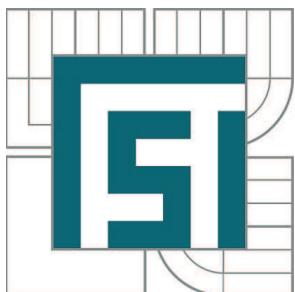


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

INTELIGENTNÍ DVOU-RAMENNÝ ŽONGLOVACÍ STROJ

INTELLIGENT DOUBLE ARM JUGGLING MACHINE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. DANIEL MERTLÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MARADA, Ph.D.

BRNO 2012

Bez pomoci a podpory spousty lidí bych studium na vysoké škole a především zpracování diplomové práce jen ztěžka zvládl, a proto je na místě jim poděkovat. Mé největší poděkování patří zejména Ing. Tomáši Maradovi, Ph.D., hlavně za trpělivost, rychlé reakce a cenné připomínky při vedení mé diplomové práce. Také bych rád poděkoval doc. Ing. Zdeňku Němcovi, CSc., za jeho odborné rady v oblastech regulace a návrhu zařízení a Ing. Radomilu Matouškovi, Ph.D., za jeho pomoc při konstrukci zařízení. Mé dík patří i mému kolegovi Bc. Janu Janečkovi, bez jehož spolupráce by zařízení nemohlo správně fungovat. Dále děkuji Ústavu automatizace a informatiky za možnost využití laboratoří a softwarového vybavení a v neposlední řadě mým nejbližším.

Prohlašuji, že jsem diplomovou práci vyřešil samostatně s využitím uvedených pramenů a literatury.

V Brně dne 18. května 2012

.....

Abstrakt

Diplomová práce se zabývá návrhem a realizací zařízení nazvaného Inteligentní dvou-ramenný žonglovací stroj. Úlohou projektu je demonstrace řízení rotačního pohybu ramene tak, aby v rámci koncepce stroje docházelo k přehazování míčku na opačnou stranu ramene. Projekt je rozdělen do dvou diplomových prací, kdy tato řeší již zmiňovaný návrh stroje a zadávání úkolů druhé kooperující úrovni řízení prostřednictvím vizualizace.

Klíčová slova: dotykový displej, komunikace, PLC, programovatelný automat, vizualizace

Abstract

This thesis describes the design and implementation of device called Intelligent two-arm juggling machine. The role of the demonstration project is to control rotational movement of the arm so that the concept of the machine there to swap the ball to the opposite side shoulder. The project is divided into two theses, which this proposal addresses the above mentioned tasks, the machine and place the second level ofcooperative governance through visualization.

Keywords: touch panel, communication, PLC, programmable logic controller, visualization

MERTLÍK, D. *Inteligentní dvou-ramenný žonglovací stroj*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. XY s. Vedoucí diplomové práce Ing. Tomáš Marada, Ph.D.

Obsah

1	Úvod	15
1.1	Úvod do problematiky	15
1.2	Cíl práce a metodika	16
2	Návrh zařízení	17
2.1	Fyzikální princip	17
2.2	Volba komponentů	18
2.3	Návrh konstrukce	20
3	Použité komponenty	25
3.1	Dotykový panel TP 12T	25
3.2	Programovatelný automat ILC 150 ETH	26
3.3	Ethernet switch	27
3.4	IL Bluetooth Access Point – 2884758	28
3.5	IL TEMP	29
3.6	Senzor PT 100	30
4	Použitá komunikace	31
4.1	Podrobný popis komunikace	32
4.2	Nutná konfigurace	32
5	Program VISU+	35
5.1	Programování ve VISU+	35
5.2	Založení projektu	35
5.3	Pracovní prostředí	36
5.4	Basic shapes (dále „základní útvary“)	37
5.4.1	Skrytí/zobrazení objektu	38
5.4.2	Obrázek jako pozadí objektu	38
5.4.3	Dynamický přesun objektu	38
5.5	Button (dále „tlačítko“)	38
5.5.1	Nastavení hodnoty proměnné	39
5.5.2	Přepnutí na jinou obrazovku	39
5.5.3	Otevření obrazovky v novém okně	39
5.5.4	Deaktivace tlačítka	39
5.5.5	EditBox – Display (dále „editační okno“)	39
5.6	Hot Region	39
5.6.1	Zapsání hodnoty proměnné pomocí numerické klávesnice	40
5.7	Trend	40
5.7.1	Přiřazení proměnné trendu	40
5.8	Připojení dynamické OPC proměnné	40
5.9	Nahrání projektu na dotykový panel	41

6 Realizace uživatelského rozhraní	43
6.1 Obrazovka MENU	43
6.2 Obrazovka SERVICE	44
6.3 Obrazovka INTRODUCTION	46
6.4 Obrazovka ABOUT	47
6.5 Obrazovka JUGGLING	48
6.6 Obrazovka hJUGGLING	50
6.7 Obrazovka iJUGGLING	52
7 Realizace řídícího programu	55
7.1 Struktura programu	56
7.2 Předdefinované funkční bloky pro komunikaci	57
7.2.1 IP_CONNECT	57
7.2.2 IP_USEND	59
7.2.3 IP_URCV	60
7.3 Vlastní funkční bloky	61
7.3.1 FB_Ethernet	61
7.3.2 FB_Height	64
7.3.3 FB_Temp_RTD	66
8 Závěr	69
9 Použité zdroje	71
10 Přílohy	73

Seznam obrázků

1	Fyzikální princip činnosti stroje	17
2	Schéma pro výpočet kroutícího momentu	18
3	Konstrukce stroje	20
4	Potisk trojúhelníku	21
5	Upínka ramene	21
6	Model krytu	21
7	Realizované zařízení	22
8	Blokové schéma použitých zařízení	23
9	Dotykový panel TP 12T	25
10	Princip funkce odporového displeje	26
11	Programovatelný automat ILC 150 ETH	26
12	Ethernet switch	27
13	IL Bluetooth Access Point – 2884758	28
14	IL TEMP	29
15	Senzor PT 100	30
16	Komunikační schéma zařízení	31
17	Pracovní prostředí programu VISU+	36
18	Použité ovládací panely programu VISU+	37
19	Všechny objekty panelu Toolbox	38
20	Připojení OPC proměnné	40
21	Okno nastavení pro nahrání projektu	41
22	Obrazovka – MENU	43
23	Servisní obrazovka – SERVICE	44
24	Upozornění při stisku tlačítka EMERGENCY STOP	45
25	Informativní obrazovka – INTRODUCTION	46
26	Informativní obrazovka – ABOUT	47
27	Graf vývoje teplot	48
28	Ovládací obrazovka – JUGGLING	49
29	Ovládací obrazovka – Direction of juggling	50
30	Ovládací obrazovka – hJUGGLING	51
31	Ovládací obrazovka – iJUGGLING	52
32	Část řídícího slova <i>mode</i>	55
33	Struktura programu	56
34	Schéma vrhu míčku z nenulové výšky	65
35	Vstupní/výstupní slova pro obsluhu měřící karty	66
36	Výstupní sparametr OUT1	67
37	Výstupní parametr OUT2	67
38	Členové týmu při účasti ve finálovém kole v Německu	70

Seznam tabulek

1	Rozdělení řídícího slova do bloků	56
2	IP_CONNECT	58
3	IP_USEND	59
4	IP_URCV	60

1 Úvod

1.1 Úvod do problematiky

Německá společnost s více než osmdesátiletou tradicí Phoenix Contact – přední světový výrobce elektrického připojení a průmyslové automatizační techniky, vyhlásila pro rok 2012 (po čtyřleté pauze každoročního pořádání) soutěž xplore New Automation Award 2012.

Soutěže se mohou účastnit nejen jedinci, ale také týmy neomezeného počtu členů, z celého světa. Úroveň vzdělanosti pro zapojení do soutěže není rozhodující a účastnit se jí mohou studenti středních a vysokých škol, lektori či profesori. Pravidla soutěže nedefinují žádná omezení, co se obsahu týče, avšak pro přijetí do soutěže je žádoucí dodržet následující aspekty projektu: systémová přístupnost, strukturované programování a moderní způsoby přenosu dat. Motivací účastníků pro zapojení se do soutěže je nejen možnost pracovat na zajímavých projektech a získání tak cených zkušeností v oboru, ale také finanční podpora 3000 Euro, které společnost Phoenix Contact poskytuje na podporu realizace vybraných projektů (ve formě nákupu jejich komponentů).

Podmínky a účast v soutěži xplore New automation Award 2012 byly natolik zajímavé, že se využilo příležitosti a byl sestaven pětičlenný tým složený ze dvou studentů Fakulty strojního inženýrství VUT a tří pracovníků ústavu Aplikované informatiky a řízení, též fakulty. Jmenovitě hlavní řešitelé Bc. Jan Janeček a Bc. Daniel Mertlík a ostatní členové týmu doc. Ing. Zdeněk Němec, CSc., Ing. Tomáš Marada, Ph.D., a Ing. Radomil Matoušek, Ph.D. Tým byl do soutěže přihlášen s projektem Intelligent double arm juggling machine, jehož podstatou je návrh a následná realizace stroje, který prezentuje řízení časově náročné úlohy zábavnou formou. Vzhledem k zaměření projektu na řízení zábavnou formou spadá projekt do soutěžní kategorie Recreation.

Podstatou projektu je sestavení žonglovacího stroje, který řeší problematiku vyzazování a chytání míčku pomocí ramene. Rameno se pohybuje rotačním pohybem, který je vytvářen motorem a je roztáčeno a brzděno takovým způsobem, aby zajistilo vyhození a chycení míčku. Aby mohlo zařízení správně fungovat je nutné vybrat vhodné komponenty pro jeho sestavení. Vzhledem k náročnosti řešené problematiky je projekt rozdělen na dvě části, a to na řízení pohybů ramene poháněného motorem a na řízení ovládání stroje a komunikace. Zmíněné dělení soutěžního projektu odpovídá rozdělení úlohy mezi hlavní řešitele, kteří zadané problematice věnují své diplomové práce.

Obsahem diplomové práce je návrh konstrukce stroje, zadávání úkolů druhé kooperující úrovni řízení a vizualizace.

1.2 Cíl práce a metodika

Aby mohla diplomová práce vůbec vzniknout, je nutné, aby se autor řádně seznámil s veškerým programovým a hardwarovým vybavením, potřebným pro realizaci žonglovacího stroje, a zlepšil své dovednosti v širokém okruhu činností (konstrukce, grafický návrh, programování apod.).

Hlavním cílem práce je v součinnosti s druhým řešitelem úkolu navrhnut celkovou koncepci stroje a algoritmy činnosti. Mezi další neméně důležité cíle patří zpracování programového vybavení, tedy vytvoření programů pro obsluhu zařízení, a ověření funkčnosti.

Práce je dělena na dvě hlavní části. První obsahuje návrh konstrukce stroje a výběr vhodných komponentů. Další část se zabývá zajištěním komunikace, složené z řešení programu vizualizace a programu v PLC.

Ke splnění těchto cílů bude v diplomové práci využito fyzikálních principů rotace a šíkmého vrhu a programového vybavení, konkrétně pro modelování program Autodesk Inventor, pro zpracování programového vybavení programy VISU+ a PCWORX, a pro podporu vizualizace bude použit program CorelDraw.

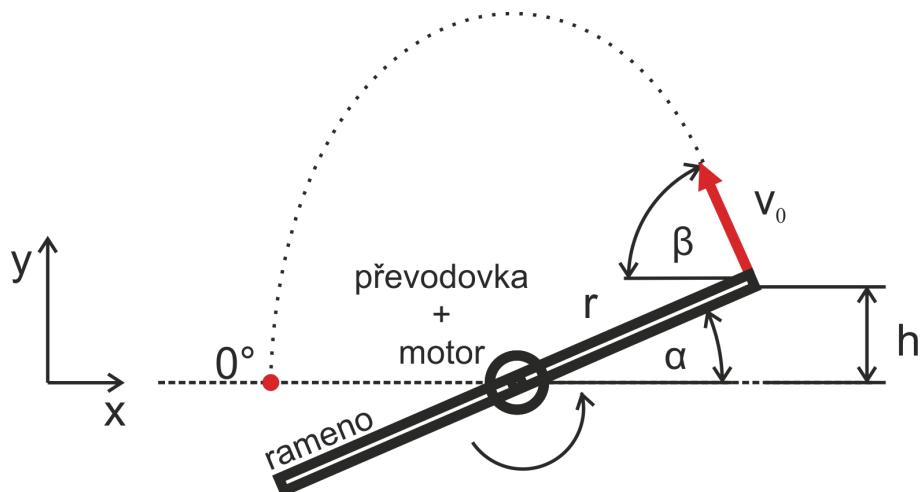
2 Návrh zařízení

Základem celého projektu je řešení rotačního pohybu ramene. V rámci zadané úlohy je třeba navrhnut celkovou koncepcí zařízení. Poté, v závislosti na zvolené koncepci, vybrat vhodné komponenty a přejít k realizaci vybraného návrhu.

2.1 Fyzikální princip

Jak již bylo řečeno v úvodu, úlohou projektu je demonstrace řízení rotačního pohybu ramene tak, aby v rámci koncepce stroje docházelo k přehazovaní míčku na opačnou stranu ramene.

Pro úspěšné vrhnutí a následné chycení míčku je potřeba úlohu, z hlediska typu pohybu, rozdělit na dvě části. Na rotaci a šikmý vrh. Jak je znázorněno na Obrázku 1, z rychlosti otáčení ramene a jeho poloměru lze vyjádřit počáteční rychlosť pro šikmý vrh míčku. Žádoucí pro řešenou úlohu je dopad vrženého míčku na konkrétní místo dané geometrií koncepce stroje a proto je potřeba kromě rychlosti zvolit současně i úhel odhazu.



kde:

r = poloměr ramene;

h = výška, ze které je vrhán;

α = úhel pootočení ramene při vypuštění;

β = elevační úhel pro výpočet šikmého vrhu;

Obrázek 1: Fyzikální princip činnosti stroje

Zdroj: autor práce, 2012

Je zřejmé, že pro vrh musí být dosaženo určitých otáček motoru, převedených přes vhodnou převodovku na rychlosť otáčení ramene. Dosažení těchto otáček musí být provedeno rychle a přesně. Stejně tak musí být přesný úhel, ve kterém dojde

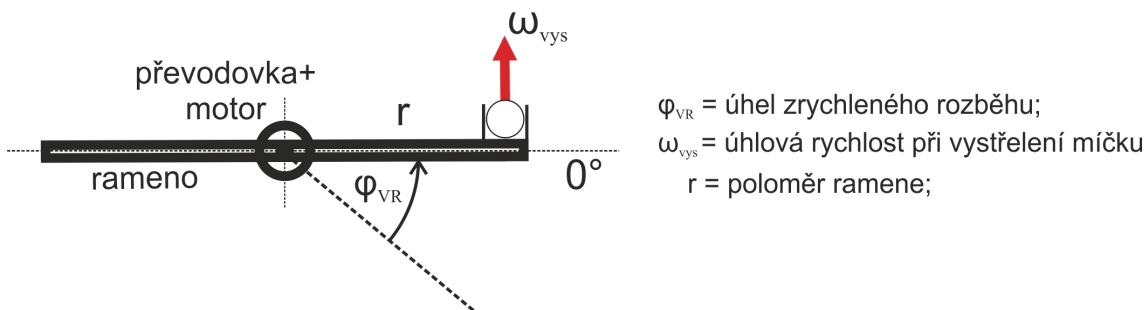
k odhozu, k němuž dojde až po dostatečném zpomalení (zastavení) otáčení ramene. Přesné řízení motoru je řešeno v druhé diplomové práci, která se podílí na splnění zadaného úkolu, regulací rychlosti a polohy.

2.2 Volba komponentů

Základem je výběr stejnosměrného motoru (dále také „motor“) a převodovky. Je důležité, aby byl motor s převodovkou schopen dosáhnout požadované rychlosti za určitý čas (resp. úhel pootočení ramenem).

Nejvíce se na výběru projevuje poloměr ramene. V rámci projektu je žádoucí držet se menších rozměrů vzhledem k dodržení přijatelné velikosti celého stroje. Delší rameno v praxi způsobí nižší požadavky na rychlosť otáček, ale zároveň je tak potřeba vrhat na delší vzdálenost. Vrhání na delší vzdálenosti potom zvětšuje pravděpodobnost chyby. Délka ramene je tedy volena v závislosti na předešlých úvahách.

Pro výběr motoru je také nutné určit kroutící moment M_k (Obrázek 2), který bude nutný pro dosažení potřebného zrychlení. Dle níže uvedených vztahů byla „pro vystřelení míčku kolmo vzhůru do výšky 1 m, vypočtena hodnota kroutícího momentu na motoru $M_{KM} = 0,07[Nm]$. Podle zjištěné hodnoty celkového kroutícího momentu na motoru je odvozen potřebný výkon motoru. Převodovku motoru volíme na základě poměru úhlové rychlosti při vystřelení míčku ku max. převoditelným otáčkám, který je 23,53.



Obrázek 2: Schéma pro výpočet kroutícího momentu

Zdroj: autor práce, 2012

Vztahy pro výpočet potřebného M_k :

$$M_k = J \cdot \varepsilon \quad (1)$$

Jedná se o rovnoměrně zrychlený pohyb:

$$\varepsilon = \frac{d\omega}{dt} \quad (2)$$

Úhlová rychlosť pri vystrelení míčku:

$$\omega_{VYS} = \varepsilon \cdot \Delta t \quad (3)$$

Úhel urychleného rozběhu:

$$\varphi_{VR} = \omega_{stredni} \cdot \Delta t = \frac{1}{2} \cdot \omega_{VYS} \cdot \Delta t = \frac{1}{2} \cdot \frac{\omega_{VYS}^2}{\varepsilon} \quad (4)$$

M_k dosazeno do φ_{VR} :

$$\varphi_{VR} = \frac{1}{2} \cdot \frac{\omega_{VYS}^2}{M_k} \cdot J \quad (5)$$

$$M_k = \frac{\omega_{VYS}^2 \cdot J}{2 \cdot \varphi_{VR}} \quad (6)$$

Moment setrvačnosti:

$$J_{micek} = r^2 \cdot m_{micek} \quad (7)$$

$$J_{miska} = r^2 \cdot m_{micek} \quad (8)$$

$$J_{ramene} = m \cdot \frac{l^2}{3} \quad (9)$$

$$J_{motor} = i^2 \cdot J_{motor} \quad (10)$$

$$J = J_{micek} + 2 \cdot J_{miska} + 2 \cdot J_{ramene} + J_{motor} \quad (11)$$

Potřebný kroutící moment na motoru:

$$M_{KM} = \frac{M_K}{i} \quad (12)$$

Podle potřebného výkonu a požadavků na převodovku byl zvolen DC motor PD4266 od firmy Transmotec. Motor obsahuje planetovou převodovku s převodovým poměrem 24:1.

Řízení motoru probíhá pomocí regulace. Pro regulaci rotačního pohybu motoru je vybrán, z katalogu Phoenix Contact, servozesilovač IB IL EC AR 48/10A. Obsahuje tři hardwarově implementované regulátory (proudový, rychlostní a polohový), které výhodně pro potřeby zařízení pracují s velmi malými periodami.

Pro zpětnou vazbu z motoru do servozesilovače je nutné připojit inkrementální snímač. Snímač je potřeba volit takový, aby bylo dosaženo co možná nejlepšího rozlišení pulzů snímače na úhel pootočení motoru. Docíleno toho je výběrem inkrementálního snímače HEDS-5540, který v režimu dekódování měří celkem 2000 pulzů na otáčku. Dopolně komponenty je potřeba řídit pomocí PLC. Pro řízení motoru v kooperaci s jmenovanými zařízeními je vybráno PLC od firmy Phoenix Contact ILC 350 ETH.

Další komponenty zajišťující řízení ovládání a komunikaci jsou také od firmy Phoenix Contact. Ke komponentům zajišťujícím komunikaci bude řečeno více v dalších kapitolách. Přehled vybraných komponentů je znázorněn na blokovém schématu (viz Obrázek 8 na konci kapitoly).

2.3 Návrh konstrukce

Základem konstrukce je rám, na kterém je upevněn motor s převodovkou a inkrementálním snímačem. Na hřídeli převodovky (motoru) je upínkou připevněné rameno a na obou stranách ramene jsou upevněny lapače míčku.

Nosný rám je svařenec ve tvaru rovnostranného trojúhelníku. V každém vrcholu je navařen plech s vyvrstanými dírami pro upevnění motoru. Je usazen na nožičkách se stavitelnými šrouby, které slouží pro vyrovnání konstrukce rámu do vodorovné polohy (Obrázek 3).



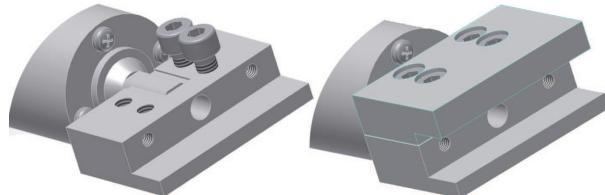
Obrázek 3: Konstrukce stroje
Zdroj: autor práce, 2012

Z obou stran je rám opatřen krytem. Přední strana je potištěna (Obrázek 4) a na zadní jsou upevněny konektory přívodního kabelu pro napájení motoru a pro inkrementální snímač.



Obrázek 4: Potisk trojúhelníku
Zdroj: autor práce, 2012

Rameno je na hřídel převodovky upevněno pomocí speciální upínky dle Obrázku 5. Na každé straně ramene je upevněn lapač míčku složený ze tří částí. První částí je podstavec z hliníkového plechu sloužící k upnutí dalších částí, kterými jsou malý válec (důležitý při vrhání míčku) a trychtýře (zvětšuje místo dopadu).



Obrázek 5: Upínka ramene
Zdroj: autor práce, 2012

Pro umístění řídích prvků (PLC, servozesilovače atd.) je použita prosklená rozvodná skříň, která má v horní části upevněnou dotykovou obrazovku. Slouží tedy i k obsluze zařízení.



Obrázek 6: Model krytu
Zdroj: autor práce, 2012

Pro přívod kabelů je předělána část rozvodné skříně. Snímatelná horní část skříně (kryt) je obrobena a osazena konektory (Obrázek 6). Je zde umístěn pří-

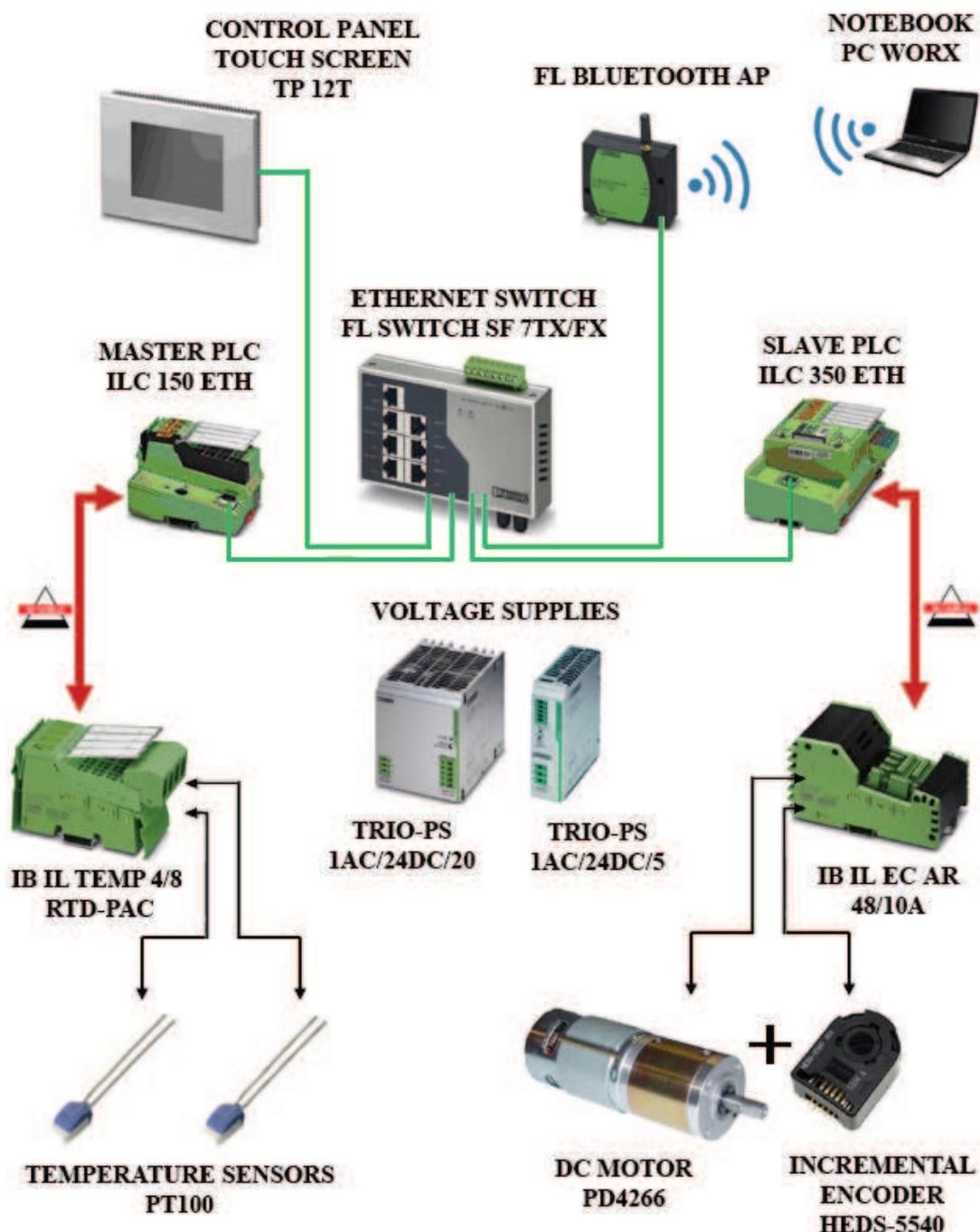
vod napájení z rozvodné sítě, vývod pro napájení motoru, vývod pro kabel inkrementálního snímače a konektory pro připojení síťového kabelu (kroucené dvoulinky s konektory RJ45).

Na následující fotografii (Obrázek 7) je zachycena realizovaná konstrukce zařízení, včetně rozvodné skříně.



Obrázek 7: Realizované zařízení

Zdroj: autor práce, 2012



Obrázek 8: Blokové schéma použitých zařízení

Zdroj: autor práce, 2012

3 Použité komponenty

Kapitola popisuje komponenty, které byly použity pro řešení této diplomové práce, tedy části projektu zabývající se ovládáním, komunikací a vizualizací.

3.1 Dotykový panel TP 12T



Obrázek 9: Dotykový panel TP 12T

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

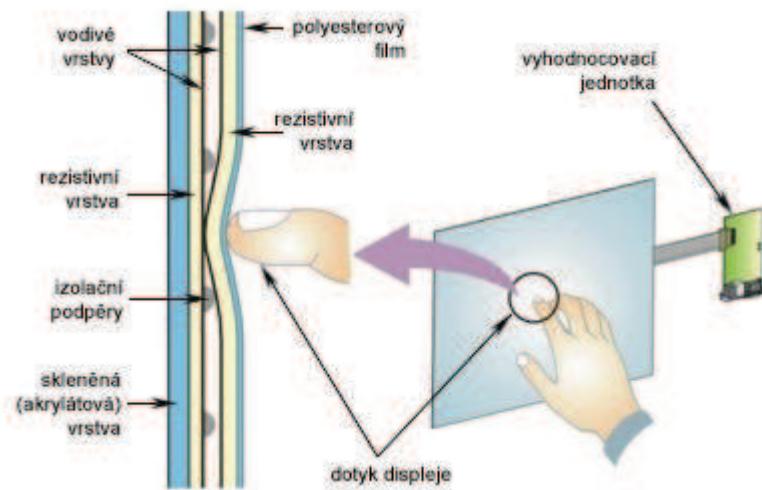
Pro zobrazení a ovládání vytvořeného uživatelského prostředí je použit uvedený dotykový panel.

Panel je v podstatě počítač osazený dotykovým displejem, který využívá rezistivní (odporovou) technologii. Displeje jsou na povrchu tvořeny pružnou membránou, která je zevnitř pokryta tenkou průhlednou kovovou vrstvou. Pod touto měkkou vrstvou je pevná vodivá vrstva. Mezi vrstvami je tenká vzduchová mezera s rastrem izolačních podpěr, které vodivé vrstvy izolují od sebe. Obě vrstvy jsou připojeny k řídícímu a vyhodnocovacímu modulu. Při dotyku se horní vrstva prohne a v daném místě se vodivě spojí s vrstvou spodní. Mezi vrstvami pak začne procházet elektrický proud a kontrolor vypočítá na základě velikosti jednotlivých proudů polohu bodu dotyku. Princip rezisitivního dotykového displeje je znázorněn na Obrázku 10. (<http://www.mobilmania.cz/default.aspx?article=1108570>)

Panel je vybaven odolnou průmyslovou dotykovou obrazovkou o rozměrech 270 x 340 x 5 mm, jejíž displej disponuje 65 536 barvami v rozlišení 800 x 600 Pixelů (SVGA) a podsvícením typu CCFL.

Hardware výbava panelu obsahuje procesor Xscale PXA320 o frekvenci 806 MHz, pracovní paměť 128 MByte SDRAM a datovou paměť 1 GByte Flash. Pro komunikaci je vybaven rozhraním typu USB Host 1.1, Compact Flash a síťovou kartou Ethernet (10/100 MBit), RJ45.

Součástí panelu je i nainstalovaný SW. Konkrétně je zde kromě operačního systému Windows CE nainstalován VISU+ RT (Run Time, vysvětleno v kapitole uživatelské rozhranní visu+) a OPC server. (<http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>)



Obrázek 10: Princip funkce odporového displeje

Zdroj: <http://www.mobilmانيا.cz/default.aspx?article=1108570>

3.2 Programovatelný automat ILC 150 ETH



Obrázek 11: Programovatelný automat ILC 150 ETH

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

Pro potřeby projektu se na pozici řídícího PLC vybrala standardní jednotka ILC 150 ETH napájená 24 V DC.

V rámci obtížnosti řízení části úlohy, kterou se zabývá tato diplomová práce části komunikace a předávání hodnot, teno automat plně dostačuje potřebám aplikace. Přes integrované rozhraní Ethernet lze parametrizovat a programovat automatizačním softwarem PC WORX, vyměňovat paralelně data se servery OPC a komunikovat s účastníky dostupnými přes TCP/IP.

Datová rozhraní

Rozhraní	místní sběrnice INTERBUS (Master)
Způsob připojení	Datové řazení Inline
Rychlosť přenosu	500 kBaud / 2 MBaud umschaltbar
Rozhraní	Zadávání parametrů / obsluha a ovládání / diagnostika
Způsob připojení	RS-232-C, MINI-DIN PS/2, Ethernet 10/100 (RJ45)
Rychlosť přenosu	max. 115,2 kBit/s
Rozhraní	Ethernet 10Base-T/100Base-TX
Způsob připojení	Zásuvka RJ45
Rychlosť přenosu	10/100 MBit/s

Funkce INTERBUS

Provedení	INTERBUS-Master
Počet připojitelných svorek Inline	63
Poznámka k počtu připojitelných svorek Inline	Je nutno respektovat proudový odběr
Počet účastníků s parametrickým kanálem (PCP)	max. 16
Počet podporovaných účastníků	max. 128
Počet vstupních/výstupních bodů	max. 4096

Systém zpracování IEC -61131

Programovací nástroje	PC WORX / PC WORX EXPRESS
Rychlosť zpracování	1,5 ms (Příkazy Mix 1 K)
Datová paměť	256 kB
Remanentní datová paměť	8 kB (NVRAM)
Počet datových prvků	(v závislosti na datové paměti)
Počet časovačů, čítačů	(v závislosti na datové paměti)
Počet úloh řídicích jednotek	8
Hodiny reálného času	ano

(<http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>)

3.3 Ethernet switch



Obrázek 12: Ethernet switch

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

Switch (česky přepínač) je aktivní síťový prvek propojující jednotlivé segmenty sítě. Toto zařízení podle toho, jaký adresát je přidělen k datům, vybere zařízení, se kterým začne komunikovat. Vlastnosti použitého přepínače jsou opět uvedeny níže.

Rozhraní

Název	Inline místní sběrnice
Způsob připojení	Datové řazení Inline
Rychlosť přenosu	500 kBit/s
Fyzika přenosu	měd'

Rozvod potenciálu Inline

Napětí logiky U_L	7,5 V DC (přes rozvod potenciálu)
Příkon z U_L	typ. 75 mA
Periferní napájecí zdroj U_{ANA}	24 V DC
Příkon z U_{ANA}	typ. 28 mA
Příkon	typ. 1,24 W

Analogové vstupy

Počet vstupů	8
Název vstupu	Analogové vstupy RTD
Popis vstupu	Vstup pro odporové teplotní senzory
Způsob připojení	Připojení tažnou pružinou
Technika připojení	2-, 3-vodič (stíněné)
Použitelné typy senzorů (RTD)	Pt, Ni, KTY, Cu, lineární odpory
Princip měření postupná approximace	16 bitů (15 bitů + znaménko)
Znázornění naměřených hodnot	max. 10 μ
Doba převodu A/D	6 ms (Provozní režim možný závisle do 230 ms)
Aktualizace procesních dat	kompatibilní s IB IL, IB ST, S7
Datové formáty	typ. 0,06 %
Přesnost	

(<http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>)

3.4 IL Bluetooth Access Point – 2884758



Obrázek 13: IL Bluetooth Access Point – 2884758

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

Jedná se o prvek pro zavedení bezdrátové komunikace mezi zařízeními standardem Bluetooth (dále také „BT“). BT je bezdrátová technologie patentovaná jako

otevřený standard pro výměnu dat na krátké vzdálenosti z pevných a mobilních zařízení. Tato technologie byla původně koncipována jako bezdrátová alternativa k rozhranní RS -232 datových kabelů. Jak již bylo řečeno dříve slouží moduly tohoto typu k možnosti vzdálené správy obsluhy. (<http://www.bluetooth.com/Pages/about-technology.aspx>)

Zařízení IL Bluetooth AP je přístupový bod bezdrátové BT komunikace.

Kromě BT rozhranní obsahuje také Ethernetou přípojku. Podporovanými komunikačními BT profily jsou LAN Access Profile(LAP), kde je možné přistupovat k jiným sítím, nicméně vzhledem k protokolu komunikace vyžaduje v síti server. Serial Port Profile (SPP) je serverovou funkcí, která emuluje sériové rozhranní, díky čemuž lze připojit zařízení původně komunikujících pouze na RS-232 rozhranní. Posledním je Personal Area Networking (PAN), kde modul může být použit jako přístupový bod nebo jako klient. Takto může komunikovat až sedm zařízení bez použití serveru. Tento profil je pomocí webové správy nastaven pro BT komunikaci použitou při řešení projektu. (<http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>)

3.5 IL TEMP



Obrázek 14: IL TEMP

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

Jedná se o svorkovnici/ měřící kartu schopnou vyčítat hodnoty až z osmi odporových snímačů teploty. Modul podporuje snímače z platiny a niklu dle DIN a směrnic SAMA. U snímačů na bázi platiny jsou kromě Pt 100, Pt 500 a Pt 1000 podporovány i Pt 10000 (např. z technického vybavení budov) i další snímače a záznamníky naměřené hodnoty.

Připojení snímačů je možno provést dvěma nebo třemi vodiči. Naměřené teploty jsou zobrazeny ve tvaru 16bitových hodnot v každém kanálu (2x 4 kanály v multiplexu dat procesu) anebo pomocí 16 bitů v každém kanálu při komunikaci PCP. Další informace o měřícím modulu jsou v následujících výitech.

Rozhraní

Název	Inline místní sběrnice
Způsob připojení	Datové řazení Inline
Rychlosť přenosu	500 kBit/s
Fyzika přenosu	měď

Analogové vstupy

Počet vstupů	8
Název vstupu	Analogové vstupy RTD
Popis vstupu	Vstup pro odporové teplotní senzory
Způsob připojení	Připojení tažnou pružinou
Technika připojení	2-, 3-vodič (stíněné)
Použitelné typy senzorů (RTD)	Pt, Ni, KTY, Cu, lineární odpory
Princip měření	postupná approximace
Znázornění naměřených hodnot	16 bitů (15 bitů + znaménko)
Doba převodu A/D	max. $10 \mu s$
Aktualizace procesních dat	6 ms (Provozní režim možný závisle do 230 ms)
Datové formáty	kompatibilní s IB IL, IB ST, S7
Přesnost	typ. 0,06 %

(<http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>)

3.6 Senzor PT 100



Obrázek 15: Senzor PT 100

Zdroj: <http://select.phoenixcontact.com/phoenix/dwl/dwlfr1.jsp?lang=en>

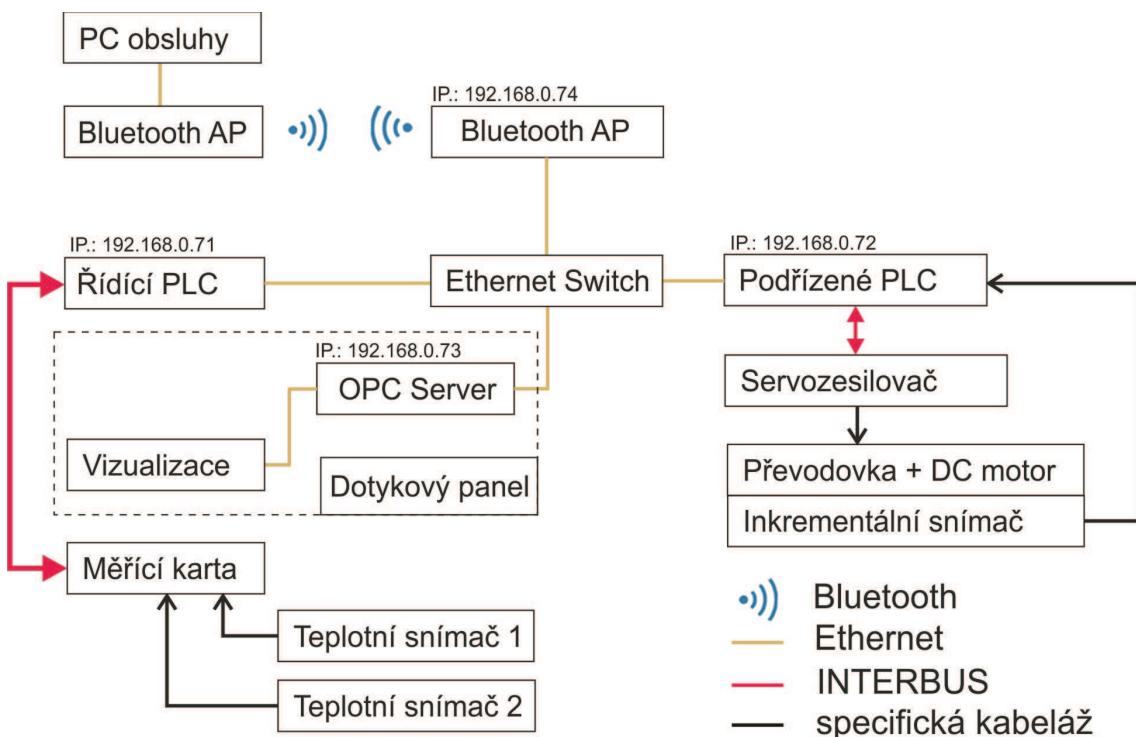
Senzor PT 100 je teplotní snímač sloužící pro určení teploty jeho okolí s využitím závislosti odporu kovů na teplotě. Atomy krystalové mřížky kovu s rostoucí teplotou zvyšují amplitudu svých kmitů a kladou tak věší odpor průchodu elektronů. S rostoucí teplotou tedy roste odpor snímače.

Pro kovové senzory teplot, stejně jako tomu je u použitého snímače, se většinou využívá platina pro její chemickou netečnost, časovou stálost, vysokou teplotu tavení a také pro možnost dosažení vysoké čistoty (pohybuje se v rozmezí 99,9 až 99,999 %). Kromě platiny je možné se setkat též s niklem, mědí, molybdenem nebo některými slitinami. V projektu jsou tyto použity dva snímače PT100, které měří teplotu v rozvodné skříni. Hodnoty teplot z nich snímá měřící karta IB IL TEMP 4/8 RTD. (<http://www.hw.cz/theorie-a-praxe/dokumentace/mereni-teploty-kovove-odporove-senzory-teploty.html>)

4 Použitá komunikace

Proto, aby mohlo zařízení pracovat, musí být zajištěna komunikace mezi všemi komponenty. Podstatou komunikace řešené úlohy je provedení příkazu podle parametrů nastavených na dotykovém panelu. Hodnoty parametrů je nutno zpracovat a zaslat programovatelnému automatu obsluhujícího ovládání motoru.

Z blokového schématu všech komponentů stroje uvedeného v kapitole Návrh zařízení je dán způsob propojení jednotlivých zařízení. Dotykový panel, Bluetooth (dále také „BT“) modul na straně rozvaděče a obě PLC jsou síťovými kably propojeny prostřednictvím rozbočovače. K servozesilovači a modulu měřící teplotu se přistupuje sběrnicí INTERBUS. Pro možnost bezdrátového přístupu jsou využity dva Bluetooth komunikační moduly, kde jeden je umístěn přímo v prostoru rozvaděče a druhý je připojen k PC obsluhy.



Obrázek 16: Komunikační schéma zařízení

Zdroj: autor práce, 2012

Základní protokoly komunikace jsou tedy tři. Pro zajištění funkce komunikace zařízení v lokální síti Ethernet, je nutné nastavit správně IP síťové adresy. Jaké byly přiřazeny IP adresy, jednotlivým komponentům je patrné z OBRÁZKU X. Pro komunikaci se zařízením připojeného pomocí sběrnice INTERBUS, je nutné provést správné nastavení projektu v programu PC WORX. V základu se jedná o výběr zařízení připojeného na sběrnici a přiřazení proměnných fyzickým adresám (ovlá-

dacím a kontrolním slovům). Pro komunikaci prostřednictvím Bluetooth je potřeba provést správnou konfiguraci BT modulů.

Nezbytnou součástí komunikace je OPC server, který umožňuje přístup k OPC proměnným, jež slouží k předávání hodnot mezi aktivními prvky vizualizace. Samotný server běží na počítači, který je součástí dotykového panelu.

Komunikaci mezi komponenty je věnován prostor i v dalších kapitolách.

4.1 Podrobný popis komunikace

K tomu, aby mohla probíhat komunikace (viz Obrázek X), je nutné vytvořit projekt v programu PC WORX a současně v rámci místní sítě mít spuštěný OPC server, na kterém se ukládají OPC proměnné.

V prvé řadě je nutné, aby obsluha zařízení nahrála program vytvořený v PC WORX pomocí BT modulů na PLC 1 popřípadě PLC 2. Následuje vybrání OPC proměnných ve vývojovém prostředí programu VISU+ a jejich přiřazení k vybrané vlastnosti objektu. Na základě provedených akcí ve vizualizaci dojde k nastavení hodnot OPC proměnných. Dle hodnot konkrétních proměnných program v PLC rozhoduje o tom, zda, nebo jaká data budou zaslána druhému automatu. Pro potřeby komunikace mezi automaty mají zasílaná data určitý tvar, který je blíže popsán v kapitole Realizace řídícího programu. Zasílání dat je uskutečňováno pomocí předdefinovaných funkčních bloků programu PC WORX.

Druhý automat, pomocí stejných funkčních bloků, zaslaná data přijímá, zpracovává a na jejich základě rozhoduje o řízení motoru. Řízení pohybu motoru je uskutečněno pomocí regulátorů implementovaných v servozesilovači. K motoru je připojen inkrementální snímač měřící natočení ramene zařízení. Hodnoty získané snímačem představují zpětnou vazbu pro regulaci.

K prvnímu PLC je připojena měřící karta, která přebírá hodnoty z teplotních snímačů vyvedených v rozvodné skříni zařízení. Pro samotnou komunikaci mezi měřící kartou a automatem je nutné nastavení ovládacích slov, problematika je blíže specifikována opět v kapitole Realizace řídícího programu. Aby bylo možno získat snímané hodnoty, nutné připojit zařízení do sběrnicové struktury a nastavit jeho vstupní a výstupní parametry odpovídajícím proměnným. Proměnné jsou opět proměnnými OPC a jejich hodnoty jsou zobrazeny ve vizualizaci.

4.2 Nutná konfigurace

Pro Ethernetovou komunikaci je potřeba nastavení IP adres jednotlivých automatů. Všechny nastavené adresy musí být jedné třídy. Komunikace mezi PLC je prováděna prostřednictvím struktury třech proměnných, a to mode, speedRelease a angleRelease (viz kapitola Realizace řídícího programu)

Na dotykovém panelu se nachází jak vizualizace spuštěná programem VISU+, tak i výše zmíněný OPC server. V konfiguraci dotykového panelu je nutno nastavit IP adresu PLC, se kterým bude prostřednictvím OPC serveru navázána komunikace.

Aby byla umožněna bezproblémová komunikace je nutno přiřadit odpovídající IP adresu (stejná třída) také samotnému dotykovému panelu, na kterém je OPC server spuštěn. Pro přístup obsluhy je nutné provést konfiguraci BT modulů. Nastavením BT profilu Personal Area Networking (PAN), jako aktivního profilu ke komunikaci je umožněno navázání spojení dvou modulů. Potom již stačí nastavit IP adresu modulu umístěného v rozvodné skříni.

5 Program VISU+

VISU+ je SCADA/HMI vizualizační software, který umožňuje změnit počítač, dotykový panel, mobilní zařízení nebo jiné pracovní stanice na zařízení schopné získávat a zpracovávat data potřebná pro řízení.

Proces řízení zařízení jako jsou PLC, termoregulátory, inteligentní karty a počítače, může probíhat pomocí sériových linek, modemů a různých komunikačních sítí, a to i bezdrátových.

VISU+ je softwarová platforma pro vytváření a spuštění projektů sloužících pro dohled a řízení různých druhů automatických systémů.

V závislosti na operačním systému nainstalovaném na zařízení nejsou některé funkce podporované. Ať už se jedná o grafické funkce, například rotaci objektů, nebo o alarmové funkce, omezení pro WinCE existují. Nicméně v rámci řešeného problému je funkční vybavenost tohoto systému plně postačující.

Program má dva pracovní režimy, jeden pro vývoj (Development) a druhý pro provoz (Runtime):

- **Vývojový režim (Development mode)** – V tomto režimu systém umožňuje vytvářet nebo upravovat soubory projektu prostřednictvím vkládání objektů (zdrojů) a konfigurovat je pomocí vlastností.
- **Provozní režim (Runtime mode)** – Tento mód umožňuje spuštění projektu. Uživatel zde může sledovat data nebo případně používat příkazy připravené programátorem.

5.1 Programování ve VISU+

VISU+ projekty mají za úkol dohlížet na výrobní či jiné procesy prostřednictvím animovaných grafických stránek tzv. obrazovek („Screens“). Na tyto obrazovky je možné rozmištit objekty jako jsou tlačítka, měříče nebo textová pole.

Vlastnosti těchto objektů jsou zpravidla vhodně propojeny se systémem proměnných, tak aby bylo docíleno funkčnosti daného řešení. K objektům a k vlastnostem se přistupuje obdobně například jako v prostředí Delfi.

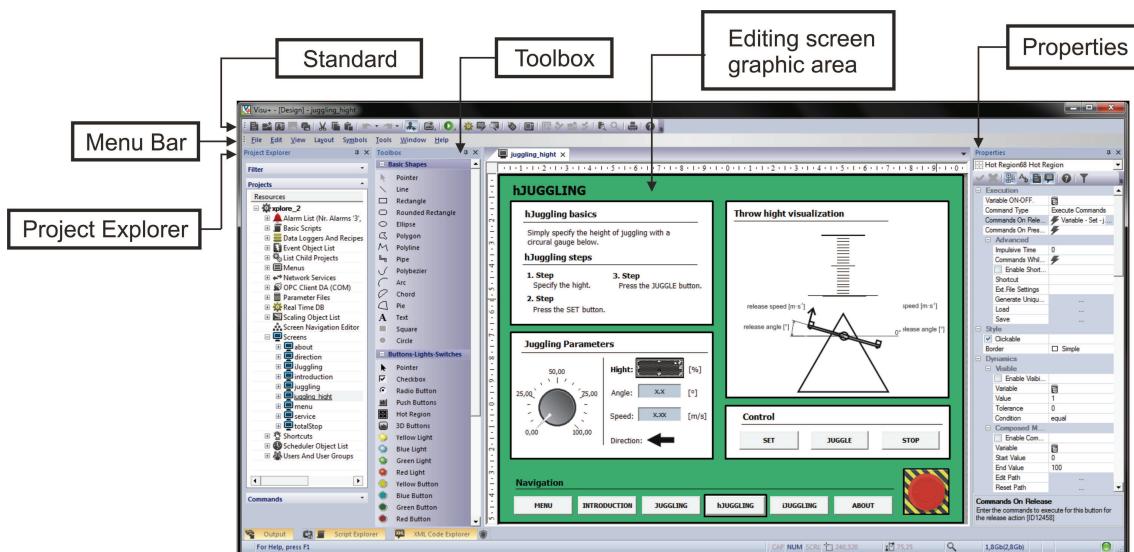
5.2 Založení projektu

Po spuštění programu vybereme v menu „File“ příkaz „New“ (Ctrl+N). Nejdříve vybereme platformu, ve které budeme projekt tvořit. Vzhledem k tomu, že je na dotykovém panelu nainstalovaný operační systém Windows CE, vybere se platforma Windows CE. Podle výběru nebudou k dispozici některé pokročilé funkce a vlastnosti objektů. Pro potřeby projektu je však toto omezení minimální.

Na další obrazovce je potřeba zadat jméno projektu a vybrat umístění kam bude projekt uložen. Projekt není chráněn heslem a není potřeba přidávat ovladače dalších zařízení proto se v následujících oknech Users, Add Comm.I/O Driver nic

nevyplňuje. Dalším okno Screens umožňuje nastavit šířku a výšku vytvářených Screens („obrazovek“) a lze nastavit kolik se jich po dokončení průvodce vytvoří. Další okna slouží k nastavení databáze a alarmů. Ukládání dat pro řešený projekt nemá význam a proto v se v těchto oknech ponechá výchozí nastavení. Všechny zvolené volby lze po dokončení průvodce změnit.

5.3 Pracovní prostředí



Obrázek 17: Pracovní prostředí programu VISU+

Zdroj: autor práce, 2012

Jako většina programovacích aplikací má i VISU+ intuitivně rozdělené ovládací panely a okna. Popsány budou jen ty, které byly pro při tvorbě projektu využity.

Základním panelem standardním pro většinu aplikací je panel nabídek. Ve VISU+ označen jako „Menu Bar“ (Obrázek 18). Zde lze především vybrat jaké panely a nástroje budou zobrazeny. Nabídka označená „Standard“ umožňuje snadný přístup k ukládání změn, spuštění a zejména k nahrání projektu na dotykový panel. Všechny dostupné zdroje právě otevřeného projektu obsahuje okno Project Explorer. Pro řešený problém jsou vzhledem k jeho zaměření využity pouze tři zdroje, které jsou popsány níže.

1. OPC Client DA

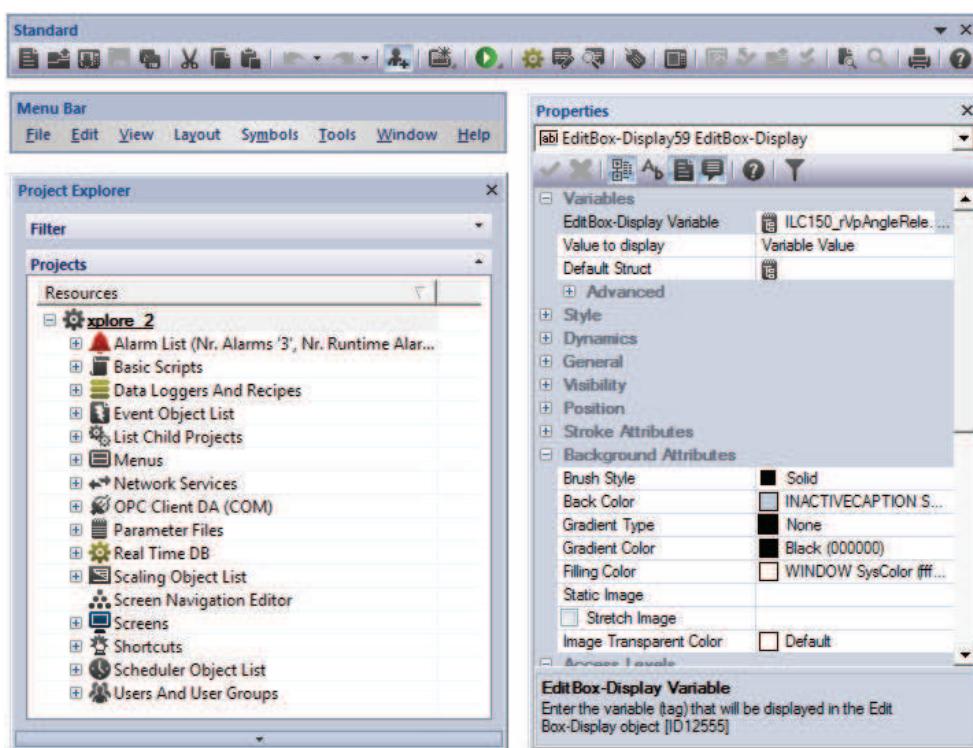
Proměnné připojené prostřednictvím OPC serveru z programu spuštěného na PLC.

2. Real Time DB

Databáze všech proměnných použitých v projektu (proměnné převzaté z OPC serveru společně s proměnnými lokálními).

3. Screens („obrazovky“)

Obrazovka je plocha, na kterou se umístují ovládací, informační a grafické objekty z panelu nástrojů „Toolbox“. Objekty jsou svázány některou svou vlastností s proměnnými a tvoří tak základ vizualizace, jejímž cílem je přiblížení řešeného problému a zajištění jeho ovládání.



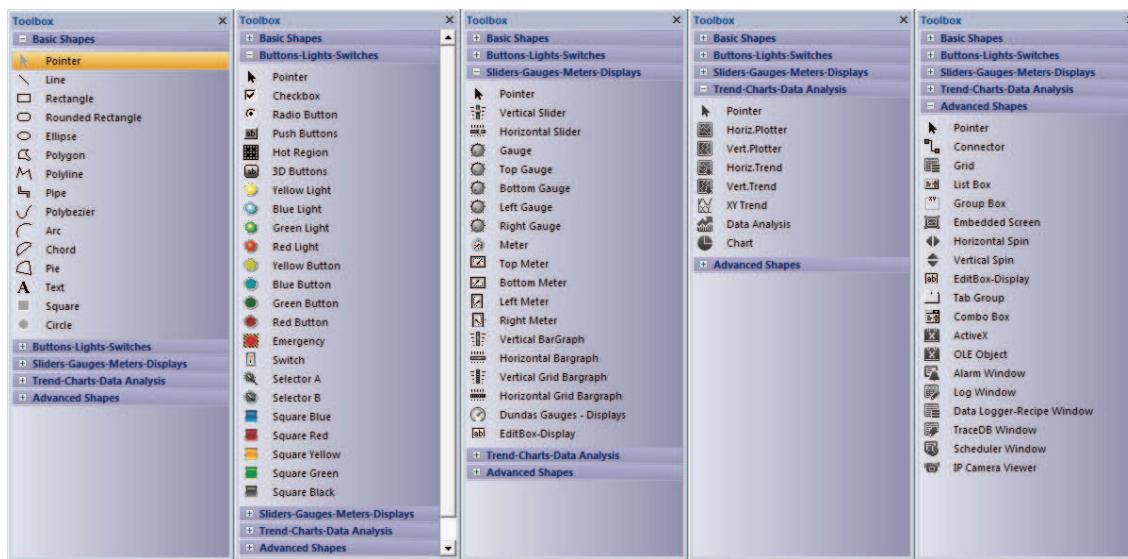
Obrázek 18: Použité ovládací panely programu VISU+
Zdroj: autor práce, 2012

Panel nástrojů „Toolbox“ obsahuje množství objektů rozčleněných do několika kategorií. Popsány jsou pouze objekty, které byly při tvorbě diplomové práce použity.

Všechny objekty mají řadu vlastností. Popsány jsou opět jen ty vlastnosti, které byly využity při tvorbě diplomové práce. Přístup k těmto vlastnostem je umožněn panelem Properties implicitně zobrazeným v pravé části pracovního prostředí programu VISU+.

5.4 Basic shapes (dále „základní útvary“)

Jedná se o jednoduché objekty typu úsečka nebo obdélník. Slouží pro docílení elegantního, přehledného a funkčního vzhledu obrazovek. Většina z nich je statická, některé jsou však provázány s proměnnými tak, že v závislosti na změnách hodnot těchto proměnných dochází k dynamickým změnám ve vizualizaci.



Obrázek 19: Všechny objekty panelu Toolbox

Zdroj: autor práce, 2012

5.4.1 Skrytí/zobrazení objektu

V projektu je využito několik objektů Rectangle (dále „obdélník“) takovým způsobem, že se v závislosti na logické hodnotě proměnné (TRUE nebo FALSE) přiřazené k vlastnosti „Visible“ po spuštění programu zobrazí nebo nezobrazí.

5.4.2 Obrázek jako pozadí objektu

Další vlastnosti, které je využito je „Static Image“. Jedná se o vlastnost zobrazeného pozadí objektu obdélník. Přiřazením vhodného obrázku na pozadí objektu, je možné přispět ke snadnějšímu pochopení funkcí naprogramovaných pro danou obrazovku.

5.4.3 Dynamický přesun objektu

Poslední z využitých vlastností pro základní útvary je „Enable Y movement“. Pokud je zaškrtnutý check-box („zaškrtávací políčko“) u této vlastnosti umožní objektu reagovat na proměnnou zadанou ve vlastnosti „Variable“. Reakcí je myšlen přesun daného objektu v horizontálním směru podle hodnot přidělené proměnné.

5.5 Button (dále „tlačítko“)

Základní objekt pro jakékoliv prostředí sloužící pro ovládání. Využito je nejvíce vlastnosti „Command On release“. Pod tuto vlastnost lze připojit příkazy, které se provedou po klepnutí na tlačítko. Příkazy použité v projektu pro objekty tlačítka jsou:

5.5.1 Nastavení hodnoty proměnné

Příkaz se nastaví klinutím do vlastnosti „Command On release“ vybraného tlačítka. Po této akci se zobrazí okno „Command List“ a vybere se „New Command“. Zde v záložce „Variable“ se v buňce „Variable“ vybere proměnná, se kterou se bude pracovat a v buňce „Action“ se vybere možnost „Set“. Do buňky „Value“ se zapíše hodnota, která má být do proměnné nastavena. Celý příkaz vypadá následovně: Variable – Set – *jmeno_promenne*.

5.5.2 Přepnutí na jinou obrazovku

Přepnutí obrazovky se nastaví obdobně jako nastavení hodnoty proměnné. Rozdílem je, že se v „Command Type“ okně vybere záložka „Screen“. V této záložce se v buňce „Screen“ zapíše jméno obrazovky, ke které se chce tímto příkazem přistoupit. V buňce „Action“ se vybere *Open normal (screen change)*. Po potvrzení zní příkaz následovně: Screen – *Open normal (screen change) – jmeno_obrazovky*.

5.5.3 Otevření obrazovky v novém okně

Nastavení tohoto příkazu se provede téměř totožně jako příkaz pro přepnutí na jinou obrazovku. Rozdíl je ve výběru vybrané akce v buňce „Action“. Zde se vybere Open frame (multi-monitor). Celý příkaz potom vypadá následovně: Screen – *Open frame (multi-monitor) – jmeno_obrazovky*.

5.5.4 Deaktivace tlačítka

Další použitá vlastnost objektu tlačítka je „Enable var.“. Logická hodnota proměnné přiřazená této vlastnosti rozhoduje o tom zda tlačítko bude nebo nebude provádět akce zadané ve vlastnosti „Command On Release“ nebo „Command On Pressed“. Nastavením hodnoty FALSE do přiřazené proměnné znemožní provádění přiřazených akcí daného tlačítka.

5.5.5 EditBox – Display (dále „editační okno“)

Objekt, který zobrazuje hodnotu proměnné přiřazené k základní vlastnosti „EditBox – Display Variable“.

5.6 Hot Region

Objekt s vlastnostmi obdobnými jako tlačítko, který je však avšak při vývoji v pracovním prostředí je reprezentovaný šedým průhledným obdélníkem a po spuštění projektu je neviditelný.

Program je vyvíjen pro dotykový panel, který nemá k dispozici klávesnici. Pro řešenou úlohu je však potřeba hodnoty zapisovat. Docíleno toho je tak, že editační okna do kterých je nutné zapisovat jsou překryta objekty Hot Region, které mají

ve k vlastnosti „Command On Release“ přiřazen příkaz nastavení hodnoty vybrané proměnné pomocí numerické klávesnice. Je tedy žádoucí aby oba tyto objekty byly svázány stejnou proměnnou. Pomocí numerické klávesnice pak dojde k nastavení proměnné zobrazené editačním oknem.

5.6.1 Zapsání hodnoty proměnné pomocí numerické klávesnice

Příkaz se nastaví téměř shodně jako v případě nastavení hodnoty proměnné pro objekt tlačítka. Změnou je výběr jiné akce v buňce „Action“. V tomto případě se nevybere možnost „Set“, ale „Numeric Pad“. Celý příkaz vypadá poté následovně: *Variable – Numeric Pad – jméno_promenne*.

5.7 Trend

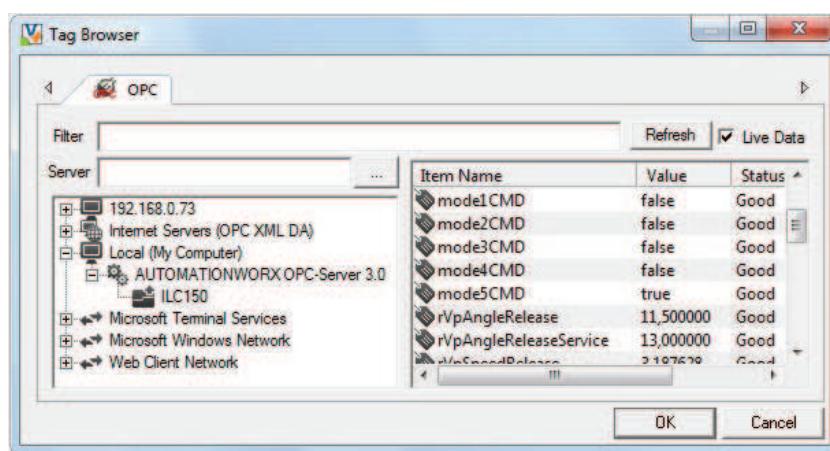
Objekt, který vykresluje aktuální hodnotu proměnné přiřazené k vlastnosti „Pen“.

Vlastnost představuje pero, které bude tuto hodnotu zapisovat, v určitých časových intervalech (délka intervalu nastavitelná ve vlastnostech), na pozadí objektu.

5.7.1 Přiřazení proměnné trendu

Přiřazení se provede kliknutím do vlastnosti „Edit Pens“. Po této akci se zobrazí dialogové okno „Pens“, dále se stiskne tlačítko „Add“, které zobrazí dialogové okno „Edit Pen Properties“ a zde se do buňky „Variable Pen“ zadá jméno proměnné, kterou chceme vykreslovat.

5.8 Připojení dynamické OPC proměnné



Obrázek 20: Připojení OPC proměnné

Zdroj: autor práce, 2012

V panelu Project Explorer se vybere zdroj OPC Klient DA(COM). Kliknutím pravého tlačítka myši se přejde na nabídku, kde se vybere „Add New OPC Item ...“. Zde se v levé straně okna „Tag Browser“ vyhledá spuštěný OPC server a na pravé se zobrazí aktivní OPC proměnné. Vybere se žádaná proměnná a výběr se potvrdí.

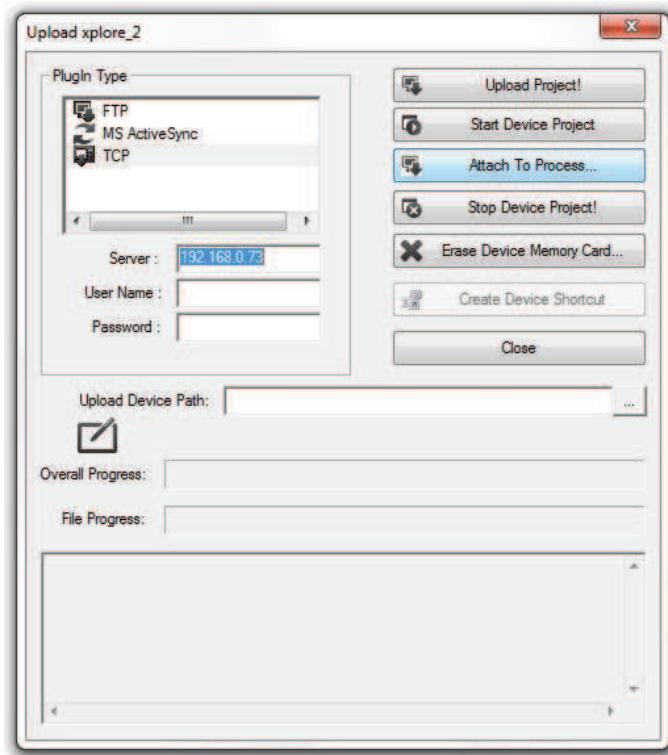
Ve zdroji jsou vidět jednotlivé připojené proměnné. Jména proměnných jsou modifikována přidáním ILC150 před jejich původní název. Například z proměnné *setCMD* tak pro potřeby VISU+ vznikne *ILC150.setCMD*.

Přidáním proměnné se současně vytvoří prvek OPC server, který ve vlastnosti „Network Node“ musí mít zadанou IP adresu zařízení, na kterém je spuštěn OPC server. V tomto případě se jedná o dotykový panel TP 12T s adresou 192.168.0.73.

5.9 Nahrání projektu na dotykový panel

V nabídce Standard se vybere možnost „Upload Project“. Zobrazí se okno kde se v části „PlugIn Type“ (typ spojení) vybere možnost TCP a do editačního okna Server se zapíše IP adresa zařízení kam má být projekt umístěn. Opět se jedná o IP adresu dotykového panelu. Samotné nahrání se provede tlačítkem „Upload Project!“.

K spuštění programu poté dojde buďto tlačítkem „Start Device Project“ nebo prostým restartem dotykového panelu.



Obrázek 21: Okno nastavení pro nahrání projektu
Zdroj: autor práce, 2012

6 Realizace uživatelského rozhraní

Kapitola popisuje konkrétní realizaci uživatelského rozhraní, kdy jednotlivé podkapitoly blíže specifikují použité obrazovky vytvořené programem VISU+. Na každou obrazovku je nahlízeno ze dvou pohledů. Vždy je nejprve uveden obecný popis obrazovky a následně popis funkčních ovládacích prvků.

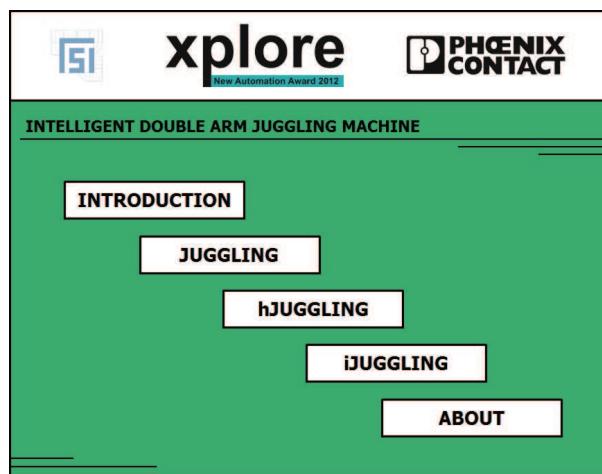
Uživatelské rozhraní je vytvořeno s důrazem na přehlednost a jednoduchost obsluhy. Jak již bylo zmíněno, prostředí je rozděleno do několika obrazovek. Pro jednodušší orientaci v textu je název jednotlivých obrazovek vždy uveden velkým písmem. Mezi jednotlivými obrazovkami se lze přepínat pomocí obrazovky MENU nebo prostřednictvím tlačítka navigace, jež slouží pro rychlé přepínání bez nutnosti vracet se k MENU. Tlačítka navigace jsou umístěna ve spodní části každé obrazovky, vyjma zmíněného MENU.

Obrazovky lze členit na informativní, servisní a ovládací. Ovládací obrazovky umožňují uživateli přímé ovládání zařízení a proto je jim věnována podstatná část kapitoly.

6.1 Obrazovka MENU

Obecný popis

Po spuštění vizualizačního projektu vytvořeného ve VISU+ se jako první zobrazí obrazovka MENU (Obrázek 22). Tohoto bylo docíleno vybráním obrazovky jako „Startup Screen“ ve vlastnostech projektu.



Obrázek 22: Obrazovka – MENU

Zdroj: autor práce, 2012

Obrazovka MENU slouží pro přechod na některou z nabízených obrazovek. Přechod je umožněn klepnutím na jednu z uvedených možností. Pomocí neviditelného objektu umístěného v levé spodní části lze také přejít do obrazovky SERVICE.

Popis funkčních ovládacích prvků

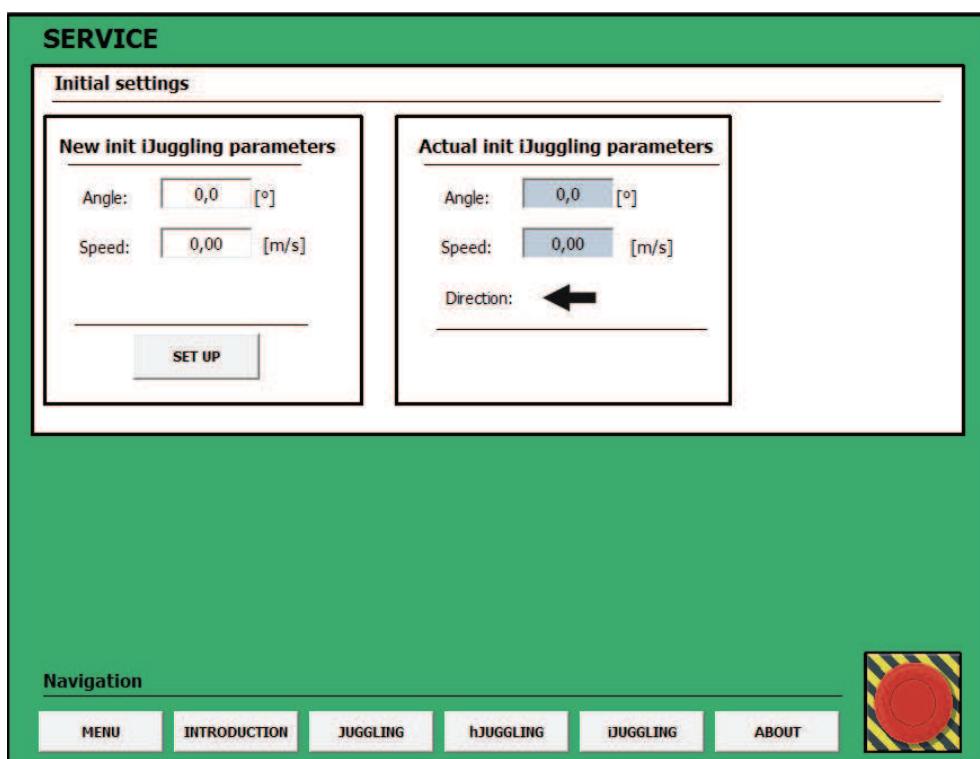
Přechod na další obrazovky je umožněn prostřednictvím objektů VISU+ zvaných Hot Region. Při tvorbě projektu v programu VISU+ tyto objekty vypadají jako šedé obdélníky. Při spuštění projektu na dotykovém panelu tyto obdélníky vidět nejsou, nicméně pokud uživatel klepe na oblast kde se nacházejí, provede se akce, kterou mají přidělenou.

V tomto případě mají obdélníky přidělený příkaz *Screen – Open normal (screen change) – název_obrazovky*, který provede přepnutí na vybranou obrazovku dle popisu na obdélníku. Obdélníky se poté chovají obdobně jako tlačítka.

6.2 Obrazovka SERVICE

Obecný popis

SERVICE je servisní obrazovka (Obrázek 23), ke které nemá uživatel přímý viditelný přístup. Přístup je možný pouze přes neviditelný objekt v levé spodní části obrazovky MENU. Slouží pro nastavení parametrů pro režim iJuggling.



Obrázek 23: Servisní obrazovka – SERVICE

Zdroj: autor práce, 2012

V levé části nazvané „New Init iJuggling parameters“ lze prostřednictvím numerické klávesnice, která se zobrazí po klepnutí na příslušné editační okno, zapsat

nové hodnoty pro úhel a rychlosť. Aby byly tyto hodnoty pro daný režim převzaty, musí být potvrzeny stiskem tlačítka SET UP.

V pravé části „Actual Init iJuggling parameters“ jsou zobrazeny aktuálně nastavené hodnoty zmíněných parametrů. Ve spodní části obrazovky, se stejně tak, jak tomu bude u všech dalších obrazovek, objevuje panel tlačítek navigace a tlačítka EMERGENCY STOP.

Jak již bylo řečeno umožňují tlačítka navigace, stejně jako MENU, přechod na zvolenou obrazovku. Funkce tlačítka EMERGENCY STOP spočívá v nouzovém vypnutí zařízení v případě havárie. Tlačítko vydá příkaz, který okamžitě odpojí motor zařízení od napájení, tím je motor zastaven a je zabráněno případnému zranění uživatele. Současně se zobrazí upozornění o použití tlačítka EMERGENCY STOP.



Obrázek 24: Upozornění při stisku tlačítka EMERGENCY STOP

Zdroj: autor práce, 2012

Popis funkčních ovládacích prvků

Podobně jako tomu je u MENU, se po klepnutí na tlačítka (v MENU na obdélníky s názvy obrazovek) navigace pomocí příkazu *Screen – Open normal (screen change)* – *název_obrazovky* provede přechod na obrazovku dle popisu na konkrétním tlačítku. Dále zde nalezneme čtyři editační okna, z toho dvě slouží jen ke zobrazení aktuálních nastavených hodnot proměnných *ILC150_rVpAngleReleaseService* a *ILC150_rVpSpeedReleaseService*. Z toho důvodu nemají tato dvě okna povolenou možnost editace.

Další dvě editační okna, sloužící pro zápis nových hodnot, jsou překryta objekty Hot Region tak, že po klepnutí na tyto objekty dojde pomocí příkazu *Variable – Numeric pad – promenna* ke zobrazení numerické klávesnice. Prostřednictvím této klávesnice zapíšeme žádanou hodnotu parametru. Tato hodnota je uložena do příslušné proměnné, kterými jsou *ILC150_rVpAngleRelease* a *ILC150_rVpSpeedRelease*.

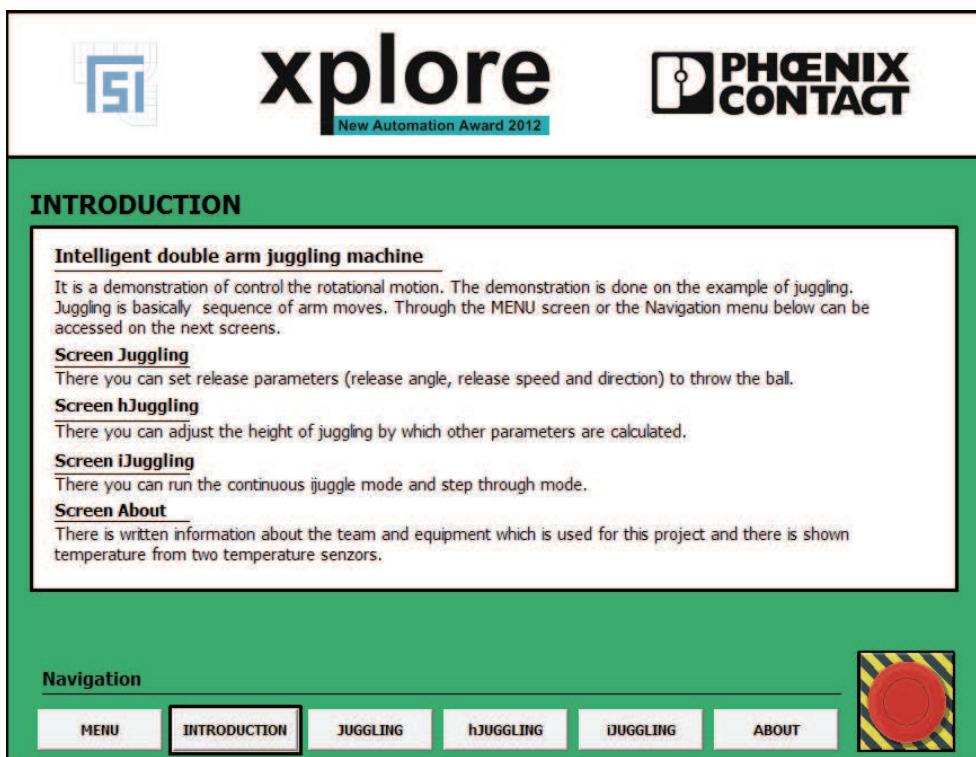
V neposlední řadě je zde a na všech dalších ovládacích obrazovkách umístěno EMERGENCY STOP. Po stisknutí tlačítka se nastaví na hodnotu TRUE proměnná *ILC150_totalStopCMD* a okamžitě odpojí motor od zařízení a provede se zobrazení obrazovky TOTALSTOP, která je vidět na obrázku výše (Obrázek 24). Na obrazovce TOTALSTOP je jen jedna možnost, a to tlačítko, které po aktivaci provede zavření obrazovky a nastaví proměnnou *ILC150_totalStartCMD* na hodnotu TRUE, čímž dojde k opětovnému zapnutí napájení motoru zařízení.

6.3 Obrazovka INTRODUCTION

Obecný popis

Jak je patrné z Obrázku 25, je INTRODUCTION obrazovkou čistě informativní. Nalezneme zde stručný popis nejen řešené problematiky, ale také toho, co obsahuje další nabízené obrazovky. Ve spodní části obrazovky, se stejně tak, jak tomu bylo u předešlé obrazovky a bude u všech dalších obrazovek, objevuje panel tlačítek navigace a tlačítko EMERGENCY STOP.

Na rozdíl od obrazovky SERVICE mají všechny ostatní obrazovky pro snadnější orientaci zvýrazněnou aktuální pozici uživatele orámováním příslušného tlačítka v panelu navigace.



Obrázek 25: Informativní obrazovka – INTRODUCTION

Zdroj: autor práce, 2012

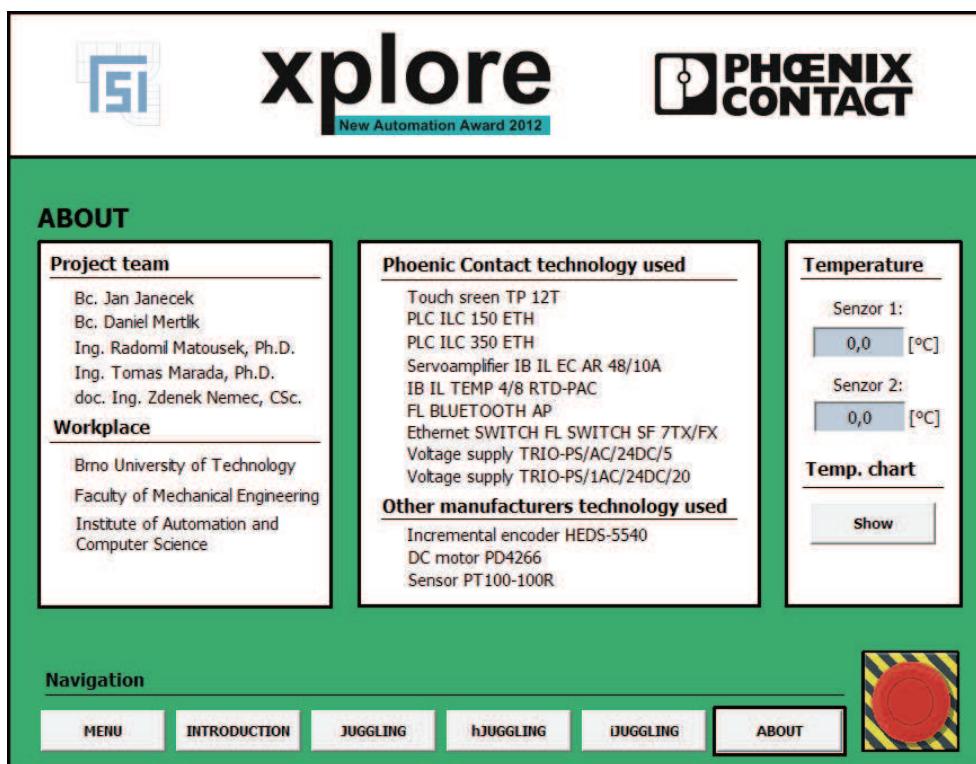
Popis funkčních ovládacích prvků

Kromě tlačítek navigace pro přepnutí na jinou z obrazovek a tlačítka EMERGENCY STOP se zde jiné funkční ovládací prvky nenachází. Návaznost tlačítek na proměnné je popsána již u obrazovky SERVICE.

6.4 Obrazovka ABOUT

Obecný popis

Stejně tak, jako první obrazovka nabízena v MENU, má i poslední ABOUT (Obrázek 26) čistě informativní charakter. Jsou zde uvedeni členové týmu, pracoviště týmu, použité komponenty a teplota.



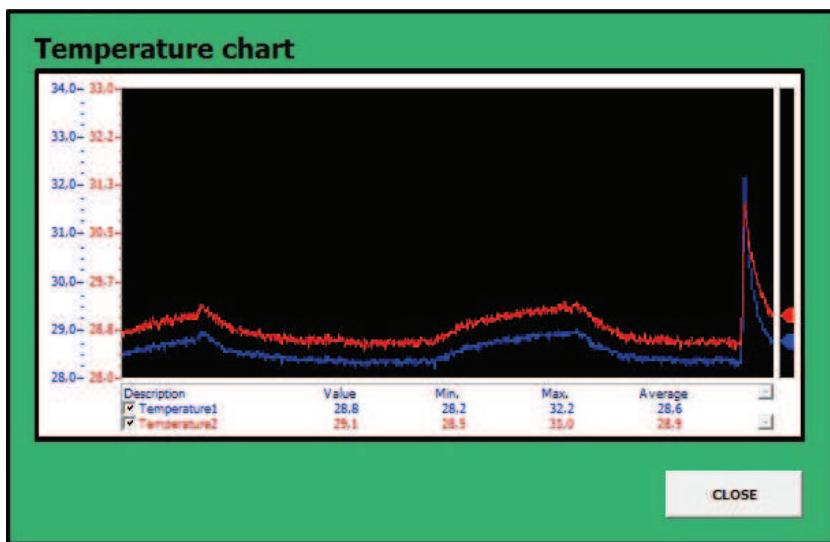
Obrázek 26: Informativní obrazovka – ABOUT

Zdroj: autor práce, 2012

Teplotou je myšlena naměřená teplota v rozvodné skříni zařízení, kdy jsou na obrazovce ABOUT vypsány aktuální hodnoty teplot ze dvou snímačů. Dále je zde umístěno tlačítko umožňující přechod na graf vývoje teplot, který je možno vidět na Obrázku 27.

Popis funkčních ovládacích prvků

Funkční prvky se zde nacházejí tři. Dvě editační okna zobrazující hodnoty proměnných *ILC150_TEPLOTA_TW1* a *ILC150_TEPLOTA_TW2* a tlačítko SHOW



Obrázek 27: Graf vývoje teplot

Zdroj: autor práce, 2012

umožňující pomocí příkazu *Screen Open frame (multi-monitor)* – *Temperature chart* zobrazit průběh měřených teplot. Průběh je zobrazen v nově otevřeném samostatném okně obsahujícím objekt TREND a jedno tlačítko CLOSE, které po klepnutí provede příkaz *Screen Close and return back – Temperature chart*, čímž okno zavře.

6.5 Obrazovka JUGGLING

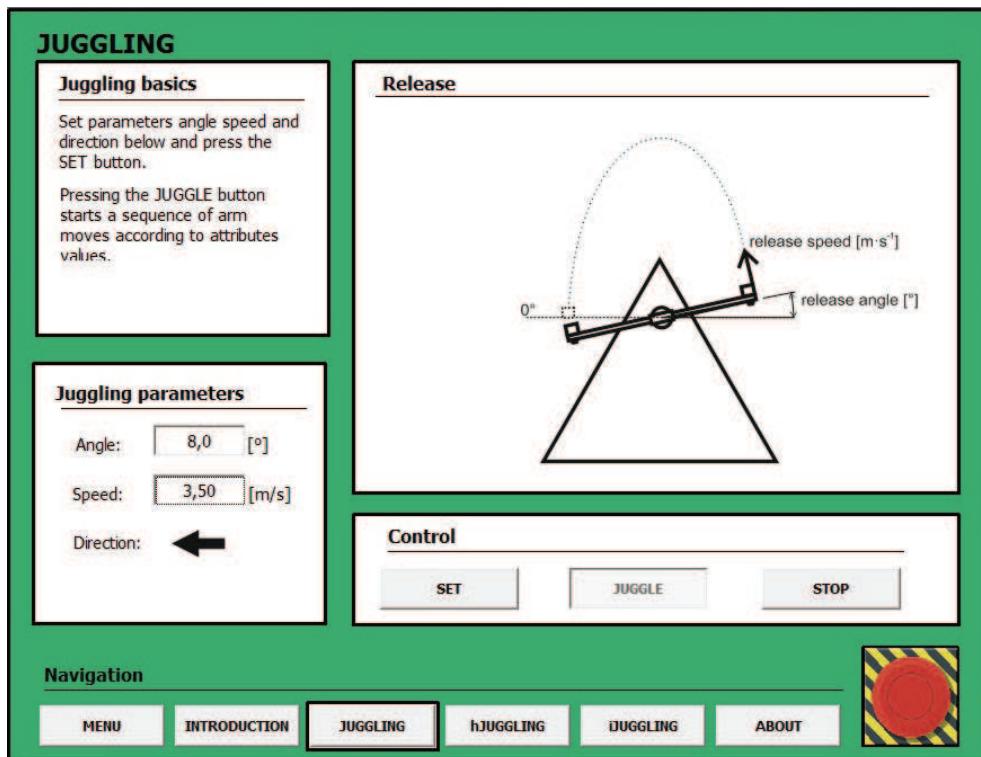
Obecný popis

JUGGLING (Obrázek 28) je první funkční obrazovka, na které lze nastavit parametry pro jeden hod míčku. Na obrazovce má obsluha možnost nastavit hodnoty pro odhadovou rychlosť, odhadový úhel, směr hodu, stisknout tlačítko SET, JUGGLE a STOP.

Obsluha obrazovky je prostá. Zapíší se požadované hodnoty a vybere se směr. Dále je potřeba stisknout tlačítko SET, čímž se hodnoty potvrzdí. Stisknutím tlačítka JUGGLE se odstartuje sekvence pohybů ramene vedoucí k hodu míčku dle zadaných parametrů. Tlačítko STOP slouží k zastavení sekvence pohybů a nastavení ramene do výchozí nulové polohy.

Zadávání hodnot je realizováno tak, že po klepnutí na konkrétní editační okno se zobrazí numerická klávesnice. Prostřednictvím této klávesnice je možné hodnotu zapsat.

Po klepnutí na šipku se zobrazí nové okno s možností výběru směru. Směrem se myslí to, kterým směrem bude hod veden, tedy pokud je například potřeba vést hod zprava doleva je potřeba vybrat šipku směřující doleva. Výběrem směru se přepne i schématický obrázek, na kterém je znázorněno, kterým směrem se bude házet a jak je definována odhadová rychlosť a odhadový úhel.



Obrázek 28: Ovládací obrazovka – JUGGLING

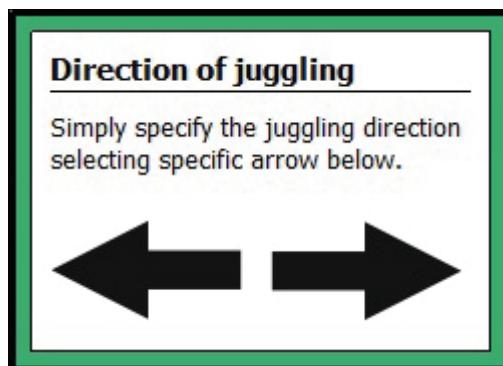
Zdroj: autor práce, 2012

Popis funkčních ovládacích prvků

Zápis do editačních editačních oken popsaných „Angle“ a „Speed“ je proveden stejným způsobem jako u obrazovky SERVICE prostřednictvím objektů Hot Region, které spustí numerickou klávesnici a zapíší hodnoty do proměnných *ILC150_rVpAngleRelease* a *ILC150_rVpSpeedRelease*. Hodnoty těchto proměnných jsou příslušnými editačními okny zobrazeny. Klepnutím na Hot Region v tomto případě provede i další akci: přiřazení hodnoty FALSE proměnné *Variable – Set – juggleBLOCK*, čímž se deaktivuje tlačítko JUGGLE.

Klepnutí na obrázek šipky se spustí příkaz *Screen Open frame (multi-monitor) – direction* a následně se zobrazí nové okno (Obrázek 29). Zde je možno vybrat směr hodu klepnutím na šipku ukazující vybraným směrem. Výběr směru znamená, že v případě klepnutí na levou šipku se proměnné *LC150_direction*, *arrow_left_button* příkazem *Variable – Set* nastaví na hodnotu TRUE a proměnné *arrow_right_button* a *juggleBLOCK* se nastaví stejným způsobem na FALSE. Klepnutím na tuto šipku se také zároveň provede zavření okna (*Screen Close and Return Back – direction*). Pro klepnutí na šipku směřující doprava se spustí stejně příkazy, ale proměnným jsou, vyjma proměnné *juggleBLOCK*, přiřazeny opačné logické hodnoty.

Proměnné *Arrow_left_button* a *Arrow_right_button* jsou přiřazeny k dynamické vlastnosti *Visible* u obrázků šipek a schématického obrázku „Vypuštění“.



Obrázek 29: Ovládací obrazovka – Direction of juggling

Zdroj: autor práce, 2012

V případě, že je proměnná logická pravda, obrázek je vidět. V opačném případě nikoliv. Díky tomuto provedení je vždy na první pohled patrné, jakým směrem bude hod veden.

V panelu označeném „Control“ se nacházejí tři ovládací tlačítka. Tlačítko SET nastaví proměnným *ILC150_setCMD*, *ILC150_mode1CMD* a *juggleBLOCK* hodnotu TRUE. Tímto se později v programu PC WORX běžícím na řídícím PLC určí, že bylo stisknuto tlačítko SET pro mód Juggling a převezmou se hodnoty vyplněných parametrů úhlu, rychlosti a směru.

Nastavením *juggleBLOCK* na TRUE se zpřístupní tlačítko JUGGLE. Stisknutím tohoto tlačítka dojde současně k nastavení *ILC150_mode1CMD* a *ILC150_juggleCMD* na TRUE. Hodnoty těchto proměnných jsou opět zpracovány programem běžícím na řídícím PLC. Podle hodnot potvrzených tlačítkem SET začne zařízení provádět sekvenci pohybů ramene vedoucí k hodu.

Poslední ovládací tlačítko STOP nastaví hodnotu *ILC150_stopCMD* na TRUE.

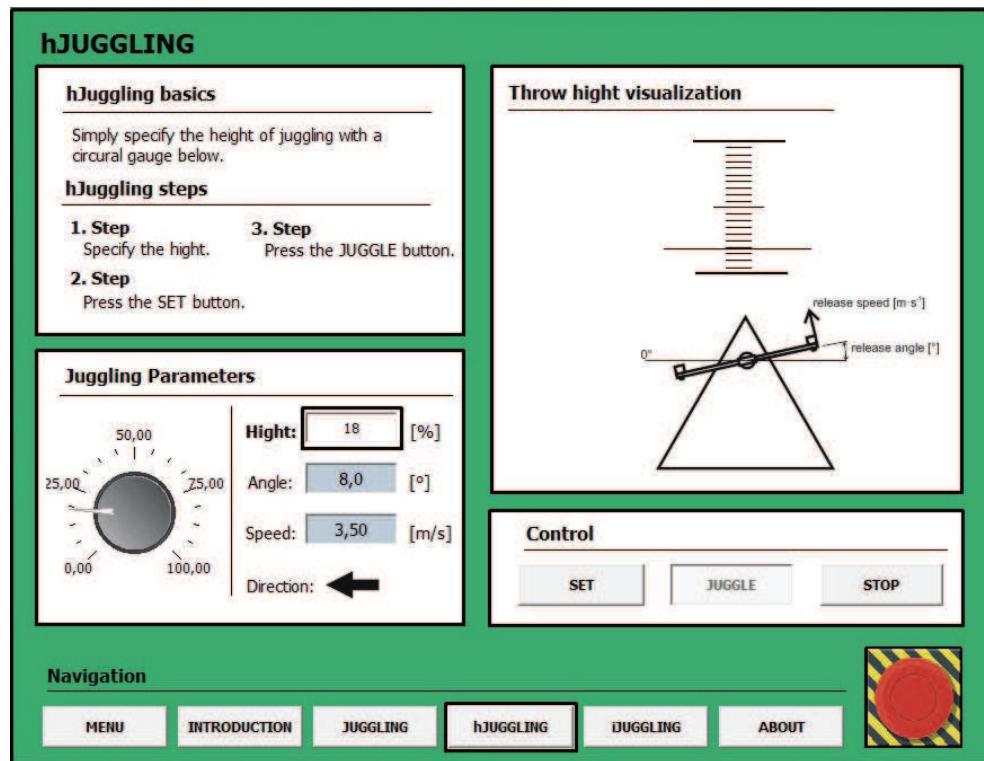
6.6 Obrazovka hJUGGLING

Obecný popis

Na obrazovce hJUGGLING (Obrázek 30), je možno volit výšku hodu míčku tak, aby byl druhou stranou ramene po odhodu chycen. Nejde o skutečnou volbu výšky hodu, nýbrž o procentuální hodnotu. Nula procent tu představuje nejmenší možnou výšku přehozu a naopak výška sto procent maximální výšku přehozu. Podle zvolené výšky se přepočítávají parametry rychlost a úhel, a také se mění poloha úsečky na stupnici v „Throw height visualization“.

Směr je možno měnit libovolně, stejně jako tomu bylo u předchozí popisované obrazovky JUGGLING. I v tomto případě se výběrem směru přepne i schématický obrázek znázorňující směr hodu. Měnit procentuální hodnotu lze pomocí kruhového jezdce (circural gauge) nebo přímo zapsáním hodnoty do editačního okna (označeného „Height“) pomocí numerické klávesnice. Po stisknutí tlačítka SET se pro

zvolenou výšku převezmou vypočítané hodnoty a lze přejít ke spuštění přehození tlačítkem JUGGLE.



Obrázek 30: Ovládací obrazovka – hJUGGLING

Zdroj: autor práce, 2012

Popis funkčních ovládacích prvků

V části obrazovky označené „Juggling parameters“ se nachází celkem pět funkčních prvků. Kruhovým jezdcem lze dotykem a tažením nastavit libovolnou hodnotu od 0 do 100. Tato hodnota je přiřazena proměnné *ILC150_iVpJuggleHeight*. S proměnnou je svázáno i editační okno „Height“, příslušný Hot Region překrývající toto okno a horizontální posuv úsečky na stupnici v části „Throw height visualization“. Posuv je zajištěn změnou hodnoty vlastnosti „Composed Movement“, podle zvolené výšky.

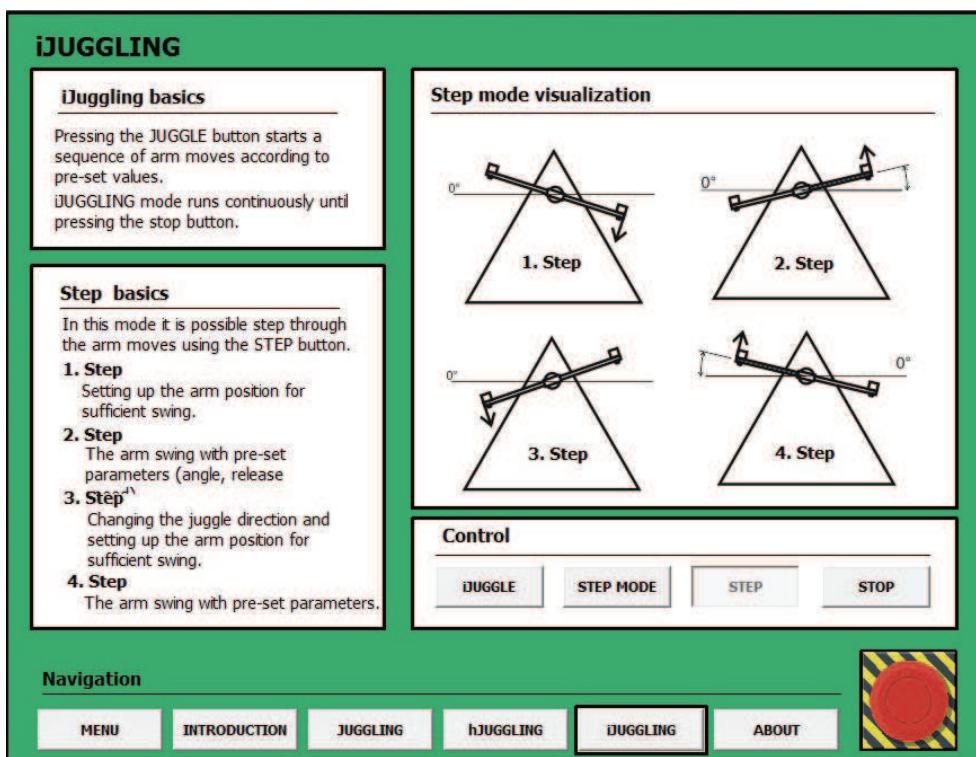
Další dvě okna slouží zobrazení hodnot proměnných *ILC150_rVpAngleRelease* a *ILC150_rVpSpeedRelease*, které jsou v programu PC WORX přeypočítány pro zvolenou hodnotu výšky. Stejným způsobem jako u obrazovky JUGGLE i zde lze měnit směr hodu. Shodně se zobrazuje odpovídající schéma hodu dle vybraného směru. Stejným způsobem pracují i tlačítka v části „Control“, až na jednu výjimku – tlačítko SET. Klepnutím na tlačítko SET se nastaví proměnná *ILC150_mode5CMD* na hodnotu TRUE. Tímto se v programu běžícím v PLC určí, že se jedná o mód hJuggling a provedou se další činnosti vedoucí k výpočtu parametrů rychlosti a výšky (viz kapitola Realizace řídícího programu).

6.7 Obrazovka iJUGGLING

Obecný popis

V módu iJuggling jsou hodnoty odhadové rychlosti a úhlu již přednastavené (Obrázek 31). Tlačítkem iJUGGLE je spuštěna sekvence pohybů vedoucích k přehození míčku z jedné strany ramene na druhou a opačně. Přehazování pokračuje neustále, dokud není stisknuto tlačítko STOP.

Prostřednictvím tlačítka STEP MODE je umožněno přepnout do režimu krokování. Následně lze pomocí tlačítka STEP krokovat sekvenci pohybů stejně jako po stisku iJUGGLE.



Obrázek 31: Ovládací obrazovka – iJUGGLING

Zdroj: autor práce, 2012

Sekvence pohybů, kterými lze krokovat:

1. Natočení ramene z výchozí polohy na „dokovací“ úhel.
2. Spuštění pohybu ramene, které vymrští míček dle přednastavených parametrů zprava doleva. Po odhození se rameno rovná zpět do výchozí polohy. Na opačnou stranu ramene dopadá míček.
3. Natočení ramene z výchozí polohy na „dokovací“ úhel, tentokrát na opačnou stranu.

4. Spuštění pohybu ramene, které vymrští míček dle přednastavených parametrů zleva doprava. Po odhození se rameno rovná zpět do výchozí polohy. Na opačnou stranu ramene dopadá míček.

Popis funkčních ovládacích prvků

Funkčními prvky jsou zde pouze tlačítka v části „Control“. Tlačítkem iJUGGLE se nastaví hodnoty *ILC150_mode2CMD* a *ILC150_juggleCMD* na TRUE a *juggleBLOCK* na FALSE. Funkce proměnné *juggleBLOCK* je shodná jako u předešlých dvou obrazovek. Rozdíl je v deaktivovaném tlačítku. Zde se deaktivuje tlačítko STEP. Dříve než je možné provést nastavení hodnot proměnných z tlačítka STEP, je potřeba aktivovat tzv. „Step mode“ klepnutím na tlačítko STEP MODE. Tímto se provede nastavení *ILC150_mode3CMD*, *ILC150_setCMD* a *juggleBLOCK* na TRUE, které zpřístupní tlačítko STEP a pro program v PLC určí, že se jedná o mód iJuggling. Tlačítkem STEP se nastavují proměnné *ILC150_mode3CMD* a *ILC150_stepCMD*, které také zpracovává program v PLC (viz kapitola Realizace řídícího programu).

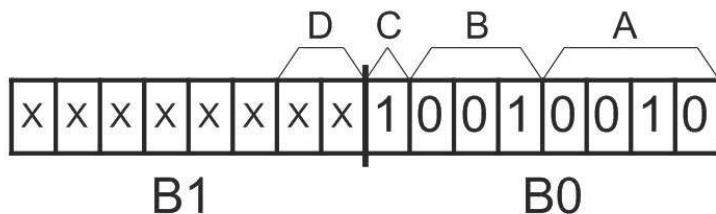
7 Realizace řídícího programu

Hlavní řídící program je vytvořen v prostředí PC WORX v kooperaci s prostředím VISU+. Tento program operuje na řídícím PLC ILC 150 ETH.

Operátor prostřednictvím OPC proměnných zajišťuje komunikaci mezi řídícím PLC a dotykovým panelem. Přebírané hodnoty z OPC proměnných nadále zpracovává. Takto zpracovaná data, v určité datové struktuře, prostřednictvím speciálního funkčního bloku, zasílá druhému automatu řešícímu přímé řízení pohybů ramene žonglujícího stroje.

Datová struktura pro předávání hodnot mezi automaty je složena ze tří proměnných. První parametr *speedRelease* reprezentuje rychlosť otáčení ramene, druhý parametr *angleRelease* reprezentuje uhel ve kterém má být míček odhozen. Třetí a nejdůležitější ze zasílaných parametrů je řídící slovo *mode*.

Prostřednictvím programu spuštěném na řídícím automatu se slovo *mode* mění na základě stisknutých tlačítek na dotykovém panelu. Pomocí tohoto slova jsou druhému automatu předávány informace o tom, jaké bylo stisknuto tlačítko, v jakém módu provozu se má operovat, o jaký se jedná směr házení a zasílá údaj o přepnutí kroku (v módu krovkání).



Obrázek 32: Část řídícího slova *mode*
Zdroj: autor práce, 2012

Na Obrázku 32 je znázorněn ukázkový příklad pro hod míčku, při využití prvních dvou bytu řídícího slova (druhý a třetí byte slova není pro řízení použit), kde jednotlivé bloky A, B, C a D představují:

- A – stisknutí tlačítka JUGGLE
- B – spuštění v módu provozu Juggling
- C – směr házení probíhá zprava doleva
- D – blok D není vyplněn, protože se nejedná o režim krovkání

Při odeslaní výše definovaného vzorového příkazu provede automat přímo zadaný trik.

Řídící slovo je rozděleno do bloků (Tabulka 1). Hodnoty v jednotlivých blocích jednoznačně určují chování žonglování.

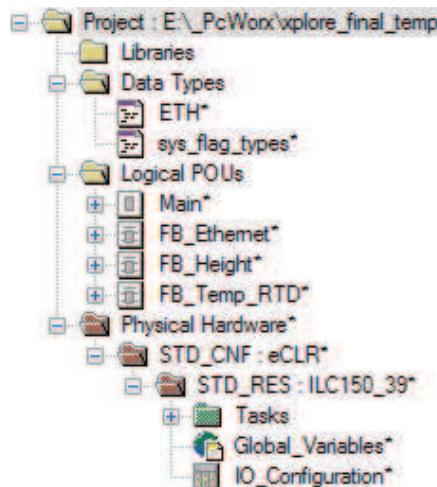
Tabulka 1: Rozdělení řídícího slova do bloků

Označení bloku	Hodnota bin	Hodnota dec	Popis
A – stisk tlačítka	0001	1	Set
	0010	2	Juggle
	0011	3	Stop
	1100	14	Reset Emergency Stop
	1110	15	Emergency Stop
B – mód provozu	001	1	Juggling
	010	2	iJuggling
	011	3	iJuggling + Step mode
	111	7	Service
C – směr házení	0	0	Right
	1	1	Left
D – krok	01	1	Step 1
	10	2	Step 2

Zdroj: autor práce, 2012

7.1 Struktura programu

Projekt se skládá ze tří funkčních bloků (dále také „FB“) a hlavního programu (dále jen „Main“). Dále je zde pro potřeby komunikace vytvořen vlastní datový typ, pojmenovaný stDataETH, který slouží k předávání hodnot mezi PLC.



Obrázek 33: Struktura programu

Zdroj: autor práce, 2012

Jak je z následující části kódu patrné, jedná se o strukturu sloučenou ze tří, výše již zmíněných, proměnných datového typu DWORD (double word).

```

TYPE
    stDataETH:
    STRUCT
        mode :      DWORD;
        angleRelease : DWORD;
        speedRelease : DWORD;
    END_STRUCT;
END_TYPE

```

V programu Main je nejprve provedeno navázání síťové komunikace s druhým automatem pomocí předdefinovaného funkčního bloku IP_CONNECT. Pokud je skutečně komunikace navázána provádí se další příkazy.

Pokud byl v uživatelském rozhraní na obrazovce hJUGGLING aktivován mód vrhání dle výšky, potom je proveden funkční blok FB_Height_1 a jeho výstupní parametry jsou přiřazeny hodnotám rychlosti a úhlu.

Dále je proveden převod hodnot rychlosti a úhlu. Z uživatelského rozhraní na dotykovém panelu jsou převzaty hodnoty úhlu odhozu ve stupních [°] a hodnotu rychlosti odhadu v [m/s], nicméně pro potřeby druhého PLC jsou hodnoty převedeny, dle následujících vzorců (kód programu), ze stupňů na počet impulsů inkrementálního snímače a z rychlosti v [m/s] na rychlosť v [ot/min] ramene.

```

(*-----Prep. rychlosť z [m/s] na [ot/min] ----- *)
dwSpeedReleaseRPM:=
REAL_TO_DWORD(((rVpSpeedRelease/rArm)
                * (30.0/rPI))
                * rGear);
(*----- Prep. uhlo v [stup.] na [impulz]----- *)
dwAngleReleaseIMP:=
REAL_TO_DWORD((rVpAngleRelease*rImpuls)/360.0);

```

Dalším funkčním blokem, ke kterému se přistupuje v rámci programu Main je FB_Ethernet_1, který má jako vstupní hodnotu identifikaci síťového připojení IP_CONNECT_1.ID a dále ony přepočítané hodnoty úhlu a rychlosti.

Posledním funkčním blokem s kterým se v projektu operuje je FB_Temp_1. Tento blok slouží pro vyčítání teploty měřené pomocí teplotního modulu.

7.2 Předdefinované funkční bloky pro komunikaci

7.2.1 IP_CONNECT

Tento FB slouží pro navázání spojení mezi dvěma zařízeními tzv. partnery, má dva vstupní a čtyři výstupní parametry.

Vstupními parametry jsou EN_C a PARTNER. Pokud se nastaví EN_C na logickou hodnotu TRUE pokusí se blok vytvořit spojení se zařízením s IP adresou

uvedenou v parametru PARTNER. Aby skutečně došlo ke spojení, musí být tento FB volán na obou zařízení. IP adresa musí být přiřazena tímto způsobem PARTNER='IP=xxx.xxx.xxx.xxx', kde x představují číslice. FB potom vypadá následovně IP_CONNECT_1(EN_C:=TRUE,PARTNER:=sEthernet_IP), kde sEthernet_IP je proměnná naplněná hodnotou '/IP=192.168.0.72', která představuje IP adresu druhého PLC, se kterým má být vytvořeno spojení.

Pokud dojde k navázání komunikace a je stále aktivní, z výstupního parametru VALID se zápisem IP_CONNECT_1.VALID získá hodnota TRUE. Výstupní parametr ID potom jednoznačně identifikuje, konkrétní připojení unikátním číslem (IP_CONNECT_1.ID). Dalšími výstupními parametry jsou ERROR a STATUS. V případě chyby se na výstupu ERROR získá hodnota TRUE a z výstupu STATUS lze vyčíst kód chyby.

Tabulka 2: IP_CONNECT

Název	Datový typ	Tok dat	Popis
EN_C	BOOL	Vstup	Jestliže se vstup nastaví na TRUE, potom se blok pokusí vytvořit spojení se zařízením s IP adresou zadanou na vstupu PARTNER. Nicméně, spojení lze vytvořit pouze pokud je FB volaný u obou komunikujících partnerů. Pro hodnotu TRUE bude spojení navázáno (ponecháno), v případě FALSE ukončeno.
PARTNER	STRING	Vstup	Vstup specifikuje, se kterým systémem se bude komunikovat, a to zadáním IP adresy partnera. Během operací s funkčním blokem nesmí být měněn řetězec, který ho identifikuje.
VALID	BOOL	Výstup	Výstup ukazuje, zda je navázáno platné spojení. Dokud připojení existuje je hodnota výstupu TRUE.
ID	INT	Výstup	Parametr identifikuje unikátním číslem navázанé spojení. Připojení může být provedeno současně více. Pro odlišení, o které se jedná, je zde právě výstup ID. ID je stále platné pokud parametr VALID nabývá hodnoty TRUE.

Zdroj: <http://eshop.phoenixcontact.cz>

7.2.2 IP_USEND

Pokud je prostřednictvím funkčního bloku IP_CONNECT navázané spojení, lze pomocí funkčního bloku IP_USEND zahájit přenos dat.

Při vytváření spojení užitím funkčního bloku IP_CONNECT, se pro každé navázané spojení vytvoří unikátní číslo ID. Toto číslo, lze přímo převzít na vstup funkčního bloku IP_USEND označeného „ID“, čímž se řekne, že zařízení s tímto ID budou přenášena data.

Vlastní posílání dat je prováděno pomocí vstupního parametru REQ. IP_USEND musí detektovat na tomto vstupu vzestupnou hranu respektive logickou hodnotu TRUE.

Přenos dat je zajištěn parametrem SD_1, který může být libovolného datového typu. Pro potřeby řešeného projektu se jedná o vlastní datový typ stDataETH, který je popsáný výše. Zda byla data úspěšně odeslána se zjistí tak, že výstup IP_USEND.DONE obsahuje hodnotu TRUE. Proto, aby byla data úspěšně odeslána, musí být na straně příjemce správně použit funkční blok IP_URCV, který slouží pro příjem dat.

Tabulka 3: IP_USEND

Název	Datový typ	Tok dat	Popis
REQ	BOOL	Vstup	S náběžnou hranou přivedenou na vstup se spustí požadavek na přenos dat. FB je po dobu přenosu dat uzamčený a nejde přerušit ani restartovat.
ID	INT	Vstup	Během navázání komunikace pomocí funkčního bloku IP_CONNECT je každému navázanému spojení přiřazeno unikátní číslo <i>ID</i> . Toto <i>ID</i> informuje blok IP_USEND, na které zařízení má data přenášet. Na tento vstup je vhodné připojit výstup <i>ID</i> funkčního bloku IP_CONNECT.
DONE	BOOL	Výstup	Výstup ukazuje, zda blok přenesl úspěšně všechna data.
SD_1	ANY	Vstup/Výstup	Velikost proměnné odeslané z výstupu SD_1 musí být stejná jako proměnná přijatá na vstupu RD_1 funkčního bloku IP_URCV. Velikost a jméno proměnné v SD_1 nesmí být během používání bloku měněna.

Zdroj: <http://eshop.phoenixcontact.cz>

7.2.3 IP_URCV

Tento funkční blok slouží pro příjem dat a příkazů. Musí být tedy použit v programu druhého PLC a být správně nastaven. Stejně jako funkční blok IP_USEND je i IP_URCV vložen do funkčního bloku FB_Ethernet a je rovněž závislý na hodnotě parametru ID bloku IP_CONNECT.

Aby bylo přijímání dat možné, musí mít vstupní parametr EN_R bloku IP_URCV hodnotu TRUE. Další nutné parametry jsou, kromě již zmíněného ID, parametry RD_1 a NDR.

Parametr RD_1 slouží k přenosu dat obdobně jako v případě IP_USEND parametr SD_1. Opět může být libovolného typu a opět, pro potřeby řešeného projektu, se jedná o typ stDataETH. Posledním využitým parametrem je NRD, který je obdobou parametru DONE funkčního bloku IP_USEND. Pokud je jeho hodnota TRUE byla data úspěšně přijata.

Tabulka 4: IP_URCV

Název	Datový typ	Tok dat	Popis
EN_R	BOOL	Vstup	Použitím se blok připraví na přijetí spojení. Pro hodnotu TRUE je přijímání dat umožněno, v případě FALSE je tomu naopak.
ID	INT	Vstup	Během navázání komunikace pomocí funkčního bloku IP_CONNECT je každému komunikačnímu spojení přiřazeno unikátní číslo ID. Toto ID informuje přijímací blok IP_URCV, od kterého partnera má přijímat data.
NDR	BOOL	Výstup	Výstup ukazuje, zda blok úspěšně přijal všechna data.
RD_1	ANY	Vstup/Výstup	Velikost proměnné přijaté na vstup RD_1 musí být stejná jako proměnná odeslaná z výstupu SD_1 funkčního bloku IP_USEND. Velikost a jméno proměnné v RD_1 nesmí být během používání bloku měněna. Dokud vstup EN_R nabývá hodnoty TRUE, je příjem dat umožněn. Jestliže byla zpráva úspěšně a kompletně přijata, tak se NDR nastaví na hodnotu TRUE na dobu jednoho cyklu. Během tohoto času lze s daty z RD_1 dále pracovat. Pokud je však NDR=FALSE, jsou hodnoty neplatné.

Zdroj: <http://eshop.phoenixcontact.cz>

7.3 Vlastní funkční bloky

Nejzásadnější je funkční blok FB_Ethernet, kde dochází ke zpracování povelů zadaných pomocí dotykového panelu. Další je FB_Height, který pro mód vrhání dle zadанé výšky, volí úhel a přepočítává rychlosť odhodu. Poslední funkční blok FB_Temp_RTD slouží pro vyčítání hodnot z tepelných snímačů.

7.3.1 FB_Ethernet

Nejdůležitější část PC WORX projektu, ve které se dle povelů předávaných pomocí OPC proměnných nastaví ovládací slovo *mode*, které se poté spolu s již převedenými hodnotami pro odhadový úhel *angleRelease* a pro odhadovou rychlosť *speedRelease* odešlou pomocí funkčního bloku IP_USEND druhému řízenému PLC.

FB má tři vstupní a tři výstupní parametry. Vstupními parametry jsou *ID*, *dwWriteAngleRelease* a *dwWriteSpeedRelease*. Do parametru *ID* se v programu Main přiřadí hodnota IP_CONNECT_1.ID, která jednoznačně identifikuje dané spojení s druhým PLC a je dále použita jako vstupní parametr pro funkční bloky IP_USEND a IP_URCV (způsobem popsaným v části o předdefinovaných funkčních blocích). Na vstupní parametry *dwWriteSpeedRelease* a *dwWriteAngleRelease* jsou v programu Main přivedeny přepočítané hodnoty převzaté z OPC proměnných nastavených prostřednictvím dotykového panelu.

Výstupními parametry jsou *dwReadMode*, *dwReadAngleRelease* a *dwReadSpeedRelease*. Slouží pouze pro kontrolu programátora, zda jsou data odeslaná v pořádku. Zjištěno je to tak, že z druhého PLC jsou přijaté hodnoty poslány zpět na řídící PLC.

Prvním krokem, který se ve funkčním bloku provádí, je vynulování struktury přijatých dat, aby později nedošlo k chybám při jejím porovnávání. Dále dojde k naplnění části datové struktury stDataETH ze vstupních parametrů rychlosti a úhlu. Přijaté hodnoty z této dvou parametrů jsou do datové struktury stDataETH přiřazeny dle níže uvedené části kódu programu. Proměnná *stWrite* je datového typu stDataETH.

```
(* hodnota úhlu a rychlosti ze vstupu na stDataETH *)
stWrite.angleRelease    := dwWriteAngleRelease;
stWrite.speedRelease    := dwWriteSpeedRelease;
```

Další část FB, je věnována převážně nastavování ovládacího slova (proměnné) *mode*. Nastavení se provádí v závislosti na logické hodnotě, konkrétních OPC proměnných. Například když se na některé z obrazovek vizualizace klepne (dále také „stiskne“) na tlačítko SET, dojde k nastavení hodnoty TRUE do proměnné *setCMD*. To dle části kódu (viz níže) způsobí, že podmínka je splněna a provedou se dané příkazy. V případě příkazů spouštěných tlačítka jde o nastavení prvních čtyřech bitů prvního bytu ovládacího slova mode (přesný popis složení slova v úvodu kapitoly).

```
(*---- Tlacitko nastavení parametru ---- *)
IF setCMD THEN
    stWrite.mode.X0 := TRUE;
    stWrite.mode.X1 := FALSE;
    stWrite.mode.X2 := FALSE;
    stWrite.mode.X3 := FALSE;
    xButton := TRUE;
END_IF;
```

Nastavení směru žonglování je definováno pouze hodnotou v posledním bitu prvního bytu a je provedeno příkazem *stWrite.mode.X7 := direction*; kde direction je proměnná typu BOOL nastavovaná pomocí zvolené šipky ve vizualizaci.

Dále se obdobně jako tlačítka nastavují tzv. módy. Jedná se opět, jako v případě příkazů pro tlačítka, o bitové proměnné nastavované na hodnotu TRUE. Podobně jako tomu je u příkazů spouštěnými tlačítky i zde dochází k nastavování konkrétních bitů řídícího slova *mode*. Pokud se například klepne na tlačítko SET na obrazovce JUGGLING dojde současně k nastavení *setCMD* i *mode1CMD* na hodnotu TRUE, čímž se splní podmínka a dojde tak přiřazení hodnot do konkrétních bitů. Rozhoduje tedy nejen to, jaké tlačítko bylo stisknuto, ale současně i to, na které obrazovce (v některých případech i ve které části obrazovky) bylo tak skutečně učiněno.

```
(*---- Mod hod ---- *)
IF mode1CMD THEN
    stWrite.mode.X4 := TRUE;
    stWrite.mode.X5 := FALSE;
    stWrite.mode.X6 := FALSE;
    xMode := TRUE;
    step := 1;
END_IF;
```

Dle předešlé kapitoly, je možné ovládat stroj ze třech hlavních obrazovek JUGGLING, hJUGGLING a iJUGGLING a z obrazovky SERVICE. Pro rozlišení způsobu ovládání ramene žongléra slouží již zmiňované módy. Módy jsou celkově čtyři, označené „Juggling“, „iJuggling“, „iJuggling + Step mode“ a „Service“.

Na obrazovkách JUGGLING a hJUGGLING je současně k proměnným, představující stisk tlačítka SET a JUGGLE, přidáno nastavení *mode1CMD* na hodnotu TRUE, čímž se definuje, že se jedná o jeden přehoz.

Na obrazovce iJUGGLING existují dvě možnosti ovládání stroje. První možností je spuštění režimu iJuggling klepnutím na tlačítko iJUGGLE, které nastaví proměnné *juggleCMD* a *mode2CMD* na TRUE, čímž si stroj začne přehazovat míček z jedné strany na druhou podle přednastavených parametrů. Druhou možností je aktivování režimu krování „iJuggling + Step mode“. Provede se tak stisknutím tlačítka STEP MODE, čímž se současně nastaví *setCMD* a *mode3CMD*. Následně je umožněno krování tlačítkem STEP. Stisknutím tohoto tlačítka se opět prove-

dou dva příkazy přiřazení hodnoty TRUE, v tomto případě pro proměnné *stepCMD* a *mode3CMD*. Krokování je umožněno nastavením ovládacího slova podle části kódu níže. Tak, že se po opětovném stisknutí tlačítka opakovaně mění hodnoty osmého a devátého bitu.

```
(*---- Tlacitko pro krokovani ----- *)
IF stepCMD THEN
    stWrite.mode.X0 := FALSE;
    stWrite.mode.X1 := TRUE;
    stWrite.mode.X2 := FALSE;
    stWrite.mode.X3 := FALSE;
    CASE step OF
        1: stWrite.mode.X8 := TRUE;
            stWrite.mode.X9 := FALSE;
        2: stWrite.mode.X8 := FALSE;
            stWrite.mode.X9 := TRUE;
            step := 0;
    END_CASE;
    xButton := TRUE;
    step := step + 1;
END_IF;
```

V poslední části funkčního bloku FB_Ethernet se provádí odesílání dat druhému PLC. Odesílání dat je provedeno funkčním blokem IP_USEND. K tomu aby měl parametr *mode* tvar (hodnotu) odpovídající požadavkům z vizualizace, je použito binárních proměnných *xReq*, *xButton* a *xMode*. Pokud by totiž nebylo podmínky (část kódu níže), mohla by při pokusu o současné nastavení proměnné pro tlačítka a proměnné pro mód nastat situace, kdy se provede přiřazení hodnot jen pro jednu z těchto proměnných. Pro tlačítka, která nepracují v závislosti na módu (např. tlačítko STOP), je tu proměnná *xReq*, která po nastavení na TRUE zajistí přenos blokem IP_USEND. U ostatních tlačítek je však použita proměnná *xButton* a u módů proměnná *xMode*. Jednoduchou podmínkou je potom zajištěno, odeslání dat z obou prvků současně a nedojde tak, ke ztrátě informace.

```
(*---- Odeslani dat druhemu automatu -----*)
IF NOT xReq THEN
    xReq := xButton AND xMode;
END_IF;
IP_USEND_1(REQ :=xReq & xEthLinkActive,
            ID :=ID,
            SD_1:=stWrite);
stWrite:=IP_USEND_1.SD_1;
```

Posledním použitým předdefinovaným funkčním blokem pro komunikaci je použit výše popsaný IP_URCV. Pomocí toho bloku jsou získána data zaslaná z druhého

PLC, která musí být stejná jako ta odeslaná pomocí IP_USEND z řídícího PLC, protože druhé PLC zasílá pro kontrolu data, která přijal.

7.3.2 FB_Height

Funkční blok, který v závislosti na zvolené procentuální hodnotě výšky, volí úhel odhadu *AngleRelease* a na základě fyzikálního principu šikmého vrhu počítá odhadovou rychlosť *SpeedRelease* tak, aby byl proveden úspěšný přehoz z jedné strany ramene na druhou. Tyto hodnoty jsou výstupními parametry FB. Jediným vstupním parametrem je *iJuggleHeight*. Na tento vstup je přivedena OPC proměnná *iVpJuggleHeight*, jejíž hodnota představuje výše zmíněnou požadovanou výšku.

V Main, je tento FB volán na základě logické hodnoty proměnné *mode5CMD*. Proměnná je svázána se stiskem tlačítka SET na obrazovce hJUGGLING. Pokud se tento FB zavolá, vypočítá hodnoty odhadového úhlu a rychlosti, které jsou přivedeny na výstup FB. Hodnoty těchto parametrů jsou poté v programu Main užitím následujícího příkazu přiřazeny příslušným OPC proměnným.

```
(*----- Funkcni blok FB_Height----- *)
IF mode5CMD THEN
    FB_Height_1(iJuggleHeight:= iVpJuggleHeight);
    rVpSpeedRelease := FB_Height_1.AngleRelease;
    rVpAngleRelease := FB_Height_1.SpeedRelease;
END_IF;
```

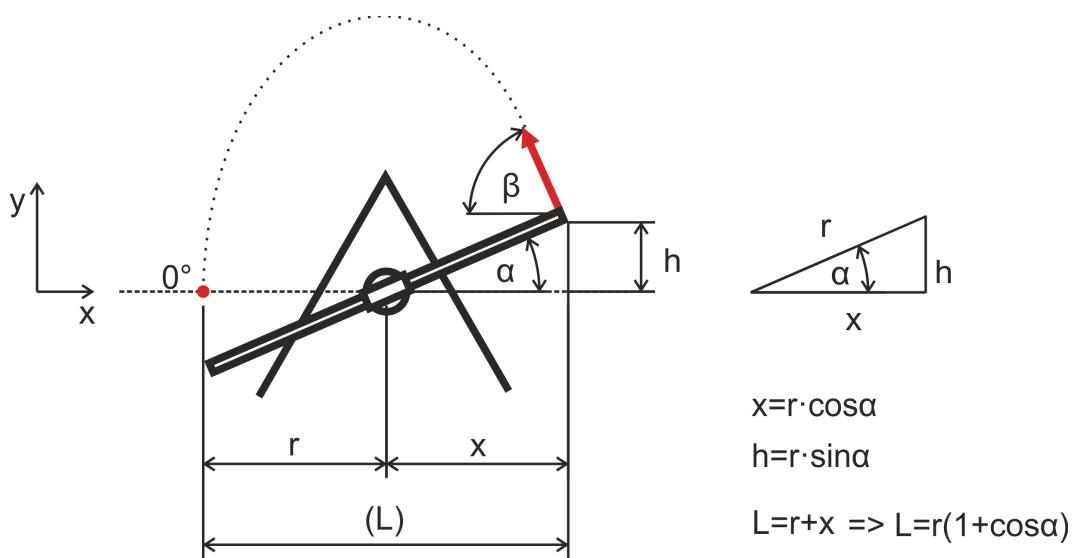
Funkční blok pracuje tak, že převeze hodnotu ze vstupu a pomocí příkazu *CASE*, podle kterého rozhodne jaký úhel se zvolí. Rozsah úhlů je vytvořen s ohledem na omezené dynamické možnosti pohybu ramene. Čím vyšší požadovaná výška, tím vyšší požadavek na odhadovou rychlosť. S vyšší rychlosťí roste případná chyba dopadu (nedojde k chycení druhou stranou ramene). Hodnotou vstupního parametru je vždy celé číslo v rozsahu od 0 do 100. Tento rozsah je v příkazu *CASE* rozdělen na osm částí (intervalů). V každé části je přidělen jiný požadovaný úhel odhozu, se kterým se dále počítá. Například pro 10 % maximální výšky přehazu bude úhel odhozu 14° .

Dále je třeba vyjádřit vzdálenost, na kterou má míček doletět. Je tak učiněno z požadovaného úhlu odhozu a poloměru ramene stroje. Pomocí zjištěné vzdálenosti a předvoleného úhlu odhozu je dle vzorce uvedeného níže vypočítána potřebná odhadová rychlosť.

Šikmý vrh

$$x_{max} = v_{0x} \cdot t \quad (13)$$

$$y = y_0 + v_{0y} - \frac{1}{2} \cdot g \cdot t^2 \quad (14)$$



Obrázek 34: Schéma vrhu míčku z nenulové výšky
Zdroj: autor práce, 2012

Vyjádření času dopadu

$$x_{max} = r \cdot (1 + \cos\alpha) \quad (15)$$

$$r \cdot (1 + \cos\alpha) = v_{0x} \cdot t \quad (16)$$

$$r \cdot (1 + \cos\alpha) = v_0 \cdot \cos\beta \cdot t \quad (17)$$

$$t = \frac{r \cdot (1 + \cos\alpha)}{v_0 \cdot \cos\beta} \quad (18)$$

Vyjádření počáteční rychlosti (odhodové)

Výška dopadu: $0 \Rightarrow y = 0$

Počáteční výška: $h \Rightarrow y_0 = h = r \cdot \sin\alpha$

Čas dopadu: $t \Rightarrow t = \frac{r \cdot (1 + \cos\alpha)}{v_0 \cdot \cos\beta}$

Dosazením výše uvedených hodnot do rovnice šíkmého vrhu

$$y = y_0 + v_{0y} \cdot t - \frac{1}{2} \cdot g \cdot t^2 \quad (19)$$

získáme rovnici

$$0 = r \cdot \sin\alpha + v_0 \cdot \frac{r \cdot (1 + \cos\alpha)}{v_0 \cdot \cos\beta} \cdot \sin\beta - \frac{1}{2} \cdot \left(\frac{r \cdot (1 + \cos\alpha)}{v_0 \cdot \cos\beta} \right)^2, \quad (20)$$

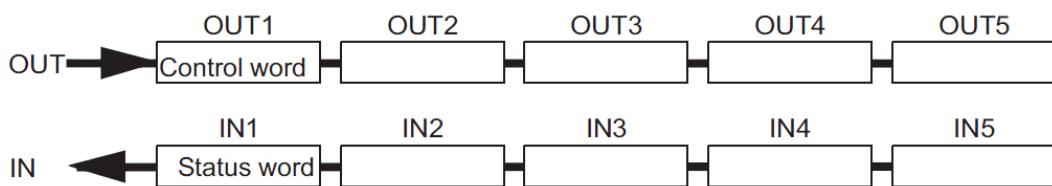
ze které následně vyjádříme počáteční (odhadovou) rychlosť

$$v_0 = \sqrt{-\frac{g \cdot r^2 \cdot \cos\beta \cdot (1 + \cos\alpha)^2}{\cos^2\beta \cdot (2 \cdot r \cdot \sin\alpha + 2 \cdot r(1 + \cos\alpha) \cdot \tan\beta)}}. \quad (21)$$

7.3.3 FB_Temp_RTD

Tento funkční blok slouží k nastavení ovládacích slov pro teplotní měřící kartu IB IL Temp 4/8 RTD – PAC tak, aby bylo možné vyčítat data z teplotních snímačů PT100.

Nejprve je nutné provést připojení zařízení do sběrnicové struktury a nastavení jeho vstupních a výstupních parametrů odpovídajícím proměnným.



Obrázek 35: Vstupní/výstupní slova pro obsluhu měřící karty
Zdroj: www.phoenixcontact.cz

Konfigurace karty se provádí pomocí výstupních parametrů $OUT1$ a $OUT2$. První z nich obsahuje příkaz a druhý parametry odpovídající tomuto příkazu. Příkazy se provádějí podle hodnoty nastavené v oblasti „Command code“, tedy kód příkazu. Těmito příkazy může být například příkaz načítání hodnot teploty z několika kanálů (pro vyčtení hodnot z prvních čtyřech kanálů se do $OUT1$ nastaví hodnota 0800hex), nebo příkaz ke konfiguraci celé karty (nastaví se 0600hex).

Výstupní parametr $OUT2$ zobrazený na Obrázku 37 definuje filter, způsob připojení (2/3), odpor teplotního snímače R0, rozlišení, formát a typ teplotního snímače.

Filter představuje z kolika naměřených vzorků se bude dělat střední hodnota, která se bude později zobrazovat. Způsob připojení uvádí zda je snímač připojen dvěma nebo třemi vodiči a rozlišením se uvádí na kolik desetinných míst se bude měření provádět a v jakých jednotkách ($^{\circ}\text{Celsia}$ nebo $^{\circ}\text{Fahrenheita}$).

OUT1																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Command code										0	0	0	0	0	0	0
0																

Obrázek 36: Výstupní sparametr OUT1

Zdroj: www.phoenixcontact.cz

OUT2																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Filter		2/3	R_0				Resolution		Format		Sensor type				
0																

Obrázek 37: Výstupní parametr OUT2

Zdroj: www.phoenixcontact.cz

V kódu funkčního bloku se nejdříve provede inicializace, kde dojde k naplnění kontrolních slov příslušnými hodnotami. V případě, že inicializace proběhla úspěšně, nastaví se řídícímu slovu *OUT1* příkaz, představující požadavek na čtení kanálů 1–4. Obdobným způsobem jsou načteny i kanály 4–8.

V programu Main jsou hodnoty z kanálů převzaty a nastaveny OPC proměnným. Ty jsou poté prostřednictvím vizualizace zobrazeny na display ovládací obrazovky ABOUT.

8 Závěr

Hlavním cílem diplomové práce „Inteligentní dvou-ramenný žonglovací stroj“ bylo ve spolupráci s druhým řešitelem úkolu realizovat celkovou koncepci stroje, zpracovat programové vybavení a ověřit funkčnost celého zařízení. Úloha je příkladem náročného řízení motoru pomocí programovatelného automatu. Principem je provádění sekvence pohybů ramene tak, aby docházelo k vrhání a následnému chytání míčku. V rámci této diplomové práce bylo žádoucí zajistit komunikaci mezi programovatelnými automaty a navrhnout vizualizaci ovládání stroje.

S tímto projektem jsme se přihlásili do soutěže pořádané společností Phoenix Contact. Netradiční řešení časově kritické úlohy řízení pohybu motoru, bez pomoci specializovaných zařízení na řízení pohybu (motion control), spolu s atraktivitou pohybu žonglování, atraktivním vzhledem zařízení a přehlednou intuitivní vizualizací zapůsobilo na komisi soutěže xplore New Automation Award 2012. Projekt nejen že byl do soutěže přijat a byl tak dotován částkou 3 000 Euro sloužící na nákup zařízení, ale dokonce postoupil i do finálového kola, kterého se účastnily týmy z celého světa. Finále probíhalo nedaleko hlavní pobočky firmy Phoenix Contact ve městě Lüdge v Německu. Boj o přední místa v soutěži probíhal několik dní, kdy účastníci převezli na místo konání svá zařízení, zprovoznili je a následně probíhaly prezentace a ukázky funkčnosti strojů. I přesto, že se řešený projekt neumístil mezi prvními třemi vítězi v kategorii, v konkurenci účastníků z 13 zemí světa mohli členové týmu čerpat od ostatních soutěžících zkušenosti a podněty pro další činnost v oboru.

Pro sestavení soutěžního zařízení byl nejprve nutný popis fyzikálních principů rotace ramene a šíkmého vrhu míčku. Na základě výpočtů podložených těmito principy byly zvoleny vhodné komponenty pro sestavení zařízení. Práce blíže specifikuje komponenty související s jejím zadáním, tedy s ovládáním, komunikací a vizualizací. Vybranými komponenty jsou jmenovitě: Dotykový panel TP 12T, Programovatelný automat ILC 150 ETH, Ethernet switch, IL Bluetooth Access Point – 2884758, IL TEMP a Senzor PT 100.

Dále práce řešila základy komunikace mezi programovatelnými automaty pomocí Ethernetu. Bylo nutno zavést datovou strukturu (stDataETH) složenou z proměnných rychlosti, úhlu a ovládacího slova *mode*. Na základě hodnoty slova *mode* je přímo dán chování zařízení.

Jednou ze stěžejních částí diplomové práce bylo vytvoření vizualizace v programu VISU+. Obsahem kapitoly Program VISU+ je popis základních přístupů k tvorbě uživatelského rozhraní (vizualizace). Vytvořená vizualizace je tvořena především ovládacími obrazovkami JUGGLING, iJUGGLING a hJUGGLING. Zmíněné obrazovky slouží pro zápis hodnot do proměnných, které jsou přes OPC server propojeny s programem běžícím na PLC. Programovatelný automat podle zadaných hodnot parametrů provede algoritmus a takto zpracovaná data zašle podřízenému PLC, které pomocí regulátorů přímo ovládá otáčení ramene.

Program vytvořený v prostředí PC WORX, běžící na řídícím automatu, zpracovává použitím třech funkčních bloků přijatá data a příkazy tak, že po jejich ode-

slání může podřízený automat vykonat žádanou činnost. Součástí programu v PC WORX jsou jak předem definované funkční bloky určené pro Ethernetovou komunikaci (IP_CONNECT, IP_USEND, IP_URCV), tak i funkční bloky vlastní. Mimo funkční bloky FB_Ethernet a FB_Height, které řeší ovládání stroje, byl vytvořen i FB_Temp_RTD, jehož úkolem je nastavování řídících slov měřící karty tak, aby byly získány hodnoty teplot ze dvou odporových snímačů k této kartě připojených.

Realizace stroje je shodně jako zajištění komunikace, řízení a ovládání úspěšně dokončena. Všechny cíle zadání byly splněny, veškeré programové vybavení je zpracováno a je docíleno bezproblémové komunikace mezi zařízeními.

Úloha svým širokým záběrem obohatila mé obzory a dosavadní profesní zkušenosti.



Obrázek 38: Členové týmu při účasti ve finálovém kole v Německu

Zdroj: <http://www.facebook.com/xploreAward>

9 Použité zdroje

Bluetooth [online]. Poslední revize 2012 [cit. 2012-04-16].

Dostupné z: <<http://www.bluetooth.com/Pages/about-technology.aspx>>.

Download-Center [online]. Poslední revize 2012 [cit. 2012-05-12].

Dostupné z: <<http://select.phoenixcontact.com/phoenix/dwlfr1.jsp?lang=en>>.

eshop Phoenix Contact [online]. Poslední revize 2012 [cit. 2012-05-12].

Dostupné z: <<http://eshop.phoenixcontact.cz>>.

facebook xplore [online]. Poslední revize 2012 [cit. 2012-05-20].

Dostupné z: <<http://www.facebook.com/xploreAward>>.

hw.cz [online]. Poslední revize 2012 [cit. 2012-04-16].

Dostupné z: <<http://www.hw.cz/teorie-a-praxe/dokumentace/mereni-teploty-kovove-odporove-senzory-teploty.html>>.

mobilmania.cz [online]. Poslední revize 2012 [cit. 2012-04-16].

Dostupné z: <<http://www.mobilmania.cz/default.aspx?article=1108570>>.

NOSKIEVIČ, P. *Modelování a identifikace systémů* Ostrava: Montanex, 1999. 276 s.
ISBN 80-7225-030-2.

Phoenix Contact [online]. Poslední revize 2012 [cit. 2012-05-12].

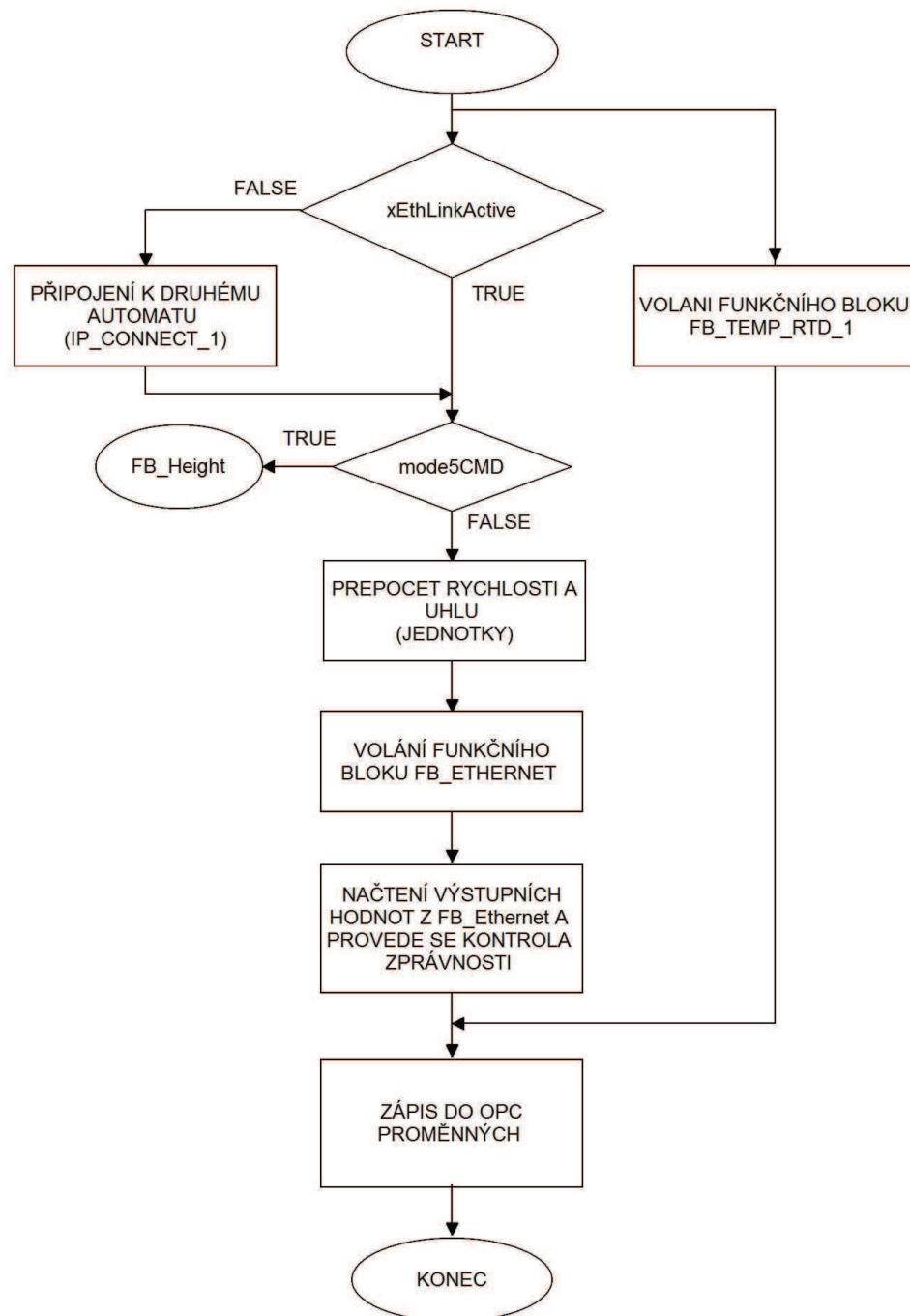
Dostupné z: <www.phoenixcontact.cz>

ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M. *Automatické řízení*. Brno: CERM, 2007.
324 s. ISBN 978-80-214-3491-2.

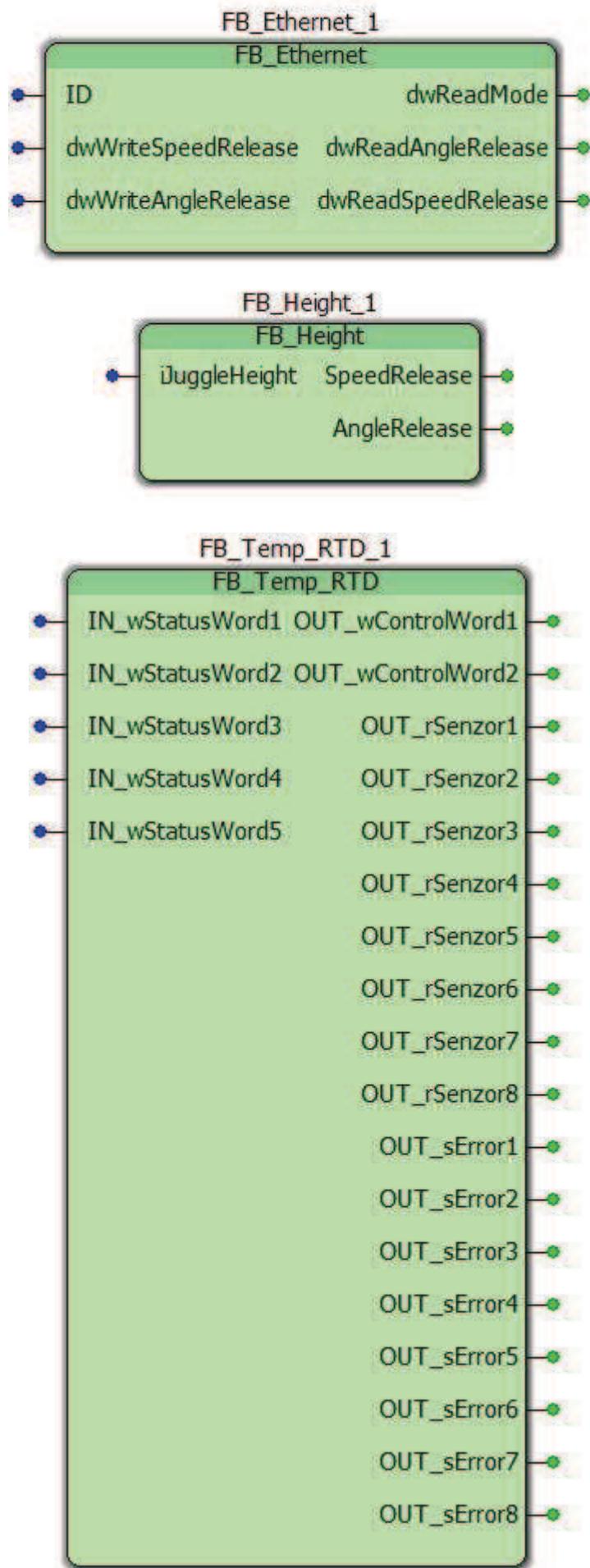
10 Přílohy

- Příloha 1 Vývojový diagram programu Main
- Příloha 2 Vlastní funkční bloky
- Příloha 3 Blokové schéma vývojového diagramu FB_Ethernet
- Příloha 4 Blokové schéma vývojového diagramu FB_Height
- Příloha 5 Blokové schéma vývojového diagramu FB_Temp_RTD

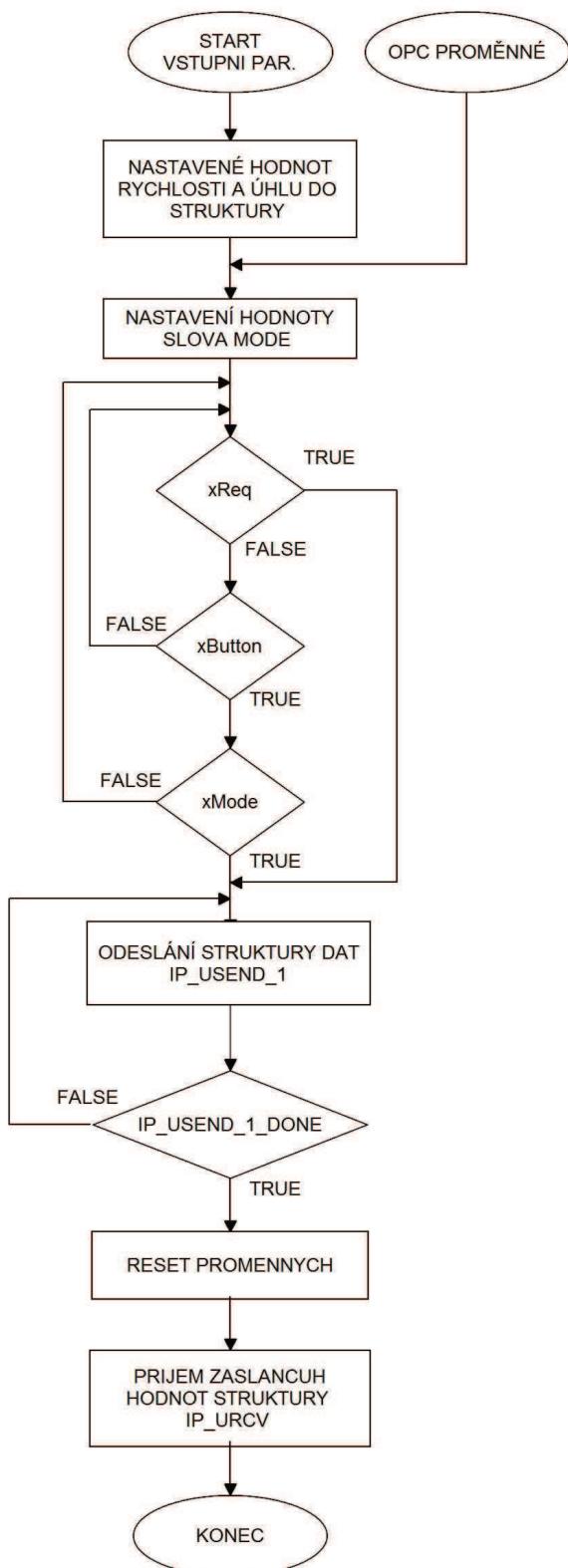
Příloha 1: Vývojový diagram programu Main



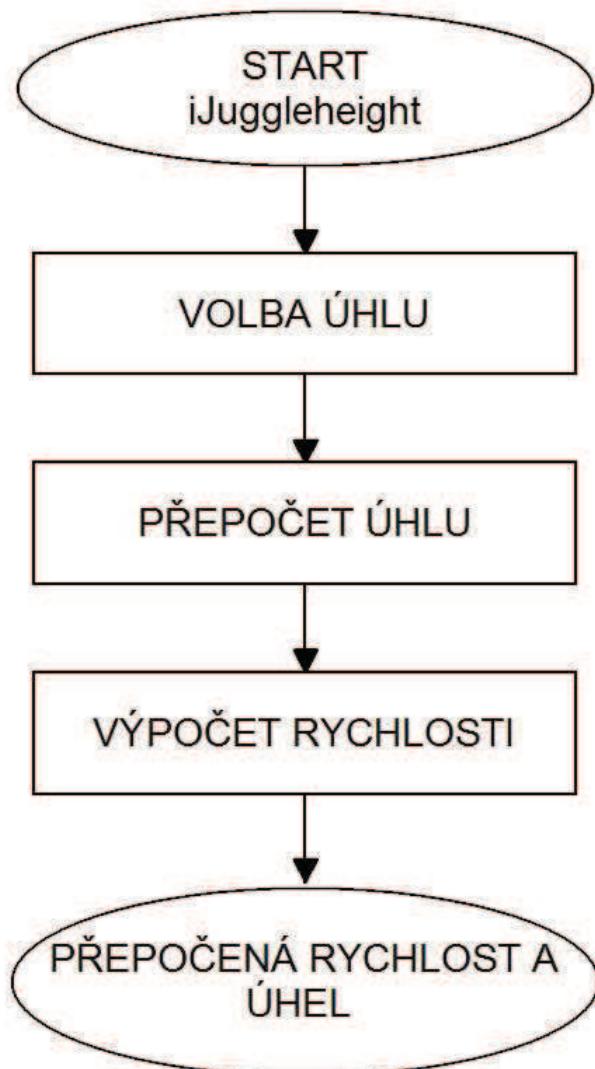
Příloha 2: Vlastní funkční bloky



Příloha 3: Blokové schéma vývojového diagramu FB_Ethernet



Příloha 4: Blokové schéma vývojového diagramu FB_Height



Příloha 5: Blokové schéma vývojového diagramu FB_Temp_RTD

