



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ABB ROBOTSTUDIO - OVLÁDÁNÍ ROBOTA Z PLC B&R AUTOMATION

ABB ROBOTSTUDIO - ROBOT CONTROL USING PLC B&R AUTOMATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Eduard Mittaš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. et Ing. Stanislav Lang, Ph.D.

BRNO 2019

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Eduard Mittaš**
Studijní program: Strojírenství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **Ing. et Ing. Stanislav Lang, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

ABB RobotStudio – ovládání robota z PLC B&R Automation

Stručná charakteristika problematiky úkolu:

Student v rámci práce vytvoří scénu s robotem v prostředí RobotStudio, samotné programování jednotlivých pohybů však bude realizovat mimo uvedené prostředí (pokud možno prostřednictvím programovatelného automatu B&R Automation, případně z prostředí Matlab & Simulink). Výsledkem bude vytvoření jednoduchých demonstračních pohybových programů.

Cíle bakalářské práce:

Proveďte stručnou rešerši v oblasti robotiky.
Nastudujte řídicí prostředky B&R Automation.
Vytvořte scénu s robotem v prostředí RobotStudio.
Vytvořte programy realizující základní pohybové příkazy pro robota.
Vybrané pohybové programy otestujte i na reálném robotu (v laboratoři).
Vytvořte stručný návod pro spojení s ABB robotem.

Seznam doporučené literatury:

ABB Group [online]. [cit. 2018-09-11]. Dostupné z: <https://new.abb.com/cz>

B&R Automation [online]. [cit. 2018-09-10]. Dostupné z: <http://www.br-automation.com/>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Cieľom tejto záverečnej práce je ukázať rôzne spôsoby komunikácie s kontrolérom robota a tiež následné ovládanie tohto robota. Simulácia úlohy bude prevedená s využitím protokolu TCP/IP, pričom komunikácia v laboratóriu s reálnymi zariadeniami bude zaistená zbernicou Profinet. V tejto práci je tiež ukázané ďalšie riešenie, ktorým je jednoduchá komunikácia a prepojenie Microsoft Visual Studia a RobotStudia prostredníctvom protokolu TCP/IP. Úvodné kapitoly popisujú spoločnosti ABB a B&R Automation, ich riadiace produkty, históriu a ďalšie informácie s nimi spojené. Nasledujúce kapitoly slúžia ako návod popisujúci vytvorenie aplikácie v každom z troch prezentovaných riešení.

ABSTRACT

The goal of this bachelor thesis is to demonstrate various ways of communication with the controller of the robot as well as consecutive controlling of this robot. The simulation will be achieved by using the TCP/IP protocol, while the communication in the laboratory with real equipment will be assured by Profinet bus. This thesis will show also another solution, which is simple communication and connection between Microsoft Visual Studio and Robotstudio by using TCP/IP protocol. Initial chapters will describe ABB and B&R Automation companies, their control systems, history and information associated with them. Following chapters will serve as a guide describing creating of application in each of three presented solutions.

KLÚČOVÉ SLOVÁ

Robot, kontrolér, programovateľný logický automat, PLC, ABB, B&R Automation, RobotStudio, Automation Studio, Microsoft Visual Studio, socket, TCP/IP, Ethernet, Profinet, POWERLINK, ANSI C, C++, RAPID, Štruktúrovaný text

KEYWORDS

Robot, controller, programmable logic controller, PLC, ABB, B&R Automation, RobotStudio, Automation Studio, Microsoft Visual Studio, socket, TCP/IP, Ethernet, Profinet, POWERLINK, ANSI C, C++, RAPID, Structured Text

BIBLIOGRAFICKÁ CITÁCIA

MITTAŠ, E. *ABB RobotStudio - ovládání robota z PLC B&R Automation*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 79 s. Vedoucí bakalářské práce Ing. et Ing. Stanislav Lang, Ph.D..

POĎAKOVANIE

Chcel by som sa poďakovať svojmu vedúcemu bakalárskej práce Ing. et Ing. Stanislavovi Langovi PhD. za odbornú a pedagogickú pomoc, za jeho priateľský prístup užitočné rady a ochotu. Ďalšie poďakovanie patrí pánovi Ing Romanovi Parákov, rovnako za jeho rady, odbornú pomoc a čas. V poslednom rade by som sa chcel poďakovať svojej rodine, za ich podporu, či už finančnú alebo psychickú.

ČESTNÉ PREHLÁSENIE

Prehlasujem, že som danú prácu vypracoval samostatne pod vedením Ing. et Ing. Stanislava Langa Ph.D a s použitím literatúry uvedenej v zozname literatúry.

V Brne dňa 24.5.2018

.....
Eduard Mittaš

OBSAH

1	ÚVOD	15
2	B&R AUTOMATION	16
	2.1 HISTÓRIA FIRMY	16
	2.2 RIADIACE PROSTRIEDKY	17
	2.2.1 Automation Studio	17
	2.2.2 X20 I/O Systém	18
	2.2.3 IPC Priemyselné počítače.....	19
	2.2.4 Zbernicové a sieťové moduly	20
3	ABB GROUP	22
	3.1 HISTÓRIA FIRMY	22
	3.2 DIVÍZIE SPOLOČNOSTI.....	22
	3.3 ROBOTIKA ABB	23
	3.3.1 Robot IRB 120.....	23
	3.3.2 Riadiaci systém IRC5	24
4	ROBOTIKA	26
	4.1 HISTÓRIA ROBOTIKY	26
	4.2 PRIEMYSELNÉ ROBOTY A MANIPULÁTORY	26
	4.3 POHONY A POHYBOVÉ JEDNOTKY	29
	4.4 PROGRAMOVANIE PREMYSELNÝCH ROBOTOV	31
5	SIMULÁCIA S VYUŽITÍM KOMUNIKÁCIE TCP/IP	33
6	APLIKÁCIA AUTOMATION STUDIO	35
	6.1 ZALOŽENIE PROJEKTU	35
	6.2 VYTVORENIE PROGRAMU	36
	6.3 IMPORTOVANIE KNIŽNÍC	38
	6.4 PRÁCA S KNIŽNICAMI.....	39
	6.5 PRINCÍP KOMUNIKÁCIE.....	40
	6.6 SPUSTENIE APLIKÁCIE	42
	6.7 POUŽÍVANIE WATCH FUNKCIE	44
7	APLIKÁCIA ROBOTSTUDIO	47
	7.1 ZALOŽENIE PROJEKTU	47
	7.2 IMPORTOVANIE ROBOTA.....	48
	7.3 PRIDANIE NÁSTROJA	49
	7.4 PRIDANIE SYSTÉMU	50
	7.5 TVORBA WORKOBJECTU.....	51
	7.6 TVORBA TARGETU	52
	7.7 RAPID PROGRAM	55
	7.8 RAPID WATCH A DEBUG MODE	55
8	MICROSOFT VISUAL STUDIO	57
	8.1 TVORBA APLIKÁCIE	57
	8.2 KONFIGURÁCIA PRIPOJENIA	58
	8.3 KOMUNIKÁCIA ROBOTSTUDIO A MICROSOFT VISUAL STUDIO ..	59
9	KOMUNIKÁCIA S VYUŽITÍM PROFINETU	61
	9.1 PRÍSLUŠENSTVO	61
	9.2 NASTAVENIE KOMUNIKÁCIE AUTOMATION STUDIO	64
	9.2.1 Nastavenie adries a premenných	64
	9.2.2 Tvorba kódu a I/O mapping	66

	9.2.3 Nastavenie komunikácie PC a PLC	66
	9.3 NASTAVENIE KOMUNIKÁCIE ROBOTSTUDIO	69
	9.3.1 Nastavenie adres a názvu stanice.....	70
	9.3.2 Vytvorenie a nastavenie premenných	71
10	ZÁVER	73
11	ZOZNAM POUŽITEJ LITERATÚRY	75
12	ZOZNAM PRÍLOH	79

1 ÚVOD

Cieľom tejto práce je ovládanie stacionárneho robota IRB 120 od firmy ABB prostredníctvom programovateľného automatu X20CP1584 od firmy B&R Automation. Bude vytvorená komunikácia medzi automatom a robotom, najskôr v simulácii medzi Robotstudiom od ABB a medzi Automation studiom od B&R pomocou TCP/IP. Neskôr bude priamo použité komunikačné rozhranie PROFINET pri práci s reálnymi zariadeniami. Tieto prostriedky budú popísané v nasledujúcich kapitolách. Ďalší spôsob komunikácie s robotom, a jeho prípadné možné ovládanie, bude realizované prostredníctvom Microsoft Visual Studia a RobotStudia na simulácii na osobnom počítači.

Táto práca má tiež ukázať integráciu B&R Automation, ktorá bola odkúpená spoločnosťou ABB, ich kooperáciu a prepojenie a tiež iný netradičnejší prístup ovládania robota, rovnako ako funkcie a vlastnosti použitých aplikácií.

Prvé kapitoly budú popisovať jednotlivé firmy, ich históriu, filozofiu. Ďalej sa zameriam na riadiace prostriedky B&R Automation, softwarové aj hardwarové. Následne popíšem spoločnosť ABB a samotného robota IRB 120, spolu s riadiacimi prostriedkami ABB. Nasleduje rešerš z oblasti robotiky, ktorý bude zameraný predovšetkým na stacionárne roboty.

Druhá časť bude pojednávať o samotnej simulácii v Automation Studiu, kde bude realizovaná simulácia kontrolného zariadenia, a RobotStudiu, kde bude realizovaná simulácia robota a jeho kontrolného systému. Ďalej o jednoduchej aplikácii v Microsoft Visual Studiu, ktorá umožní komunikovanie s RobotStudiom. Nasledovať bude praktické prevedenie v laboratóriu, ktoré bude demonštrované jednoduchými pohybmi robota.

2 B&R AUTOMATION

2.1 HISTÓRIA FIRMY

Spoločnosť B&R Automation započala svoju činnosť v roku 1979, teda pred 40 rokmi, v Rakúskom meste Eggelsberg, kde sa dodnes nachádza ich hlavná centrála. Názov spoločnosti vznikol ako iniciály priezvisk jej zakladateľov, pánov Erwina Berneckera a Josefa Reineru. Ich slogan je „Dokonalosť v automatizácii“, čo je zároveň aj ich heslom. V súčasnosti pôsobia po celom svete, celkovo vo viac ako 70 krajinách, pričom pobočky majú aj v Českej republike, napríklad v Prahe a v Brne, a aj na Slovensku, v Novom Meste nad Váhom a v Košiciach. V roku 2017 bola táto spoločnosť odkúpená firmou ABB. Ich úloha v skupine ABB, kde slúžia ako centrum pre strojnú a továrnu automatizáciu, posilnila ich pozíciu medzi konkurenciou.[1,2]



Obr. 1 Hlavné sídlo spoločnosti v Eggelsbergu[2]

V súčasnosti patria k popredným výrobcom automatizačnej techniky, ktorí prinášajú a vymýšľajú jedny z najlepších riešení v priemyselnej automatizácii a pýšia sa mnohými významnými patentami. Zároveň spolupracujú s mnohými univerzitami a poskytujú bezplatné školenia, vďaka čomu získavajú nových zamestnancov a propagujú svoje produkty a tým neustále narastajú.



Obr. 2 Pobočka B&R Automation Brno[3]

2.2 RIADIACE PROSTRIEDKY

B&R Automation ponúka široké spektrum produktov slúžiacich pre ovládanie strojov a systémov s vysokou rýchlosťou a precíznosťou riadenia. B&R zaručuje tieto podmienky vďaka veľkej rozšíriteľnosti ich riadiacich prostriedkov. Toto umožňuje, aby pokryli všetky požiadavky jedinou platformou, od najmenších kontrolérov až po CNC, robotiku a aplikácie slúžiace na ovládanie procesov. Navyše táto platforma môže byť naprogramovaná a nakonfigurovaná bez ohľadu na použitý hardware s Automation Studiom. Táto kompatibilita ušetrí výrobné náklady a súčasne ochráni investície počas celého životného cyklu stroja.[4]

2.2.1 AUTOMATION STUDIO

Automation Studio, v súčasnej najnovšej verzii 4, je vrcholný univerzálny nástroj pre strojné a systémové inžinierstvo. Už od jeho vytvorenia v roku 1997 sa neustále zdokonaľuje, dokáže pokryť každý aspekt automatizácie, od ovládania pohybu, technológií až po HMI, vizualizáciu, či integrované bezpečnostné technológie. Vďaka tomuto minimalizuje náklady na vývoj, urýchľuje uvedenie na trh a umožňuje opätovné použitie hardwaru svojou plne grafickou hardware konfiguráciou. Obrovskou prednosťou Automation Studia je podpora objektovo orientovaného programovania, takže okrem všetkých jazykov IEC 61131-3 umožňuje navyše programovanie aj v jazyku C++, ale aj v jazykoch C alebo CFC. Programátori tak majú na výber zo širšej škály jazykov a môžu si zvoliť svoj preferovaný. [5,7]



Obr. 3 Funkcie Automation Studia reprezentujúce jeho univerzálnosť[5]

Ďalšou prednosťou sú PLCopen funkčné bloky pre riadenie pohybu alebo bezpečnostnú techniku alebo konektory pre integráciu automaticky generovaného kódu zo simulačných systémov. Projektové súbory sú vo formáte XML, ktorý zaisťuje otvorenú komunikáciu so systémami tretích strán ako je riadenie materiálu alebo plánovanie

výroby. Zároveň umožňuje s použitím SQL rozhraní priamy prístup k dátam uloženým v databázach.

Automation Studio používa protokol OPC UA. Architektúra tohto protokolu umožňuje použitie užitočných funkcií a kompatibilitu so systémami tretích strán, čím uľahčuje prácu vývojárov.

Zároveň odstraňuje bariéry medzi systémami, takže dokáže spolupracovať s rôznymi ďalšími aplikáciami, napríklad E-PLAN, MATLAB alebo už spomínané SQL, tým, že využíva funkcie, ktoré vznikajú zo simulácií v týchto programoch.[5,7]



Obr. 4 Aplikácie schopné spolupráce s Automation Studiom[5]

Vďaka priamej integrácii všetkých zbernicových systémov je k dispozícii celé spektrum dostupných periférií a optimálna podpora integrácie systémov tretích strán.

Bezpečnostné riešenia B&R Automation sú založené na openSAFETY. Toto je prvý a jediný zbernicovo nezávislý bezpečnostný štandard, využívajúci princíp Black Channel, ktorý je prístupný pre všetky priemyselné Ethernetové a zbernicové riešenia. To poskytuje rýchlejšie odozvy a zároveň zvyšuje produktivitu. Celé toto riešenie je naprogramované v Automation Studiu. [6]



Obr. 5 Logo štandardu openSAFETY[6]

2.2.2 X20 I/O SYSTÉM

Vďaka výkonným procesorom jako sú napríklad Intel® Atom™ alebo ARM procesory, dokáže systém X20 zvládnuť všetky úlohy, malé aj veľké. Tento systém je extrémne kompaktný a modulárny, vďaka systému „kariet“ uložených vedľa seba, ktoré sú tvorené svorkami slúžiacimi na upevnenie, radovou svorkovnicou, modulu s elektronikou a zbernicového modulu ktorými sú spojené. Integrované prepojenia tvorené zbernicami poskytujú voľnosť pre decentralizované stroje a systémové koncepty. Celý systém sa tak skladá z viacerých modulov. [8]



Obr. 6 Zložený X20 systém[9]



Obr. 7 Modul X20 systému[8]

K systému X20 B&R ponúka mnoho vstupno/výstupných modulov, napríklad v analógovom alebo digitálnom prevedení, rôzne napájacie zdroje, čítače, moduly pre CPU, tento modul obsahuje vstupy pre RS232 alebo USB, Ethernet, ale môže obsahovať aj vstupy pre ďalšie komunikačné rozhrania. Ďalej špeciálne moduly na prácu s enkodermi, PWM signálom, či meranie teploty alebo komunikačné moduly napríklad pre Etherne, CAN, PROFIBUS alebo PROFINET a tiež pre ich vlastný komunikačný štandard Ethernet POWERLINK, čo je priemyselný typ Ethernetu. Firma ponúka aj safety moduly (na obrázku 3 žltej farby), ktoré slúžia pre bezpečnostné technológie. [9]

X20 systém je veľmi rýchly, dokáže zvládnuť dokonca aj rýchlosť cyklu 100 μ s. Jeho ďalšou prednosťou je jeho už mnohokrát spomínaná vysoká modularita, vďaka ktorej je schopný prepojiť až 250 modulov a 3000 kanálov. Zároveň je tento systém kompatibilný so systémom X67, čo je vzdialený vstupno/výstupný modul, ktorý sa môže umiestniť priamo na stroj, a jeho konštrukcia spolu s ochranou IP67 zaručuje odolnosť v najdrsnejších prostrediach. Samozrejme, že aj systémy X20 sa vyrábajú v oplášťovanom prevedení zaručujúcim ochranu pred kondenzáciou a koróziou.[9,10,11]

2.2.3 IPC – PRIEMYSELNÉ POČÍTAČE

Počítače od spoločnosti B&R Automation sú navrhnuté aby spĺňali požiadavky klientov pre maximálnu odolnosť, spoľahlivosť a čo najdlhšiu funkčnosť. Tieto počítače využívajú procesory od spoločnosti Intel, a to v širokej škále od rady Atom až po najvýkonnejšie rady „i-core“ série, čím zaručujú plynulý chod aplikácii s vysokými systémovými požiadavkami. V ponuke sú architektúry jedno, dvoj a štvorjadrové. Samozrejme, že tieto priemyselné počítače sa vyrábajú v modulárnom prevedení, čo im umožňuje používať SDL a SDL4, teda Smart Display Link, pre všetky generácie produktov. SDL prenáša všetky komunikácie, cez jediný kábel a prepája tak každé panelové zariadenie s IPC. Zároveň integrujú všetky dôležité rozhrania a to dva porty pre Gigabit Ethernet, jeden pre USB 2.0 a jeden pre USB 3.0, ďalej už spomínaný

POWERLINK a tiež CAN alebo RS232. K dispozícii sú tiež sloty pre CFast, ktorý nahrádza Compact Flash karty.

Celá konštrukcia je bez ventilátorov a využíva SSD disky, čo zabezpečuje maximálnu odolnosť v priemyselnej aplikácii odstránením rotujúcich častí. Zariadenie môže byť ovládané operačným systémom Windows od Microsoftu a to konkrétne verziou 7, 8.1 Industry Professional alebo 10 IoT Enterprise alebo systémom Linux. V neposlednej rade je tiež k dispozícii run – time operačný systém od samotnej spoločnosti B&R a to Automation Runtime. Kombinácia Automation Runtime a Windowsu sa používa pri aplikáciách vyžadujúcich hard real – time. Pokiaľ je architektúra procesoru viacjadrová, real – time OS beží na jednom jadre, kým ostatné jadrá sú rezervované pre Windows.[12,13,14,15,16]



Obr. 8 Automation PC 2100[13]

2.2.4 ZBERNICOVÉ A SIEŤOVÉ MODULY

Pri aplikáciach je veľmi dôležitá otvorenosť a práve preto B&R ponúka zbernicové pripojenia v rôznych formách pre všetky ich riadiace systémy. V Automation Studiu je kompletne integrované pripojenie zariadení tretej strany a to použitím GSD a EDS

súborov alebo FDT/DTM technológie. Aj zbernicové slave moduly umožňujú integráciu vstupno / výstupných systémov od B&R na zariadeniach tretích strán.[18]

Ethernet POWERLINK, ktorý bol už viackrát spomínaný, je priemyselný štandard Ethernetu vyvinutý spoločnosťou B&R, pričom sa jedná už o druhú generáciu priemyselných zbernic. Jeho obrovskou prednosťou je práca v reálnom čase, pričom vďaka protokolu POWERLINK dokážeme prepojiť Ethernet až na najnižšiu úroveň, teda k samotným snímačom a akčným členom, s cyklami kratšími ako 200 μ s a časovaniu presnému na mikrosekundy. Môžeme ho využiť aj na vizualizáciu alebo výmenu dát medzi PLC. Oproti svojmu predchodcovi rozširuje štandard o aplikačnú vrstvu. POWERLINK dokáže rolišovať domény reálneho času a tie, ktoré v ňom nepracujú. Dokáže pracovať v dvoch režimoch, a to v režime Ethernetu, kedy nie je potrebný prenos dát v reálnom čase, pričom tento režim je východzí a v režime POWERLINK, pre real – time aplikácie. Týmto zaisťuje bezpečnosť vo všetkých hľadiskách. Ďalšou prednosťou tejto zbernice je, že zariadenia POWERLINK sú vybavené niekoľkými Ethernetovými portmi, vďaka čomu môžeme realizovať zapojenie prakticky v akejkolvek topológii. [17,21]



Obr. 9 Logo POWERLINKu[17]



Obr.10 Logo Profinetu[19]

Profinet je ďalší veľmi dôležitý komunikačný štandard. Založený na TCP/IP a IT štandardoch, toto priemyselné prevedenie Ethernetu je vyrobené firmou PROFIBUS & PROFINET International. B&R využíva Profinet vo variantách NRT, čo je non-real-time, ďalej RT, čo je real-time prevedenie a nakoniec IRT, čo je izochrónne prevedenie. NRT je používané pri aplikáciách, ktoré nie sú časovo kritické, takže je pomalší a to pri rýchlostiach cyklu menej ako 100 ms. Profinet RT vstupno/výstupné dáta vymieňa prostredníctvom Ethernetového protokolu, zatiaľ čo diagnostika a komunikácie sú sprostredkované prostredníctvom UDP/IP. Vďaka tomuto rozdeleniu dosahuje rýchlosti cyklu menej ako 10 milisekúnd. Profinet IRT je izochrónny real-time Profinet, ktorý sa používa špeciálne pre aplikácie riadiace pohyb. Je veľmi rýchly s rýchlosťou cyklu pohybujúcej sa nad 1 milisekundou. B&R Automation poskytuje v svojom sortimente dve zbernicové karty pre Profinet, pričom každá poskytuje dva konektory RJ45.[19]

CAN zbernica používaná predovšetkým pri riadení strojnej výroby má vysokú odolnosť voči rušeniu, rýchly prenos dát a tiež real-time aplikovateľnosť. Je vhodná na použitie pri vzdialených vstupno/výstupných uzloch. Pri komplexnejších strojoch je však vhodnejšie využiť POWERLINK. B&R vo svojom portfóliu ponúka tri komunikačné moduly CAN, pričom sa vyskytujú aj v prevedení s POWERLINKOM alebo X2X Link master rozhraním.[20]

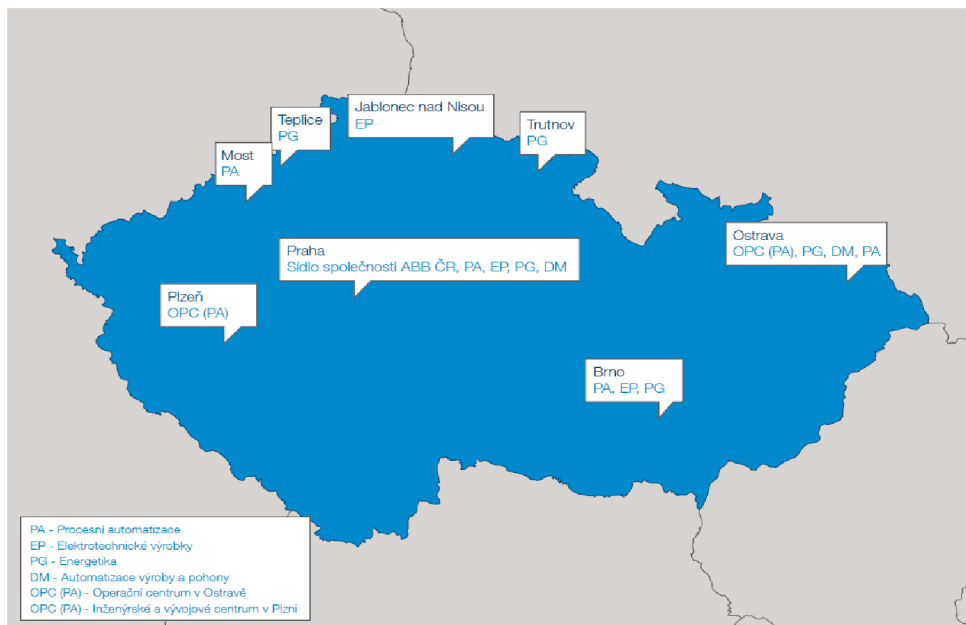
3 ABB

3.1 HISTÓRIA FIRMY

Počiatky spoločnosti ABB siahajú až do roku 1883, kedy vznikla prvá z divízií a to spoločnosť ASEA, čo je skratka pre Allmänna Svenska Elektriska Aktiebolaget. Táto firma sa spočiatku zaoberala elektrotechnickými a elektroenergetickými zariadeniami a to generátormi a svetlami. Neskôr, v roku 1974 dokonca predstavila aj priemyselné roboty. V roku 1988 sa spojila so švajčiarskou spoločnosťou BBC, teda Brown, Boveri & Cie, ktorá bola založená v roku 1891 a rovnako pracovala v odvetví elektrotechniky, a to najmä v produkcii elektromotorov, generátorov a transformátorov.

V Českej republike začali prostredníctvom poskytovania svojich služieb a výrobkov pôsobiť už v roku 1970. V súčasnosti sa najvýznamnejšie výzkumné a inžinierske centrá, spolu s výrobnými závodmi nachádzajú v Prahe, Brne, Plzni či v Ostrave.

S hlavným sídlom v Bratislave a ďalšími centrami v Žiline, Košiciach, či Banskej Bystrici pôsobí ABB aj na Slovensku. [22]



Obr. 11 Centrá spoločnosti ABB v Českej republike[22]

3.2 DIVÍZIE SPOLOČNOSTI

Prevádzková činnosť ABB sa delí celosvetovo do štyroch divízií. Každá z divízií sa skladá z jednotlivých podnikových jednotiek, pričom každá sa zameriava na určité odvetvia a výrobkové kategórie. Jednotlivé divízie sú:

- Energetika
- Elektrotechnické výrobky
- Priemyselná automatizácia
- Robotika a pohony

Vďaka svojej viac než 120 ročnej tradícii a silnému zameraniu na výzkum a vývoj, spolu s veľkým množstvom patentov, inovácií a širokému pôsobeniu sa ABB radí medzi najvýznamnejšie spoločnosti na svete.[23]



Obr. 12 Hlavná centrála ABB v Zürichu[24]

3.3 ROBOTIKA ABB

Spoločnosť ABB je vedúci dodávateľ priemyselných robotov, automatizovaných výrobných systémov a služieb s nimi spojených. Celkovo ABB už nainštalovalo viac ako 300 000 robotov po celom svete. Do tejto divízie spadajú motory, generátory, frekvenčné meniče, mechanický prenos energie a robotika.

3.3.1 ROBOT IRB 120

Tento robot sa radí do portfólia malých robotov. Napriek jeho nízkej hmotnosti a malým rozmerom disponuje kompletnou funkcionalitou štandardných priemyselných ABB robotov. Jeho rozmery a hmotnosť nie sú nevýhodou, naopak umožňujú jeho umiestnenie do stiesnených priestorov výrobných buniek, na stroje, či ako súčasť výrobných liniek. [25]

Využíva sa najmä v elektrotechnickom, strojnóm priemysle a potravinárstve, dá sa však použiť napríklad aj vo farmácii. V potravinárskom priemysle sa vyskytuje ako varianta Clean Room, pričom spĺňa ISO triedu 5, aby bol vhodný na prevádzku v prostredí vyžadujúcom čistotu. [25]

Koniec efektoru môže vážiť maximálne 3 kilogramy aj so záťažou. Iné zariadenia s maximálnou hmotnosťou 0.3 kilogramu môžu byť umiestnené na horné rameno.

Horizontálny dosah ramena je 580 milimetrov, pričom robot dokáže dosiahnuť aj pod svoju základňu a to až 112 milimetrov. [25]



Obr.13 Robot IRB 120[25]

Konštrukcia je hliníková, čo v kombinácii s riadiacim systémom IRC5 zaručuje rýchlu akceleráciu a presnosť. Pre aplikácie „pick and place“ s vysokým nárokom na krátky cyklový čas sa používa rýchlejšia varianta IRB 120T. [25]

Robot obsahuje bezpečnostné a núdzové zastavenia a zároveň dohľad aby zmenšil riziko ohrozenia. Samozrejme, že robot môže byť vybavený ďalšími podpornými software pre podporu aplikácii ako je zváranie alebo lepenie, ďalej komunikačnými rozhraniami, kontrolu senzorov alebo multitaskingom. [25]

3.3.2 RIADIACI SYSTÉM IRC5

Celkovo už piata generácia riadiaceho systému robotov ABB nazývaná IRC5 využíva technológie pre riadenie pohybu a to TrueMove, ktorý zaručuje presnosť a QuickMove, ktorý znižuje dĺžku cyklov a tým zvyšuje rýchlosť. Vďaka týmto technológiám IRC5 optimalizuje programovateľnosť, výkonnosť a synchronizáciu, či už s ľuďmi alebo externými zariadeniami. Systém SafeMove2 umožňuje flexibilnú prácu robotov, pričom zaručuje bezpečnosť aj pri kooperácii s ľuďmi.[27,28]

Všetky systémy ABB sú programované v RAPIDE. Tento jazyk, ovplyvnený jazykom C, bol vyvinutý spoločnosťou ABB v roku 1994 zaručuje flexibilitu a vysoko-úrovňové programovanie.

Systém IRC5 nepotrebuje prakticky žiadnu údržbu, vďaka vstavaným diagnostickým funkciám zaručuje rýchle obnovenie produkcie. Obsahuje tiež vzdialenú monitorovaciu technológiu, nazývanú ABB Ability Connected Services, ktorá umožňuje rýchlu analýzu chýb a real-time monitorovanie stavu robota počas celej jeho životnosti.[27,28]

Jedná sa o multiprocessorový systém vybavený procesormi Intel Pentium®. IRC5 má v sebe tiež integrované PLC, AC500, od spoločnosti ABB. Ako pamäťové jednotky slúžia Flash disky alebo hard disky, oboje schopné uchovávať veľké objemy dát, pričom zálohovanie pamäte je energeticky zabezpečené. Tento systém dokáže spracovať až 8192 vstupných či výstupných signálov v analógovom alebo digitálnom prevedení. Je vybavený mnohými komunikačnými rozhraniami ako je sériová linka RS232 alebo RS422, PROFINET, PROFIBUS DP, Ethernet alebo DeviceNet.[26,27,28]

Vyrába sa v kompaktnom alebo panelovom prevedení, pričom ovládací panel, nazývaný FlexPendant, môže byť vbudovaný alebo voľne uložený. Tieto ovládacie panely, slúžiace ako užívateľské rozhranie, obsahujú grafický farebný dotykový displej, joystick, tlačítko núdzového zastavenia a tiež podporujú USB pamäť. FlexPendants umožňujú online programovanie robotov. [26,27,28]



Obr. 14 Rôzne typy prevedení IRC5 prvý z prava kompaktný[28]

Jedným kontrolérom dokážete ovládať až štyroch robotov, pričom pre každého ďalšieho ovládaného robota je nutné pridať kompaktný pohonný modul. Systém IRC5 je modulárny. Prichádza v rôznych variantách, moduly sa môžu ukladať na seba, vedľa seba alebo môžu byť uložené v bunke, čo zabezpečuje ich optimálne rozmiestnenie. [26]



Obr. 15 Modulárne prevedenie systému IRC5 vrchný panel môže predstavovať ovládací modul a spodný pohonný modul[31]

Tieto kontroléry majú často väčšie rozmery, načo prišlo ABB s odpoveďou vo forme OmniCore. Ide o novú rodinu kontrolérov, ktoré sú založené na IRC5, pričom sú o 50% menšie. Zároveň prinášajú flexibilné možnosti upevnenia. Využívajú RobotWare7, operačný systém od ABB a už vyššie spomínané FlexPendantsy, ktoré však majú nový dizajn, aby uľahčili prácu používateľovi [29,30]



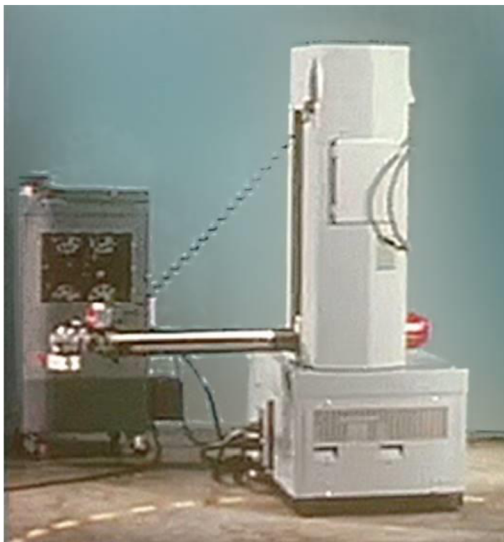
Obr. 16 OmniCore kontrolér[29]

4 ROBOTIKA

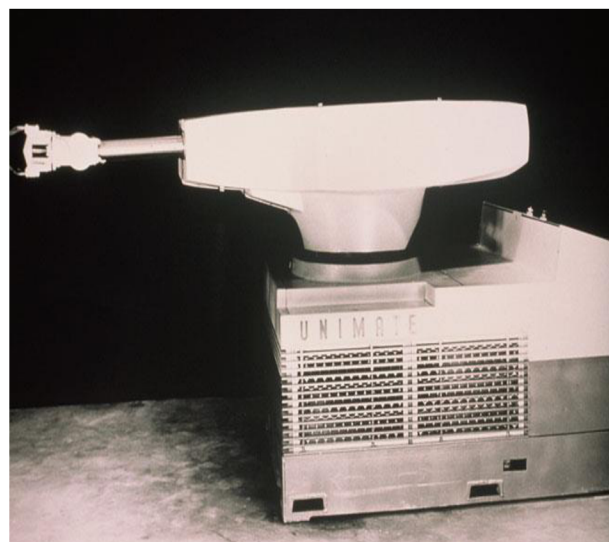
4.1 HISTÓRIA ROBOTIKY

Počiatky robotiky siahajú ďaleko do minulosti. Ako jedny z prvých pokusov by sa dali označiť napríklad figuríny poháňané parou od mechanika a konštruktéra Heróna Alexandrijského. Taktiež je veľmi dôležité spomenúť technika, umelca a vynálezcu Leonarda da Vinciho, ktorý zostrojil napríklad mechanického leva schopného chôdze a zdvihnutia prednej laby. V 18. storočí sa vďaka rozvoju prírodných vied vynašli mnoho dokonalejšie prostriedky v stavbe automatov. Mnohé z týchto mechanických konštrukcií sa využívali pri mechanických konštrukciách umelých ľudí. Išlo o automaty, schopné samostatnej činnosti. Mnohé z týchto automatov však boli podvody, nakoľko často vo svojom vnútri ukrývali človeka. Slovo robot prvýkrát použil český spisovateľ Karel Čapek. Toto slovo má základ zo staroslovanského slova „-rob-“ od ktorého je tiež odvodený výraz „robota“.[32,34]

Priemyselná výroba využíva stroje plniace určité funkcie za človeka. Tieto stroje sa však často označujú skôr názvom automat ako robot. V roku 1958 prišla americká firma Unimation so zariadením pod názvom „priemyselný robot UNIMATE“, ktorý ale našiel svoje uplatnenie neskôr. Ďalším mnohoúčelovým automatom bol „priemyselný robot VERSATRAN“ od americkej firmy AMF v roku 1961. Zastával funkciu človeka u výrobného stroja. Na prelome 20. a 21. storočia došlo k symbióze priemyselných robotov a NC výrobných systémov, čo umožnilo vznik plne automatizovaných tovární.[32,34]



Obr. 17 robot VERSATRAN 500[33]



Obr. 18 robot UNIMATE[32]

4.2 PRIEMYSELNÉ ROBOTY A MANIPULÁTORY

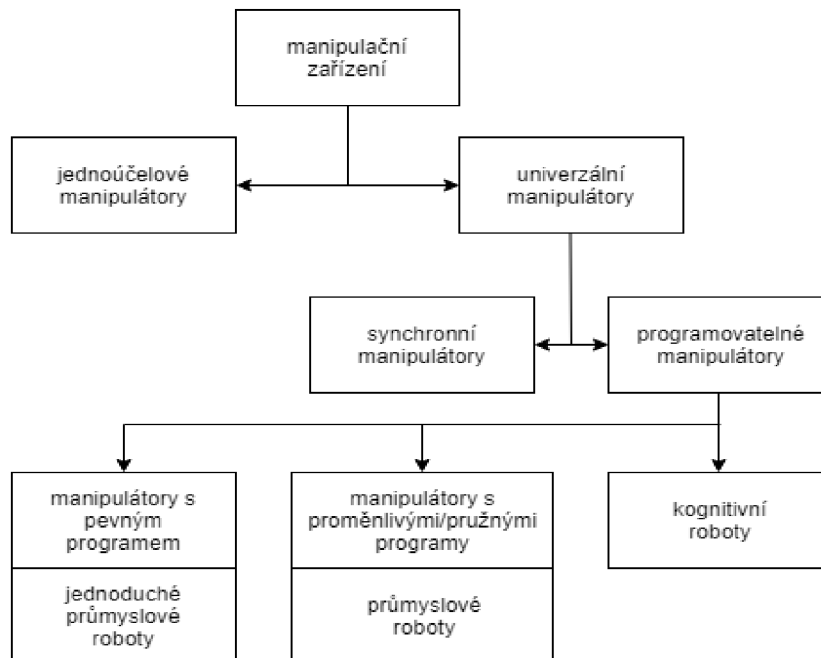
Aby sme mohli používať pojem „robot“, musí mať toto zariadenie väčšinu z nasledujúcich vlastností: [34,35,36]

- Manipulačná schopnosť, to znamená schopnosť uchopovať objekty, prenášať ich a prevádzať na nich úpravy, poprípade vykonávať manipuláciu s nástrojmi, v zmysle práce s nimi

- Univerzálnosť, čím je myslené, že zariadenie môže slúžiť k viacerým účelom a teda po zmene programu, chápadiel, či nástrojov je možné použiť ho na iný účel, alebo na inom pracovisku
- Väzba s prostredím, čo znamená vnímanie prostredníctvom senzorov, ktoré u robota predstavujú zmysly človeka. Ide napríklad o vizuálnu väzbu, akustickú, dotykovú alebo rôzne iné druhy väzieb, ktoré prenášajú informácie pre človeka nezaznamenateľné.
- Autonómna činnosť chovania, to je schopnosť vykonávať automaticky zložitú postupnosť úloh podľa určitého programu. Tento program by mal byť voliteľný človekom alebo vlastným zariadením.
- Priestorová sústredenosť jednotlivých zložiek do jedného celku, pokiaľ je to možné. Nejedná sa priamo o riadiaci systém robota. Pokiaľ je robot integrovaný do jedného celku, jeho premiestnenie je jednoduchšie. Samozrejme, že v niektorých prípadoch je potrebné, aby bol robot mobilný.
- Označenie „robot“, sa používa pre manipulačné mechanizmy vykonávajúce úkony blížiacie sa typickým úkonom človeka, pričom sú riadené počítačom. Robot sa svojimi manipulačnými schopnosťami nemusí úzko blížiť manipulačným schopnostiam ľudskej ruky.

„Priemyselný robot je autonómne fungujúci stroj–automat, ktorý je určený k reprodukcii niektorých pohybových a duševných funkcií človeka pri predvádzaní pomocných a základných výrobných operácii bez bezprostrednej účasti človeka, a ktorý je k tomuto účelu vybavený niektorými jeho schopnosťami (sluchom, zrakom, hmatom, pamäťou a podobne), ako sú schopnosť samovýuky, samoorganizácie a adaptácie, t.j. prispôsobivosť k danému prostrediu.“[50]

Manipulačné zariadenia robotov je možné kategorizovať podľa rôznych kritérií. Napríklad podľa počtu stupňov voľnosti, kinematickej štruktúry, druhu použitých pohonov, geometrie pracovného priestoru, spôsobu riadenia a programovania alebo pohybových charakteristík.[35]

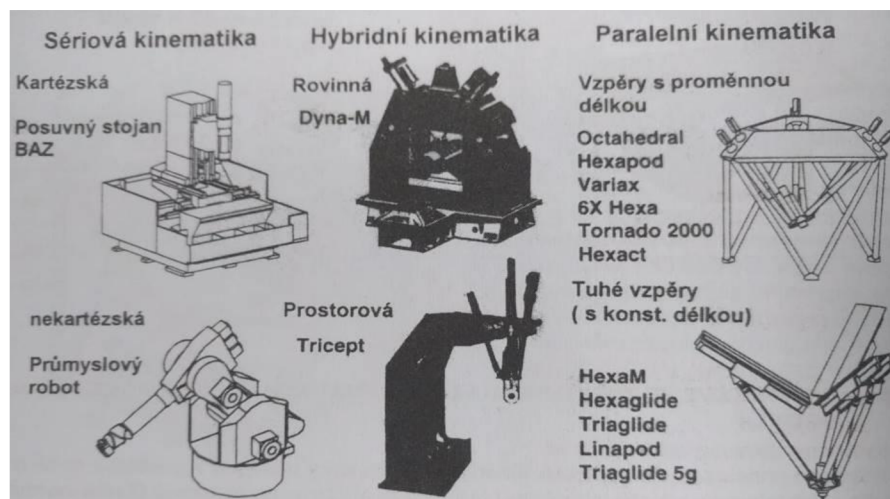


Obr. 19 Rozdelenie manipulačných zariadení [34]

Priemyselné roboty využívajú iba dva druhy kinematických dvojíc, teda väzieb, vo svojich kinematických reťazcoch. Ide o rotačnú a translačnú väzbu. [34]

Podľa kinematickej štruktúry delíme roboty na :[34,35]

- Sériové roboty, teda s otvoreným kinematickým reťazcom manipulátoru, to znamená, že využívajú sériovú kinematiku, čiže kombináciu rotačných a translačných pohybov. Každý pohon si ovláda svoje otáčanie, čím sa odstraňuje problém s pripojením pohonu, ale narastá priestorová zástavba a hmotnosť celku. Ich výhodou je jednoduchšie riadenie, jednoduchá kalibrácia a malé väzby. Na základe svojich rozmerov dokážu dosiahnuť aj veľký pracovný priestor. Naopak sú pomalšie, majú horšiu flexibilitu a ako už bolo spomínané, sú ťažšie.
- Paralelné roboty, teda s uzavretým kinematickým reťazcom manipulátoru. Pohyb je daný simultánnym riadením a kontrolou pohybu rôzneho počtu ramien, od dvoch po šesť, s premennou, či nepremennou dĺžkou. Využívajú kĺby s dvomi alebo niekedy aj tromi stupňami voľnosti. Ich výhodou je veľká rýchlosť a nízka hmotnosť, dobrá flexibilita a dynamika. Medzi nevýhody patrí nelinearita väzieb, komplikované riadenie a menší pracovný priestor a nutnosť zložitých osových transformácií.
- Hybridné roboty, ktoré kombinujú oba typy reťazcov



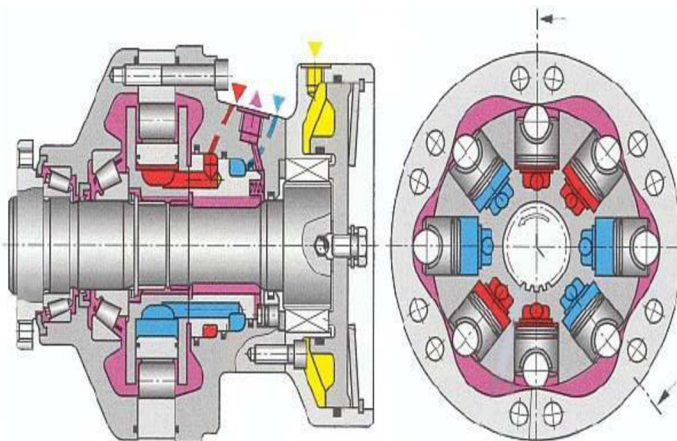
Obr.20 Morfologické triedenie kinematických štruktúr stavby strojov a robotov[34]

4.3 POHONY A POHYBOVÉ JEDNOTKY

Pojem pohon predstavuje zariadenie pre premenu energie, pričom riadi túto premenu a uvádza poháňaný stroj do požadovaného stavu. Podľa druhu energie, ktorá je využitá k premene na mechanickú energiu, rozlišujeme hydraulické, pneumatické, elektrické, mechanické a kombinované pohony. [34,35,36]

Mechanické pohony sú pohony, pri ktorých dochádza ku transformácii mechanickej práce na pohyb zariadenia alebo jeho častí. Aby k transformácii došlo, sú použité rôzne technické prostriedky, ako sú vačky, pákové mechanizmy, krokovacie mechanizmy, nárazkové mechanizmy a rôzne iné. Práve kvôli realizácii pohonu prostredníctvom týchto prostriedkov nie je trajektória pohybu programovateľná a univerzálna. Z tohoto dôvodu sa využívajú u jednocelových mechanizmov nevyžadujúcich flexibilitu a univerzálnosť. Ich prednosťou sú presnosť, rýchlosť, životnosť a spoľahlivosť. Dnes sa mimo hodinárstva tento typ pohonov nepoužíva, výnimku tvoria oblasti, kde je nevhodné používať elektronické zariadenia, napríklad v dôsledku elektromagnetického rušenia. [34,35,36]

Hydraulické pohony a servopohony sú jedny z najdôležitejších pohonov priemyselných robotov a manipulátorov. Ich vysoká účinnosť a spoľahlivosť, vysoká tuhosť a možnosť plynulého riadenia rýchlosti a dosiahnutia žiadanej polohy sú príčinami ich rozsiahleho použitia, napriek tomu, že v poslednej dobe bývajú nahrádzané elektrickými pohonmi. Výhodné dynamické vlastnosti sú zabezpečené vďaka malej hmotnosti pohyblivých častí, pričom tieto pohony dokážu dosiahnuť malých rýchlostí pohybu dokonca bez prevodov. Naopak vyššie rýchlosti sa dosahujú pomerne obtiažne. Zároveň tento pohon vyžaduje samostatný oddelený energetický blok. Ďalšou nevýhodou je, že pracovné kvapaliny, používané na realizáciu pohybu sú horľavé a menia viskozitu v závislosti od teploty. Hydraulické pohony používajú rôzne typy hydromotorov. Najrozšírenejšie sú piestové axiálne rotačné hydromotory a priamočiare hydromotory. Medzi menej používané sa radia zubové, lamelové a skrutkové alebo kyvné. [34,35,36]



Obr.21 Radiálny piestový hydromotor[38]

Pneumatické pohony sú vyhovujúcim riešením pre menšie priemyselné roboty alebo manipulátory a s jednoduchšími pracovnými cyklami. Vysokotlakové mechanizmy sa u manipulátorov nepoužívajú. Najčastejšie sa využívajú motory s priamočiarym pohybom, teda pneumatické valce. Umožňujú dosiahnutie vysokých rýchlostí a to 2 až 3 metre za sekundu, čo je ich obrovskou výhodou voči hydraulike, avšak kvôli komplikáciám s brzdením a veľkými zotrvačnými silami sa tak vysoké rýchlosti ani nedajú využiť. Aplikujú rovnaké typy priamočiarych motorov ako hydraulické pohony, okrem plunžrového prevedenia ale na rozdiel od hydrauliky používajú omnoho častejšie dvojčinné aj jednočinné membránové valce. Vďaka rôznym materiálom použitým na konštrukciu vlnovcov je možné dosiahnuť väčších zdvihov. Rotačné pneumatické motory sa používajú zriedkavo u robotov a manipulátorov, kvôli vysokým otáčkam, huku a obtiažnemu ovládaniu. [34,35,36]



Obr.22 Pneumatický priamočiary valec [37]

Elektrický pohon robotov alebo manipulátorov je subsystém tohto zariadenia, ktorý realizuje a riadi pohyb prostredníctvom premeny elektrickej energie na mechanickú prácu. Tento subsystém je najčastejšie tvorený niekoľkými elektromotormi, ktoré sú spojené s napájacími meničmi, ďalej snímačmi stavových veličín a príslušnými riadiacimi obvodmi. Pri jednoduchých manipulátoroch sa elektrický pohon realizuje prostredníctvom narážkového riadenia, takže pracuje v otvorenej regulačnej slučke, avšak najrozšírenejším riešením súčasnosti sú pohony s uzavretou regulačnou slučkou. Prioritné postavenie zaujímajú rotačné elektrické motory a servomotory. Podľa spôsobu vytvárania momentu rozlišujeme tie servomotory, ktoré vytvárajú točivý moment vzájomným pôsobením magnetického poľa statoru a rotoru, a reluktančné seromotory, ktoré točivý moment vytvárajú zmenou magnetického odporu vzduchovej medzery medzi statorom a rotorom. Samozrejme motory delíme aj podľa spôsobu napájania a to na striedavé a jednosmerné. [34,35,36]

Elektrické pohony sú najpoužívanejšie druhy pohonov v oblasti robotiky. Ostatné typy pohonov sa používajú k dosiahnutiu pohybov jednotlivých osí manipulátorov, čím vznikajú kombinované pohony, ako sú napríklad elektro–pneumatické alebo elektro–hydraulické. [34]

4.4 PROGRAMOVANIE PRIEMYSELNÝCH ROBOTOV

V dnešnej dobe existuje mnoho spôsobov programovania priemyselných robotov. Každý výrobca používa pre svojich robotov špecifický jazyk.

Dnes najpoužívanejšia metóda programovania robotov je metóda online. Princíp tejto metódy spočíva v tom, že obsluha programuje robota priamo na pracovisku prostredníctvom ručného ovládacieho panelu, ktorý bol napríklad spomenutý v kapitole 3 pri pojednávaní o riadiacich prostriedkoch ABB, tzv FlexPendant, všeobecne nazývaný teach–pendant alebo len pendant. Tento panel slúži ako užívateľské rozhranie, pričom robot môže byť navádzaný alebo sa môže programovať riadiaca aplikácia priamo na pracovisku s jeho prítomnosťou. Pendantsy sú s riadiacim systémom prepojené pomocou káblu a rozdeľujú sa podľa dizajnu na dve skupiny. Prvú tvoria ovládacie panely s dizajnom na výšku. Druhú, dizajn na šírku, obsahuje väčší displej a umožňujú ovládanie robota v ručnom režime vďaka prítomnosti joysticku alebo priestorovej myši.[34]



Obr.23 Teach pendant od spoločnosti KUKA[39]

Princíp offline programovania, druhej veľmi rozšírenej metódy, využíva softwarové systémy, prostredníctvom ktorých sa vytvorí 3D návrh robotizovaného pracoviska vo virtuálnom prostredí, založenom na kinematickom alebo dynamickom simulačnom modeli vybraného robota. Umožňuje napríklad definovať dráhy robota zapisovať body

alebo rôzne iné činnosti. Vďaka tejto simulácii sa dá chod robota optimalizovať. Napriek mnohým výhodám, ktoré offline programovanie ponúka sa často musí výsledný program, aj keď len čiastočne, upravovať na pracovisku. Príkladom jazykov používaných pri offline programovaní sú napríklad jazyk RAPID od firmy ABB alebo KRL od firmy KUKA. Program sa do robota nahrá prostredníctvom USB alebo CD či DVD mechaniky. Táto metóda je vhodná pri frézovaní, brúsení alebo lakovaní a podobných technológiách, pričom nie je účelné ju používať napríklad pri zvaraní alebo manipulácii, teda pri nesúvislých pohyboch. [34]

```

PROC DelBackups()
  VAR dir directory;
  VAR string filename;
  VAR string path:="hd0a/HOME";
  OpenDir directory,path;
  WHILE ReadDir(directory,filename) DO
    !If the next entry is a file, remove it.
    IF IsFile(filename) THEN
      RemoveFile(filename);
    ELSE
      !If not a file, logic needs to Open Directory and delete all files in it.
    ENDIF
  ENDWHILE
  CloseDir directory;
ENDPROC

```

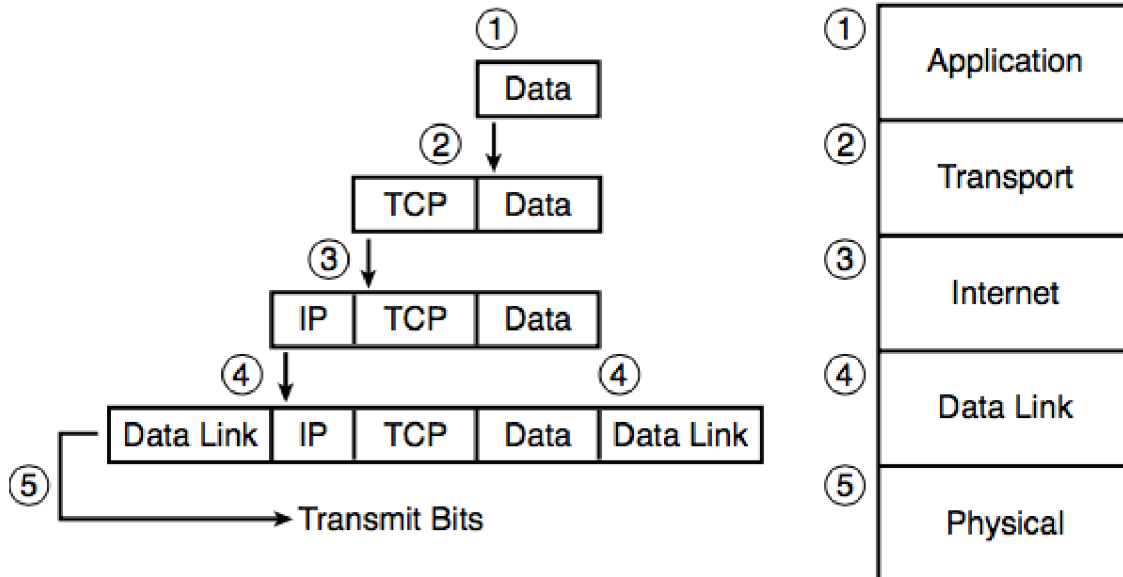
Obr.24 Príklad kódu v jazyku RAPID[40]

5 SIMULÁCIA S VYUŽITÍM KOMUNIKÁCIE TCP/IP

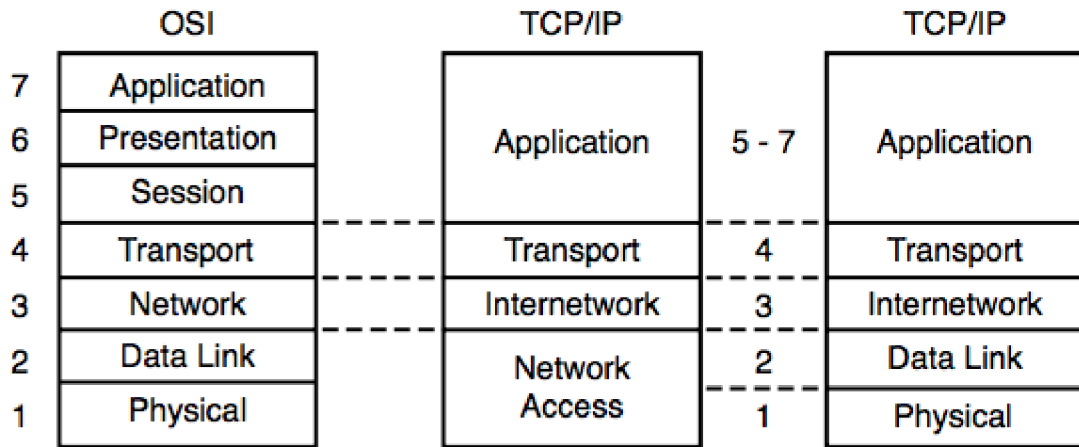
Komunikácia medzi aplikáciami bude prebiehať prostredníctvom rodiny protokolov TCP/IP. Tento prístup bol zvolený, pretože Ethernet, z ktorého PROFINET vychádza, využíva práve tento štandard, a pretože komunikáciu prostredníctvom PROFINETu nedokážeme simulovať. PLC bude v reálnej aplikácii využívať moduly, ktoré mu umožnia komunikovať s robotom. Ide o modul X20BC1083, zabezpečujúci rozhranie POWERLINK a napojený priamo na PLC, a modul X20IF10E1_1, ktorý bude tvoriť rozhranie pre PROFINET. Integráciou týchto modulov vytvoríme komunikačný kanál pre prácu s robotom.

Ethernet však neumožňuje real-time komunikáciu, preto toto riešenie nie je vhodné na aplikáciu do priemyslu.

Samotná aplikácia bude teda, podľa vzoru TCP/IP rozdelená na dve časti. Stranu serveru bude predstavovať RobotStudio od ABB a stranu klienta bude zastávať Automation studio od B&R. Prostredníctvom Automation studia budeme generovať reťazce informácií, ktoré budeme posielat' RobotStudiosu na základe príslušnej logiky. RobotStudio tieto informácie prijme, spracuje a upraví, aby s nimi dokázal robot pracovať. Komunikácia sa ukončí po odoslaní všetkých údajov a robot sa vráti do východiskovej polohy.



Obr.25 TCP/IP zapúzdrenie dát a architektúra[41]



Obr.26 TCP/IP Porovnanie architektúry s OSI modelom[41]

6 APLIKÁCIA AUTOMATION STUDIO

Stranu klienta vytvoríme v Automation studiu. Využil som Educational License a verziu 4.4.4.108.

6.1 ZALOŽENIE PROJEKTU

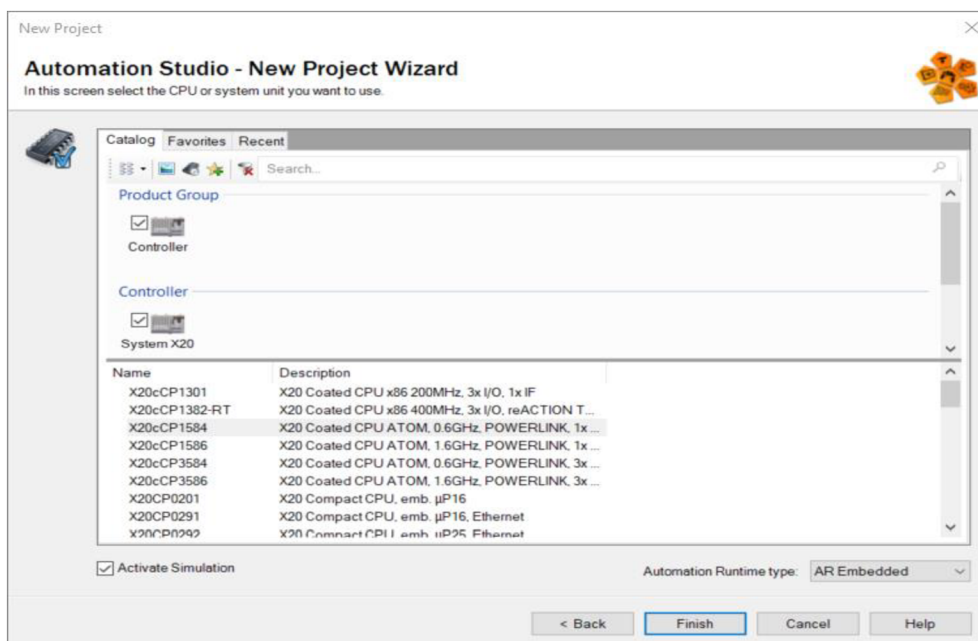
Po spustení Automation studia sa nám zobrazí *Start Page*. Tu vyberieme možnosť *New Project*, z panelu na ľavo a zvolíme *An empty project*.

Pokiaľ sa nám toto okno neotvorí, v ľavom hornom rohu zvolíme *File* a možnosť *New Project*.

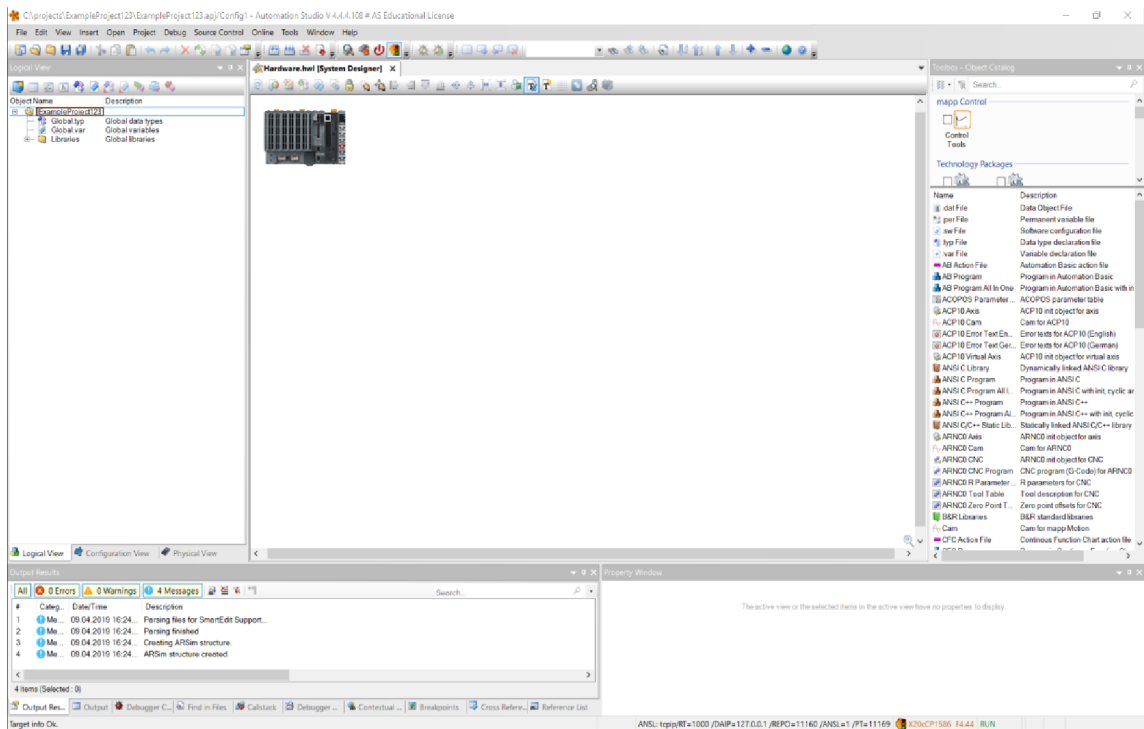
Po zvolení tejto možnosti sa nám otvorí *New Project Wizard*, v ktorom pomenujeme aplikáciu a nastavíme mu príslušný priečinok, v ktorom sa vytvorí. Môžeme pridať tiež popis projektu. Pokiaľ zvolíme možnosť *Copy Automation Runtime files into project*, tak verzia AR, ktorú používame sa nám skopíruje do projektu. Výhodou je, že program bude lepšie prenositeľný a teda pokiaľ by sa púšťal na počítači, ktorý potrebnú verziu neobsahuje, dovoľí zkompilovanie súboru aj napriek tomu. Avšak táto možnosť zväčší veľkosť projektu. V našom prípade sme túto možnosť zvolili.

V ďalšej časti nastavujeme parametre konfigurácie. Pomenujeme ju a zvolíme, ako konfiguráciu budeme definovať. Pokiaľ zvolíme možnosť *Identify hardware configuration online*, hardware konfigurácia bude nahraná zo systému prostredníctvom online pripojenia. Môžeme tiež zvoliť možnosť *Reference an existing hardware configuration*, čo nám umožní použiť konfigurácie, ktoré sme už vytvorili alebo použiť konfiguráciu z projektu niekoho iného. My zvolíme možnosť *Define a new hardware configuration manually*, čo nám umožní vybrať komponenty v ďalších krokoch.

Nakoniec sa ocitneme v poslednej časti, kde volíme CPU alebo systém. Zvolíme možnosť *Controller* z *Product Group* a následne z rady *Controller System X20*, aby sme vyfiltrovali naše možnosti. V okne pod filtrovaním nájdeme CPU X20CP1584. Pokiaľ chceme spustiť simuláciu ihneď po otvorení projektu, zvolíme možnosť *Activate Simulation*. Nakoniec potvrdíme *Finish*.



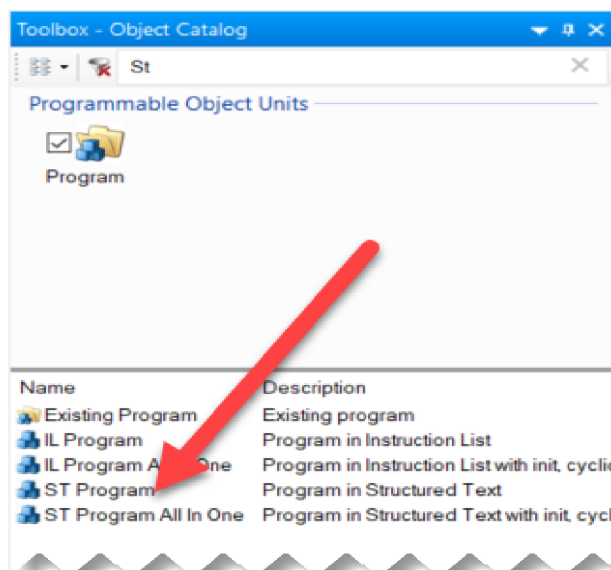
Obr.27 Manuálna konfigurácia hardware



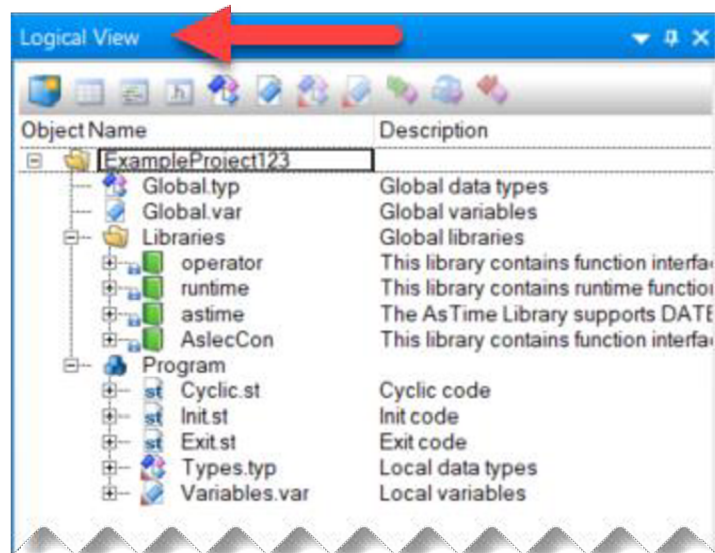
Obr.28 Založený projekt

6.2 VYTVORENIE PROGRAMU

Po vytvorení projektu nám vznikne prázdna aplikácia. Pokiaľ chceme pridať program, najskôr klikneme na náš projekt (teda akokoľvek sme si ho nazvali), v *Logical View* a prejdeme do *Toolbox-u*, ktorý sa nachádza na pravo v užívateľskom rozhraní. Náš program je písaný v jazyku štruktúrovaný text. Vo filtroch nastavíme *Program* a vyberieme možnosť *ST Program*. Ďalšia možnosť, ako pridať program je napísať do kolónky *Search* *ST Program*. Dvakrát klikneme na možnosť *ST Program*. Po rozkliknutí môžeme vidieť jednotlivé časti programu. Tie tvoria časti pre samotný kód a pre deklaráciu a definíciu lokálnych premenných

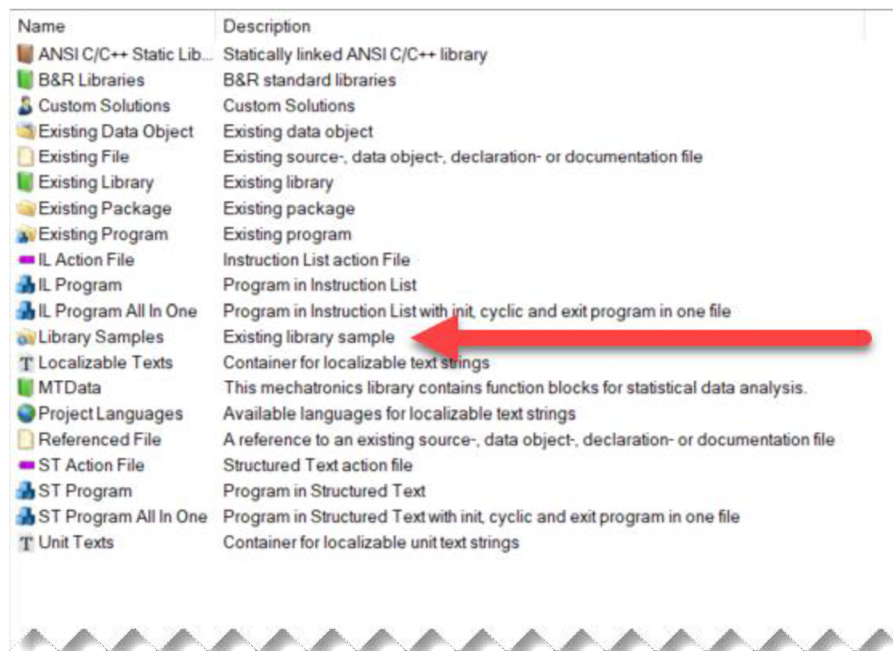


Obr.29 Toolbox – výber programu



Obr.30 Logical View – časti programu

Automation studio tiež ponúka mnoho vzorových programov v *Library samples*, ktoré demonštrujú prácu s týmito knižnicami. Niektoré sú tvorené iba ako demo programy, iné riešia rôzne konkrétne úlohy.



Obr.31 Toolbox – Library Samples

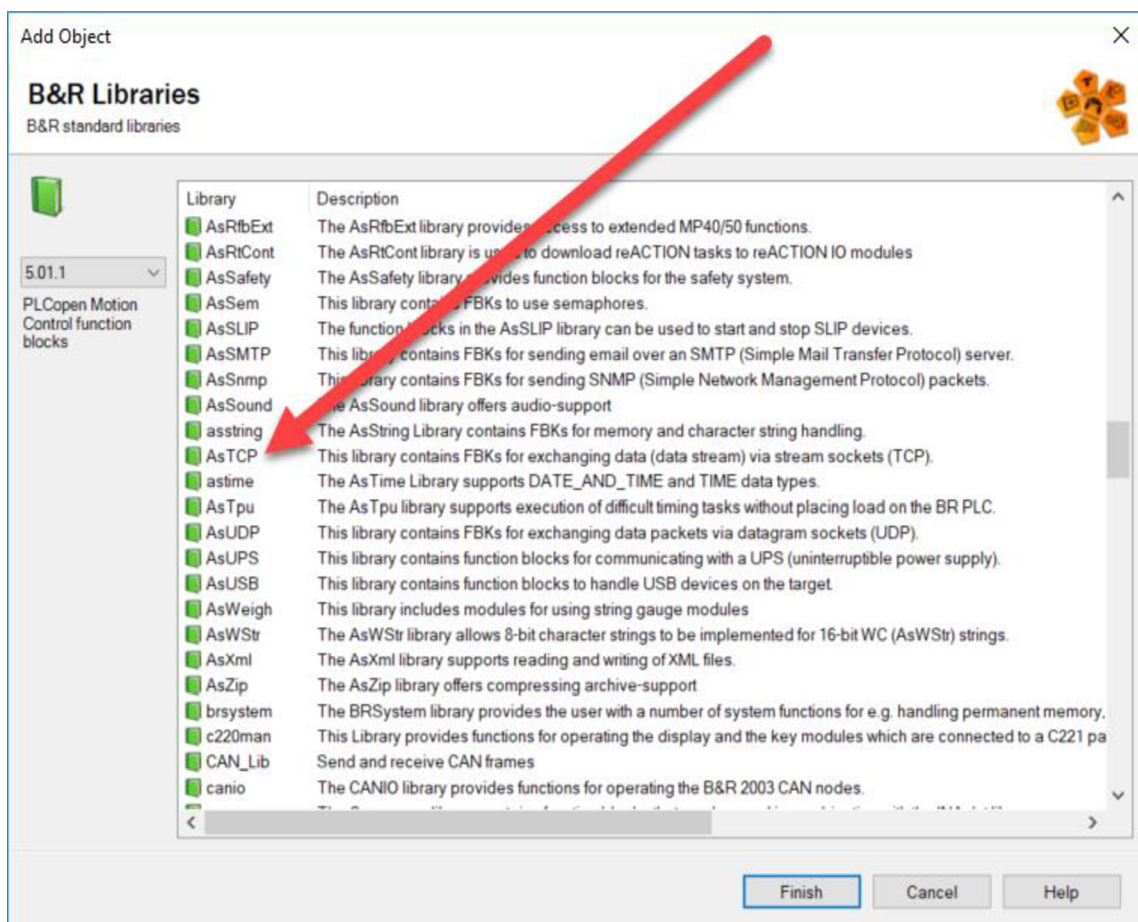
Tieto Library Samples prichádzajú v rôznych jazykoch, avšak nie vo všetkých. Tento objekt však umožňuje užívateľovi pridať aj vlastnú knižnicu. Veľa z týchto programov sa dá importovať a následne upraviť podľa vlastných požiadaviek a predstáv.

6.3 IMPORTOVANIE KNIŽNÍC

Aby sme mohli pracovať s TCP/IP komunikáciou, je nutné využívať správne knižnice. Automation Studio ponúka mnoho knižníc na prácu s TCP/IP, avšak veľa z nich sú už zastarané a slúžia na kooperáciu so staršími projektami. Naš program bude používať knižnicu *AsTCP*, ktorá obsahuje funkčné bloky pre data stream prostredníctvom soкетов.

Na importovanie knižnice klikneme na Libraries v našom projekte v *Logical View* a v *Toolboxe* vyberieme B&R Libraries. Môžeme použiť aj filter na knižnice v sekcii *Toolboxu Programmable Object Units*.

Po dvojkliku na B&R Libraries sa nám otvorí okno s ponukou pridať objekt. Tu si môžeme vybrať z širokej ponuky knižníc ktoré nám Automation Studio ponúka. Nájdem knižnicu *AsTCP* a klikneme Finish. Knižnicu môžeme pridať aj do priečinku programu a následne ju pomocou kurzoru pretiahnuť a vložiť do Libraries.



Obr.32 Rozhranie Add Object po rozkliknutí B&R Libraries

Ďalšia knižnica, ktorú budeme potrebovať je knižnica umožňujúca prácu s premennými typu string. Naše správy vo forme string budeme posielat' pomocou TCP/IP do RobotStudia. Automation Studio nám dáva na výber niekoľko knižníc umožňujúcich prácu s reťazcami, avšak pre najnovšie aplikácie odporúča používanie knižnice *AsBrStr*. Ostatné knižnice existujú iba na zabezpečenie kompatibility. Knižnicu pridáme rovnakým spôsobom, ako bolo popísané.

6.4 PRÁCA S KNIŽNICAMI

Po importovaní knižníc by sme ich mali vidieť v *Logical View*, v sekcii Libraries. Máme viacero možností, ako prísť k jednotlivým funkciám a premenným, ktoré knižnice ponúkajú. Jednou z možností je kliknutie na znak + nachádzajúci sa pred názvom knižnice. Ďalšou je dvojklik ľavým tlačidlom myši. Tu môžeme vidieť jednotlivé funkcie a funkčné bloky, výstupy ale tiež premenné a ich typy.

Name	Type	Constant	Replicable	Value	Description [1]
tcpERR_INVALID_IDENT	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32600	invalid ident
tcpERR_NOMORE_IDENTS	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32601	no more idents
tcpERR_ALREADY_EXIST	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32602	socket already exists
tcpERR_PARAMETER	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32603	invalid parameter, check parameter
tcpERR_INVALID_IOCTL	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32604	io control not supported
tcpERR_NO_DATA	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32605	no data to receive
tcpERR_SENTLEN	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32606	length sent error, call again
tcpERR_WOULDBLOCK	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32607	would block, call again
tcpERR_INVALID	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32608	invalid argument
tcpERR_NOT_CONNECTED	UINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	32609	socket is not connected

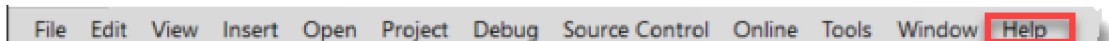
Obr.33 Deklarácia premenných knižnice AsTCP

Pri funkčných blokoch môžeme pozorovať ich vstupné a výstupné parametre a ich typy. Zároveň každý obsahuje stručný popis, rovnako ako aj premenné.

Name	Type	Reference	Scope	Constant	Retain	Replicable	Redundancy	Value	Description [1]
TcpOpen		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Usable in context		opens a TCP socket
TcpOpenSsl		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Usable in context		opens a TCP socket with SSL
TcpServer		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Usable in context		starts a TCP server
TcpClient		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Usable in context		establishes a connection
enable	BOOL	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			enables execution
ident	UDINT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			identifier returned from server
pServer	UDINT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			pointer to the ip address
portserv	UINT	<input type="checkbox"/>	VAR_INPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			port number of server
status	UINT	<input type="checkbox"/>	VAR_OUTPUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			execution status: Error
i_state	UINT	<input type="checkbox"/>	VAR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			internal variable
i_result	UINT	<input type="checkbox"/>	VAR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			internal variable
i_tmp	UDINT	<input type="checkbox"/>	VAR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			internal variable
TcpClose		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Usable in context		closes a TCP socket

Obr.34 Deklarácia funkcií a funkčných blokov AsTCP

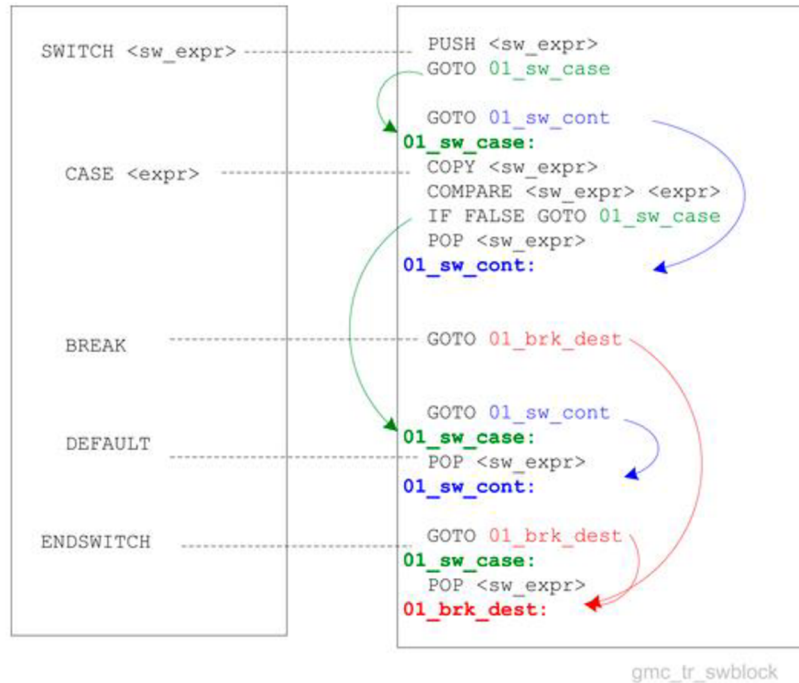
Informácie o knižniciach sa tiež nachádzajú v Help-e. Stačí napísať názov knižnice do kolonky *Search*.



Obr.35 Panel príkazov Automation Studia

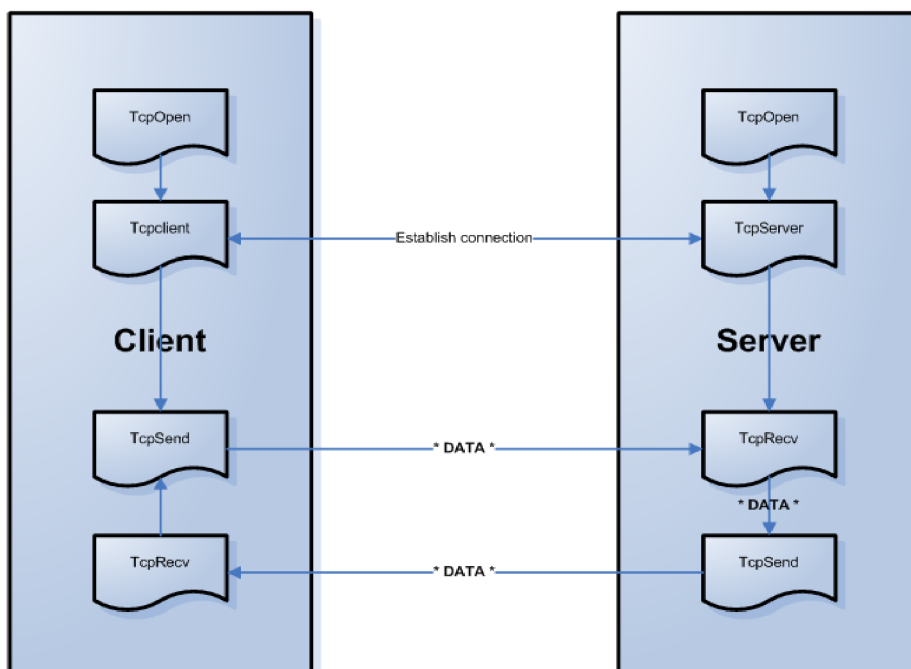
6.5 PRINCÍP KOMUNIKÁCIE

Pri TCP/IP komunikácii je dôležité dodržiavať stanovený postup definície a deklarácie premenných, volania funkcií a funkčných blokov a tiež predávanie parametrov. Pri PLC programe by mal kód prebiehať v štýle stavového automatu, teda za použitia SWITCH – CASE princípu.



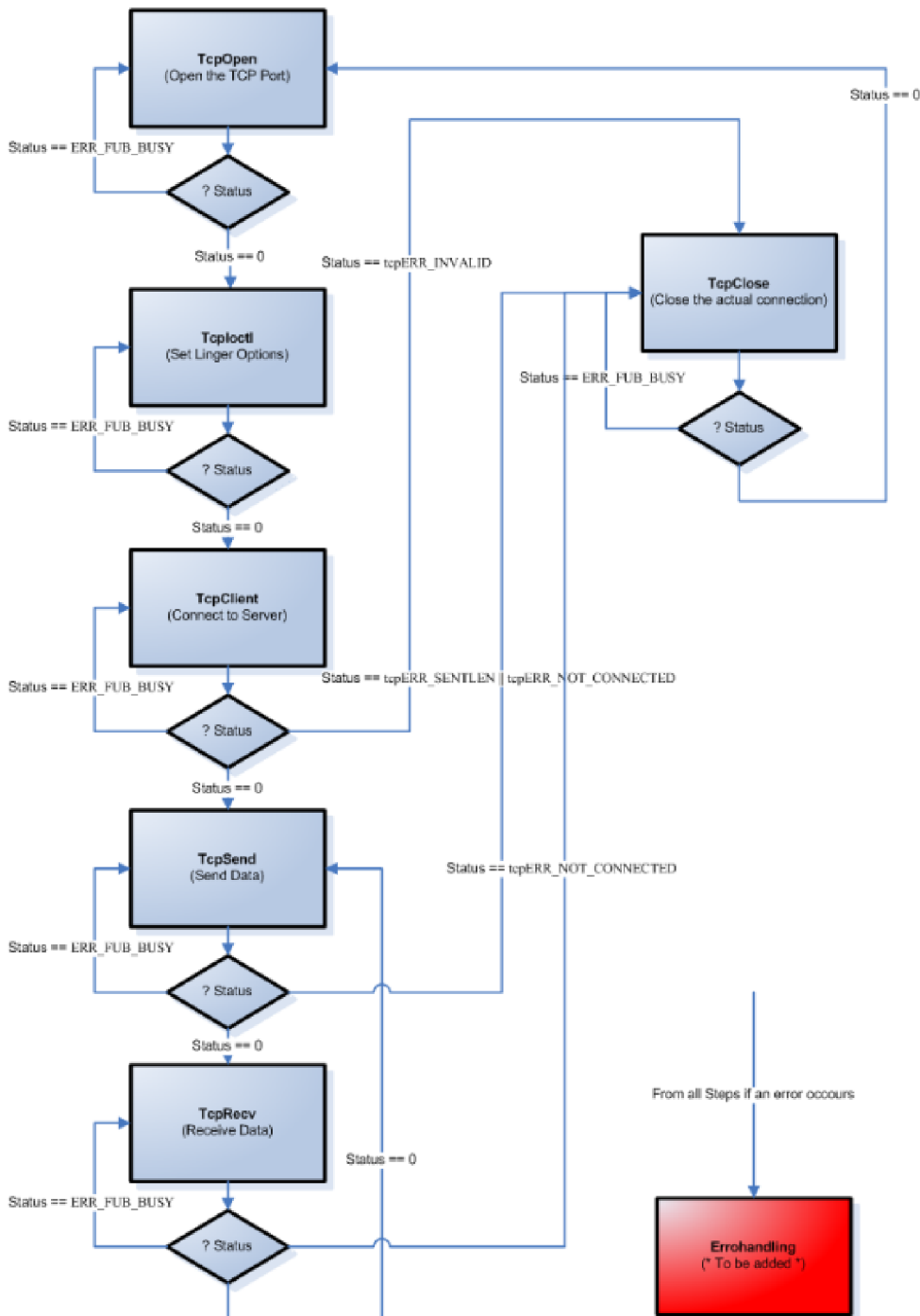
Obr.36 SWITCH – CASE riadiaca štruktúra[42]

Kód komunikácie by mal byť teda napísaný v rámci tohto pravidla. Zároveň musí dodržiavať postupnosť, ako bolo zmienené vyššie. Túto postupnosť popisuje nasledujúci vývojový diagram.



Obr.37 Vývojový diagram TCP/IP komunikácie[42]

Tieto informácie je možné nájsť v Help-e, pri zadaní kľúčových slov. Ďalším dôležitým krokom je správna konfigurácia adresy a portov. Klient a server môžu bežať na oddelených systémoch alebo na jednom spoločnom. V prípade našej simulácie celá komunikácia prebieha na našom osobnom počítači. Preto by mala byť IP adresa nastavená na Localhost, čo znamená 127.0.0.1. Zložitejšie konfigurovanie adresy, ako je špecifikovanie rodiny protokolov alebo jej naviazanie na port, zaobstará Automation Studio, a teda ho nie je potrebné riešiť. Port môžeme zvoliť akýkoľvek, výhodnejšie je však zvoliť vyššie číslo, aby sme si boli istí, že port nie je obsadený. Toto číslo bude pradtavovať rovnako číslo portu na strane servera.



Obr.38 Vývojový diagram TCP/IP komunikácie na strane klienta[42]

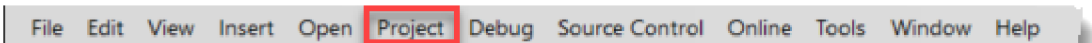
6.6 SPUSTENIE APLIKÁCIE

Po napísaní programu, zadeklarovaní a zedefinovaní premenných môžeme spustiť našu aplikáciu. Na paneli nástrojov klikneme na ikonu *Build* alebo stlačíme klávesu F7.



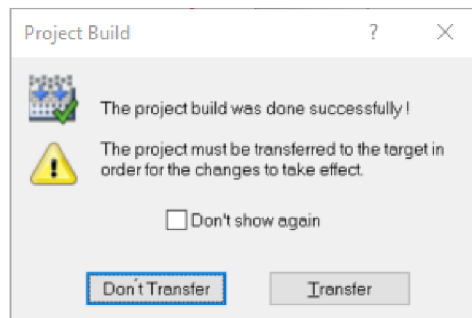
Obr.39 Panel nástrojov Automation Studio

Ďalšou možnosťou, ako spustiť program je cez panel príkazov v Automation studiu. Zvolíme možnosť *Project* a ďalej *Build Configuration* alebo pokračovať do *Project Installation* a zvoliť možnosť *Transfer To Target*.



Obr.40 Panel príkazov Automation Studio

Po spustení sa nám program skontroluje, v okne *Output Results* môžeme vidieť priebeh tvorby aplikácie. V prípade chýb sa budovanie zastaví a neumožní nám nahráť program do virtuálneho PLC. Po dokončení budovania aplikácie sa zobrazí ďalšie okno, ktoré bude čakať na potvrdenie prenosu programu do PLC.

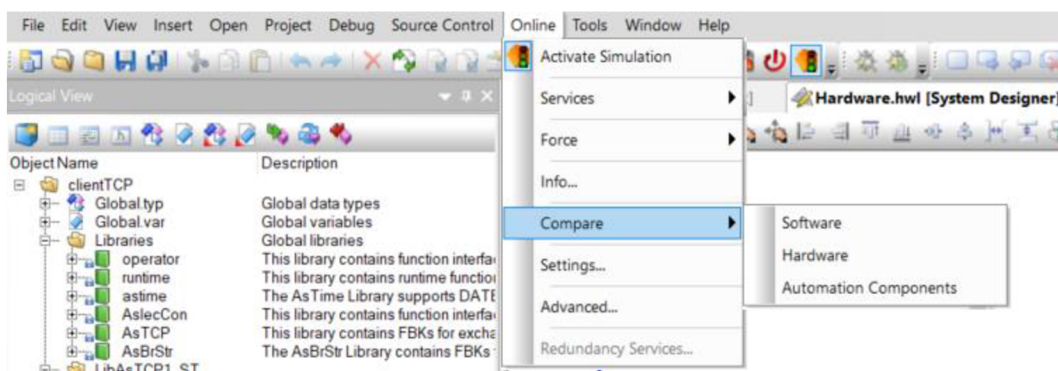


Obr.41 Okno Project Build

Pokiaľ sme zvolili priamo možnosť *Transfer To Target* cez project installation predchádzajúci krok nenastane.

Zvolíme možnosť *Transfer* a zobrazí sa nám ďalšie okno. Toto rozhranie nám umožní nahratie programu do virtuálneho PLC a to zvolením možnosti *Transfer*. Pokiaľ sú v programe rozdiely oproti tomu, ktorý sa nachádza v PLC, či už hardware alebo software, táto skutočnosť sa nám zobrazí.

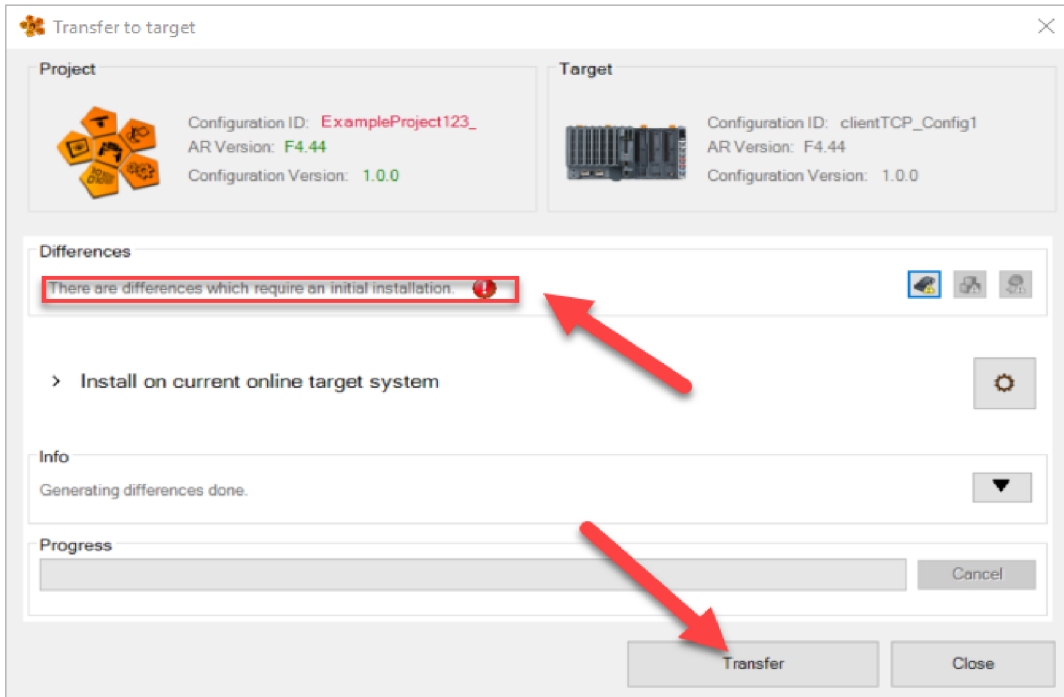
Pokiaľ by sme chceli pozorovať rozdiely medzi aktuálnym programom a tým, ktorý je nahraný v PLC môžeme zvoliť možnosť *Online* na paneli príkazov a následne zvoliť možnosť *Compare*. Tu si budeme môcť vybrať, či chceme sledovať hardwareové, softwareové rozdiely alebo rozdiely v komponentoch.



Obr.42 Pozorovanie odlišností

Pokiaľ však nemáme spustenú simuláciu alebo pripojené PLC, nič sa nám na porovnanie nezobrazí.

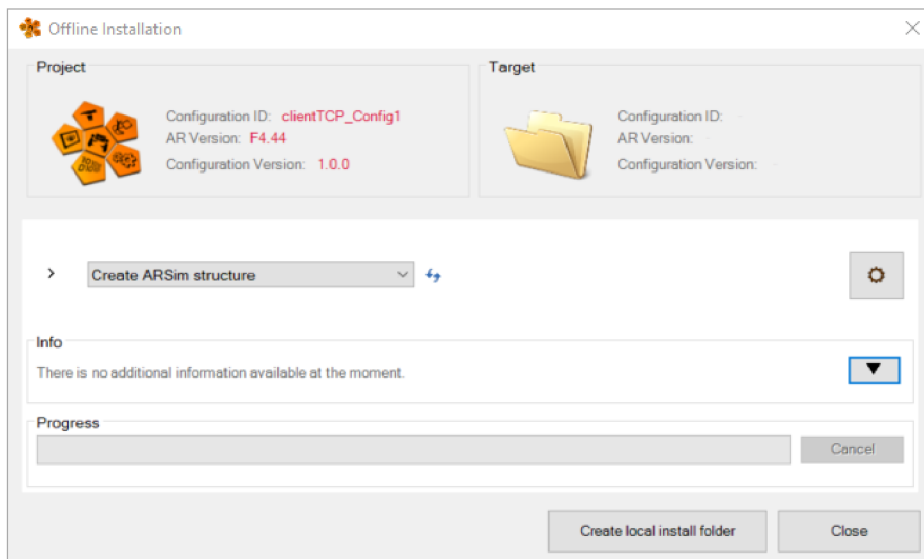
Po zvolení *Transfer Automation Studio* vymaže pôvodný program z Targetu a prepíše ho aktuálnym nahrávaným. Zároveň nás program na túto skutočnosť upozorní po zvolení tejto možnosti.



Obr.43 Rozhranie nahrávania programu

Po dokončení prenosu, ktorého priebeh môžeme sledovať pomocou *Progress* baru, sa simulácia spustí a program beží na virtuálnom PLC.

Ďalšou možnosťou, ako nahráť program do PLC je prostredníctvom offline inštalácie. Offline inštalácia vytvorí Compat Flash pre PLC. V prípade simulácie je vytvorená iba simulačná štruktúra na lokálnom disku. Offline inštaláciu spustíme prostredníctvom voľby *Project* na paneli funkcií, následne zvolíme možnosť *Project Installation* a nakoniec *Offline Installation*. Po zvolení tejto možnosti sa nám zobrazí rozhranie, v ktorom vytvoríme priečinok pre inštaláciu programu, obsahujúci program.



Obr.44. Rozhranie Offline inštalácie

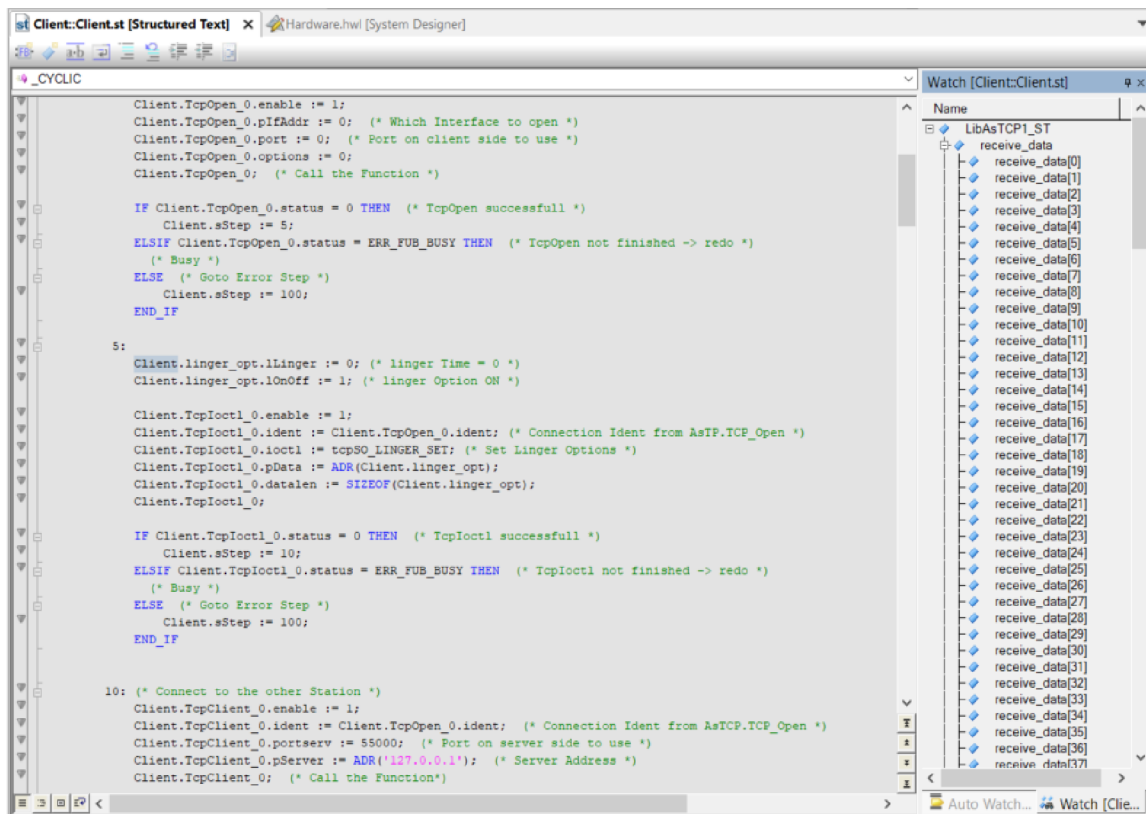
6.7 POUŽÍVANIE WATCH FUNKCIE

Pri analyzovaní programu je veľmi dôležité poznať stavy premenných. V Automation Studiu môžeme využiť funkciu *Watch* aby sme monitorovali ich aktuálne hodnoty. Túto funkciu môžeme spustiť rôznymi spôsobmi. Jedným z nich je zvoliť *Open* na paneli príkazov a následne zvoliť *Watch*. Ďalším spôsobom je použiť klávesy CTRL + W alebo kliknúť na ikonu na paneli nástrojov. Funkciu *Watch* nie je možné spustiť pokiaľ nebeží simulácia.



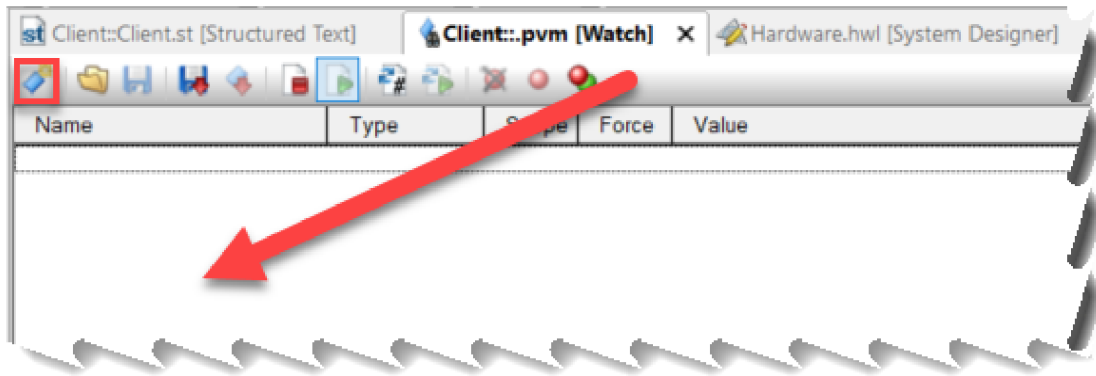
Obr.45 Možnosti spustenia Watch funkcie

Pokiaľ otvoríme *okno Watch* priamo v jednej zo záložiek, premenné môžeme pridať tak, že ich vyznačíme a pretiahneme pomocou kurzoru do okna *Watch-u*. *Watch* vypneme tak, že znovu klikneme na ikonku *Lupy*.



Obr.46 Okno Watch v záložke Aplikácie

V prípade, že zvolíme možnosť cez *Open*, otvorí sa nám nová záložka *pvm[Watch]*. Premenné tu môžeme pridať kliknutím pravým tlačidlom myši do prázdneho poľa a zvolením možnosti *Insert Variable* alebo pomocou ikony na lokálnom paneli nástrojov záložky.



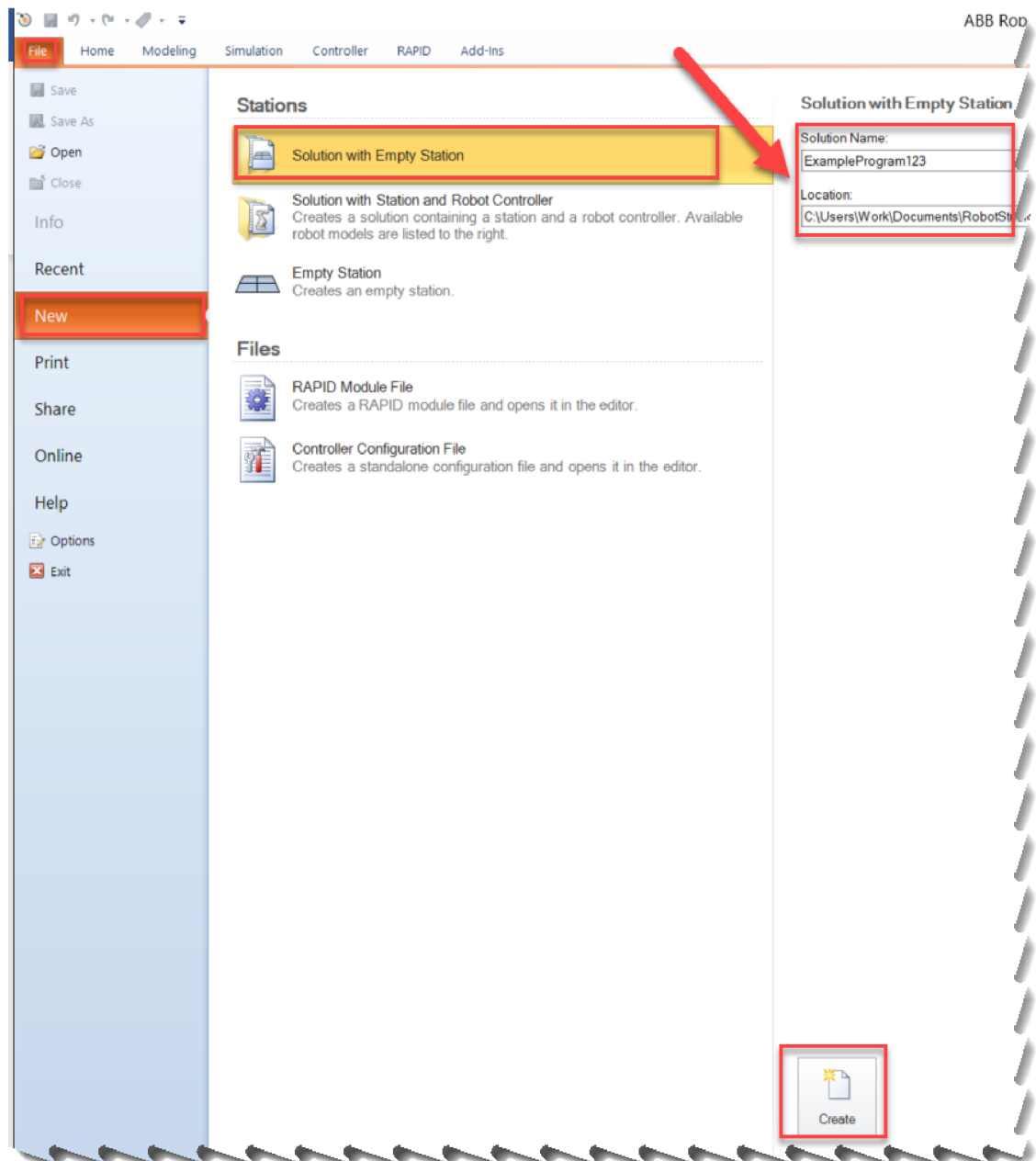
Obr.47 Záložka Watch, možnosti pridania premenných

7 APLIKÁCIA ROBOTSTUDIO

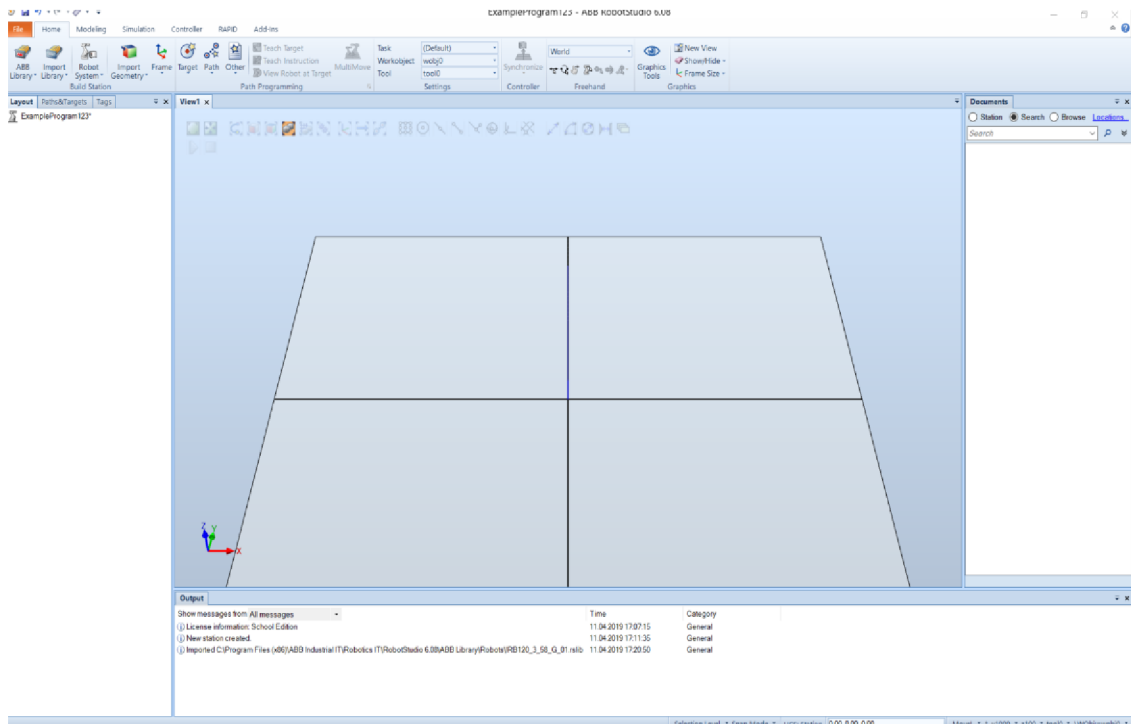
Software RobotStudio poslužil na vytvorenie strany serveru. Využíval som verziu 6.08 a licenciu School Edition. Rád by som upozornil, že je potrebné aby cesta vedúca do priečinku, v ktorom sa nachádza program, neobsahovala v názve žiadne medzery. V opačnom prípade sa Controller v RobotStudiosu nespustí.

7.1 ZALOŽENIE PROJEKTU

Po spustení aplikácie RoboStudio sa ocitneme v rozhraní *File* a v záložke *New*. Tu zvolíme možnosť *Solution with Empty Station*, pomenujeme (popríklad zvolíme uložisko) a následne klikneme na tlačidlo *Create*.



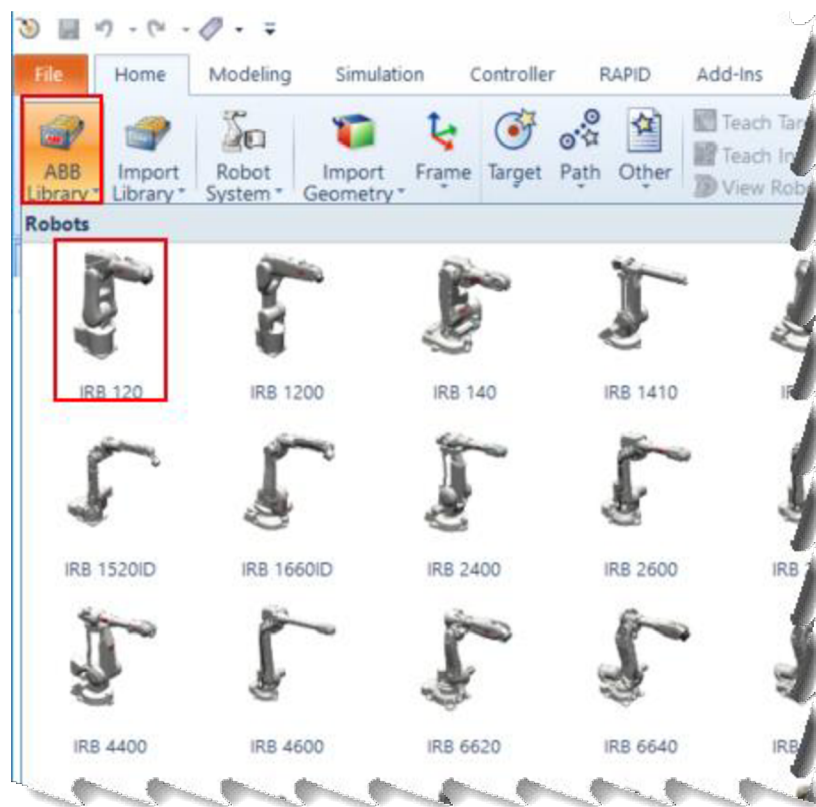
Obr.48 Tvorba stanice v RoboStudiosu



Obr.49 Prázdna stanica

7.2 IMPORTOVANIE ROBOTA

Keď už máme vytvorenú prázdnu stanicu môžeme vložiť robota. Robot, s ktorým budeme pracovať je model IRB120. Na paneli funkcií klikneme na záložku *ABB Library* a nájdem robota IRB120.

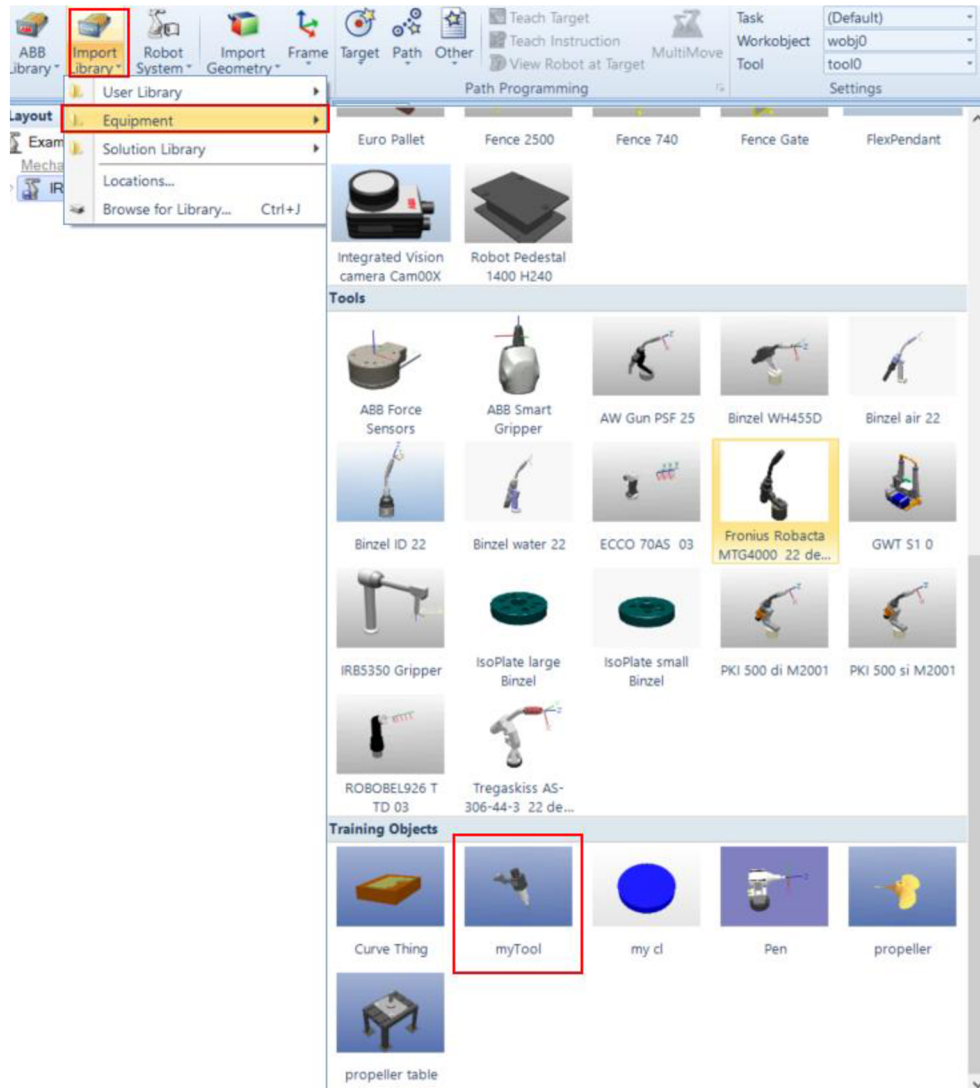


Obr.50 Vloženie robota

Po zvolení príslušného robota sa nám zobrazí okno, v ktorom dostaneme na výber z jeho prevedení. My budeme pracovať s štandardnou variantou IRB120.

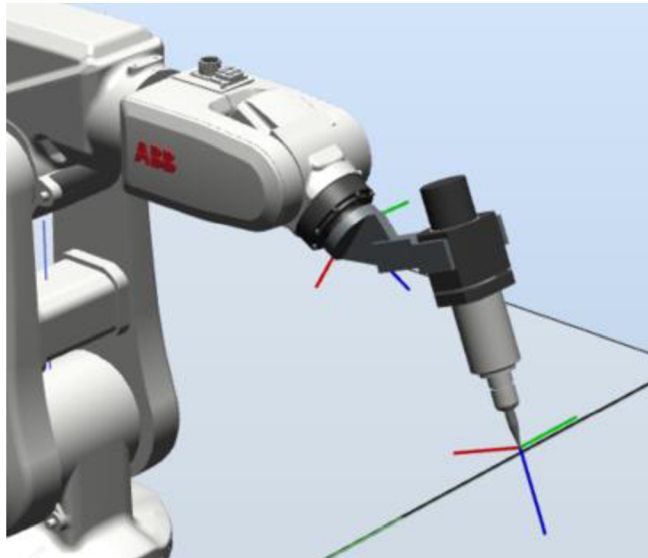
7.3 PRIDANIE NÁSTROJA

RobotStudio umožňuje pridať širokú škálu nástrojov a objektov. Nástroj pre nášho robota pridáme tak, že klikneme na záložku *Import Library*, potom zvolíme zložku *Equipment* a vyberieme nástroj *myTool* z pomedzi *Training objects*.



Obr.51 Voľba nástroja

Tento nástroj sa nám však nespojí s robotom automaticky. Vytvorí sa v nulových súradniciach a bude potrebné ho k robotu pridať manuálne. To spravíme tak, že ľavým kurzorom myši zvolíme a pretiahneme položku *myTool* na položku nášho robota, v tomto prípade IRB120. Program sa nás opýta, či si prajeme zmeniť polohu nášho nástroja.



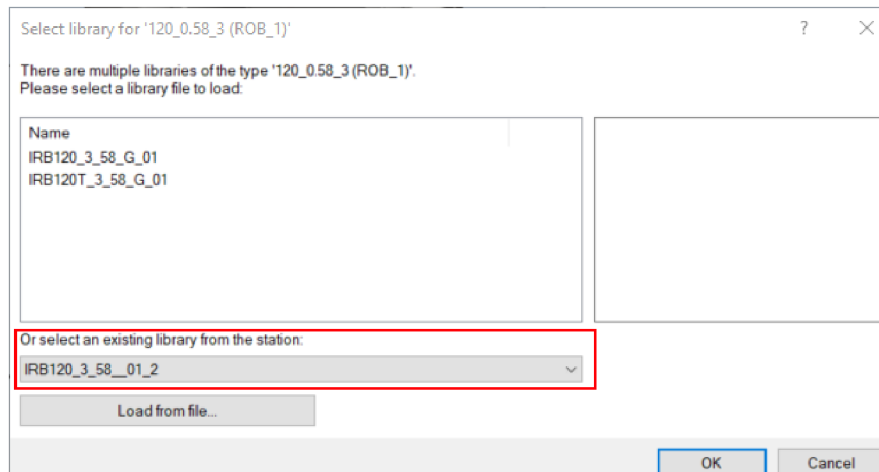
Obr.52 Správne umiestnený nástroj

7.4 PRIDANIE SYSTÉMU

Aby sme mohli robota ovládať, potrebujeme pridať systém. To prevedieme tak, že na paneli funkcií zvolíme záložku *Robot System* a ďalej možnosť *New System*.

Obr.53 Pridávanie nového systému

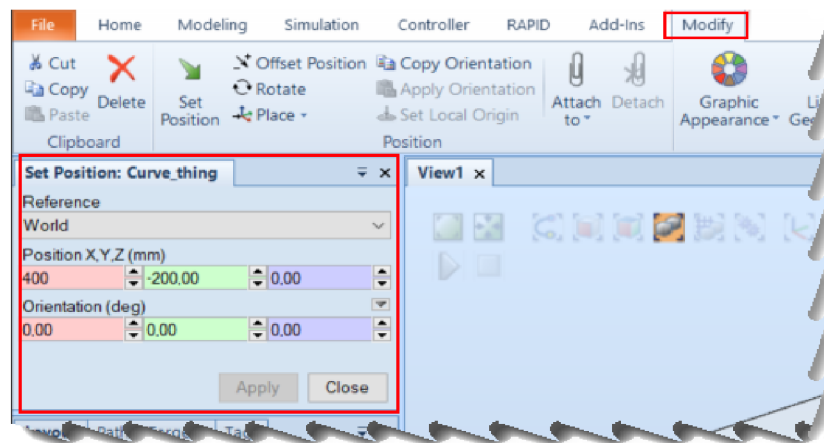
Po potvrdení sa náš systém začne vytvárať. Nakoniec sa nás program opýta, akú knižnicu má načítať. Zvolíme možnosť, aby načítal knižnicu z našej stanice, potvrdíme a dokončíme.



Obr.54 Voľba knižnice pre stanicu

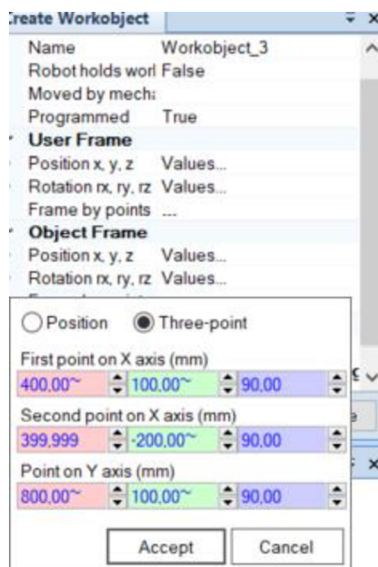
7.5 TVORBA WORKOBJECTU

Predtým, než vytvoríme workobject (čo je pracovný priestor), by sme mali pridať objekt ku ktorému ho vytvoríme. Rovnako ako sme pridali nástroj, pridáme cez *Import Library – Equipment – Curve Thing*. Objekt sa nám však nevytvoril v mieste, v ktorom by sme chceli. Jeho pozíciu upravíme prostredníctvom funkcie *Set Position*. Tú nájdeme v záložke *modify* na hlavnom paneli RobotStudia. Po kliknutí na funkciu sa nám zobrazí rozhranie na určovanie pozície. Tu môžeme upraviť súradnice, aby sa objekt nachádzal v rovine s robotom.



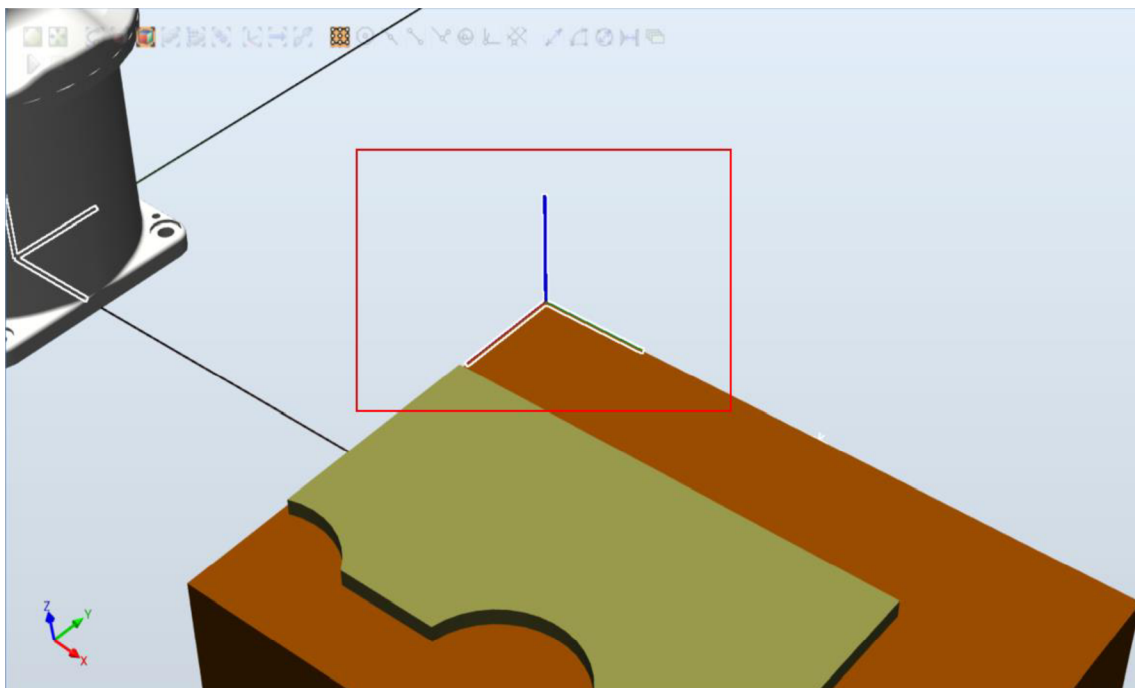
Obr.55 Zmena pozície objektu

Workobject môžeme vytvoriť viacerými spôsobmi. Jeden z nich je, že zvolíme záložku *Other*, na paneli funkcií a následne zvolíme možnosť *Create Workobject*. Pokiaľ chceme, môžeme ho ľubovoľne pomenovať. My sa zameriame na jeho definíciu pomocou bodov. Tú prevedieme tak, že v rozhraní *Create Workobject* v záložke *User Frame* zvolíme možnosť *Frame by points* a nakoniec *Three-point*. Vďaka objektu, ktorý sme vytvorili, môžeme súradnicový systém workobjectu vzťahovať k nemu. V okne *View1* zvolíme z funkcií *Snap Object* a *Body Selection*. Kliknutím do jednotlivých okienok pre súradnice a následným zvolením súradníc pomocou kurzoru na našom objekte nastavíme súradnice pre náš workobject.



Obr.56 Zadávanie súradníc workobjectu

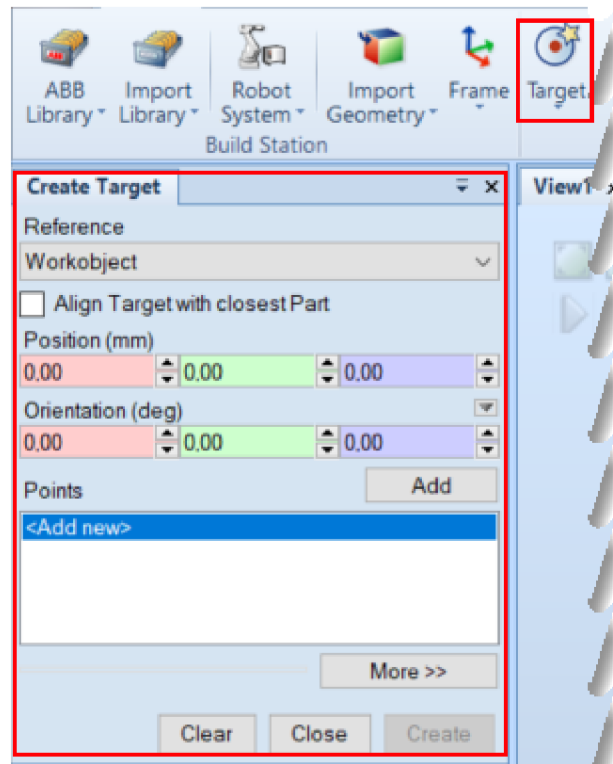
Týmto sme vytvorili súradnicový systém pre nášho robota, ktorý budeme používať.



Obr.57 Vytvorený workobject

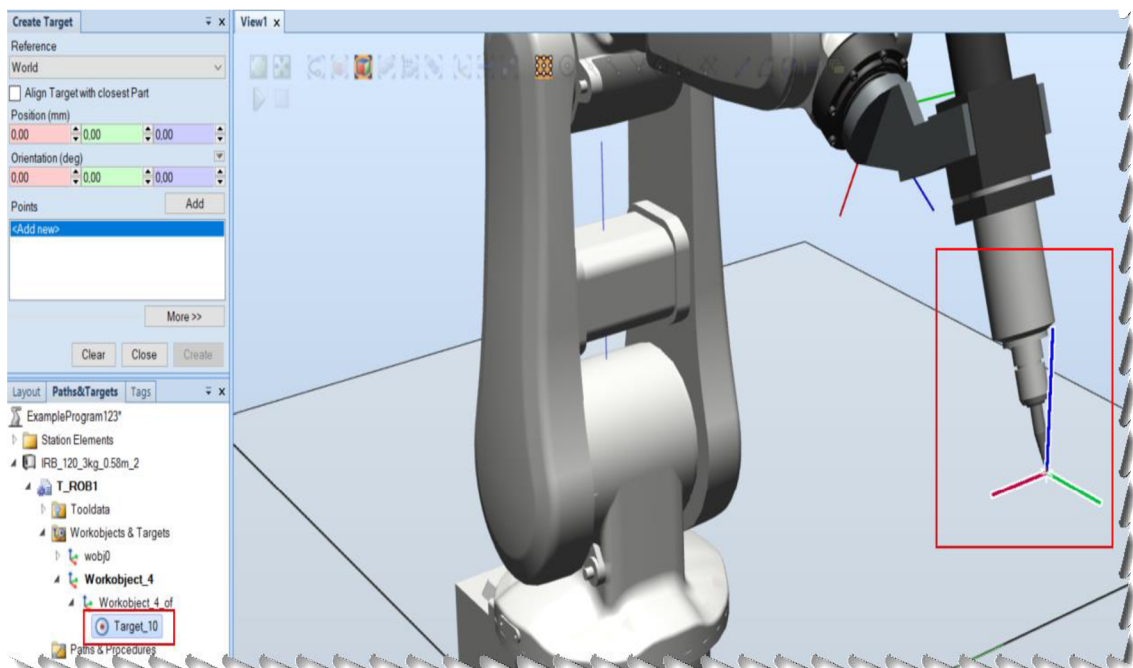
7.6 TVORBA TARGETU

Targety sú pozície, na ktoré sa robot môže presunúť. V tejto práci však tieto pozície budeme posielat' pomocou TCP/IP. Napriek tomu si vytvoríme Target v základnej pozícii robota, aby sa do nej mohol po ukončení komunikácie vrátiť. Rovnako v tomto prípade existuje mnoho spôsobov, ako vytvoriť Target. Môžeme ho vytvoriť pomocou RAPID kódu alebo obdobne ako sme vytvorili Workobject. Na paleni funkcii zvolíme *Target* a potom *Create Target*. Zobrazí sa nám rozhranie, v ktorom musíme zadať súradnice, kde sa bude Target nachádzať.



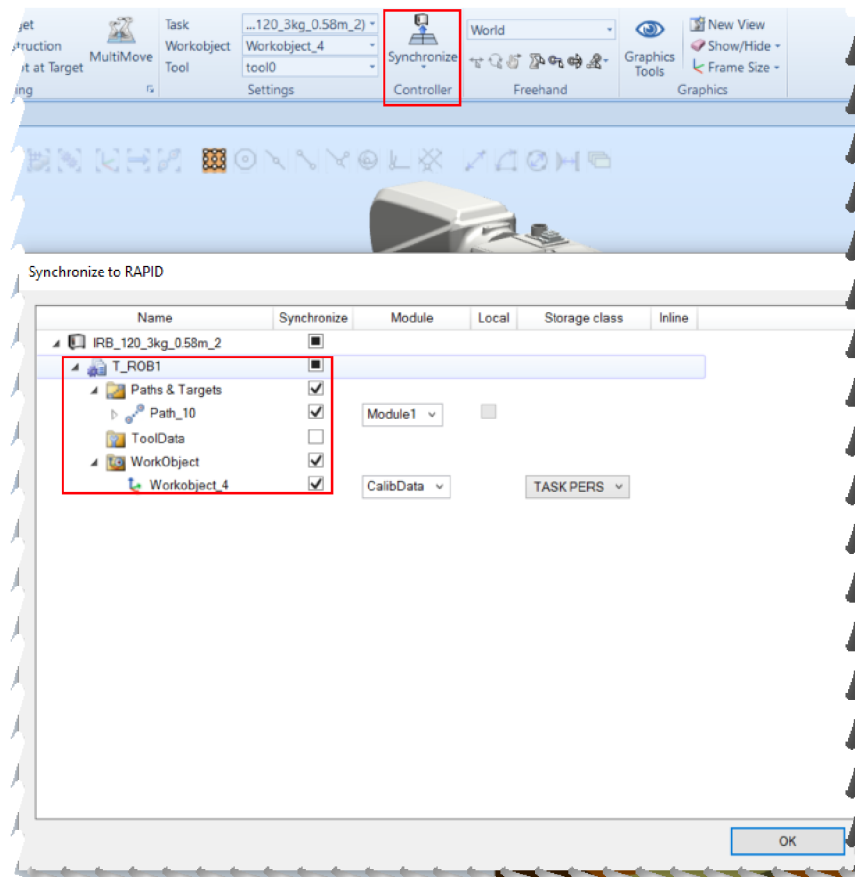
Obr.58 Zadávanie súradníc Targetu

Súradnice zadáme obdobne ako pri Workobjekte. Znovu zvolíme možnosť *Body Selection* a *Snap Object* v okne *View*. Rovnako ako v predchádzajúcej úlohe, najskôr klikneme do poľa pre pozíciu a následne pomocou kurzoru vyznačíme polohu nástroja v základnej polohe, pre pozíciu aj pre orientáciu. Stačí prvé pole, RobotStudio ostatné súradnice dopočíta. Nakoniec zvolíme create.



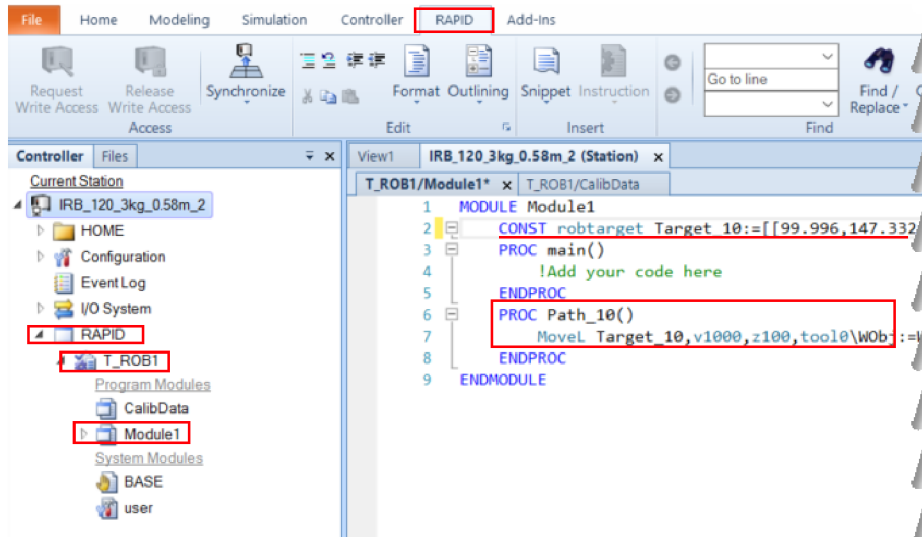
Obr.59 Vytvorený Target základnej polohy

Po tom ako sme vytvorili Target, musíme ho pridať do Path-u, teda cesty. Pokiaľ by sme to neurobili, RAPID by nám po zosynchronizovaní nevygeneroval kód Targetu. Target do Path pridáme tak, že ľavým tlačidlom myši klikneme na Target a zvolíme možnosť *Add to new path*. Teraz musíme zosynchronizovať Controller s RAPID-om, aby nám automaticky vygeneroval kód nášho Targetu, teda aby sme zmeny prevedené v prostredí preniesli do RAPID-u. To prevedieme tak, že z panela funkcií zvolíme voľbu *Synchronize* a potom *Synchronize to RAPID*. Zobrazí sa nám okno, v ktorom potvrdíme zmeny.



Obr.60 Synchronizácia Controlleru

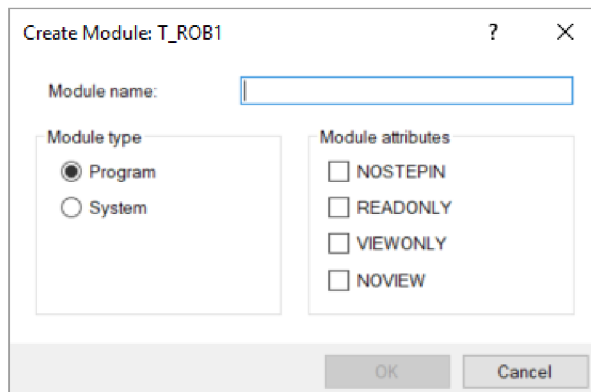
Teraz nám RAPID vygeneroval kód Targetu, ale aj proces, ktorým robota do tejto polohy dostane, teda Path, do ktorého sme náš Target pridali. Do RAPIDU sa dostaneme cez hlavný panel zvolením možnosti *RAPID* a následným rozkliknutím záložky s rovnakým pomenovaním. Tu sa nám zobrazí ďalšia záložka, predstavujúca *task*, s názvom T_ROB. Pokračujeme v rozkliknutí a zobrazí sa nám obsah tasku. Ten je rozdelený na program a system moduly. Náš program sa nachádza v *Program Modules*, v Module1.



Obr.61 RAPID rozhranie

7.7 RAPID PROGRAM

Kód TCP/IP komunikácie budeme písať do procesu main() v Module1. Bolo by však výhodné vytvoriť ďalšie moduly a procesy, aby bol kód prehľadnejší. Nový modul pridáme tak, že klikneme ľavým tlačidlom myši na task T_ROB1 a zvolíme možnosť *New Module*. Otvorí sa nám okno, v ktorom môžeme zvoliť jeho typ a atribúty.



Obr.62 Tvorba Modulu

7.8 RAPID WATCH A DEBUG MODE

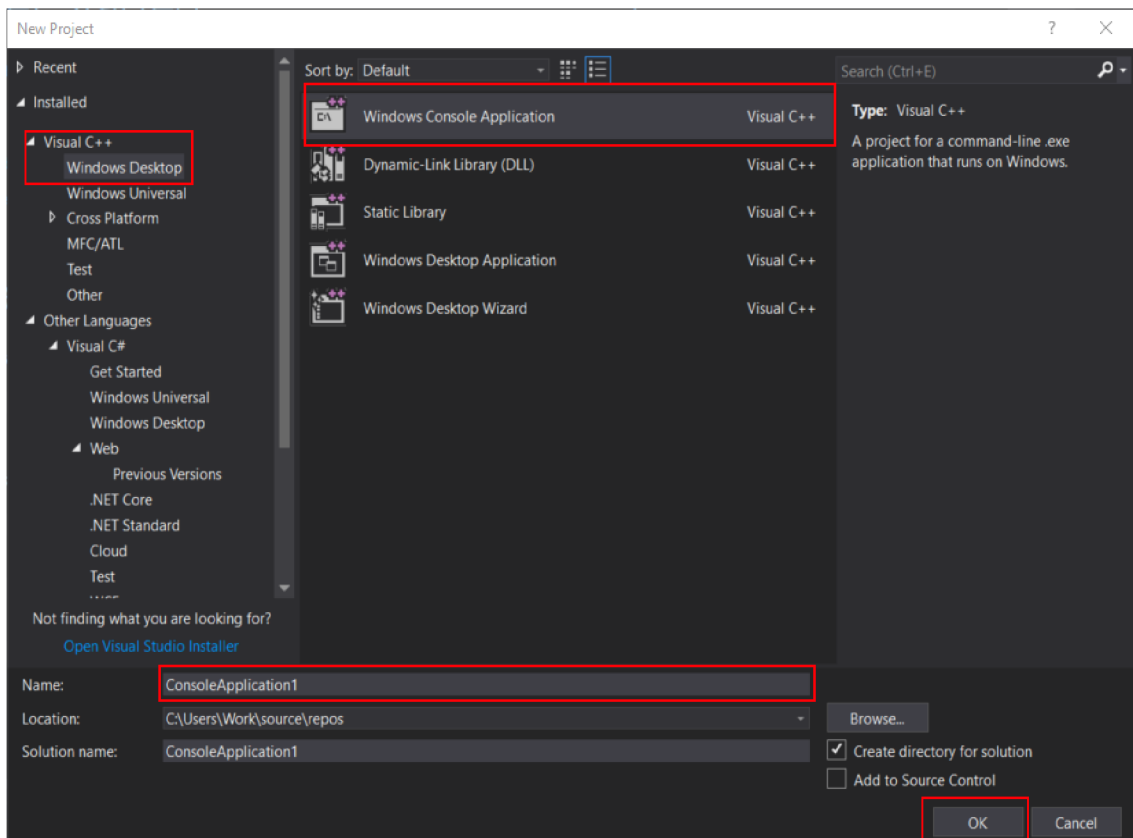
Okrem klasického debuggovania ponúka RAPID aj funkciu WATCH, pomocou ktorej môžeme sledovať aktuálny stav premenných. Do tejto funkcie sa dostaneme buď prostredníctvom okna RAPID Watch, alebo tak, že pridáme premennú z modulu. To spravíme kliknutím ľavým tlačidlom myši a zvolením možnosti *Add Watch*. Pokiaľ bude teda simulácia aktívna, môžeme takto sledovať hodnoty premenných v našom programe. V prípade debuggovania je viacmenej zbytočné používať ju, nakoľko pri ňom môžeme stav kontrolovať v každom kroku. Debug mód spustíme prostredníctvom tlačidla štart v RAPID záložke hlavného panelu. Samozrejme, je nutné pridať breakpoint, inak spustíme iba simuláciu. Taktiež môžeme presúvať pointery v programe manuálne prostredníctvom funkcie *Program Pointer*, rovnako v záložke RAPID. Tieto funkcie veľmi uľahčujú budovanie programu a hľadanie chýb v jeho priebehu.

8 MICROSOFT VISUAL STUDIO

Na komunikáciu prostredníctvom protokolu TCP/IP s RobotStudiom môžeme využiť aj inú aplikáciu. Ďalší zo spôsobov, ako sa spojiť s RobotStudiom a poslať mu inštrukcie, je naprogramovať stranu klienta v Microsoft Visual Studiu.

8.1 TVORBA APLIKÁCIE

Aplikáciu v Microsoft Visual Studiu vytvoríme tak, že po spustení samotného programu zvolíme na lište funkcií možnosť *File*, následne *New* a potom *Project*. Nie sme limitovaný voľbou jazyka, môžeme zvoliť prakticky takmer akýkoľvek, avšak treba mať doinštalované potrebné doplnky. V našej aplikácii budeme používať jazyk C++. Aby sme vytvorili aplikáciu zvolíme v ponuke *Visual C++* možnosť *Windows Desktop* a následne si vyberieme typ aplikácie, v tomto prípade *Windows Console Application*. Projekt pomenujeme a potrdíme OK.



Obr.63 Tvorba nového projektu

Po potvrdení sa nám vytvorí aplikácia, ktorá bude obsahovať metódu *main*. Naš program nebude potrebovať veľa knižníc. Pokiaľ chceme tvoriť túto komunikáciu na operačnom systéme Windows, budeme musieť zahrnúť v programe knižnicu *WS2tcpip*, ktorá nám umožní pracovať so socketami. [51,52]

```
1 #include<iostream>
2 #include<WS2tcpip.h>
```

Obr.64 Vloženie knižnice

8.2 KONFIGURÁCIA PRIPOJENIA

Prvým krokom bude vytvorenie socketu, ktorý nám bude slúžiť ako bod pripojenia. Ten vytvoríme ako typ SOCKET, pričom preberá 3 parametre. Prvým je rodina adresy. Keďže používame verziu IPV4, nastavíme ju ako AF_INET. Ďalší parameter je typ socketu. Chceme vytvoriť TCP socket, takže parameter, ktorý predáme bude SOCK_STREAM. Posledným parametrom je špecifikácia protokolu, ktorý nastavíme na nulu, keďže vieme, že chceme používať TCP. Túto hodnotu by sme nastavovali keby sme používali tzv. Raw sockety, čo sú sockety nezávislé na protokole, napríklad v takom prípade aký používame teraz.

```
SOCKET my_SOCKET = socket(AF_INET, SOCK_STREAM, 0);
```

Obr.65 Tvorba socketu

Knižnica, ktorú sme pridali obsahuje funkcie na prácu so špeciálnym typom adresovej štruktúry, pomocou ktorej naviažeme port a adresu na socket. Zadeklarujeme teda štruktúru typu sockaddr_in. Vďaka štruktúre máme prístup k poliam, ktoré potrebujeme nastaviť. Prvú nastavíme rodinu adresy portu, na ktorý sa chceme pripojiť. Bude rovnaká, ako nášho portu, teda AF_INET. Ďalší musíme špecifikovať port, na ktorý sa chceme pripojiť. Musíme však použiť funkciu htons, ktorá nám zaistí konverziu do správneho tvaru, nakoľko nemôžeme predať číslo portu ako integer, pretože by nebolo v správnom bytovom zoradení. Nakoniec nastavíme IP adresu serveru, na ktorý sa budeme pripájať. Nastavíme ju ako INADDR_ANY, nakoľko pracujeme na našom lokálnom systéme, čo je rovnaké ako keby sme sa chceli pripojiť na adresu 0.0.0.0.

```
27     sockaddr_in hint;
28     hint.sin_family = AF_INET;
29     hint.sin_port = htons(54000);
30     hint.sin_addr.S_un.S_addr = INADDR_ANY;
```

Obr.66 Naviazanie adresy a portu na socket

Na prijímanie dát si musíme založiť buffer. Je dôležité aby sme pred každým prijatím vyčistili buffer od predchádzajúcich dát.

Pokiaľ chceme dáta odosielať, použijeme funkciu send, pričom nesmieme zabudnúť pridať nulujúci znak pri počte odosielaných dát, v prípade prijímania používame funkciu recv. Preberajú podobné parametre, pričom ako socket im predáme náš vytvorený my_SOCKET. Funkcia recv preberá buffer ako argument pre prichádzajúce dáta a funkcia send preberá ako argument užívateľský vstup cez konzolu. Tieto funkcie vracajú hodnotu typu int, ktorá signalizuje či prebehli úspešne. V prípade úspešného priebehu vrátia hodnotu predstavujúcu počet bytov, ktoré sme prijali alebo odoslali. V prípade úspešného zatvorenia komunikácie recv vráti nulu. Pokiaľ došlo k chybe, obe funkcie vrátia hodnotu SOCKET_ERROR.

```
int sendResult = send(sock, userInput.c_str(), userInput.size() + 1, 0);
```

Obr.67 Funckia send

```
int bytesRecieved = recv(sock, buf, 4096, 0);
```

Obr. 68 Funckia recv

Po skončení komunikácie zatvoríme socket.

8.3 KOMUNIKÁCIA ROBOTSTUDIO A MICROSOFT VISUAL STUDIO

Na strane RobotStudia vytvoríme jednoduchú aplikáciu na TCP komunikáciu. ABB RobotStudio nemá konzolový vstup, takže hodnoty budeme zapisovať prostredníctvom funkcie Watch. Tento program je možné jednoducho upraviť tak, že budeme odosielať súradnice alebo potvrdenia aby robot uskutočnil nejakú akciu. Prvý vždy spustíme server, ktorý bude čakať na pripojenie klienta.



```

C:\Users\Work\source\repos\tcpipCLIENT\Debug\tcpipCLIENT.exe
Ja>Ahoj server!
SERVER> Ahoj Klient!
Ja>Suradnice12345
SERVER> Ahoj Klient!
Ja>Suradnice a text
SERVER> Odpoved123456
Ja>
  
```

Obr.69 Komunikácia Server/Klient Console App

	Name	Value	Type	Source
	message	"Suradnice12345\00"	string	System1/RAPID/T_ROB1/MOD1/message
	sendMessage	"Odpoved123456"	string	System1/RAPID/T_ROB1/MOD1/sendMessage

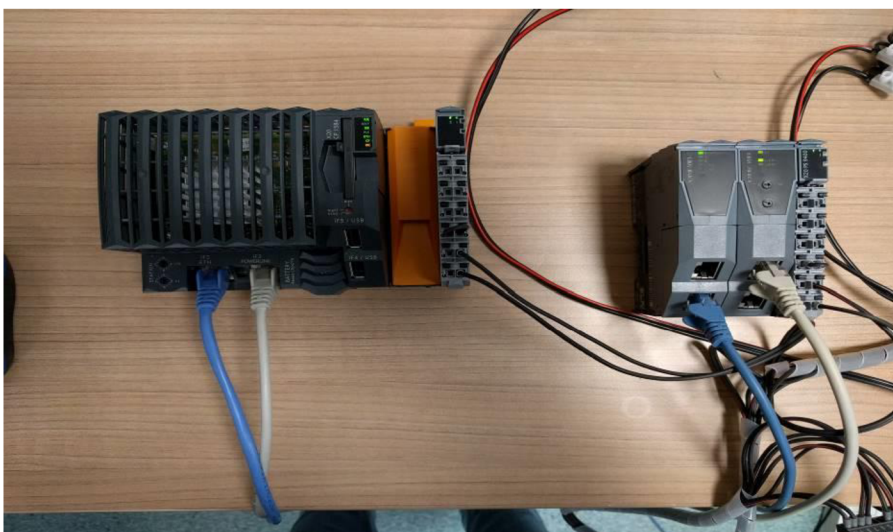
Obr.70 Komunikácia Server/Klient RobotStudio-Watch

9 KOMUNIKÁCIA S VYUŽITÍM PROFINETU

Posledným spôsobom komunikácie, ktorý táto práca popisuje a ktorý bude demonštrovaný na praktickom prevedení v laboratóriu bude komunikácia prostredníctvom siete Profinet. Profinet využíva ako prenosovú vrstvu Ethernet, čo je samozrejme rovnako zapísané v norme IEEE 802.3. Komunikačné zariadenia v tejto sieti sú si rovnocenné, čo znamená, že nie sú rozdelené na nadriadené a podriadené alebo master-slave. Prístup k sieti nie je nutné riadiť, nakoľko bezkolízna komunikácia je daná použitím prepínačov. Tieto sieťové prepínače alebo inak nazývané switche pracujú na úrovni druhej vrstvy komunikačného modelu ISO/OSI. Vďaka použitiu prepínačov je teda možné flexibilne navrhnuť topológiu siete a preto sa objavujú v praxi rôzne kombinácie základných topológií. Konceptia architektúry komunikačného systému Profinet je koncipovaná ako modulárna, takže jeho schopnosti je možné do istej miery voliť. Profinet, zavádza koncept mien priradenia. Tieto mená sú doplnkom k adrese MAC (hardware adrese). Mená zariadení sa udeľujú prostredníctvom protokolu DCP (Discovery Configuration Protocol). Tento protokol je priamo nad vrstvou Ethernetu. Ako už bolo v tejto práci spomenuté, rozdeľuje sa na Profinet CBA, čo je štandardné prevedenie a Profinet IO, čo je real-timeové prevedenie priemyselného Ethernetu. Práve prevedenie Profinet IO budeme používať v tejto práci na komunikáciu PLC a kontroléru IRC5. [45,48]

9.1 PRÍSLUŠENSTVO

Pri prevedení v laboratóriu budeme používať prídavné moduly k PLC na komunikáciu prostredníctvom Profinetu. Samotné PLC od B&R nemá priamy vstup na Profinet, ako napríklad PLC od Siemensu, za to má však vstup na POWERLINK. Pridaním modulov zaistíme komunikačný kanál s kontrolérom. Modul X20BC1083 bude prostredníctvom Ethernetu POWERLINK spojený priamo s PLC, pričom tento modul bude spojený s modulom X20IF10E1_1, umožňujúci komunikáciu prostredníctvom Profinetu. Tieto moduly budú spolu umiestnené na spoločnej základni X20BB81, ktorá zaistí ich vzájomnú komunikáciu. V tejto základni bude zapojený aj slot pre napájacie napätie. Modul Profinetu bude následne spojený so switchom, ktorý bude spájať kontrolér IRC5 a PLC. Do toho switchu bude privedený aj Ethernet z osobného počítača, prostredníctvom ktorého budeme programovať PLC a kontrolér robota.



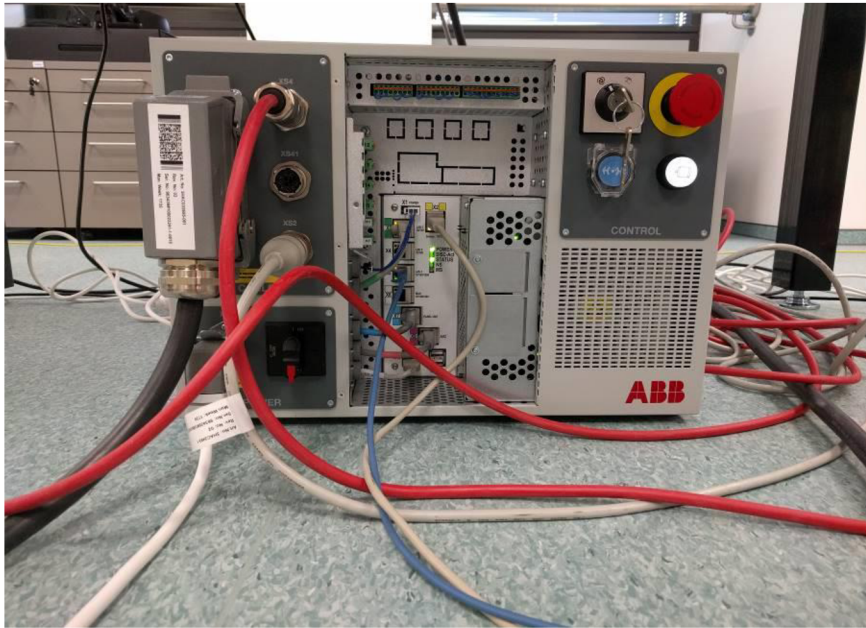
Obr.70 PLC X20CP1584 spolu s modulmi X20IF10E1_1 a X20BC1083



Obr.71 Switch prepájajúci kontrolér IRC5, PLC X20CP1084 a osobný počítač



Obr.72 Laboratórny zdroj EA-PS 3032-10 B napájajúci PLC komunikačné moduly



Obr.73 Kontrolér IRC5 ovládající robota IRB120



Obr.74 Robot IRB120

9.2 NASTAVENIE KOMUNIKÁCIE AUTOMATION STUDIO

Po založení projektu a pridaní príslušného PLC musíme pridať komunikačné moduly a Profinet zariadenie. Aby sme však mohli pridať modul Profinetu a jeho zariadenie, najskôr musíme importovať súbor GSDML. To spravíme tak, že z riadku funkcií zvolíme *Tools* a potom *Manage 3rd-Party Devices*. Otvorí sa nám rozhranie umožňujúce import súborov tretích strán. GSDML súbor vložíme prostredníctvom možnosti *Import DTM Device(s)*. Nakoniec vyberieme súbor *GSDML-V2.32-ABB Robotics-INTERNAL-Robot Device-20170621.xml* a potvrdíme voľbu.

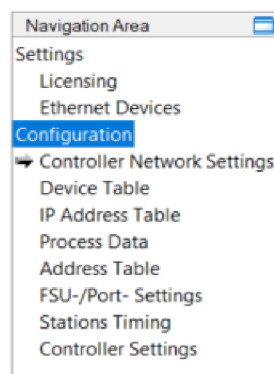
V *Hardware catalogu* nájdeme základňu X20BB81 a pridáme ju v *Physical View*. Následne k nej pridáme rovnakým postupom modul POWERLINKU a Profinetu, ktorý sa nám do katalógu pridal po importovaní GSDML súboru. Po rozkliknutí modulu Profinetu nájdeme port zariadenia a pridáme k nemu zariadenie BASIC V1.2, čo bude predstavovať náš kontrolér IRC5.



Obr.75 System Designer – Výsledná Hardware konfigurácia

9.2.1 NASTAVENIE ADRIES A PRIDANIE PREMENNÝCH

Po tom ako sme vytvorili správnu hardware konfiguráciu, musíme nastaviť adresy a názov zariadenia. To spravíme tak, že pravým tlačidlom myši klikneme na Profinet(DTM) v module X20IF10E1_1 a zvolíme možnosť Device Configuration.



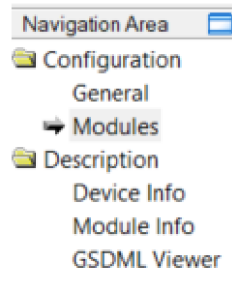
Obr.76 Rozhranie Device Configuration Profinet(DTM)

Tu v možnosti Configuration nastavíme správnu IP adresu, sieťovú masku a bránu. Ďalej v *IP Adress Table* nastavíme IP adresu spájaného zariadenia, v našom prípade teda kontroléru IRC5. Zároveň tu nastavíme sieťovú masku a bránu. IP adresa a brána musia byť v oboch prípadoch nastavovania v tvare špecifickom pre Profinet.

IP Address Table				
Name of station	IP address /	Inherit	Network mask	Gateway address
robotbasicio	192.168.124.3	<input checked="" type="checkbox"/>	255.255.255.0	192.168.124.1

Obr.77 Nastavenie IP Adresy stanice

Informácie o premenných, ktoré budeme používať na komunikáciu uvidíme v záložke *Address Table*. Najskôr ich ale musíme pridať. To prevedieme tak, že pravým tlačidlom myši klikneme na *Internal_BASIC V1.2*, nachádzajúci sa v *Profinet(DTM)* a zvolíme možnosť *Device Configuration*.



Obr. 78 Rozhranie Device Configuration BASIC_V1_2

Po zobrazení rozhrania zvolíme záložku *Modules* a následne možnosť *Add Module*. Pridáme jeden modul digitálnych vstupov a jeden modul digitálnych výstupov. Pokiaľ je to potrebné, môžeme zmeniť veľkosť týchto modulov kliknutím na ich názov v tabuľke.

	Slot	Sub Slot	!	Module
	0			BASIC V1.2 [0]
	1			DI 256 bytes
	2			DO 256 bytes

Obr.79 Založenie premenných v záložke Modules

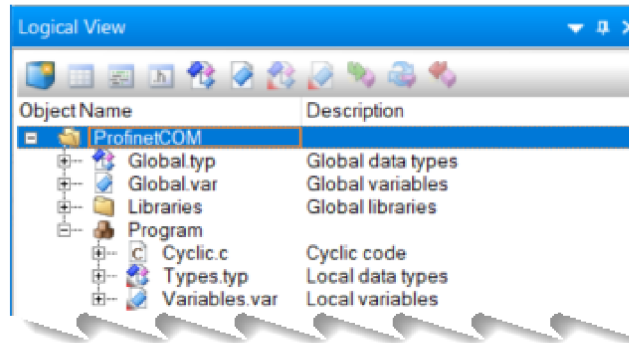
Po potvrdení zmien sa nám tabuľka *Address Table* upraví o nami pridané premenné.

Address Table						
<input checked="" type="checkbox"/> Auto addressing	Display mode: Decimal		CSV Export			
Inputs:						
Name of station	Module	Submodule	Type	Length	Address	
robotbasicio	DI 256 bytes <Slot 1>	DI 256 bytes <Subslot 1>	IB	256		0
Outputs:						
Name of station	Module	Submodule	Type	Length	Address	
robotbasicio	DO 256 bytes <Slot 2>	DO 256 bytes <Subslot 1>	OB	256		0

Obr.80 Adress Table

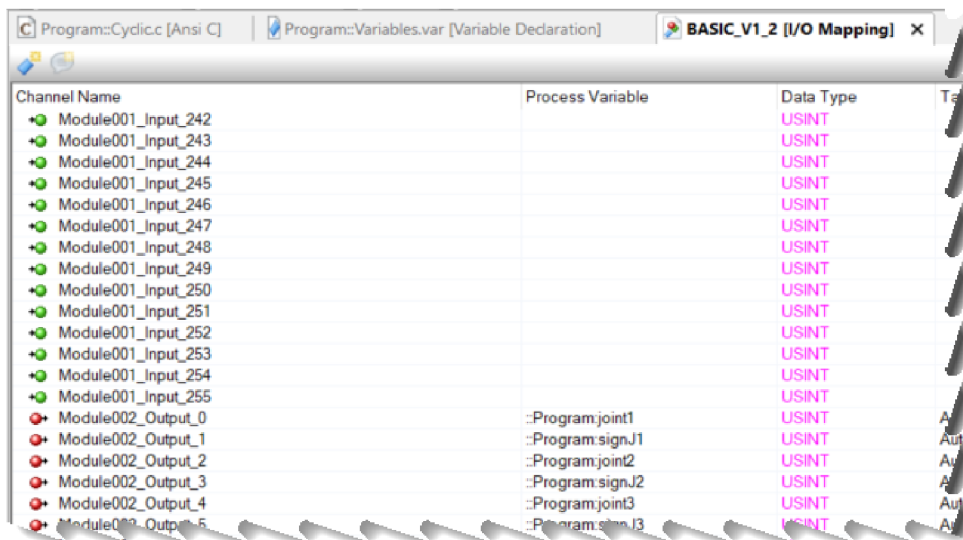
9.2.2 TVORBA KÓDUA I/O MAPPING

Po nastavení konfigurácie Profinet device môžeme pridať premenné a napísať kód, prostredníctvom ktorého budeme riadiť pohyby robota. V tejto práci využijeme ako programovací jazyk jazyk C. V *Logical View* zvolíme priečinok s názvom nášho programu a z *Toolboxe* pridáme objekt *ANSI C Program*. Štruktúru kódu tvoríme, rovnako ako pri príklade so simuláciou s využitím TCP/IP protokolu, podľa SWITCH-CASE metodiky.



Obr.81 Logical View-ANSI C program

Po pridaní premenných do *Variables.var* ich musíme namapovať na vstupy a výstupy Profinet device, ktoré sme založili v predchádzajúcej podkapitole. Do *I/O Mappingu* sa dostaneme tak, že v *Physical View* dvakrát klikneme na zariadenie *BASIC_V1_2* nachádzajúce sa v Profinet(DTM). Následne nastavíme danú premennú ako *Process Variable*. Je dôležité, aby mali premenné, ktoré mapujeme na vstupy a výstupy, rovnaký dátový typ ako vstupno/výstupné premenné. Keďže chceme zaistiť obojstrannú komunikáciu, je nutné nastaviť aj vstupné premenné, prostredníctvom ktorých bude kontrolér IRC5 komunikovať s PLC X20CP1584, aj výstupné premenné, ktoré zaisťujú komunikáciu v opačnom smere.

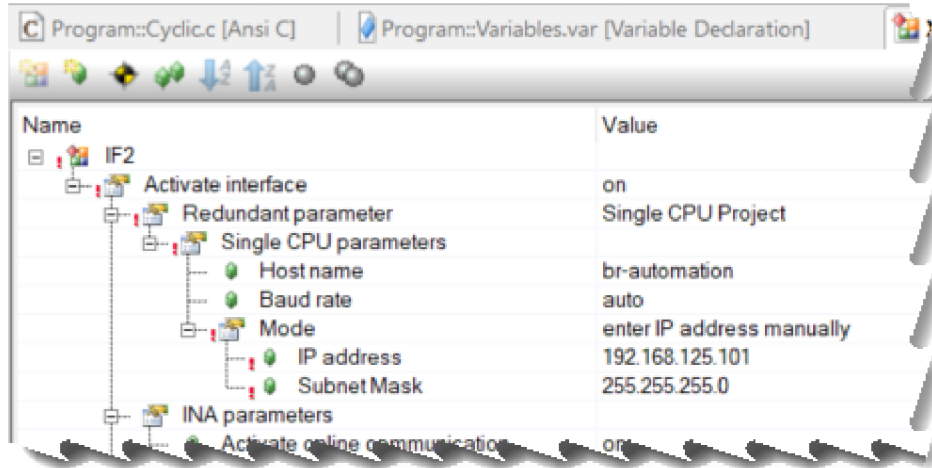


Obr.82 I/O Mapping premenných

9.2.3 NASTAVENIE KOMUNIKÁCIE PC-PLC

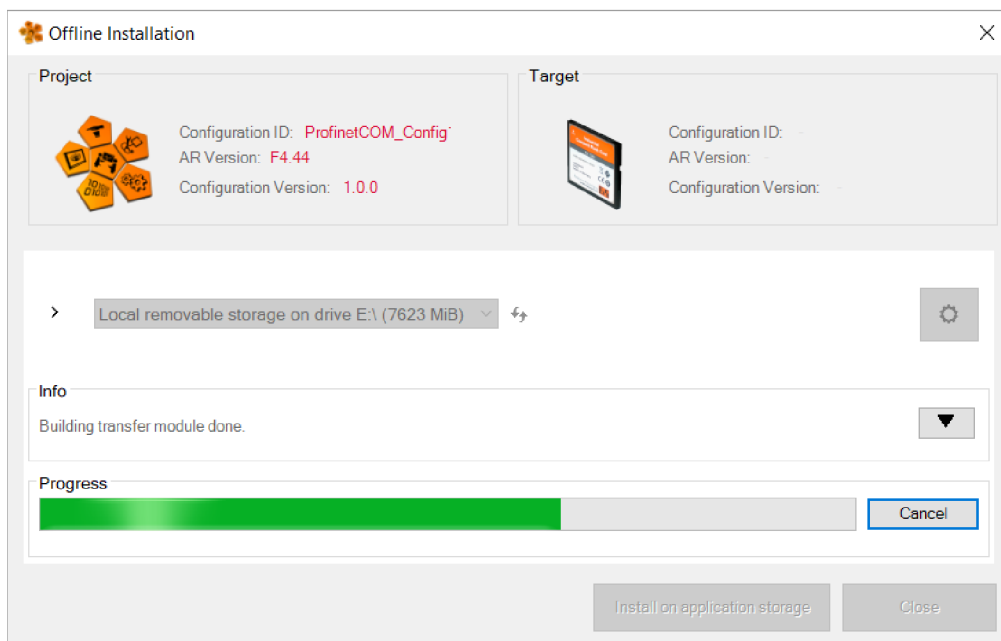
Po tom, ako sme napísali kód a naviazali premenné, je na čase nahráť kód do PLC. Keďže sme náš kód písali na osobnom počítači, musíme ho spojiť s PLC. Prvým krokom, ktorý musíme urobiť je nahráť konfiguráciu z počítača do PLC. To urobíme

prostredníctvom Compact Flash karty, na ktorú ju uložíme. Najskôr nastavíme IP adresu Ethernetového portu PLC. To prevedieme tak, že vo *Physical View* klikneme pravým tlačidlom myši na ETH a zvolíme *Device Configuration*. Tu rozklikneme *IF2->Active interface->Redundant parameter->Single CPU parameter->Mode*. Keď sa dostaneme do *Mode*, klikneme na *Value* a zmeníme ju na *enter IP address manually*. Potom vhodne nastavíme IP adresu a sieťovú masku.



Obr.83 Konfigurácia IP adresy a masky Ethernetového pripojenia

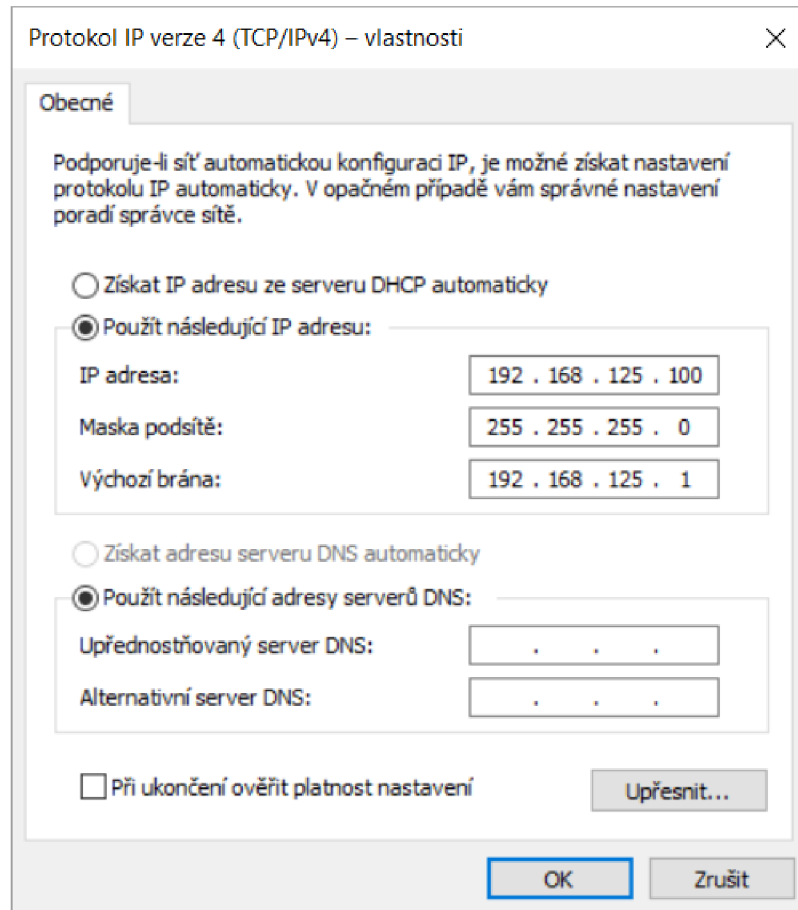
Po týchto nastaveniach môžeme nahráť projekt na Compact Flash pamäťovú kartu. Tú pripojíme ku počítaču pomocou USB adaptéra. Následne zvolíme možnosť *Offline Installation* z *Project-Project Installation*. Po zobrazení rozhrania zvolíme príslušnú Compact Flash a potvrdíme.



Obr.84 Nahrávanie projektu a konfigurácie na Compact Flash pamäť

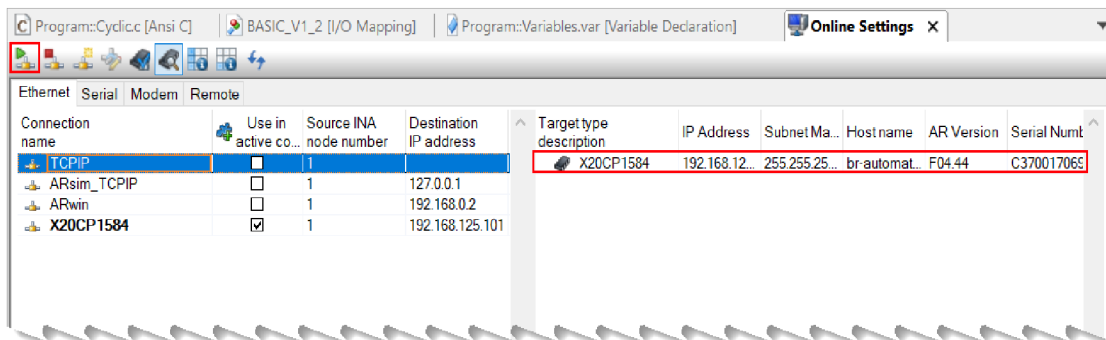
Po dokončení nahrávania vložíme kartu do PLC a počkáme, kým do seba nahrá novú konfiguráciu a projekt. Treba sa uistiť, že sú naša hardware konfigurácia zhodná s reálnym zapojením, v opačnom prípade projekt nebude fungovať.

Ďalším krokom je nastavenie Ethernetového adaptéra počítača. Chceme sa pripojiť priamo na PLC, takže musíme správne nakonfigurovať komunikáciu. Otvoríme nastavenia sietí a internetu, zvolíme Ethernet a ďalej možnosť *Zmeniť možnosti adaptéru*. Tu klikneme pravým tlačidlom myši na Ethernet a zvolíme *Vlastnosti*. Po zobrazení vlastností nájdeme Protokol IP verzia 4(TCP/IPv4) a znovu zvolíme vlastnosti. Nakoniec nastavíme možnosť *Použiť nasledujúcu IP adresu* a nastavíme IP adresu, masku podsiete a pokiaľ je to nutné aj východiskovú bránu.



Obr.85 Sieťové nastavenia počítača

Aby sme sa dokončili pripojenie, zvolíme z panelu funkcií *Online* a ďalej *Settings*. Tu v záložke Ethernet môžeme vidieť stranu počítača a prístupné pripojenia. Zvolíme pripojenie, ktoré sa nám zobrazilo, teda naše PLC a presunieme ho kurzorom na ľavú stranu. Následne zvolíme možnosť *Connect* v ľavom hornom rohu.



Obr.86 Pridanie a aktivovanie pripojenia

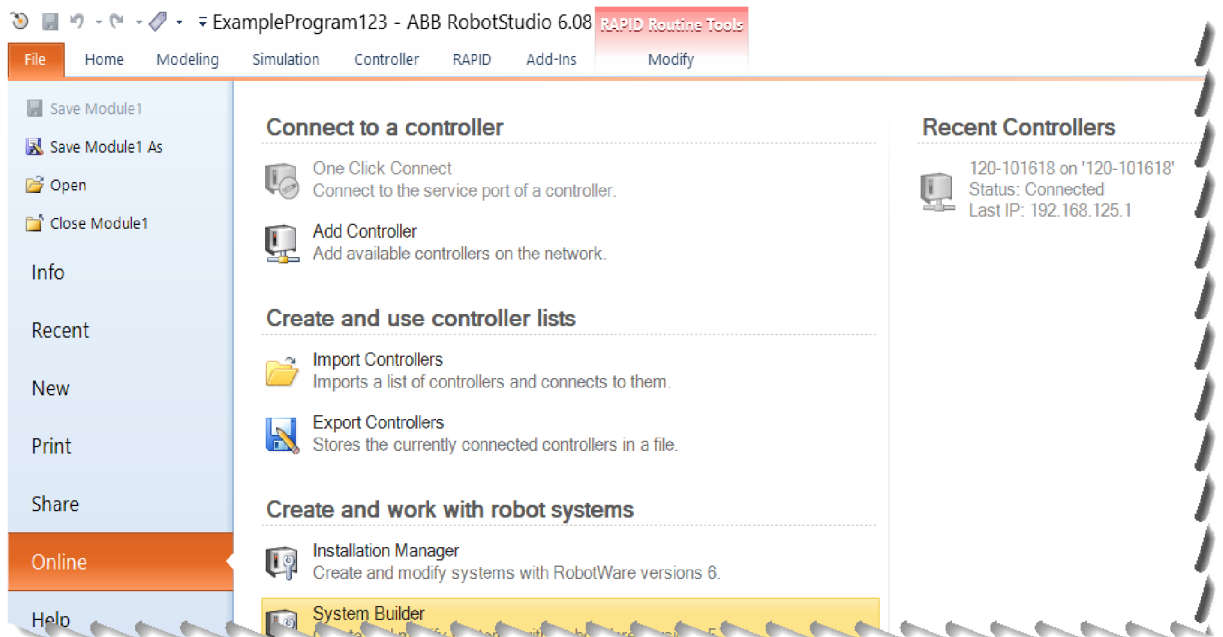
Po prevedení nasledujúcich krokov a zreštartovania PLC a jeho prechodu do módu RUN bude naša komunikácia aktívna.

ANSL: tcpip/RT=1000 /SDT=5 /DAIP=192.168.125.101 /REPO=11159 /ANSL=1 /... X20CP1584 F4.44 RUN

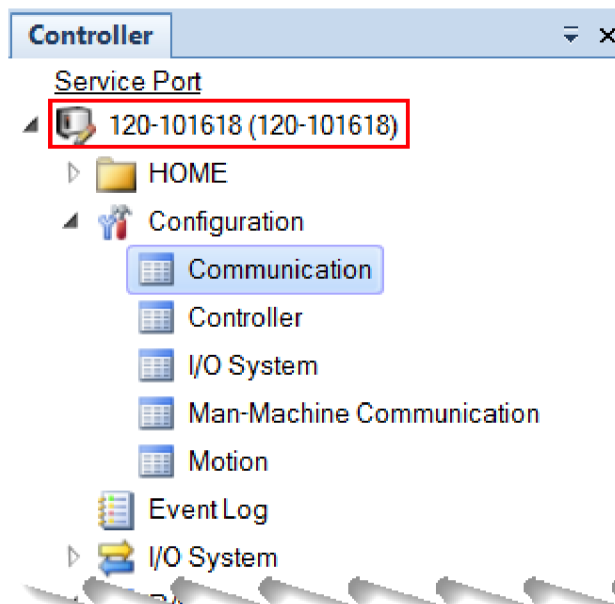
Obr.86 Úspešná komunikácia medzi PC a PLC

9.3 NASTAVENIE KOMUNIKÁCIE ROBOTSTUDIO

Konfiguráciu na strane RobotStudia budeme robiť priamo na kontroléry IRC5. Prvým krokom bude pripojenie kontroléru do počítača. Ethernetový adaptér počítača je už nastavený z predchádzajúcej kapitoly, v prípade že nie, nastavíme ho rovnako. Potom zapneme RobotStudio a zvolíme možnosť *Online* v záložke *File*. Tu sa nám zobrazí prístupný kontrolér, pričom zvolíme pripojenie *One Click Connect*.



Obr.87 Pripájanie ku kontroléru IRC5

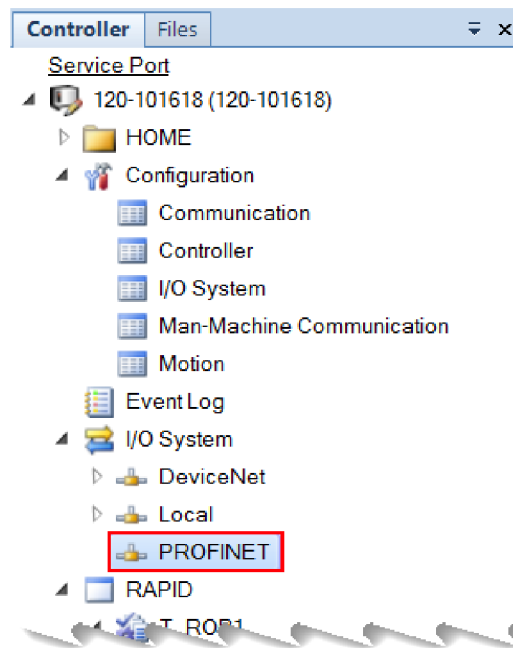


Obr.88 Pripojený kontrolér

Po úspešnom pripojení na kontrolér môžeme nakonfigurovať komunikáciu a vstupno výstupné premenné.

9.3.1 NASTAVENIE ADRIES A NÁZVU STANICE

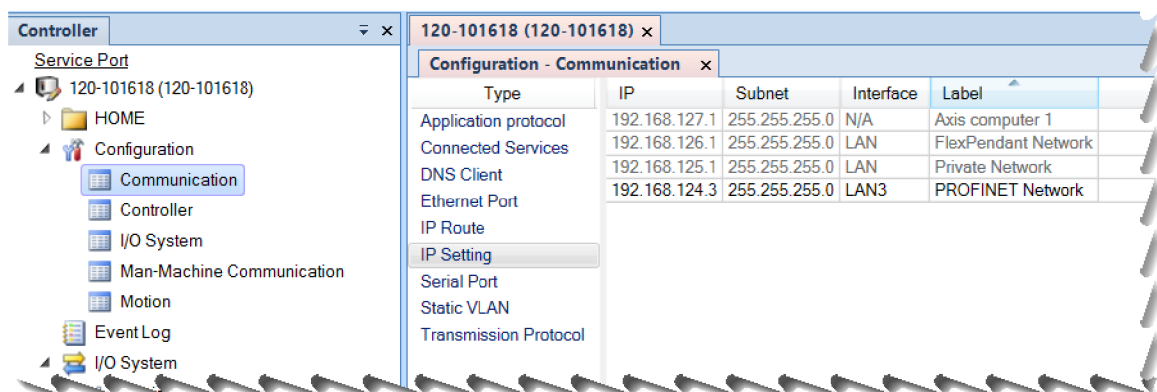
Komunikačné rozhranie Profinet nemusíme kontroléru pridávať, nakoľko si ho vyhľadá a pridá samo. Môžeme si to potvrdiť pri rozkliknutí *I/O System* a nahliadnutí na zariadenia. V simulácii sa rozhranie pridáva tak, že pravým tlačidlom myši klikneme na kontrolér a zvolíme možnosť *Change options*, pričom Profinet môžeme nájsť pod možnosťou *Industrial Networks*.



Obr.89 Pripojenia kontroléru

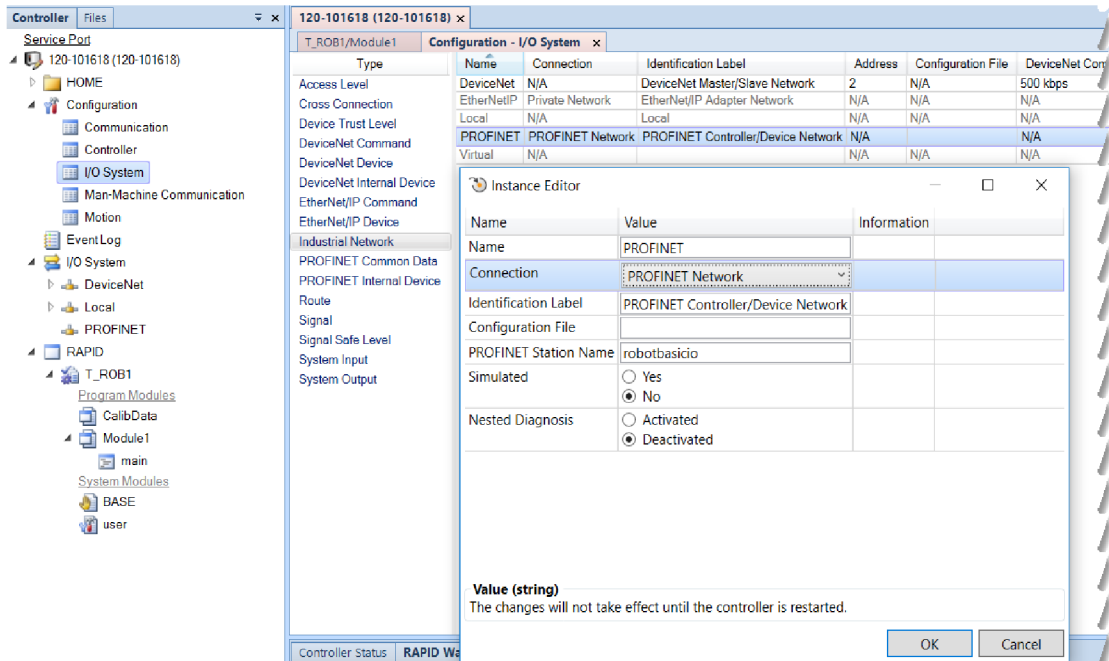
Adresy a názvy nastavíme v záložke *Configuration* a následne *Communication*.

Po otvorení rozhrania zvolíme možnosť *IP Settings* a nájdeme Label Profinet Network. Tu nastavíme IP adresu a sieťovú masku. IP adresa musí byť v tvare špecifickom pre Profinet, pričom musí byť rovnaká, akú sme nastavili zariadeniu v Automation Studiu.



Obr.90 Nastavenie adresy a masky

Po nastavení adresy a masky musíme nastaviť správny názov stanice. Ten musí byť rovnaký, ako názov stanice v Automation Studiu. Názov stanice nastavíme v *Configuration* a *I/O System*, následne zvolíme *Industrial Network*. Meno nastavíme v *PROFINET Station Name*.

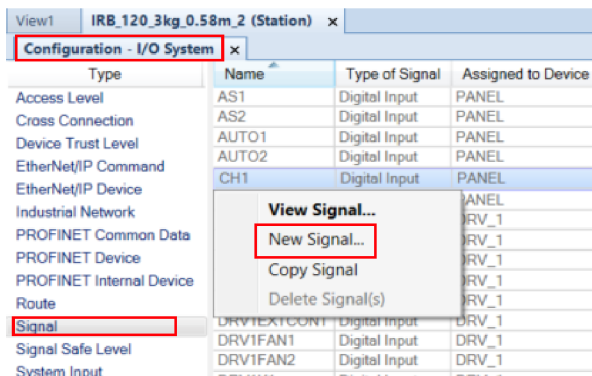


Obr.91 Nastavenie názvu stanice

Po nastavení tejto konfigurácie reštartujeme kontrolér, aby sa potvrdili zmeny. Teraz môžeme kontrolér odpojiť od počítača a pripojiť ho do switchu. Komunikácia je nastavená. Na overenie správnosti jej chodu použijeme v Automation Studiu funkciu *Watch* v *I/O Mapping*, kde sme mappovali premenné. Budeme sledovať *Channel Name ModuleOk*, pričom v *Process Variable* potrebujeme *BOOL* hodnotu *True*.

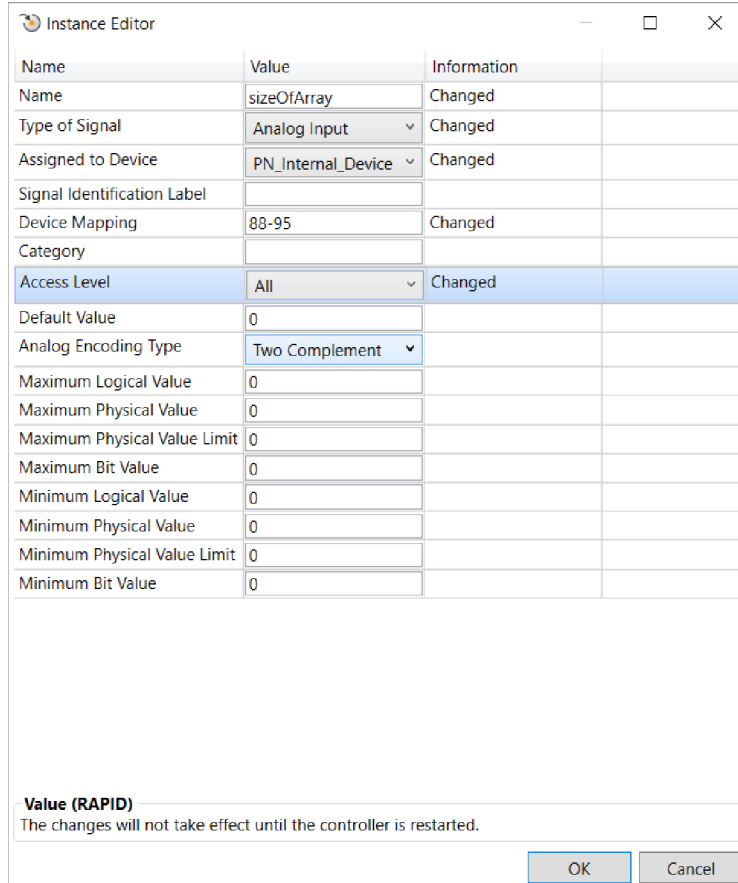
9.3.2 VYTVORENIE A NASTAVENIE PREMENNÝCH

Po úspešnom nastavení komunikácie musíme vyvoriť premenné, pomocou ktorých budeme odosielať/prijímať dáta do/od Automation Studia. Novú premennú, ktoré sú nazývané aj signály, pridáme tak, že v rozhraní kontroléru zvolíme záložku *Configuration* a následne *I/O System*. Tu zvolíme *Signal*. Následne sa nám zobrazí tabuľka signálov popisujúca ich názov, typ, východiskovú hodnotu a tak ďalej. Klikneme pravým tlačidlom myši a zvolíme možnosť *New Signal*.



Obr.92 Tvorba nového signálu

Po prevedení tohoto kroku sa nám zobrazí *Instance Editor*, v ktorom budeme nastavovať parametre signálu. Dôležité sú predovšetkým parametre *Assigned to Device* a *Device Mapping*. Parameter *Assigned to Device* nastavíme k nášmu Profinet zariadeniu, pričom *Device Mapping* nastavujeme v súlade s mapovaním, ktoré sme prevádzali v Automation Studiu. To znamená, že napríklad prvá vstupná premenná v Automation Studiu bude naviazaná na bity 0-7, druhá 8-15 a tak ďalej.



Name	Value	Information
Name	sizeOfArray	Changed
Type of Signal	Analog Input	Changed
Assigned to Device	PN_Internal_Device	Changed
Signal Identification Label		
Device Mapping	88-95	Changed
Category		
Access Level	All	Changed
Default Value	0	
Analog Encoding Type	Two Complement	
Maximum Logical Value	0	
Maximum Physical Value	0	
Maximum Physical Value Limit	0	
Maximum Bit Value	0	
Minimum Logical Value	0	
Minimum Physical Value	0	
Minimum Physical Value Limit	0	
Minimum Bit Value	0	

Value (RAPID)
The changes will not take effect until the controller is restarted.

OK Cancel

Obr.93 Instance Editor-Špecifikácia parametrov signálu

Na priradenie hodnoty vstupného signálu v programe RAPID nejakej premennej voláme jej meno. Pokiaľ chceme priradiť hodnotu výstupnému signálu, používame metódu SetAO.

10 ZÁVER

Táto práca predstavila spoločnosti ABB a B&R, ich históriu a riadiace systémy, pričom sa zameriava hlavne na tie, ktoré boli využité pri realizácii praktického prevedenia, ale aj pri tvorbe simulácie. Každý z firiem je venovaná osobitná kapitola. Následne bol v krátkosti predstavený a popísaný protokol TCP/IP, ktorý bol použitý pri tvorbe a spracovaní komunikácie pri simulácii.

Ďalšia časť prezentovala tri riešenia komunikácie a ovládania robota, ktoré slúžia ako návod k vytvoreniu podobnej aplikácie. Prvé dve riešenia sú iba v prevedení simulácie. Riešenie vo Microsoft Visual Studiu je nepovinné, ide o ukážku využitia ďalšieho riadiaceho systému a jeho spojenia s RobotStudiom pomocou osobného počítača. Samotná aplikácia neriadi pohyby robota, ide iba o komunikáciu klient/server, ktorá po jednoduchých úpravách dokáže poslúžiť rovnakej funkcii ako aplikácia bežiaci na PLC. Riešenie s využitím simulácie PLC a robota bolo natočené a zdokumentované.

Tretí a posledný spôsob je demonštrovaný na robotovi IRB120 v laboratóriu ústavu Automatizácie a informatiky. Tento postup bol rovnako zdokumentovaný a taktiež bolo natočené video úspešnej demonštrácie programu.

11 ZOZNAM POUŽITEJ LITERATÚRY

- [1] B&R. *Filozofie*[online]. 2019, [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/about-us/philosophy/>>
- [2] OHŘÁL, Jan. *B&R Perfection in Automation*[PDF dokument]. 2015, [cit. 2019-04-20]. Dostupné z: <<http://www.bicbrno.cz/getattachment/e65460cf-714a-4e94-b270-49e9d0d1348a/BR-automatizace-Ohral.aspx> >
- [3] B&R Automation Brno[obrázok].google.com, 2017,[cit. 2019-04-20]. Dostupné z: <<https://www.google.com/maps/@49.2127921,16.5689214,3a,39y,143.51h,86.06t/data=!3m6!1e1!3m4!1sWgyNlv7LG5EjuhjLPDH7Mg!2e0!7i13312!8i6656>>
- [4] B&R. *control-systems*[online]. 2019, [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/control-systems/>>
- [5] B&R *software* [online]. 2019,[cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/software/>>
- [6] B&R *openSAFETY* [online]. 2019,[cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/technologies/opensafety/>>
- [7] B&R *Automation Studio 4*[PDF document]. 2014,[cit. 2019-04-20]. Dostupné z: <https://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000000289549/Automation%20Studio%204%20MM-E01325.041.pdf>
- [8] B&R *Io-systems*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/products/io-systems/>>
- [9] B&R *x20-system*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/products/io-systems/x67-system/>>
- [10] B&R *coated-x20-system*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/products/io-systems/coated-x20-systems/>>
- [11] B&R *x67-system*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/en-gb/products/io-systems/x67-system/>>
- [12] B&R *industrial-pcs*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/industrial-pcs/>>
- [13] B&R *automation-pc-2100*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/industrial-pcs/automation-pc-2100/>>
- [14] B&R *operating-systems*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/industrial-pcs/additional-information/operating-systems/>>
- [15] B&R *automation-pc-910*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/industrial-pcs/automation-pc-910/>>
- [16] B&R *smart-display-link-4*[online]. 2019, [2019-04-20]. [cit. 2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/industrial-pcs/smart-display-link-4/>>
- [17] B&R *powerlink-ethernet-tcpip*[online]. 2019, [2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/networks-and-fieldbus-modules/powerlink-ethernet-tcpip/>>

- [18] B&R *networks-and-fieldbus-modules*[online]. 2019, [2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/networks-and-fieldbus-modules/>>
- [19] B&R *profinet*[online]. 2019, [2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/networks-and-fieldbus-modules/profinet/>>
- [20] B&R *can-bus*[online]. 2019, [2019-04-20]. Dostupné z: <<https://www.br-automation.com/cs/produkty/networks-and-fieldbus-modules/can-bus/>>
- [21] POWERLINK *ethernet-powerlink*[online]. 2019, [2019-04-20]. Dostupné z: <<https://www.ethernet-powerlink.org/>>
- [22] ABB *zakladni-udaje*[online]. 2019, [2019-04-20]. Dostupné z: <<https://new.abb.com/cz/o-nas/zakladni-udaje>>
- [23] ABB *divize*[online]. 2019, [2019-04-20]. Dostupné z: <<https://new.abb.com/cz/o-nas/divize>>
- [24] ABB *HQ-Zürich* [obrázok]. glassdoor.co.uk, 2019, [2019-04-20]. Dostupné z: <<https://www.glassdoor.co.uk/Photos/ABB-Office-Photos-IMG269487.htm>>
- [25] ABB *Robot-IRB120* [PDF dokument]. 2018, [2019-04-20]. Dostupné z: <<https://search-ext.abb.com/library/Download.aspx?DocumentID=3HAC035960-001&LanguageCode=en&DocumentPartId=&Action=Launch>>
- [26] ABB *Controller-IRC5* [PDF dokument]. 2011, [2019-04-20]. Dostupné z: <https://library.e.abb.com/public/c13e1c5490c61230c125796000515137/IRC5%20datasheet%20PR10258%20EN_R13.pdf>
- [27] ABB *system-irc5-technicka-data* [online]. 2019, [2019-04-20]. Dostupné z: <<https://new.abb.com/products/robotics/cs/ridici-systemy/irc5/irc5-technicka-data>>
- [28] ABB *system-irc5-data-sheet*[PDF dokument]. 2018, [2019-04-20]. Dostupné z: <<https://search-ext.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch>>
- [29] ABB *Controller-Omnicores*[online]. 2019, [2019-04-20]. Dostupné z: <<https://new.abb.com/products/robotics/controllers/omnicore>>
- [30] ABB *Controller-Omnicores-data-sheet* [PDF dokument]. 2018, [2019-04-20]. Dostupné z:<<http://search.abb.com/library/Download.aspx?DocumentID=9AKK107046A3855&LanguageCode=en&DocumentPartId=&Action=Launch>>
- [31] ABB *Compact-Robot-Controller*[obrázok]. directindustry.com, 2019, [2019-04-20]. Dostupné z: <<http://www.directindustry.com/prod/abb-robotics/product-30265-169114.html>>
- [32] ROBOTIKA *História* [online]. 2019, [2019-04-20]. Dostupné z: <<https://ifr.org/robot-history>>
- [33] ROBOTIKA *Versatran* [obrázok]. researchgate.net, 2019, [2019-04-20]. Dostupné z: <https://www.researchgate.net/figure/ersatran-from-14_fig1_329590080>
- [34] KOLÍBAL, Zděnek. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně-nakladatelství VUTIUM, 2016 ISBN 978-80-214-4828-5

- [35] SKAŘUPA, Jiří. *Průmyslové roboty a manipulátory* [online]. Ostrava: Vysoká škola báňská-Technická univerzita, [2008] [cit.2019-04-20]. ISBN 978-80-248-1522-0.
- [36] VACHÁLEK, Ján, TAKÁCS, Gergély. *Robotika*. V Bratislave: Slovenská technická univerzita v Bratislave, 2014. Edícia vysokoškolských učebníc (Slovenská technická univerzita). ISBN 978-80-227-4163-7
- [37] ROBOTIKA *pneumatické-pohony* [obrázok]. parker.cz, 2018, [2019-04-20]. Dostupné z: <<http://www.parker.cz/nova-rada-pneumatickych-valcu-plf-podle-iso-15552/>>
- [38] ROBOTIKA *hydraulické-pohony* [obrázok]. docplayer.cz, 2009, [2019-04-20]. Dostupné z: <<https://docplayer.cz/1998963-Radialni-pistovy-hydromotor-a-hydrogenerator.html>>
- [39] ROBOTIKA *teach-pendant* [obrázok]. robot-store.co.uk, 2019, [2019-04-20]. Dostupné z: <<https://robot-store.co.uk/product/teach-pendant-krc4-id403>>
- [40] ROBOTIKA *RAPID* [obrázok]. forums.robotstudio.com, 2016, [2019-04-20]. Dostupné z: <<https://forums.robotstudio.com/discussion/9784/rapid-removing-a-directory-with-contents>>
- [41] TCP/IP *sieťový-model* [obrázok]. notes.shichao.io, 2019, [2019-04-20]. Dostupné z: <<https://notes.shichao.io/icnd1/part1/#tcpip-networking-model>>
- [42] B&R Automation. Automation Studio [počítačový program]. Ver. 4.3., 2018 [citováno 2019-04-25]. Dostupné z {BrAutomation}\AS43\Bin-en\BR.Help.Explorer.exe. Uživatelská nápověda. Vyžaduje Windows Vista / 7 / 8 / 10
- [43] Profinet *Priemyselná komunikácia* [online]. 2017, [2019-04-20]. Dostupné z: <<https://www.foxon.cz/blog/prakticka-teorie/206-a-tohle-jste-o-profinetu-vedeli-2-dil-prumyslove-komunikace>>
- [44] Profinet *Profinet* [online]. 2017, [2019-04-20]. Dostupné z: <<https://www.profibus-profinet.cz/profinet>>
- [45] Profinet *Princípy komunikácie a diagnostika sietí Profinet* [PDF dokument]. 2013, [2019-04-20]. Dostupné z: <http://www.automa.cz/Aton/FileRepository/pdf_articles/10377.pdf>
- [46] Profinet *Spríevodca návrhom siete* [PDF dokument]. 2014, [2019-04-20]. Dostupné z: <https://web.foxon.cz/downloads/profibus-profinet/profinet/PROFINET_Design_8062_V114_Dec14_cs_FOXON.pdf>
- [47] Profinet *ABB Application manual Profinet Controller/Device with IO Configurator* [PDF dokument]. 2018, [2019-04-20]. Dostupné z: <<https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC065546%20AM%20PROFINET%20Controller%20Device%20with%20IO%20Configurator%20RW%206-en.pdf>>
- [48] Profinet *Průmyslový Ethernet VII: Ethernet Powerlink, Profinet* [PDF dokument]. 2018, [2019-04-20]. Dostupné z: <<http://www.odbornecasopisy.cz/res/pdf/37288.pdf>>
- [49] ŠTEFANAČEK, Peter. *Konstrukcia priemyselných robotov a manipulátorov*. 1. vyd. Bratislava: Alfa 1985, 265s.

- [50] TCP Client *Creating TCP/IP Client in C++* [online]. 2017, [2019-04-20]. Dostupné z: <https://www.youtube.com/watch?v=0Zr_0Jy8mWE>
- [51] Socket programming *Socket programming Tutorial In C For Beginners* [online]. 2017, [2019-04-20]. Dostupné z: <<https://www.youtube.com/watch?v=LtXEMwSG5-8>>
- [52] Matlab and RobotStudio *Example socket communication between Matlab and Robotstudio* [online]. 2017, [2019-04-20]. Dostupné z: <<https://www.youtube.com/watch?v=JGJt17c69Bk>>

12 ZOZNAM PRÍLOH

Obsah elektronickej prílohy:

- Projekt v Automation Studiu a RobotStudiu obsahujúci simuláciu komunikácie PLC s kontrolérom a ovládania robota s využitím protokolu TPC/IP
- Projekt v Microsoft Visual Studiu a RobotStudiu obsahujúci simuláciu komunikácie medzi kontrolérom robota a osobným počítačom s využitím protokolu TCP/IP
- Projekt v Automation Studiu a RobotStudiu obsahujúci programy zabezpečujúce komunikáciu s kontrolérom a ovládanie robota s využitím priemyselnej zbernice Profinet IO
- Video simulácie komunikácie prostredníctvom protokolu TCP/IP a ovládania robota v RobotStudiu prostredníctvom Automation Studia
- Video simulácie TCP/IP komunikácie medzi Microsoft Visual Studiom a RobotStudiom
- Video praktickej demonštrácie komunikácie s kontrolérom a ovládania robota IRB120 cez PLC X20CP1584 v laboratóriu Ústavu Automatizace a Informatiky