWM. Většina průmyslových analogových přístrojů a akčních členů je však navržena pro proudové smyčky 4–20 mA, proto byl tento standard použit v demonstračním projektu. [4, 7]

Demonstrační zařízení využívá standardní průmyslové komponenty: průtokoměr Rosemount 3051SFP Integral Orifice a regulační ventil Fisher Easy-Drive, oba poskytované společností Emerson Automation Solutions. Oba komponenty jsou běžným typem zařízení v průmyslovém prostředí. Nejprve bylo nutné převést proudový signál 4–20 mA na napěťovou úroveň 1-5 V. Ačkoliv je možné převodník pro tento účel koupit, v tomto projektu není použit, z důvodu zachování DIY charakteru. Projekt pro převod úrovní používá jednoduchý 250 Ω rezistor. Náročnějším úkolem byl převod PWM na 4-20 mA. Nedostatek komerčních řešení naznačuje, že se nejedná o běžný převod. Pro účely projektu musel být tedy vyroben úplně od začátku. Převodník viz obr. 3, byl sestaven z opto izolátoru signálových filtrů a operačního zesilovače. Všechny součástky převádějící signál se vešly na stejný shield. [7]



Obrázek 3 Arduino s vytvořeným převodníkem úrovní [7]

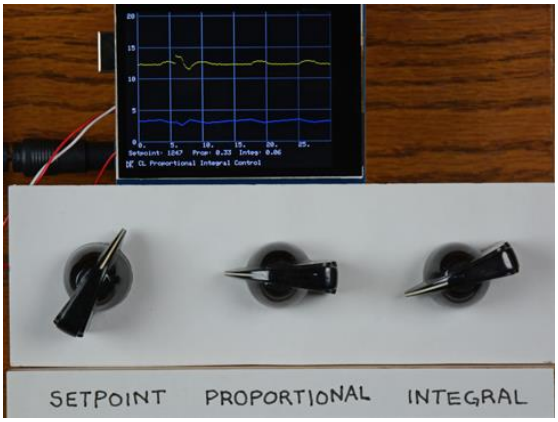
Arduino nemá napájecí zdroj ani žádný typ rozhraní člověk-stroj (HMI), ale je možné použít malý grafický displej [3]. Pro porovnání, PLC má obvykle více možností rozhraní HMI vyžadujících pouze jednoduchou konfiguraci. [7] Připojení vodičů je také základní, takže připojení externích zařízení nějakou dobu trvá. Pro montáž desek Arduino je také nutné vymyslet instalační kryt, protože Arduino v základu neobsahuje kryt ani žádné praktické doplňky, jako je montáž na lištu DIN.

Napájení projektu bylo realizováno stejnosměrným zdrojem o napětí 24 V. Ten je vhodný pro napájení průtokoměru a regulačního ventilu. Pro napájení Arduina bylo nutné napětí snížit na 12 V, k tomu byl použit jednoduchý lineární regulátor. [7]

Vytvoření kódu pro implementaci PI algoritmu na Arduinu muselo být prováděno od nuly. Princip kódu je následovný [7]:

- čtení proudové smyčky,
- porovnání rozdílu mezi žádanou hodnotou a vstupními daty,
- výpočet integrální a proporcionální korekce,
- odeslání výsledku na výstup proudové smyčky PWM.

Výkon řídicího systému je zobrazován jako procesní proměnná na připojeném displeji. Nastavení provozních vstupů je realizováno pomocí potenciometrů připojených k analogovým vstupům viz obr. 4. Jeden ovládá požadovanou hodnotu průtoku, další dva poskytují proporcionální a integrální zisky pro řídicí smyčku a jsou nastavovány tak, aby vyvážily stabilitu a dobu odezvy systému.



Obrázek 4 Proporcionální řízení [7]

Průtokoměr má vlastní displej zobrazující průtok v galonech za minutu a diferenční tlak v palcích vody, což poskytuje externí indikaci funkčnosti smyčky. V demonstrační sestavě je vložen obtokový ventil průtoku, kterým lze provádět změny v procesu a nutit smyčku k nastavení ventilu pro dodržení požadované hodnoty průtoku. [7]

Druhá fáze demonstračního projektu byla postavena s použitím low-end PLC BRX (BX-DM1E-10ED23-D) od společnosti AutomationDirect viz obr. 5. PLC má jeden analogový výstup a jeden analogový vstup, oba konfigurovatelné pro provoz proudové smyčky 4-20 mA, spolu s několika samostatnými vstupy a výstupy. Programovací přístup k PLC je z externího počítače s Windows přes Ethernet. Kompletní konfigurace PLC je pomocí softwaru „Do-more Designer“. [7]

Obrázek 5 Porovnávané PLC [7]

Pomocí softwaru byl analogový vstup a výstup konfigurován pro proudovou smyčku 4-20 mA. Tato operace byla mnohem jednodušší než navrhnout a postavit smyčkové rozhraní pro Arduino. PLC na rozdíl od Arduina nabízí propracovanou metodu PID, která umožňuje nastavit různé parametry smyčky pro ruční nebo automatické ovládání. Po konfiguraci bylo PLC integrováno do demonstračního projektu. Připojení k průtokoměru bylo rychlé a jednoduché, stejně tak jako připojení řídicího ventilu. Úprava proporcionálních a integrálních koeficientů funkce PID byla provedena pomocí řídícího softwaru Do-more Designer. [7]

Oba systémy vykázaly dostatečný výkon pro obsluhu této jednoduché úlohy. [7] Většina průmyslových aplikací však bude vyžadovat celou řadu dalších diskrétních a analogových řídicích funkcí. Výhodou PLC je určitý rozsah integrace těchto funkcí. Při využití Arduina bude nutné psát tyto funkce od nuly. U PLC je možné očekávat vyšší životnost, jednoduše proto, že je pro nasazení v těchto provozech designováno. Zároveň je většina PLC součástí produktových rodin, nabízí škálovatelnost a rozšiřování nativních schopností. Ačkoliv se Arduino jeví jako levnější varianta, tak při zahrnutí nákladů na nutné dodatečné komponenty, programování a vyšší časovou náročnost, tomu tak být nemusí.

2.3.2 Inteligentní autobusová zastávka

Dalším popisovaným projektem je návrh inteligentní zastávky pro autobusovou dopravu. Tento projekt je od samého začátku založen čistě na platformě Arduino. Projekt zkoumá možné způsoby, jak lze autobusové zastávky vylepšit pomocí IoT a poskytnout tak veřejnosti užitečné informace během čekání na spoj. Prototyp této zastávky je cílen na zobrazování inteligentní reklamy v autobusových zastávkách. Projekt se zaměřuje na využití digitálního zobrazování informací na veřejných prostranstvích, jako jsou autobusové zastávky. Obsahem, na který je projekt zaměřen, jsou digitální inzerce od podniků a institucí ve městě. Integrace IoT do inteligentní zastávky autobusu zahrnuje použití několika druhů senzorů, které shromažďují kontextová data od cestujících a také z okolního prostředí. Na základě těchto dat pak systém zobrazuje cílený obsah. Vedlejším využitím instalovaných senzorů je snížení spotřeby energie, kdy je pomocí senzorů řízeno např. stmívání světel, vypínaní displejů atd. Chytrá zastávka obsahuje hlavní klíčové funkcionality jako konektivitu, interakci s okolím, digitální mapové podklady a informační displeje. Datové připojení je realizováno pomocí 3G modemu, skrze který je zastávka připojena do informačního cloudu, který obsahuje veškerá potřebná data. K řízení chytré zastávky je použito Arduino UNO společně s ethernetovým shieldem. [8]

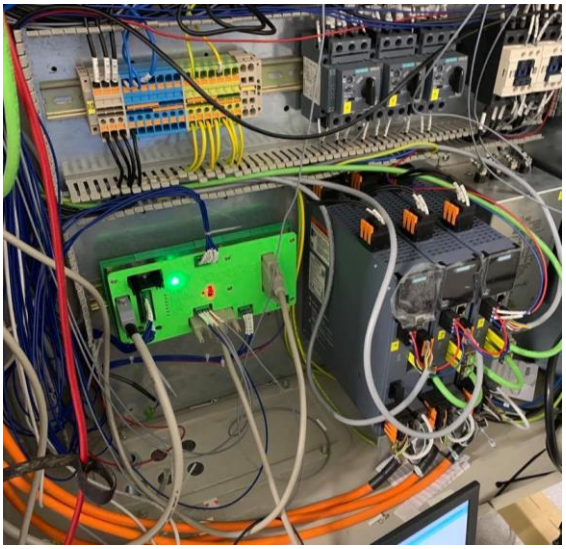
2.3.3 Arduino jako komunikační člen pro PLC

Společnost OPTIXS, s.r.o., která se zabývá komplexními dodávkami laserové a přístrojové techniky pro materiálové zpracování, vysoce přesnou metrologii a výzkum, úspěšně implementovala Arduino do řídícího systému přístroje ILS 600 viz obr. 6, od společnosti Innolas. Přístroj slouží pro průmyslové mikro gravírování materiálů při výrobě elektronických součástek. [9]

Obrázek 6 Přístroj ILS 600 [9]

Společnost OPTIXS, s.r.o. v tomto projektu řešila výměnu laserové hlavy tohoto přístroje. Během přípravy projektu bylo zjištěno, že výrobce laseru dodává již pouze nové modely, které nejsou zpětně kompatibilní s původním PLC. Výměna laserové hlavy včetně PLC by představovala téměř 100% zvýšení nákladu na servis. Po provedení studie proveditelnosti se společnost rozhodla vyřešit kompatibilitu nové laserové hlavy a starého PLC pomocí desky Arduino vlastní výroby, které zde bude sloužit jako komunikační mezičlen. [9]

PLC v tomto případě řídí obsluhu pohonů, interlocků a posílá data pro řízení laseru a jeho intervalové spouštění. Celkem se jedná o 4 diskrétní signály. Arduino tyto signály přijímá a následně na základě jejich vyhodnocení ovládá laserovou hlavu pomocí jednoho analogového výstupu. [9]



Obrázek 7 Zástavba Arduina do rozvaděče [9]

V projektu byla využita deska Arduino, které byla navržena společností OPTIXS, s.r.o. přímo pro tyto účely viz obr. 7. Deska obsahuje veškerou podpůrnou elektroniku a průmyslové konektory. Pro instalaci do prostoru rozvaděče byla navržena jedinečná instalační krabička s úchytem na lištu DIN. Ovládací program Arduina byl vytvořen společností OPTIXS, s.r.o. Ačkoliv použití Arduina vyžadovalo poměrně složitý návrh hardwarové a softwarové části, bylo oproti pořízení nového PLC ušetřeno cca 70 % nákladů. V tomto případě se sice jedná o kusový vývoj, nicméně využitelný pro budoucí projekty. [9]

2.4 Průmyslové platformy postavené na deskách Arduino

2.4.1 Industrial shields

Společnost Industrial shields nabízí řadu produktů postavených na různých variantách Arduina nebo Raspberry PI. Jedná se o první zařízení založené na technologii Arduino určené pro průmyslové použití. Průmyslová řada Arduino M-DUINO viz obr. 8, nabízí možnost rozšíření až 127 modulů prostřednictvím sběrnice I^2C, to dohromady umožňuje mít k dispozici až 7 100 vstupů / výstupů, každý modul je v plastovém pouzdře s konektory s montáží na DIN lištu. Toto „*PLC*“ Arduino lze programovat pomocí standardního prostředí Arduino IDE. Nahrávání softwaru je možné přes rozhraní USB nebo ethernet. Zařízení rovněž nabízí vzdálenou správu a nepřetržité sledování stavů všech proměnných, vstupů, výstupů atd. Přístroj je kompatibilní se zařízeními Ardbox a Touchberry Pi. [10]

Obrázek 8 Industrial shields Arduino M-DUINO [10]

2.4.2 Controllino

Dalším průmyslovým zařízením postaveným na platformě Arduino je projekt Controllino. Zařízení se prezentuje jako PLC s otevřeným softwarem. Controllino na rozdíl od společnosti Industrial shields, nabízí pouze jeden produkt ve dvou variantách, rozdělených podle počtu vstupů a výstupů. V nejvyšší variantě viz obr. 9, obsahuje 10 vstupů (analog / digital), 2 digitální vstupy, 12 digitálních PWM výstupů s možnou zátěží až 2 A a napěťovou hladinou 5–24 V, 10 reléových výstupů pro maximální napětí 250 V a proud 16 A. Dále disponuje rozhraními I^2C, SPI, ethernet a RS485, má také vlastní obvod reálného času. Produkty Controllino jsou navrhované s ohledem na průmyslové využití, proto jsou dodávány v plastovém pouzdře určeném pro montáž na DIN lištu. [11]

Obrázek 9 Controllino MAXI [11]

2.5 Průmyslové nasazení desek Arduino

Platforma Arduino nabízí řešení od hardwaru po cloud a IoT. Přístup s otevřeným zdrojovým kódem umožňuje kontrolovat způsob připojení zařízení ke cloudu. Pro možnosti cloudu je možné využít profesionální služby Arduino IoT Cloud, nebo se připojit ke cloudovým službám třetích stran, jako např. Amazon Web Services, Google Cloud Platform nebo Microsoft Azure. Výhody knihoven Arduino včetně podpory HTTP, MQTT, X.509 a JSON, umožňují připojení k jakékoli webové službě, která je upřednostňována uživatelem. [6]

Desky Arduino lze téměř libovolně modifikovat s pomocí široké nabídky shieldů. Tak je možné rozšířit desku Arduino např. o IoT, Ethernet, řízení pro motory, senzory nebo čtečku SD karet.

Díky těmto vlastnostem si desky postavené na platformě Arduino nalezly široké uplatnění i v průmyslovém využití. Tato kapitola uvádí příklady průmyslového využití desek Arduino.

2.5.1 Monitorování bezpečnostních prvků pro výtahy

Společnost ESCM Manufacturing vyvíjí monitorovací systémy pro bezpečnostní prvky výtahů a v roce 2020 úspěšně využila Arduino pro vytvoření monitoru bezpečnostních prvků výtahů, který vyhovuje platným předpisům.

Výtahy mají oddělené napájecí přívody, které zahrnují šachtu, kabinu a strojovnu. Jedním z největších problémů při instalaci a opravách výtahů je odstraňování problémů s elektrickými bezpečnostními prvky. Technik musí provést demontáž krytu a následné měření, tento postup je nutné opakovat pro každý bezpečnostní prvek a neúměrně se tak prodlužuje čas potřebný pro provedení opravy. [12]

Obrázek 10 Monitor stavu bezpečnostních prvků[12]

Monitorovací systém bezpečnostních prvků ESCM2000 viz obr. 10, je založený na Arduinu Nano, produkt implementuje desku mikrokontroleru a kontrolní desku vlastní výroby. Stav kontaktu je signalizován pomocí dvou diod, červená blikající kontrolka představuje otevřený kontakt a zelená svítící dioda signalizuje uzavřený kontakt. Tyto stavové kontrolky poskytují technikovi přehled o stavu bezpečnostního prvku bez nutnosti demontáže krytu a dodatečného měření viz obr. 11. ESCM2000 umožňuje vzdálený dohled pomocí sběrnice RS485 nebo CAN. Adresování zařízení umožňuje DIP přepínač s rozsahem adres 00–99. [12]

Obrázek 11 ESCM v praxi [12]

2.5.2 Fluid Eye – monitor kvality oleje

Společnost Fluid Intelligence prosazuje efektivnější nakládání s oleji. Olej je často měněn předčasně pouze dle nastavených intervalů, olej by však měl být ideálně měněn na konci své životnosti. Realizace tohoto úkolu potřebuje zařízení, které dokáže monitorovat parametry oleje nejlépe v reálném čase. [13]

Obrázek 12 Fluid Eye [13]

Fluid Eye viz obr. 12, je instalován přímo do olejové vany viz obr. 13, a monitoruje parametry oleje v reálném čase. Získává data o kvalitě a teplotě oleje, poskytuje pohled na mikroskopickou čistotu oleje (velikost, počet a typ částic nečistot) a provádí syntézu a korelační analýzu dat v reálném čase. [13]

Obrázek 13 Nasazení FluidEye v praxi [13]

Zařízení je připojeno ke cloudové službě Fluid, pomocí algoritmů strojového vidění a učení provádí potřebné analýzy a poskytuje přehled měsíčních a ročních reportů. Provoz zařízení je zajištěn deskou Arduino MKR, přičemž se v těchto drsných prostředích ukazuje jako snadno použitelná a dobře fungující. Pro komunikaci s cloudem využívá běžné síťové protokoly spolu s mobilními sítěmi v závislosti na typu a umístění zařízení. [13]

2.5.3 TerraSmart – zemědělství 4.0

TerraSmart je italský startup, který navrhuje a vyvíjí propojené systémy a řešení pro zemědělství založené na inteligentních senzorech a IoT technologiích.

Hlavním cílem zemědělců je zvýšit výnosy plodin a současně snížit spotřebu vody a chemikálií. Prostřednictvím aktivního monitorování rostlin a prostředí, může toto řešení umožnit farmářům pěstovat zdravější a udržitelnější plodiny. Konečným cílem je snížit provozní náklady a nabídnout vysoce kvalitní produkty za velmi konkurenceschopnou cenu. [14]

Obrázek 14 TerraSmart [14]

Spolehlivost systémů nasazených v terénu je technickou výzvou, protože musí fungovat s nepřetržitým provozem ve vzdálených a drsných podmínkách. [14]

Řešení je založeno na digitálních a analogových senzorech viz obr. 14, které shromažďují data týkající se stavu půdy, rostlin (např. teplota, úroveň vlhkosti půdy atd.) a místního podnebí (např. vlhkost vzduchu, úroveň slunečního záření, rychlost a směr větru atd.). Obsluhu těchto senzorů zajišťuje Arduino MKR, které umožňuje vzájemnou komunikaci mezi senzory pomocí sběrnice Modbus a portálem terraWeb. TerraWeb je online portál, který umožňuje zemědělcům sledovat jejich plodiny v reálném čase za účelem optimalizace jejich produktivity. Portál přijímá data ze senzorů terraSense přes uzly Arduino, monitorování probíhá v reálném čase. Pro připojení ke službě terraWeb používá zařízení mobilní síť. [14]

3 Zhodnocení možností Arduino v průmyslové praxi

Arduino je obecně jedním z nejpopulárnějších mikrokontrolerů a může se pochlubit velkou online komunitou, která vytvořila tisíce projektů. Arduino poskytuje širokou možnost využití, ať už je to monitorování parametrů, nebo řízení robota, proto našlo cestu do mnoha edukačních a hobby aplikací.

Díky variabilitě nabízených desek a procesorů, nízké ceně a široké komunitě se Arduino stále více prosazuje do průmyslové praxe. V průmyslové praxi jsou pro automatizaci nejběžněji používané zařízení PLC. Obvykle jsou to robustní přístroje vyrobené pro přímé použití ve výrobních závodech. PLC se liší od ostatních výpočetních zařízení právě tím, že jsou určeny pro náročné podmínky ve výrobních provozech. To znamená, že zvládnou prach, vysokou nebo naopak nízkou teplotu a vlhkost. Pro tyto podmínky není Arduino vyrobeno ani navrženo. PLC mají také větší variabilitu pro připojení senzorů a akčních členů. Tato variabilita spočívá v možnostech zpracování širšího spektra vstupních a výstupních signálů, než je možné u platformy Arduino, které dokáže pracovat pouze v rozmezí 0-5 V pro analogové vstupy a analogové výstupy jsou pouze v režimu PWM. V některých projektech může být problémem i licence Arduina. Arduino spadá do open source pod licenci LGPL, to znamená, že je Arduino možné komerčně užívat za předpokladu uvolnění kódu pro veřejnost. Tato povinnost může některé subjekty odrazovat od nasazení Arduina v průmyslových projektech. [1, 2, 4]

I když jsou PLC hlavním nástrojem pro automatizaci v průmyslovém odvětví, neznamená to, že Arduino nemůže být pro tento sektor užitečné. Příklady z předchozích kapitol ukázaly, že je Arduino výborným řešením pro doplnění PLC, kdy se stará o sběr dat, nebo jako komunikační mezičlen. [7, 8, 9, 12, 13, 14]

Od nasazení Arduina v průmyslových projektech jsou společnosti odrazovány také malou možností ladit kód. To se sice zlepšuje s nástupem 32bitových procesorů, ale průmyslové mikrokontrolery často přicházejí s rozsáhlým vývojovým prostředím. To většinou obsahuje možnost ladit program a množství testovaných knihoven. U knihoven určených pro Arduino se může stát, že daná knihovna nebyla nikdy testována, případně je jejich vývoj roky opuštěn. [15]

Problémem v průmyslovém nasazení může být také budoucí dostupnost použitých desek, snadno se může stát, že typ desky Arduino použitý v projektu, nebude za pár let dostupný a v případě servisu bude nutné sáhnout po jiné verzi, pro kterou ale může být potřebné upravit software nebo okolní hardware. [15]

I přes výše popsané nedostatky platformy Arduino přibývá v poslední době mnoho průmyslových řešení postavených přímo na platformě Arduino. Tento trend je silně podpořen příchodem IoT, kde možnosti Arduina převyšují možnosti PLC. A to především díky velké flexibilitě programování s dostupností velkého množství knihoven schopných obsluhovat většinu známých komunikačních protokolů. Tento trend podpořila sama platforma Arduino vydáním desek MKR a Portenta H7, které jsou navrhovány pro vysoký výkon a možnosti IoT. Společnost také spustila profesionální cloudovou službu Arduino IoT Cloud, která ještě více usnadňuje použití Arduina v profesionálních projektech. Díky modulárnosti celé platformy umožňuje Arduino komunikaci s většinou bezdrátových a drátových standardů. To společně s nízkými pořizovacími náklady a volnou rukou programátora posouvá využití Arduina stále více do průmyslové praxe. [6]

Zajímavá jsou i řešení, která vytváří průmyslové jednotky typu PLC na platformě Arduino. Tyto řešení spojují odolnost PLC a variabilitu Arduina a jsou doplněny o užitečné členy, jako jsou reléové výstupy nebo PWM řízení vyšších proudových zátěží. [10, 11]

4 Návrh vlastní případové studie – propojení Arduino a stavebnice Fishertechnik

4.1 Zadání

Cílem vlastní případové studie je demonstrovat možnosti propojení průmyslového modelu Fishertechnik s mikrokontrolerem Arduino. Zadání projektu bylo vytvořit komunikační program pro PC, který bude komunikovat se stavebnicí Fishertechnik pomocí Arduina, které bude v roli překladače mezi sériovou linkou a sběrnicí I^2C.

Průmyslový model využitý v této práci, je Fishertechnik Sorter. Jedná se o model robotického ramene, které třídí puky podle přečtené barvy do tří úložných boxů. Tento model je vybaven řídící jednotkou RoboPro TXT. Celý popis je uveden v kap. 4.2.1.

Součástí projektu je vytvoření programových vrstev použitých zařízení a provedení jejich fyzického propojení. Pro programovou výbavu PC byly zvoleny jazyky C# a WPF. Program výsledného řešení musí v reálném čase vyčítat definované hodnoty z průmyslového modelu Fishertechnik, tyto hodnoty opatřit časovým razítkem a zaznamenat je do textového souboru. Program dále musí umožňovat spuštění manuálního režimu a uživatelský výběr úložného boxu. Níže je uveden soupis požadovaných výstupů a vstupů.

Výstupy z modelu Fishertechnik

- Robot ve výchozí pozici
- Zaznamenaný puk na vstupu
- Spuštění činnosti robota
- Detekovaná barva

Vstupy do modelu Fishertechnik

- Manuální režim
- Manuální založení puku do zvoleného boxu

4.2 Použité komponenty

4.2.1 Průmyslový model Fishertechnik

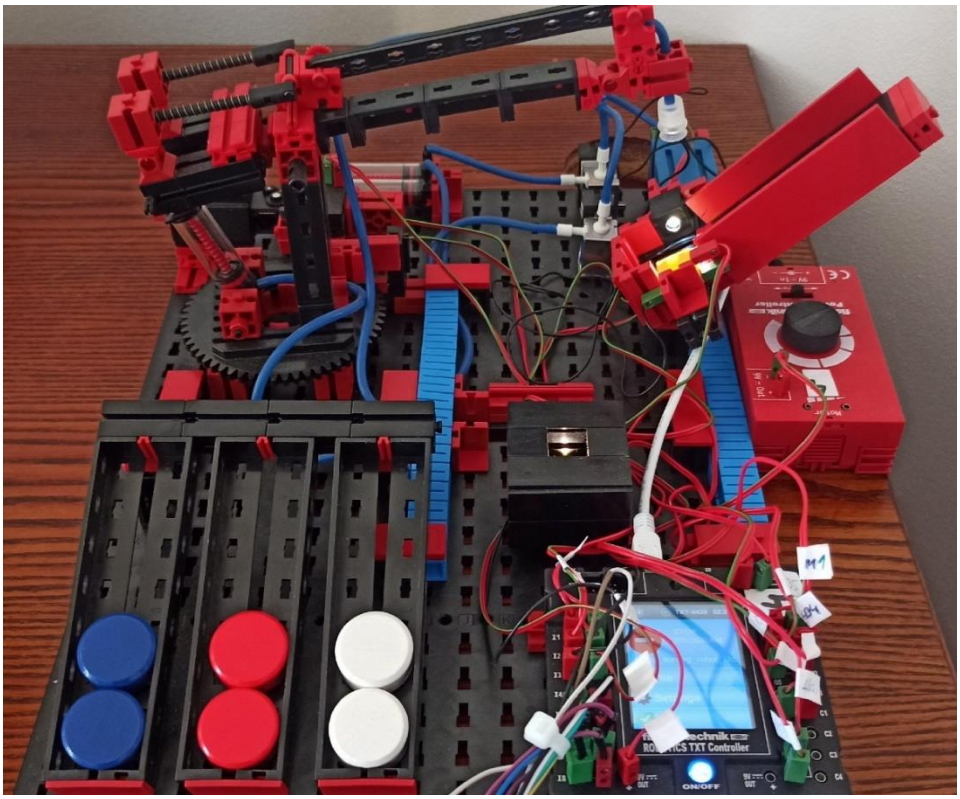
Fishertechnik STEM Engineering je studijní demonstrační stavebnice průmyslových provozů, která obsahuje komplexní přehled robotických, kódovacích a automatizačních systémů. Stavebnice se programuje pomocí objektového programovacího prostředí ROBO Pro. [17]

V projektu použitý průmyslový model Fishertechnik viz obr. 15, byl zapůjčen katedrou průmyslového inženýrství a managementu. Jedná se o model robotického ramene s vakuovým efektorem, vstupním zásobníkem s opto závorou, čidlem RGB barev, třemi úložnými boxy, pneumatickým systémem a řídící jednotkou RoboPro TXT.

Jednotka RoboPro TXT je tvořená dvoujádrovým 32bitový procesorem ARM Cortex A8 s rychlostí 500 MHz. Je vybavena pamětí RAM o velikosti 128 MB a 64 MB pamětí typu flash pro ukládání dat. Tato paměť je rozšiřitelná pomocí SD karty. Na přední straně jednotky RoboPro je umístěný barevný dotykový displej s rozlišením 320x240, 16 vstupů a výstupů (8 univerzálních vstupů, 4 čítací vstupy a 4 výstupy pro motor nebo LED). Rozšířená konektivita jednotky umožňuje komunikaci s vnějším světem přes rozhraní USB, infraport, wifi nebo Bluetooth. [25]

V základním režimu pracuje robot tak, že při vložení barevného puku do zásobníku řídící jednotka detekuje přerušení optické závory a spustí robotické rameno. To uchopí puk, přiloží ho na čidlo barvy, tam dojde k odečtení analogové hodnoty reprezentující jednu ze tří barev. Na základě těchto dat vydá řídící jednotka povel k uložení do boxu přiřazeného k barvě.

Pohyb robota je realizován pomocí motoru s mechanickým odečtem polohy. Pneumatický systém je pak složen z kompresoru, elektro ventilů a soustavy pístů pro vytvoření zdroje vakua.

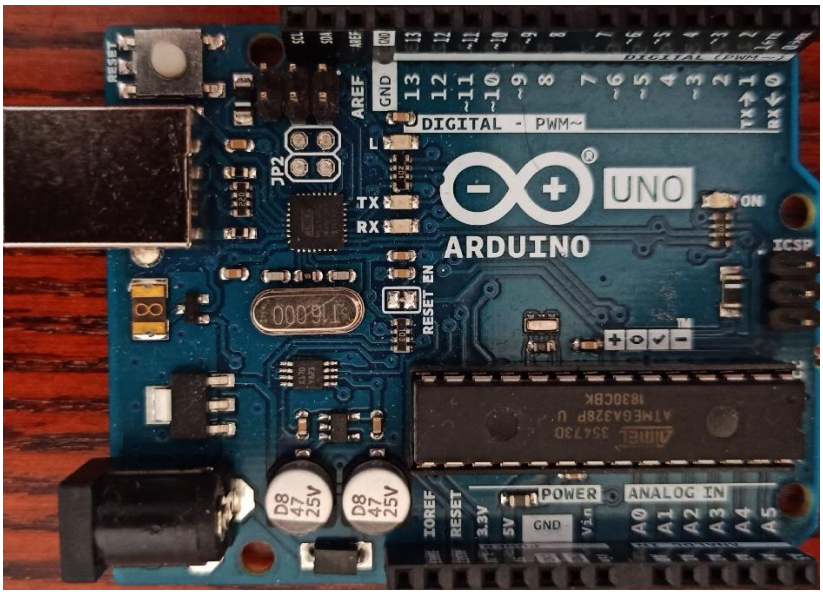


Obrázek 15 Průmyslový model Fishertechnik

4.2.2 Arduino Uno

Pro účel řešeného projektu byla použita originální deska Arduino Uno viz obr. 16. Tato verze byla zvolena s ohledem na to, že se jedná o jednu z nejvíce používaných desek. Řešení v této práci bude fungovat i na neoriginálních klonech Arduina, nicméně může docházet k neočekávanému chování.

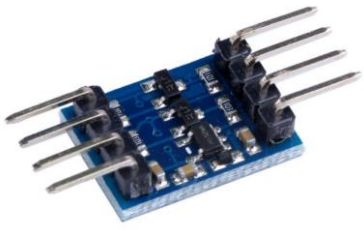
Deska je založená na čipu ATmega328. Má 14 digitálních vstupních / výstupních pinů (6 lze použít jako PWM), 6 analogových vstupů, 16 MHz oscilátor, připojení USB, DC napájecí konektor, konektor ICSP a resetovací tlačítko. Uno bylo první verzí desky Arduino pro USB připojení, která byla vydaná společně s vývojovým prostředím Arduino IDE 1.0. [2]



Obrázek 16 Arduino UNO

4.2.3 Převodník napěťových úrovní

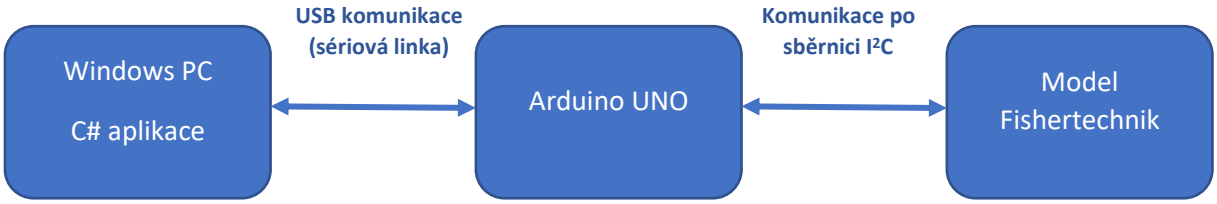
Kvůli rozdílným napěťovým hladinám I^2C sběrnic na Arduinu a RoboPro, je nutné použít převodník napěťových úrovní 3,3 V a 5 V viz obr. 17. Jedná se o klíčový prvek komunikace I^2C, a bez jeho použití by mohlo dojít k poškození jednotky RoboPro TXT. Pro sběrnici I^2C musí být použitý obousměrný převodník, jinak dané řešení nebude fungovat.



Obrázek 17 Převodník napěťových úrovní [24]

4.3 Návrh komunikačního rozhraní

V projektu jsou použity dva typy komunikačního rozhraní a mikrokontroler Arduino mezi nimi slouží jako překladač viz obr. 18. Toto řešení rozšiřuje možnosti použití stavebnice Fishertechnik s počítačem a zároveň nabízí uživatelsky přívětivější možnost konfigurace tohoto propojení.

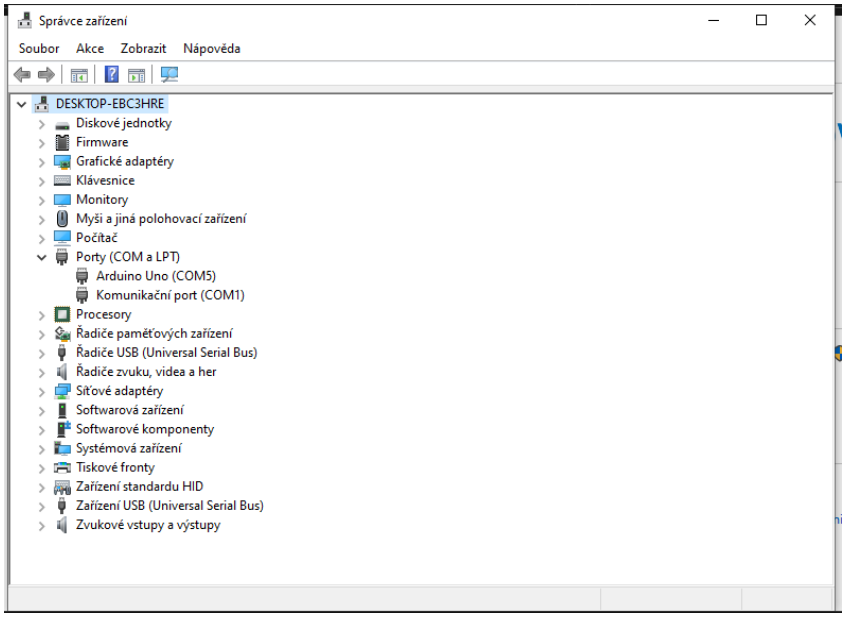


Obrázek 18 Komunikační schéma případové studie

4.3.1 USB sériová linka

Sériová komunikace, jak už název napovídá, posílá data po 1 bitu v řadě za sebou. Sériová linka je dvouvodičová, jeden vodič je přijímací a druhý vysílací. Díky tomu dokáže sběrnice současně číst i zapisovat pomocí dvou vláken programu, kdy jedno vlákno zajišťuje čtení a druhé odesílání dat. Pro správnou funkci sériové komunikace musí mít obě zařízení nastavenou shodnou rychlost baud (jednotka modulační rychlosti přenosu dat v asynchronním komunikačním kanále) [22]. Ta zajišťuje synchronizaci přenosu a integritu přenášených dat. Jelikož je standard RS-232 výhradou již spíše v průmyslu a u osobních počítačů byl nahrazen univerzálním sériovým portem (USB), je v tomto projektu použitá komunikace pomocí virtuálního COM portu. Rychlosti sériové linky dosahují až 115200 baud, se vzrůstající rychlostí ale může docházet k vyšší chybovosti. Pro většinu projektů je dostačující rychlost 9600 baudů, ta je použita i v řešeném projektu. [18]

USB virtuální COM port slouží pro připojení desky Arduino UNO k osobnímu počítači. Čip FTDI, který zprostředkovává tento port je implementován přímo na desce Arduino, ta se jen připojí k PC pomocí portu USB. Po nastavení správného COM portu v komunikačním rozhraní, pak přes tento port probíhá obousměrná sériová komunikace počítače a mikrokontroleru Arduino. Virtuální COM port s připojeným mikrokontrolerem Arduino je možné zjistit ve správci zařízení daného PC viz obr. 19.



Obrázek 19 Zjištění správného COM portu ve správci zařízení

4.3.2 I^2C

Sběrnice I^2C byla vyvinuta společností Philips v 80. letech minulého století. Jedná se o dvou vodičovou sběrnici, která umožňuje jednoduchou komunikaci mezi jednotlivými komponenty. Jedná se o *multi-master* sběrnici, to znamená, že sběrnice dokáže pracovat s více master jednotkami a má mechanismy, které zajišťují, aby nedocházelo ke kolizím a ztrátám dat. [21]

Název I^2C vzešel ze slova Inter IC, sběrnice se tak často označuje jako IIC nebo I^2C. Základní rychlost komunikace je 100 kbit, tato rychlost je použita i v řešeném projektu. Sběrnice nicméně umožňuje i takzvaný „FASTMODE“ s rychlostí 400 kbit a od roku 1998 je dokonce dostupná rychlost 3,4 Mbit. [21]

Sběrnice se často používá i pro komunikaci s jednotlivými periferními zařízeními, jako jsou různé snímače, displeje a jiné typy komponent. Zařízení zapojená do komunikační linky jsou adresovatelná a mohou fungovat jako *master,* nebo *slave*. Zařízení *master* se stará o časování sběrnice pomocí vodiče CLK, odpadá tak nastavování rychlosti baud. *Slave* kontinuálně přijímá data od *master* jednotky, odesílá je však pouze pokud je dotázán. [21]

Pro komunikace využívá sběrnice dvou vodičů. Jeden vodič používá pro synchronizaci (SCL) a druhý pro data (SDA). Sběrnice funguje na napěťových úrovních 5 V nebo 3,3 V. Jelikož zařízení použitá v tomto projektu používají odlišné napěťové úrovně, je v práci použit převodník napěťových úrovní pro I^2C, viz kap. 4.2.3.

5 Realizace vlastní případové studie

Návrh řešení spočíval v otestování komunikačních rozhraní poskytnutých Ing. Miroslavem Malagou. V první řadě byla otestována obousměrná komunikace mezi počítačem a Arduinem. V rámci testování byly ověřeny možnosti odesílání a přijímaní dat na obou zařízeních. Testovací řešení spočívalo v Arduinu, potenciometru a LED diodě. Aktuální nastavení hodnoty potenciometru se přenášelo do zkušební aplikace v C#, naopak z aplikace bylo možné ovládat diodu připojenou k Arduinu. [18]

Další krok spočíval v ověření komunikace Arduina a RoboPro přes I^2C sběrnici. K tomuto řešení bylo využito rozhraní napsané Ing. Miroslavem Malagou. Zkušební zapojení zůstalo stejné jako v předchozím případě, tedy Arduino s potenciometrem a LED diodou. Testovací program zobrazoval na vyžádání hodnotu potenciometru na displeji jednotky RoboPro TXT. A z displeje jednotky RoboPro TXT bylo možné ovládat LED diodu připojenou k Arduinu. [16]

V posledním kroku bylo nutné tyto testovací rozhraní spojit do jednoho testovacího celku. Zkušební model se sestával z Arduina propojeného s jednotkou RoboPro, k té byla připojena dioda a optická závora. Zkušebním úkolem bylo vytvoření programů pro možnost zobrazování stavu optické brány v PC a z programu také ovládat LED diodu připojenou k jednotce RoboPro. [16, 18]

Během realizace posledního zkušebního modelu byla zjištěna omezující vlastnost jednotky RoboPro. Jednotka RoboPro je defaultně nastavena jako master a používá interně uloženou I^2C adresu, kterou není možné změnit, a ačkoliv je sběrnice I^2C *multi-masterová*, nedokáže jednotka RoboPro spolupracovat s další *master* jednotkou připojenou na sběrnici. V rámci zkušebního řešení bylo testováno několik variant, ale řešení bylo funkční pouze v režimu RoboPro jako *master* a Arduino jako *slave*.

Toto omezení se vztahuje na jednotku RoboPro naprogramovanou pomocí vývojového prostředí RoboPro. Je pravděpodobné, že s použitím pokročilejšího programování, pomocí jazyka C++, by bylo možné toto omezení odstranit.

S tímto omezením je možné realizovat odesílání informací z RoboPro do Arduina, v opačném směru to ale lze pouze na vyžádání master jednotky, to znamená, že RoboPro nejprve musí Arduino o tyto data požádat. Arduino není schopné samovolně poslat příkaz do jednotky RoboPro tak, aby ho zaznamenala. V rámci řešení bylo nutné vymyslet způsob, jak tuto situaci vyřešit.

První varianta byla naprogramování jednotky RoboPro pro vyčítání dat z I^2C sběrnice v určitých intervalech. Tato varianta se ukázala jako nevhodná, jelikož pokud si RoboPro vyžádalo data v době, kdy nebyla dostupná, tak došlo k zablokování celé sběrnice. To vyplývá z procesu požadavku na zaslání dat. RoboPro v tu chvíli zasílá následující sekvenci informací [26]:

1. START bit
2. Adresa slave
3. Požadavek Read
4. Čeká na odpověď
5. STOP bit

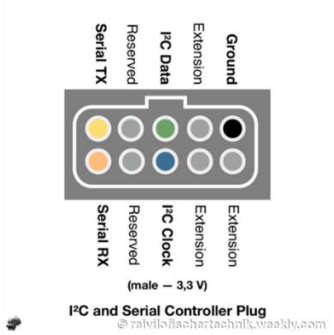
Pokud nedostane odpověď, tak zůstává linka otevřená a blokuje komunikaci. [26]

Další varianta spočívala v určení definovaných intervalů, kdyby současně RoboPro ověřovalo změnu dostupných dat a Arduino mělo připravenou jejich poslední verzi. Toto řešení vykazovalo vysokou chybovost, problémy se synchronizací intervalů a zvýšené reakční prodlevy, proto bylo vyřazeno jako nevyhovující.

Poslední varianta spočívala v použití digitálních I/O (vstupně/výstupních) pinů pro základní komunikaci mezi jednotkami. Použití této varianty spočívalo v úpravě programu RoboPro tak, aby v určitých krocích kontrolovalo stavy určených digitálních vstupů a podle toho řídilo svůj program. Použití I^2C sběrnice se omezilo na odchozí data z jednotky RoboPro.

Vzhledem k jiné napěťové hladině I/O pinů bylo nutné ověřit, zda RoboPro bude detekovat 5 V jako stav HIGH, jelikož standardně používá 9 V. V rámci zkoušky bylo ověřeno, že tato varianta je funkční a RoboPro spolehlivě detekuje 5 V jako stav HIGH.

Nyní bylo možné zkompletovat celé řešení dohromady. Jednotka průmyslového modelu Fishertechnik byla připojena pomocí I^2C převodníku k Arduinu pomocí rozšiřujícího rozhraní EXT viz obr. 20. Pro dodatečnou komunikaci s modelem byly použity 4 I/O digitální piny viz tab. 1.



Obrázek 20 Rozšiřující rozhraní jednotky RoboPro [23]

Tabulka 1 Přiřazení doplňkových pinů

Funkce	Arduino pin	RoboPro pin
Změna režimu	9	I8
Spuštění sekvence	10	I7
Adresa zvoleného boxu	11	I6
	12	I5

Po fyzickém zapojení všech částí dohromady bylo dalším krokem vytvoření řídících programů na všechny použité platformy tak, aby spolu v reálném čase komunikovaly a plnily funkce definované v zadání.

5.1 Software RoboPro

Výchozí program pro model stavebnice Fishertechnik byl poskytnut Ing. Miroslavem Malagou. Program je složen ze základních funkčních bloků a vnořených sekvencí. Vnořené sekvence jsou použity pro komplexnější funkční celky, jako je pohyb robota, ovládání efektoru, nebo identifikace barvy a jsou v programu označeny zeleným podbarvením.

V programu bylo nutné provést úpravy pro potřeby I^2C komunikace. Do zájmových částí kódu byly vloženy bloky zápisu na sběrnici I^2C. RoboPro tak vždy v požadovaném místě odešle stavovou informaci. Ta je vyjádřena unikátním číslem, které odpovídá různým stavům uvedeným v tab. 2. Tyto zápisové bloky obsahují veškerou nezbytnou konfiguraci pro odeslání dat po I^2C a konfigurují se nezávisle.

Tabulka 2 Přiřazení znaků pro jednotlivé stavy

Znak	Stav
1	Puk detekován
2	Zahájení pohybu ramene
3	Detekována bílá
4	Detekována červená
5	Detekována modrá
6	Uložení do boxu
7	Výchozí pozice – konec
8	Vstupní pozice je prázdná

Příchozí požadavky – příkazy z PC je v tomto řešení nutné zpracovávat pomocí čtení stavu digitálních vstupů. Program RoboPro ve výchozím stavu kontroluje stav vstupu I8 pro určení automatického nebo manuálního režimu. Pokud je v automatickém režimu, tak při detekci puku provede jeho zatřídění na základě změřené barvy. Pokud je ale tento vstup ve stavu HIGH, tak program místo kontroly optické závory začne kontrolovat stav digitálního vstupu I7, který spouští manuální zatřídění do zvoleného boxu. Pokud je na vstupu I7 detekován stav HIGH, vyčte RoboPro stav vstupů I5 a I6 a na základě jejich kombinace provede zatřídění do zvoleného boxu.

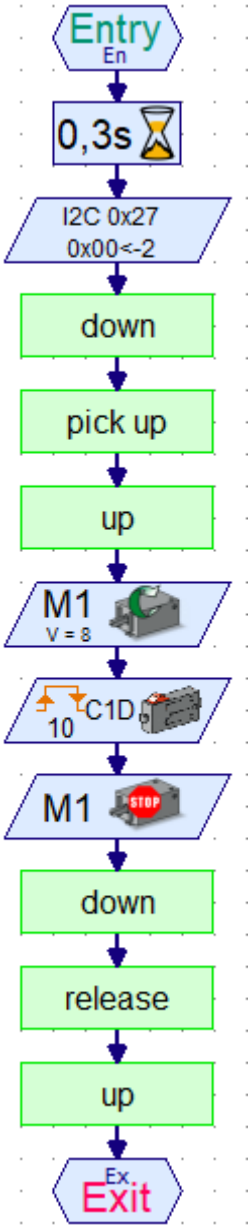
Pokud se v té době nenachází puk na vstupní pozici, program zatřídění neprovede a zašle chybový kód pomocí sběrnice I^2C, ten následně zpracuje Arduino pod stavem „Vstupní pozice je prázdná“.

Určení požadovaného boxu probíhá pomocí komparátoru na vstupech I5 a I6 a nabývá tří hodnot viz tab. 3:

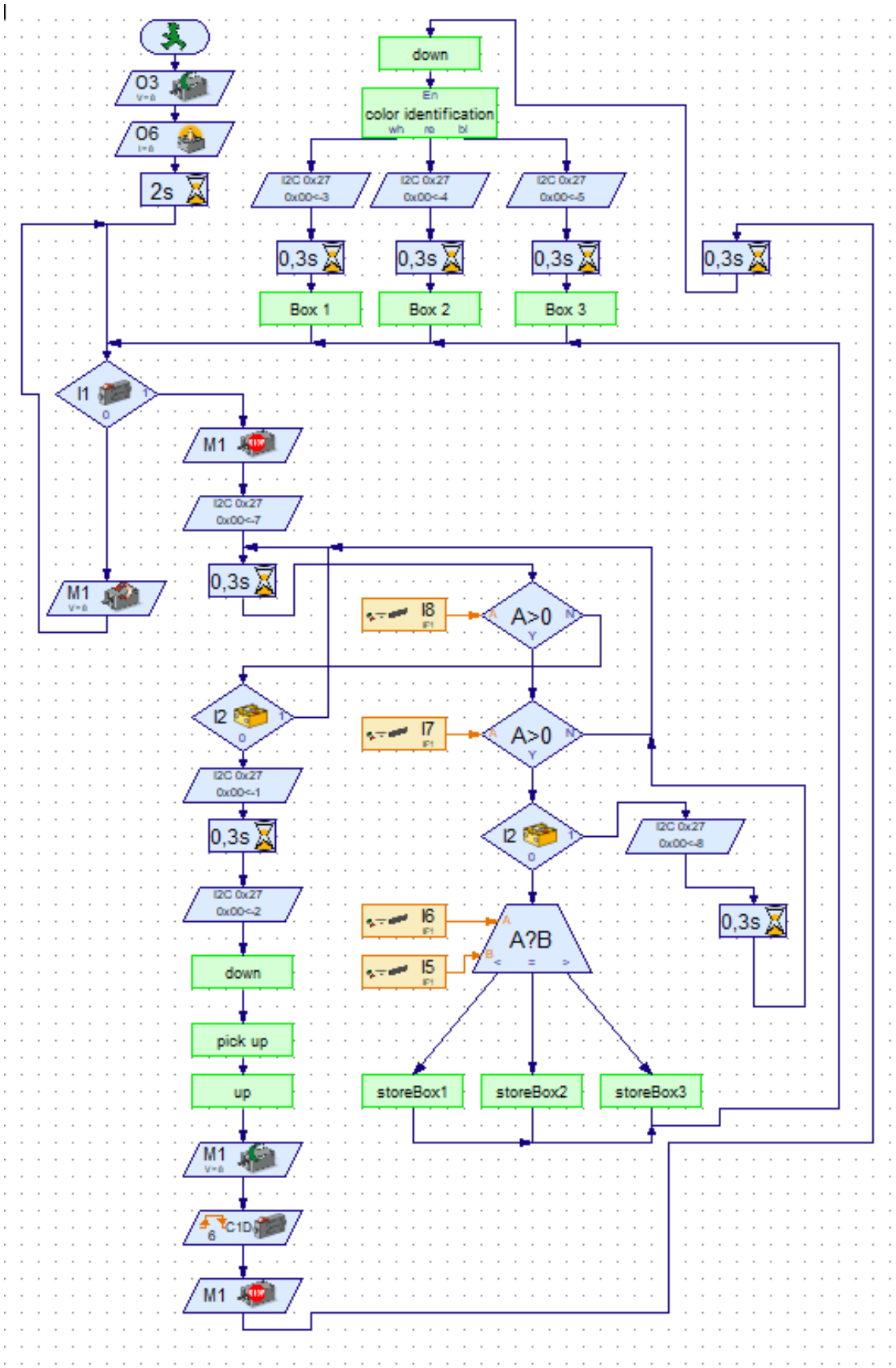
Tabulka 3 Rozhodovací podmínky pro určení zvoleného boxu

I5 > I6	Box 1 (modrá)
I5 = I6	Box 2 (červená)
I5 < I6	Box 3 (bílá)

Pro založení do zvoleného boxu byla vytvořena nová pohybová sekvence robota pro každý box. Její ukázka je uvedena na obr. 21, jednotlivé sekvence se navzájem liší počtem impulsů na čítacím pinu *C1D*. V úvodu funkce je uveden příklad funkce I^2C, která odesílá informaci o zahájení pohybu robota a je použita v různých částech kódu. V každé části však odesílá rozdílnou hodnotu. Blokové zobrazení celého programu RoboPro je uvedeno na obr. 22.



Obrázek 21 Příklad nově vytvořené sekvence



Obrázek 22 Vytvořený program RoboPro

5.2 Software Arduino

Úkolem Arduina je v tomto projektu překládat komunikaci mezi sériovou linkou a sběrnicí I^2C. Pro I^2C sběrnici je Arduino nakonfigurováno jako slave a zpracuje každou přijatou informaci ze strany master jednotky, tu přeloží do *string* a odešle na sériovou linku. Pokud Arduino detekuje příjem bitu na vstupním zásobníku sériové linky, vyčte vstupní *string* a provede vybrané části kódu.

Pro práci se sérovou linkou není nutné využívat externí knihovny, metody pro ovládání sériové linky jsou součástí základního kódu. Otevření sériového portu probíhá pomocí metody *Serial.begin(),* hodnota v závorce uvádí zvolenou rychlost v baudech. Tato metoda je volána funkci *setup()*, která je vykonána vždy pouze jednou při startu programu. V další části kódu je pak možné s připojením pracovat pomocí sady metod, které jsou k tomu určené. Základní metody pro práci se sériovou linkou jsou:

Serial.avalible()	Vyčte počet bajtů dostupných na sériové lince, tyto data jsou v příchozím zásobníku
Serial.print()	Odeslání dat bez zalomení řádku
Serial.println()	Odeslání dat se zalomením řádku
Serial.readString()	Načtení příchozích dat ve formě string

Komunikace se sběrnicí I^2C funguje trochu odlišně. Aby bylo možné pracovat se sběrnicí I^2C, je nutné použít v programu knihovnu *wire.h,* ta je definována na začátku kódu společně s proměnnými použitými v programu viz obr. 23. Proměnné využívané kódem jsou popsány v tab. 4. V části *setup()* se provede základní nastavení rozhraní. Metoda *Wire.begin()* iniciuje knihovnu a připojení k I^2C sběrnici, v závorce je uvedena adresa, které zařízení využívá pro komunikaci na sběrnici. Metody *Wire.onReceive()* a *Wire.onRequest* zaregistrují funkce, které se vyvolají při požadavku na příjem, nebo odeslání dat na sběrnici. Základní metody knihovny *Wire.h* jsou:

Wire.send()	odesílá po sběrnici data v několika typech (value, string, data). Použití v režimu *slave* umožňuje odeslat data na vyžádání jednotkou *master*. Pro odeslání dat z jednotky *master* je nutné volat tuto metodu mezi metodami *beginTransmission()* a *endTransmission().*
Wire.read()	vyčítá přijatá data z I^2C

```
#include <Wire.h>
const byte myAddress = 0x27;
int fromF = 0;
String pcRead;
const int enable = 9;
const int a = 10;
const int b = 11;
const int apply = 12;
```

Obrázek 23 Definice použité knihovny a proměnných

Tabulka 4 Seznam proměnných použitých v programu RoboPro

Proměnná	Význam
myAddress	Adresa zařízení (konstanta)
fromF	Proměnná pro ukládání přijatých dat z RoboPro
pcRead	Proměnná pro ukládání dat přijmutích z PC
enable	Pin volby režimu (konstanta)
a	Pin volby boxu (konstanta)
b	Pin volby boxu (konstanta)
apply	Pin pro spuštění sekvence (konstanta)
numBytes	Přijímaná data ve funkci receiveEvent

Celý program je složen z několika bloků, které jsou popsány níže.

V části kódu setup viz obr. 24, je definováno nastavení digitálních pinů, rychlost baud pro sériové spojení a část kódu pro obsluhu sběrnice I^2C. V té je definováno zahájení I^2C komunikace s přidělenou adresou, a také funkce, která se spouští v případě příchozích dat na sběrnici I^2C. [27] Na konci této části je pak výchozí nastavení výstupních pinů *enable* a *apply* do pozice *LOW*. To zaručuje, že po restartu zařízení bude RoboPro vykonávat činnost v automatickém režimu.

```
void setup() {
pinMode(enable, OUTPUT);
pinMode(a, OUTPUT);
pinMode(b, OUTPUT);
pinMode(apply, OUTPUT);
Serial.begin(9600);
Wire.begin(myAddress);
Wire.onReceive(receiveEvent);
Wire.onRequest(requestEvent);
digitalWrite(enable, LOW);
digitalWrite(apply, LOW);
}
```

Obrázek 24 Funkce setup v kódu Arduino

V hlavní smyčce programu viz obr. 25, probíhá kontrola příchozích dat na sériové lince. Pokud program detekuje příchozí bit, provede načtení celého příchozího *string* pomocí funkce *Serial.readString()*. [28] Podle vyčteného *string* se následně provede nastavení konfigurace výstupních pinů a odeslání potvrzující zprávy do PC. V tab. 5 je uveden přehled používaných stringů. Před deaktivací potvrzujícího pinu *apply* je nastavena prodleva 0,5 s tak, aby byl zaručen prostor pro zaregistrování změny jednotkou RoboPro.

Tabulka 5 Seznam použitých proměnných string a jejích význam

String	Význam
0	Manualni trideni vypnuto
1	Manualni trideni zapnuto
Box1	Zvolen Box 1
Box2	Zvolen Box 2
Box3	Zvolen Box 3

```
void loop() {
    while (Serial.available() > 0) {
    pcRead = Serial.readString();
    if (pcRead == "1")
    {
     Serial.println("Manualni trideni zapnuto");
     digitalWrite(enable, HIGH);
    }
    if (pcRead == "0") {
      Serial.println("Manualni trideni vypnuto");
     digitalWrite(enable, LOW);
    }
    if (pcRead == "Box1") {
      Serial.println("Zvolen Box1");
    digitalWrite(enable, HIGH);
    digitalWrite(a, LOW);
    digitalWrite(b, HIGH);
    digitalWrite(apply, HIGH);
    delay(500);
    digitalWrite(apply, LOW);
    }
    if (pcRead == "Box2") {
      Serial.println("Zvolen Box2");
     digitalWrite(enable, HIGH);
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(apply, HIGH);
     delay(500);
    digitalWrite(apply, LOW);
    }
    if (pcRead == "Box3") {
      Serial.println("Zvolen Box3");
      digitalWrite(enable, HIGH);
    digitalWrite(a, HIGH);
    digitalWrite(b, LOW);
    digitalWrite(apply, HIGH);
     delay(500);
    digitalWrite(apply, LOW);
    }
  }
}
```

Obrázek 25 Funkce loop v kódu Arduino

Funkce *receiveEvent* viz obr. 26, která byla definovaná v části setup, se stará o zpracovaní příchozích dat ze sběrnice I^2C. Při zjištění příchozích dat provede jejich uložení do proměnné *fromF* a pomocí prvku switch odešle odpovídající zprávu na sériovou linku. Přiřazení je uvedeno v tab. 2.

```
void receiveEvent(int numBytes) {
  while (Wire.available() > 0) {
    fromF = Wire.read();
    switch (fromF) {
      case 1:
        Serial.println("Puk detekovan");
        break;
      case 2:
        Serial.println("Zahajeni pohybu ramene");
        break;
      case 3:
        Serial.println("Detekovana bila");
        break;
      case 4:
        Serial.println("Detekovana cervena");
        break;
      case 5:
        Serial.println("Detekovana modra");
        break;
      case 6:
        Serial.println("Ulozeni do boxu");
        break;
      case 7:
        Serial.println("Vychozi pozice - konec");
        break;
      case 8:
        Serial.println("Vstupni pozice je prazdna!");
        break;
    }
  }
}
```

Obrázek 26 Funkce receiveEvent v kódu Arduino

5.3 Aplikace pro počítač

Aplikace byla naprogramována pomocí programovacích jazyků C# a WPF. Jazyk WPF byl použitý pro návrh a vytvoření grafického rozhraní aplikace, jazyk C# byl použitý pro obslužný kód.

Na obr. 27 je zobrazen kód grafického rozhraní v XAML. Za povšimnutí stojí použití metody *GetPortNames* jako zdroje dat pro prvek *ComboBox* určený pro výběr COM portů. Zmíněná metoda vrátí aktuálně dostupné COM porty v daném PC. [19]

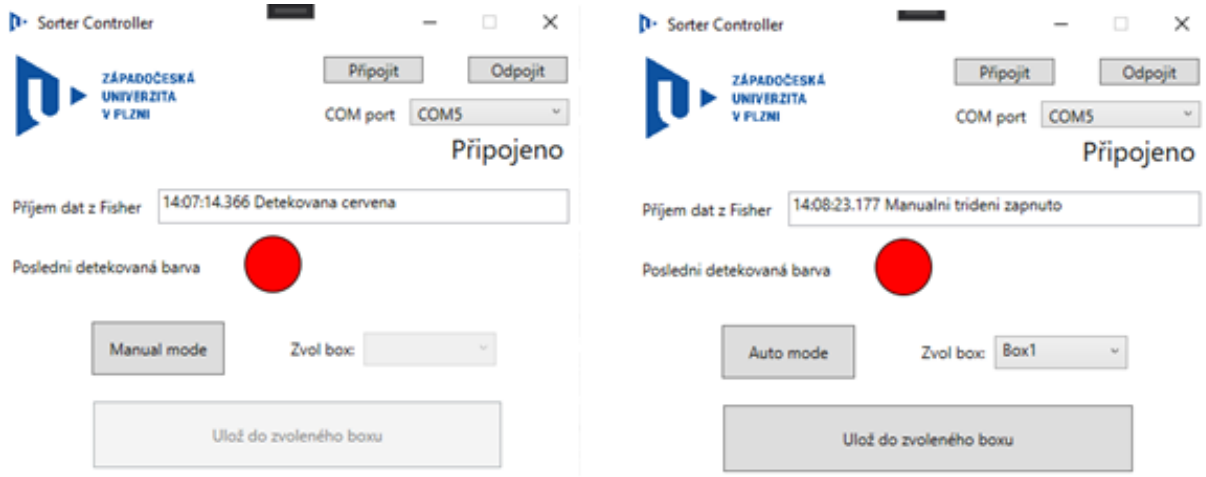
```
<Window x:Class="SorterController.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:ports="clr-namespace:System.IO.Ports;assembly=System"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:SorterController"
        mc:Ignorable="d"
        Title="Sorter Controller" Icon="WinIco.ico" Height="370" Width="455"
        WindowStyle="SingleBorderWindow" ResizeMode="CanMinimize">
    <Window.Resources>
        <ObjectDataProvider ObjectType="{x:Type ports:SerialPort}" MethodName="GetPortNames" x:Key="portNames"/>
    </Window.Resources>
    <Grid Margin="0,0,0,0">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="450"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="100"/>
            <RowDefinition Height="100"/>
            <RowDefinition Height="60"/>
            <RowDefinition Height="70"/>
        </Grid.RowDefinitions>

        <ComboBox x:Name="COM" ItemsSource="{Binding Source={StaticResource portNames}}"
                  HorizontalAlignment="Left" Margin="310,42,0,0" VerticalAlignment="Top" Width="120"
                  ToolTip="Zvolte COM" Grid.Row="0" Grid.Column="0" Height="20" />
        <TextBlock x:Name="status" FontSize="20" HorizontalAlignment="Left" Margin="341,65,0,0"
                   TextWrapping="Wrap" Text="Odpojeno" VerticalAlignment="Top" Height="28" Width="120"
                   Grid.Row="0" Grid.Column="0"/>
        <Button x:Name="connect" Content="Připojit" HorizontalAlignment="Left" Margin="245,10,0,0"
                VerticalAlignment="Top" Width="75" Click="connect_Click" Grid.Row="0" Grid.Column="0" Height="20"/>
        <Button x:Name="disconnect" Content="Odpojit" HorizontalAlignment="Left" Margin="354,10,0,0"
                VerticalAlignment="Top" Width="75" Click="disconnect_Click" Grid.Row="0" Grid.Column="0" Height="20"/>
        <Label Content="Příjem dat z Fisher" HorizontalAlignment="Left" VerticalAlignment="Top"
               Margin="5,10,0,0" Grid.Row="1" Grid.Column="0" Height="26" Width="107"/>
        <TextBox x:Name="fromF" HorizontalAlignment="Left" Height="26" Margin="120,10,0,0" TextWrapping= "NoWrap"
                 VerticalAlignment="Top" Width="310" Text="Přijem dat.."
                 VerticalContentAlignment="Center" Grid.Row="1" Grid.Column="0"/>
        <Ellipse x:Name="color"  Fill="Black" HorizontalAlignment="Left" Height="43" Margin="185,45,0,0"
                 Stroke="Black" VerticalAlignment="Top" Width="43" Grid.Row="1" Grid.Column="0"/>
        <Label Content="Posledni detekovaná barva" HorizontalAlignment="Left" VerticalAlignment="Top"
               Margin="5,55,120,0" Width="172" Height="24" Grid.Row="1" Grid.Column="0"/>
        <Button x:Name="manual" Content="Manual mode" HorizontalAlignment="Center" Margin="0,0,210,0"
                Width="100" Height="40" Click="manual_Click" Grid.Row="2" VerticalAlignment="Center" />

        <ComboBox Name="boxPick" SelectedValuePath="Content" IsEnabled="False"  HorizontalAlignment="Center"
                  Height="25" Margin="200,0,0,0" VerticalAlignment="Center" Width="100"
                  SelectionChanged="boxPick_SelectionChanged" Grid.Column="0" Grid.Row="2">
            <ComboBoxItem>Box1</ComboBoxItem>
            <ComboBoxItem>Box2</ComboBoxItem>
            <ComboBoxItem>Box3</ComboBoxItem>
        </ComboBox>

        <Label Content="Zvol box:" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="40,0,0,0"
               Grid.Row="2" Height="26" Width="60"/>
        <Button x:Name="send" Content="Ulož do zvoleného boxu" HorizontalAlignment="Center" Margin="0,0,0,0"
                VerticalAlignment="Center" Width="308" Height="50" IsEnabled="False" Click="send_Click" Grid.Row="3" Grid.Column="0"/>
        <Image HorizontalAlignment="Left" Height="65" Margin="12,8,0,0" VerticalAlignment="Top" Width="145"
               Source="zcu-logo.png" Grid.Column="0" Grid.Row="0"/>
        <Label Content="COM port" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="241,40,0,0"
               Grid.Column="0" Grid.Row="0" Height="26" Width="63"/>

    </Grid>
</Window>
```

Obrázek 27 XAML kód grafického rozhraní

Grafický návrh aplikace byl navržen s ohledem na funkčnost a uživatelskou orientaci. V horní pravé části se nachází komunikační rozhraní. Pro výběr COM portů je připraven *ComboBox* s automatický načítáním dostupných portů COM. Nad výběrem COM portu se nachází tlačítka pro připojení a odpojení vybraného COM portu. Status připojení je signalizován pod výběrem COM portu, program signalizuje stavy odpojeno a připojeno.

Pod tímto oddílem jsou umístěny prvky pro čtení dat a ovládání modelu. V poli příjem dat z RoboPro se vyčítají informace zasílané jednotkou RoboPro. Tyto informace jsou opatřeny časovým razítkem a v souladu se zadáním jsou ukládány do textového souboru *log.txt* v kořenovém adresáři projektu. Pod tímto polem se nachází grafické znázornění poslední detekované barvy. Posledním oddílem je oddíl s ovládacími prvky. Ve výchozím stavu je robot v automatickém režimu a z ovládacího oddílu je aktivní pouze tlačítko *Manual Mode*, ostatní položky jsou uzamčeny. Po stisku tlačítka *Manual mode* se robot a aplikace přepne do manuálního řízení. V tomto okamžiku jsou odemčeny i položky pro volbu boxu a tlačítko pro provedení založení. K volbě boxu je připraven další *ComboBox* s přednastavenými hodnotami. Tlačítko *Ulož do zvoleného boxu* slouží k provedení příkazu, před odesláním příkazu do jednotky RoboPro program ověří, zda je vybrán nějaký box, pokud by tomu tak nebylo, odeslání neprovede a zobrazí chybovou hlášku.

V programu byly ošetřeny výjimky v případě požadavku na akci, která není možná. Ukázka programu je na obr. 28, vlevo je zobrazen program s připojeným Arduinem v automatickém režimu, vpravo je pak zobrazení programu v manuálním režimu.



Obrázek 28 Ukázka navrhnuté aplikace Sorter Controller

Obslužný kód grafického rozhraní je napsaný v programovacím jazyku C#. Základním prvkem celého programu je práce se sériovou linkou. Aby bylo možné se sériovou linkou pracovat je nutné použít obor názvů *System.IO*. [20] Pro samotné vytvoření definice sériového portu je použita proměnná typu *SerialPort* s názvem *sp*. Ta je vytvořena pomocí třídy *SerialPort a* slouží pro práci s portem sériové linky v kódu programu. Jednotlivé úkony se sériovým portem jsou popsány v odpovídajících odstavcích funkcích, v kterých jsou použity. [20]

Používané obory názvů jsou uvedeny na začátku kódu viz obr. 29, dále začíná samotný kód programu *SorterController*, v kterém jsou definované proměnné používané v programu viz obr. 29.

```
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.IO.Ports;
using System.Windows.Threading;
using System.IO;

namespace SorterController
{
    public partial class MainWindow : Window
    {
        private delegate void UpdateUiTextDelegate(string text);
        SerialPort sp = new SerialPort();
        string boxPicked;
```

Obrázek 29 První část obslužného kódu

Struktura programu se dále skládá z jednotlivých funkcí, které jsou volány na základě nějaké situace. Tyto funkce jsou následně uvedeny.

O běh programu se stará obslužný kód napsaný v programovacím jazyku C#. Struktura kódu je rozdělena do několika funkcí. Níže jsou popsány hlavní funkce programu.

Funkce *MainWindow*

Jedná se o hlavní funkci programu viz obr. 30, která se stará o update grafického rozhraní a při příjmu dat po sériovém portu volá funkci *DataReceivedHandler*. Tato funkce probíhá ve smyčce od zapnutí programu do jeho ukončení. [20]

```
public MainWindow()
{
    InitializeComponent();
    sp.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);
}
```

Obrázek 30 Funkce MainWindow v kódu C#

Funkce *connect_Click*

Funkce *connect_Click* viz obr. 31, obsluhuje tlačítko *Připojit*, při stisku se zkusí připojit na zvolený COM port a nastaví *textBlock* status na hodnotu *Připojeno*. Pokud se jí nepovede připojit, zobrazí chybové okno.

V této funkci je definováno připojení k portu COM. Do proměnné *portName* se načítá položka načtená z *ComboBoxu* COM. Data pro tento ComboBox jsou načítány metodou *GetPortNames* v XAML kódu grafického rozhraní, ta se stará o automatické naplnění ComboBoxu dostupnými porty COM. Dále jsou definovány parametry pro připojení. Jedná se o *sp.PortName*, ten definuje zvolený port COM, vyčtený z proměnné *PortNames*, a *sp.BaudRate*, tedy přenosovou rychlost v baudech. Metoda *sp.Open* pak vytváří samotné připojení s vloženými parametry. Pokud se připojení nepovede, vznikne ošetřená výjimka, která zobrazí chybovou zprávu na obrazovce. [20]

```
private void connect_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string portName = COM.SelectedItem as string;
        sp.PortName = portName;
        sp.BaudRate = 9600;
        sp.Open();
        status.Text = "Připojeno";
    }
    catch (Exception)
    {
        MessageBox.Show("Zvolte platný COM port!");
    }
}
```

Obrázek 31 Funkce connect_Click v kódu C#

Funkce *DataReceiveHandler*

Tato funkce viz obr. 32, pokud je otevřený sériový port, vyčítá řádky příchozích dat po sériové lince. Tato data ukládá do proměnné typu *string RxString*, zároveň volá funkci *ProcessData*, které se stará o zpracování načteného řádku.

Funkce pracuje se sériovým portem pomocí proměnné *sp* a metod *Is.Open a ReadLine()*. Funkce na svém začátku pracuje s podmínkou *if*, která ověřuje stav sériového portu pomocí metody *sp.IsOpen*, pokud je podmínka splněna, spustí se metoda *sp.ReadLine*, ta uloží příchozí řádek do proměnné *RxString*. Metoda *Dispatcher.Invoke* následně spouští funkci *ProcessData* s parametrem *RxString* ve vedlejším synchronním vlákně. [20]

```
public void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
{
    if (sp.IsOpen)
    {
        string RxString = sp.ReadLine();
        Dispatcher.Invoke(DispatcherPriority.Send, new UpdateUiTextDelegate(ProcessData), RxString);
    }
}
```

Obrázek 32 Funkce DataReceiveHandler v kódu C#

Funkce *ProcessData*

Tato funkce viz obr. 33 se stará o kompletní zpracování načteného příchozího řádku. Funkce nejprve vytvoří časové razítko, to spojí s příchozím řádkem a provede uložení do textového souboru *log.txt*, umístěného v kořenovém adresáři programu, pokud by tento soubor neexistoval, automaticky jej vytvoří. Záznamy se přidávají vždy na konec souboru na nový řádek. Zbývající část kódu se stará o zobrazování detekované barvy. Každý příchozí řádek se prohledává na klíčová slova, pokud je některé z těchto slov detekováno, zobrazí se příslušná barva v grafickém prvku *color*. Zápis dat do souboru probíhá pomocí metody *File.AppendAllText()*. [29]

```
private void ProcessData(string RxString)
{
    DateTime timeStamp = DateTime.Now;
    string timeStr = timeStamp.ToString("HH:mm:ss.fff");
    string output = timeStr + " " + RxString;
    fromF.Text = output;
    File.AppendAllText("log.txt", output);

    string stringToCheck = RxString;
    if (stringToCheck.Contains("modra"))
    {
        color.Fill = Brushes.Blue;
    }
    if (stringToCheck.Contains("cervena"))
    {
        color.Fill = Brushes.Red;
    }
    if (stringToCheck.Contains("bila"))
    {
        color.Fill = Brushes.White;
    }
    if (stringToCheck.Contains("prazdna"))
    {
        MessageBox.Show("Vstupní pozice je prázdná!");
    }
}
```

Obrázek 33 Funkce ProcessData v kódu C#

Funkce *manual_Click*

Jedná se o obslužnou funkci pro tlačítko *manual mode* viz obr. 34. Toto tlačítko přepíná režim z automatického do manuálního a opačně. Při stisku tlačítka funkce nejprve zkontroluje aktuálně nastavený režim a následně provede změnu nastavení. Ta spočívá v odeslání informace do Arduina a přenastavení prvků v grafickém prostředí. Arduino po příjmu těchto dat odešle potvrzující zprávu o provedení změny. Tato zpráva je signalizována v příchozích datech a ukládaná do *log.txt*. Pokud se funkci nepodaří odeslat data po sériové lince, zobrazí chybovou zprávu.

Funkce pracuje se sériovým portem metodou *sp.Write()*. V uvozovkách v závorce jsou uvedena odesílaná data. Změny grafických prvků, jsou provedeny úpravou jednotlivých parametrů *Content* a *IsEnabled*.

```
private void manual_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if ((manual.Content as string) == "Manual mode")
        {
            sp.Write("1");
            manual.Content = "Auto mode";
            boxPick.IsEnabled = bool.Parse("True");
            send.IsEnabled = bool.Parse("True");
        }
        else
        {
            sp.Write("0");
            manual.Content = "Manual mode";
            send.IsEnabled = bool.Parse("False");
            boxPick.IsEnabled = bool.Parse("False");
        }
    }
    catch
    {
        MessageBox.Show("Nejprve se připojte!");
    }
}
```

Obrázek 34 Funkce manual_Click v kódu C#

Funkce *disconnect_Click*

Jedná se o funkci obsluhující tlačítko *Odpojit* viz obr. 35. Po stisknutí tlačítka se zavolaná funkce pokusí uzavřít sériový port, zároveň provede nastavení položky status na hodnotu *Odpojeno* a nastaví program do výchozích hodnot. Pokud se jí nepovede odpojit sériovou komunikaci, zobrazí chybovou zprávu.

Nejprve je metodou *sp.Write* odeslán požadavek na zapnutí automatického režimu, následuje uzavření sériového portu pomocí metody *sp.Close()*. Nastavení do výchozích hodnot probíhá úpravou parametrů u jednotlivých grafických prvků. [20]

```
private void disconnect_Click(object sender, RoutedEventArgs e)
{
    try
    {
        sp.Write("0");
        sp.Close();
        status.Text = "Odpojeno";
        manual.Content = "Manual mode";
        send.IsEnabled = bool.Parse("False");
        boxPick.IsEnabled = bool.Parse("False");
        fromF.Text = "-";
    }
    catch (Exception)
    {
        MessageBox.Show("Nejprve se připojte!");
    }
}
```

Obrázek 35 Funkce disconnect_Click v kódu C#

Funkce *send_Click*

Funkce viz obr. 36, obsluhuje tlačítko *Ulož do zvoleného boxu*, funkce odesílá pomocí metody *sp.Write()* do Arduina informaci o zvoleném boxu z příslušného *comboBox*. O načtení hodnoty do proměnné *boxPicked* se stará obslužná funkce prvku *boxPick*.

```
private void send_Click(object sender, RoutedEventArgs e)
{
    try
    {
        sp.Write(boxPicked);
    }
    catch
    {
        MessageBox.Show("Není vybrán box!");
    }
}
```

Obrázek 36 Funkce send_Click v kódu C#

Funkce *boxPick_SelectionChanged*

Jedná se o poslední funkci programu viz obr. 37. Ta se stará o obsluhu prvku *ComboBox* s názvem *boxPick*. Po zvolení požadovaného boxu v grafickém prostředí, je touto funkcí zvolená položka zapsána do proměnné *boxPicked* jako hodnota typu *string*.

```
private void boxPick_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    boxPicked = ((ComboBoxItem)boxPick.SelectedItem).Content.ToString();
}
```

Obrázek 37 Obslužné funkce comboBoxů v kódu C#

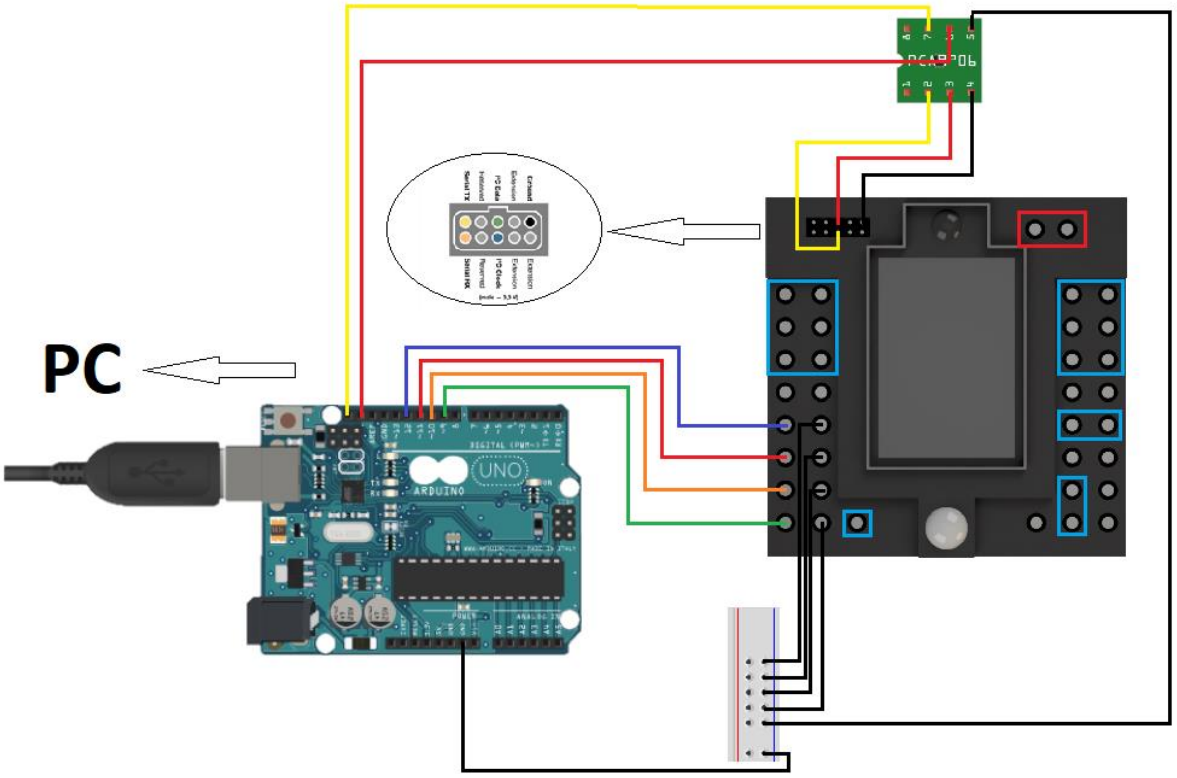
5.4 Zapojení

Tato podkapitola popisuje finální zapojení řešeného projektu. K propojení I²C byl použitý obousměrný převodník logických úrovní 5 V a 3,3V. Do převodníku jsou zapojeny výstupy SDA a SCL z mikrokontroleru Arduino z 5 V strany a z jednotky RoboPro TXT ze strany 3,3 V. Dále je nutné provést propojení společných záporných pólů, pro tento účel má převodník 2 piny GND: Záporné připojení GND je použito zároveň u spínaných logických stavů na úrovni propojení digitálních I/O pinů mezi jednotkami. Pro rozdělení záporného pólu byla použita napěťová část nepájivého pole. Zapojení převodníku již nevyžaduje zapojené piny VCC, jelikož má každá jednotka vlastní napájení. Napěťové strany jsou rozlišeny písmeny A (5 V) a B (3,3 V). Propojení jednotlivých pinů převodníku je uvedeno v tab. 6. Digitální piny Arduina a RoboPro jsou propojeny napřímo dle tab. 1.

Diagram zapojení finálního řešení je znázorněn na obr. 38. K vytvoření diagramu byla použita z části vlastní grafika a z části grafika z open source softwaru Fritzing [30]. Piny jednotky RoboPro použité pro připojení robota jsou označeny modrým rámečkem. Červeným rámečkem jsou označeny napájecí piny jednotky RoboPro. K napájení RoboPro je použitých 9 V zdroj Fishertechnik, Arduino je napájeno pomocí USB portu (5 V), kterým je zároveň připojeno k PC.

Tabulka 6 Zapojení I²C převodníku

Arduino	I²C převodník		RoboPro TXT
GND	AGND	BGND	GND
SDA	ASDA	BSDA	SDA
SCL	ASCL	BSCL	SCL
-	AVCC	BVCC	-



Obrázek 38 Diagram finálního zapojení

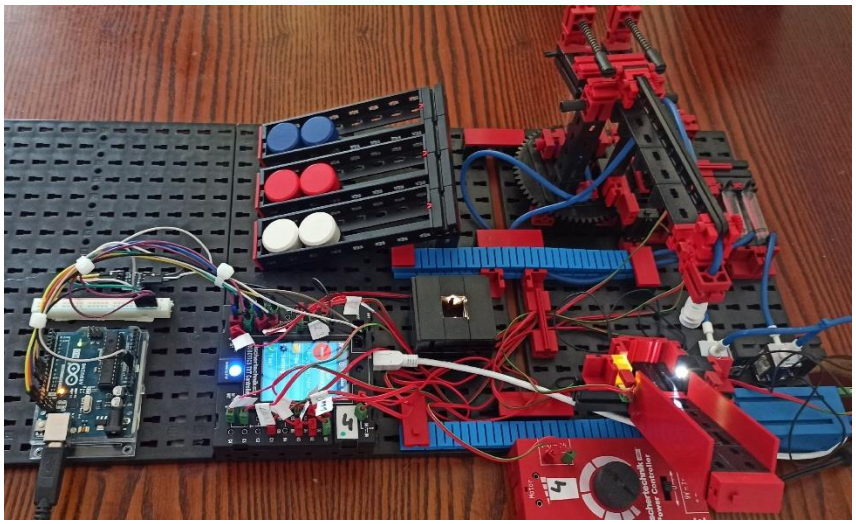
5.5 Popis funkce výsledného řešení

Po zapnutí průmyslového modelu Fishertechnik a Arduina je možné spustit program Sorter Controller. V programu se nejprve provede volba COM portu, na kterém je připojené Arduino, následně je možné připojit sériovou linku pomocí tlačítka *Připojit.* Úspěšné připojení je signalizováno změnou stavu pod *comboBox* pro výběr COM portů.

V tomto režimu čeká model robotického zakladače na vložení barevného puku do zásobníku. Při detekci puku provede model automatické zatřídění, tak jak je popsané v kap. 4.2.1. Tento režim probíhá zcela automaticky a odesílá požadované informace o jednotlivých stavech do PC, kde jsou tyto záznamy zapisovány do textového souboru *log.txt.*

Při stisknutí tlačítka *Manual mode,* dojde k zapnutí manuálního režimu. V tomto režimu přestane robot automaticky třídit a čeká na pokyny uživatele. Při obdržení pokynu na zatřídění, provede robot kontrolu, zdali je na vstupní pozici umístěn puk, pokud ano, provede jeho zatřídění, bez určování barvy, přímo do zvoleného boxu. Pokud robot při požadavku na manuální založení zjistí, že na vstupní pozici se nenachází puk, odešle chybovou hlášku, která se zobrazí uživateli programu na PC.

Pro ukončení práce slouží tlačítko *Odpojit*. To se postará o nastavení robota zpět do automatického režimu, odpojení sériové linky a výchozího nastavení prvků grafického rozhraní. Model celého zařízení je zobrazen na obr. 39.



Obrázek 39 Finální řešení projektu

6 Závěr a hodnocení

V teoretické části diplomové práce byl proveden průzkum využití platformy Arduino v průmyslové oblasti. Tato část obsahuje souhrn nejběžnějších desek Arduino vhodných pro využití v automatizaci a IoT. Možnosti Arduina jsou popsány v zajímavých projektech z průmyslové sféry, práce obsahuje posouzení vhodnosti Arduina pro průmyslové nasazení.

Praktická část diplomové práce se zabývá realizací případové studie propojení Arduina a průmyslového modelu FisherTechnik. V té bylo navrženo řešení, které splňuje všechny podmínky zadání. Arduino slouží jako překládací prvek mezi PC a modelem Fishertechnik. Během řešení byly otestovány možnosti a hranice použití I^2C komunikace s modelem RoboPro. Při návrhu funkčního řešení bylo nutné vyřešit několik problémů, popsaných v práci. Tyto problémy se podařilo vyřešit drobnými úpravami výsledného řešení.

Nejvýznamnější změnou je použití digitálních I/O pinů pro komunikaci ze strany Arduino do jednotky RoboPro. Ačkoliv se podařilo najít funkční řešení, jedná se stále o značně omezující problém. Řešení dodatečné komunikace pomocí digitálních pinů je omezeno celkovým počtem pinů na jednotce RoboPro. Pro úsporu I/O pinů byla v práci ověřena varianta použití PWM výstupu Arduina s analogovým vstupem RoboPro, ta se ovšem neosvědčila. RoboPro nedokáže spolehlivě vyčítat hodnoty PWM. Možným řešením by bylo použití vhodného D/A převodníku, tak by bylo možné přes jeden vstup posílat více úrovní rozlišitelných jako různé informace na straně jednotky RoboPro. Použití vhodného D/A převodníku nebylo v této práci řešeno, ale bude předmětem dalšího výzkumu.

Výsledné řešení projektu vykazuje vysokou spolehlivost provozu. Během testování se projevilo minimum problémů. Převážně se jednalo o problémy s komunikací na sběrnici I^2C, které se občas projevily během zapínání zařízení Arduino a modelu Fishertechnik. K odstranění problému stačilo provést reset mikrokontroleru Arduino. Jiné problémy se za dobu testování neprojevily. Všechny programy a zdrojové kódy, vytvořené v této práci, jsou přiloženy v digitální podobě na CD nosiči.

Výsledky této práce budou uplatněny nejen pro vzdělávací účely, ale mají využití při návrhu modelů digitálních podniků, či podobném nasazení přímo v průmyslu. Použití STEM modelu Fishertechnik a Arduina umožňuje využít jednoduchost sériové komunikace a data získávaná z modelu libovolně zpracovávat. To například usnadňuje propojení fyzického modelu Fishertechnik a virtuálního modelu např. v Plant Simulation.

Navrhnuté řešení spadá do konceptu STEM a představuje užitečné znalosti v oblasti průmyslového inženýrství. Výsledky této práce mohou být použity při optimalizaci výrobních, vzdělávacích nebo logistických procesů uvnitř podniku. Vzhledem k širokým možnostem používaných technologiích, může být práce dále rozvíjena a zlepšována. Další rozvoj může být například propojení fyzického průmyslového modelu a jeho virtuálního dvojčete, to by pak sloužilo jako vzdělávací pomůcka při školení a rozvoji zaměstnanců. Virtuální a fyzické modely mohou sloužit pro porovnání a rozhodování o volených technologických operacích. V průmyslových provozech je možné tyto výsledky využít pro sledování a optimalizaci provozních parametrů výrobních procesů.

Cíle této práce byly navrhnutým řešením splněny a práce může být použita jako výchozí model pro specifické řešení v průmyslu nebo jako podklad pro další vývoj v oblasti návrhu a prezentace digitálních podniků s modely Fishertechnik a mikrokontrolerů Arduino.

Použitá literatura

[1.] Arduino – Introduction. Arduino – Home [online]. [cit. 14-11-2020]. Dostupné z: https://www.arduino.cc/en/guide/introduction.

[2.] Arduino Official Store | Boards Shields Kits Accessories. Arduino Official Store | Boards Shields Kits Accessories [online]. [cit. 14-11-2020]. Dostupné z: https://store.arduino.cc/.

[3.] Voda, Z., Arduino – Průvodce světem Arduina 2.vydání, Nakladatelství Martin Stříž, Bučovice 2017, ISBN 978-80-87106-93-8

[4.] Monk, S., Programming Arduino: Getting Started with Sketches, Second Edition (Tab) 2nd Edition, McGraw-Hill Education 2016, ISBN 978-1259641633

[5.] Blum, J., Exploring Arduino: Tools and Techniques for Engineering Wizardry, Second Edition, John Wiley & Sons 2020, ISBN:978-111940537

[6.] Arduino Pro. Arduino – Home [online]. [cit. 14 11 2020]. Dostupné z: https://www.arduino.cc/pro/hardware.

[7.] RENEKER, D.: PLC vs. Arduino for industrial control. Control Design, 2017 [cit. 14-11-2020]. Dostupné z: https://www.controldesign.com/articles/2017/arduino-vs-plc-for-industrial-control/.

[8.] SRIVASTAVA, A.: Intelligent Transportation System In City Bus Shelter. Arduino Project Hub, 2017 [cit. 14-11-2020]. Dostupné z: https://create.arduino.cc/projecthub/technoid/intelligent-transportation-system-in-city-bus-shelter-cf076e?ref=tag&ref_id=industrial&offset=5.

[9.] Svoreň M.: Projektová dokumentace – Realizace řízení laserové hlavy nového typu. OPTIXS s.r.o. Praha, 2020.

[10.] Industrial Arduino PLC. The Arduino Industrial shield for projects. [online]. [cit. 14-11-2020]. Dostupné z: https://www.industrialshields.com/.

[11.] CONTROLLINO MAXI – CONTROLLINO. CONTROLLINO – 100 % Arduino compatible PLC | Industry-ready hardware [online]. [cit. 14-11-2020]. Dostupné z: https://www.controllino.com/product/controllino-maxi/.

[12.] ESCM MANUFACTURING INC: Monitoring safety systems for elevators with ESCM Manufacturing. [online]. [cit. 14-11-2020]. New York, 2020. Dostupné z https://www.arduino.cc/pro/case-studies/escm-manufacturing.

[13.] FLUID INTELLIGENCE: Monitoring oil performance in real-time with Fluid Intelligence. [online]. [cit. 14-11-2020]. Finsko, 2019. Dostupné z https://www.arduino.cc/pro/case-studies/fluid-intelligence.

[14.] TERRASMART: terraSmart – Agriculture of the Future. [online]. [cit. 14-11-2020]. Itálie, 2019. Dostupné z https://www.arduino.cc/pro/case-studies/terrasmart.

[15.] Is Arduino Used in Real Life Products? - The Robotics Back-End. The Robotics Back-End – Program Robots Like a Boss [online]. [cit. 14-11-2020]. Dostupné z: https://roboticsbackend.com/is-arduino-used-in-real-life-products/.

[16.] MALAGA, Miroslav, ULRYCH, Zdeněk. Základy řízení robotů pro strojní inženýrství. 1. vyd. Plzeň: Západočeská univerzita v Plzni, 2020. 145 s. ISBN 978-80-261-0486-5.

[17.] MALAGA, Miroslav a Zdeněk ULRYCH. Koncept STEM se zaměřením na problematiku Industry 4.0. In: Průmyslové inženýrství 2019: Mezinárodní studentská

vědecká konference [online]. Západočeská univerzita v Plzni, 2019. - [cit. 08-12-2020]. ISBN 9788026108948. Dostupné z: doi:10.24132/PI.2019.08948.094-100

[18.] Serial Communication - learn.sparkfun.com. *Learn at SparkFun Electronics - learn.sparkfun.com* [online]. [cit. 10-03-2021] Dostupné z: https://learn.sparkfun.com/tutorials/serial-communication/all

[19.] SerialPort.GetPortNames Metoda (System.IO.Ports) | Microsoft Docs. [online]. Copyright © Microsoft 2021 [cit. 20.03.2021]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/api/system.io.ports.serialport.getportnames?view=dotnet-plat-ext-5.0

[20.] Communicating With Serial Port In C#. *C# Corner - Community of Software and Data Developers* [online]. Copyright ©2021 C [cit. 06.04.2021]. Dostupné z: https://www.c-sharpcorner.com/uploadfile/eclipsed4utoo/communicating-with-serial-port-in-C-Sharp/

[21.] I2C - What's That? - I2C Bus. *I2C - What's That? - I2C Bus* [online]. [cit. 08-4-2021] Dostupné z: https://www.i2c-bus.org/

[22.] ŘÍHA, Petr. Slovník počítačové informatiky: výkladový slovník pro práci s informacemi : hardware a software včetně počítačových sítí, internetu a mobilních technologií. Ostrava: Montanex, 2002. Informační technologie. ISBN 80-7225-083-3. [cit. 08-04-2021]

[23.] How to Connect an I²C Device to the Robotics TXT? - Rei Vilo's fischertechnik Corner. *Rei Vilo's fischertechnik Corner - Home* [online]. [cit. 03-03-2021]. Dostupné z: https://reivilofischertechnik.weebly.com/how-to-connect-an-isup2c-device-to-the-robotics-txt.html

[24.] Převodník úrovní 5V - 3V | GM electronic, spol. s.r.o.. *GM electronic | elektronické součástky, komponenty . | GM electronic, spol. s.r.o.* [online]. [cit. 02-04-2021]. Dostupné z: https://www.gme.cz/prevodnik-urovni-5v-3-3v#

[25.] MALAGA, M., BROUM, T, ULRYCH, Zdeněk. Industry 4.0 basic knowledge transfer: 36th IBIMA Conference, 2020. 145 s. ISBN 978-0-9998551-5-7.[cit. 08-4-2021]

[26.] I2C tutorial. *Robot Electronics* [online]. [cit. 08-4-2021]. Dostupné z: https://www.robot-electronics.co.uk/i2c-tutorial

[27.] Arduino - Wire . *Arduino - Home* [online]. [cit. 12-4-2021]. Dostupné z: https://www.arduino.cc/en/reference/wire

[28.] Serial - Arduino Reference. Arduino - Home [online]. [cit. 06-4-2021] . Dostupné z: https://www.arduino.cc/reference/en/language/functions/communication/serial/

[29.] File.AppendAllText Metoda (System.IO) | Microsoft Docs. [online]. Copyright © Microsoft 2021 [cit. 12.04.2021]. [cit. 06-4-2021]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/api/system.io.file.appendalltext?view=net-5.0

[30.] Fritzing. *Fritzing* [online]. [cit. 12-4-2021]. Dostupné z: https://fritzing.org/

Seznam obrázků

Seznam tabulek