

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**Využití modulu MES Mitsubishi RD81MES96 pro řízení výrobního procesu
na základě požadavků z SQL databáze**

Marek Henzl

Bakalářská práce

2019

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne XX. XX. XXXX

Marek Henzl

Poděkování

Děkuji panu Ing. Jiřímu Klimešovi za metodické a ochotné vedení celou praktickou částí bakalářské práce a také děkuji za návrh a výtisk výrobků na 3D tiskárně. Dále musím poděkovat panu Ing. Danielu Honcovi Ph.D. za pomoc s teoretickou částí a pomoc s administrativními záležitostmi, které jsou spojeny s bakalářskou prací. Poděkovat musím i společnosti B:TECH a.s. za poskytnutí možnosti pracovat na této bakalářské práci a za poskytnutí potřebného vybavení. Poděkování patří také Petru Pavlíčkovi za předání cenných zkušeností v manipulaci s robotem a tvorbě programu pro robota. Dále děkuji Kryštofu Henzlovi za grafickou pomoc při změně anglických textů v ilustracích za texty v českém jazyce a pomoc při tvorbě ukázkového videa.

V Pardubicích dne XX.XX.XXXX

Podpis

Marek Henzl

ANOTACE

Bakalářská práce se zabývá tříděním dílů podle barvy v robotickém pracovišti. Tyto možnosti zajišťuje MES systém, který pracuje s databází, kde jsou uloženy informace o všech načtených dílech. MES systém díky informacím z databáze rozhodne, jakou barvu má díl. Databáze je vytvořena v MS SQL Server, PLC, MES modul i robot je značky Mitsubishi. Díly jsou rozpoznávány pomocí kamery, kdy se jednotlivé díly vyznačují různým značením. Kamera je značky Sick a rozhraní pro práci s kamerou je SOPAS Engineering Tool.

KLÍČOVÁ SLOVA

Průmysl 4.0, MES, robotické pracoviště, databáze, PLC

TITLE

Usage of Mitsubishi RD81MES96 MES module for manufacturing proces control depending on requests stored in SQL database

ANNOTATION

Bachelor work deals with sorting of parts, depending on their colour, in the robotic station. This option ensures MES system, which is working with database, where is saved information about all loaded parts. MES system with information from the database will decide how coloured is the part. Database is in MS SQL Server. PLC, MES module, robot and interfaces to these devices are from Mitsubishi. Parts are recognizing by camera, where different parts have different marking. Camera and interface for work with this camera is from company Sick.

KEYWORDS

Industry 4.0, MES, robotic work-station, database, PLC

OBSAH

	SEZNAM ILUSTRACÍ.....	7
	SEZNAM TABULEK	9
	SEZNAM ZKRATEK A ZNAČEK	10
	ÚVOD.....	11
1	TEORETICKÁ ČÁST	12
1.1	Průmysl 4.0	12
1.1.1	6 stavebních kamenů Průmyslu 4.0	13
1.1.2	„Chytrá“ továrna	13
1.1.3	Pojmy spojené s Průmyslem 4.0	14
1.2	Automatizační pyramida.....	15
1.3	MES	17
1.3.1	Základní funkcionality MES systémů	18
1.3.2	Výhody MES systému ve výrobě	20
1.4	PLC	20
1.4.1	Základní definice	20
1.4.2	Historie PLC	20
1.4.3	Typy PLC	21
1.4.4	Programování PLC	22
1.5	Robotika.....	27
1.5.1	Klasifikace robotických zařízení dle úrovně řízení	27
1.5.2	Pohybové struktury konvenčních robotů	28
1.5.3	Programování robota	33
1.5.4	Souřadnicové systémy robota	35
1.5.5	Dělení průmyslových robotů z hlediska bezpečnosti	38
1.6	Komunikační protokoly v průmyslu	38
1.6.1	Sériové protokoly	39
1.6.2	Ethernetové protokoly	40
1.7	Strojové vidění.....	41
1.7.1	Použití strojového vidění	42
1.7.2	Typy zařízení se strojovým viděním	43

2	PRAKTICKÁ ČÁST	47
2.1	Naplánování budoucí práce	47
2.2	Blokové schéma.....	49
2.3	Zásobníky	50
2.4	Díly do zásobníku	51
2.4.1	Příprava dílů	51
2.4.2	Značení dílů	52
2.5	Konfigurace síťových adaptérů	53
2.6	Kamera.....	53
2.6.1	Konfigurace kamery	54
2.6.2	Kontrola dílů.....	55
2.6.3	Parametry získané z kamery	57
2.7	Robot	57
2.7.1	Programy robota	59
2.8	PLC	62
2.8.1	Program	63
2.9	MES modul.....	63
2.9.1	Konfigurace modulu	64
2.9.2	Vytvoření „tagů“ zařízení	64
2.9.3	Nastavení tabulek z databáze.....	65
2.9.4	Nastavení činností MES modulu	66
2.10	Porovnání MES systémů	72
2.10.1	Architektury systémů.....	72
2.10.2	Načítání dat z PLC.....	73
2.10.3	Cena	74
2.10.4	Uživatelská přívětivost	74
2.10.5	Závěrečné porovnání	76
2.11	SQL databáze.....	76
2.11.1	Tabulka všech zápisů.....	77
2.11.2	Tabulky jednotlivých zásobníků.....	78
2.11.3	Vymazání tabulek pro zásobníky	78
	ZÁVĚR	79
	POUŽITÁ LITERATURA	80
	PŘÍLOHY	83

SEZNAM ILUSTRACÍ

Obrázek 1.1 – Cesta k Průmyslu 4.0 (Rojko, 2017)	12
Obrázek 1.2 – Automatizační pyramida (Cope, 2020)	16
Obrázek 1.3 – Sériové zapojení v Ladderu	24
Obrázek 1.4 – Paralelní zapojení v Ladderu	24
Obrázek 1.5 – Příklad kódu v Ladderu	24
Obrázek 1.6 – Příklad kódy v Grafsetu	25
Obrázek 1.7 – Příklad programu ve funkčním blokovém schématu (IDEC, 2017).....	26
Obrázek 1.8 – Schéma TTT struktury (TUKE, 2011)	29
Obrázek 1.9 – Pracovní prostor TTT struktury (TUKE, 2011)	29
Obrázek 1.10 – Robot s TTT strukturou (Endeavor Business Media, 2013)	29
Obrázek 1.11 – Pracovní prostor RTT struktury (TUKE, 2011)	30
Obrázek 1.12 – Schéma RTT struktury (TUKE, 2011)	30
Obrázek 1.13 – Robot s RTT strukturou (Cylindrical Robots, 2020).....	30
Obrázek 1.14 – Pracovní prostor RRT struktury (TUKE, 2011)	30
Obrázek 1.15 – Schéma RRT struktury (TUKE, 2011).....	30
Obrázek 1.16 – Robot s TTT strukturou (Endeavor Business Media, 2013)	31
Obrázek 1.17 – Pracovní prostor RRR struktury (TUKE, 2011).....	31
Obrázek 1.18 – Schéma RRR struktury (TUKE, 2011)	31
Obrázek 1.19 – Robot s RRR strukturou (Alibaba, 1999-2020)	32
Obrázek 1.20 – Schéma SCARA struktury (TUKE, 2011)	32
Obrázek 1.21 – Pracovní prostor SCARA struktury (TUKE, 2011)	32
Obrázek 1.22 – Robot se SCARA architekturou (EPSON, 2020).....	33
Obrázek 1.23 – Programování metodou <i>vézt přes</i> (Programming Robots, nedatováno).....	34
Obrázek 1.24 – Programování pomocí pendantu (Programming Robots, nedatováno)	35
Obrázek 1.25 – Osový souřadnicový systém (TUKE, 2011)	36
Obrázek 1.26 – Souřadnicové systémy (Coordinates of a robot systém, nedatováno).....	37
Obrázek 1.27 – Souřadnicový systém příruby (Shah, 2016)	37
Obrázek 1.28 – Rodina systémů zpracovávajících informace (Owen-Hill, 2016)	42
Obrázek 1.29 – Příklad 1D strojového vidění (Introduction to machine vision, 2020).....	43
Obrázek 1.30 – Princip postupného skenování (Introduction to machine vision, 2020).....	44
Obrázek 1.31 – Princip stereofonního vidění (Hariyama, 2008).....	45
Obrázek 1.32 – Princip ToF skenování (Rooms, 2019)	46

Obrázek 1.33 – Kontrola povrchu (Introduction to machine vision, 2020).....	46
Obrázek 2.1 – Vývojový diagram.....	48
Obrázek 2.2 – Blokové schéma	49
Obrázek 2.3 – Zdrojový zásobník s díly	50
Obrázek 2.4 – Úložné zásobníky	51
Obrázek 2.5 – Díl.....	51
Obrázek 2.6 – Označení dílů.....	52
Obrázek 2.7 – Štítky na kódové označení dílů	52
Obrázek 2.8 – Kamera VSPM-6F2113.....	53
Obrázek 2.9 – Referenční snímek.....	55
Obrázek 2.10 – Kontrola jednotlivých dílů	55
Obrázek 2.11 – Snímek dílu z kamery.....	56
Obrázek 2.12 – Výsledné parametry dílu z kamery.....	56
Obrázek 2.13 – Robot MELFA RV-4FRL s pendantem	58
Obrázek 2.14 – PLC R04CPU	62
Obrázek 2.15 – MES modul RD81MES96.....	64
Obrázek 2.16 – Nastavení komponent v "tagu".....	65
Obrázek 2.17 – Načtení tabulky z databáze do MES modulu	66
Obrázek 2.18 – Spuštění činnosti jednou událostí.....	67
Obrázek 2.19 – Spuštění činnosti dvěma událostmi.....	67
Obrázek 2.20 – Spuštění činnosti událostí událostmi v log. kombinaci OR	67
Obrázek 2.21 – Spuštění činnosti událostmi v log. kombinaci AND.....	67
Obrázek 2.22 – Spuštění činnosti událostí při splnění předpokladu.....	68
Obrázek 2.23 – Vytvoření spouštěcí podmínky pro zápis z tabulky všech zápisů do PLC.....	69
Obrázek 2.24 – Výchozí okno pro nastavení operací u dané činnosti.....	71
Obrázek 2.25 – Architektura MES systému při použití MES modulu	72
Obrázek 2.26 – Architektura MES systému při použití operátorského panelu	72
Obrázek 2.27 – Zjednodušení MES systémů (bez ERP)	73
Obrázek 2.28 – Ukázka nemožnosti načtení dat z PLC přes OP	73
Obrázek 2.29 – Tvorba činnosti pro OP	74
Obrázek 2.30 – Tvorba činnosti pro MES modul.....	75
Obrázek 2.31 – Přidání databázového serveru pro OP	75
Obrázek 2.32 – Přidání databázového serveru pro MES modul.....	76
Obrázek 2.33 – Databázový server	77

SEZNAM TABULEK

Tabulka 1.1 – Dělení PLC	21
Tabulka 1.2 – Dělení kinematických dvojic	28
Tabulka 2.1 – IP adresy jednotlivých komponent	53
Tabulka 2.2 – Důležité údaje o kameře	54
Tabulka 2.3 – Parametry získané z kamery	57
Tabulka 2.4 – Programy robota	59
Tabulka 2.5 – Čísla zásobníku pro jednotlivé barvy	61
Tabulka 2.6 – Použité "tagy"	65
Tabulka 2.7 – Vzorek z tabulky všech záznamů	78

SEZNAM ZKRATEK A ZNAČEK

1D	Jednodimenzionální
2D	Dvoudimenzionální
3D	Trojdimenzionální
BAC	Building automation and control (Automatizace a řízení budovy)
CAN	Controller area network (Ovladač oblastní sítě)
CPU	Central processing unit (Centrální procesorová jednotka)
DB	Database (Databáze)
ERP	Enterprise resource planning (Plánování podnikových zdrojů)
GUI	Graphical user interface (Grafické uživatelské rozhraní)
HMI	Human machine interface (Rozhraní člověk-stroj)
I/O	Input / Output (Vstup/Výstup)
ICT	Information and communication technology (Informační a komunikační technologie)
IoS	Internet of services (Internet služeb)
IoT	Internet of things (Internet věcí)
IP	Internet protocol (Internetový protokol)
MES	Manufacturing execution system (Výrobní prováděcí systém)
OEE	Overall equipment effectiveness (Celková efektivita výroby)
OP	Operační panel
PLC	Programmable logic controller (Programovatelný logický automat)
SCADA	Supervisory control and data acquisition (Kontrola superdozorce a získávání dat)
SQL	Structured query language (Strukturovaný dotazovací jazyk)
SW	Software
TCP	Transmission control protocol (Protokol řízení přenosu)
TUKE	Technická univerzita v Košiciach (Technická univerzita v Košicích)
USB	Universal serial bus (Univerzální sériová sběrnice)

ÚVOD

Druhé desetiletí 21. století si historie bude pamatovat jako dobu, kdy začala Průmyslová revoluce 4.0. Tato revoluce se promítá do všech oblastí našich životů. Většina lidí má čtvrtou průmyslovou revoluci spojenou s roboty, kteří nahrazují lidskou práci. Není to tak úplně pravda. Robotizace je spojená už se třetí průmyslovou revolucí, v nynějších letech je jen možné nahrazovat širší a širší spektrum činností, u čtvrté průmyslové revoluce jde o získávání informací, vzájemnou propojeností přístrojů, oddělení ale i celých firem. Pro čtvrtou průmyslovou revoluci je typické, že už se nevytváří věci na sklad, ale podle požadavků přímo pro trh.

Důvodem a motivací pro tuto bakalářskou práci je fakt, že je nutné se čtvrté průmyslové revoluci přizpůsobit a přijmout ji jako fakt a příležitost. Proto jsem se rozhodl pro bakalářskou práci, která se zabývá použitím MES systému ve výrobě. MES systém je sice známý delší dobu a nevznikl až s nynější průmyslovou revolucí, ale získává čím dál větší prostor a je to systém, který charakterizuje fakty se kterými je Průmyslová revoluce 4.0 spojena – využití informací a jejich analýza.

Tato bakalářská práce se zabývá možností využití MES systému. V případě této práce se využije možnosti komunikace s databází. Databáze slouží jak pro správné informování robota o dílu (na jakou pozici ho má uložit), tak pro následnou analýzu dílů.

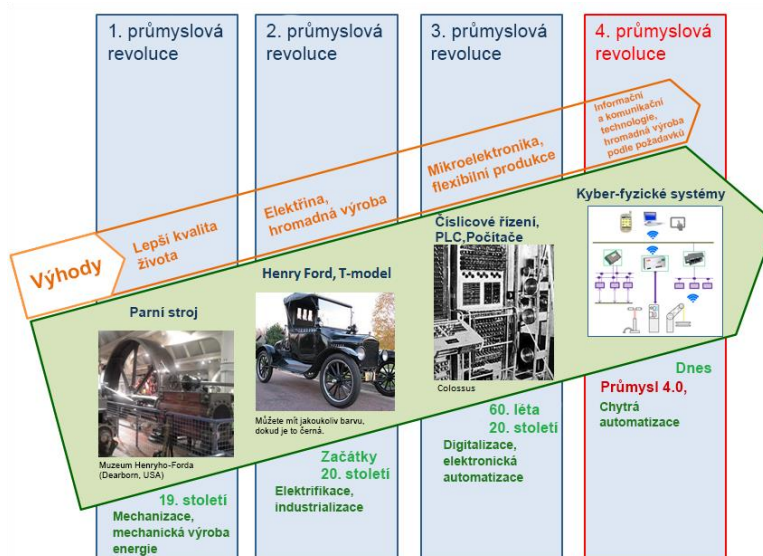
1 TEORETICKÁ ČÁST

1.1 Průmysl 4.0

Průmysl 4.0 je také někdy nazýván čtvrtou průmyslovou revolucí. Cesta k tomuto konceptu průmyslu vedla 200 let od první průmyslové revoluce, která charakterizovala 19. století. První průmyslová revoluce začala s mechanizací a mechanickou silou, která znamenala přechod od manuální práce k prvním průmyslovým procesům. Největším kladem této revoluce byla zlepšená kvalita života. Samozřejmě mezi zápory můžeme stavět začínající závislost na fosilních palivech a environmentální dopady.

Druhá průmyslová revoluce byla spuštěna postupnou elektrifikací, která dovoľovala industrializaci a masovou produkci. Druhá průmyslová revoluce je často spojována s výrokem Henryho Forda, který řekl o autě Ford T-Model: „Můžete mít jakoukoli barvu, pokud je to černá“. Tato věta skvěle charakterizuje možnost masové výroby, která je ale bez možnosti přizpůsobení produktů zákazníkům.

Třetí průmyslová revoluce je charakterizována digitalizací se zavedením mikroelektroniky a automatizace. Ve výrobě to znamenalo flexibilnější produkci, kde různé druhy produktů byly produkovány na flexibilních průmyslových linkách s programovatelnými výrobními stroji. Avšak tato revoluce stále neznamena flexibilitu týkající se množství produkce. Ta přišla až s průmyslem 4.0 (Rojko, 2017).



Obrázek 1.1 – Cesta k Průmyslu 4.0 (Rojko, 2017)

1.1.1 6 stavebních kamenů Průmyslu 4.0

- **interoperabilita** – schopnost propojení a komunikace mezi kyber-fyzickými systémy, lidmi a „inteligentními továrnami“ navzájem pomocí IoT a IoS;
- **virtualizace** – virtuální kopie „inteligentní“ továrny, která je propojená s úrovní senzorů (přímé monitorování výrobních procesů), s virtuálními a simulačními modely;
- **decentralizace** – schopnost kyber-fyzikálních systémů dělat nezávislé rozhodnutí,
- **real-time** – schopnost sběru a analýzy dat k řízení výroby v reálném čase,
- **orientace na služby** – nabídka služeb pomocí IoS,
- **modularita** - Flexibilní adaptace „inteligentní“ továrny na požadavky k změně jednotlivých modulů (Brynda, 2019).

1.1.2 „Chytrá“ továrna

Nejdůležitějším procesem v továrně, který ji posunuje do Průmyslu 4.0, je konverze digitálních dat na fyzické procesy v operativně konfigurovatelné výrobě. Takový systém je nejnovějším pokrokem ve vývoji výrobních systémů. Díky operativně konfigurovatelné výrobě je možná nejen výroba různých výrobků na té stejné lince, ale je možné i ovládat i množství vyrobených produktů, které závisí na dynamicky se měnícím trhu a požadavcích zákazníků.

Zařízení v Průmyslu 4.0 jsou kyber-fyzickým systémem. Jedná se o nějaký fyzický systém s integrovanými ICT komponenty. Systémy jsou tím pádem autonomní a zvládají se rozhodovat v závislosti na učebním algoritmu, reálných datech, analýze výsledků a záznamu úspěšnosti v minulosti.

Produkty v takové továrně jsou také „chytré“. Mají zabudované senzory, které jsou pomocí bezdrátového spojení používány pro sběr dat v reálném čase. Taková data jsou použita pro lokalizaci, měření stavu produktu a podmínek okolí. To znamená, že takové produkty mohou sami ovládat logistickou cestu produkcí (Rojko, 2017).

1.1.3 Pojmy spojené s Průmyslem 4.0

Cloud

Kvůli nutnosti okamžitého přístupu k datům a čím dál většimu objemu dat přestávají být fyzické disky atraktivní a dostačující. Pro ukládání dat začíná být čím dál více využíváno „cloudové“ ukládání dat. To znamená, že se data ukládají na databázové servery, přičemž jsou vytvořeny i zálohy těchto souborů. Výhodou je i okamžitý přístup k takovým souborům, kdy stačí jen přístup k internetu, v případě firemních „cloudů“ je nutný přístup do firemní sítě. Nevýhodou takového ukládání je nižší bezpečnost, protože data nejsou na fyzickém disku, který má uživatel u sebe, ale jsou umístěna na internetu. Trh s „cloudovými“ službami by měl v roce 2020 dosáhnout hodnoty 500 miliard dolarů.

BPA (Business Process Automation)

Tento pojem můžeme přeložit jako automatizaci procesů činnosti organizace. Jedná se o technologie, které je možné naprogramovat tak, aby plnily řadu úkolů, které dříve vyžadovaly lidskou práci. Znamená to automatizaci úkolů, častěji ale spíše částečnou, protože stále vyžaduje lidský zásah. Tyto softwary mohou vykonávat činnost, avšak nedokážou „myslet“ (jako *umělá inteligence*) ani „se učit“ (jako *strojové učení*).

RPA (Robotic Process Automation)

V překladu tento pojem znamená Robotická automatizace procesů. Jde o vylepšení BPA, kdy software má zapojenou *umělou inteligenci* i *strojové učení*. Tyto procesy přináší větší efektivitu do současných velkoobjemových a na práci náročných procesů a snižují tak provozní náklady.

IoT (Internet of Things)

Internet věcí je propojeným prostředím, kde spolu všechna zařízení v nějakém daném prostoru (například v jedné domácnosti) komunikují a vyměňují si data. Internet věcí začal pronikat na masový trh začátkem roku 2014 a dnes je na celém světě připojené 26,66 miliard zařízení. Výhodou je usnadnění života, avšak nebezpečím je skutečnost, že zvyky, preference a potřeby uživatelů jsou umístěny na internetu a při nedostatečném zabezpečení se k nim

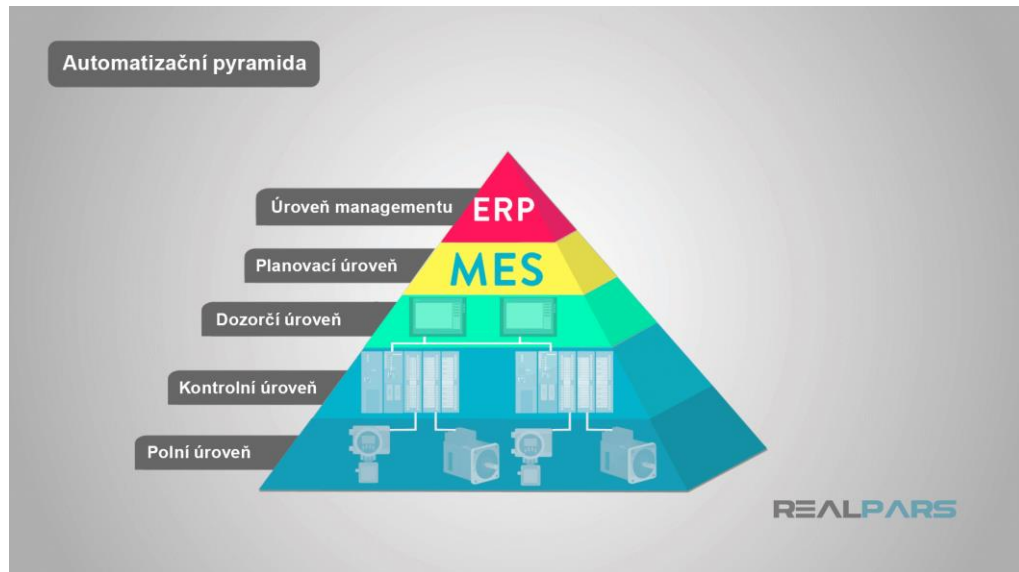
útočník může snadno dostat. V průmyslovém prostředí nazýváme internet věcí EIoT (Enterprise Internet of Things).

Digitální ekonomika

Jde o přímý produkt Průmyslu 4.0. Tímto pojmem se rozumí ekonomická aktivita, která je výsledkem milionů každodenních online spojení mezi lidmi, podniky, zařízeními, daty a procesy. Páteří digitální ekonomiky je *hyperkonektivita*, která znamená růst vzájemného propojení lidí, podniků a strojů (Cope, 2020). Růst digitální ekonomiky mění a podkopává tradiční představy o struktuře ekonomiky a je výzvou pro společnosti, obyvatele i státy. O změně struktury ekonomiky poznamenal TechCrunch, internetový ekonomický zpravodaj, toto: „*Uber, největší světová taxi společnost, nevlastí žádné vozy. Facebook, světově nejpulárnější vlastník médií, nevytváří žádný obsah. Alibaba, nejhodnotnější maloobchod, nemá zásoby. A Airbnb, největší světový poskytovatel ubytování, nevlastní žádnou nemovitost. Něco zajímavého se děje.*“. Ano, věci jako jsou sdílená ekonomika či sociální sítě od základu mění fungování ekonomiky a každý se musí této změně přizpůsobit.

1.2 Automatizační pyramida

Jde o grafické hierarchické zobrazení jednotlivých technologických úrovní automatizace a jejich vzájemné integrace. V automatizační pyramidě je rozlišeno 5 úrovní. Komunikace mezi jednotlivými úrovněmi je realizována průmyslovou komunikací.



Obrázek 1.2 – Automatizační pyramida (Cope, 2020)

Polní úroveň

Jde o nejnižší úroveň této pyramidy. Na této úrovni se vykonává fyzická práce a monitorování. Do této úrovně spadají elektrické motory, senzory, spínače apod. Dá se říci, že až do třetí průmyslové revoluce probíhalo veškeré automatizování výroby pouze na této úrovni. Další úrovně přišly až s rozvojem mikroelektroniky a digitalizace.

Kontrolní úroveň

Kontrolní úroveň ovládá zařízení na polní úrovni a pomocí této úrovně může být vykonávána fyzická práce polní úrovně. Kontrolní úroveň také sbírá veškeré informace ze senzorů, spínačů, či z jiných vstupních zařízení, pomocí těchto informací rozhoduje, jaké zapne výstupy podle toho, jak je dané zařízení na kontrolní úrovni naprogramováno (Cope, 2020). Typickým představitelem kontrolní úrovně je PLC, do kterého může být integrován například PID regulátor pro efektivnější regulaci.

Dozorčí úroveň

Obecně se dá říci, že dozorčí úroveň je zastoupena systémem SCADA. Tato úroveň umožňuje přístup k datům a kontrolu systémů nižších úrovní z jedné lokace. Plusem je také grafické uživatelské rozhraní (GUI) pro vzdálené ovládání funkcí. SCADA není limitována jedním zařízením jako HMI (Cope, 2020).

Plánovací úroveň

Typickým představitelem této úrovně je MES systém, který je součástí této bakalářské práce. Plánovací úroveň monitoruje a rozhoduje o výrobních procesech ve společnosti. MES má informace o požadavcích a také má informace o momentálních zdrojích pro výrobu. MES tedy na základě těchto dat rozhodne o tom, jaký výrobek se začne vyrábět a jaké suroviny je nutno dodat. Detailnější rozbor tohoto systému je v kapitole 1.3.

Úroveň managementu

Jedná se o nejvyšší úroveň automatizační pyramidy. Zástupcem této úrovně je ERP. Tato úroveň je pro nejvyšší vedení dané společnosti. V této úrovni vedení dohlíží na společnost a také ji pomocí této úrovně ovládá. ERP je obvykle sada různých aplikací, které monitorují veškeré činnosti uvnitř společnosti. Sjednocuje všechny předchozí úrovně a přidává další SW pro dosažení vyšší úrovně integrace. Dovoluje monitorovat všechny úrovně společnosti od výroby, přes prodej a nákup k výplatám. Zatímco plánovací úroveň má dosah maximálně pro výrobu a dodávku zdrojů, tato úroveň monitoruje absolutně všechny činnosti ve firmě (Cope, 2020).

1.3 MES

Vychází z anglického označení *Manufacturing Execution Systems*. To můžeme volně přeložit jako *Výrobní informační systémy*. Jde o počítačové systémy, které se používají ve výrobních podnicích pro řízení a monitoring výrobních procesů, které vedou k přeměně surovin na hotové výrobky. Napomáhají rozhodujícím pracovníkům ve výrobě přijímat důležité rozhodnutí anebo odhalit případný problém co nejdříve. MES tím pádem vede ke zvýšení efektivity výroby. K hlavním rysům MES systému je, že pracují v reálném čase (MES centrum, nedatováno(a)).

Z automatizační pyramidy, která byla popsána výše, jde vidět, že MES je součástí plánovací úrovně a tvoří spojení mezi dozorčí úrovní a úrovní managementu.

1.3.1 Základní funkcionality MES systémů

Organizace *MESA International* v roce 1992 zveřejnila MESA-11 model. Ten se postupně vyvíjel přes různé moduly až po MESA model (2006). Tento model obsahuje funkcionality, které by měl MES systém obsahovat. Níže budou popsány základní funkcionality MES systémů.

Detailní plánování a rozvrhování výroby

Pomocí nějakého algoritmu je vytvořena fronta práce, která definuje pořadí, v jakém se budou na výrobním zdroji zpracovávat jednotlivé příkazy. Algoritmus může být jednoduchý, kdy je založen pouze na prioritě jednotlivých zakázek, nebo může být i komplexní, kdy je založen na genetických algoritmech. Genetické algoritmy jsou relativně mladé algoritmy, které slouží k optimalizaci úloh. Tyto algoritmy tvoří frontu práce, která určuje, v jakém pořadí se na výrobním zdroji budou zpracovávat jednotlivé příkazy. Fronta práce je samozřejmě sestavena tak, aby nedocházelo ke zbytečným prostojům, spotřebě energie apod..

Správa výrobních zdrojů

Správné přidělení a sledování zdrojů a kapacit, které jsou potřebné pro výrobní proces. Zdroji mohou být osoby, materiál, zařízení, ale i energie. Informace je založena na aktuálním stavu a budoucích rezervách těchto zdrojů. Zjišťuje také informaci o dostupnosti zdrojů pro přiřazený úkol.

Správa výrobních postupů

Tato funkcionality zahrnuje evidenci, správu verzí a výměnu kmenových dat s okolními systémy, například s výrobními pravidly finálních výrobků, výrobními zdroji apod. Tyto informace pak slouží k definici, která popisuje tvorbu finálního produktu.

Řízení a monitoring výroby

Souhrn aktivit, které řídí tok výroby přiřazováním práce jednotlivým zařízením a osobám, zajišťuje potřebné množství surovin a energie, sleduje aktuální stav výroby, operativně řeší výpadky apod. Řízení výroby v MES systémech je velice důležité vzhledem k propojení na úroveň managementu (ERP systémy) a také pro případné online zpřístupnění informací o rozpracované výrobě.

Sběr dat

Zajištění sběru a historizace procesních a výrobních dat. Počet získávaných informací je velice různorodý, založen na automatizaci a komplikovanosti výroby. Ve velmi automatizované a komplikované výrobě může docházet i ke sběru tisíců hodnot každou minutu.

Sledování výrobků a jejich rodokmen

Souhrn aktivit, které zajišťují shromažďování informací a poskytování informací o zdrojích, které jsou aktuálně použity pro výrobu finálního produktu. Také může sledovat spotřebu materiálů, výrobu meziproduktů apod. Tato aktivita je důležitá z legislativních důvodů a pro případné audity či pro řešení reklamací.

Výkonnostní analýzy

Tyto analýzy, které je možné pomocí MES systému vytvořit, jsou podniky používány pro vyhodnocení jejich úspěchu, případně k vyhodnocení úspěchů v jednotlivých oblastech celého výrobního procesu. Pro každý podnik jsou důležité jiné ukazatele. Záleží především na strategii, jakou daný podnik razí. Jedním z nejznámějších ukazatelů je *OEE (Overall Equipment Effectiveness – Celková efektivita výroby)* (MES centrum, nedatováno(b)). Tento ukazatel násobí dostupnost, (podíl skutečného času výroby s plánovaným), výkon (podíl skutečně vyrobeného množství s teoreticky možným množstvím) a kvalitu (poměr správných výrobků ke skutečnému množství). Výsledkem je efektivita výroby vyjádřená v %. Většinou se bere výsledek nad 85 % jako výborný výsledek.

1.3.2 Výhody MES systému ve výrobě

MES systém pomáhá vytvořit bezchybné výrobní procesy a pomáhá utvářet jednotný pohled na výrobní data. Další výhody MES systému pro výrobu jsou:

- dohledatelnost výroby,
- zajištění přesných dat z výroby,
- snížení prostojů, neshodné výroby, či zkrácení seřizovacích časů,
- zvýšení Celkové efektivity výroby (OEE),
- snížení nutných skladových zásob (dnes je v průmyslu snaha o co nejmenší skladové zásoby, protože je to drahé),
- zavedení bezpapírové výroby (vše elektronicky),
- možnost přesného ekonomického vyhodnocení výroby (MES centrum, nedatováno(a)).

1.4 PLC

1.4.1 Základní definice

Jedná se o celkem malý průmyslový počítač, který se používá pro automatizaci procesů v reálném čase s co možná nejkratší dobou odezvy. Pro PLC je příznačné, že jeho programy se vykonávají v cyklech. Proto je tak vhodný např. pro průmyslové linky. Dalším znakem PLC je použití periférií, které mohou být rovnou napojeny na technologický proces. Nejdůležitějšími perifériemi jsou vstupy a výstupy (I/O) Ty mohou být buď digitální (DI/DO) nebo analogové (AI/AO). S rozvojem automatizace jsou stále častější tzv. funkční moduly (FM) a komunikační procesory (CP).

1.4.2 Historie PLC

Prapočátek PLC se nachází v roce 1968 v automobilce General Motors. Zdejší inženýři hledali alternativu ke komplexním elektromechanickým (reléovým) řídicím systémům.

Nevýhody reléového propojení, které předcházelo PLC, spočívaly v pevném propojení, na tuto nevýhodu se váže obtížná změna řídicí funkce. Nevýhodou bylo i nákladné uvedení do provozu i nákladná údržba. Požadavky na nový systém tedy byly:

- jednoduché programování,
- změny programu bez nutnosti měnit nějak zapojení,
- jednoduché a levné údržbě,
- menší a spolehlivější než reléové zapojení,
- jednoduché uvádění do provozu,

V 70. letech tedy vznikly první PLC. Dnes je PLC klíčovým prvkem průmyslové automatizace. Jeho hlavní výhodou je přizpůsobení provozu. PLC je odolný proti nárazům, výkyvům teplot, elektromagnetickému záření apod..

1.4.3 Typy PLC

Tabulka 1.1 – Dělení PLC

Dělení podle:	Typ	Poznámka
Provedení	Kompaktní	Vše v jednom pouzdře
	Modulární	
	Zabudovaný operátorský panel	
Velikosti	Mikro PLC	Malá paměť, bez komunikace, do 20 I/O
	Malá PLC	Možnost komunikace, do 128 I/O
	Střední výkonnostní třída	Modulární provedení, 128 až 512 I/O
	Nejvyšší výkonnostní třída	
Počtu procesorů	Jednoprocesorová	Někdy možnost multitaskingu
	Víceprocesorová	Většinou modulární PLC (každý modul má svůj procesor)

1.4.4 Programování PLC

Programovací jazyky pro PLC dělíme do 2 hlavních kategorií. Na **textové** a **grafické** jazyky. Mezi textové programovací jazyky patří **Seznam instrukcí (Instruction list)** a **Strukturovaný text (Structured text)**.

Seznam instrukcí

Jazyk, který je velice podobný Assembleru. Při použití tohoto jazyku jsou použity jednoduché operace jako LD (Load), AND, OR apod. Každá instrukce je na novém řádku. PLC zpracovává instrukce postupně.

Výhodou je kompaktnost programu napsaného v tomto jazyce. Nevýhodou je nepřehlednost takového programu a komplikované opravování chyb v programu.

Příklad kódu:

LD A (Načti A)

AND B (And B)

ST Q (Ulož výsledek do Q)

Konstanta *A* je načtena do paměťového registru. Následuje logická operace *AND* s konstantou *B*, výsledek operace je uložen do *Q*.

Strukturovaný text

Textový jazyk na „vyšší“ úrovni, Podobný například programovacímu jazyku *C*. Zvládne zpracovat komplexnější operace pomocí algoritmů a matematických funkcí spolu s opakujícími se úkoly.

Stejně jako v programovacím jazyku *C* se instrukce oddělují středníky, tyto instrukce změny vstup, výstup nebo proměnně. Také některé instrukce jsou stejné jako v jazyku *C*, například *IF*, *ELSE*, *FOR* apod.. Pokud má programátor zkušenost s některým z jazyků, které mají tomuto jazyku blízko, kromě jazyka *C* je to například *Pascal* či *Basic*, tak je pro něj relativně jednoduché se naučit pracovat s tímto jazykem.

Mezi výhody patří dobré zpracování velkých matematických výpočtů a použití některých instrukcí, které není možné použít v jiných jazycích pro PLC. Mezi nevýhody patří komplikovaná úprava online, složitost v ladění a možná složitost konečného kódu.

Příklad kódu:

```
IF Start_PB AND NOT Stop_PB THEN  
MOTOR_RUN_RELAY := 1;  
END_IF;  
IF Stop_PB THEN  
MOTOR_RUN_RELAY := 0;  
END_IF;
```

Jednoduchý program, který spouští a vypíná motor. První podmínky *IF* spustí motor, pokud je zmáčknuto startovací tlačítko a není zmáčknuto vypínací tlačítko. Druhá podmínka vypne motor, pokud je zmáčknuto vypínací tlačítko. Červeně jsou vyznačeny vstupy a výstupy.

Mezi grafické programovací jazyky patří **Příčkový diagram (Ladder diagram)**, **Sekvenční diagram (Sequential function chart)** a **Function block diagram (Funkční blokový diagram)**.

Příčkový diagram

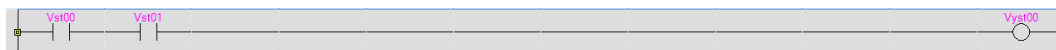
Jazyk, který vychází z reléové logiky a jeho kořeny sahají až do druhé světové války. Využíval se pro zakreslení elektrotechnického schématu a díky jeho jednoduchosti z něj byl vytvořen i programovací jazyk pro PLC.

Jazyk funguje na bázi dvou svislých příček, na levé příčce jsou vstupy, na pravé výstupy. Ty vodorovně propojujeme. Vstupy spouští výstupy, vstupy je možné různé kombinovat, například sériovým zapojením vytvoříme podmínku AND pro spuštění výstupu, paralelním zapojením vytvoříme podmínku OR pro spuštění výstupu. Možné jsou i časovače (ošetření zámků, činnost jen na určitý čas a podobně).

Výhodou je dobrá organizace pomocí příček a snadné sledování chodu programu. Snadné dokumentování komentářů v programu a pomocí ladderu je velice snadná online úprava programu. Ladder patří k nejpobulárnějším jazykům a podle autora je nejpřívětivějším programovacím jazykem pro programování PLC a také nejjednodušší k naučení se pro úplné začátečníky. Jako výhodu bych nazval i snadnou simulaci, kdy pro otestování funkčnosti stačí jednoduše spouštět a vypínat jednotlivé vstupy.

Hlavní nevýhodou je nemožnost naprogramování určitých operací, kdy Příčkový diagram naráží na své limity. Tato věc ztěžuje programování pohybu či dávkování.

Sériové zapojení, aby byl výstup sepnut, musí být oba 2 vstupy sepnuty:



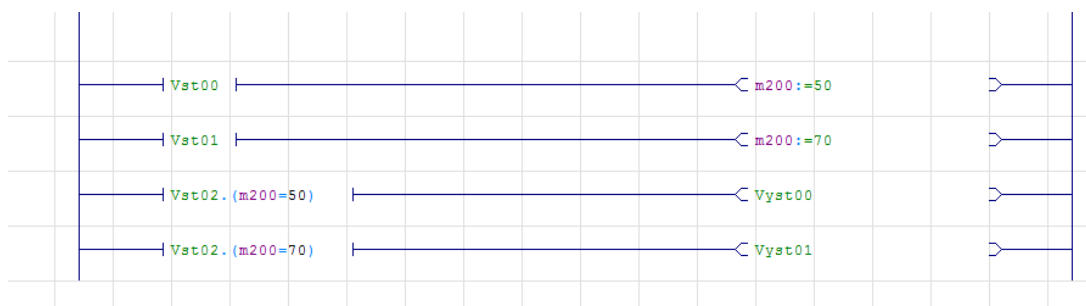
Obrázek 1.3 – Sériové zapojení v Ladderu

Paralelní zapojení, aby byl výstup sepnut, stačí, aby byl spuštěn jen jeden vstup:



Obrázek 1.4 – Paralelní zapojení v Ladderu

Příklad kódu:



Obrázek 1.5 – Příklad kódu v Ladderu

Jedna ze cvičných středoškolských úloh pro seznámení se s Příčkovým diagramem. Tento program má 3 vstupy a 2 výstupy. Je zde použita i pomocná proměnná *m200*. *Vst00* nastaví proměnnou na hodnotu 50, *Vst01* nastaví proměnnou na hodnotu 70. Podle informace, jaký z těchto vstupů jsme naposled spustili, se po spuštění *Vst02* spustí výstup. Pokud jsme před tím spustili *Vst00*, spustí se *Vyst00*, při spuštění *Vst02* se spustí *Vyst01*. Pokud jako první spustíme *Vst02*, nestane se nic, protože v daném vstupu se jedná o logickou funkci *AND* a proměnná je ze začátku na 0, takže podmínka nemůže být splněna. V jednom vstupu to musí být z důvodu, že se jedná o pomocnou proměnnou. V jednu chvíli nemůže být spuštěn *Vyst0* a *Vyst01*. Pokud je *Vst02* spuštěn, ale je přepsána pomocná *m200*, spustí se druhý výstup. Proměnná je nastavena na hodnotě posledně spuštěného vstupu.

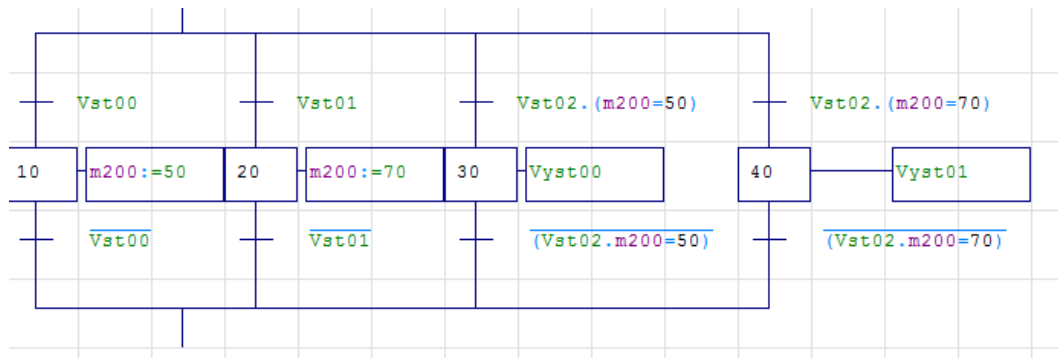
Sekvenční diagram

Někdy se tento jazyk také nazývá **Grafcet**, což je francouzské označení tohoto jazyku. Jedná se o jazyk, který má blízko k vývojovým diagramům. Tento jazyk se skládá z **kroků** a **přechodů**. Každý krok je následován přechodem a každý přechod je následován krokem. Kroky jsou hlavní funkcí tohoto programu. Během kroků se vykonává nějaká akce, kterou programátor připojil k danému kroku. Přechody slouží k tomu, aby programátor určil, za jaké podmínky program přejde do dalšího kroku. Jakmile je tato podmínka splněna, program opouští daný krok a přechází do dalšího, který je znovu spojen s nějakou akcí.

Na rozdíl od klasických vývojových diagramů, sekvenční diagramy mohou mít více cest, které se vykonávají v jeden čas. Programátor tak může spustit více kroků v jeden okamžik.

Mezi výhody patří fakt, že procesy mohou být rozděleny do hlavních kroků, což činí odstraňování chyb rychlejší a jednodušší. Výhodou také je, že programátor má přímý přístup k informaci, kde došlo k chybě na některém ze zařízení. Díky tomu, že do daného kroku se dá opakovaně vracet a vykonat akci znovu, může být vytvoření logiky rychlejší a jednodušší (Dixon, 2020). Nevýhodou je, že určité aplikace není možné pomocí sekvenčních diagramů sestavit.

Příklad kódu:



Obrázek 1.6 – Příklad kódu v Grafcet

Jedná se o stejnou středoškolskou úlohu jako u příčkového diagramu. Některé programy, například program od Mitsubishi *GX Works3* nedovolí, aby programy začínal přechodem a ne krokem. Avšak prostředí *Automgen*, pomocí kterého je vytvořen tento program, s tím zvládne pracovat. Pokud to možné není, dá se to snadno obejít prvním krokem, který nezpůsobuje žádnou akci.

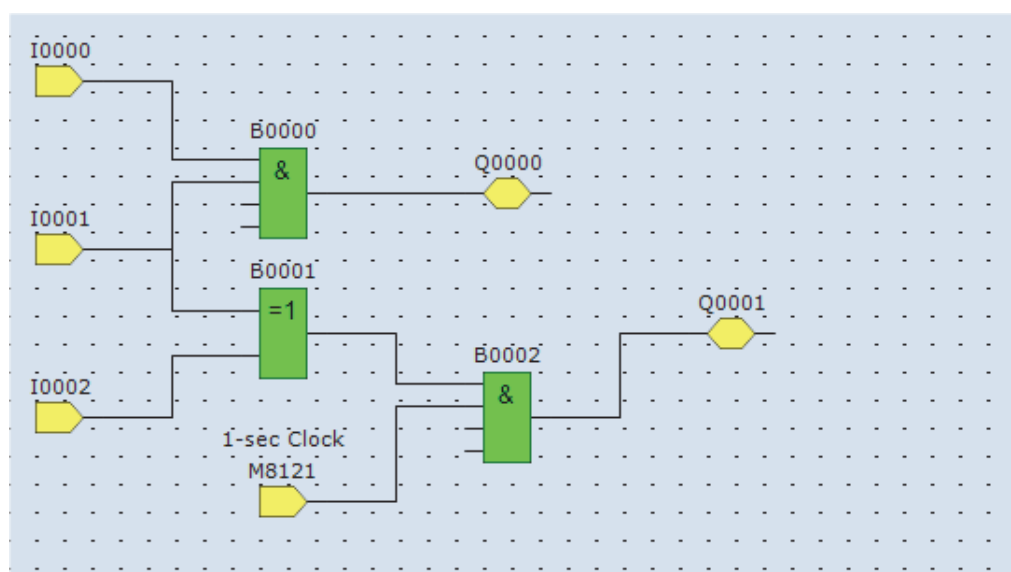
Tento program není s opakováním, to by se vyřešilo navedením konce na začátek programu. Proměnná m200 se nastaví na hodnotu podle spuštěného vstupu. Na 50 se nastaví při spuštění vstupu *Vst00*, na 70 při spuštění vstupu *Vst01*. Po spuštění vstupu *Vst02* přejde do kroku podle hodnoty v proměnné. Tento krok program opustí, pokud se vypne *Vst02* nebo se přepíše hodnota proměnné na druhou hodnotu.

Funkční blokové schéma

Poslední z grafických jazyků. Jedná se o sestavení blokového schématu, které reprezentuje tok signálu mezi jednotlivými prvky schématu. Jedná se o analogii k toku signálu v elektronických obvodech. Tato schémata popisují chování funkcí, funkčních bloků a programů jako množiny propojených grafických bloků (Balda, nedatováno). Na rozdíl od příčkového diagramu není možné toky přímo spojovat, pro tvorbu podmínek je nutné použít dané bloky. Výstup z bloku je možné rozvést. Před blok je možné dát vstupní proměnnou, za blok je možné dát výstupní proměnnou, ale je možné i bloky propojit mezi sebou.

Mezi výhody patří dobrá funkčnost programu při ovládání pohybu. Tato metoda je pro některé uživatele jednodušší k pochopení díky její názornosti. Avšak největší výhodou je, že mnoho řádků programování se dá spojit do jednoho či několika funkčních bloků (Dixon, 2020). Nevýhodou je možná neorganizovanost tohoto programu, protože bloky je možné umístit kdekoli, to může činit problém i při ladění.

Příklad kódu:



Obrázek 1.7 – Příklad programu ve funkčním blokovém schématu (IDEC, 2017)

Zde je velice jednoduchý příklad funkčního blokového schématu. Na blocích vlevo je možné vidět symbol $=I$, tento symbol ve funkčním blokovém schématu reprezentuje podmínku OR. Tento program má **3 vstupy, 2 výstupy a 1 generátor časových impulzů**, který se vypíná/zapíná každou sekundu. Výstup $Q0000$ je spuštěn, pokud jsou sepnuty výstupy $I0000$ a $I0001$. Výstup $Q0001$ se spíná každou sekundu, pokud je sepnut jeden z výstupů $I0001$ a $I0002$.

1.5 Robotika

Robotika a roboti jsou velice široké téma a jejich využití je prakticky všude. Od lékařství, přes průmysl až po vesmírný program.

1.5.1 Klasifikace robotických zařízení dle úrovně řízení

Roboti s ručním ovládáním

Manipulátor bez řídicího systému, který je ovládán člověkem, jeho funkcí je pouze násobit sílu a eliminovat nedokonalosti člověka (Roboty a manipulátory, 2008). Použití můžeme hledat ve stavebnictví, v lékařství či ve škodlivém prostředí.

Roboti s pevným programem

Zařízení s automatickým řídicím systémem, kde je změna programu značně problematická a jedná se o podstatný zásah do systému. Jejich výhodou bývá především nízká cena, jednoduchost a vysoká spolehlivost (Roboty a manipulátory, 2008). Použití nalezneme u dopravníků, jednoúčelových balících strojů či podavačů.

Roboti s flexibilním programem

Zařízení s automatickým řídicím systémem, kde změna programu není složitá a je mnohem rychlejší než u robotů s pevným programem.

Roboty s flexibilním programem dělíme na 3 druhy. Prvním druhem jsou **průmyslové roboty**, které nachází použití hlavně v průmyslových odvětvích, jde o roboty, které lze adaptovat pomocí změn v programu. Můžou vykonávat různé činnosti, mezi ně patří například paletizace, osazování strojů, svařování a podobně.

Dalším druhem robotů s flexibilním programem jsou **adaptivní průmyslové roboty**, jde o roboty s vyšší úrovní řízení, které na rozdíl od průmyslového robotu mohou měnit svůj uložený program, tak aby splnil požadovanou činnost. Svě chování upravují podle okolí díky informacím ze senzorů.

Posledním typem robota s flexibilním programem je **kognitivní robot**, jedná se o robota se schopností autonomní interakce s reálným světem, kde je účelem splnění nějakého cíle. Umělá inteligence je v základě člověkem vložené množství algoritmů.

1.5.2 Pohybové struktury konvenčních robotů

V obrázcích a v názvech jsou písmena T a R. **T** znamená *translační* neboli *posuvný* a **R** znamená *rotační*.

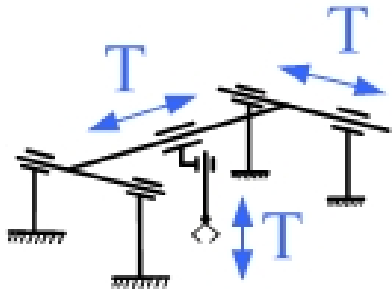
V pohybových strukturách je důležitý pojem *kinematická dvojice*. Jedná se o dva členy akčního mechanismu (dva tělesa), které jsou vzájemně pohyblivě spojeny vazbou. Pohyblivost jednoho člena je vůči druhému omezena. Z členů vzájemně spojených posuvnými a rotačními dynamickými dvojicemi je možné sestavovat libovolné kinematické řetězce (TUKE, 2011).

Tabulka 1.2 – Dělení kinematických dvojic

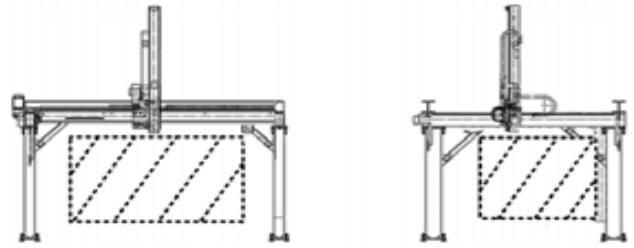
Posuvné kinematické dvojice	
Suportové	Po delším vedení se posouvá kratší těleso
Smykové	V kratším vedení se posouvá delší těleso
Výsuvné	Teleskopické, jedno těleso vyjíždí z druhého
Rotační kinematické dvojice	
Otočné	Jedno těleso se otáčí uvnitř druhého
Kyvadlové	Jedno těleso je zavěšeno na druhém a otáčí se na něm

Kartézská souřadnicový systém (TTT kinematická struktura)

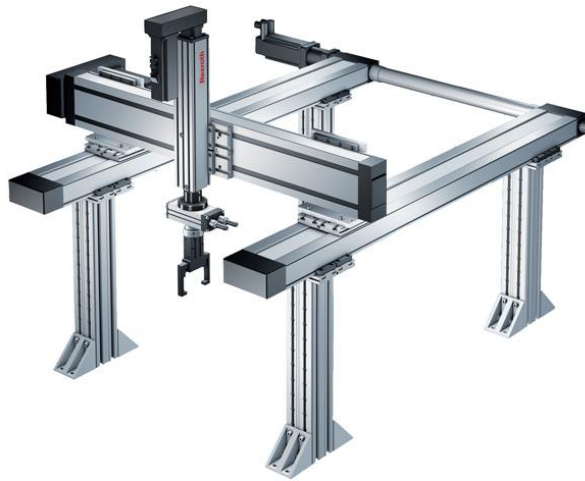
Kinematický řetězec sestavený ze tří posuvných pohybových dvojic. Tyto tři dvojice jsou na sebe vzájemně kolmé. Tato struktura používá pravoúhlý souřadnicový systém. TTT kinematická struktura je velice stabilní a z hlediska kinematického rozboru se jedná o nejpřesnější kinematickou strukturu a také o strukturu s jednoduchým řízením. Nevýhodou je nižší prostorová pohyblivost. Použití nachází hlavně ve velkých manipulačních prostorech. Pracovní prostor robota je tvořen krychlovým tělesem, konkrétně hranolem nebo kostkou (TUKE, 2011).



Obrázek 1.8 – Schéma TTT struktury
(TUKE, 2011)



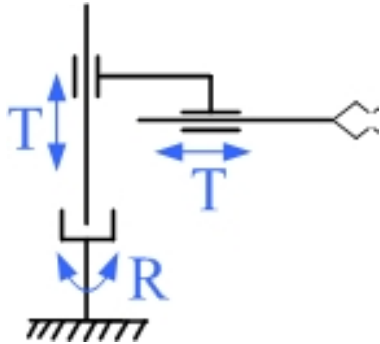
Obrázek 1.9 – Pracovní prostor TTT struktury
(TUKE, 2011)



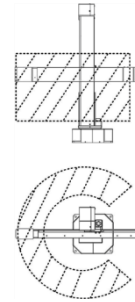
Obrázek 1.10 – Robot s TTT strukturou (Endeavor Business Media, 2013)

Válcový souřadnicový systém (RTT kinematická struktura)

Jde o kinematický řetězec, který je sestaven z jedné rotační pohybové dvojice (rotační pohybová jednotka) a dvou na sobě vzájemně kolmých posuvných pohybových dvojic. Charakteristická je svou robustností a jednoduchým řízením. Pracovní prostor je tvořen válcovým tělesem, konkrétně válcem nebo jeho částí (TUKE, 2011).



Obrázek 1.12 – Schéma RRT struktury
(TUKE, 2011)



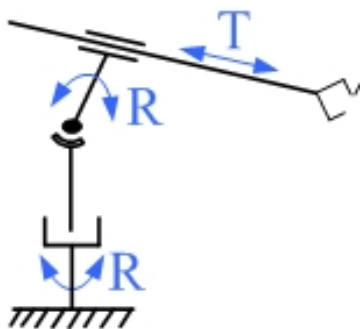
Obrázek 1.11 – Pracovní prostor RRT struktury (TUKE, 2011)



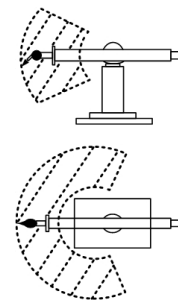
Obrázek 1.13 – Robot s RRT strukturou (Cylindrical Robots, 2020)

Sférický souřadnicový systém (RRT kinematická struktura)

Jedná se o kinematický řetězec, který je sestaven ze dvou rotačních pohybových dvojic a jedné posuvné pohybové dvojice. Pracovní prostor je ohraničen kulovou plochou.



Obrázek 1.15 – Schéma RRT struktury
(TUKE, 2011)



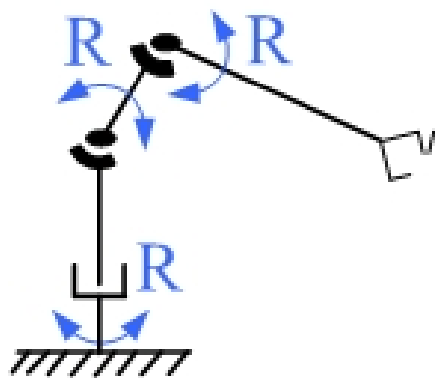
Obrázek 1.14 – Pracovní prostor RRT struktury (TUKE, 2011)



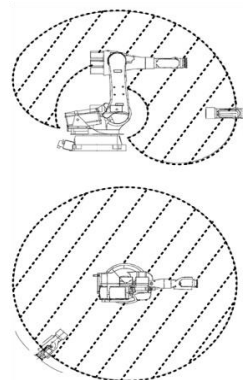
Obrázek 1.16 – Robot s TTT strukturou (Endeavor Business Media, 2013)

Angulární souřadnicový systém (RRR kinematická struktura)

Jde o kinematický řetězec, který je sestaven ze tří rotačních pohybových dvojic. Vyznačuje se dobrou manipulační schopností a tím se dobře vyhýbá překážkám. V poslední době je nejpoužívanější. Pracovní prostor je tvořen multiúhlovým tělesem (TUKE, 2011). RRR kinematickou strukturu má i robot použitý v této bakalářské práci.



Obrázek 1.18 – Schéma RRR struktury (TUKE, 2011)



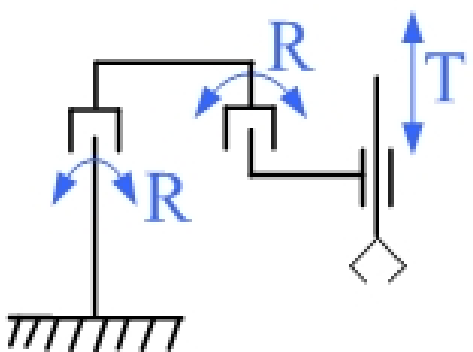
Obrázek 1.17 – Pracovní prostor RRR struktury (TUKE, 2011)



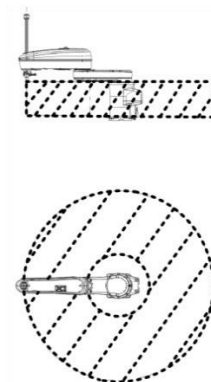
Obrázek 1.19 – Robot s RRR strukturou (Alibaba, 1999-2020)

Kinematická struktura typu SCARA

Jedná se o kinematický řetězec, který je sestaven ze dvou rotačních pohybových a jedné posuvné pohybové dvojice. Výhoda, která ho odlišuje od sférického souřadnicového systému, je dobře umístěná zóna obsluhy a vyšší pohyblivost. Nevýhodou je menší pracovní prostor a složitější řízení. Jsou předurčeny pro pracovní operace prováděné svisle shora a jejich uplatnění najdeme při plošných montážích. Mají velkou rychlost pohybu a vysoké zrychlení. Pracovní prostor je ohraničen prstencem (TUKE, 2011).



Obrázek 1.20 – Schéma SCARA struktury (TUKE, 2011)



Obrázek 1.21 – Pracovní prostor SCARA struktury (TUKE, 2011)



Obrázek 1.22 – Robot se SCARA architekturou (EPSON, 2020)

1.5.3 Programování robota

Off-line programování

Off-line programování je nutné hlavně ve výrobě, když je nutné pokračovat ve výrobě a neztratit čas odstavením robota pro online programování.

Ve většině případů rozlišujeme 2 druhy Off-line programování. Prvním druhem je **textové programování**, jedná se o tradiční metodu programování. Spočívá v napsání programu off-line v textovém editoru a následném nahrání do robota. Pokud se používá programovací jazyk výrobce, je možné získat větší funkcionalitu použitím této metody. Nevýhodou je nutnost většího ladění než za použití jiných metod.

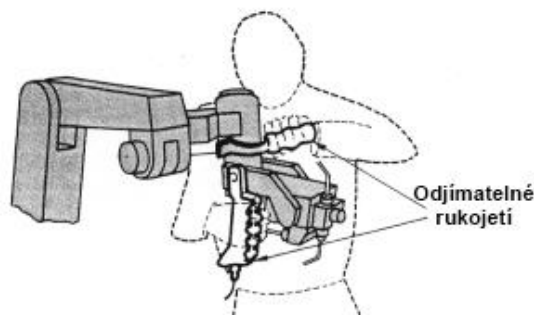
Druhým druhem off-line programování je **grafické programování**. Pomocí této metody programátor pohybuje reálným robotem v simulovaném prostředí. Po dokončení a odladění programu je možné tento program nahrát do robota. Pokud je prostředí dobře vytvořeno, nabízí tato metoda to nejlepší z online programování se všemi výhodami off-line programování.

Online programování

V online programování je programátor přímo připojen na robota a takzvaně robota „učí“. Je nutné naučit pozice, které jsou potřeba pro vykonání činnosti. To programátor udělá pomocí ovládacího panelu, díky kterému může přemístit robota do dané pozice a tuto pozici nahrát do paměti robota. Druhý krok, který je nutný, je naprogramování logické části řízení a periferních řízení. To už se většinou píše na počítači, ale v online programování je možné

program ihned nahrát do robota a vyzkoušet funkčnost, což je výhoda tohoto typu programování. Další výhodou je práce v reálném prostředí. Nevýhodou je nutnost odstavení robota, což je může být při plném provozu, kdy je nutné změnit program, značně komplikované.

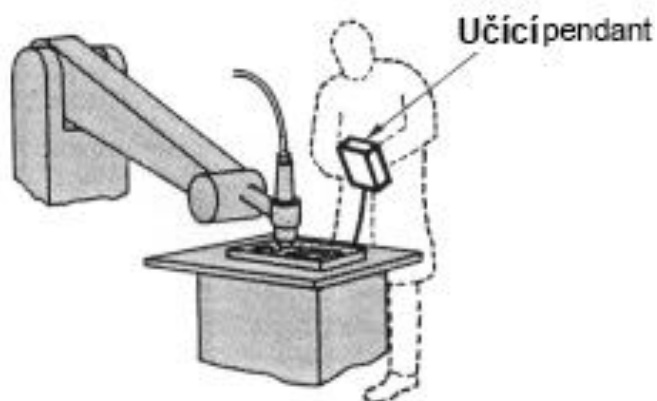
Rozlišujeme 2 druhy online programování. Prvním je **metoda Lead-through**, volně přeloženo jako *vézt přes*. Tato metoda je vhodná v případě, pokud se jedná o jednoduchý program, kde není nutná velká přesnost pohybu. Jedná se o fyzické vedení robota do cílené pozice, kde si robot v pravidelných krátkých intervalech zaznamenává aktuální pozici a tuto cestu pak dokáže robot replikovat. Výhodou tohoto programování je, že je velice intuitivní. Nevýhodou je už zmíněná nižší přesnost, často potřeba dalších senzorů a také nutnost zcela nového naprogramování při změně programu.



Obrázek 1.23 – Programování metodou *vézt přes* (Programming Robots, nedatováno)

Další metodou je **programování pomocí pendantu**. Jedná se o komplexnější metodu online programování, která je dnes mnohem častěji používána. Pendant se přímo připojí do robota. Pomocí něj je možné přemístit robota do požadované pozice a uložit tyto polohy. Pomocí pendantu je možné i ovládat rychlost takového pohybu, pokud potřebuji velice pomalu přejít do přesné polohy, použiji pomalou rychlost, pokud se potřebuji přemístit na vzdálený bod, použiji větší rychlost robota. Pro tento způsob online programování nemusí být programátor tak zkušený jako pro metodu *Lead-through*.

Nevýhodou může být možné nebezpečí, pokud operátor musí být blízko robotovi, dnes však roboti mívají vysoký stupeň zabezpečení, takže se nebezpečí čím dál více minimalizuje, přesto je však opatrnost na místě.



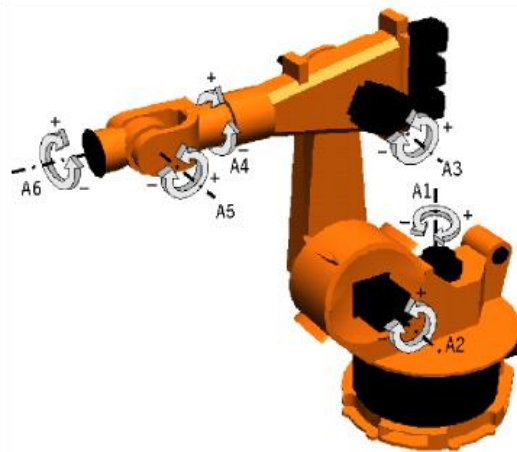
Obrázek 1.24 – Programování pomocí pendentu (Programming Robots, nedatováno)

1.5.4 Souřadnicové systémy robota

Pohyb robota je u většiny robotů možné realizovat v 6 různých souřadnicových systémech. Přičemž systémy *ROBROOT* a *FLANGE* jsou spíše doplňkovými souřadnicovými systémy. Některé z těchto systémů jsou důležité hlavně pro kalibraci robota, jiné jsou určeny pro jednodušší manipulaci s robotem a jeho programování.

Osový souřadnicový systém (JOINT)

V tomto systému je možné pohybovat s každou jednotlivou osou robota zvlášť. S jednotlivými osami je možné pohybovat do záporného i do kladného směru. Pokud se například jedná o rotační osu, tak se může tato osa udávat ve stupních od -180° do 180° .



Obrázek 1.25 – Osový souřadnicový systém (TUKE, 2011)

Nástrojový souřadnicový systém (TOOL)

Jedná se o pravoúhlý (kartézský) souřadnicový systém, jehož nulové souřadnice leží v nástroji. Orientace se zpravidla používá tak, že jeho osa X je identická s pracovním směrem nástroje. Souřadnicový systém TOOL tak stále sleduje pohyb nástroje (TUKE, 2011). Většinou se nástrojem myslí nástroj, který daný robot používá pro práci.

Externí souřadnicový systém (BASE)

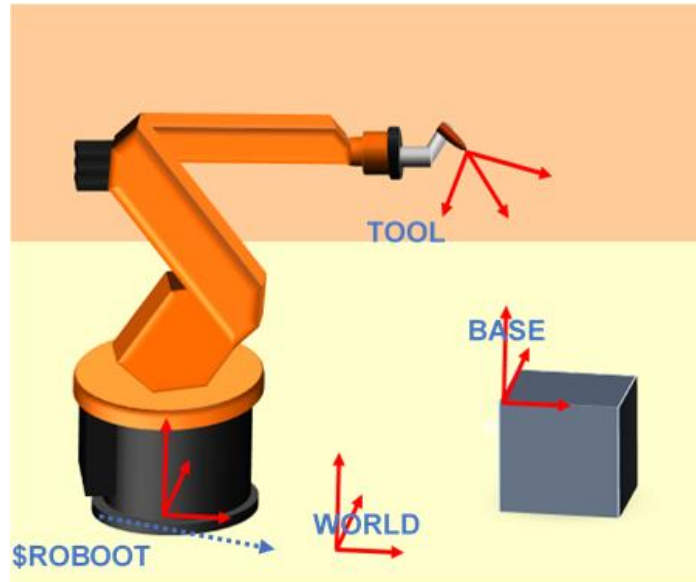
Jedná se o pravoúhlý (kartézský) souřadnicový systém, jehož začátek leží v externím nástroji. To mohou být například svářečské kleště. Při práci v tomto režimu se stává směrodatným souřadnicový systém externího nástroje (TUKE, 2011). V systému BASE se robot pohybuje paralelně s osami souřadnicového systému zpracovaného předmětu. Výhodou může být snadné přepočítání báze do nových souřadnic, pokud se například změní umístění přípravku, ke kterému je souřadnicový systém vztažen.

Základní souřadnicový systém (WORLD)

Jedná se o absolutní, pravoúhlý souřadnicový systém, jehož začátek zpravidla leží uvnitř podstavy robota. V tomto režimu zůstává robot v nulovém bodě, pokud se robot pohybuje jen ve své poloze, tudíž může být použit pro detekování posunutí robota mimo jeho základní polohu.

Souřadnicový systém robota (ROBROOT)

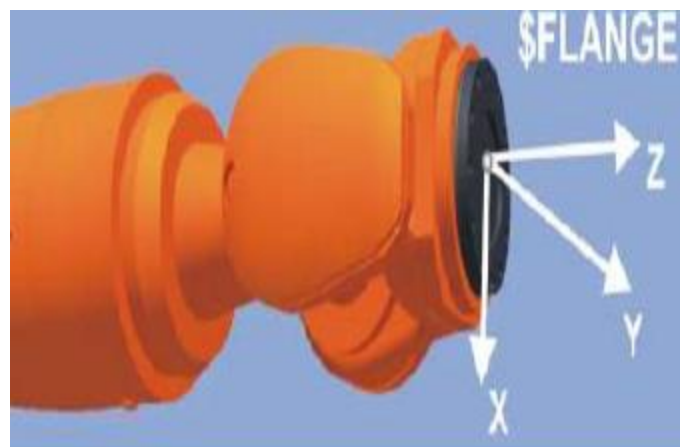
Tento systém se používá, pokud máme nulový bod systému *WORLD* jinde než ve středu podstavy robota. Pak tento souřadnicový systém značí pozici vztaženou k základnímu souřadnicovému systému.



Obrázek 1.26 – Souřadnicové systémy (Coordinates of a robot systém, nedatováno)

Souřadnicový systém příruby (FLANGE)

Jedná se o souřadnicový systém, jehož střed je umístěn v přírubě robota. Na přírubu se nasazuje pracovní nástroj. Je použitelný jako střed pro nástroj. Tento systém je spíše doplňkovým souřadnicovým systémem k souřadnicovému systému *TOOL*.



Obrázek 1.27 – Souřadnicový systém příruby (Shah, 2016)

1.5.5 Dělení průmyslových robotů z hlediska bezpečnosti

Z pohledu bezpečnosti dělíme průmyslové roboty na tzv. „klasické“, kooperativní a kolaborativní. Toto dělení se vztahuje k možnosti přítomnosti člověka u robota.

„Klasičtí“ průmysloví roboti

Pokud se jedná o klasického průmyslového robota, tak v jeho rozsahu pohybů nesmí být člověk při chodu přítomný. Proto je nutné takového robota oplotit. Takový typ robota bývá často v automobilce, kde roboti zajistí veškerou činnost a člověk nemusí nijak do chodu zasahovat.

Kooperativní průmysloví roboti

U kooperativních robotů už je přítomnost člověka možná, ale robot nesmí být v chodu, pokud je člověk přítomný. To se zajišťuje například optickou závorou. Takový robot nachází uplatnění například v provozu, kdy zaměstnanec dává robotovi díly a robot vykoná žádanou činnost na dílech.

Kolaborativní průmysloví roboti

U těchto robotů může být člověk přítomen i při chodu robota. Jedná se o novou věc, proto mají kolaborativní roboti pouze specifikaci a nemají ještě danou normu. Takový robot by měl při kontaktu s cizím objektem okamžitě zastavit. Nevýhoda těchto robotů je v nižší rychlosti kvůli nutnosti zastavit. Takový robot je také značně odlehčený oproti klasickým robotům, při vyšší váze by kvůli setrvačnosti nemohl v okamžiku zastavit, to robota v činnosti může limitovat také.

1.6 Komunikační protokoly v průmyslu

Existuje více druhů průmyslových komunikací, avšak některé jsou nebo se stávají spíše věcí minulosti. V minulosti se v průmyslu používaly hlavně sériové protokoly, dnes získává majoritní postavení *Ethernet*. Sériové protokoly se však stále používají. USB komunikace se až tak v průmyslu nepoužívá kvůli své krátké vzdálenosti možné komunikace.

1.6.1 Sériové protokoly

CAN

Původně byl tento protokol vyvinut pro automobilový průmysl, dnes však našel širší uplatnění. Rychlost dosahuje až 1 Mb/s . Existují 2 protokoly – *CANopen* a *DeviceNet*. Tyto protokoly jsou vyšší úrovně a zajišťují interoperabilitu se zařízeními ve stejné průmyslové síti. *CANopen* podporuje 127 síťových uzlů, zatímco *DeviceNet* podporuje 64 síťových uzlů (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

Jako reálný příklad použití můžu zmínit zabezpečovací systém na detekci kouře, teploty a vlhkosti, kdy sběrnice CAN slouží k sériovému propojení jednotlivých detektorů, kdy centrální zařízení získává informace ze všech detektorů.

Modbus

Jedná se o jednoduchý open-source protokol, který umožňuje propojit až 247 uzlů. Jeho přenosová rychlost je 115 kb/s a je snadno implementovatelný na linkách *RS-232* a *RS-485* (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

PROFIBUS

V dnešní době se používá hlavně v průmyslových automatizačních systémech. Určen pro komunikaci procesních a pomocných dat až do rychlostí 12 Mb/s při maximální podpoře 126 adres (Ethernet vs. sériové protokoly v průmyslu, 1997-2014). Vzdálenost, kterou je možné tímto protokolem dosáhnout je 15 km.

1.6.2 Ethernetové protokoly

PROFINET

Široce používaný protokol, který je používán hlavními výrobci zařízení v průmyslovém Ethernetu. Topologie tohoto protokolu je buď *Branch* (postupné rozvětvení) či *Star* (hvězdicové).

PROFINET se dělí na tři třídy. První je **Třída A**. Ta používá vzdálené volání procedur TCP/IP. Vzdálené volání procedur je technologie, která dovoluje programu vykonat kód na jiném místě, než je umístěn volající program (Remote procedure call, 2001). Tato třída vytváří most mezi rozhraním Ethernet a PROFIBUS, kdy proxy zajistí přístup k síti PROFIBUS. Většinou se využívá pro nastavení parametrů a cyklickou obsluhu vstupů/výstupů, kdy cyklus trvá přibližně 100 ms. Používá se obvykle pro infrastrukturu a automatizaci budov (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

Další je **Třída B**. V této třídě je doba cyklu kratší, přibližně 10 ms a třída je založena na softwaru, který pracuje v reálném čase. Někdy se také označuje *PROFINET Real-Time*. Využitelná například pro PLC, díky schopnosti tohoto protokolu pracovat v reálném čase. Obvykle se používá v továrně a automatizaci procesů (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

Poslední třídou je **Třída C**. Tato třída vyžaduje speciální hardware, aby bylo možné zkrátit dobu cyklu na méně než 1 ms, což umožňuje dostatečně přenášet výkon v reálném čase pro operace řízení pohybu. Někdy se také nazývá *PROFINET IRT* (izochronní reálný čas) (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

EtherNet / IP

Protokol aplikační vrstvy, který využívá *Common Industrial Protocol (CIP)* přes protokol TCP / IP. CIP je možné používat na více fyzických médiích, jako je sběrnice *CAN DeviceNet* nebo *Ethernet (EtherNet/IP)*. *Ethernet/IP* může mít neomezený počet uzlů systému díky využití standardního ethernetu a switche. Tento protokol je kompatibilní s řadou standardních internetových a ethernetových protokolů. Nevýhodou jsou však omezené schopnosti v reálném čase (Ethernet vs. sériové protokoly v průmyslu, 1997-2014).

Modbus/TCP

ento protokol je rozšířením protokolu Modbus pro ethernetové prostředí. Umožňuje zasílání zpráv přes *TCP/IP* na Ethernetu. Implementace je jednoduchá, protože se provádí na standardní síti Ethernet. Nevýhodou je však nezaručená časová a deterministická komunikace.

1.7 Strojové vidění

V angličtině se tento termín nazývá *Machine vision*. Jedná se o funkci, kdy umožňujeme průmyslovým zařízením „vidět“. Některá zařízení obsahovala strojové vidění už v 50. letech, ale až v průběhu 80. let můžeme mluvit o jeho skutečném rozvoji.

Strojové vidění typicky obsahuje tyto komponenty:

- senzory,
- zařízení k digitalizaci snímku,
- kamery (digitální nebo analogové),
- dostatečné osvětlení pro kamery (kvůli získání kvalitních snímků),
- software schopný zpracovat a analyzovat snímky,
- algoritmy, které identifikují vzorek.

Proces strojového vidění začíná tím, že senzor detekuje přítomnost vzorku. Senzor sepnou zdroj osvětlení a ten osvětlí prostor, kamera vytvoří snímek vzorku nebo části vzorku. Pak je snímek digitalizován, čímž se získává digitální výstup. Tento soubor je uložen v paměti a může být analyzován softwarem. Software porovnává soubor s přednastavenými parametry pro identifikaci. Pokud například slouží k identifikaci chybného vzorku, při vzorku, který neodpovídá parametrům, vzorek je vyhodnocen jako chybný.

Strojové vidění je podkategorií počítačového vidění (*computer vision*). Počítačové vidění zvládá komplikovanější úlohy než strojové vidění. Strojové vidění zvládá zpracovat vždy jen snímek či video. Počítačové vidění zvládne analyzovat „snímek“ z pohybových detektorů, infračervených senzorů či jiných senzorů, není omezeno pouze na snímek či video.

Sesterskou kategorií strojového vidění je robotické vidění (*robotic vision*). Tyto dvě kategorie se ve spoustě věcí překrývají, hlavní odlišností je výstup, který je získán z těchto dvou typů počítačového vidění. Zatímco u strojového vidění obrazu je získána nějaká informace, u robotického vidění je to pohyb robota.



Obrázek 1.28 – Rodina systémů zpracovávajících informace (Owen-Hill, 2016)

1.7.1 Použití strojového vidění

Základním využitím je kontrola kvality, kdy se vzorek kontroluje, zda splňuje dané limity. Avšak strojové vidění nachází využití i k identifikaci, inspekci a podobně. Níže budou popsány některé možnosti použití strojového vidění.

Oprava chyb na průmyslové lince

Kromě použití strojního vidění k identifikaci vadných vzorků může strojní vidění pomoci i s identifikací, kde vznikl problém, že je vzorek vadný a umožnit tak provedení nápravných opatření.

Sledování a sledovatelnost produktů

Ve velmi regulovaných průmyslových odvětvích, jako je farmacie, je velice důležitá schopnost sledovat přísady, sériová čísla produktů a sledovat expirační data. Tyto záležitosti strojové vidění velice zjednodušuje.

Zemědělství

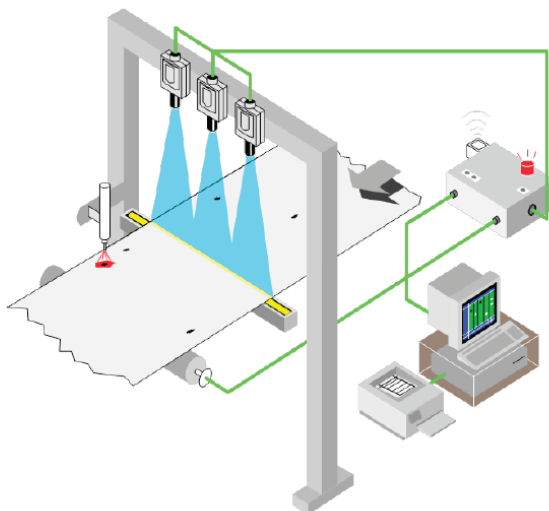
Strojní vidění je například také používáno sklízecími stroji k detekci umístění hroznů na vinné révě. Robotické sklízecí stroje tak mohou sbírat hrozny bez poškození samotných hroznů. Strojové vidění se v zemědělství také používá ke sledování plodin a k detekci případných chorob na rostlinách (Marr, 2019).

1.7.2 Typy zařízení se strojovým viděním

Strojové vidění se v základu dělí na rozměrové prostory, které zvládne zařízení zpracovat:

Jednorozměrné systémy (1D)

Takové systémy analyzují v jeden okamžik, pouze jednu linii produktu místo celého objektu. Jejich využití se nachází například v čárových kódech či detekci vad na materiálu, který byl vytvořen kontinuálním procesem. Mezi takové materiály patří např. papír či kov. Na obrázku 1.29 je kontrola povrchu prostěradla pomocí jednorozměrného strojového vidění.



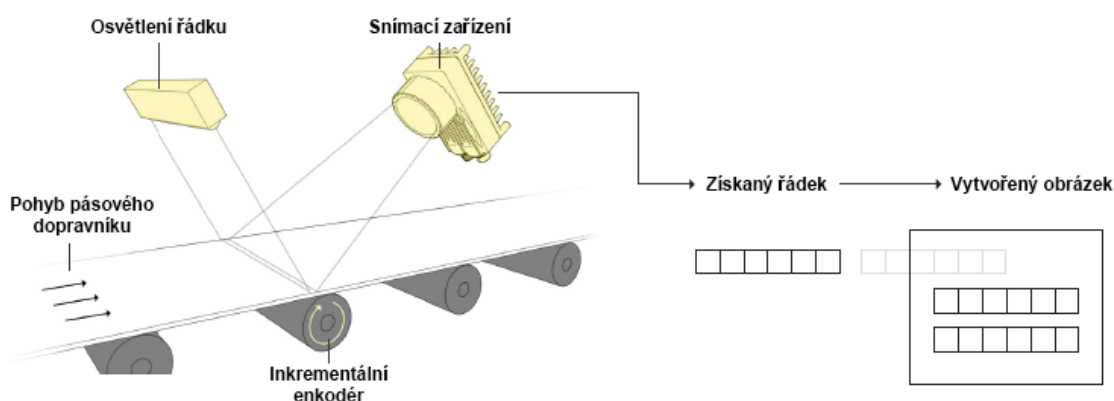
Obrázek 1.29 – Příklad 1D strojového vidění (Introduction to machine vision, 2020)

Dvourozměrné systémy (2D)

Dnes je nejpoužívanějším systémem. Dvourozměrné vidění stačí pro většinu úkonů, takže je zbytečné pro daný úkon používat dražší a komplikovanější trojrozměrný systém.

2D systémy strojového vidění se dělí na dva druhy. Prvním typem je **plošné skenování**. To spočívá v zachycení jednoho snímku, který má nějaké rozlišení. Jedná se o klasické fotografování. Výhodou plošného skenování je určitě jednoduchost, kdy se vytvoří jeden snímek a je hotovo, není potřeba, aby se výrobek postupně hýbal a tak mohl být naskenován celý prostor.

Druhým typem dvourozměrných systému je **Plošné skenování**. Zde se finální obraz tvoří řádek po řádku. Tento způsob je podobný jednorozměrnému systému, zde však systém jednotlivé řádky skládá do dvourozměrného obrazu. Postupné skenování má výhodu ve větším rozlišení finálního snímku, možnosti rozbalení válcového modelu, kdy získáme plošný snímek pouhým otočením kulatého výrobku před skenovacím zařízením. U plošného skenování by bylo potřeba více kamer. Postupné skenování je také výhodnější při nutnosti umístění do stísněných prostor, například mezi válce dopravníku, kdy je nutné skenování dna výrobku. Protože postupné skenování potřebuje výrobek v pohybu k vytvoření celkového snímku, je jeho využití velmi výhodné pro výrobky, které jsou v neustálém pohybu.



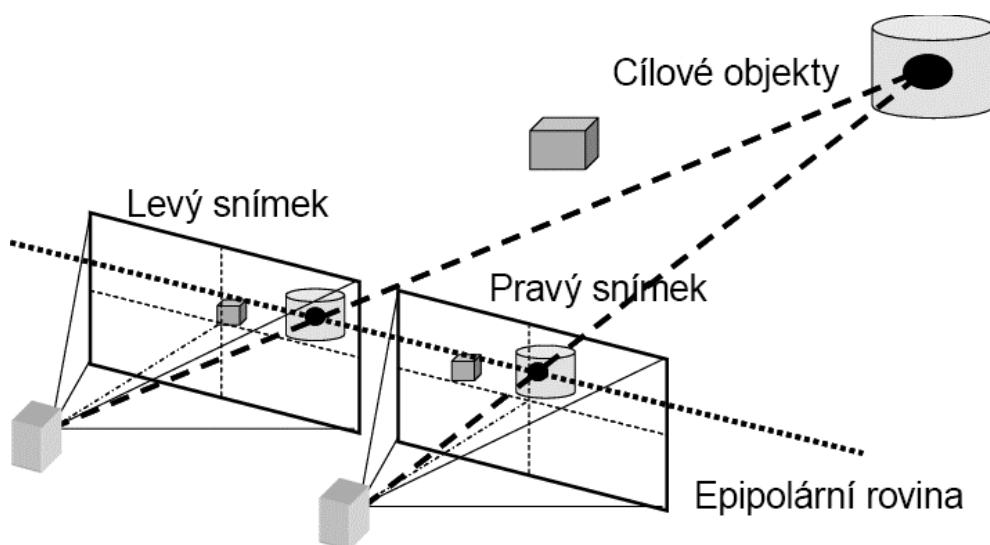
Obrázek 1.30 – Princip postupného skenování (Introduction to machine vision, 2020)

Trojrozměrné systémy (3D)

3D systémy jsou komplexním strojovým viděním, které zažívá v posledních letech velký růst. Poskytuje přesné informace v reálném čase a zlepšuje tak výkon aplikací. Výhodou je také detekce objektu bez ohledu na polohu, díky tomu je takový systém více flexibilní a nezávislý. Nevýhodou je složitější implementace a vyšší cena takových systémů. Avšak díky rozvoji techniky se tyto rozdíly pomalu smazávají. Dnes už nejsou výjimkou systémy, které potřebují jen jednu kameru pro trojrozměrný snímek. Roboti, kteří využívají trojrozměrné vidění, mohou například vědět, co se nachází před nimi a jak správně reagovat.

Tvorba trojrozměrného obrazu je možná mnoha způsoby. Mezi ně patří i **stereofonní vidění**. Tento systém využívá principu lidského vidění. Trojrozměrný snímek se získá pozorováním společného znaku ze dvou snímků, které byly vytvořeny na různých místech. Pomocí vzdálenosti jednotlivých bodů z různých míst snímání jsou vytvořeny hodnoty jednotlivých os. Takové řešení může být velice levné. Postačující je jedna 2D kamera, která je umístěna na robotovi, který umí kameru přemístit na dvě různá místa pro vytvoření snímku (Emerging 3D Vision Technologies for Robot Vision & Machine Vision, 2009-2020). Použit se můžou i dvě dvourozměrné kamery.

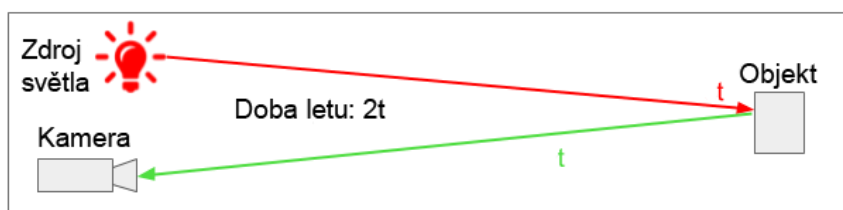
Nevýhodou je nutnost většího prostoru, ať už pro zařízení, které pohybuje kamerou, či prostor pro 2 kamery. Hlavní nevýhodou je však komplikovaná synchronizace. Obě kamery musí být dobře zkalibrovány. Částečně se tento problém dá vyřešit použitím jedné kamery a zrcadel, kdy je hlavní výhodou, že odpadá problém synchronizace.



Obrázek 1.31 – Princip stereofonního vidění (Hariyama, 2008)

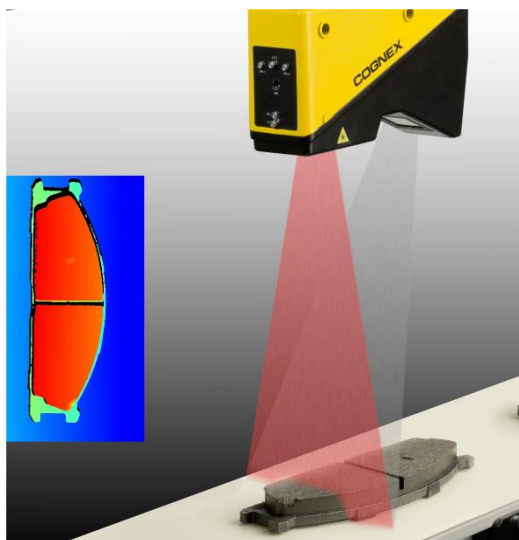
Dalším způsob tvorby trojrozměrného obrazu je **Time of Flight**, který je možné volně přeložit jako „dobu letu“. Tento systém trojrozměrného vidění je založen na měření, jak dlouho trvá světlu, než se dostane k objektu a zase zpět do senzoru. Tímto způsobem se získá z každého senzoru informace o hodnotě osy Z a je možné tak získat informace o objektu (Emerging 3D Vision Technologies for Robot Vision & Machine Vision, 2009-2020). Výhodou je vysoká rychlost snímání a například pro kontrolu povrchu materiálů je takový systém ideální. Nevýhodou může být rušení, kdy si jednotlivé senzory mohou navzájem rušit měření.

Nevýhodou je také možná chybovost, kdy je možné se k objektu dostat různými cestami a tak je naměřená vzdálenost větší oproti skutečné vzdálenosti.



Obrázek 1.32 – Princip ToF skenování (Rooms, 2019)

Na obrázku 1.33 je znázorněna kontrola povrchu brzdové destičky pomocí trojrozměrného strojového vidění.



Obrázek 1.33 – Kontrola povrchu (Introduction to machine vision, 2020)

2 PRAKTICKÁ ČÁST

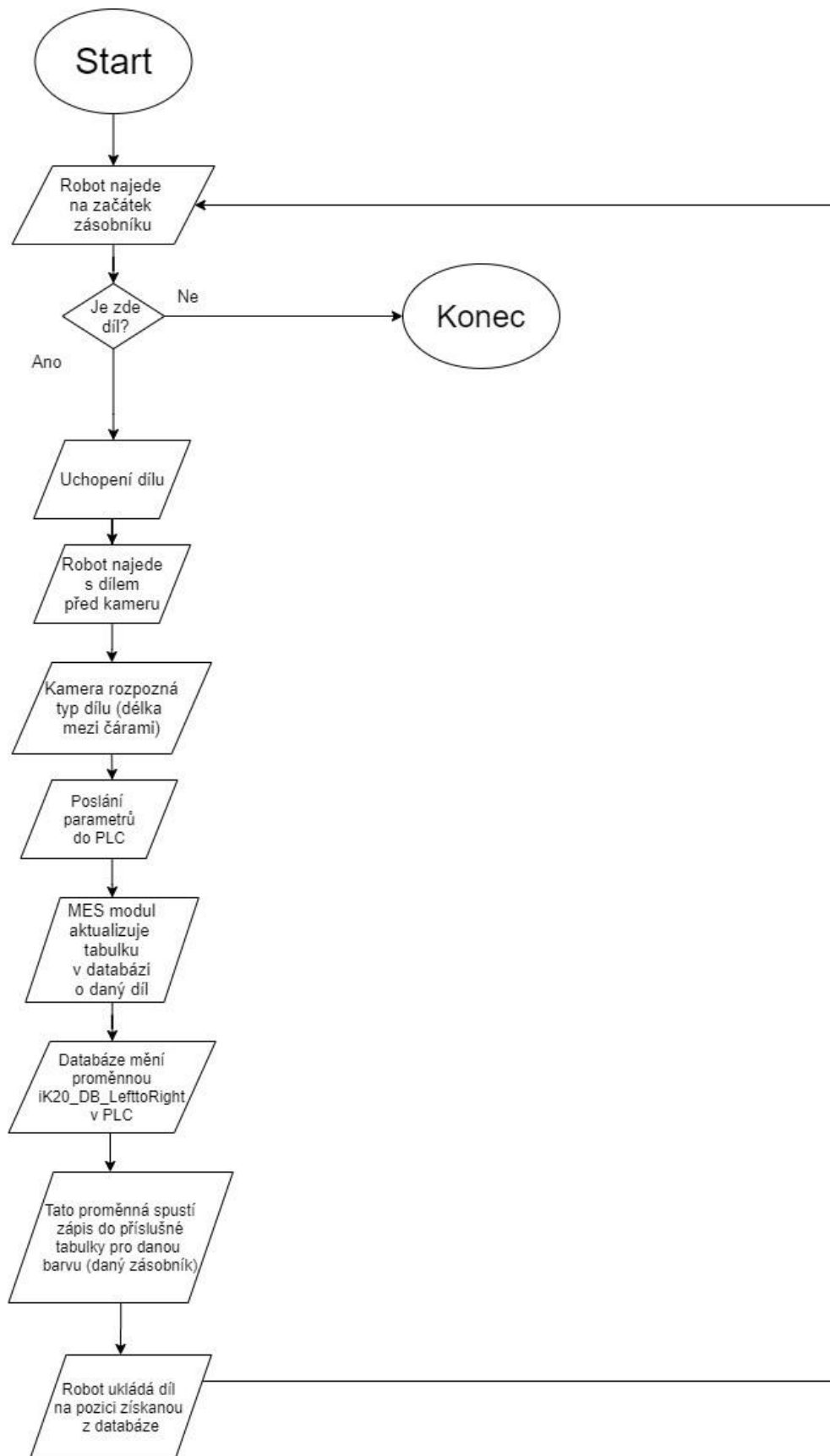
2.1 Naplánování budoucí práce

Jako cíl této práce bylo naplánováno zprovoznění a sladění těchto činností:

- robot si bude brát ze zásobníku jednotlivé díly, které budou stejné velikostí, ale budou různých barev a každý bude jinak označen (délka mezi svislými čarami), tyto díly ukáže kameře, která rozpozná, zda je díl správně natočen, pokud ne, díl otočí, pokud ano, tak délka mezi čarami, bude parametrem, který bude jednotlivé díly (barvu dílů) odlišovat;
- informace o dílu se zapíše do tabulky všech záznamů v databázi, databáze pak pošle do PLC celočíselný údaj o délce mezi čarami, díky této informaci se díl zapíše do příslušné tabulky pro danou bravu (daný zásobník). Z dané tabulky se zapisuje do PLC číslo zásobníku a počet záznamů (počet dílu v zásobníku). Z těchto informací robot díl správně uloží;
- robot skládá tak dlouho, dokud má díly, jakmile zjistí, že žádný nemá, program se přerušuje a robot se vrací zpět do domovské pozice.

Vývojový diagram

Pro zjednodušení, zpřehlednění a pro lepší pochopení fungování celého projektu, jsem si vytvořil vývojový diagram, kde popisují veškeré činnosti zahrnující veškeré činnosti programu od uchopení dílu až po jeho výsledné uložení na správnou pozici.



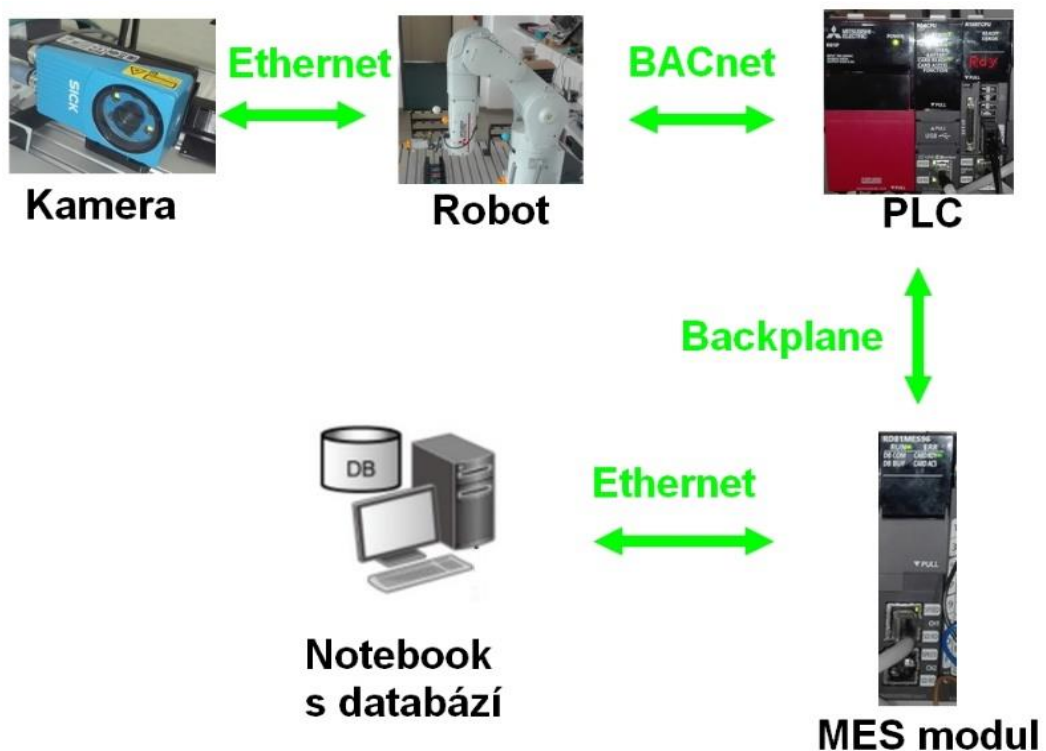
Obrázek 2.1 – Vývojový diagram

2.2 Blokové schéma

Blokové schéma značí, jaká zařízení mezi sebou komunikují a jakou komunikaci používají. Jako miniatury byly použity fotky zařízení, až na miniaturu počítače s databází, kde jsem použil obrázek z manuálu kvůli naznačení, že v notebooku je uložena databáze.

Komunikace mezi kamerou a robotem a komunikace mezi počítačem s databází a MES modulem se provádí klasickým Ethernetem. Komunikace mezi Robotem a PLC se provádí za pomoci *BACnet*. *BACnet* je komunikace, která je přímo určená pro použití v průmyslu a díky této komunikaci bylo například snadné posílání dat mezi robotem a PLC (např. informace o poloze dílu.)

Komunikace mezi PLC a MES modulem se provádí pomocí *Backplane*, MES modul je jen přidružený modul k PLC. Díky tomuto typu komunikace může MES modul pracovat s registry z PLC, což je značné zjednodušení. Další věcí je, že díky této architektuře může být MES modul v jiné síti než PLC. Tato výhoda bude podrobněji popsána v kapitole o porovnání MES systémů.



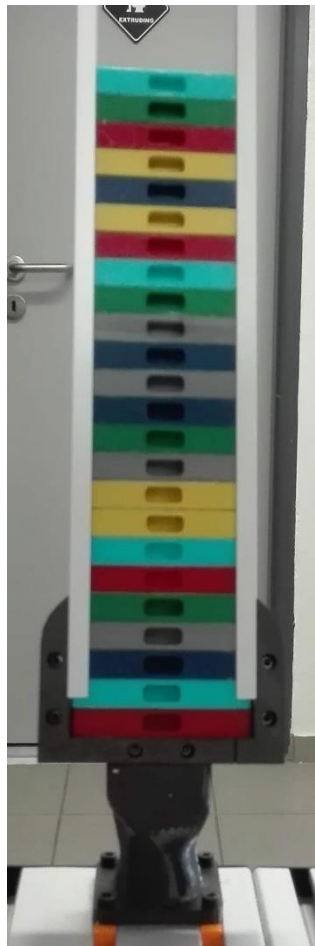
Obrázek 2.2 – Blokové schéma

2.3 Zásobníky

Pro tuto úlohu byl potřeba zdrojový zásobník, odkud si robot díly bral a úložné zásobníky, kam díly ukládal, těch bylo potřeba 6 pro 6 barev, každý zásobník byl pro 4 díly.

Zdrojový zásobník

Podstavec tohoto zásobníku byl vytisknut na 3D tiskárně a pro uložení dílu a jejich následný sjezd dolů byly použity kovové lišty. Výhodou tohoto zásobníku je, že může být adaptovatelný i na jiné díly, horní část je odnímatelná a může se na ní namontovat horní část pro díly jiné velikosti.



Obrázek 2.3 – Zdrojový zásobník s díly

Úložné zásobníky

Tyto zásobníky byly také vytisknuty na 3D tiskárně. Po vytisknutí je bylo potřeba opracovat (opilovat), aby tam díly správně zapadly. Jedná se o 2 objekty, kde každý objekt obsahuje 3 zásobníky. Proto bylo potřeba zásobníky namontovat ve správné vzdálenosti, aby i vzdálenost mezi objekty byla stejná jako vzdálenost mezi jednotlivými zásobníky v rámci objektu.



Obrázek 2.4 – Úložné zásobníky

2.4 Díly do zásobníku

2.4.1 Příprava dílů

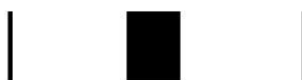
Díly byly vytisknuty na 3D tiskárně. Bylo vytvořeno 24 dílů, vždy 4 díly pro každou barvu. Díly bylo nutné opracovat, tak aby dobře sjížděly ve zdrojovém zásobníku a i dobře pasovaly do ukládacího zásobníku. Ukládací zásobník bylo nutné opracovat též. Zásobníky byly také vytisknuty na 3D tiskárně.



Obrázek 2.5 – Díl

2.4.2 Značení dílů

Každý díl bylo nutné označit tak, aby kamera identifikovala, že se jedná o díl, který je určen pro tento úkon a jakou barvu má díl. K tomu posloužil kód, který se nalepil na boky dílu. Kódem byl bílý čtverec s dvěma bílými čarami na stranách. Tyto objekty byly na černém pozadí. Čtverec sloužil k identifikaci objektu a vzdálenost čar sloužila k identifikaci, jakou barvu má daný díl. Z druhé strany byl pouze čtverec. Pokud robot kameře ukázal stranu se čtvercem, robot díl otočil na druhou stranu.



Obrázek 2.6 – Označení dílů

Na obrázku 2.6 jsou černé objekty na bílém pozadí, nakonec to bylo kvůli lepšímu čtení pro kameru zinvertováno na bílé objekty na černém pozadí.

Tyto štítky bylo nutné na díly nalepit, k tomu byl použit lepicí papír, který se používá v modelování. Tento druh papíru se namočí a poté je možné ho vyjmout z podkladu a nalepit na díl. Kvůli ochraně proti setření bylo pak tyto díly nutné přelepit matnou lepicí páskou. Matná lepicí páska je nutná, protože při použití normální pásky by se pak díl kameře lesknul a kamera by měla problémy s rozeznáním obrazců.



Obrázek 2.7 – Štítky na kódové označení dílů

2.5 Konfigurace síťových adaptérů

Veškerá komunikace bude probíhat síťově protokolem TCP/IP. Je použito adresování IPv4, které je obvyklé u PLC. Tudiž bylo nutné nastavit síťový adaptér na mém notebooku a nastavit adresu v MES modulu. Adresy bylo nutné nastavit tak, aby byly kompatibilní s adresací, která už byla vytvořena pro robota, PLC a kameru. Tato adresace je: **172.16.16.X/24**.

Tabulka 2.1 – IP adresy jednotlivých komponent

Komponenta	IP adresa
PLC	172.16.16.1
MES modul	172.16.16.2
Robot	172.16.16.20
Kamera	172.16.16.30

2.6 Kamera

Kamera je pro žádanou činnost velice důležitá. Slouží k rozpoznání jednotlivých dílů, posílá zjištěné informace robotovi, který pak s těmito informacemi dále pracuje, vykonává činnost či posílá informace databázi. Pracoval jsem s kamerou typu *VSPM-6F2113* od české firmy *SICK AG*.



Obrázek 2.8 – Kamera VSPM-6F2113

Tabulka 2.2 – Důležité údaje o kameře

Technologie	2D záznam jednotlivých snímků a jejich následná analýza
Senzor	Maticový senzor CMOS, odstíny šedé
Pracovní vzdálenost s integrovaným osvětlením	50 mm ... 200 mm
Rozlišení	640 px × 480 px

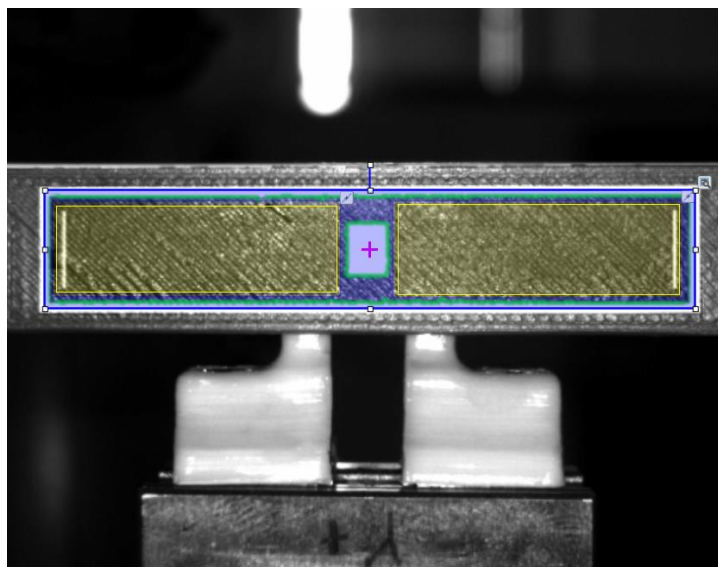
Software pro konfiguraci kamery

Pro konfiguraci kamery pro daný účel se používá *SOPAS Engineering Tool*. Tento SW je volně dostupný na internetových stránkách výrobce. Po instalaci je nutné přidat ovladač k danému zařízení. V našem případě to je *InspectorPIM60 – V2.0.0*. Poté, pokud je notebook připojený do sítě a správně nakonfigurován, je možné se připojit a nastavovat danou kameru. Po otevření editovacího okna se otevře obrazovka, kde je možné kameru konfigurovat.

2.6.1 Konfigurace kamery

Konfigurace kamery se provádí za pomoci vzorového obrázku. Kamera si zaznamená referenční obrázek a na tomto obrázku se pak manuálně nastaví, co všechno má kamera rozpoznávat.

Pro tento účel se tedy musel vyznačit bílý obdélník, který ohraničuje štítek, pak prostřední čtverec, který značí, zda se jedná o správný díl a postranní čáry, jejichž vzdálenost značí, o jaký díl se jedná.



Obrázek 2.9 – Referenční snímek

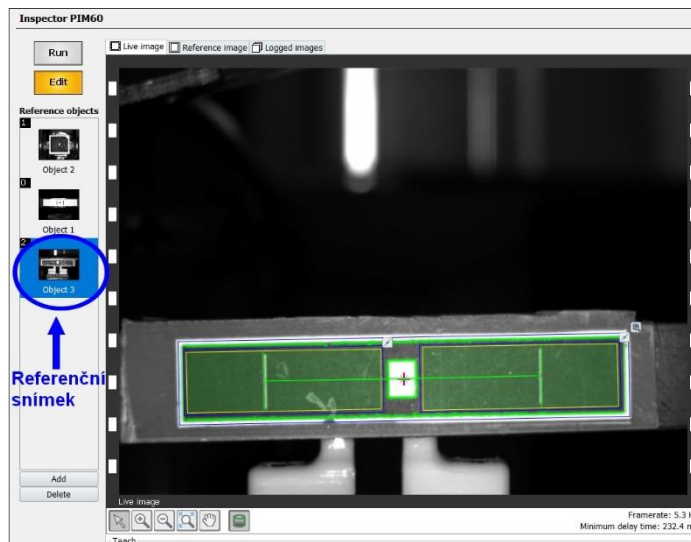
2.6.2 Kontrola dílů

Díly bylo nutné zkontrolovat, zda kamera nemá problém s jejich přečtením a také musela proběhnout kontrola, zda ukazuje správnou vzdálenost mezi díly. Robot se připravil do polohy před kamerou, kdy kamera správně čte díl a jednotlivé díly se pak nasazovaly do svěráku a na počítači bylo kontrolováno, jak kamera rozeznává daný díl.



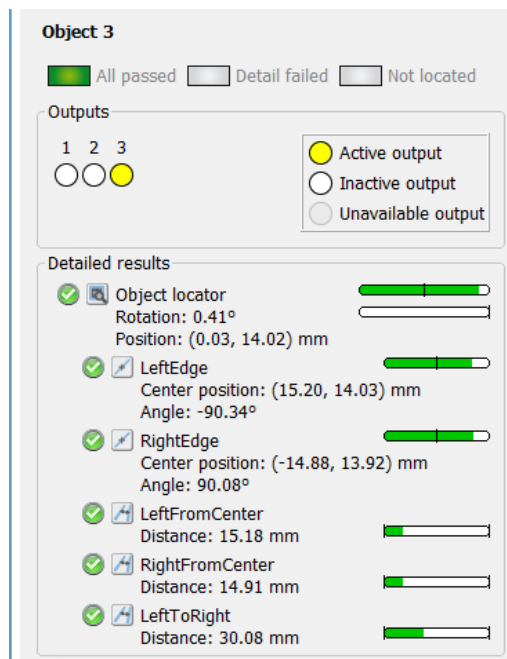
Obrázek 2.10 – Kontrola jednotlivých dílů

Na obrázku 2.11 je snímek dílu z kamery. Tento snímek kamera porovnává s referenčním snímkem a podle objektů, které jsou nastaveny v referenčním snímku, získává údaje o dílu.



Obrázek 2.11 – Snímek dílu z kamery

Výsledné parametry o dílu z kamery je přímo možné vidět v programu, tyto výsledky se zasílají robotovi, který s nimi dále pracuje.



Obrázek 2.12 – Výsledné parametry dílu z kamery

2.6.3 Parametry získané z kamery

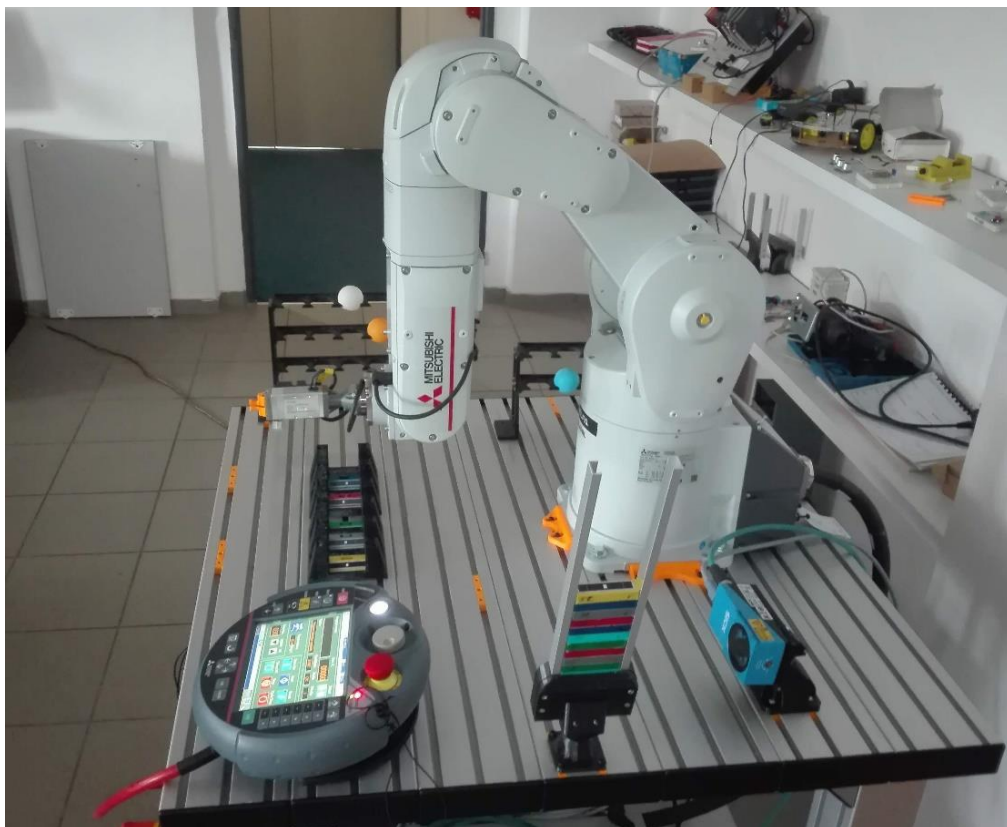
Po vyfotografování snímku kamera zpracuje daný snímek a získá z něj parametry, které charakterizují daný díl. Tyto parametry jsou posílány robotovi, který s nimi dále pracuje.

Tabulka 2.3 – Parametry získané z kamery

Parametr	Informace kterou nese
iK20_CAM_LocatorScore	Shoda v % s čtverečkem z referenčního obrázku
iK20_CAM_LocatorX	Parametr středu na vodorovné ose
iK20_CAM_LocatorY	Parametr středu na svislé ose
iK20_CAM_LocatorAngle	Naklonění čtverce
iK20_CAM_LeftScore	Shoda v % s levou čárou z referenčního obrázku
iK20_CAM_LeftPosition	Pozice levé čáry od čtverce
iK20_CAM_RightScore	Shoda v % s pravou čárou z referenčního obrázku
iK20_CAM_RightPosition	Pozice pravé čáry od čtverce
iK20_CAM_LeftToRight	Vzdálenost mezi levou a pravou čárou

2.7 Robot

Robot, přesněji robotické rameno, je zařízení, které vykonává fyzickou práci v žádané činnosti. Robot, který tuto práci vykonává, je **MELFA RV-4FRL**. Robot má v základu 6 os, je možnost přidání dalších, ale pro moje účely bohatě stačilo 6 os. Softwarem pro komunikaci s robotem je **RT ToolBox3**, který je stejně jako robot od firmy *Mitsubishi*.



Obrázek 2.13 – Robot MELFA RV-4FRL s pendantem

Naučení pozic

Základem pro vytvoření programu robota je naučení pozic, kam se má pohybovat. Jsou dvě cesty, buď naučit úplně všechny pozice, nebo naučit jen pár pozic a poté si pomoci odečítáním/přičítáním souřadnic jednotlivých os. Z praktického hlediska je rozhodně lepší použít funkci, kdy, pokud je to možné, se pouze odečte hodnota pro jednu z os. Protože při přílišném počtu pozic to začne být chaotické a je jednodušší udělat chybu, například v pohybu na špatnou pozici, což může mít i někdy fatální následky a také je mnohem složitější tuto chybu objevit. V případě, že je potřebné robota přemístit do jiného místa, je často také nutné „nakrokovat“ tento přesun do více pozic, protože bez mezipozic se robot přetočí mimo rozsah svých os.

Psaní programu

Programovací jazyk pro programování robota je podobný klasickým programovacím jazykům, jako je *C*, *Python* apod. Co je odlišné od klasického programovacího jazyka, je fakt, že před každou proměnnou je třeba psát písmenko na začátku, které značí, co daná proměnná je. Pro pozice musí být na začátku *p*, pro pozice v souřadnicovém systému *JOINT* je to *j*. Proměnné v programu musí mít na začátku písmeno *m*.

2.7.1 Programy robota

Celý program robota je z důvodu jednodušší údržby, testování a přehlednosti rozdělen do více podprogramů. Program obsahuje hlavní program, 3 podprogramy, které tvoří činnost robota, program ve kterém jsou definovány funkce a posledním programem je program, kde jsou nahrány globální proměnné, které se používají ve více programech.

Veškeré popisy pozic platí pro souřadnicový systém *TOOL*, který se váže ke svěráku.

Při změně pozic byly použity příkazy *Mov* a *Mvs*. *Mov* se pohybuje po nejkratší možné cestě, zatím co *Mvs* jede přímo. *Mvs* se tak používá, pokud se má robot pohybovat jen po jedné ose, například při zajištění do zásobníku.

Tabulka 2.4 – Programy robota

Program	Funkce programu
MHBMAIN	Hlavní program, z něj jsou volány jednotlivé podprogramy
MHBTAKE	Podprogram, který vybírá díl ze zásobníku a přesune robota do polohy, kdy kamera čte díl
MHBCAM	Podprogram, kde kamera snímá díl a informace zasílá PLC, které s nimi dále pracuje
MHBSTOCK	Podprogram, kdy robot uloží díl na správnou pozici
MHBFCNS	Podprogram, který obsahuje funkce, které robot volá. Zjednodušení, že není nutné definovat funkce pro každý program
PRGUSR	Program, který obsahuje globální proměnné pro tento účel, ale obsahuje i globální proměnné, které využívá každý program

MHBMMAIN

Tento program byl vytvořen až na konec. Je ze všech programů nejjednodušší a nejkratší, protože jeho činnost spočívá jen v aktualizaci stavů na prvotní stavy a cyklickém opakování jednotlivých podprogramů.

MHBTAKE

Tento program pro vyjmutí dílu ze zdrojového zásobníku obsahuje 4 pozice. První pozice *pTakeIn00* je počáteční, do této pozice je hlavně nutné najet při běhu programu po uložení dílu do zásobníku, protože při absenci této pozice by došlo k přetočení robota. Druhá pozice *pTakeIn01* je mezipozicí pro úspěšný přesun před zdrojový zásobník dílů. Poté se robot přesunuje do pozice *pTakeIn*, která je umístěná v zásobníku a slouží k uchopení dílu. Nejdříve se najíždí to této pozice, která je jiná v ose z. A poté se se přesunuje jen po z-ové ose do této pozice. Tím je zařízeno správné najetí do dílu a jeho správné uchopení. Poté následuje roztáhnutí svěráku. Pokud zde není díl, či dojde k jiné chybě při uchopení dílu, program se přerušuje. Při úspěšném uchopení dílu program znovu po z-ové ose vyjíždí ze zásobníku a přesunuje se do pozice *pCAM*. Tím je tento program ukončen a hlavní program volá podprogram *pCAM*.

MHBCAM

Jde o program, ve kterém se vykoná snímání kamery a nahrání parametrů získaných ze snímku do PLC. Program nemá žádnou pozici, protože do pozice před kamerou ho přesune *MHBTAKE* a po vykonání tohoto programu následuje podprogram *MHBSTOCK*, který začíná na pozici u kamery. Program začíná definicí portů pro komunikaci s kamerou a zahájením komunikace. Poté se jednotlivé porty otevrou. Následuje vybrání vzorového snímku, se kterým kamera bude porovnávat získaný snímek a spuštění kamery (kamera vyfotí snímek). Parametry, které kamera zjistí ze snímku, se načtou do proměnných v robotovi. Parametr *LeftToRight* je pak použit v podmínce, která zjišťuje, zda je díl správně natočen, pokud je tento parametr roven 0, tak to znamená, že byl na kameru díl natočen obráceně (tj. nemá štítek s informací o barvě dílu) a díl je otočen a kamera vyfotografuje díl. Při správně otočeném dílu se aktualizují parametry z kamery v PLC a uzavírají se porty pro komunikaci.

MHBSTOCK

Tento program slouží k uložení dílu na správné místo ve správném zásobníku. Program začíná definicí proměnných, které slouží pro načtení informací z PLC o pozici dílu. Poté za pomoci mezipozic je robot přesunut do polohy nad prvním zásobníkem. Když je robot nad zásobníkem, dochází k načtení informace o zásobníku a jeho pozici v něm pro daný díl. Když jsou informace načteny, je vypočtena správná pozice. Nejdříve se najíždí nad správný zásobník, tudíž se robot pohybuje po ose y. Po najetí nad správný zásobník dochází k výpočtu hodnoty pro osu z a robot po této ose najíždí do zásobníku. Při najetí do zásobníku je omezena rychlost na 40 % z bezpečnostních důvodů. Jakmile je na správné pozici, zavírá se svěrák (tj. uvolňuje se díl) a robot vyjíždí zpátky. Na konec se variabilní pozice, která slouží pro uložení dílu, přepisuje zpět na výchozí hodnotu nad 1. zásobníkem a rychlost se znovu nastavuje na 100 %.

Výpočet správné pozice pro daný díl:

$$PoziceVar.Y = PoziceVych.Y - (MStock \times 49,9) \quad (2.1)$$

PoziceVar.Y - parametr cílové pozice v ose y

PoziceVych.Y - parametr výchozí pozice v ose y

MStock – číslo zásobníku

$$PoziceVar.Z = PoziceVych.Z - 60 + ((MNumberofpart - 1) \times 12,6) \quad (2.2)$$

PoziceVar.Z - parametr cílové pozice v ose z

PoziceVych.Z - parametr výchozí pozice v ose z

MNumberofpart – kolikátý díl je to v daném zásobníku

Tabulka 2.5 – Čísla zásobníku pro jednotlivé barvy

Číslo zásobníku	Barva
0	Žlutá
1	Šedá
2	Zelená
3	Červená
4	Modrá
5	Tyrkysová

MHBFNCS

Jedná se o program, ve kterém jsou definovány funkce, které jsou použity v různých podprogramech. Program obsahuje funkci, která aktualizuje stav programu, tato aktualizace například slouží pro informování PLC, které díky této informaci sepne bity pro zápis parametrů do databáze a činnosti, které následují po zápisu do databáze. Další funkcí je aktualizace parametrů z kamery v PLC. Posledními funkcemi jsou funkce pro svěrák, patří mezi ně otevření svěráku, zavření svěráku a nastavení svěráku do výchozí pozice.

PRGUSR

Program pro definici globálních proměnných, které je možné použít ve více programech najednou. Jsou zde definovány globální proměnné, které se využívají pro veškeré programy i speciální globální proměnné, které se využívají v této bakalářské práci.

2.8 PLC

Pro tuto bakalářskou práci bylo použito PLC **R04CPU** od firmy *Mitsubishi*. PLC v této práci slouží prakticky jako přestupní stanice mezi robotem a MES modulem. Do PLC se ukládají parametry z kamery, které si MES modul načítá a z PLC si robot načítá informace pro výpočet výsledné pozice. V PLC jsou i „triggery“ (spouštěcí bity), které zajišťují zápis do databáze až po načtení nového snímku kamerou.



Obrázek 2.14 – PLC R04CPU

2.8.1 Program

Program byl napsán v příčkovém diagramu (Ladder). Dá se říci, že program se sestává z 2 částí. V první části probíhá nahrání parametrů z kamery, které jsou přeposlány přes robota do registrů v PLC. Díky funkci *INT2FLT* je parametr změněn z celočíselného formátu (integer) do reálného čísla (float). Pak nastane korekce desetinné čárky o 2 řády doleva, u porovnávacích údajů se tím získá údaj v procentech a pro vzdálenostní údaje se tím získá hodnota v mm. V druhé části probíhá spínání bitů, které spouštějí operace v MES modulu a nahrání informací dílu do robota. První bit *M1000* spustí zápis parametrů z kamery do tabulky všech zápisů v databázi. Druhý bit *M1001* spustí zápis z databáze do registrů v PLC. Díky změně registru *iK20_DB_LefttoRight* se spustí zápis do tabulky pro zásobník, který odpovídá barvě daného dílu. Po sepnutí třetího bitu *M1002* se zapíší informace o dílu do robota. Přesnější řečeno se tato akce spouští časovačem, který ale spouští bit *M1002*. Mezi jednotlivými operacemi je časovači naprogramován rozdíl 3 sekund.

Výměnná dat mezi robotem a PLC probíhá speciální kombinací, kdy je za pomoci programu od firmy *Mitsubishi* získán parametr, který nahraje žádaná data do robota a proměnná, která slouží k využití parametru v programu robota. Například hodnota se v PLC nahraje do registru *U3E0\HG157*, tuto hodnotu v robotovi získáme zadáním *M_inW(12512)*. V této proměnné je uložena hodnota z PLC.

2.9 MES modul

MES modul slouží v tomto zadání pro komunikaci s databází. MES modul získává informace z databáze a také informace do databáze zapisuje. Pro roztřídění dílů jsem pracoval s MES modulem *RD8IMES96* od firmy *Mitsubishi*. Práce s MES modulem probíhala v prostředí *MELSEC iQ-R MES Interface Module*, který byl stejně jako výrobek od firmy *Mitsubishi*.



Obrázek 2.15 – MES modul RD81MES96

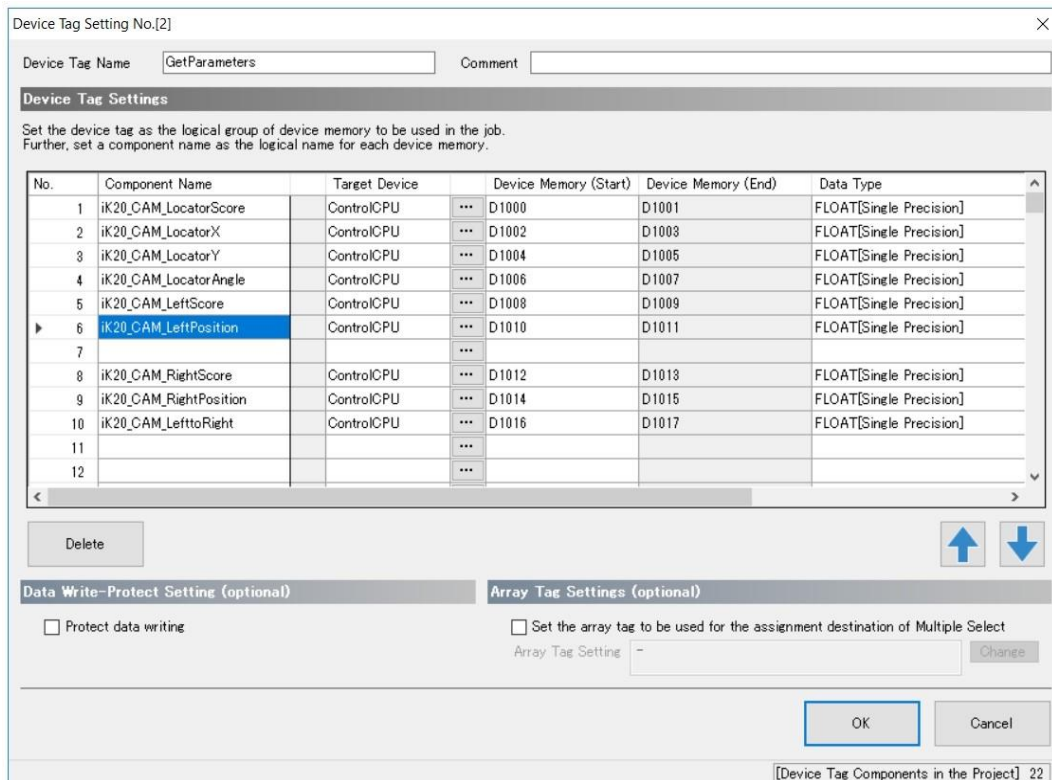
2.9.1 Konfigurace modulu

Bylo nutné nastavit komunikaci mezi modulem a databází, která se nacházela v počítači. V MES modulu tedy bylo nutné zřídit komunikaci s databázovým serverem a bylo nutné také zřídit komunikaci prostředí s MES modulem. Po úspěšném vyzkoušení komunikace bylo možné začít pracovat s MES modulem.

2.9.2 Vytvoření „tagů“ zařízení

Aby bylo možné pracovat s informacemi z PLC, bylo nutné vytvořit „tagy“ zařízení. Slovo tag se dá přeložit jako štítek. Díky těmto „tagům“ pak je v činnostech možné zapisovat údaje z PLC do databáze nebo údaje z databáze zapsat do registrů v PLC. V případě této bakalářské práce do nich taky bylo nutné zapsat bity, které spouštěly jednotlivé činnosti. V jednotlivých „tagách“ je možné nastavit více komponentů, pro přehlednost byly jednotlivé komponenty rozděleny do více „tagů“ podle jejich použití. Pro nastavení komponentu v „tagu“ bylo nutné komponent pojmenovat, nastavit jeho datový typ a na jakém místě v paměti PLC tento komponent začíná. Podle datového typu už se automaticky nastavil, kde daný komponent končí. Nastavení je identické k nastavování komponent pro PLC.

„Tag“ může být také ochráněn proti přepsání (*Data Write-Protect Settings*) a z daného „tagu“ může být vytvořeno pole (*Array Tag Settings*)



Obrázek 2.16 – Nastavení komponent v "tagu"

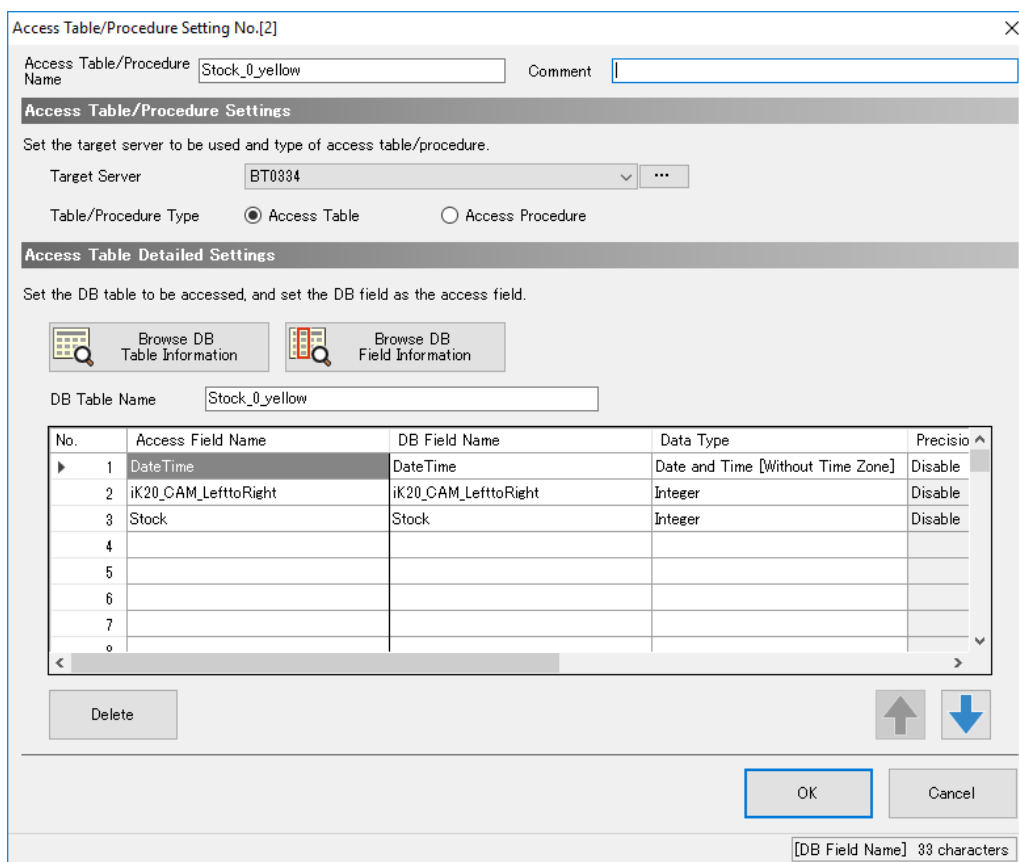
Tabulka 2.6 – Použité "tagy"

Jméno "tagu"	Použití
CycleStep	Bit, který značí dokončení snímání kamerou, může tak začít nahrávání do databáze
GetParameters	Získané parametry z kamery
DB_to_Variable	Bit, který spouští zápis z databáze do PLC
SendParameters	Parametry z databáze do PLC (nakonec využít pouze jediný komponent)
Position_to_PLC	Informace o zásobníku a počtu dílů v něm

2.9.3 Nastavení tabulek z databáze

Nastavení tabulek v MES modulu probíhá v možnosti *Acces Table/Proc. Settings*. Po pojmenování dané tabulky je nutné zadat databázový server, ve kterém se chtěná tabulka nachází. V možnosti *Browse DB Table Information* je nutné vybrat danou tabulku. V možnosti *Browse DB Field Information* se pak vyberou sloupce, které se použijí. Není problém vybrat všechny a použít jen některé. V případě této práce, kdy sloupců není tolik, to problém není, ale

v případě velkých databází, které obsahují mnoho sloupců, je pro přehlednost lepší si do MES modulu načíst jen sloupce, se kterými se bude pracovat.



Obrázek 2.17 – Načtení tabulky z databáze do MES modulu

2.9.4 Nastavení činností MES modulu

Činnosti, které MES modul provádí, se nastavují v možnosti *Job Settings*. Nastavení činnosti se provádí až v 5 krocích (při rozšířené konfiguraci až v 7 krocích). Pro činnosti, které byly potřeba vytvořit pro tuto bakalářskou práci, jsem musel měnit nastavení jen ve 2 krocích. Zbytek jsem nechal na výchozích hodnotách.

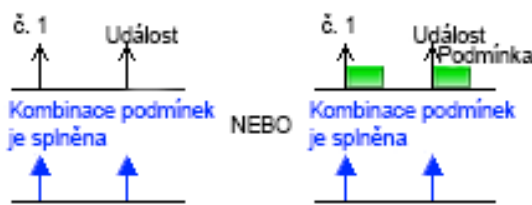
Job Configuration

Slouží jen pro výběr hlavní konfigurace (*Main Configuration*) nebo rozšířené konfigurace (*Extended configuration*). V případě této práce stačila hlavní konfigurace, která je i nabízena jako výchozí. Nebylo tedy nutné konfigurovat, co má MES systém vykonat před výkonem hlavní činnosti (*pre-processing*) ani po vykonání hlavní činnosti (*post-processing*). V

pre-processingu je možné nastavit operaci uvnitř databázového serveru a v *post-processingu* je možné nastavit operaci uvnitř MES modulu.

Trigger Conditions

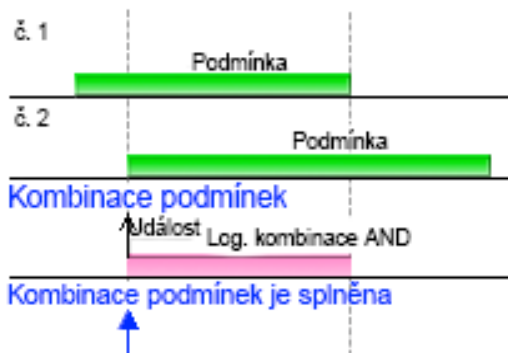
V této části konfigurace se nastavuje, co spustí danou činnost, nastaví se tzv. „*trigger*“. Může to být jedna událost, dvě události, kdy každá z nich spustí činnost, také se může spuštění činnosti nastavit událostmi v logické kombinaci (OR a AND). Logická kombinace OR se od spuštění dvěma událostmi liší v tom, že zatímco při dvou událostech se událost spustí vždy, když se splní jedna z podmínek, u logické kombinace OR se událost nespustí při splnění druhé, pokud je první stále v pravdě a podmínka pro spuštění je tak stále splněna. Poslední možností je kombinace, která obsahuje předpoklad a událost. Činnost započne po nastání události, ale jen při splněním předpokladu.



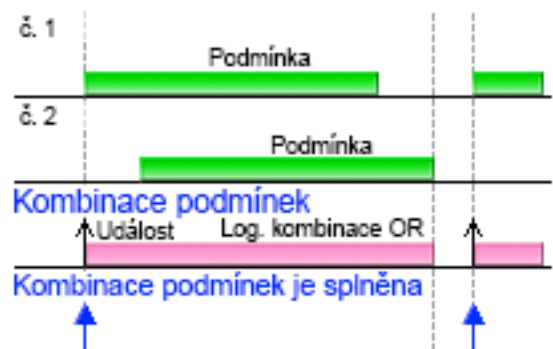
Obrázek 2.18 – Spuštění činnosti jednou událostí



Obrázek 2.19 – Spuštění činnosti dvěma událostmi



Obrázek 2.21 – Spuštění činnosti událostmi v log. kombinaci AND



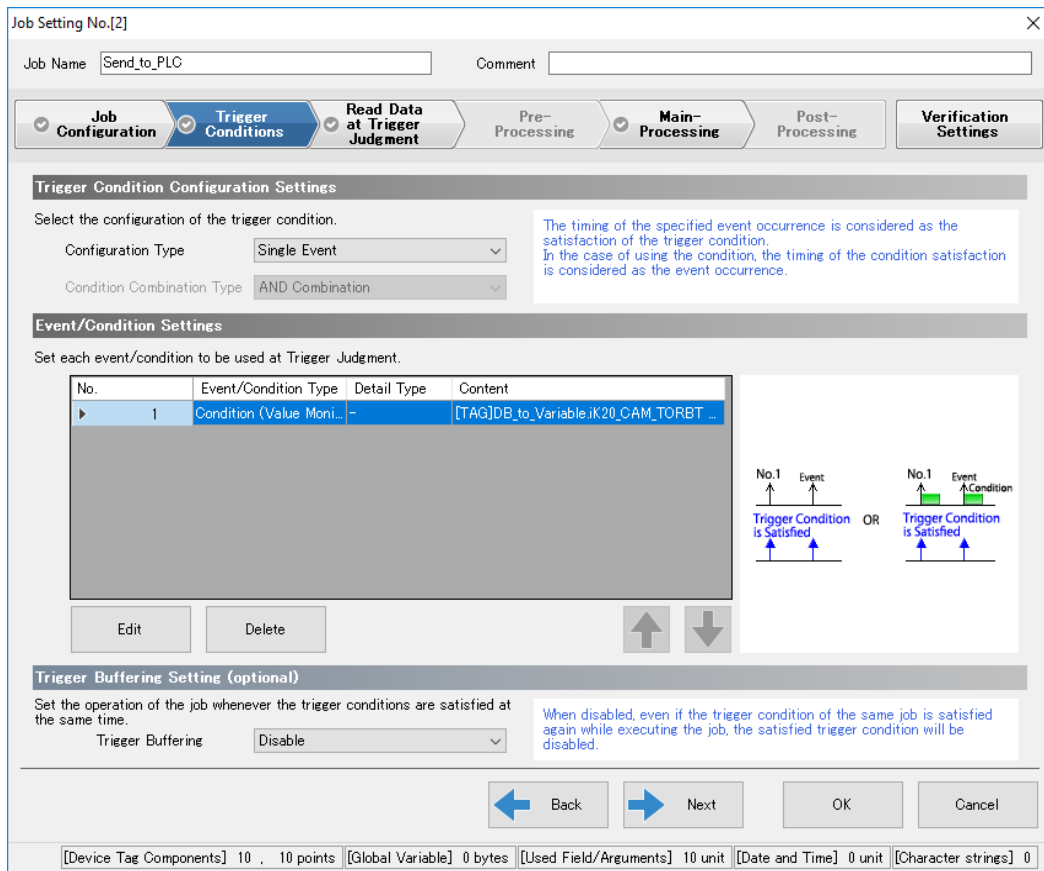
Obrázek 2.20 – Spuštění činnosti událostmi v log. kombinaci OR



Obrázek 2.22 – Spuštění činnosti událostí při splnění předpokladu

V možnosti více událostí je možné zadat pouze dvě události. Pokud je nutné činnost spustit více událostmi, musí se daná činnost zkopírovat a použít jiné spouštěcí události pro stejnou činnost. V této práci bylo použito pouze spuštění jednou událostí nebo dvěma událostmi. Jednou událostí byla spuštěna činnost pro zápis do tabulky všech zápisů a při čtení z této tabulky. Dvěma událostmi pak byla spuštěna činnost zápisu do tabulek pro jednotlivé zásobníky a následné nahrání informací z této tabulky do proměnných v PLC. Výjimku tvořily některé barvy, kdy 2 díly měly stejnou vzdálenost mezi čarami a tak pro 4 díly byly pouze 3 vzdálenosti, v druhé činnosti zbyla tedy pouze jedna podmínka. Zápis do tabulky všech zápisů a následné čtení z ní bylo provedeno spuštěním bitů v PLC, které následují po snímání dílu kamerou. Zápis do tabulky pro jednotlivé úložné zásobníky byly provedeny proměnnou *iK20_DB_LefttoRight*, která je celočíselným vyjádřením vzdálenosti mezi čarami a je získána z tabulky všech zápisů v databázi. Podle její hodnoty se spustí zápis do tabulky pro danou barvu (pro daný zásobník).

Další věc, kterou je v tomto kroku nutné nastavit, je zda se činnost začne provádět i při případu, že se činnost provádí a spouštěcí událost nastane znovu. Při zakázání (disable) se činnost nezačne při jejím vykonávání provádět znovu. Při povolení (enable) se může provádět znovu, i když je zrovna prováděna. V mém případě je to vždy zakázáno, což je i výchozí nastavení, správně by stav opětovného spuštění podmínky při vykonání činnosti neměl nastat, ale pro jistotu je to zakázáno, protože by to mohlo způsobit případné problémy.



Obrázek 2.23 – Vytvoření spouštěcí podmínky pro zápis z tabulky všech zápisů do PLC

Read Data at Trigger Judgment

V tomto kroku se nastavuje typ přístupu. Výchozím typem je obecný přístup (*General acces*), ten znamená, že MES modul může přistupovat na jakékoli zařízení (i na databázi). V tomto přístupu se nastavuje, jak dlouhou dobu má MES modul číst data při přístupu po splnění spouštěcí podmínky. Výchozí hodnou je 100 ms, tato hodnota nebyla v této práci změněna, protože byla plně dostačující, avšak dostatečně krátká. Pak jsou zde vysokorychlostní přístupy (*High-Speed Accesses*), kdy MES modul při dané činnosti komunikuje jen s PLC, první je stejně jako obecný přístup pro nějaký časový interval, druhý přístup čte data pokaždé, kdy PLC provede skenování. Ve skenování si PLC aktualizuje stav vstupů a výstupů.

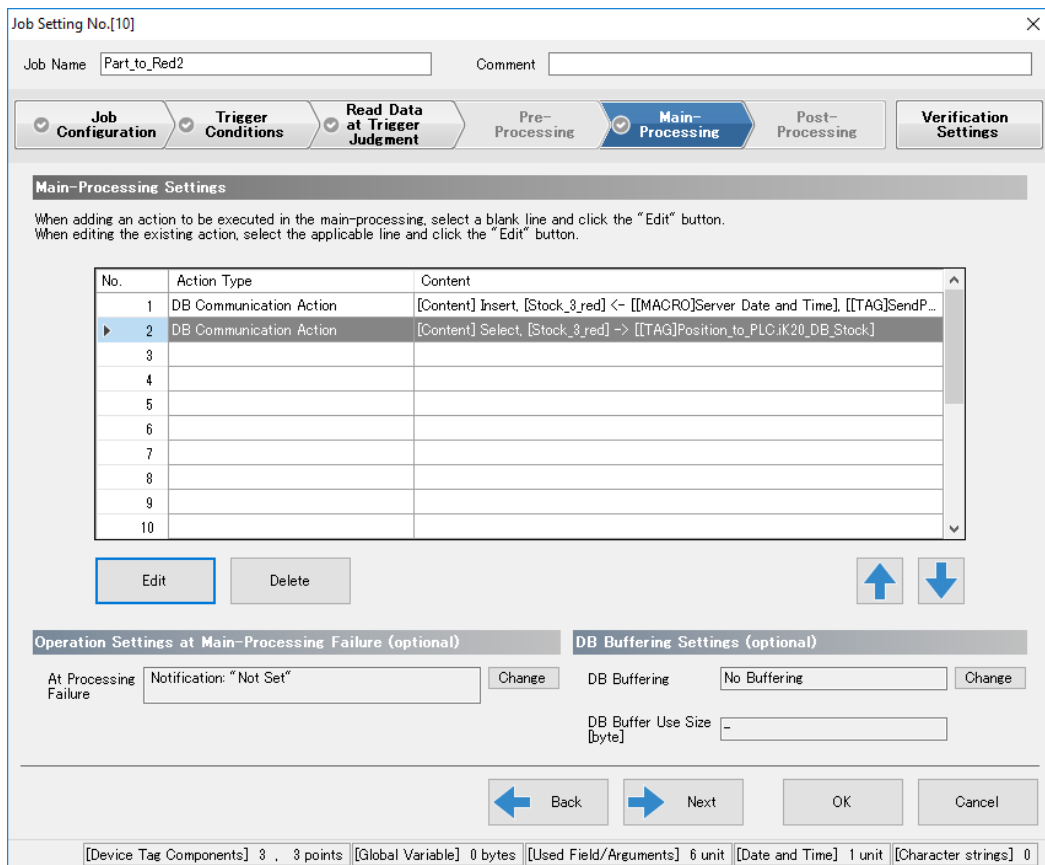
Main-Processing

Tato část je nejdůležitější částí konfigurace činnosti. Zde se nastavují operace, které provede daná činnost. Mezi možné operace patří komunikace s databází, komunikace s nějakým externím zařízením a vnitřní operace, kdy je možné pracovat s proměnnými, jak s proměnnými v PLC, ale také je možné pracovat s proměnnými, které jsou uloženy pouze v MES modulu.

V této práci byly použity pouze operace pro komunikaci s databází. Buď byly do tabulky zapisovány z hodnot proměnných v PLC, nebo byly do proměnných nahrávána data z tabulek v databázi. Při nahrávání z databázové tabulky bylo také nutné nastavit, aby byl vždy čten poslední zápis, protože díky absenci primárních klíčů nejsou údaje v databázi seřazeny podle času. Při nahrávání údajů z tabulek pro jednotlivé úložné zásobníky bylo také nutno nastavit, aby se do registru *iK20_DB_Parts_in_stock* nahrál počet záznamu v dané tabulce. Díky tomuto údaji pak robot ví, kolik už dílů je v zásobníku a může tak najet na správnou pozici. Poslání údaje o zásobníku, kam má robot díl uložit, je vytvořeno zapsáním čísla zásobníku do sloupce *Stock* v tabulce a tento údaj se pak nahrává do registru *iK20_DB_Stock*.

Je také možno nastavit notifikaci, pokud operace selže. To se provede tak, že je možné do nějaké proměnné nahrát obsah jiné proměnné, případně nějakou hodnotu. Příkladem může být bit *Error*, do kterého by se při chybě zapsala jednička a spustila by se sekvence událostí pro chybu. V této práci toto není řešeno, protože komunikace s databází byla bezchybná. V případě ostrého provozu by ale taková pojistka byla žádoucí.

V tomto kroku je také možné nastavit ukládání příkazů (*DB Buffering Settings*), to by znamenalo, že při nevykonání příkazů se příkaz uloží a při jeho možnosti se všechny příkazy vykonají. V této práci toto není nastaveno, protože zpětné vykonání příkazů by provedlo více škody než užitku. Informaci o dílu robot potřebuje okamžitě, a pokud ji nedostane, neví kam díl uložit, pokud by ji dostal zpětně, mohlo by například dojít k chybnému vyhodnocení a kolizi s už uloženým dílem.



Obrázek 2.24 – Výchozí okno pro nastavení operací u dané činnosti

Verification Settings

V tomto posledním kroku je možné nastavit tvorbu historie, kde se zaznamenávají činnosti. Historii je možno ještě povýšit záznamem detailního záznamu (*detailed log*). Druhou věcí, kterou je v tomto posledním kroku možno nastavit, je potlačení výstupu dat do cílového zařízení či cílového serveru při ověřování operace. Poslední možností, kterou je zde možno nastavit, je vypnutí činnosti. Nastavené operace pro činnost se nevykonají, přestože bude podmínka splněna. Tato funkce nachází své uplatnění, pokud je činnost momentálně nechťená, ale mohla by být žádaná v budoucnu, není poté potřeba činnost tvořit znovu či nahrávat jiný projekt, stačí vypnout toto nastavení a činnost se bude znovu provádět.

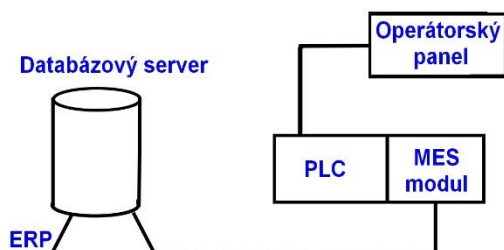
2.10 Porovnání MES systémů

Pro porovnání budou sloužit 2 typy MES systémů, protože funkcionality MES se dá dosáhnout nejen **MES modulem**, jak bylo ukázáno v této práci, ale také pomocí **operátorského panelu**. Tyto typy se liší architekturou, možnostmi, počtem HW a cenou. Každý systém přináší jiné výhody, takže výběr systému záleží na použití. Porovnání proběhlo mezi prostředím *MELSEC iQ-R MES Interface Module*, pomocí kterého se konfigurují MES moduly a prostředím *MELSOFT GT Designer3*, pomocí kterého se konfigurují operátorské panely.

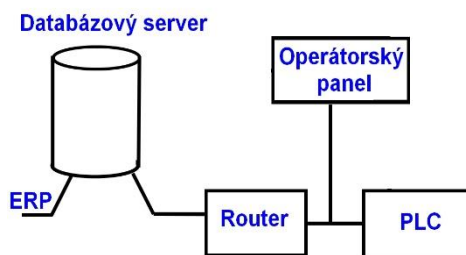
Možností pro velice základní MES funkce je i ukládání chyb ve výrobě na paměťovou kartu v PLC. Toto řešení je ale značně limitující.

2.10.1 Architektury systémů

Díky tomu, že MES modul je propojený s PLC přes Backplane, poskytuje možnost, že PLC a MES modul budou připojeny na různých sítích. Zatímco při operátorském panelu, musí být databázový server i výroba připojeny v 1 síti. Částečně se to dá vyřešit přidáním routeru, ale jedná se o zařízení navíc a stále to není tak účinné jako fyzické oddělení těchto sítí. Proto je MES modul při režimu, kdy se chce za pomoci ERP řídit celá výroba, mnohem výhodnější.

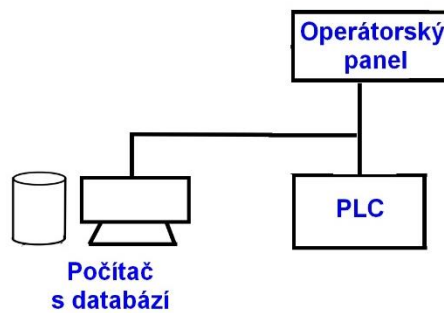


Obrázek 2.25 – Architektura MES systému při použití MES modulu



Obrázek 2.26 – Architektura MES systému při použití operátorského panelu

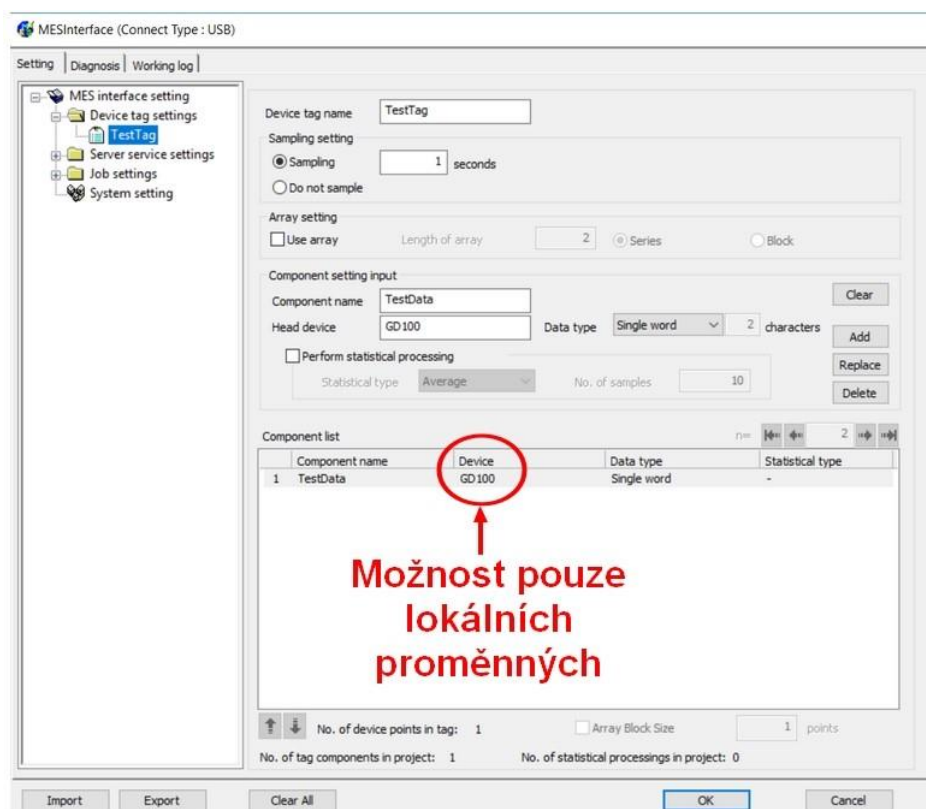
Avšak vytvoření MES systému pomocí operátorského panelu může být výhodné, pokud se nevyužívá ERP systém a jako databázový server se použije počítač přímo ve výrobě. Takový systém slouží jen pro záznamy ve výrobě a pro případné sledování kvality, úspěšnosti apod. V tomto případě je použití pomocí operátorského panelu plně dostačující.



Obrázek 2.27 – Zjednodušení MES systémů (bez ERP)

2.10.2 Načítání dat z PLC

Při použití MES modemu je načítání dat z PLC absolutně bezproblémové a můžeme použít data z PLC i pro zápis či čtení z databáze. Při MES systémech, které jsou tvořeny pomocí operátorského panelu, je to komplikovanější, tam mohou být použity pouze lokální proměnné a případné nahrání těchto proměnných se dá udělat až pomocí skriptu.



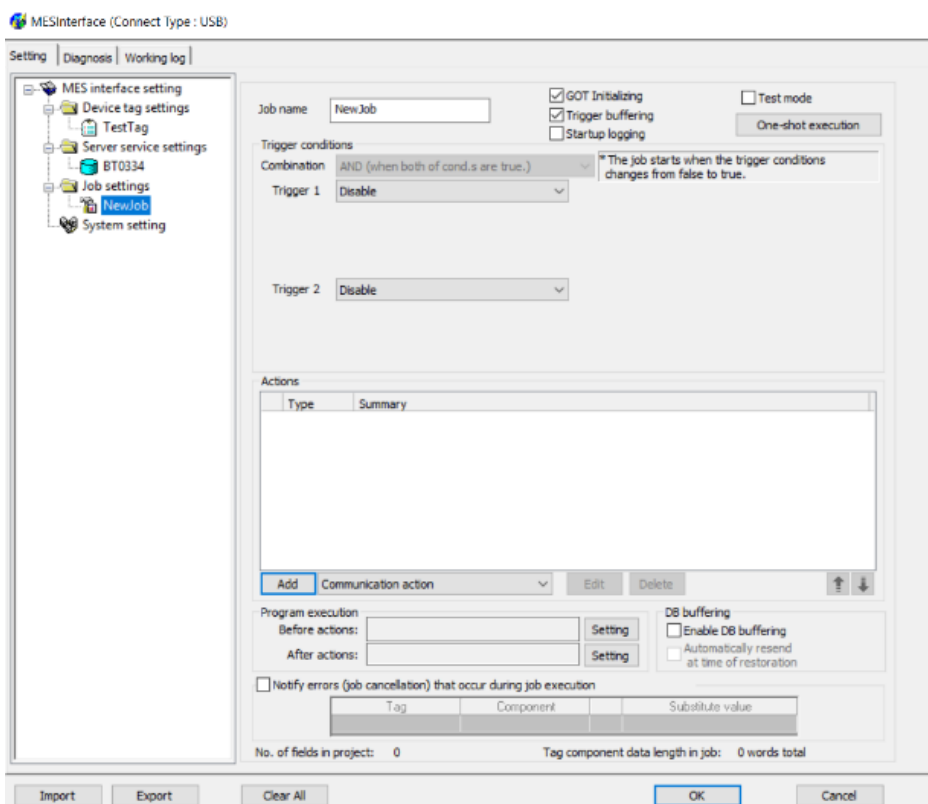
Obrázek 2.28 – Ukázka nemožnosti načtení dat z PLC přes OP

2.10.3 Cena

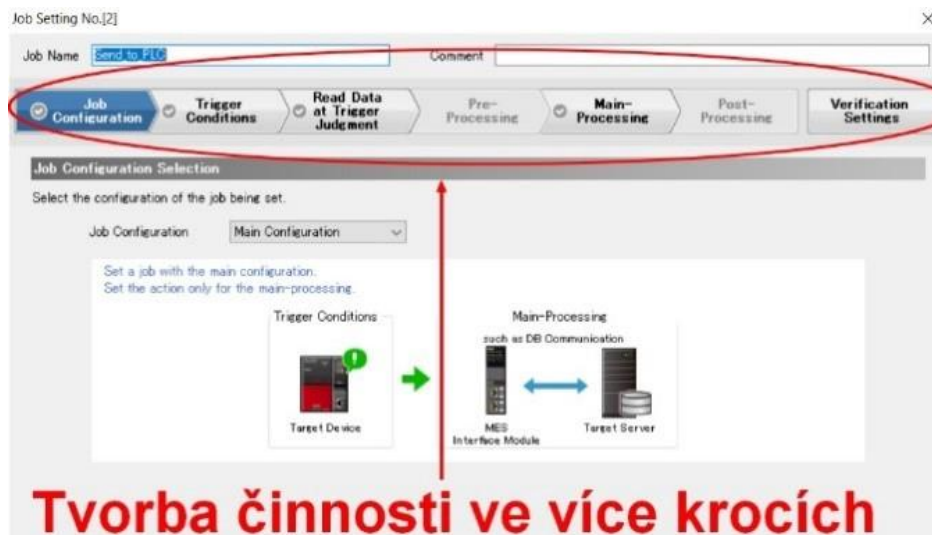
Zde naopak vychází mnohem lépe operátorský panel. MES modul je mnohem dražší než tvorba MES systémů pomocí operátorského panelu. Avšak při tomto rozmyšlení se musí počítat i s nutným časem, který programátor stráví s nastavením MES systému, kdy práce s MES modulem je rychlejší a čas je dnes velice cenná komodita. Společnost musí zvážit či se ušetřené peníze při koupi operátorského panelu vyplatí oproti času navíc, který s tím programátor stráví.

2.10.4 Uživatelská přívětivost

Práce s MES systémem v modulu a operátorském panelu je podobná, ale konfigurování v MES modulu je více intuitivní a přehlednější. Například zatímco vytváření činnosti probíhá v MES modulu ve více krocích, které za sebou přehledně navazují a například u tvorby spouštěcích „triggeru“ je i grafické znázornění, kdy bude v dané kombinaci spouštěcích signálů činnost spuštěna, v operátorském panelu veškeré nastavování probíhá v jednom okně.

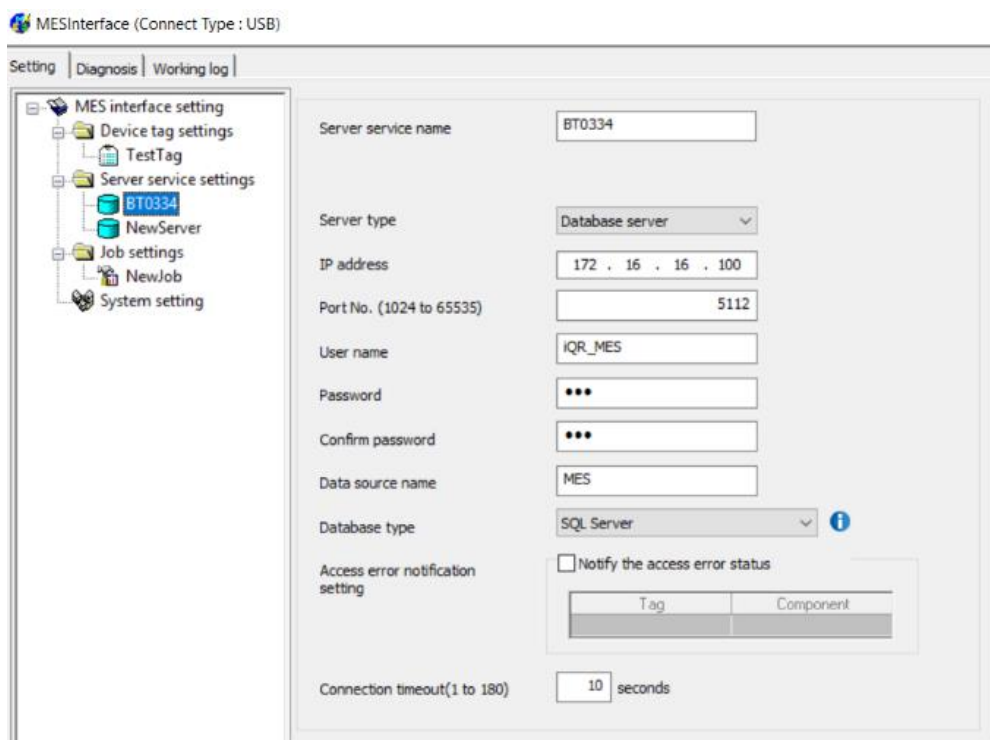


Obrázek 2.29 – Tvorba činnosti pro OP

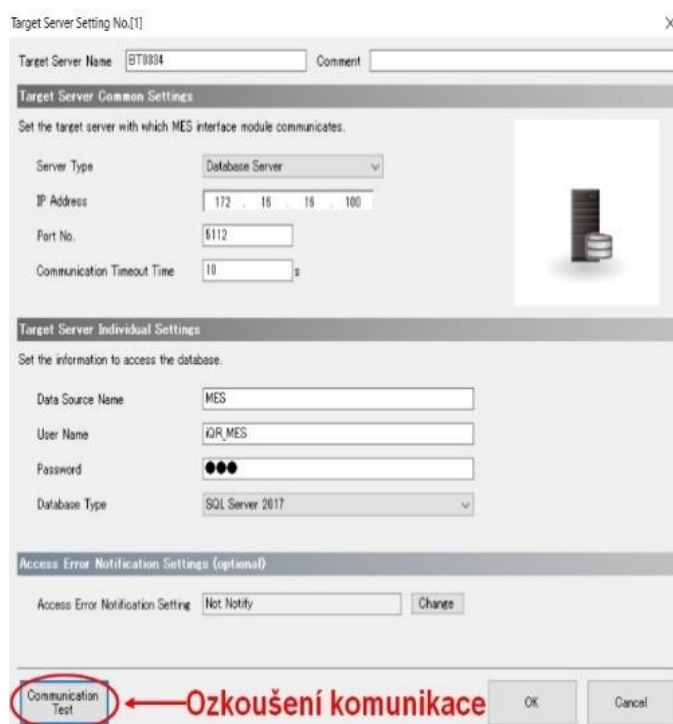


Obrázek 2.30 – Tvorba činnosti pro MES modul

Připojení k databázovému serveru je podle mě také lepší v programu pro MES modul. Hlavní výhodu vidím v ozkoušení komunikace mezi MES modulem a databázovým serverem, to v programu *GT Designer* pro OP není možné. Zde by odzkoušení muselo proběhnout až samotným vyzkoušením funkčnosti činnosti.



Obrázek 2.31 – Přidání databázového serveru pro OP



Obrázek 2.32 – Přidání databázového serveru pro MES modul

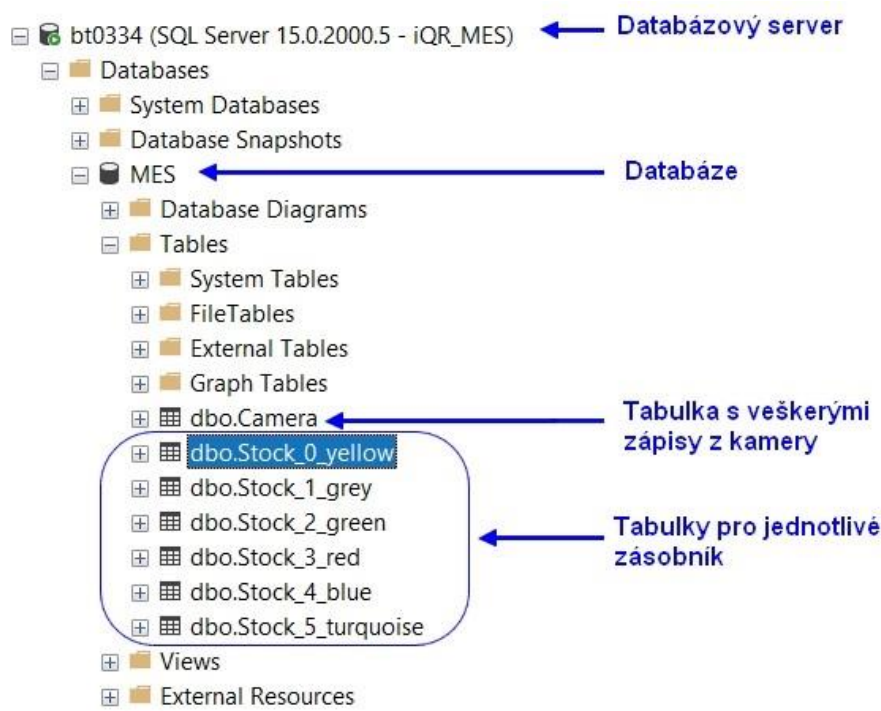
2.10.5 Závěrečné porovnání

Jak je možno vidět z výše uvedených porovnání, operátorský panel má prakticky jedinou výhodu, za to však velice důležitou výhodu a tou je cena. Pokud má MES modul vykonávat komplikované operace ve velké výrobě (zvláště za použití ERP) se však MES modul i tak vyplatí. Programátor stráví méně času s konfigurací, sítě jsou oddělené, takže bude docházet k méně problémům a odstraní se i případný problém při komunikaci s PLC, kdy operátorský panel může pro MES funkce používat pouze lokální proměnné a jejich nahrání do PLC se musí zajistit skriptem. Operátorský panel se díky nižší ceně vyplatí pro jednodušší úlohy, kdy například bude sloužit jen ke statistickému ukládání dat.

2.11 SQL databáze

SQL byla vytvořena v prostředí **MS SQL 2019** od firmy *Microsoft*. Po úspěšné instalaci jsem si vytvořil databázi **MES**, která je používána pro komunikaci s PLC. Byl vytvořen lokální účet, protože přihlašování pomocí lokálního účtu Windows by nebylo možné kvůli nemožnosti PLC dosáhnout autentizace. Tomuto účtu bylo také nutné dát oprávnění kvůli možnosti práce s tabulkou. Byla vytvořena tabulka, kde se zaznamenávala veškerá čtení z kamery. Pak bylo

vytvořeno 6 tabulek, které sloužily pro identifikaci barvy (tj. do jakého zásobníku se má díl uložit) a kolik už dílů daný zásobník obsahuje.



Obrázek 2.33 – Databázový server

2.11.1 Tabulka všech zápisů

Do této tabulky se provede zápis pokaždé, když kamera nahraje parametry do PLC. To znamená pokaždé, když se provede podprogram v robotovi *MHBCAM*. Jsou zde veškeré získané parametry z kamery. Tato tabulka byla z praktického hlediska použita pro získání parametru *LeftToRight* v celočíselné formě (integer). Avšak taková tabulka by v reálném prostředí byla užitečná z hlediska kontroly kvality. Například na parametrech *LocatorScore*, *LeftScore* a *RightScore* je možné vidět, jak kvalitně kamera snímá a zda už se neblíží hranici, kdy už to pro ni bude nerozeznatelné. To může být způsobeno docházejícím tonerem v tiskárně, nečistotě na čočce kamery, svítícím sluncem apod. Kontrolor pak díky těmto datům může včas reagovat a zabránit tak zastavení výroby či jiným nákladným problémům.

Databáze není seřazená podle stáří, protože neobsahuje *primární klíč*. Pro tento účel to ale nevádí, přímo v SQL databázi se dá seřadit pomocí příkazu a v MES modulu se dá pomocí nastavení *jobu* (úkolu) získat nejnovější záznam.

V tabulce 2.7 je zobrazeno 5 záznamů z této tabulky. Tabulka je transponovaná, v databázi jsou názvy jednotlivých položek vodorovně a jednotlivé hodnoty jsou za sebou svisle.

Tabulka 2.7 – Vzorek z tabulky všech záznamů

DateTime	2020-05-04 15:30:11.643	2020-05-04 15:31:10.647	2020-05-04 15:35:42.640	2020-05-04 15:36:48.637	2020-05-04 15:37:35.647
LocatorScore	94	83	90	93	92
LocatorX	3,48	2,99	3,81	3,85	2,73
LocatorY	14,24	14,5	14,45	13,64	14,18
LocatorAngle	0,08	0,08	0,08	0,08	0,08
LeftScore	4,31	14,87	14,23	9,26	8,31
LeftPosition	85,65	85,5	85,64	85,86	85,7
RightScore	3,92	14,62	13,94	8,86	7,95
RightPosition	3,92	14,62	13,94	8,86	7,95

2.11.2 Tabulky jednotlivých zásobníků

Tyto tabulky slouží pro správnou identifikaci, na kterou pozici má robot díl uložit. Tabulka má 3 sloupce. První dva jsou z čistě informačního hlediska, kde je možné vidět, jaký díl se do databáze zapsal v jaký čas. Třetí sloupec *Stock* je stejný pro každý díl a značí číslo zásobníků. Toto číslo je pak zapsáno do registru v PLC, ze kterého si pak robot vypočítá potřebnou pozici. Údaj o počtu dílů v zásobníku se získá pomocí počtu zápisů do dané tabulky. Získání tohoto údaje a jeho zapsání do registru v PLC se nastavuje v MES modulu.

2.11.3 Vymazání tabulek pro zásobníky

Protože pro výpočet počtu dílu v zásobníku je použit počet záznamů a v této úloze už se po uložení s díly dále nepracuje, tak je nutné při spuštění programu od začátku vymazat dané tabulky. Tohoto se docílí příkazem `DELETE FROM jmeno_tabulky`. Pro rychlejší práci byl vytvořen skript, který vymaže všechny tabulky najednou.

ZÁVĚR

Průmyslová revoluce 4.0 je záležitostí, která mění svět a nabízí mnoho možností, proto jsem rád, že jsem měl možnost pracovat na práci, která se zabývá aspekty této revoluce. Data momentálně hýbou světem a mnoho firem proto pracuje na tom, aby získávaly co největší objem informací. Další věcí je práce s těmito informacemi, kdy se toto odvětví také rozvíjí velkou rychlostí.

Výstupem této práce je třídící systém, kdy robot třídí díly podle barvy a to na základě značení na jedné straně dílu. Rozpoznávání zajišťuje kamera, která tato data posílá do robota, který je předává do PLC. Z PLC si tato data přebírá MES modul a ten už díky komunikaci s databází rozhodne, o jakou barvu dílu se jedná. Z databáze se zasílají data do PLC, které robotovi slouží pro uložení dílu na správnou pozici. Tato práce mi rozšířila znalosti v robotice, v MES systémech, v konfigurování přístrojů pro strojové vidění, osvěžila znalosti programování PLC, konkrétně programování pomocí příčkového diagramu a celkově mi poskytla nahlédnout více do záležitostí spojených s Průmyslem 4.0

Tato práce nabízí možnosti rozšíření a mohla by být použita pro další práci. Například vytvoření analýzy, která bude umět pracovat se získanými daty a tato data tak využít, dále je možné vytvoření požadavků, kdy by robot tvořil z roztříděných dílů nějaké zadané kombinace. Proto bych také chtěl pokračovat v magisterském studiu a plánuji tuto bakalářskou práci použít jako základ pro diplomovou práci.

POUŽITÁ LITERATURA

- ALIBABA. 1999-2020. Competitive price 4 axis articulated robot arm for sale.
In: *Alibaba* [online]. Hangzhou: Alibaba.com, [cit. 7.5.2020]. Dostupné z:
https://www.alibaba.com/product-detail/Competitive-price-4-axis-articulated-robot_60511641588.html
- BALDA, Pavel. Informační a řídicí systémy I. Programování PLC III. – IEC 61131-3.
In: *Vendulka* [online]. Plzeň: Oddělení Informačních a řídicích systémů [cit. 7.5.2020].
Dostupné z: https://vendulka.zcu.cz/Download/Free/IRS1/IRS1-04_IEC_3.pdf
- BRYNDA, Petr. 2019 15 let digitalizace výroby. Prezentováno na: 15 let digitalizace výroby, 18.6.2019 , Mitsubishi Electric FA Center Praha, Česká Republika
- Coordinates of a robot system. In: *Learnchannel-TV.com* [online]. Learnchannel-TV.com [cit. 7.5.2020]. Dostupné z: <http://learnchannel-tv.com/robot/robot-coordinate-systems/>
- COPE, Kevin. 2020. What is the automation pyramid? In: RealPars [online]. Rotterdam: RealPars B.V., 11.6.2018 [cit. 7.5.2020]. Dostupné z: <https://realpars.com/automation-pyramid/>
- Cylindrical robots. 2020 In: *Industrial Electricity: Information and Resources* [online]. 2020, 9.9.2007 [cit. 7.5.2020]. Dostupné z: https://www.industrial-electronics.com/industrial-electricity-com/4_Cylindrical_Robots.html
- DELOITTE. 2020. What is digital economy?: Unicorns, transformation and the internet of things. Deloitte Malta [online]. Birkirkara: Deloitte, [cit. 7.5.2020]. Dostupné z: <https://www2.deloitte.com/mt/en/pages/technology/articles/mt-what-is-digital-economy.html>
- DIXON, Mary. 2020. What are the most popular programming languages?. *RealPars* [online]. Rotterdam: RealPars B.V., c2020, 26.11.2018 [cit. 7.5.2020]. Dostupné z: <https://realpars.com/plc-programming-languages/>
- Emerging 3D Vision Technologies for Robot Vision & Machine Vision.
2009-2020 *Vision Online - Vision Systems* [online]. Ann Arbor (Michigan): AIA, [cit. 7.5.2020]. Dostupné z: <https://www.visiononline.org/blog-article.cfm/Emerging-3D-Vision-Technologies-for-Robot-Vision-Machine-Vision/194>
- ENDEAVOR BUSINESS MEDIA. 2013. The Difference between Cartesian, Six-Axis and SCARA Robots. 2013 In: *Machine Design* [online]. 2.12.2013 [cit. 7.5.2020]. Dostupné z: <https://www.machinedesign.com/mechanical-motion-systems/article/21831692/the-difference-between-cartesian-sixaxis-and-scara-robots>
- EPSON. 2020. Epson SCARA T3 series *Epson* [online]. Nagano: Seiko Epson Corporation, [cit. 7.5.2020]. Dostupné z: <https://www.epson.cz/products/robot/epson-scara-t3-series>
- Ethernet vs. sériové protokoly v průmyslu. 1997-2014. *Elektronika v automatizaci* [online]. Praha: HW server s.ro., 4.10.2018 [cit. 7.5.2020]. Dostupné z: <https://automatizace.hw.cz/ethernet-vs-seriove-protokoly.html>

- HARIYAMA, M., N. YOKOYAMA a M. KAMEYAMA. 2008. Design of a Trinocular-Stereo-Vision VLSI Processor Based on Optimal Scheduling. *IEICE Transactions on Electronics* [online]. E91-C(4), 479-486 [cit. 7.5.2020]. DOI: 10.1093/ietele/e91-c.4.479. ISSN 0916-8524. Dostupné z: <http://joi.jlc.jst.go.jp/JST.JSTAGE/transele/E91.C.479?from=CrossRef>
- IDEA. 2017. FBD Programming Manual. In: *IDEA Global* [online]. Osaka: IDEA Corporation, [cit. 7.5.2020]. Dostupné z: https://www.idec.com/language/chinese_s/AO/FT9Y-B1386.pdf
- Introduction to machine vision: A guide to automating process & quality improvements. 2020. *Assembly Magazine* [online]. Troy (Michigan): BNP Media, [cit. 7.5.2020]. Dostupné z: https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf
- MARR, Bernard. 2019 What is Machine Vision And How Is It Used In Business Today? *Forbes* [online]. Jersey City (New Jersey): Forbes Media, 11.10.2019 [cit. 7.5.2020]. Dostupné z: <https://www.forbes.com/sites/bernardmarr/2019/10/11/what-is-machine-vision-and-how-is-it-used-in-business-today/#7af97b426939>
- MES CENTRUM. MES systém (Manufacturing Execution System). *MES Center* [online]. MES centrum [cit. 7.5.2020]. Dostupné z: <http://mescenter.org/cz/clanky/5-co-je-to-mes-system>
- MES CENTRUM. MES systém - základní funkcionality. *MES Center* [online]. MES centrum [cit. 7.5.2020]. Dostupné z: <http://mescenter.org/cz/clanky/6-zakladni-funkcionality-mes-systemu>
- OWEN-HILL. 2016. Alex. Robot Vision vs Computer Vision: What's the Difference? In: *Robotiq* [online]. 7.11.2016 [cit. 7.5.2020]. Dostupné z: <https://blog.robotiq.com/robot-vision-vs-computer-vision-whats-the-difference>
- Programming robots. In: *Tony Hargreaves CV* [online]. Yorkshire [cit. 7.5.2020]. Dostupné z: <https://thnet.co.uk/thnet/robots/33.htm>
- Remote procedure call. 2001- In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, [cit. 2020-05-14]. Dostupné z: https://en.wikipedia.org/wiki/Remote_procedure_call
- Roboty a manipulátory: Učební text VOŠ a SPŠ Kutná Hora. 2008 In: *Ing. Luděk Kohout* [online]. Kutná Hora: Luděk Kohout, [cit. 7.5.2020]. Dostupné z: http://www.edumat.cz/texty/Roboty_manipulatory.pdf
- ROJKO, Andreja. 2017. Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies (iJIM)* [online], 11(5), 77-90 [cit. 7.5.2020]. ISSN 1865-7923. Dostupné z: <https://online-journals.org/index.php/ijim/article/view/7072>

ROOMS, Filip. 2019. Point Clouds – 2: Pointless without scanners. *Bricrys CAD blog* [online]. Gent: Bricrys NV, [cit. 7.5.2020]. Dostupné z: <https://blog.bricsys.com/point-clouds-scanners/>

SHAH, Bhavin C., Dr. Devendra NAGAL a Dr. Swati SHARMA. 2016. Coordinate systems for industrial robots. In: *International Conference on Emerging Technologies in Engineering, Biomedical, Medical and Scienc* [online]. International Journal For Technological Research In Engineering, November 2016, s. 191-193 [cit. 7.5.2020]. ISSN 2347 - 4718. Dostupné z: International Journal for Technological Research in Engineering

TECHNICKÁ UNIVERZITA V KOŠICIACH. 2011. Automatizační a robotická technika. *Strojnická fakulta TUKE* [online]. Košice: Strojnická fakulta TU v Košiciach, September 2011 [cit. 7.5.2020]. Dostupné z: http://www.sjf.tuke.sk/vitalab/files/vystupy/prirucka_cz_final.pdf

PŘÍLOHY

A – CD

Příloha k bakalářské práci

Využití modulu MES Mitsubishi RD81MES96 pro řízení výrobního
procesu na základě požadavků z SQL databáze

Marek Henzl

CD

OBSAH

1. Text bakalářské práce ve formátu PDF
2. Databáze s tabulkami pro prostředí MS SQL Management Studio 2019
3. Projekt s programy pro robota
4. Projekt s programem pro PLC
5. Projekt pro MES modul
6. Projekt s referenčním obrázkem pro kameru
7. Video s ukázkou výsledné činnosti