

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra aplikované elektroniky a telekomunikací

BAKALÁŘSKÁ PRÁCE

Návrh řešení automatické navíječky cívek

Originál (kopie) zadání BP/DP

Abstrakt

Předkládaná bakalářská práce se zabývá návrhem automatického navíjecího zařízení pro výrobu samonosných cívek pro elektrotechniku. Součástí práce bylo vytvoření mechanické konstrukce s vhodným hardwarem a vytvoření obslužného softwaru pro nastavení parametrů navíjení. V hlavní části bakalářské práce je řešena problematika řízení servopohonů pomocí programovatelně logického automatu. Dále je v práci uveden způsob ovládání zhotoveného programu pomocí dotykového panelu.

Klíčová slova

Navíjení, cívka, servomotor, servodriver, servozesilovač, PLC automat, modul, Mitsubishi, GX Works3, poziční tabulka, jog, automat, navíječka, dotykový panel

Abstract

The bachelor theses deals with the design of an automatic winding device for the production of self-supporting coils for electrotechnics. Part of the work was to create a mechanical construction with the appropriate hardware and to create a service software for setting winding parameters. The main part of the bachelor thesis deals with the issue of servo drive control using a programmable logic automaton. Further, the thesis describes the form of program controlling by the touch panel.

Key words

Winding, coil, servo motor, servo driver, servo amplifier, PLC automaton, module, Mitsubishi, GX Works3, position table, jog, automat, winder, touch panel

Prohlášení

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

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 5.6.2017

Josef Viktora

Poděkování

Tímto bych rád poděkoval společnosti P&V Elektronik spol. s.r.o. za veškerou materiálovou a finanční podporu. Dále bych rád poděkoval Ing. Tomášovi Burdovi a panu Františku Cifreundovi za cenné profesionální rady, připomínky k práci a poskytnutí softwaru k vytvoření programového řešení.

Obsah

ÚVOD

SEZNAM SYMBOLŮ A ZKRATEK

1	NAVÍJENÍ CÍVEK.....	10
1.1	RUČNÍ NAVÍJENÍ	12
1.2	AUTOMATICKÉ NAVÍJENÍ.....	13
2	NÁVRH MECHANIKY AUTOMATICKÉHO ZAŘÍZENÍ	14
2.1	MECHANICKÁ KONSTRUKCE	15
2.2	BLOKOVÉ SCHÉMA ŘÍZENÍ	16
2.3	ZVOLENÝ POHON.....	16
2.3.1	<i>Pohon vodítka</i>	<i>19</i>
2.3.2	<i>Pohon hřídele.....</i>	<i>19</i>
2.3.3	<i>Pneumatické zařízení.....</i>	<i>19</i>
2.4	ŘÍZENÍ A REGULACE	20
2.5	PLC.....	21
2.5.1	<i>Typy programovacích jazyků PLC automatů</i>	<i>22</i>
2.5.2	<i>CPU programovatelných logických automatů</i>	<i>24</i>
2.5.3	<i>PLC automat Mitsubishi MELSEC iQ-F series</i>	<i>25</i>
2.5.4	<i>Polohovací Simple Motion modul FX5-40SSC-S</i>	<i>26</i>
2.6	DOTYKOVÝ PANEL.....	27
3	NÁVRH PROGRAMOVÉHO ŘÍZENÍ	28
3.1	PROSTŘEDÍ GX WORKS3	28
3.1.1	<i>Programování v GX Works3</i>	<i>29</i>
3.2	PROSTŘEDÍ EASYBUILDER PRO	30
3.2.1	<i>Programování v EasyBuilder PRO.....</i>	<i>30</i>
4	ŘEŠENÍ PROGRAMU.....	31
4.1	MENU PROGRAMU.....	31
4.2	MANUÁLNÍ OVLÁDÁNÍ STROJE.....	33
4.2.1	<i>Joggování.....</i>	<i>34</i>
4.2.2	<i>Poziční data.....</i>	<i>34</i>
4.2.3	<i>Homování.....</i>	<i>36</i>
4.3	SPUŠTĚNÍ ULOŽENÉHO NAVÍJECÍHO PROGRAMU.....	38
4.3.1	<i>Inicializace.....</i>	<i>40</i>
4.3.2	<i>Načtení kroku.....</i>	<i>41</i>
4.3.3	<i>Identifikace typu modulu.....</i>	<i>42</i>
4.4	VYTVOŘENÍ / EDITACE NAVÍJECÍCH PROGRAMŮ.....	46
4.5	DALŠÍ FUNKCE.....	48
4.6	PROGRAMOVÉ MODULY.....	51
4.6.1	<i>Navíjecí modul</i>	<i>51</i>
4.6.2	<i>Koordinální modul</i>	<i>52</i>
4.6.3	<i>I/O modul</i>	<i>52</i>
4.6.4	<i>Resetovací modul</i>	<i>53</i>
4.6.5	<i>Opakovací modul</i>	<i>53</i>
5	ZÁVĚR.....	54
	SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ.....	56

PŘÍLOHY

Úvod

Náplní této bakalářské práce je návrh a realizace programového řízení navíječky cívek pro danou mechanickou konstrukci vyrobenou ve spolupráci s firmou P&V Elektronik spol. s.r.o..

Práce je rozdělena do dvou částí. V první části je popsána mechanická konstrukce zařízení, použité řídicí jednotky, pohony, jejich vlastnosti a možnosti jejich použití. Druhá část je věnována programovému řešení. Tato část obsahuje popis vývojových prostředí GX Works3 a EasyBuilder PRO. Dále je zde popsána struktura programu a postup programování. To obnáší např. nastavení parametrů PLC, Simple Motion modulu a servozesilovačů. V příloze v závěru práce je ukázáno výsledné řešení zhotoveného programu, které zprostředkovává práci s zobrazovacím zařízením. Část programu odpovědná za automatický chod zařízení je uložena na CD přiloženého k této práci.

Seznam symbolů a zkratk

<i>PLC</i>	Programovatelný logický automat
<i>PC</i>	Osobní počítač
<i>IL</i>	Instrukční list
<i>LD</i>	Ladder diagram
<i>FBD</i>	Diagram funkčních bloků
<i>ST</i>	Strukturovaný text
<i>SFC</i>	Sekvenční funkční diagram
<i>CPU</i>	Centrální procesorová jednotka
<i>HMI</i>	Human-Machine interface
<i>DI / DO</i>	Digitální vstupy / výstupy
<i>AI / AO</i>	Analogové vstupy / výstupy
<i>RAM</i>	Random Access Memory
<i>USB</i>	Universal Serial Bus
<i>PWM</i>	Pulse Width Modulation
<i>HP</i>	Home Position
<i>HPR</i>	Home Position Return
x_{set} / x_{act}	Hodnota pozice požadovaná / aktuální
n_{set} / n_{act}	Hodnota rychlost otáček požadovaná / aktuální
I_{set} / I_{act}	Hodnota proudu požadovaná / aktuální

1 Navíjení cívek

Cívka je na první pohled jednoduchá součástka, avšak její výroba již tak jednoduchá být nemusí. Při její výrobě se musí brát v úvahu hned několik parametrů, od jejích rozměrů až po tloušťku drátu a jeho izolaci. Podle účelu cívek jsou na ně kladeny určité nároky. Konstrukce a provedení jsou ve většině případů závislé na konkrétním zapojení elektrického obvodu. Kritériem například může být, zda se jedná o cívky navinuté na kostře, o samonosné cívky a nebo o cívky rozličných tvarů. Od jednoduchých válcových, přes anténní, až po cívky nestandardních tvarů. Tam, kde nelze dosáhnout žádaného tvaru cívky využitím tuhosti vodiče, lze přistoupit k technologii spékání. V takovém případě je zapotřebí speciálních vodičů, jež jsou opatřeny vrstvou zpevňovací hmoty. Ta za kontinuálního ohřevu vodiče umožní navinout díl v žádaném tvaru a po následném ochlazení fixuje dosažený tvar. Druhů cívek a jejich provedení je velmi mnoho, stejně tak způsobů navíjení, od ručního, až po plně automatické.

1.1 Ruční navíjení

Jedná se o způsob navíjení, při kterém operátor cívku navíjí manuálně pouze rukama. Počítá si navinuté závitě, případně, při jejich větším počtu, si jejich počet musí zapisovat. Tento způsob navíjení není vhodný aplikovat u cívek s velkým počtem závitů a např. u samonosných spěkaných cívek je tato metoda neproveditelná. Je vhodná pro navíjení cívek (zvláště toroidních) s malým počtem závitů (jednotky až desítky závitů) a s takovou tloušťkou vodiče, aby byl uživatel schopný utáhnout závitě vlastní silou.

Při připojení kostry nebo jádra cívky na hřídel s elektromotorem je navíjení značně urychleno. Dají se tak navinout cívky s větším počtem závitů, a pokud je k hřídeli připojeno počítadlo závitů, nemusí už závitě počítat operátor. Posun drátu zůstává v tomto případě stále na operátorovi.

1.2 Automatické navíjení

Pokud je předchozí metoda obohacena o podavač drátu a elektronické počítadlo, na kterém operátor nastaví cílový počet závitů, dostaneme nejjednodušší poloautomatickou navíječku. Nastavená hodnota na počítadle se při navíjení dekrementuje a po dopočítání do nuly zastaví motor. Při navíjení je důležité mít drát stále dostatečně napnutý. Přesně to zajistí podavač drátu. Zastavení motoru je tedy možné i při přetržení drátu pomocí snímače v podavači. Taková navíječka si vystačí s jednoduchou logikou bez použití jednočipových mikropočítačů. Posun drátu se zde však stále neřeší. S potřebou posunu drátu už jednoduchá logika nestačí. Pro pohyb takzvaného „vodítka“ drátu je zapotřebí druhý elektromotor. Vodítka slouží pro posuv drátu tak, aby nedocházelo k jeho navíjení na jednom místě, ale rozložilo jej rovnoměrně. Rychlost posuvu je třeba nastavit v závislosti na tloušťce drátu a zároveň sesynchronizovat s rychlostí otáčení hřídele. Je tedy nutné ovládat dva elektromotory.

U složitějších navíječek jsou možnosti v nastavení daleko větší. Uživatel si v programu jednoduše nastaví parametry navíjení (např. počet závitů, rychlost navíjení, posuv vodiče, atd.). Navíječka může obstarávat další funkce jako např. zapnutí/vypnutí různých mechanických zařízení. Takové navíječky již umožňují navíjet i toroidní cívky.

2 Návrh mechaniky automatického zařízení

Návrh zařízení, který je popisován v této práci, jsem vytvořil ve spolupráci s firmou P&V Elektronik spol. s.r.o.. Od zařízení je očekáváno plně automatické navíjení malých spékaných cívek. Parametry navíjení bude možné zadat do programu pomocí dotykového displeje. Po spuštění programu bude zařízení automaticky navíjet jednu cívku za druhou. Během navíjení bude na displeji zobrazován průběh navíjení. Tím se rozumí např. počet navinutých závitů právě navíjené cívky, celkový počet již navinutých cívek nebo aktuální pozice os.

Na trhu lze najít mnoho poskytovatelů servopohonů. Kvůli své ceně a podpoře všech potřebných funkcí této navíječky jsme společně s firmou P&V Elektronik spol. s.r.o. vybrali řídicí systém a systém servopohonů značky Mitsubishi. Samotný výrobce Mitsubishi nabízí velké množství servopohonů, v nabídce tedy bylo mnoho kombinací servozesilače a servomotoru. Srdcem zařízení je tedy PLC automat. Obsahuje integrovaný modul vstupů / výstupů a byl zvolen tak, aby jejich počet byl pro návrh dostatečný.

Pohyb hřídele obstarává střídavý servomotor o výkonu 750W. Ten je řízený výkonovým servozesilovačem, který je kompatibilní právě s tímto typem servomotoru. Pohyb vodítka je zajištěn méně výkonným střídavým servomotorem, který je rovněž řízen svým kompatibilním servozesilovačem. Tento servomotor má výkon 200W. Servozesilovače jsou propojeny optickou přenosovou komunikací s Simple Motion modulem. Simple Motion modul je připojený k PLC automatu a zprostředkovává řízení servomotorů. Zodpovídá např. za řízení otáček, řízení točivého momentu, synchronní řízení atd. Pro pohyb dodatečných mechanických částí k celkové automatickosti zařízení byl použit pneumatický systém na stlačený vzduch s lineárními pneumatickými válci.

Parametry programu navíjení bude možné zadávat na dotykovém LCD displeji. Během chodu programu zde budou zobrazovány aktuální informace o navíjení. Displej je spojen přes switch s PLC automatem pomocí sítě ethernet. Switch je zde použit pro možnost připojení k PC pro případ, že bude uživatel potřebovat nějak upravit/změnit hlavní program a nahrát jej do PLC.

2.1 Mechanická konstrukce

Mechanická konstrukce zařízení byla vytvořena tak, aby svou stavbou nejlépe vyhovovala použití pro plně automatické navíjení. Při návrhu a výrobě mechanické konstrukce byla nacházena inspirace u jiných automatických navíjecích strojů (např. u navíjecích automatů od společnosti Ingrid West Machinery Ltd. Hlavní konstrukce je vyrobena z hliníkových profilů (60 x 60 mm). Důležitým kritériem je, aby konstrukce byla pevná, stabilní a navíjení neovlivňovaly vibrace způsobené pohony. Vzhled sestrojeného zařízení lze nalézt v příloze [A] v závěru práce.

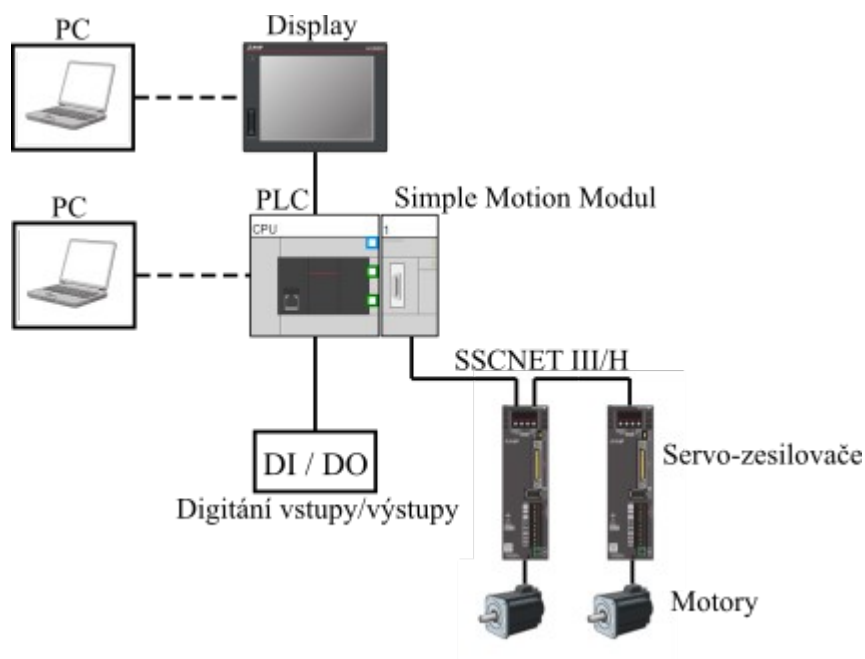
Větší 750-wattový motor pro pohyb hřídele je umístěn pod deskou „stolu“ konstrukce a pohyb je přenesen řemeny k navíjecí hřídeli. Menší 200-wattový motor je umístěn na vrchu konstrukce a pohybuje kuličkovým šroubem se stoupáním 4 mm. Kuličkový šroub převádí rotační pohyb na pohyb přímočarý. K matici tohoto šroubu je připevněné vodítko společně s jedním ze vzduchových pístů. Tento píst slouží pro zdvih drátu při navíjení. V programovém řešení je nutné se stoupáním šroubu počítat. Zadané hodnoty parametrů musí odpovídat pro správný posuv vodítka v milimetrech na otáčku.

O dodatečné mechanické pohyby je postaráno využitím vzduchových pístů. Ty obstarávají např. uchopení vodiče na začátku navíjení, jeho ustřižení při dokončení navíjení, vysunutí/zasunutí navíjecí hřídelky atd.

Celý chod stroje je následující: Před začátkem navíjení je vodič přidržen příslušným vzduchovým pístem a další píst obstará zasunutí hřídelky do navíjecího trnu. V momentě, kdy je tak učiněno, je zahájeno navíjení. Pohyb vodítka během navíjení zajišťuje rovnoměrné rozložení vodiče. Po navinutí požadovaného počtu závitů je pohyb hřídele a vodítka zastaven. Příslušnými vzduchovými písty je přestřižen vodič a vysunuta hřídelka z trnu. Tím cívka odpadne do zásobníku. Vodítko a osa hřídele přejedou do svých původních pozic a celý cyklus se může opakovat.

2.2 Blokové schéma řízení

Jak již bylo nastíněno v předchozí kapitole, pohyb hřídele a vodítka navíječky je realizován pomocí dvou servomotorů (HG-KN73BJ a HG-KN23J). Ovládání každého motoru zajišťují servozesilovače (MR-JE70B a MR_JE20B), kterým jsou posílány řídicí signály ze Simple Motion modulu (MELSEC FX5-40SSC-S) s využitím komunikace SSCNET III/H. Simple Motion modul je spojený s průmyslovým PLC automatem (MELSEC iQ-F Series) a efektivním způsobem doplňuje integrované polohovací funkce této jednotky.



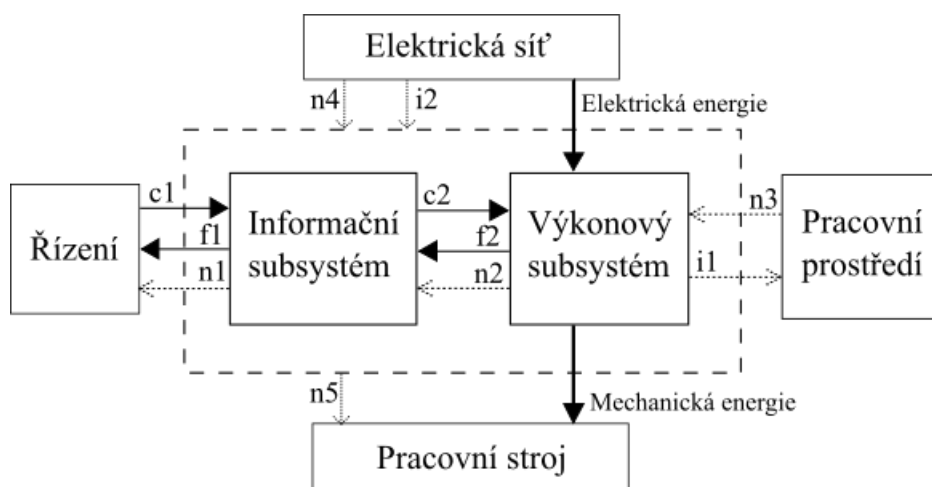
Obr. 2.1 Schéma řazení jednotlivých zařízení

2.3 Zvolený pohon

K pohybu hřídele a vodítka navíječky je použit systém elektrických servopohonů. V obecné terminologii je „Servo“ termín, používaný pro pohony aplikované v systémech s uzavřenou regulační smyčkou. Elektrický servopohon je regulační pohon, který je složen ze servomotoru a z pohonné – silové a řídicí elektroniky, která je často nazývána servozesilovač. Hranici mezi silovou částí a řídicí elektronikou je někdy obtížné určit, protože většina zesilovačů také obsahuje řídicí elektroniku. Na rozdíl od běžného elektrického pohonu, který může pracovat bez zpětné vazby tj. v otevřené regulační smyčce, je servopohon vždy zapojen se zpětnou vazbou v uzavřené regulační smyčce.

Servomotor obsahuje zpětnovazební snímače rychlosti a polohy např. enkodér. Ten je umístěn na hřídeli servomotoru. Signály z enkodéru jsou posílány zpět do servozsilovače, který tato data neustále kontroluje. Servozsilovač přijímá zadaný (příkazový) signál od řídicího systému, zesiluje jej a přenáší elektrický proud do servomotoru tak, aby vytvořil pohyb úměrný žádanému signálu. Zadaným (příkazovým) signálem může být například požadovaná rychlost, může ale také představovat točivý moment nebo polohu. Snímač připojený k servomotoru hlásí aktuální (skutečný) stav motoru. V regulačních obvodech se porovnává signál ze snímače o skutečném stavu motoru se stavem žádaným. Na základě rozdílu těchto stavů servozsilovač nastavuje motor do stavu požadovaného. Právě regulační parametry, tj. přesnost, rychlost regulace a jejich spolehlivost jsou dominantními požadavky na servopohony. Podmínkou pro pohon je umožnění čtyřkvadrantového provozu (otáčení do obou směrů a oba směry momentu).

Tak jako všechny pohonné systémy je i systém elektrických servopohonů popsán svou strukturou a vnitřními a vnějšími vazbami. Systém samotného servopohonu lze rozdělit na dva dílčí systémy: výkonový a informační. Do výkonového pod systému lze zařadit napájecí výkonové měniče a k nim příslušné obvody (transformátor, jisticí a spínací zařízení, atd.). Informační pod systém tvoří řídicí a regulační elektronické obvody. Mezi vnější pod systémy servopohonu patří: řízení tj. nadřazené komunikační a řídicí systémy, např. PC, mikročipové kontroléry, PLC automaty, ovládací panely apod., napájecí síť nebo jiný zdroj elektrické energie, poháněné zařízení (pracovní stroj) a pracovní prostředí. Do pracovního prostředí můžeme zahrnout např. teplotu okolí, vlhkost apod. Struktura servopohonu a jeho vnitřní a vnější vazby jsou zobrazeny na obr. 2.2.



Obr. 2.2 Komponenty asynchronního motoru

Vazby 'c1' a 'c2' představují řídicí signály a žádané hodnoty, např. polohy, rychlosti nebo točivého momentu. Signály zpětné vazby znázorňují 'f1' a 'f2'. Jsou to signály z čidel a snímačů a ostatní zpětná hlášení, např. provozní nebo poruchové stavy.

Negativní vazby jako např. rušivá vysokofrekvenční napětí a rušivá elektromagnetická pole způsobená spínáním výkonových prvků servopohonu představují 'n1' a 'n2'. Chvění, tepelné ztráty, hluk a další nepříznivé vlivy způsobené pohonem na pracovního prostředí a na pracovní stroj znázorňují 'n3' a 'n5'. Zpětné rušivé vlivy na napájecí síť značí 'n4'. Vnější negativní vlivy z okolního prostředí na servopohony jako např. teplota, vlhkost apod. 'i1'. Pokles napětí a přepětíové špičky jsou negativní vlivy napájecí sítě 'i2'.

Servopohony, používané v automatizačních systémech, požadují přesné ovládání pohybu, velice rychlou reakci na příkazové signály a přesné spolehlivé polohování. Nejčastěji používané servopohony jsou elektromechanické AC servo systémy. AC, samozřejmě, odkazuje na průběh střídavého proudu v servomotoru během provozu. Ze střídavých servomotorů jsou nejpoužívanější takzvané synchronní servomotory s permanentním polem. Tyto servomotory jsou několikanásobně výkonnější než jiné typy motorů o stejné velikosti. Jejich výkonné magnety zajišťují, že jsou mimořádně rychlé a přesné.

Samotný servozsilovač ale k řízení servomotoru nestačí. K jeho řízení potřebuje dostávat příkazové řídicí signály od nadřazeného systému. To obstarává takzvaný pohybový kontrolní systém (Motion control systém). Na vstupu do servozsilovače je kontroler, který poskytuje příkazové signály, které servomotoru říkají, kdy a jak se má pohybovat. Kontroler posílá příkazové signály do zesilovače ve formě analogových a nebo digitálních signálů. Samotný kontroler je přitom propojen k vyššímu operačnímu rozhraní. Rozhraní se používá k spuštění, zastavení a nastavení různých funkcí zařízení.

2.3.1 Pohon vodítka

Pro účely pohybu vodítka v tomto zařízení byl zvolen AC servomotor s označením 'HG-KH23J' a k němu kompatibilní servozeslovač 'MR-JE20B'. Jedná se o typ servomotoru s nízkým momentem setrvačnosti. Jmenovitý výstupní výkon servomotoru je 0,2 kW a jmenovitou rychlostí 3000 ot/min. Tento servomotor neobsahuje elektromagnetickou brzdu. Servomotor je vybaven optickým enkodérem s 131072 pulsy na otáčku. Umožňuje tedy lehký chod, velmi přesné polohování a rovnoměrné otáčení. To je pro účely vodítka velmi důležité. Nepředpokládá se, že se tento motor bude pohybovat ve vysokých otáčkách, právě naopak musí zajistit hladké a pomalé otočení. Všechny parametry servomotoru lze najít v příloze [B] a parametry servozsilovače v příloze [C].

2.3.2 Pohon hřídele

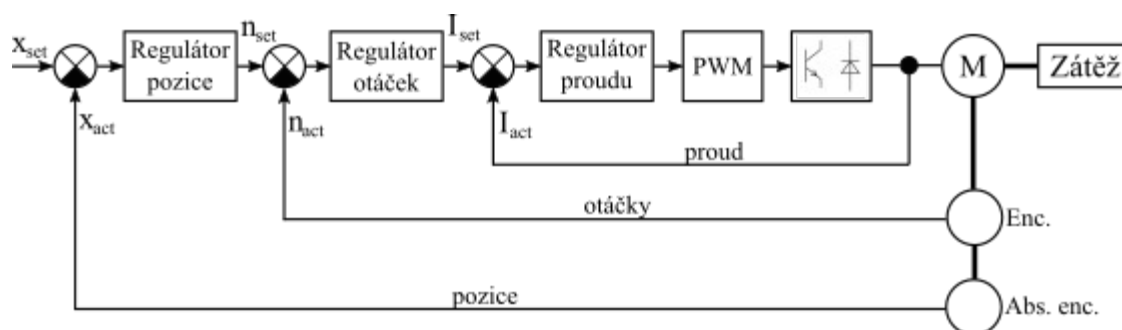
Pro pohyb hřídele byl zvolen servomotor typu 'HG-KN73BJ' a k němu příslušný servozsilovač 'MR-JE70B'. Jedná o servomotor s jmenovitým výstupním výkonem 750W se zabudovanou elektromagnetickou brzdou. Servozsilovače komunikují po optickém rozhraní 'SSCNET III/H'. Zde použitý servozsilovač ('MR-JE70B') umožňuje po tomto rozhraní propojení s dalším servozsilovačem v řadě. Tyto servozsilovače lze kombinovat s moduly pro jednoduché pohyby typu Simple Motion. Servomotor má opět zabudovaný optický enkodér s 131072 pulsy na otáčku. U tohoto servomotoru je počítáno s chodem ve vysokých otáčkách. Maximální jmenovitá rychlost otáčení je 5000 ot/min a přípustná okamžitá rychlost otáčení až 5750 ot/min. Svými rozměry je opoznáni robustnější než 200-wattová varianta. Maximální nosnost na hřídeli je 4kg.

2.3.3 Pneumatické zařízení

Dodatečné mechanické pohyby, potřebné k automatizaci navíječky jsou řešeny pomocí pneumatického systému. Konkrétně pomocí jednočinných lineárních pneumatických válců, které vyžadují přísun stlačeného vzduchu (kompresor). Aktivaci válců obstarávají elektronické ventily, kde ke každému válci přísluší právě jeden ventil. Ventily jsou ovládány signály z výstupů PLC automatu.

2.4 Řízení a regulace

Servopohony se používají v mnoha aplikacích k řízení polohy nebo rychlosti. Řídicí systém je obvykle konstruován v kaskádové sestavě tak, že se řídicí systémy vzájemně překrývají. Nejvnitřnější řídicí systém řídí proud. Pouze tento regulátor může řídit točivý moment. Překrytím regulátoru proudu pomocí uzavřené smyčky regulátoru otáček je umožněno řídit rychlost otáčení. Dalším překrytím regulátoru proudu pomocí regulátoru polohy je umožněno ovládat polohu. Řízení servopohonu je tedy složeno ze tří řídicích obvodů (smyček). Na obrázku 2.3 je zobrazena základní řídicí struktura obecného systému servopohonu. Jedná se pouze o obecný popis a nemusí tedy platit pro mnou zhotoveného zařízení, protože řídicí veličina se může lišit podle typu motoru.



Obr. 2.3 Struktura řízení servomotoru

1. Regulace proudu: Takzvaná proudová (momentová) smyčka ('proud') je vnitřní obvod, který řídí proudy do vinutí motoru. Pohybový účinek servopohonu je určen přímo točivým momentem motoru. Sám točivý moment je určen proudy v motoru. Pro dosažení požadované hodnoty točivého momentu co nejdynamičtěji je nutné nejen řídit napětí na motoru, ale také řídit proudy. Korekční proměnnou proudového regulátoru je napětí na motoru. Napětí motoru je specifikováno pulsně-šířkovou modulací (PWM) v časově diskrétní formě. S PWM modulací o kmitočtech jednotky až desítky kHz je proudová smyčka velmi rychlá, s dobou odezvy řádově 0.1-1 ms. Úkolem proudové smyčky je i ochrana motoru před přetížením a omezení maximálního momentu, kterým může působit motor na mechanickou soustavu.

2. Regulace rychlosti: Proudové smyčce je nadřazena rychlostní smyčka ('otáčky'). Zde je k detekci polohy a rychlosti využíváno různých druhů enkodérů. Podle dvou hlavních kritérií je můžeme rozdělit na enkodéry s absolutní informací o poloze a enkodéry s čistě přírůstkovou informací o poloze. Oba typy enkodérů však pouze poskytují informace o poloze. Z těchto informací určí řídicí systém servopohonu rychlost v pravidelných intervalech. Rychlost se vypočítá z rozdílu v poloze dvou vzorkovaných hodnot polohy přes známý časový rozdíl. Z toho lze usoudit, že rychlost není k dispozici jako okamžitá hodnota v určitém časovém okamžiku, ale spíše jako střední hodnota vzorkovacího intervalu. Hodnota rychlosti z enkodéru se porovnává s požadovanou hodnotou, která může pocházet z řídicího nebo polohovacího systému. Tato smyčka je pomalejší než proudová smyčka. Rychlostní smyčka má často integrující charakter, aby zajistila stabilní a přesné otáčky.

3. Polohová regulace: Regulátor polohy je většinou realizován jako regulátor typu P. Takzvaná polohová smyčka ('pozice') je nejpomalejší ze všech. Je nadřazená všem ostatním, proto musí počkat na výsledek změn v rychlostní smyčce, než může vydat pokyny pro hodnoty pozice. Skutečná hodnota z absolutního pozičního enkodéru je porovnávána s požadovanou hodnotou a tím je řízena proudová a rychlostní smyčka tak, aby byla dosažena minimální odchylka.

2.5 PLC

Protože k řízení navíječky je použit PLC automat, bylo by vhodné obecně přiblížit, co to PLC automat je a k čemu slouží. Zařízení PLC (Programmable Logic Controller), v češtině označované pojmem programovatelný logický automat, je v podstatě relativně malý průmyslový počítač sloužící k automatizaci, řízení a regulaci různých aplikací a technologických procesů (např. řízení strojů nebo výrobních linek). Od běžných počítačů se ale liší jednak cyklickým způsobem zpracování programu a jednak i tím, že jejich periferie jsou přímo uzpůsobeny pro napojení na technologické procesy. V tomto případě periferie tvoří analogové vstupy/výstupy (AI/AO) pro zpracování spojitých signálů a dále převážně digitální vstupy/výstupy (DI/DO) pro snadné připojení senzorů, spínačů, tlačítek a různých dalších přístrojů a zařízení.

Podle struktury můžeme PLC rozdělit na dvě skupiny. První skupinu tvoří takzvané „kompaktní“ (fixní) systémy s pevně danou hardwarovou strukturou. Jedná se o systém, kde jeden základní modul obsahuje CPU (Central Procesor Unit), paměti, digitální a analogové I/O a základní komunikaci. Rozšířitelnost struktury tohoto typu je omezena. Druhou skupinou jsou „modulární“ (flexibilní) systémy s modulárně upravitelnou hardwarovou strukturou. Komponenty jsou rozděleny do jednotlivých modulů. Celý systém PLC se potom skládá z modulů v podobě několika vzájemně propojených "krabiček". Od zdroje, CPU, vstupů/výstupů, paměťových modulů až po další funkční moduly (FM) např. pro polohování, komunikační procesory (CP) pro sběr a přenos dat a další specifické moduly podle výrobce konkrétního systému. Takovouto strukturu lze tedy s ohledem na systémové limity dále rozšiřovat, a to v daleko větším rozsahu než u kompaktních systémů.

Pro PLC je samozřejmostí možnost datové komunikace. Ta využívá zejména fyzických vrstev jako je RS232, RS485 nebo ethernet a průmyslových protokolů (DH+, profibus, atd.). Funkce celého PLC a připojených prvků je řízena uloženým uživatelským programem, který lze vytvořit pomocí výrobcem dodávaného vývojového softwaru pro běžná PC. Software obvykle umožňuje mimo programování i průběžnou grafickou simulaci a testování.

2.5.1 Typy programovacích jazyků PLC automatů

K vytvoření uživatelského programu nabízejí PLC systémy specializované jazyky, původně navržené pro snadnou, názornou a účinnou realizaci logických funkcí. Jazyky systémů různých výrobců nebyly vždy stejné a přenositelnost programů pro PLC byla proveditelná pouze mezi systémy téhož výrobce. Dnes jsou jazyky jednotlivých výrobců normovány a je tudíž možná přenositelnost např. u programů v jazyce ST. Pro psaní programů a algoritmů, které mají programovatelné automaty vykonávat, jsou využívány následující formy jazyků.

Instruction List - „IL“ - Seznam instrukcí nebo také „jazyk mnemokódů“ patří do skupiny textových jazyků. Označuje se také jako jazyk pokynů (povelů) připomínající Assembler (jazyk symbolických adres) u počítačů. Program je složen ze sekvence instrukcí, kde na každém řádku je pouze jedna instrukce. Každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka. Poskytuje tedy obvyklý aparát symbolického označení jako např. označení návěští, symbolická jména číselných hodnot a proměnných atd. Program vykonává instrukce jednoho řádku a pak se posune o řádek níž, vykoná další instrukci a takto pokračuje, dokud nedojde na poslední řádek programu. Dnes se již s jazykem IL často nesetkáme vzhledem k požadavkům na stále více rozsáhlejší aplikace, které snižují přehlednost a orientaci v kódu.

Ladder Diagram - „LD“ - Žebříkový diagram nebo také „jazyk kontaktních schémat“ je velice přehledný grafický jazyk. Základní logické operace jsou v programu zobrazené jako schémata obvyklá pro práci s reléovými a kontaktními prvky. Obvykle je na levé straně umístěna takzvaná napájecí sběrnice, ke které se připojují další prvky. Těmito prvky mohou být spínací a rozpínací kontakty, funkční bloky (např. čítače, časovače) a mnoho dalších prvků. Tento jazyk je pro svou přehlednost nezastupitelný při požadavku rychlé diagnostiky. Pokud ladicí prostředky dovolují zvýraznit na schématu "vodivou cestu", je nalezení závady na stroji otázkou několika minut.

Function Block Diagram - „FBD“ - Diagramů funkčních bloků nebo také „jazyk funkčních bloků“ je opět grafickým jazykem, který základní logické operace programu popisuje pomocí hradel (AND, OR a NOT) a funkčních bloků. Takto je sestaven celý program. Funkční bloky jsou stejné jako u jazyka LD. Výhodou FBD je přehlednost u rozsáhlejších zapojení. Jazyk funkčních diagramů vychází vstříc uživatelům, kteří jsou zvyklí kreslit logická schémata u zařízení s integrovanými obvody.

Structured Text - „ST“ - Neboli „jazyk strukturovaného textu“ je obdobou výkonných vyšších programovacích jazyků pro PC (např. Pascal nebo C). Syntaxe jazyka je dána povolenými výrazy a příkazy. Je definováno několik typů příkazů (přiřazení, vyvolání funkce, návrat, výběr apod.), které jsou navzájem odděleny středníkem. Může jich tedy, na rozdíl od „IL“, být více na jednom řádku. Jazyk ST umožňuje úsporný a názorný zápis algoritmů.

Sequential Function Chart - „SFC“ - Sekvenční funkční diagram popisuje sekvenční chování řídicího programu. Umožňuje systematicky rozložit úlohu řízení na realizovatelné části při zachování přehlednosti o chování celku. K popisu struktury je využíváno značek kroků (stavů), přechodů a větvení. Každý krok reprezentuje stav řízeného systému, k němuž je přiřazen blok akcí. Přechod je definován podmínkami, které musí být splněny, aby mohl být deaktivován krok předcházející přechodu, a naopak aktivován krok následující. Jazyk umožňuje i větvení programu. Chování v jednotlivých stavech nebo definování podmínek přechodů lze obvykle popsat kterýmkoli z dříve popsaných jazyků.

2.5.2 CPU programovatelných logických automatů

Centrální procesorová jednotka - „CPU“ (Central Processing Unit) je mozek celého PLC automatu. Skládá se z mikroprocesoru, paměťového prostoru a dalších integrovaných obvodů pro řízení logiky, monitorování a komunikaci. CPU realizuje soubor instrukcí a systémových služeb, zajišťuje základní komunikační funkce s vlastními i vzdálenými moduly, s nadřazeným systémem a s programovacím přístrojem. Paměťový prostor je obvykle rozdělen na dvě části. První část je určena pro uložení uživatelského programu, datových bloků a tabulek. Její obsah se zadává v edičním režimu a během vykonávání programu se obvykle nemění. U druhé části se jedná o takzvanou operační část paměti. V ní jsou lokalizovány veškeré uživatelské registry, čítače a časovače, obrazy vstupů a výstupů, komunikační, časové a jiné systémové proměnné (systémové registry).

V souboru instrukcí všech PLC automatů jsou vždy obsaženy instrukce pro základní logické operace s bitovými operandy. Těmi jsou např. operace jako logický součet a součin, negace, výlučný součet či jiné kombinační logické funkce. Dále např. instrukce pro realizaci paměťových funkcí a klopných obvodů, instrukce čítačů, časovačů, posuvných registrů, krokových řadičů a jiných funkčních bloků. Současné PLC nabízejí instrukční soubor ještě podstatně bohatší. U vyspělých PLC obvykle v souboru instrukcí nechybí instrukce pro aritmetiku a operace s čísly. Od nejzákladnějších, jako je např. sčítání, odčítání a porovnávání, až po výpočty s pevnou nebo plovoucí desetinnou čárkou. Nechybí ani logické instrukce s číselnými operandy a přenosy dat. Obvyklé jsou instrukce pro organizaci programu, např. skoky v programu, volání podprogramů a návraty. U některých PLC se můžeme setkat se specializovanými instrukcemi pro komplexní operace. To jsou

instrukce vhodné např. k realizaci regulátorů a jejich automatickému seřizování. Dále se můžeme setkat s instrukcemi pro fuzzy logiku a fuzzy regulaci, pro operace s daty a s datovými strukturami, pro realizaci ucelených funkčních bloků, pro ukládání dat do zásobníků a pro podporu komunikací. Tyto specializované instrukce usnadňují programování, protože nabízejí již hotové ucelené funkce. Zvyšují však požadavek na výpočetní výkon PLC. Systémovými službami jsou označovány prostředky, které centrální PLC poskytuje nad rámec instrukčního souboru. Řadíme sem soubor systémových registrů obsažených v systémovém programu.

Podle doby vykonávání instrukcí můžeme posoudit takzvanou výkonnost PLC automatu. U výkonných automatů se obvyklé hodnoty pohybují v řádu jednotek mikrosekund na instrukci. U malých systémů bývají časy řádově v jednotkách až desítkách mikrosekund na instrukci. Uváděné časové údaje však často odpovídají jen nejzákladnějším logickým instrukcím, ostatní instrukce mívají tyto časy mnohonásobně delší, a to někdy i desetkrát či stokrát. V předchozím odstavci již bylo zmíněno, že požadavek na výpočetní výkon PLC zvyšují specializované instrukce, které realizují komplexní funkční bloky (PID regulátory, fuzzy regulátory, krokové řadiče, generátory funkcí atd.) nebo které dovolují efektivní zpracování souborů dat a datových struktur. Výkonnost systému bude záviset na typu úloh a na prostředcích, které uživatel použije při psaní programu. Z tohoto důvodu závisí výkonnost systému na kvalifikovanosti programátora. A to proto, jaké a jak výkonné instrukce, postupy a systémové služby využije.

2.5.3 PLC automat Mitsubishi MELSEC iQ-F Series

Srdcem zařízení je PLC automat značky Mitsubishi MELSEC řady iQ-F Series, konkrétně model 'FX5U-32MT/ESS'. Jedná se o novou generaci kompaktních PLC automatů, které umožňují připojení velkého množství rozšiřujících modulů. Základní modul obsahuje integrovaný síťový zdroj, CPU a integrované vstupy/výstupy. Tento konkrétní model obsahuje 16 tranzistorových vysokorychlostních digitálních vstupů a výstupů. Standardně je vybavený zabudovanými vysokorychlostními čítači, polohovacími výstupy, ethernetovým rozhraním a slotem pro SD kartu. PLC automat je napájen ze sítě

230V / 50Hz. Na jeho čelní straně se nacházejí kontrolní led diody signalizující současný stav vstupů a výstupů a také stav samotného PLC. Např. signalizaci pokud je PLC v chybě a je vyžadován restart. Parametry tohoto PLC automatu lze nalézt na webových stránkách výrobce.

2.5.4 Polohovací Simple Motion modul FX5-40SSC-S

K zvládnutí složitých polohovacích funkcí je k základní jednotce připojen rozšiřující polohovací Simple Motion modul 'FX5-40SSC-S', který doplňuje jeho integrované polohovací funkce. Se základním modulem je kvůli rychlosti přenosu propojen přes paralelní sběrnici. Simple Motion modul obsahuje vlastní CPU optimalizované pro rychlé zpracování dat (poziční data, ovládací signály: start, stop, jog,..) do formátu, kterému servozesilovače rozumí a zajišťuje jejich přenos. Dále zprostředkovává výpočty křivek náběhů/doběhů, vyhodnocuje příchozí data (identifikace chyb, blokování poruchových chodů). Obsahuje převodník pro optickou komunikaci a knihovny všech podporovaných servozesilovačů. Podporuje lineární (až čtyři osy) a kruhovou interpolaci (max. 2 osy). Pokud je zapotřebí větší počet os, je možné k základnímu modulu připojit další 3 rozšiřující Simple Motion moduly. K přenosu dat a řídicích signálů k servozesilovačům slouží vysokorychlostní komunikační síť SSCNETIII/H, založena na přenosu přes optický kabel. Umožňuje plně duplexní komunikaci s přenosovými rychlostmi až 150 Mb/s. Optická komunikace je zde využita kvůli své schopnosti odolávat elektromagnetickému rušení, způsobené servomotory. Díky této skutečnosti odpadá např. potřeba stínění či zemnění.

2.6 Dotykový panel MT8100iE2

HMI panel Weintek MT8100iE patří k nejlevnějším 10" panelům na českém trhu, přesto velmi dobře postačí pro vytvoření mnoha ovládacích rozhraní různých strojů, zařízení a aplikací. Důležitou vlastností je přímá podpora komunikace s většinou vyráběných PLC napříč všemi výrobci a dodavateli (např. právě Mitsubishi). Není tedy nutné mít obavy, že by napojení na PLC systém bylo problematické. To byl i důvod použití tohoto panelu k obsluze navíjecího zařízení.

Panel je vybaven dotykovým TFT LCD displejem 800x400, dvěma COM porty podporující sériové rozhraní RS-232 nebo RS-485, jedním USB a jedním ethernet portem. Na přenosovou vrstvu RS-485 a ethernet je implementován firmware pro podporu většiny protokolů. Tedy i pro MELSEC - protokol vytvořený výrobcem Mitsubishi. Vnitřním jádrem HMI panelu je CPU v podobě 600 MHz 32-bitového RISC procesoru Cortex-A8 doplněného o 128 MB flashpaměti pro program a 128 MB RAM paměti pro data. Veškeré parametry panelu lze nalézt v příloze [D]. Dále panel podporuje VNC server vzdálené zprávy. Pokud je připojen k internetu, lze se s pomocí PC či jiného zařízení spojit s panelem prakticky odkudkoli. To poslouží např. k vzdálené kontrole alarmových hlášení atd.

3 Návrh programového řízení

Jednou z nejdůležitějších částí tohoto zařízení je samozřejmě samotný program. A protože od zařízení je očekáváno plně automatické navíjení cívek, je potřeba k tomu vytvořit odpovídající program. Návrh je takový, že po zvolení sestaveného navíjecího programu jej uživatel spustí a stroj už bude navíjet cívky automaticky, aniž by se o něj kdokoli musel starat. V případě potřeby navíjení nového druhu cívek bude k dispozici funkce, pomocí které uživatel bude moci editovat stávající navíjecí program nebo vytvořit zcela nový. K sestavení navíjecího programu bude k dispozici 5 programových modulů. Těmi jsou: navíjecí, koordinační, resetovací, opakovací a I/O modul. Každý z modulů má svoji specifickou funkci viz kapitola [4.5]. Jednotlivé moduly bude možné použít v navíjecím programu vícekrát a jejich posloupnost bude taková, jakou bude daná aplikace vyžadovat. Maximální počet modulů v jednom programu je 50. Celý průběh navíjení bude zobrazován na displeji (např. celkový počet navinutých kusů, počet závitů aktuálně navíjeného kusu atd.).

K vytvoření programu byl použit vývojový software GX Works3, který je konkrétně určený pro rodinu Mitsubishi PLC, tedy pro PLC použité v mém zařízení (MELSEC iQ-F, model 'FX5U-32MT/ESS'). Pro vytvoření grafického rozhraní pro displej byl použit software EasyBuilder PRO od výrobce panelu (Wientek, model 'MT8100iE2').

3.1 Prostředí GX Works3

Software GX Works3 je inženýrský nástroj pro konfiguraci nastavení, programování, ladění a provádění údržby pro programovatelné automaty rodiny Mitsubishi. GX Works3 podporuje standardizované programovací jazyky jako jsou LD (Ladder diagram), ST (Structured text) nebo FBD (Function block diagram). V rámci stejného projektu umožňuje použití několika různých jazyků zároveň. Mezi nimi je pak možné jednoduše přepínat v záložkách nabídky. Software dále obsahuje kompilátor, obsáhlý grafický konfigurátor, pokročilý diagnostický systém, simulátor a spoustu dalšího.

V aplikaci má vše své místo a vše je seřazeno do specifických bloků. V levé části se nachází navigační panel projektu, ze kterého má uživatel přístup prakticky kamkoli. Nalezneme zde jak samotný program, tak i konfiguraci CPU a všech rozšiřujících modulů (v případě tohoto zařízení Simple Motion modul). Dále zde lze nalézt nastavení parametrů funkcí, proměnné, registry a mnoho dalšího. Vzhled prostředí je ukázán v příloze [E].

Vzhledově vypadá prostředí GX Works3 jednoduše a přívětivě. Software je však poměrně nový, a proto je částečně nestabilní. Před vydáním nových aktualizací měla aplikace často nepříjemný zlovyk během kompilace projektu „zamrznout“. Jednodušeji řečeno aplikace přestala odpovídat. To vyžadovalo restart aplikace a následně pokračovat v psaní programu od místa posledního uložení. Prostor je však stále v intenzivním vývoji a nové aktualizace, které řeší takovéto nedostatky, vycházejí velmi často. Nedocenitelnou výhodou softwaru je jeho obsáhlá možnost nápovědy, grafický konfigurátor a diagnostický systém. Kromě prvotních pádů šlo především o drobné záležitosti a při delší práci si na prostředí člověk brzy zvykne.

3.1.1 Programování v GX Works3

Prvním krokem na začátku tvorby programu je nezbytné vybrat správné segmenty z obsáhlé nabídky grafické modulové konfigurace 'Module Configuration' (viz příloha [E]). Je tedy potřeba zvolit konkrétní součásti, které jsou fyzicky použité v zařízení, tedy CPU modul 'FX5U-32MT/ESS' a Simple Motion modul 'FX5-40SSC-S'.

Zvolením daných segmentů jsou načteny všechny registry a funkce, které tento model podporuje. Dalším krokem je konfigurace vlastních hardwarových komponentů ('System Configuration'). V systémové konfiguraci Simple Motion modulu jsou vybrány konkrétní servozesilovače ('MR-JE20B' a 'MR-JE70B'). Pro jeden Simple Motion modul mohou být vybrány až 4 servozesilovače. Pokud bude nakonfigurován větší počet jednotek než počet jednotek skutečně připojených k modulu, systém jejich absenci rozpozná a zahlásí chybu konfigurace. Jednotlivými parametry servozesilovačů lze nastavit v okně 'Parameters'. Zde se zadávají požadované osy ('Axis #1', 'Axis #2',...). Každé ose lze v 'Compute Basic Parameters' vybrat příslušný mechanismus, kterým osa fyzicky pohybuje. Např. pro pohyb vodička je vybrán horizontální kuličkový šroub (viz příloha [F]). Systém přepočítá zadané

hodnoty a výsledné hodnoty doplní do tabulky parametrů. Pro správné nastavení všech předchozích parametrů je nyní možné zadávat hodnoty do poziční tabulky. Kromě jiného se zde definují také hodnoty pro synchronizaci (posun druhé osy podle pohybu osy první).

Při psaní programu v GX Works3 bylo využito především jazyku funkčních bloků (FBD) a strukturovaného textu (ST). Část programu, která řeší automatický chod a přepočty, je psána v jazycích FBD a LD. V jazyce ST je psána část programu obsluhující práci s displejem a je ukázána v závěru práce v příloze [I].

3.2 Prostředí EasyBuilder Pro

Tento software umožňuje vytváření designu zobrazení jednotlivých obrazovek a kompletní nastavení funkcí dotykového panelu. Obsahuje plnohodnotnou online i offline simulaci na PC. Grafické vytváření obrazovek a konfigurace parametrů je díky přehlednému prostředí velmi jednoduché. Vývojové prostředí nabízí širokou paletu prvků, se kterými lze dále pracovat. Vytvořené obrazovky jsou přehledně seřazeny v levé části prostředí. Vzhled prostředí je ukázán v příloze [G].

3.2.1 Programování v EasyBuilder Pro

Prvkům, které jsou vloženy na pracovní plochu (jednotlivá okna na displeji) lze nakonfigurovat jejich parametry (registr, se kterým prvek komunikuje, barva, tvar, dynamické chování, pozice, přístupnost). Veškeré funkce se konfigurují formou objektů, které se kompilují pro příslušnou obrazovku. Výsledný program nahraný do displeje je tedy sadou podprogramů (oken), které pracují nezávisle na sobě (formou vláken). Aktivní je vždy pouze okno uživateli zobrazené. Pokročilá nastavení a funkce se dají, kromě prostředí, nastavovat zápisem z PLC do systémových registrů (oprávnění pro zápis je možné nastavit při konfiguraci). Zkompilovaný program v displeji je možné zpětně stáhnout do vývojového prostředí v případě ztráty zdrojového projektu. Pro komerční nasazení je možné tuto funkci zablokovat nebo chránit heslem. Komunikace mezi počítačem a displejem probíhá po ethernetu s TCP/IP protokolem.

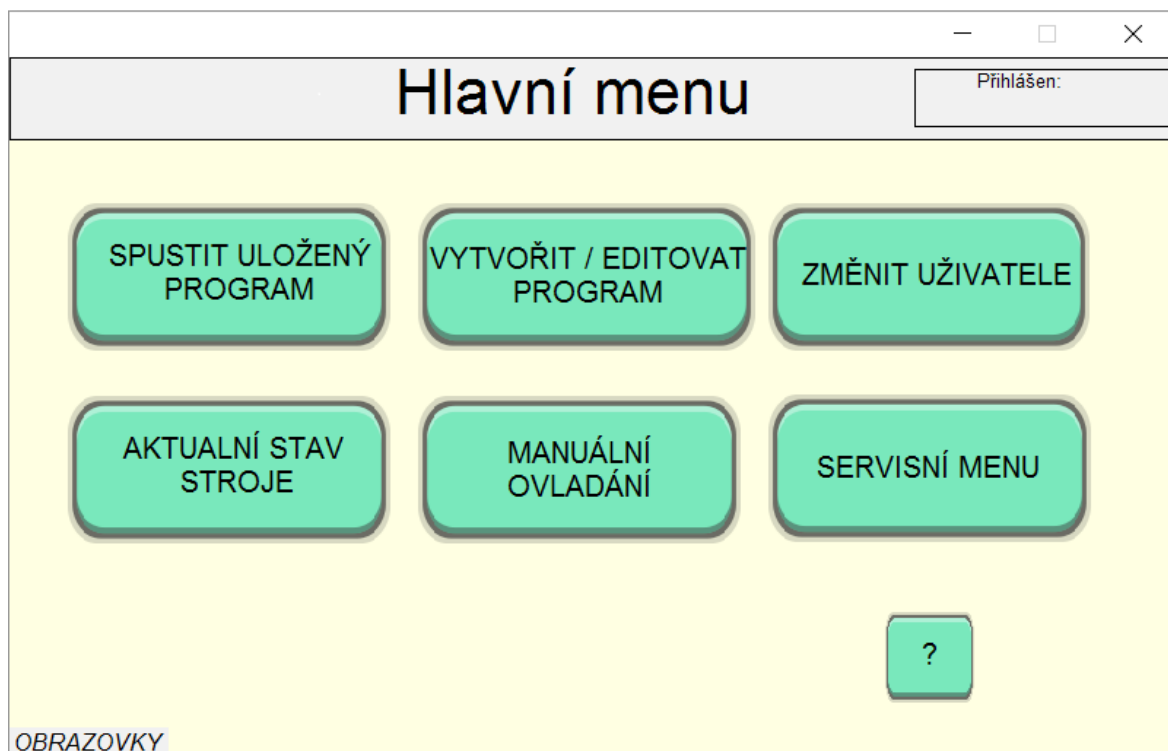
4 Řešení programu

Úplně prvním krokem před samotným psaním programu bylo důležité ujasnit si, jaké funkce budu od zařízení očekávat. Z tohoto důvodu bylo nejprve vytvořeno hlavní menu, ve kterém vyberu právě jednu z požadovaných funkcí (viz Obr. 4.1). Druhým krokem bylo ověření funkce zapojení a následná aktivace a rozpohybování servomotorů. To bylo nejprve realizováno v manuálním režimu prostřednictvím takzvaného joggování (jog = nepřetržitý posuv o délce trvání signálu). Tento krok byl klíčový a po úspěšném rozpohybování servomotorů bylo možné z této části navázat a začít řešit strukturu modulů pro navíjení. Následující krok obnášel grafické vytvoření obrazovek modulů a realizaci zadávání jejich parametrů. Dále bylo zapotřebí vyřešit jejich funkcionality a řešení skládání modulů za sebe k utvoření proveditelného programu.

Vzhledem k rozsáhlosti programu by nebylo vhodné celou strukturu programu zakreslit do jediného vývojového diagramu. Takovýto diagram by byl velmi chaotický a nepřehledný, proto je struktura rozdělena na jednotlivé dílčí vývojové diagramy. Na těchto subdiagramech je již struktura celého programu zřejmá.

4.1 Menu Programu

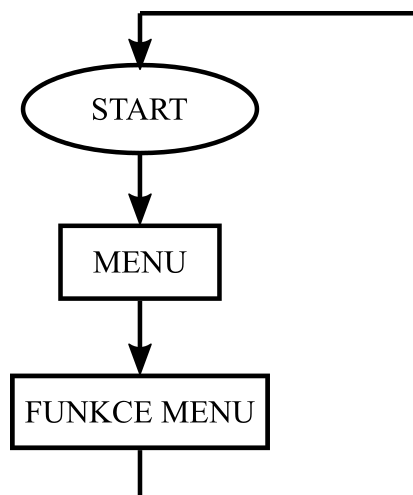
Při zapnutí zařízení a po naběhnutí operačního systému se na displeji zobrazí úvodní obrazovka vyobrazující 'Hlavním menu'. Hlavní menu je základní segment, ze kterého je možné přistoupit ke všem funkcím programu a na který se vždy program vrátí po vykonání zadané operace. Menu nabízí 6 různých funkcí. Spuštění uloženého programu, vytvoření nového či editaci stávajícího programu, změnu uživatele, zobrazení aktuálního stavu stroje, manuální ovládání stroje a servisní menu. Každá z těchto funkcí bude podrobněji popsána v následujícím textu společně s vlastním diagramem. Úvodní obrazovka je ukázána na obrázku 4.1 a vývojový diagram na obrázku 4.2.



Obr. 4.1 Ukazka úvodní obrazovky – Hlavní menu

Jednu z funkcí menu by přesto bylo dobré stručně vysvětlit hned na začátku. Jedná se o funkci 'ZMĚNIT UŽIVATELE'. Hlavní menu nabízí dalších 5 funkcí, tyto funkce však nejsou po startu přístupné všem uživatelům. Jedná se o specifický způsob zabezpečení, který zabraňuje neoprávněným uživatelům v přístupu k určitým funkcím. Nekvalifikovaná osoba by neměla mít přístup k vytváření či editaci programu. Proto je zde funkce 'ZMĚNIT UŽIVATELE', která dává uživateli na výběr, zda se přihlásí jako 'operátor' nebo jako 'programátor'. Operátor může například spustit uložený program nebo manuálně ovládat stroj, nemá však oprávnění editovat či vytvářet nové programy a měnit hesla uživatelů v servisním menu. Tyto funkce přísluší pouze uživateli přihlášenému jako programátor.

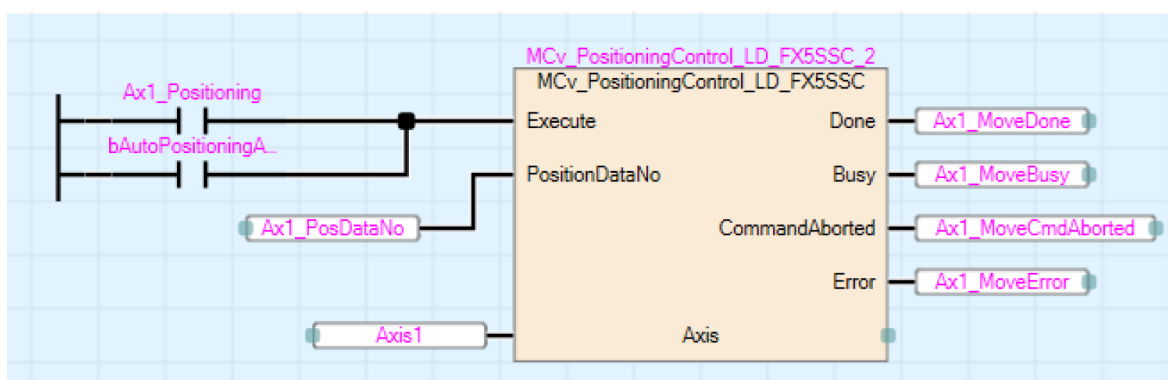
Již v úvodu bylo řečeno, že programy jsou v PLC automatech vykonávány cyklicky. To platí rovněž i v případě mého zařízení. Po zapnutí a načtení úvodní obrazovky mám možnost zvolit jednu z nabízených funkcí. Ať už jde například o spuštění či editaci uloženého programu, a nebo jednoduché manuální ovládání. Výběr funkce z hlavního menu symbolizuje blok 'MENU' v diagramu na obr. 4.2. Průběh této funkce symbolizuje blok 'FUNKCE MENU'. Po dokončení úkolu vybrané funkce se program vrací zpět na start, tedy do hlavního menu.



Obr. 4.2 Vývojový diagram chodu navijecího programu

4.2 Manuální ovládání stroje

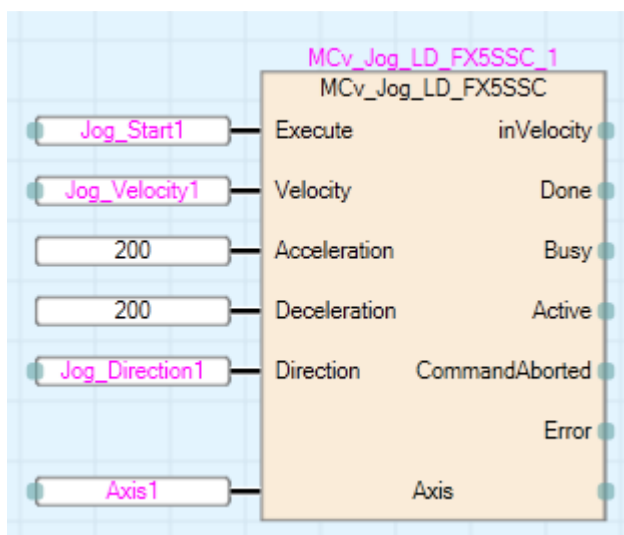
K ovládním pohybu motoru bylo nejprve nutné softwarově aktivovat servopohon. Pro spouštění serva se používá funkční blok 'MCv_PositioningControl_LD_FX5SSC_2' (označení MC označuje databázi Motion Control) na obr. 4.3, který lze nalézt v knihovně PLCopen. Blok obsahuje dlouhou předdefinovanou řadu registrů (viz příloha [H]), které zprostředkovávají aktivaci servopohonu. Vstupními parametry jsou poziční data odkazující do poziční tabulky ('Ax1_PosDataNo') a číslo osy ('Axis1,2,3,4,..'). Vstupním bitem (např. stisknutí tlačítka) je provedena aktivace ('Execute').



Obr. 4.3 Ukázka funkčního bloku vstupních parametrů pro servo

4.2.1 Jogování

Servopohon lze programově ovládat dvěma způsoby. Prvním způsobem je takzvané jogování. Jedná se o obyčejné krokování a představuje nejjednodušší možnost ovládní pohybu. Servomotor je uveden do pohybu nadefinovanou rychlostí po dobu trvání řídicího signálu (stisknuté tlačítko = motor se točí, tlačítko v klidu = motor stojí). Právě jogováním bylo zahájeno manuální ovládní. V knihovnách PLC open dodaných výrobcem se podařilo najít funkční modul ('Mcv_Jog_LD_FX5SSC_1' viz obr. 4.4), který řídí jednotlivé registry popsané v manuálu servopohonu. PLC open jsou obecné knihovny pro pozicování. V podstatě se jedná o veřejně dostupné knihovny upravené na parametry zařízení Mitsubishi. Pro jogování je nutné dodávat servopohonu tato data: rychlost ('Velocity'), zrychlení ('Acceleration'), zpomalení ('Deceleration') a směr otáčení ('Direction') viz obr. 4.4. Použité registry pro proměnné jsou přímo provázané s příslušnou obrazovkou viz obr. 4.9.



Obr. 4.4 Ukázka programu manuálního jogování

4.2.2 Poziční data

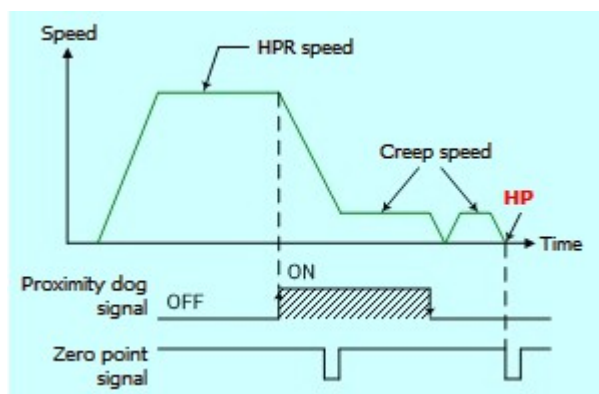
Druhým způsobem manuálního ovládní servopohonu je využití pozičních dat. Tento způsob umožňuje komplexnější pohyb, jehož parametry jsou vloženy do poziční tabulky. Tato tabulka bude nadále využívána k programovému řízení a ovládní stroje. V další fázi vývoje programu tedy bylo zapotřebí nalezení funkčních bloků (funkcí), které umožňují zápis do poziční tabulky. Poziční tabulka s jednotlivými parametry je ukázána na obr. 4.5.

No.	Operation pattern	Control method	Axis to be interpolated	Acceleration time No.	Deceleration time No.	Positioning address	Arc address	Command speed	Dwell time	M-code
49	<Positioning Comment>									
50	<Positioning Comment>									
51	1:CONT <Positioning Comment>	02h:INC Linear 1	-	0:200	0:200	3600000.0 µm	0.0 µm	100000.00 mm/min	0 ms	0
52	1:CONT <Positioning Comment>	83h:LOOP	-	0:200	0:200	0.0 µm	0.0 µm	0.00 mm/min	5 ms	2
53	1:CONT <Positioning Comment>	08h:INC Linear 2	#2	0:200	0:200	3600000.0 µm	0.0 µm	100000.00 mm/min	0 ms	0
54	1:CONT <Positioning Comment>	08h:INC Linear 2	#2	0:200	0:200	3600000.0 µm	0.0 µm	100000.00 mm/min	0 ms	0
55	1:CONT <Positioning Comment>	84h:LEND	-	0:200	0:200	0.0 µm	0.0 µm	0.00 mm/min	5 ms	0
56	0:END <Positioning Comment>	08h:INC Linear 2	#2	0:200	0:200	3600000.0 µm	0.0 µm	100000.00 mm/min	0 ms	0
57	1:CONT <Positioning Comment>	84h:LEND	-	0:200	0:200	0.0 µm	0.0 µm	0.00 mm/min	0 ms	0
58	0:END <Positioning Comment>	08h:INC Linear 2	#2	0:200	0:200	3600000.0 µm	0.0 µm	100000.00 mm/min	0 ms	0

Obr. 4.5 Ukázka části poziční tabulky s nastavenými parametry

4.2.3 Homování

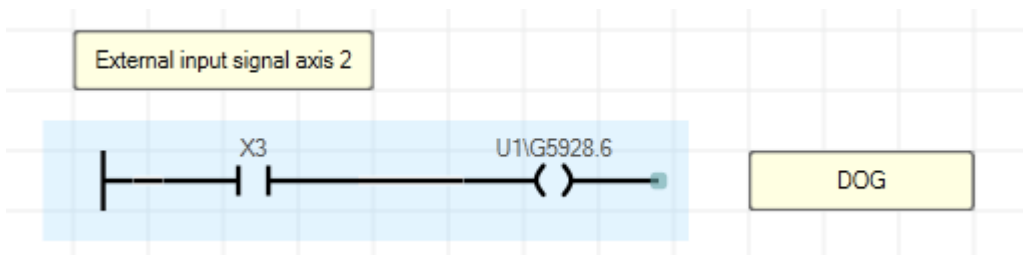
Další nezbytnou součástí bylo zprovoznění funkce 'Home', neboli vynulování osy posuvu vodička. Na to byla využita funkce „homování“ HPR (Home Position Return), také dostupná ze sady PLC open. Pro její správné fungování bylo nutné pečlivě nastavit parametry v tabulce 'HPR Basic Parameters'. Těmito parametry jsou např.: metoda homování, směr pohybu, rychlost, pozice referenčního bodu (např. na jaké čidlo se má homovat) atd. V případě mého zařízení je využita metoda 'Proximity Dog Method' neboli metoda blízkosti nebo také metoda dojezdu na čidlo. Pro tuto metodu je potřeba nastavit dojezdové vlastnosti tak, abychom nenarazili do hrany a zároveň zastavili v požadované vzdálenosti. To bylo zjištěno experimentálně.



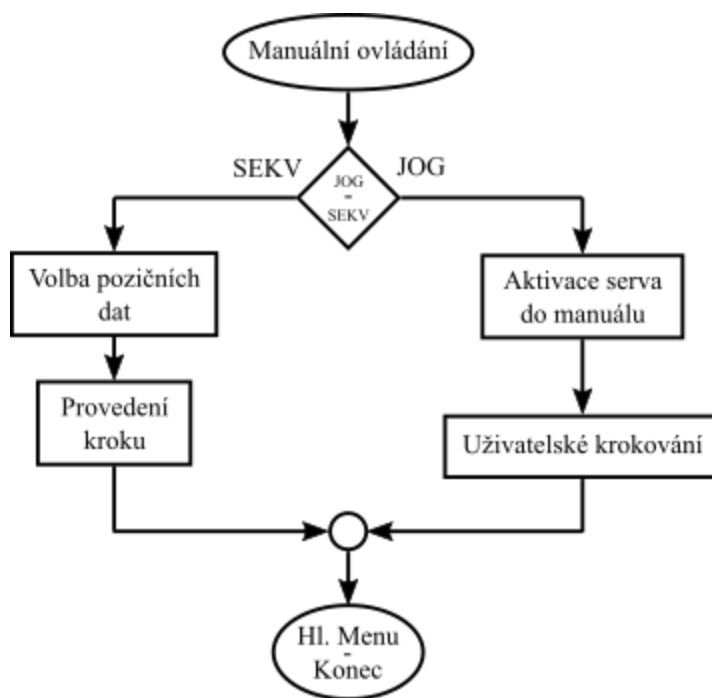
Obr. 4.6 Proximity Dog Method - metoda homování

Obrázek 4.6 znázorňuje funkci zvolené metody homování. Při vzniku požadavku najetí stroje do referenční HP ('Home Position') pozice je pohon uveden do pohybu nadefinovaným směrem a rychlostí ('HPR speed'). Při dojezdu na čidlo je vyslán signál ('Proximity dog signal'), při kterém je pohyb zpomalen na tzv. creepovou rychlost ('Creep speed'). Skončením signálu je pohyb zastaven a následně creepovou rychlostí dokončen dojezd do referenční pozice 'HP', kterou inicializuje signál nulového bodu ('Zero point signal').

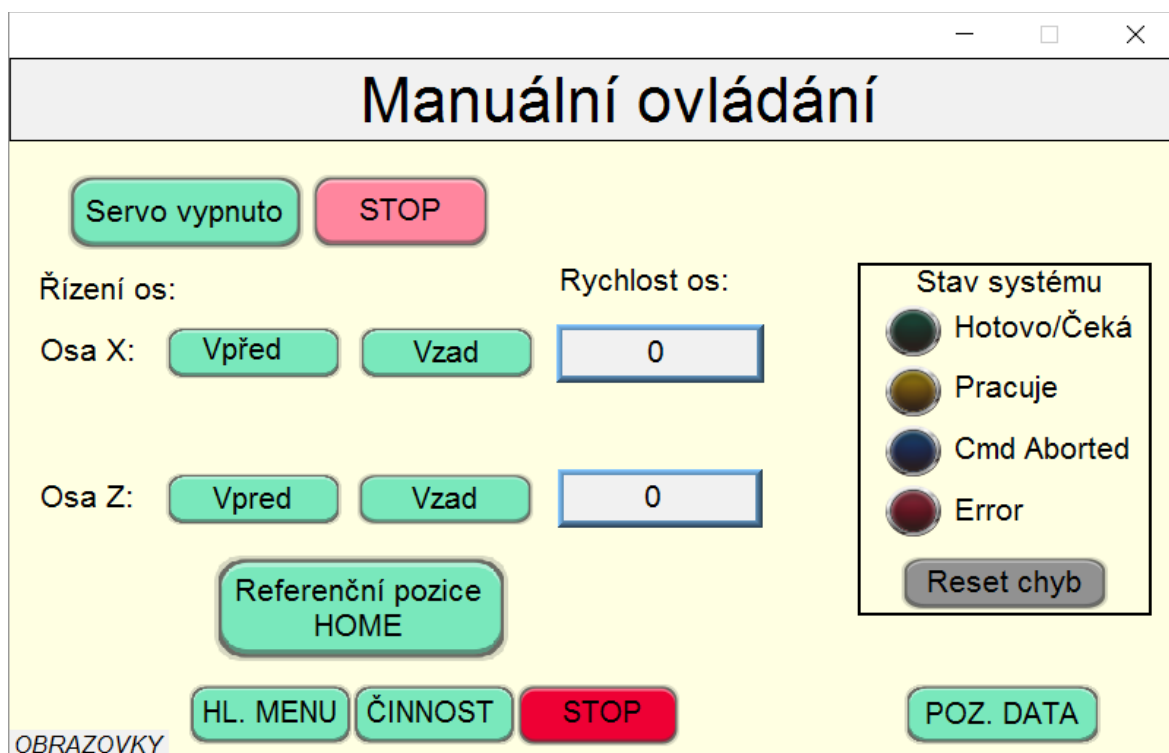
Zápis do registru Simple Motion modulu, kterým byl zařízen přenos signálu z homovacího čidla, je ukázán na obr. 4.7. Čidlo je připojené ke vstupu X3 a zapisováno je do požadovaného registru homovací funkce. Jedná se o bitový zápis.



Obr. 4.7 příklad zápisu do registru simple motion modulu



Obr. 4.8 Vývojový diagram manuálního ovládání

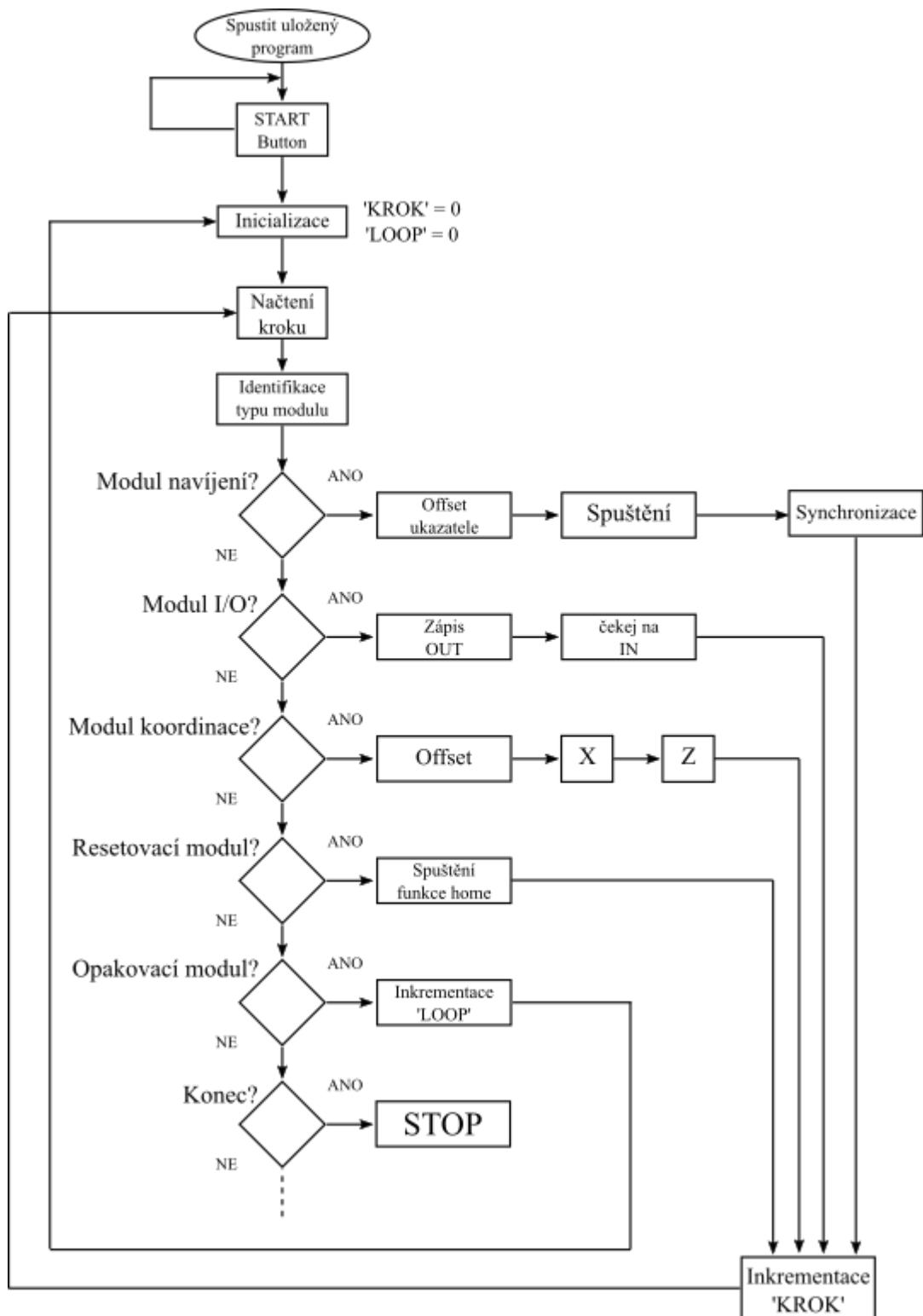


Obr. 4.9 Ukázka obrazovky manuálního ovládání z displeje

4.3 Spuštění uloženého navíjecího programu

Spustit uložený program je funkce v menu, do které má přístup každý uživatel. Při výběru této funkce je na displej vyvoláno okno se seznamem uložených navíjecích programů. Uživatel jednoduše zvolí příslušný program pro potřebnou navíjecí operaci vybraného typu cívky a spustí program. Pokud bylo vše správně připravené k navíjení, stroj bude sám automaticky navíjet, dokud program nedojde do přednastaveného počtu cyklů.

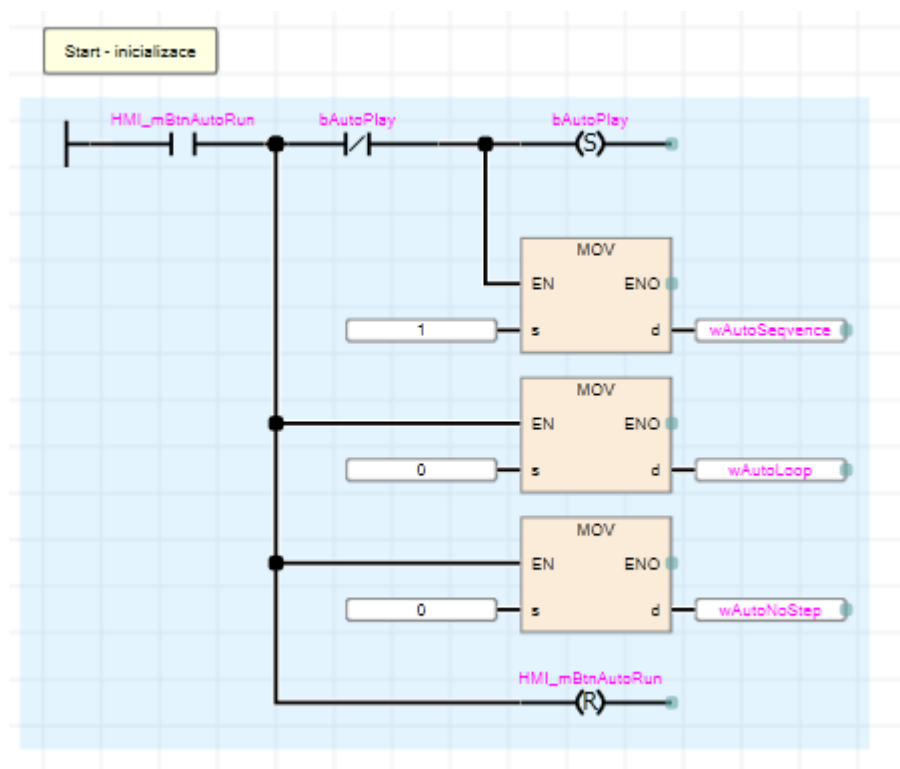
Průběh funkce sestává z několika po sobě jdoucích kroků. Po startu navíjecího programu je zahájena inicializace proměnných. Po inicializaci následuje podle aktuálního kroku načtení prvního programového modulu. Porovnávací funkcí je identifikován a vybrán správný modul a je zahájeno jeho zpracování. Po provedení funkce vybraného modulu program přejde k výběru následujícího modulu a je opět zahájeno jeho zpracování. Pokud je vybrán modul koncový, je provedena ukončovací sekvence a následný návrat na obrazovku 'Spuštění programu'. Průběh celé funkce je zobrazen na vývojovém diagramu (obr 4.10).



Obr. 4.10 Vývojový diagram manuálního ovládání

4.3.1 Inicializace

Po stisknutí start tlačítka a nahození start bitu se spustí režim inicializace. Hlavním úkolem inicializace je příprava programu pro spuštění. To znamená nastavení proměnných určujících krok do nuly a nastavení proměnné 'LOOP'. Tato proměnná určuje počet opakování navíjecího programu. Dále se inicializují provozní proměnné jako např. aktuální stav automatu, vlajky (flagy) spouštěcích tlačítek, nastavení proměnné pro opakování a autosekvenci (vnitřní krokování automatu).

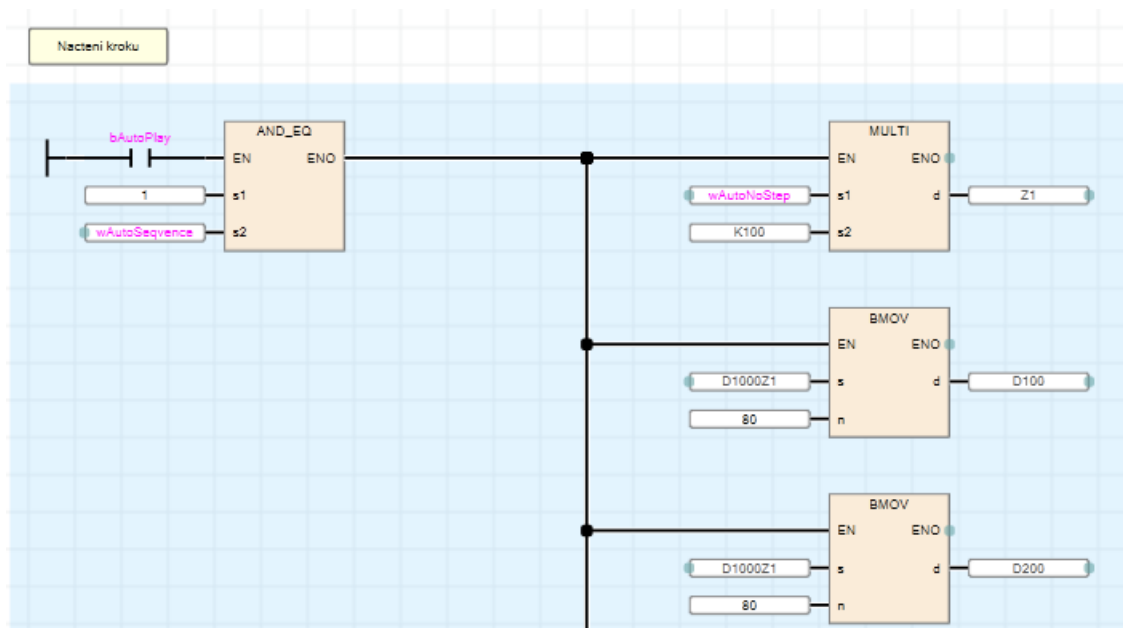


Obr. 4.11 Ukázka programu inicializační části

Výše uvedený segment kódu (Obr. 4.11) znázorňuje inicializaci proměnných po uživatelském spuštění programu (tzn. aktivaci tlačítka navázaného na proměnnou HMI_mBtnAutoRun). Jako první se nastaví proměnná automatického režimu 'bAutoPlay'. Tato proměnná se používá k indikaci provozu na displeji. Zároveň se nastaví další provozní proměnné.

4.3.2 Načtení kroku

Další fází je načtení modulu dle aktuálního kroku. To je řešeno načtením dat z místa paměti určeného výpočtem offsetu dle aktuálního požadovaného kroku. Vybraný blok dat je zkopírován do paměťového rozsahu začínajícího registrem D100 a je použit k zobrazování na displej. Zároveň ta samá data jsou zkopírována do paměťového rozsahu začínajícího registrem D200. Tato data slouží k budoucímu zpracování. Krok 0 je namapován od registru D1000 s rozestupem kroků 100. Požadavek zadání je maximum 50 kroků. Alokuje tedy paměťový prostor 100x50. Tedy až do registru D6000 (PLC obsahuje 9999 16-bitových registrů). Rozestupy po 100 krocích (100x16 bitů) jsou zvoleny s jistou rezervou, kterou umožňuje dostatek paměti v PLC. Pro přehledné adresování jsou do paměťového prostoru uloženy registry všech dostupných modulů. Nejobsáhlejší modul navíjení obsahuje 12 32-bitových registrů.

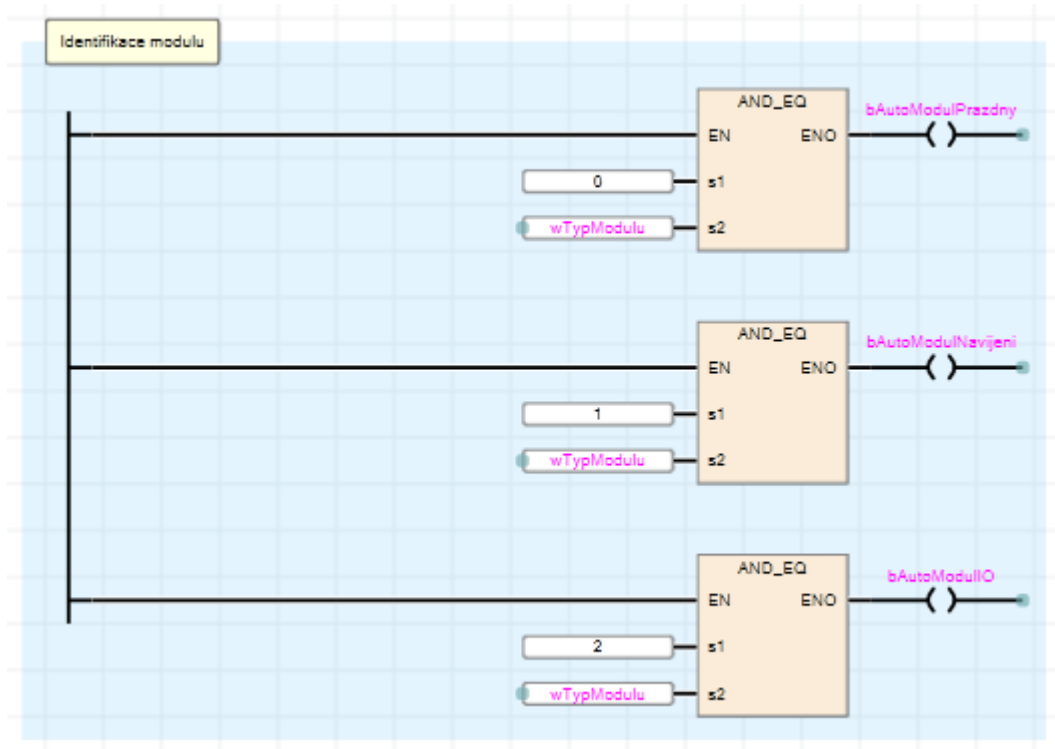


Obr. 4.12 Ukázka části programu nastavení kroku

Ve výše zobrazené ukázce programu (Obr.4.12), Z1 slouží jako ukazatel zpracovávaných dat a zápis D1000Z1 funguje jako nepřímé adresování pro přesun bloku dat. Funkční blok 'BMOV' zprostředkovává blokový přesun.

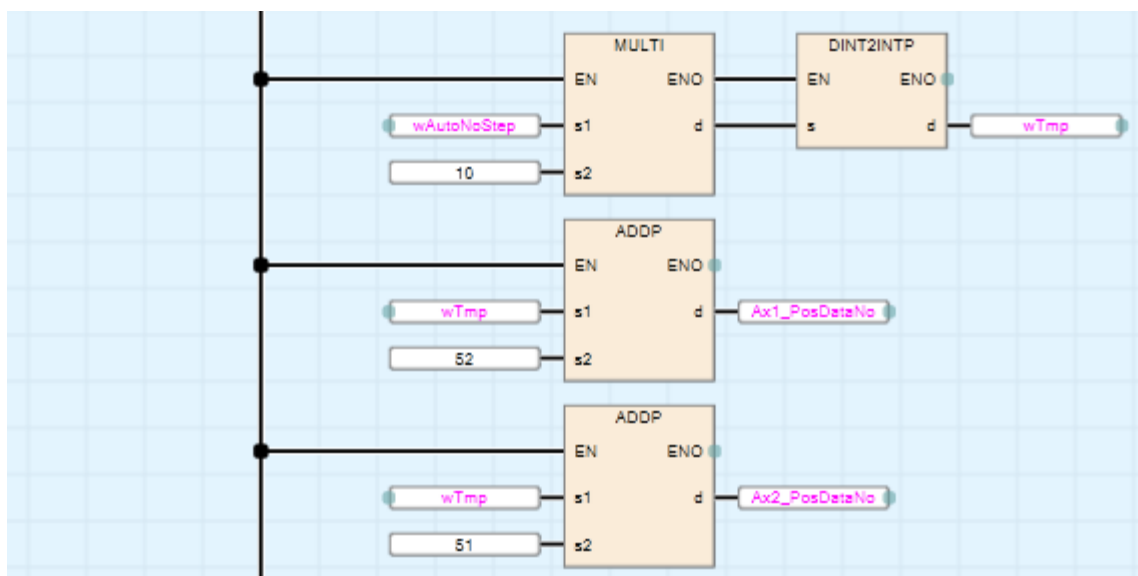
4.3.3 Identifikace typu modulu

Jedná se o fázi programu, v níž se nahlíží na paměťové místo D202, kde je uložen typ aktuálního modulu (viz vytváření/editace - kapitola [4.4]).



Obr. 4.13 Ukázka části programu identifikace modulu

Navíjecí modul:



Obr. 4.14 Ukázka části programu modulu navíjení

Výše uvedená ukázka programu (Obr. 4.13) představuje funkci 'CASE'. Na základě hodnoty bajtu stavového slova 'wTypModulu' se porovnávací funkcí vybere a nasetuje bit správného modulu. V dalším zpracování tento bit povoluje v sekvenčním zpracování programu pouze požadovanou část bloku odpovídající identifikovanému modulu.

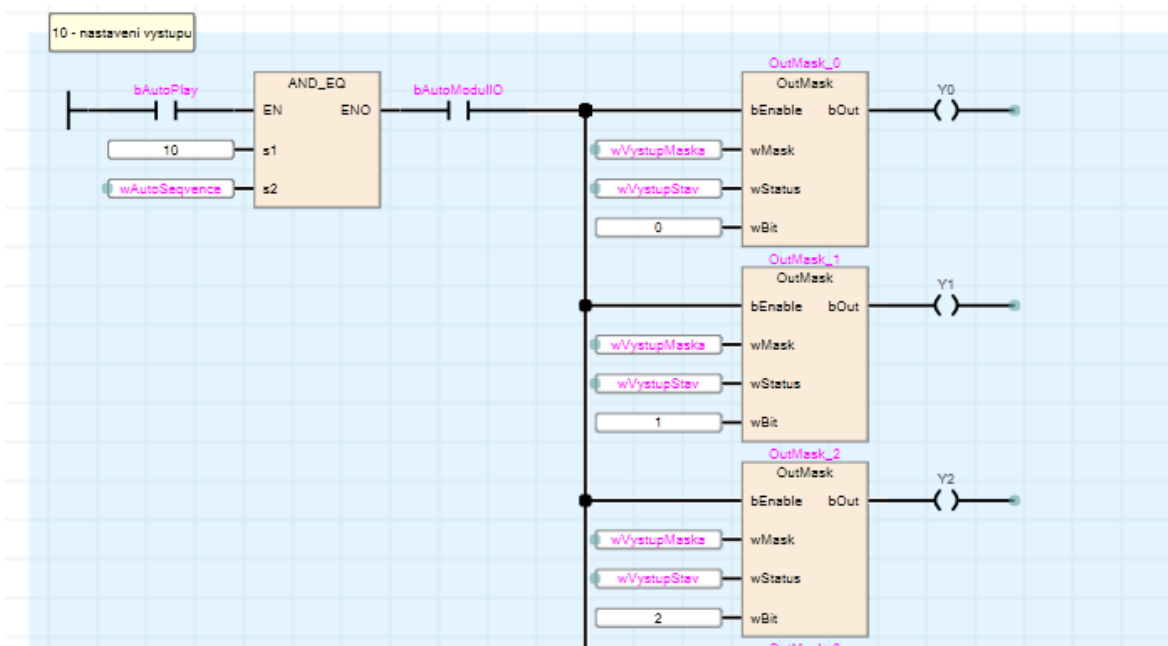
Ukázka části programu (Obr. 4.14) popisuje zápis do poziční tabulky. V bloku 'MULTI' (Multiplication - násobení) je prováděno adresování pro aktuální krok - aktuální krok x 10. Výstupem funkce 'MULTI' je 32-bitová proměnná. Pro další použití musí být převedena na 16-bitový integer. To řeší funkční blok 'DINT2INTP' (double integer to integer). V dalším programovém kroku je zapisován výsledek aktuálního kroku poziční tabulky do registru pozičních dat požadované osy ('Ax1_PosDataNo'). „Aktuální krok + 52“ odkazuje na místo v poziční tabulce, kde začínají data pro pohyb os. Na posledním zobrazeném funkčním bloku 'ADDP' na obr. 4.14 je odkazováno na poziční data osy č.2 (osy vodítka), která zajistí nastavení stroje do správné pozice. Pořadí bloků v tomto případě není důležité. Samotné spuštění pozicování bude aktivováno až po provedení těchto bloků a poziční tabulka je zpracovávána sekvenčně.

Koordinální modul:

Princip modulu je velmi podobný modulu navíjení s tím rozdílem, že poziční tabulka řeší pouze posuv os. V tomto případě není využíváno synchronního posuvu, ale naopak je každá osa posouvána separátně (každý pohyb je uložen do samostatného řádku poziční tabulky). Spuštění sekvence posuvů os je stejné jako v případě navíjení.

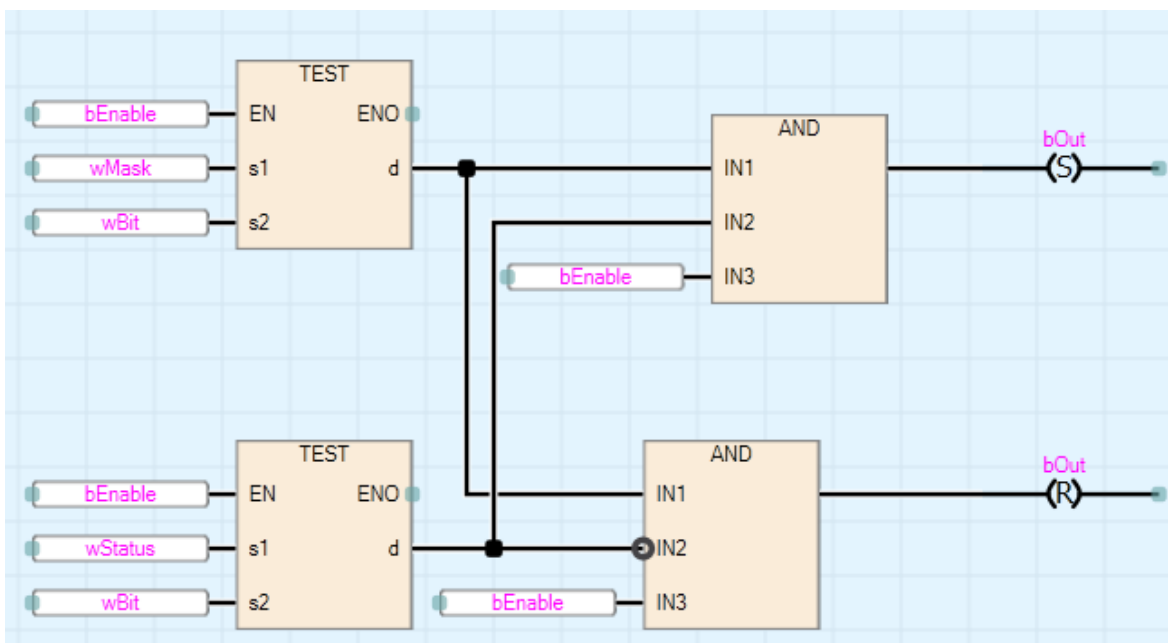
Modul I/O:

Následující ukázka části programu popisuje řízení ovládání výstupů. Funkční bloky 'OutMask' na základě uživatelem zadané masky a požadovaného stavu výstupů ovládají bit nastavený proměnnou 'wBit'. Vnitřní zapojení je vidět na obrázku níže (Obr. 4.15).



Obr. 4.15 Ukázka části programu nastavení výstupů

Následující obrázek (Obr. 4.16) ukazuje nejnižší úroveň programu (psanou v FBD pro snadnou diagnostiku). Vstupní proměnné předchozí funkce jsou zavedené do bloků 'TEST' na základě přivedeného čísla bitu. Tento bit je vybrán ze vstupního registru masky nebo stavu. Následující zapojení bloků 'AND' povolí výstup pouze v případě, kdy maska povolí svým stavem požadovaný výstup. Bit 'bEnable' povoluje celou funkci v sekvenčním vykonávání programu pouze pokud je program v dané části zpracování.



Obr. 4.16 Ukázka kódu výstupní masky

Resetovací modul:

Obstarává spouštění samostatné funkce homování jednotlivých os (viz manuální ovládání – kapitola [4.2]). Při tomto kroku se vůbec nezasahuje do poziční tabulky.

Opakovací modul:

Modul opakování čte stav proměnné 'LOOP'. Pokud hodnota proměnné odpovídá nastavenému počtu opakování, dojde ke skoku na koncový modul. Pokud je hodnota proměnné nižší než nastavená hodnota 'LOOP', dojde k programovému skoku zpět do části inicializace. Zde však tentokrát nedojde k inicializaci proměnné 'LOOP', ale dojde k její inkrementaci.

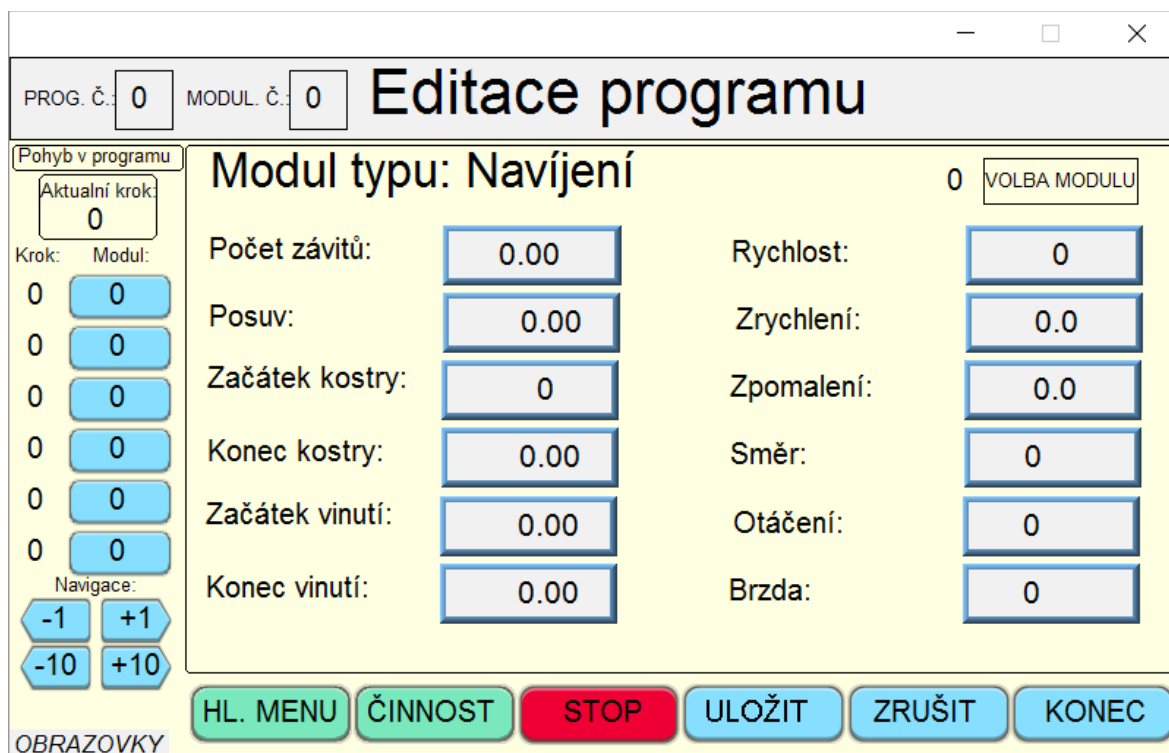
Při dokončení činnosti každého modulu dojde k programovému skoku do části načtení kroku (viz vývojový diagram Obr. 4.10). Obecný popis funkce modulů je v kapitole [4.6]

Konec programu:

Konec navíjecího programu je indikován nulovou hodnotou v proměnné typu modulu. Při dosažení této hodnoty se provede ukončovací sekvence. Shození provozních bitů a návrat na obrazovku 'spuštění programu'.

4.4 Vytváření / editace navíjecích programů

Jedná se o funkci vizualizace, ve které má uživatel na výběr, zda bude editovat (upravovat) již uložený navíjecí program, anebo vytvoří svůj zcela nový. Na obr. 4.17 je ukázána prvotní obrazovka při vytváření nového navíjecího programu. V levé části obrazovky se nachází posloupnost kroků navíjecího programu s přiřazenými moduly.



Obr. 4.17 Ukázka obrazovky editace programu

Stisknutí tlačítka editace:

Po stisknutí volby editace je nastavením příslušného bitu v displeji a jeho přenosem do PLC automat informován o zvoleném programu. Následně PLC provede vynulování registru aktuálního programu. Spočítá offset začátku dat (krok 0 – začíná registrem D1000) a data překopíruje do bufferů (registrové rozsahy od D100 a od D200). Provede se vyhodnocení typu modulu (informace je nyní uložena v registrech D102 a D202). Následně PLC zapíše do registru číslo požadované obrazovky a nastaví příslušný bit – flag. Displej tento bit pravidelně kontroluje a při jeho aktivaci provede změnu obrazovky. Stejně tak PLC pravidelně kontroluje obsah systémového registru displeje a porovnává jej s požadovanou obrazovkou. Pokud se obsahy shodují, je bit pro změnu obrazovky opět nastaven do nuly (takovéto řešení je nezbytné, protože displej neumožňuje pokročilejší skriptování).

Stisknutí tlačítka čísla kroku:

Po stisknutí kteréhokoliv tlačítka kroku se opakuje postup stejný jako při načtení editace. Tentokrát však s tím rozdílem, že do počítaného offsetu pro zkopírování bloku dat je započítán i krok požadovaný (viz obr. 4.18).

```

IF(HMI_mAktualizujEditOkno) THEN
  HMI_mAktualizujEditOkno:=FALSE;

  MHI_wAktKrok:= MHI_wLedKrok - MHI_wDispPrvniKrok;
  DECO(TRUE,MHI_wLedKrok,3 ,HMI_mAktivBlok0);
  Z1:=MHI_wAktKrok*100;
  // HMI_mAktBlok00_09:=TRUE;
  BMOV( TRUE , D1000Z1 , 80 , D100 );

  //pro zobrazeni kroku
  HMI_wProgKrok0:=MHI_wDispPrvniKrok;
  HMI_wProgKrok1:=HMI_wProgKrok0+1;
  HMI_wProgKrok2:=HMI_wProgKrok1+1;
  HMI_wProgKrok3:=HMI_wProgKrok2+1;
  HMI_wProgKrok4:=HMI_wProgKrok3+1;
  HMI_wProgKrok5:=HMI_wProgKrok4+1;
  // pro zobrazeni typu modulu
  Z1:= HMI_wProgKrok0*100;
  HMI_wProgBlok0:=D1000Z1;
  Z1:= HMI_wProgKrok1*100;
  HMI_wProgBlok1:=D1000Z1;
  Z1:= HMI_wProgKrok2*100;
  HMI_wProgBlok2:=D1000Z1;
  Z1:= HMI_wProgKrok3*100;
  HMI_wProgBlok3:=D1000Z1;
  Z1:= HMI_wProgKrok4*100;
  HMI_wProgBlok4:=D1000Z1;
  Z1:= HMI_wProgKrok5*100;
  HMI_wProgBlok5:=D1000Z1;

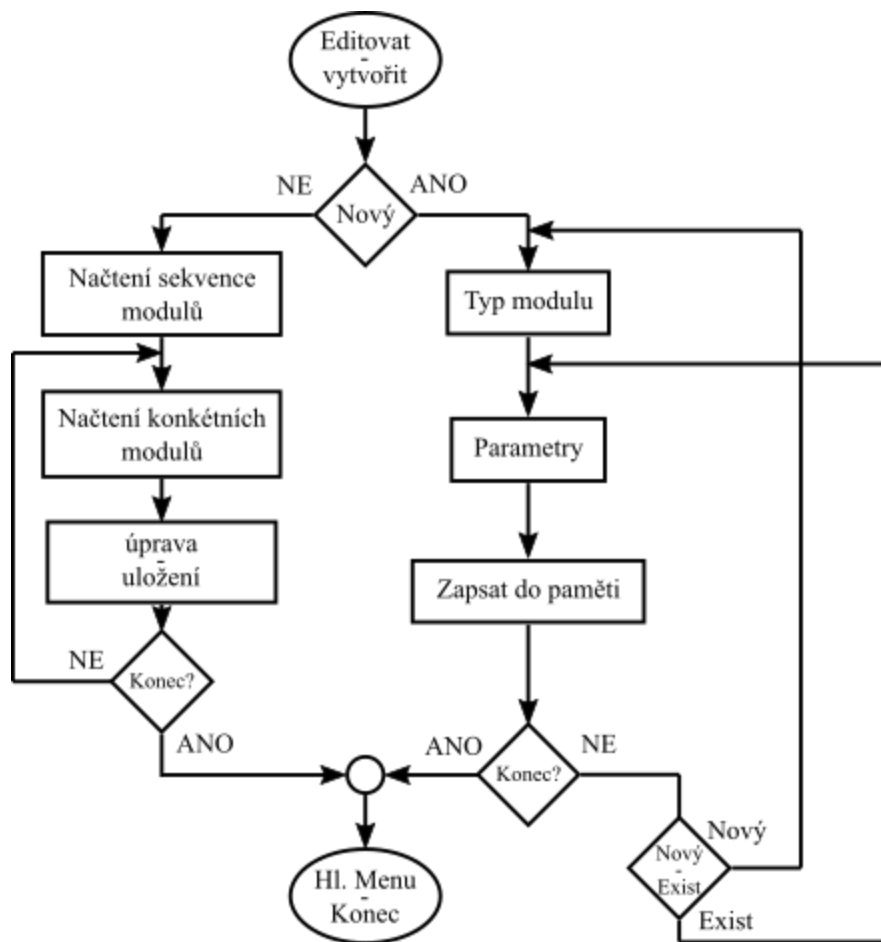
  HMI_mSetTypModulu:=TRUE;
END_IF;

```

Obr. 4.18 Ukázka kódu obsluhující práci s displejem

Stisknutí tlačítka uložit:

Při stisku tlačítka uložit dojde v PLC k dopočítání hodnot do poziční tabulky. Použita jsou aktuální data z bufferu (textová pole na displeji jsou přímo navázána na buffer). Data jsou dále na základě čísla kroku zapsána do paměti dat (opačný postup oproti načítání dat po stisknutí tlačítka kroku). Přepočítaná data jsou následně zapsána na příslušné pozice do poziční tabulky Simple Motion modulu.



Obr. 4.19 Vývojový diagram editace / vytváření navigačních programů

4.4.1 Další funkce

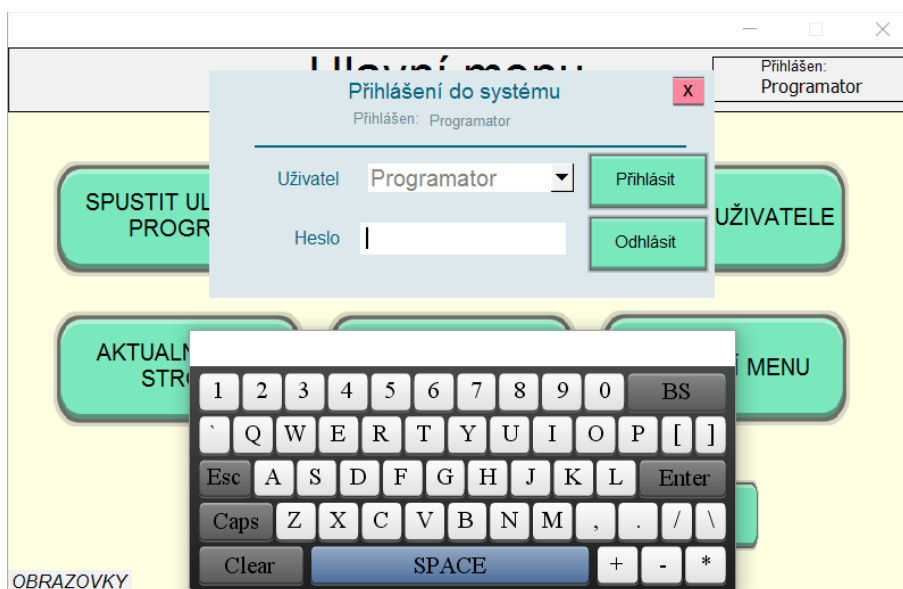
Změna uživatele:

Při vytváření jednotlivých funkcí menu byl zároveň vytvořen seznam uživatelů. Tato funkce je poskytnuta přímo v prostředí EasyBuilder PRO. S její pomocí lze v systémových parametrech jednoduše přiřadit každému uživateli rozsah oprávnění. Následně je možné libovolným prvkům v programu přidělit přístupová práva (oprávnění potřebné k interakci s daným objektem). Defaultně jsou všechny prvky přístupné všem uživatelům.

Vnitřně představuje jednotlivé uživatele sada registrů. Po zvolení jednoho z uživatelů jsou příslušné registry porovnávány s registrem, v němž je uloženo tomuto uživateli příslušející heslo. Pokud se shodují, do aktuálního registru uživatelů se nahraje daný uživatel. Na základě registru uživatelů systém přidělí přístupová práva. Průběh funkce přihlašování uživatele je znázorněn ve vývojovém diagramu (Obr. 4.20). Následuje ukázka obrazovky přihlašování uživatele (Obr. 4.21).



Obr. 4.20 Vývojový diagram editace / vytváření navijecích programů



Obr. 4.21 Ukázka obrazovky změny uživatele

Service menu:

Service menu obsahuje pouze funkci pro změnu hesla. Výběrem jednoho z uživatelů a napsáním nového přihlašovacího hesla jsou po potvrzení přepsány příslušné registry.



Obr. 4.22 Ukázka obrazovky servisního menu

Aktuální stav stroje:

Tato funkce vyobrazuje na displeji stav, ve kterém se stroj právě nachází. Zobrazována je aktuální pozice obou os, aktuálně prováděný krok a počet provedených cyklů navíjení.



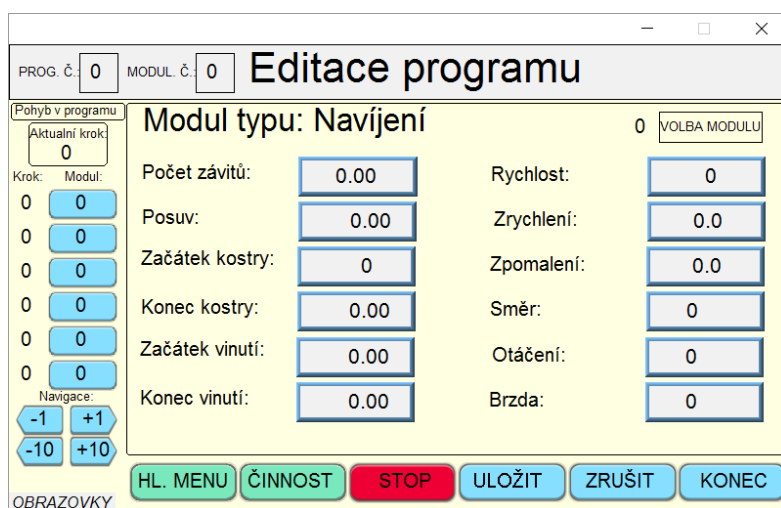
Obr. 4.23 Ukázka obrazovky stavu stroje

4.5 Programové funkční moduly

Jak již bylo v předchozích kapitolách řečeno, navíjecí program je poskládán z jednotlivých programových modulů. Každý z modulů má svojí specifickou funkci.

4.5.1 Navíjecí modul

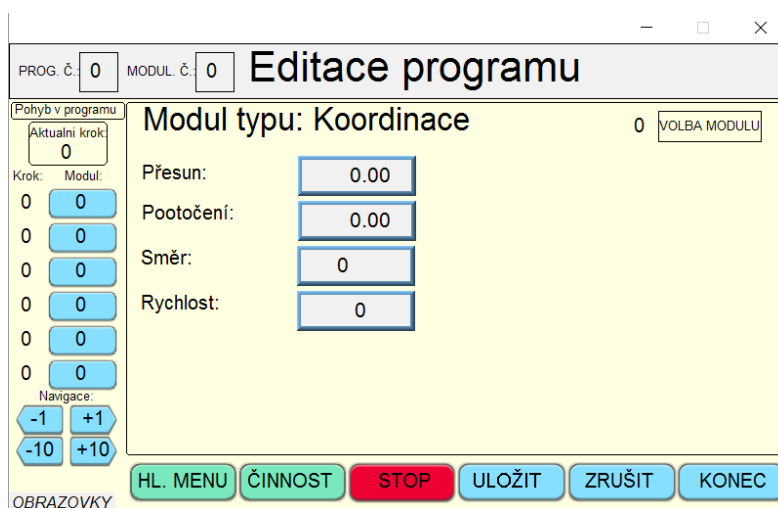
Jde o nejdůležitější část celého navíjecího programu. Modul plní funkci navíjení podle uživatelem nadefinovaných parametrů. Těmi jsou např.: počet závitů cívky, posuv drátu, rychlost navíjení atd. (viz obr. x.x).



Obr. 4.24 Ukázka obrazovky modulu navíjení

4.5.2 Koordinační modul

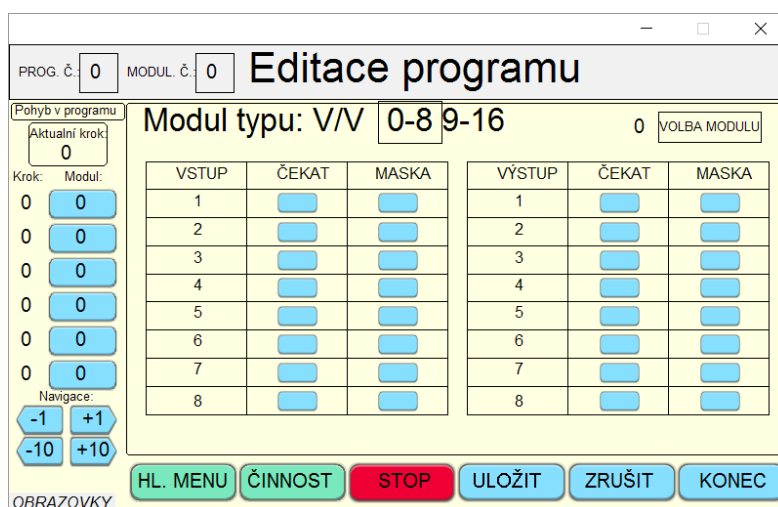
Modul koordinace slouží k dodatečnému pohybu jednotlivých os stroje. Do parametru ('přesun') se zadává pozice, na kterou má najet vodítko. Pro pohyb hřídele slouží zbylé parametry. Tedy o kolik stupňů se má hřídel potočit, jakým směrem a jakou rychlostí.



Obr. 4.25 Ukázka obrazovky modulu koordinace

4.5.3 I/O modul

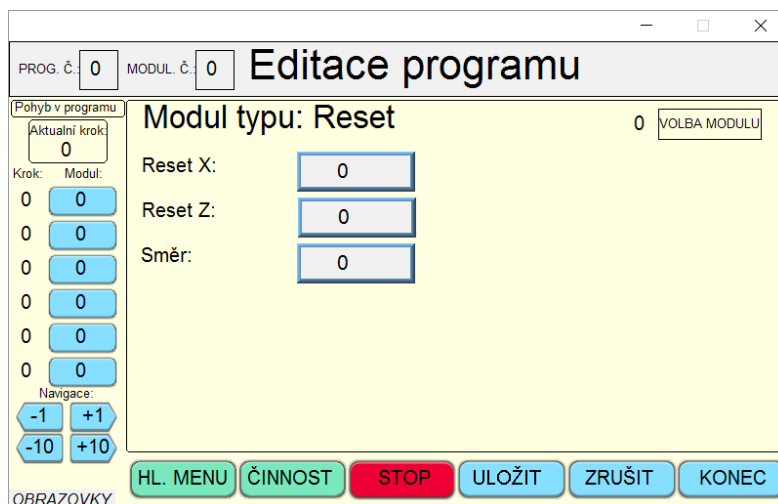
Modul vstupů/výstupů je používán k aktivaci připojených zařízení. Výstupy spouští např. vzduchové píсты či jiné mechanismy, na vstupy naopak mohou být od těchto mechanismů přiváděny signály, které reprezentují dokončení provedené operace. Pokud je vyžadováno čekání stroje právě na takovýto signál, lze označit příslušný vstup/výstup parametrem 'čekat'. Pokud vstup/výstup nebude využíván, lze jej zamaskovat.



Obr. 4.26 Ukázka obrazovky modulu vstupů / výstupů

4.5.4 Resetovací modul

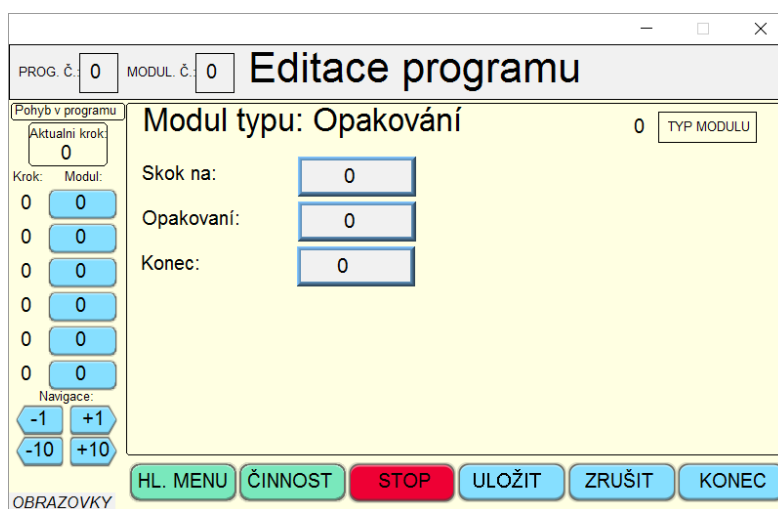
Resetovací modul spouští funkci homování, a to jak u osy vodítka (osa X), tak u osy hřídele (osa Z). Pro osu hřídele je dále možné nastavit směr, kterým do referenční pozice najede.



Obr. 4.27 Ukázka obrazovky resetovacího modulu

4.5.5 Opakovací modul

Jak už název napovídá, v tomto modulu se nastavuje počet opakování navíjení. Zapiše se požadované množství opakování a číslo kroku, na který bude proveden skok. Program bude pokračovat modulem, který se v tomto kroku nachází. Dále může tento modul sloužit jako koncový. V takovém případě se nedefinuje počet opakování, pouze se zapiše hodnota '1' do parametru konec.



Obr. 4.28 Ukázka obrazovky resetovacího modulu

Závěr

Cílem práce bylo zhotovení plně automatického navíjecího zařízení. Jak vytvoření mechanické konstrukce s vhodným hardwarem, tak vytvoření obslušného softwaru pro nastavení parametrů navíjení. Tento cíl byl dosažen jen částečně. Byla vytvořena funkční mechanická konstrukce, bohužel v části programového řešení z časových důvodů nebyl zprovozněn koordinační a opakovací modul. Zbylé programové moduly jsou funkční a možnost jejich řazení do kompletního programu rovněž. Část programu pro ukládání receptů (programů) na více druhů cívek se z důvodů složitosti nepovedlo uvést do funkčního stavu. Protože modul opakování prozatím není zprovozněn, nelze zhotovit kompletní program, který by běžel sám bez potřeby obsluhy. Ale díky faktu, že navíjecí, resetovací a I/O moduly jsou funkční, lze stroj použít jako poloautomatickou navíječku k navíjení jednoduchých cívek.

Dle mého názoru byla mechanická konstrukce vytvořena velmi nápaditým způsobem. Její řešení je praktické a nenáročné. Výkon aktuálně použitého servomotoru ('HG-KN23J') pro pohyb vodítka nebude plně využit. Bohatě by stačilo použití servomotoru s nižším výkonem, např. model 'HG-KH13J', který se nachází v nabídce společnosti Mitsubishi. Použití kuličkového šroubu pro lineární pohyb vodítka bylo nejjednodušším řešením pro tento mechanismus.

Během vytváření programu jsem se setkal s několika překážkami. První překážkou je fakt, že software GX Works3 je poměrně nový a neexistuje k němu český návod. Seznamování s prostředím teby bylo ze začátku velmi náročné. Orientovat v prostředí jsem se začal až po několika konzultacích s odborníky. Další nepříjemností byly časté pády aplikace během kompilace programu, zapříčiněné nestabilitou aplikace. Po vydání nové aktualizace již k pádům nedocházelo. Nejsložitější úlohou při řešení programu byla konfigurace os a zápis dat do poziční tabulky. Zhotovenou část programu, která je zodpovědná za automatický chod a přepočty, se podařilo realizovat v jazyce FBD a LD. Tento jazyk je pro pochopení a programování velmi jednoduchý. Umožňuje snadnou diagnostiku a obsluha při zásahu do programu nemusí být seznámena s vyšší formou programovacích jazyků.

Do budoucna se počítá s dokončením stanoveného cíle, tedy k zprovoznění všech programových modulů. Po kompletním vytvoření funkčního navíjecího automatu je další plánovanou funkcí automatizované odebírání cívek přímo z hřídele navíječky pomocí průmyslového robota řady LR-Mate od společnosti FANUC. Po dokončení navíjení bude modulem I/O vyslán signál k aktivaci programové sekvence robota. Ten cívku odebere z hřídele automatu a uloží na patřičné místo v zásobníku. Po odebrání cívky vyšle signál zpět do automatu PLC. Tento signál bude indikovat úspěšné odebrání cívky a tedy připravenost navíječky k novému cyklu navíjení.

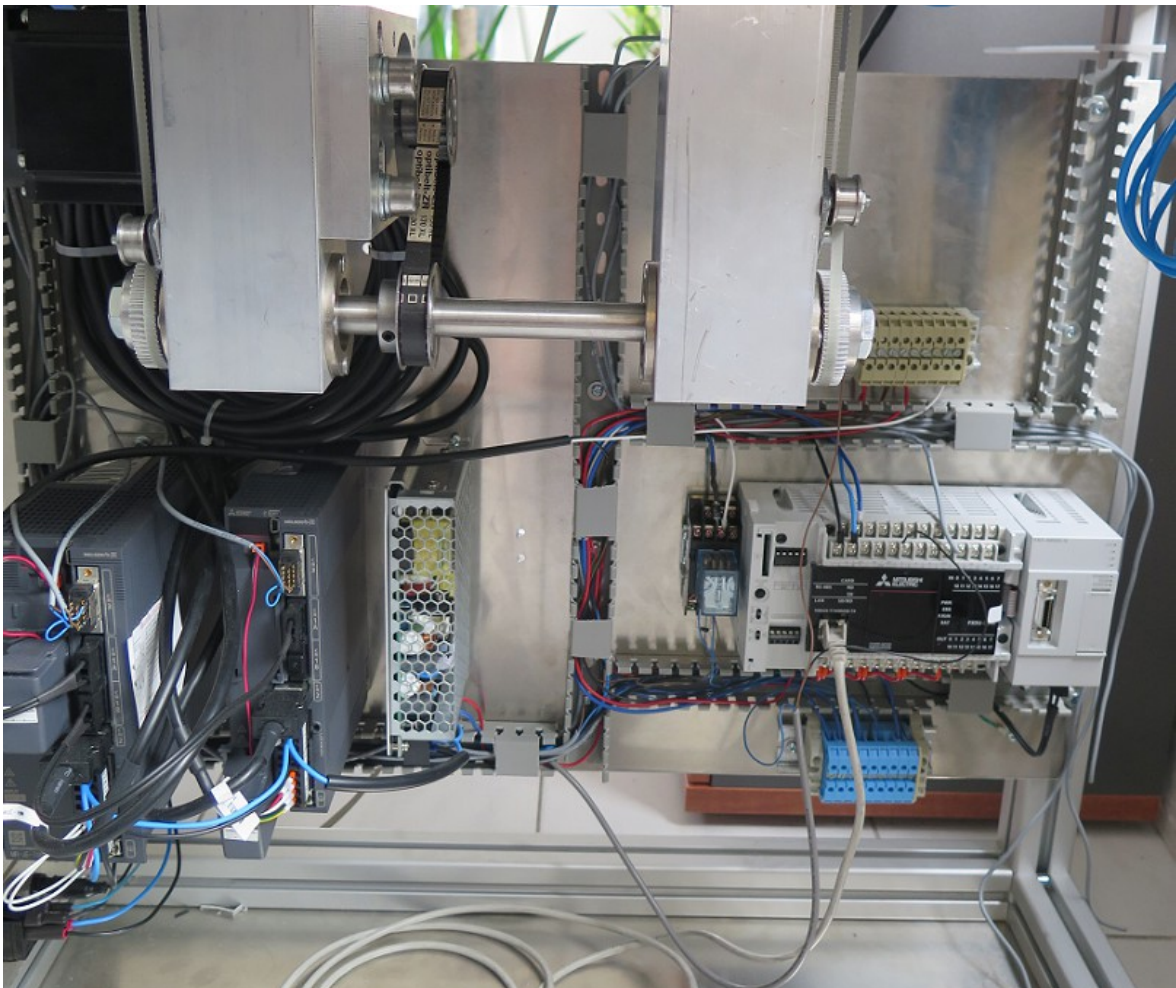
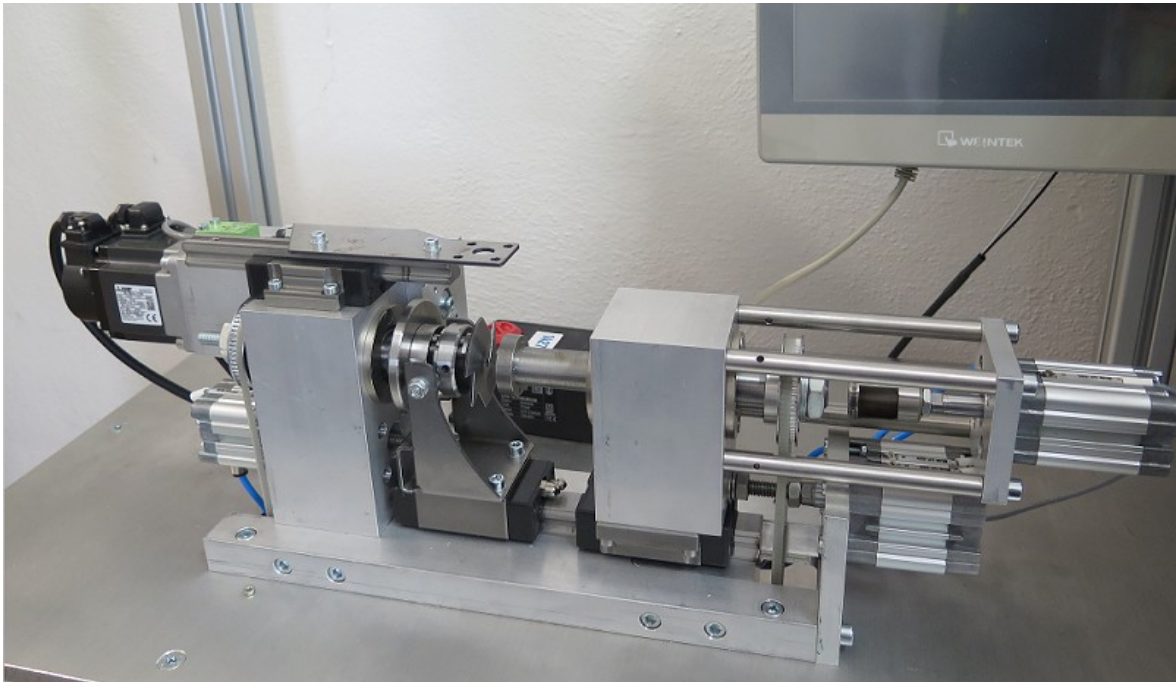
Tato práce, včetně zhotovených programů, je součástí přiloženého CD.

Seznam literatury a informačních zdrojů

- [1] BERNSTEIN, Herbert. *SPS-Workshop mit Programmierung*. 1. vyd. Berlin: VDE VERLAG GMBH, Berlin und Offenbach, 2007. 332 s.
- [2] SEITZ, Matthias. *Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation*. 4. überarbeitete und erweiterte Auflage. München: Carl Hanser Verlag, c2015. 351 s. ISBN 978-3-446-44273-3.
- [3] Skalický, Jiří. [online]. *Elektrické servopohony*. https://www.vutbr.cz/www_base/priloha.php?dpid=33400.
- [4] SEW EURODRIVE. [online]. *Drive Engineering - Practical Implementation - Servo Technology*. Edition 09/2006. Bruchsal. September 2006. 11322810/EN.
- [5] Jonata Pavel, Kondr Jan. [online]. *Průmyslové využití servozesilovačů*. 16.11.2004. <http://www.mmspektrum.com/clanek/prumyslove-vyuziti-servozesilovacu.html>.
- [6] Barua Sajib. [online]. *Understanding The Theory Of Servo Amplifier And Its Use In Servo Motor Mechanism*. <http://protorit.blogspot.cz/2011/08/understanding-theory-of-servo-amplifier.html>.
- [7] MELSEC. [online]. *Programovatelný logický automat MELSEC iQ-F Series*. Uživatelský manuál, programový manuál, Reference funkčních bloků. Dostupné z: <https://cz3a.mitsubishielectric.com/fa/cs/products/cnt/plcf/items/>

Přílohy

Příloha A – Ukázka závěrečného zpracování navíjecího stroje



Příloha B – Tabulka parametrů z katalogového listu servomotoru HG-KN23J

Servo motor model	HG-KN	13(B)J	23(B)J	43(B)J	73(B)J
Compatible servo amplifier model		Refer to "Combinations of Servo Motor and Servo Amplifier" on p. 2-1 in this catalog.			
Power supply capacity ¹¹	[kVA]	0.3	0.5	0.9	1.3
Continuous running duty	Rated output [W]	100	200	400	750
	Rated torque ^(Note 3) [N·m]	0.32	0.64	1.3	2.4
Maximum torque	[N·m]	0.95	1.9	3.8	7.2
Rated speed	[r/min]	3000			
Maximum speed	[r/min]	5000			
Permissible instantaneous speed	[r/min]	5750			
Power rate at continuous rated torque	Standard [kW/s]	12.9	18.0	43.2	44.5
	With electromagnetic brake [kW/s]	12.0	16.4	40.8	41.0
Rated current	[A]	0.8	1.3	2.6	4.8
Maximum current	[A]	2.4	3.9	7.8	14
Regenerative braking frequency ^{12, 13}	[times/min]	(Note 4)	(Note 5)	276	159
Moment of inertia J	Standard [10 ⁻⁴ kg·m ²]	0.0783	0.225	0.375	1.28
	With electromagnetic brake [10 ⁻⁴ kg·m ²]	0.0843	0.247	0.397	1.39
Recommended load to motor inertia ratio ^(Note 1)		15 times or less			
Speed/position detector	Combination with MR-JE-B	Absolute/incremental 17-bit encoder (resolution: 131072 pulses/rev)			
	Combination with MR-JE-A	Incremental 17-bit encoder (resolution: 131072 pulses/rev)			
Oil seal		Installed. Without oil seal is also available.			Installed
Insulation class		130 (B)			
Structure		Totally enclosed, natural cooling (IP rating: IP65) ^(Note 2)			
Environment ¹⁴	Ambient temperature	Operation: 0 °C to 40 °C (non-freezing), storage: -15 °C to 70 °C (non-freezing)			
	Ambient humidity	Operation: 80 %RH maximum (non-condensing), storage: 90 %RH maximum (non-condensing)			
	Ambience	Indoors (no direct sunlight); no corrosive gas, inflammable gas, oil mist or dust			
	Altitude	1000 m or less above sea level			
Vibration resistance ¹⁵		X: 49 m/s ² Y: 49 m/s ²			
Vibration rank		V10 ¹⁷			
Compliance to global standards		Refer to "Conformity with global standards and regulations" on p. 19 in this catalog.			
Permissible load for the shaft ¹⁶	L [mm]	25	30	30	40
	Radial [N]	88	245	245	392
	Thrust [N]	59	98	98	147
Mass	Standard [kg]	0.6	0.98	1.5	3.0
	With electromagnetic brake [kg]	0.8	1.4	1.9	4.0

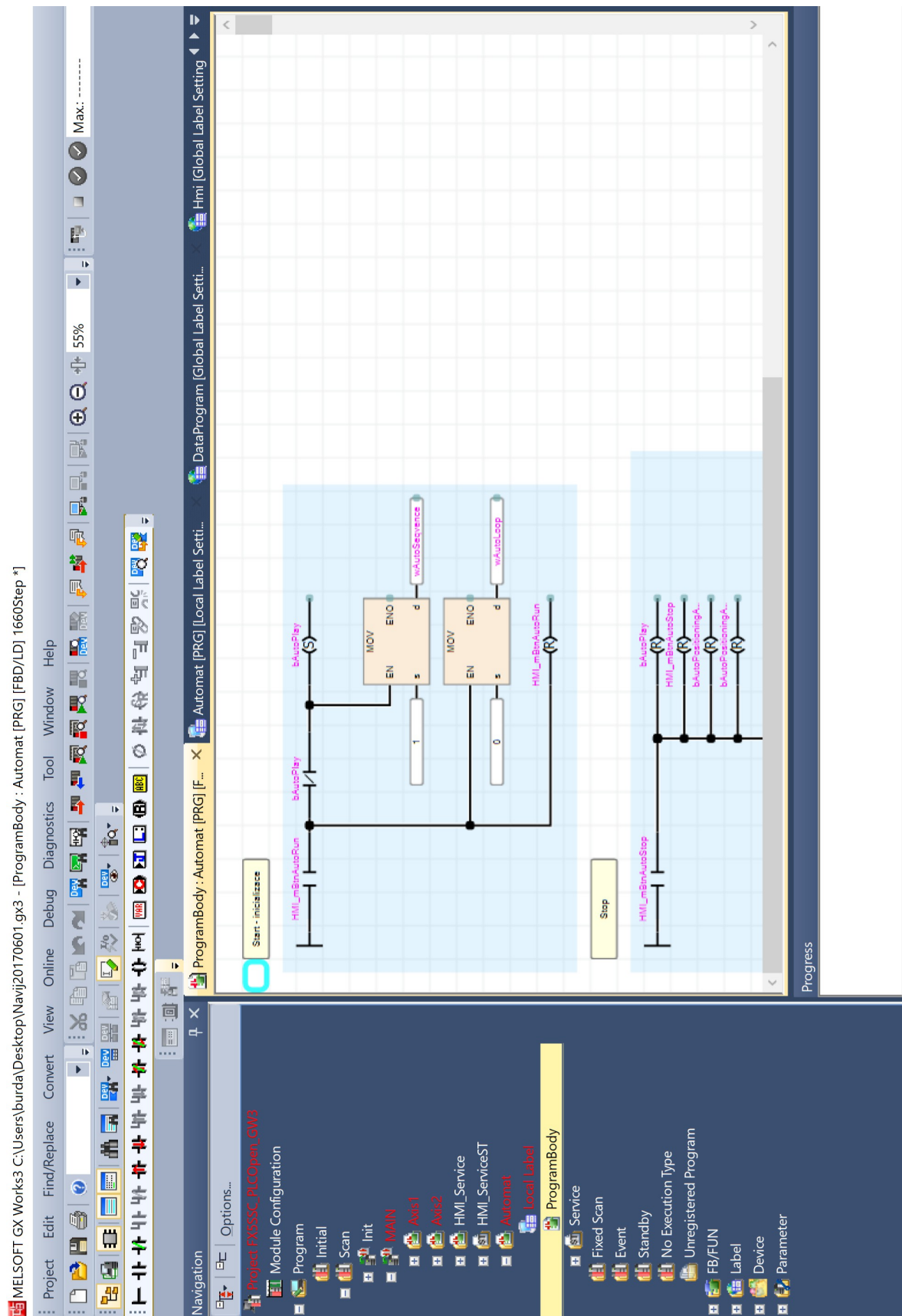
Příloha C - Tabulka parametrů z katalogového listu servozesilovače HG-KN20B

Servo amplifier model MR-JE-		10B	20B	40B	70B	100B	200B	300B	
Output	Rated voltage	3-phase 170 V AC							
	Rated current [A]	1.1	1.5	2.8	5.8	6.0	11.0	11.0	
Power supply input	Voltage/frequency ^(Note 1)	3-phase or 1-phase 200 V AC to 240 V AC, 50 Hz/60 Hz				3-phase or 1-phase 200 V AC to 240 V AC, 50 Hz/60 Hz ^(Note 8)		3-phase 200 V AC to 240 V AC, 50 Hz/60 Hz	
	Rated current ^(Note 7) [A]	0.9	1.5	2.6	3.8	5.0	10.5	14.0	
	Permissible voltage fluctuation	3-phase or 1-phase 170 V AC to 264 V AC				3-phase or 1-phase 170 V AC to 264 V AC ^(Note 8)		3-phase 170 V AC to 264 V AC	
	Permissible frequency fluctuation	±5% maximum							
Interface power supply		24 V DC ± 10% (required current capacity: 0.1 A)							
Control method		Sine-wave PWM control/current control method							
Tolerable regenerative power of the built-in regenerative resistor ^(Note 2, 3) [W]		-	-	10	20	20	100	100	
Dynamic brake		Built-in ^(Note 4)							
SSCNET III/H command communication cycle ^(Note 5)		0.444 ms, 0.888 ms							
Communication function		USB: Connect a personal computer (MR Configurator2 compatible)							
Servo function		Advanced vibration suppression control II, adaptive filter II, robust filter, auto tuning, one-touch tuning, tough drive function, drive recorder function, tightening & press-fit function, machine diagnosis function, power monitoring function, lost motion compensation function							
Protective functions		Overcurrent shut-off, regenerative overvoltage shut-off, overload shut-off (electronic thermal), servo motor overheat protection, encoder error protection, regenerative error protection, undervoltage protection, instantaneous power failure protection, overspeed protection, error excessive protection, hotline forced stop function ^(Note 9)							
Compliance to global standards		Refer to "Conformity with global standards and regulations" on p. 19 in this catalog.							
Structure (IP rating)		Natural cooling, open (IP20)					Force cooling, open (IP20)		
Close mounting ^(Note 5)	3-phase power supply input	Possible							
	1-phase power supply input	Possible			Not possible		-		
Environment	Ambient temperature	Operation: 0 °C to 55 °C (non-freezing), storage: -20 °C to 65 °C (non-freezing)							
	Ambient humidity	Operation/Storage: 90 %RH maximum (non-condensing)							
	Ambience	Indoors (no direct sunlight); no corrosive gas, inflammable gas, oil mist or dust							
	Altitude	1000 m or less above sea level							
Vibration resistance		5.9 m/s ² at 10 Hz to 55 Hz (directions of X, Y and Z axes)							
Mass [kg]		0.8	0.8	0.8	1.5	1.5	2.1	2.1	

Příloha D - Tabulka parametrů z katalogového listu HMI panelu MT8100iE

Display	Display	10.1" TFT
	Resolution	800 x 480
	Brightness (cd/m ²)	300
	Contrast Ratio	500:1
	Backlight Type	LED
	Backlight Life Time	>30,000 hrs.
	Colors	262K
Touch Panel	Type	4-wire Resistive Type
	Accuracy	Active Area Length(X)±2%, Width(Y)±2%
Memory	Flash	128 MB
	RAM	128 MB
Processor		32 bits RISC Cortex-A8 600MHz
I/O Port	SD Card Slot	N/A
	USB Host	USB 2.0 x 1
	USB Client	N/A
	Ethernet	10/100 Base-T x 1
	COM Port	COM1: RS-232/RS-485 2W/4W, COM3: RS-485 2W
	RS-485 Dual Isolation	Yes
	CAN Bus	N/A
	HDMI	N/A
	Audio Output	N/A
	Video Input	N/A
RTC		Built-in
Power	Input Power	24±20%VDC
	Power Isolation	Built-in
	Power Consumption	maximum 600mA@24VDC, Class2
	Voltage Resistance	500VAC (1 min.)
	Isolation Resistance	Exceed 50MΩ at 500VDC
	Vibration Endurance	10 to 25Hz (X, Y, Z direction 2G 30 minutes)
Specification	PCB Coating	Yes
	Enclosure	Plastic
	Dimensions WxHxD	271 x 213 x 40 mm
	Panel Cutout	260 x 202 mm
	Weight	Approx. 1.3 kg
	Mount	Panel mount
Environment	Protection Structure	UL Type 4X (indoor use only) / NEMA4 / IP65 Compliant Front Panel
	Storage Temperature	-20° ~ 60°C (-4° ~ 140°F)
	Operating Temperature	0° ~ 50°C (32° ~ 122°F)
	Relative Humidity	10% ~ 90% (non-condensing)
Certificate	CE	CE marked
	UL	cULus Listed
Software		EasyBuilder Pro V5.00.01 or later versions

Příloha E – Prostředí aplikace GX Works3



Příloha F – Nastavení parametrů pro servozesilovač 'Compute Basic Parameters'

Compute Basic Parameters 1 - Axis #2 ✕

Entry

Select the machine components, and enter the machine data to automatically set the basic parameters 1 (unit setting, No. of pulses per rotation, movement amount per rotation and unit magnification).

Machine Components : Ball Screw, Horizontal

Unit Setting 0:mm

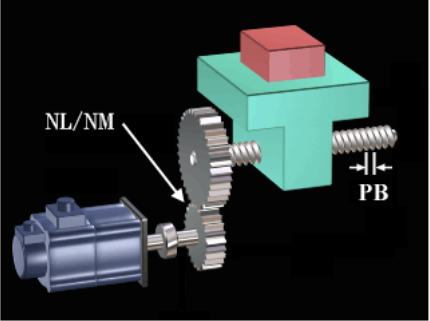
Lead of Ball Screw (PB) 4000.0 [μ m]

Reduction Gear Ratio (NL/NM)
= 1 / 1

Calculate reduction ratio by teeth or diameters Reduction Ratio Setting

Encoder Resolution 131072 [pulse/rev]

Setting Range



↓ Compute Basic Parameters 1

Calculation Result

Basic Parameters 1	Unit Setting		Movement Amount per Pulse
	No. of Pulses per Rotation		
	Movement Amount per Rotation		
	Unit Magnification		

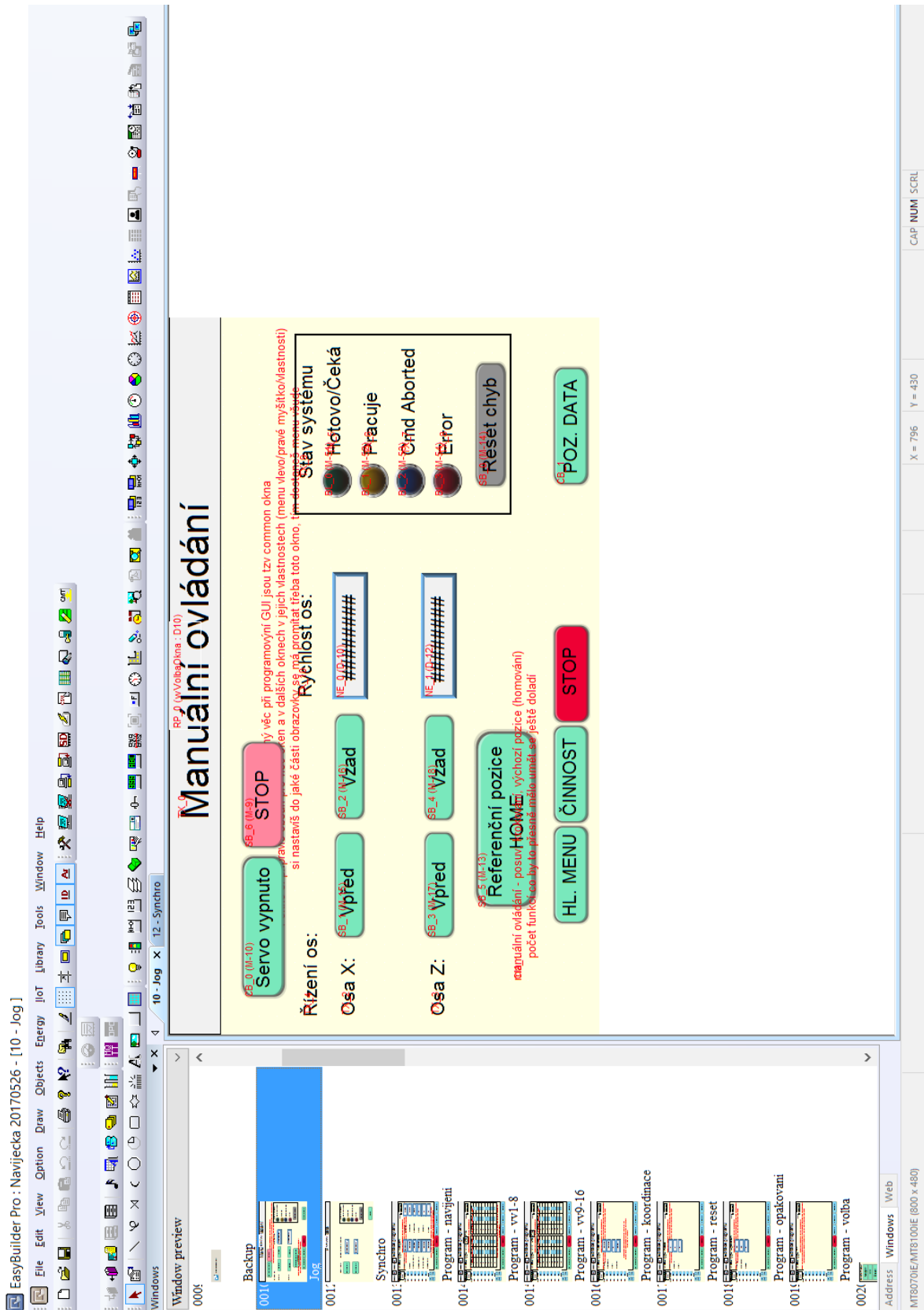
As a result of calculation, no error occurs in the movement amount.

Applying the calculation result above,

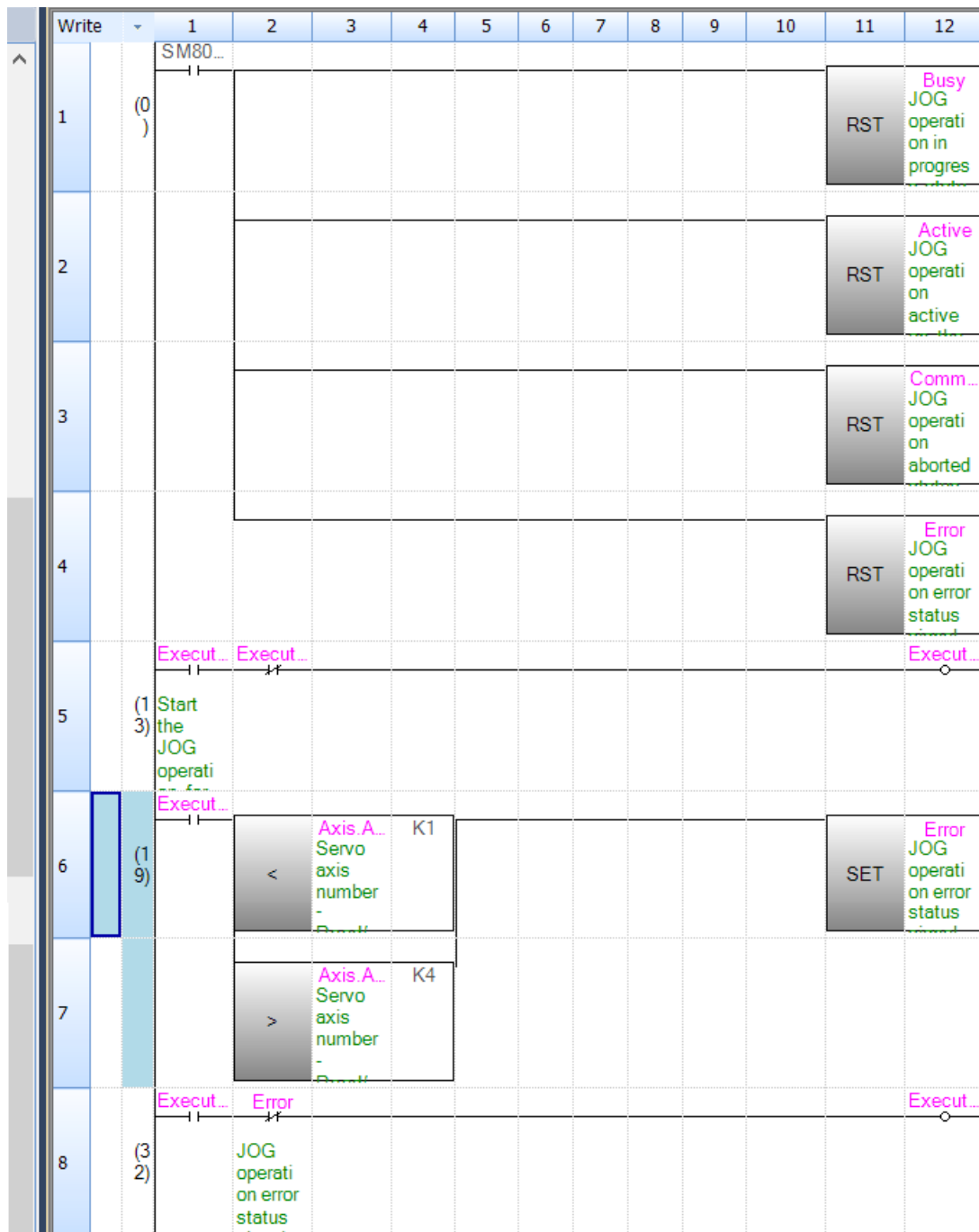
you want to perform is about 0.0 [μ m] the error for the movement amount 0.0 [μ m] Error Calculation

OK Cancel

Příloha G – Prostředí aplikace EasyBuilder PRO



Příloha H – Ukázka obsahu funkčního bloku pro spuštění serva



Příloha I – Zhotovený program v jazyku ST pro obsluhu práce s panelem

```
IF(Hmi_mBtnProgramEdit OR HMI_mBtnProgramNovy) THEN
    Hmi_mBtnProgramEdit:=FALSE;
    HMI_mBtnProgramNovy:=FALSE;
    MHI_wDispPrvniKrok:=0;
    MHI_wLedKrok:=0;
    MHI_wAktKrok:=0;
    HMI_mAktualizujEditOkno:=TRUE;
END_IF;

//ENCO( TRUE , HMI_mAktivBlok0 , 4 , MHI_wAktBlokJednotky );
//ENCO( TRUE , HMI_mAktBlok00_09 , 3 , MHI_wAktBlokDesitky);

IF(HMI_mSetTypModulu) THEN

    HMI_mSetTypModulu:=FALSE;

    IF(HMI_wTypModulu = 0) THEN
        HMI_wVolbaOkna :=19;
        HMI_mPrepniOkno:= TRUE;
    END_IF;
    IF(HMI_wTypModulu = 1) THEN
        HMI_wVolbaOkna :=13;
        HMI_mPrepniOkno:= TRUE;
    END_IF;
    IF(HMI_wTypModulu = 2) THEN
        HMI_wVolbaOkna :=14;
        HMI_mPrepniOkno:= TRUE;
    END_IF;
    IF(HMI_wTypModulu = 3) THEN
        HMI_wVolbaOkna :=16;
        HMI_mPrepniOkno:= TRUE;
    END_IF;
    IF(HMI_wTypModulu = 4) THEN
        HMI_wVolbaOkna :=17;
        HMI_mPrepniOkno:= TRUE;
    END_IF;
    IF(HMI_wTypModulu = 5) THEN
        HMI_wVolbaOkna :=18;
        HMI_mPrepniOkno:= TRUE;
    END_IF;

END_IF;

IF(HMI_mPrepniOkno) THEN
    IF(HMI_wVolbaOkna = HMI_wAktivniOkno) THEN
        HMI_mPrepniOkno:=FALSE;
    END_IF;
END_IF;

IF(HMI_mBtnPopupSmazat) THEN
    HMI_mBtnPopupSmazat:=FALSE;
    HMI_wTypModulu:=0;
END_IF;
```



```
// priznak stisku tlacitka pro vyber bloku

HMI_mSetBtnKrok:=(HMI_mBtnBlok0 OR HMI_mBtnBlok1 OR HMI_mBtnBlok2 OR
HMI_mBtnBlok3 OR HMI_mBtnBlok4 OR HMI_mBtnBlok5 OR HMI_mBtnBlok6 OR
HMI_mBtnBlok7 OR HMI_mBtnBlok8 OR HMI_mBtnBlok9);
IF(HMI_mSetBtnKrok) THEN
    HMI_mSetBtnKrok:=FALSE;

    ENCO( TRUE , HMI_mBtnBlok0 , 4 , MHI_wBtnKrok );
    MHI_wLedKrok:=MHI_wBtnKrok;

    // reset priznaku tlacitek
    MHI_wBtnKrok:=0;
    HMI_mBtnBlok0:=FALSE;
    HMI_mBtnBlok1:=FALSE;
    HMI_mBtnBlok2:=FALSE;
    HMI_mBtnBlok3:=FALSE;
    HMI_mBtnBlok4:=FALSE;
    HMI_mBtnBlok5:=FALSE;
    HMI_mBtnBlok6:=FALSE;
    HMI_mBtnBlok7:=FALSE;
    HMI_mBtnBlok8:=FALSE;
    HMI_mBtnBlok9:=FALSE;

    // reset priznaku tlacitek
    HMI_mBtnPlus1:=FALSE;
    HMI_mBtnMinus1:=FALSE;
    HMI_mBtnPlus10:=FALSE;
    HMI_mBtnMinus10:=FALSE;
    HMI_mLedRun:=FALSE;

    HMI_mAktualizujEditOkno:=TRUE;

END_IF;

// popup menu "Volba modulu"
IF(HMI_mBtnPopupNavijeni) THEN
    HMI_mBtnPopupNavijeni:=FALSE;
    HMI_wTypModulu:=1;
END_IF;
IF(HMI_mBtnPopupV_V) THEN
    HMI_mBtnPopupV_V:=FALSE;
    HMI_wTypModulu:=2;
END_IF;
IF(HMI_mBtnPopupRizeni) THEN
    HMI_mBtnPopupRizeni:=FALSE;
    HMI_wTypModulu:=3;
END_IF;
IF(HMI_mBtnPopupReset) THEN
    HMI_mBtnPopupReset:=FALSE;
    HMI_wTypModulu:=4;
END_IF;
IF(HMI_mBtnPopupOpak) THEN
    HMI_mBtnPopupOpak:=FALSE;
    HMI_wTypModulu:=5;
END_IF;
IF(HMI_mBtnPopupSmazat) THEN
    HMI_mBtnPopupSmazat:=FALSE;
    HMI_wTypModulu:=0;
END_IF;
```



```
//*****

IF(HMI_mAktualizujEditOkno) THEN
    HMI_mAktualizujEditOkno:=FALSE;

    MHI_wAktKrok:= MHI_wLedKrok - MHI_wDispPrvniKrok;
    DECO(TRUE,MHI_wLedKrok,3 ,HMI_mAktivBlok0);
    Z1:=MHI_wAktKrok*100;
//    HMI_mAktBlok00_09:=TRUE;
    BMOV( TRUE , D1000Z1 , 80 , D100 );

//pro zobrazeni kroku
    HMI_wProgKrok0:=MHI_wDispPrvniKrok;
    HMI_wProgKrok1:=HMI_wProgKrok0+1;
    HMI_wProgKrok2:=HMI_wProgKrok1+1;
    HMI_wProgKrok3:=HMI_wProgKrok2+1;
    HMI_wProgKrok4:=HMI_wProgKrok3+1;
    HMI_wProgKrok5:=HMI_wProgKrok4+1;

// pro zobrazeni typu modulu
    Z1:= HMI_wProgKrok0*100;
    HMI_wProgBlok0:=D1000Z1;
    Z1:= HMI_wProgKrok1*100;
    HMI_wProgBlok1:=D1000Z1;
    Z1:= HMI_wProgKrok2*100;
    HMI_wProgBlok2:=D1000Z1;
    Z1:= HMI_wProgKrok3*100;
    HMI_wProgBlok3:=D1000Z1;
    Z1:= HMI_wProgKrok4*100;
    HMI_wProgBlok4:=D1000Z1;
    Z1:= HMI_wProgKrok5*100;
    HMI_wProgBlok5:=D1000Z1;

    HMI_mSetTypModulu:=TRUE;
END_IF;

// navrat k puvodnim hodnotam editovaneho modulu

IF(HMI_mBtnZrusit) THEN
    HMI_mBtnZrusit := FALSE;
    Z1:= MHI_wAktKrok*100;
    BMOV( TRUE , D1000Z1 , 80 , D100);
END_IF;

// ulozeni editovaneho modulu
IF(HMI_mBtnUloz) THEN
    HMI_mBtnUloz := FALSE;

//*****

// ulozeni modulu do PLC
    Z1:= MHI_wAktKrok*100;
    BMOV( TRUE , D100 , 80 , D1000Z1);

//*****
// prepočet a uložení modulu do "simple motion"
_ServoBlockSaveWinding(indexBlock:= MHI_wAktKrok);

END_IF;
```

```
//*****  
// obsluha popup okna  
//*****  
  
HMI_dPocetZavitu:=HMI_dPocetZavitu;  
HMI_dPosuv:=HMI_dPosuv;  
HMI_dZacatekKostry:=HMI_dZacatekKostry;  
HMI_dKonecKostry:=HMI_dKonecKostry;  
HMI_dZacatekVinuti:=HMI_dZacatekVinuti;  
HMI_dKonecVinuti:=HMI_dKonecVinuti;  
HMI_dRychlostNavijeni:=HMI_dRychlostNavijeni;  
HMI_dZrychleni:=HMI_dZrychleni;  
HMI_dZpomaleni:=HMI_dZpomaleni;  
HMI_wSmerPosuvu:=HMI_wSmerPosuvu;  
HMI_wSmerToceni:=HMI_wSmerToceni;  
HMI_wBrzdit:=HMI_wBrzdit;  
HMI_mSetProgram:=HMI_mSetProgram;  
  
;
```