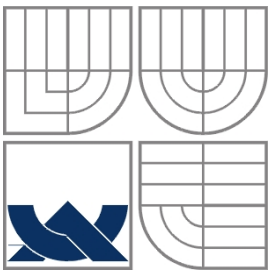


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ŘÍZENÍ MODELŮ PROCESŮ EDU-MOD PROGRAMOVATELNÝM AUTOMATEM ŘADY FX3U

EDU-MOD MODEL CONTROLL BY PROGRAMMABLE LOGIC CONTROLLER FX3U

BAKALÁŘSKÁ PRÁCE
BACHELOR THESIS

AUTOR PRÁCE **DAVID MAREK**
AUTHOR

VEDOUCÍ PRÁCE **ING. TOMÁŠ MARADA, PH.D.**
SUPERVISOR

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): David Marek

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

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

Řízení modelů procesů EDU-mod programovatelným automatem řady FX3U

v anglickém jazyce:

EDU-mod model control by programmable logic controller FX3U

Stručná charakteristika problematiky úkolu:

V laboratoři programovatelných automatů A1/731a jsou čtyři modely soustav EDU-mod. Jedná se o soustavu posuvné jednotky, mysící jednotky, automatické pračky a křižovatky.
Cílem práce je navrhnout a realizovat řízení modelu křižovatky a automatické pračky programovatelným automatem Mitsubishi FX3U. K těmto modelům dále vytvořit vzorová zadání a vypracování. Funkčnost ověřit řízením pomocí PLC.

Cíle bakalářské práce:

1. Seznamte se s modely soustav EDU-mod v laboratoři A1/731a.
2. Seznamte se s programovatelným automatem Mitsubishi FX3U.
3. Vytvořte vzorová zadání s vypracováním řízení úloh.
4. Ověřte funkčnost řízením pomocí PLC.

Seznam odborné literatury:

[1] Šmejkal, L., Martinásková, M., PLC a automatizace, Praha: BEN, 1999.
[2] Firemní materiály o programovatelných automatech fy Mitsubishi FX3U-32M.

Vedoucí bakalářské práce: Ing. Tomáš Marada, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne

L.S.

Ing. Jan Roupec, Ph.D. Ředitel ústavu	prof. RNDr. Miroslav Doupovec, CSc., dr. h. c. Děkan fakulty

;

ABSTRAKT

Tato bakalářská práce byla vytvořena jako podpora výuky na fakultě Strojního inženýrství Vysokého učení technického na Ústavu automatizace a informatiky pro předmět Programovatelné automaty. Cílem této práce bylo seznámit se s modely soustav EDU-mod, programovatelným automatem Mitsubishi FX3U-32M a vývojovým prostředím GX IEC Developer 7.01. Poté vytvořit vzorová zadání s vypracováním řízení úloh a následně ověřit jejich funkčnost pomocí PLC.

ABSTRACT

This work was created as a support for teaching at the Faculty of Mechanical Engineering of the Brno University of Technology at the Institute of Automation and Computer Science for the study of Programmable automatic machines. The aim of this work was to acquaint with the models of EDU-mod system, Mitsubishi FX3U-32M programmable automatic machine and the GX IEC Developer 7.01 development environment, then to create the sample settings with the tasks controlling elaboration and to verify their functionality by PLC consequently.

KLÍČOVÁ SLOVA

EDU-mod, Mistubishi FX3U-32M, GX IEC Developer 7.01, PLC, EasyBuilder, MT6050i

KEYWORDS

EDU-mod, Mitsubishi FX3U-32M, GX IEC Developer 7.01, PLC, EasyBuilder, MT6050i

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem tuto práci vypracoval samostatně, s využitím uvedené literatury a podkladů, na základě konzultací a pod vedením vedoucího bakalářské práce.

BIBLIOGRAFICKÁ CITACE

MAREK, D. *Řízení modelů procesů EDU-mod programovatelným automatem řady FX3U.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 79 s. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D.

V Brně dne 18. 5. 2012

……
Podpis

Poděkování

Nyní bych rád poděkoval vedoucímu mé bakalářské práce panu Ing. Tomášovi Maradovi, Ph.D. za mnohé rady při vytváření této bakalářské práce.

Obsah

;

ÚVOD

V této bakalářské práci se budeme zabývat vytvářením vlastních programů na již existující modely EDU-mod. Máme k dispozici dva modely, přičemž na každý z nich vytvoříme tři funkční programy. Každý program bude realizován ve 4 programovacích jazycích. Od napsání až k ověření funkčnosti těchto programů budeme potřebovat vývojové prostředí, PLC a modely. Jako vývojové prostředí nám poslouží GX IEC Developer. PLC jsme použili Mitsubishi FX3U-32M, z modelů křižovatku a automatickou pračku. Každá část je popsána ve vlastní kapitole.

1 EDU-MOD

EDU-mod jsou modely sloužící k simulaci technologických procesů. Tyto modely slouží především k výuce logických systémů realizovaných pomocí programovatelných automatů (PLC). Dělí se podle napěťové úrovně. Existují 5-ti voltové modely a 24 voltové modely. 5-ti voltové modely se díky logickým signálům s úrovní TTL spojují s PLC realizovanými na bázi stavebnic číslicových I/O, programovatelných polí (PLD) atd. Zatímco 24 voltové modely díky logickým signálům s úrovní 24V ss jsou univerzální a dají se spojit s libovolným PLC. Pro připojení obou modelů použijeme 20-ti žilový kabel. U obou modelů si ukážeme zapojení.[1]

1.1 Křižovatka

Jak již nadpis napovídá, jde o model simulující chování křižovatky. Obsahuje tři semafory. Hlavní, vedlejší a semafor pro přechod pro chodce. Celkem tedy tento modul obsahuje 8 výstupů, které jsou zastoupeny barevnými diodami. Nyní si ukážeme, jak jsou jednotlivé diody přiřazeny k výstupům.[2]

Hlavní semafor: červená, oranžová, zelená

Vedlejší semafor: červená, oranžová, zelená

Přechod pro chodce: červená, zelená

Vodič	Význam	Svorka		Vodič	Význam	Svorka
1	GND	M		11	Červená přechod	Y6
2	24V	L+		12	Zelená přechod	Y7
3	Červená hlavní	Y0		13	x	x
4	Oranžová hlavní	Y1		14	x	x
5	Zelená hlavní	Y2		15	x	x
6	Červená vedlejší	Y3		16	x	x
7	GND	M		17	x	x
8	24V	L+		18	x	x
9	Oranžová vedlejší	Y4		19	x	x
10	Zelená vedlejší	Y5		20	x	x

Tab.1: Zapojení modelu křižovatky pomocí 20-ti žilového kabelu

Obr.1: Křižovatka [2]

1.2 Automatická pračka

Tento model slouží k simulaci chování automatické pračky. Je vybaven 6-ti vstupy a 6-ti výstupy. Pomocí výstupů napouštíme vodu, ohříváme vodu, otáčíme bubnem doleva či doprava, aktivujeme rychlejší otáčení bubnu a nakonec zapínáme čerpadlo, které slouží k vypouštění vody. Přiřazení vstupů a výstupů ke zmiňovaným funkcím můžeme vidět níže.[3]

Vodič	Význam	Svorka		Vodič	Význam	Svorka
1	x	x		11	Čerpadlo	Y6
2	x	x		12	x	x
3	Buben vpravo	Y0		13	x	x
4	Buben vlevo	Y1		14	x	x
5	otáčky	Y2		15	Hladina 50%	X0
6	x	x		16	Hladina 100%	X1
7	GND	M		17	Teplota 90°C	X2
8	24V	L+		18	Teplota 60°C	X3
9	Topení	Y4		19	Teplota 40°C	X4
10	Voda	Y5		20	Teplota 30°C	X5

Tab.2: Zapojení modelu automatické pračky pomocí 20-ti žilového kabelu

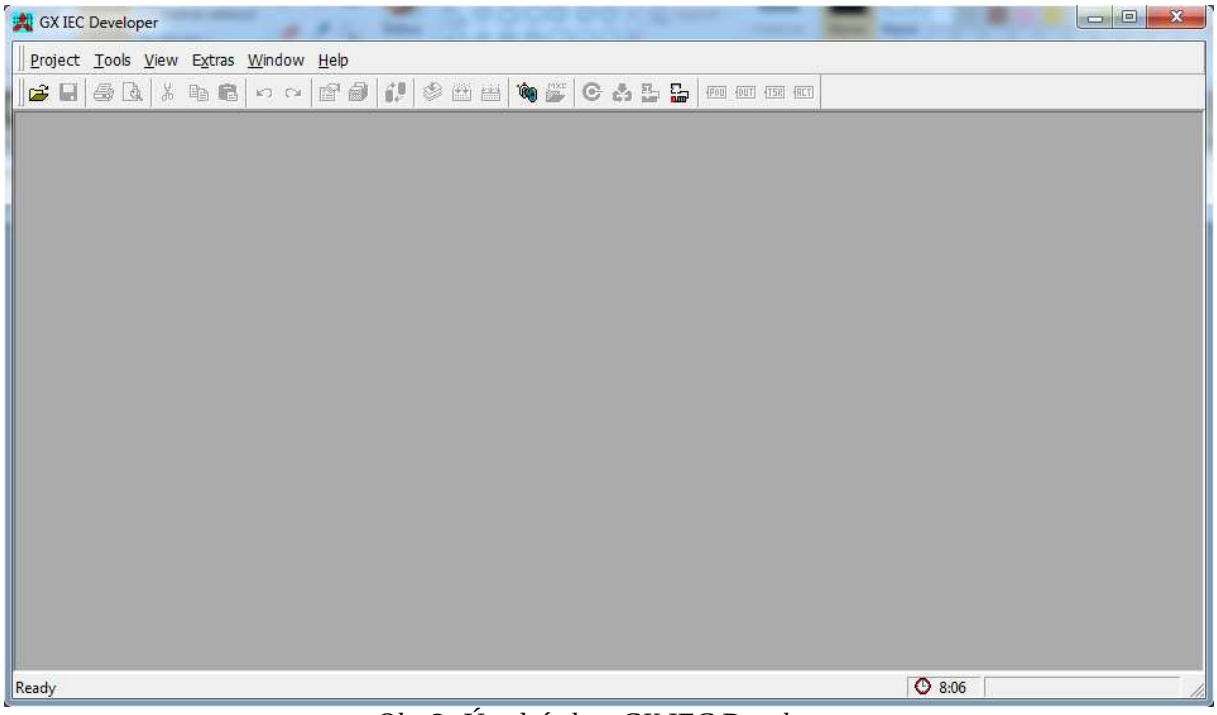
Obr.2: Křižovatka [3]

2 GX IEC DEVELOPER

Vývojové prostředí pro PLC značky Mitsubishi je GX IEC Developer. Vychází z GX Developeru a umožňuje navíc programování normou IEC 61131-3. V této kapitole si popíšeme, jak v něm vytvořit, zkompilovat a nahrát projekt do PLC. GX IEC Developer podporuje 6 programovacích jazyků(IL,ST,FBD,LD,SFC,MELSEC IL). Co tyto zkratky znamenají je podrobně rozebráno dále v textu.

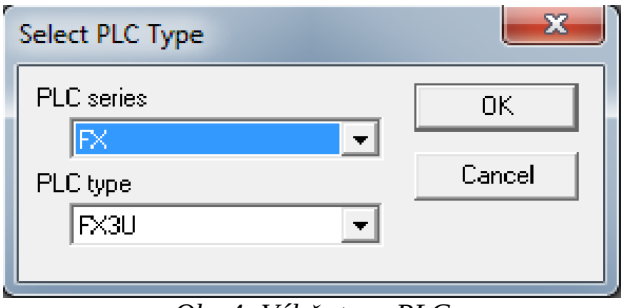
2.1 Vytvoření projektu

Po kliknutí na ikonu GX IEC Developer na vaší ploše vám naskočí *Obr.3*.



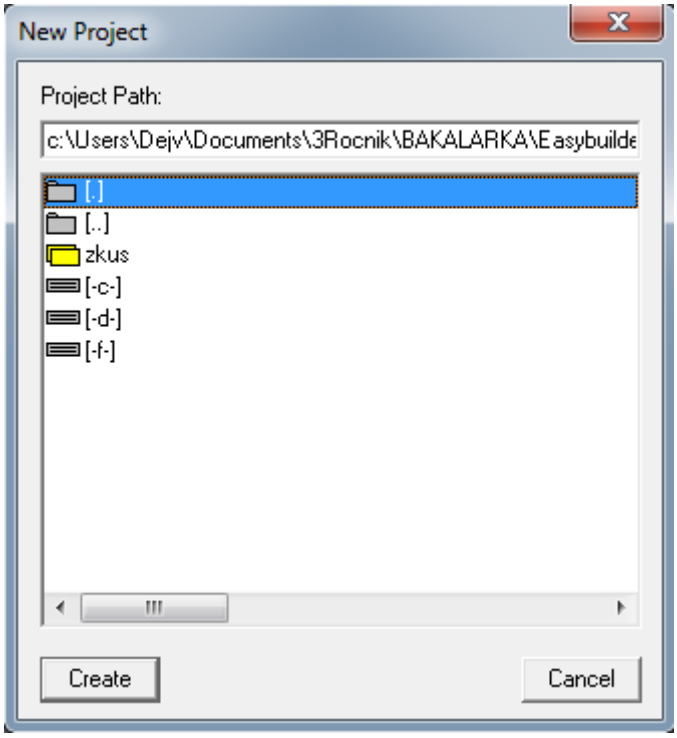
Obr.3: Úvodní okno GX IEC Developer

Nyní klikneme na Project a následně na New. Poté se zobrazí okno s výběrem typu PLC.



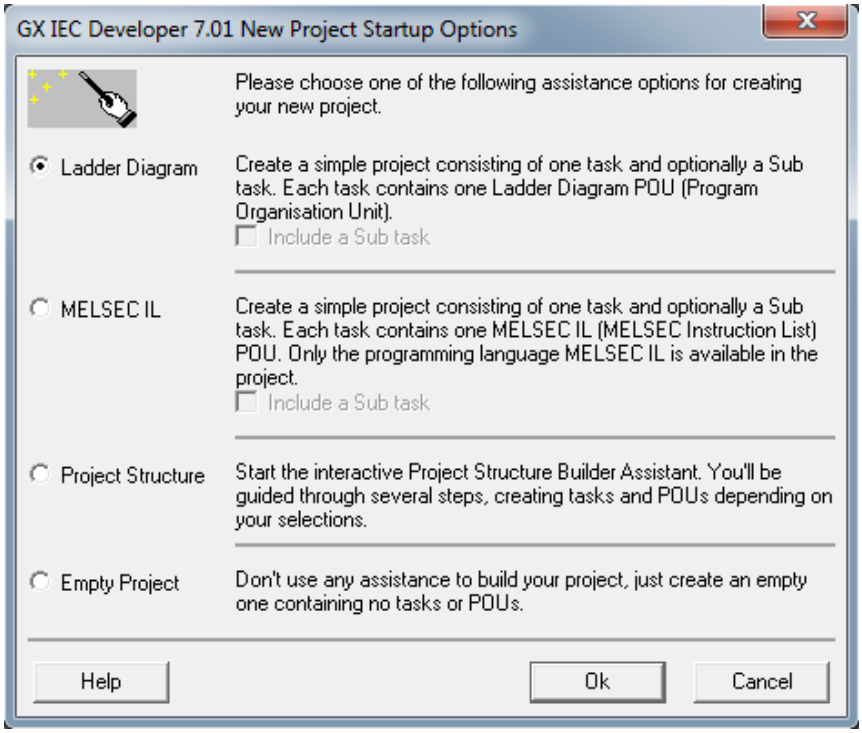
Obr.4: Výběr typu PLC

Jak je vidět na *Obr.4*, vybrali jsme si podle zadání FX3U. Potvrdíme výběr stisknutím tlačítka OK. Následně po nás program chce výběr místa na disku, kam projekt chceme uložit. Viz *Obr.5*.



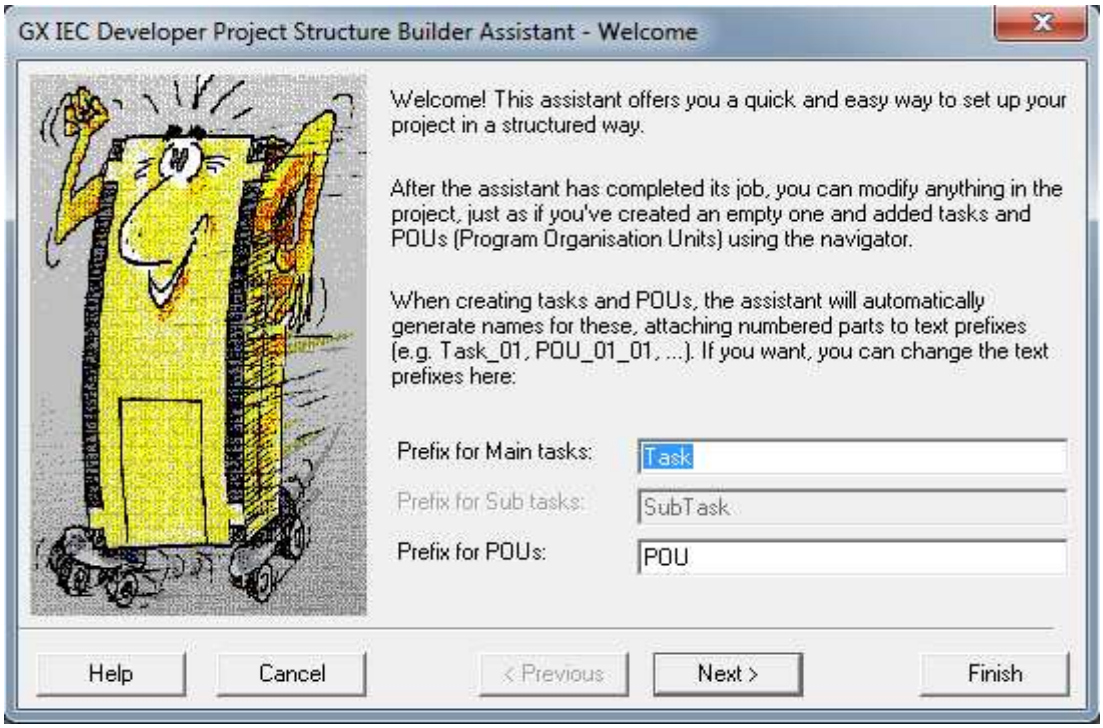
Obr.5: Výběr místa uložení projektu

Po stisknutí tlačítka Create nám program dá vybrat, zda chceme programovat v jazyku LD, MELSEC IL, vytvořit prázdný projekt, či, což je náš případ, si pomocí Project Structure náš projekt krok po kroku nastavit.



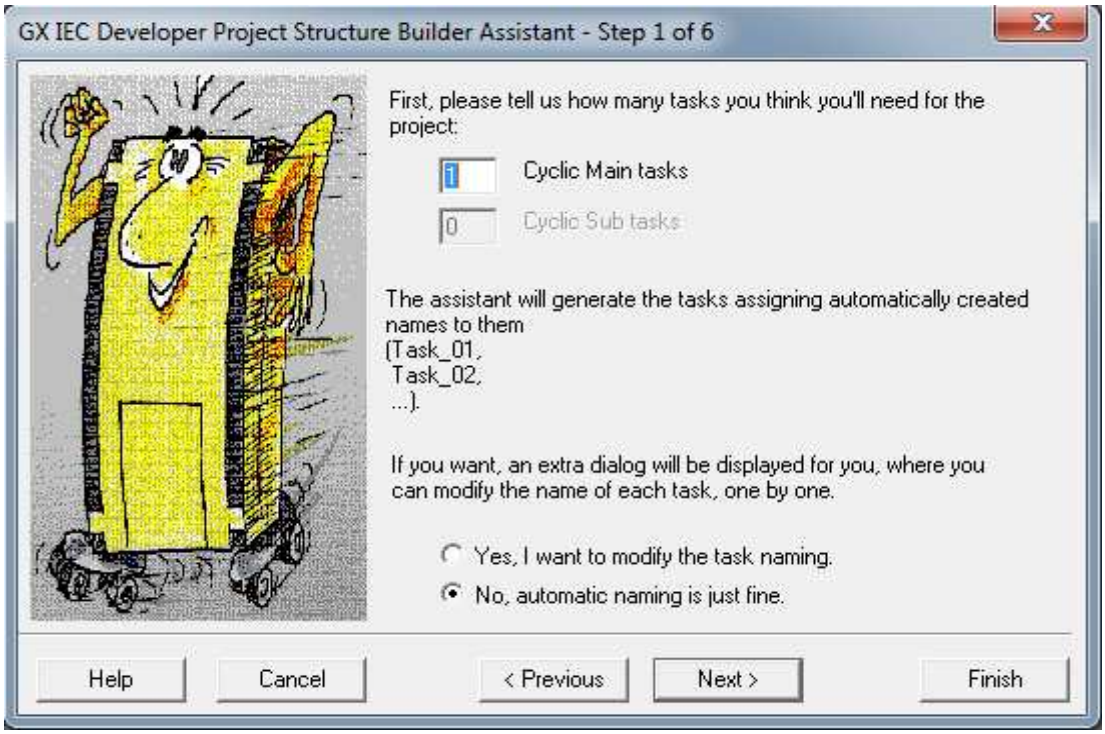
Obr.6: Výběr programovacího jazyka

V tomto kroku zvolíme název skupiny úloh (Task) a skupiny programových modulů (POU). Co Tasky a POU dělají se dozvíme v kapitolách 3.2.1. a 3.2.2..



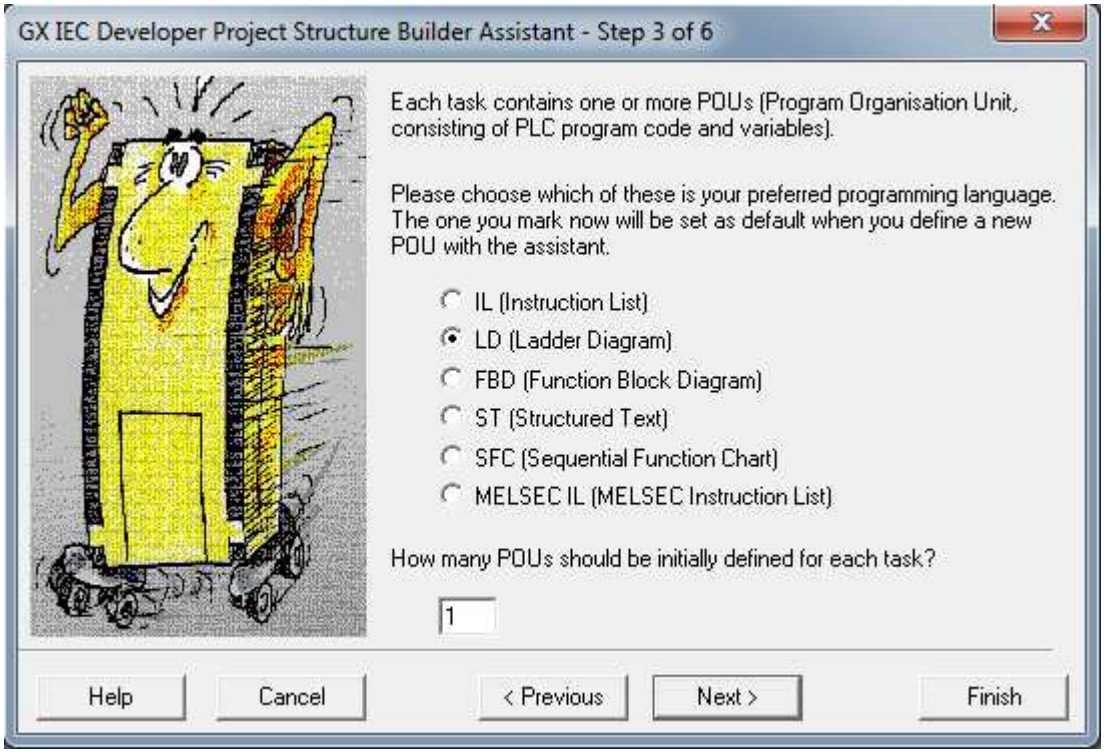
Obr.7: Výběr názvu Task a POU

V následujícím okně si můžeme zvolit počet úloh, které si myslíme, že budeme potřebovat pro náš projekt a zvolíme automatické, nebo ruční pojmenování úloh. Jelikož se ve vytvořeném projektu dají úlohy pohodlně přidávat i přejmenovat, vybereme možnost NO, automatic naming is just fine, čímž se nám budou úlohy automaticky pojmenovávat Task1, Task2, atd..



Obr.8: Výběr počtu úloh

Jak vidíme na *Obr.9*, tak si kromě LD a MELSE IL můžeme vybrat mezi IL, FBD, ST a SFC, které jsou definovány normou IEC 61131-3. Kromě toho nastavíme počet POU pro každý Task.



Obr.9: Výběr programovacího jazyka

2.1.1 Přehled programovacích jazyků

Textové:

IL(Instruction List): Nízkoúrovňový jazyk, je velmi rychlý, skládá se z posloupnosti instrukcí a je podobný assembleru. Při rozsáhlejších úlohách se stává nepřehledným

ST(Structured Text): Vyšší programovací jazyk než IL, podobný jazyku Pascal a částečně i jazyku C. Je vhodný zejména pro implementaci složitějších výpočetních algoritmů.

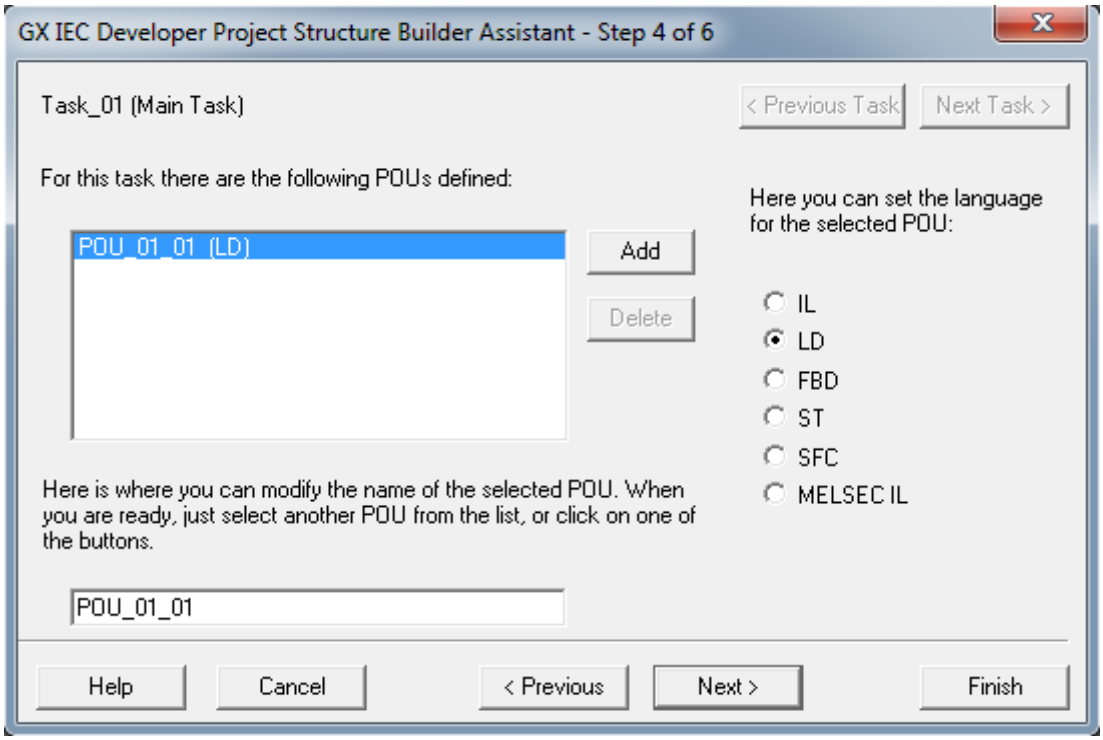
Grafické:

FBD(Function Block Diagram): Jedná se o zápis funkčních bloků. Přehledný i při velkém počtu logických signálů.

LD(Ladder Diagram): Nejrozšířenější programovací jazyk pro PLC. Jde o jazyk kontaktních schémat, jedná se o grafickou reprezentaci IL.

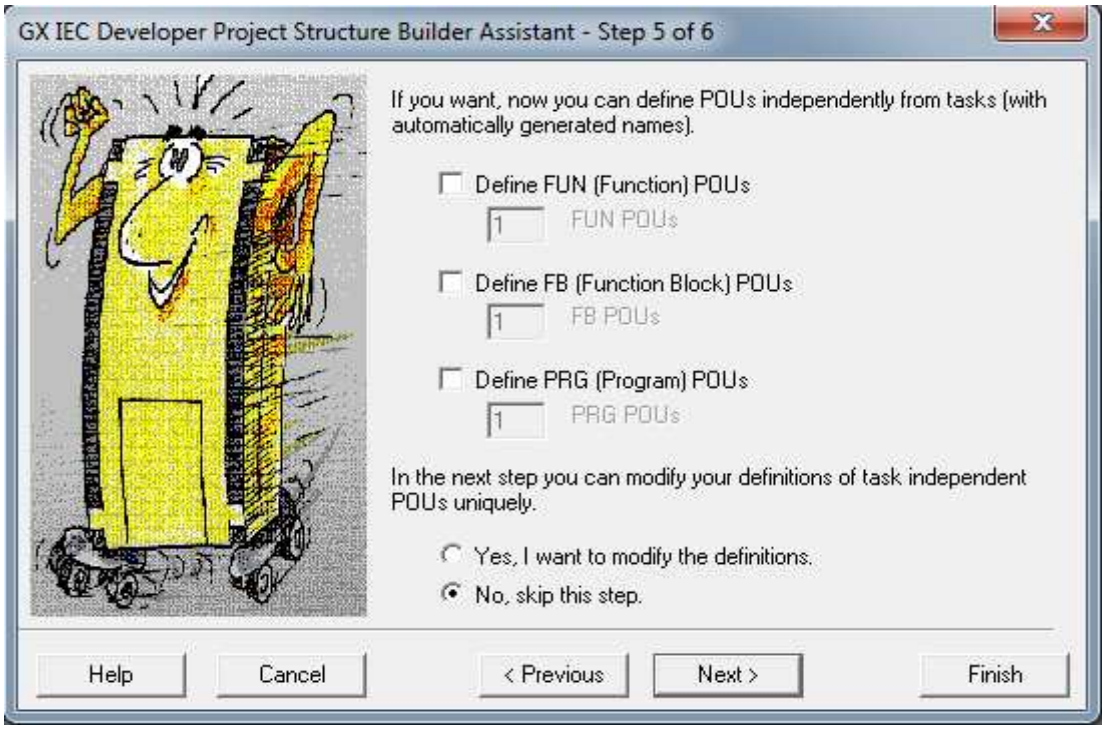
SFC(Sequential Function Chart): Jedná se o vývojový diagram. Umožňuje snadnou definici chování programu.

V tomto kroku nám program umožňuje zvolit pro každou POU jiný programovací jazyk, jestliže chceme programovat celý projekt v jednom jazyku, tak přejdeme na další krok pomocí tlačítka Next.



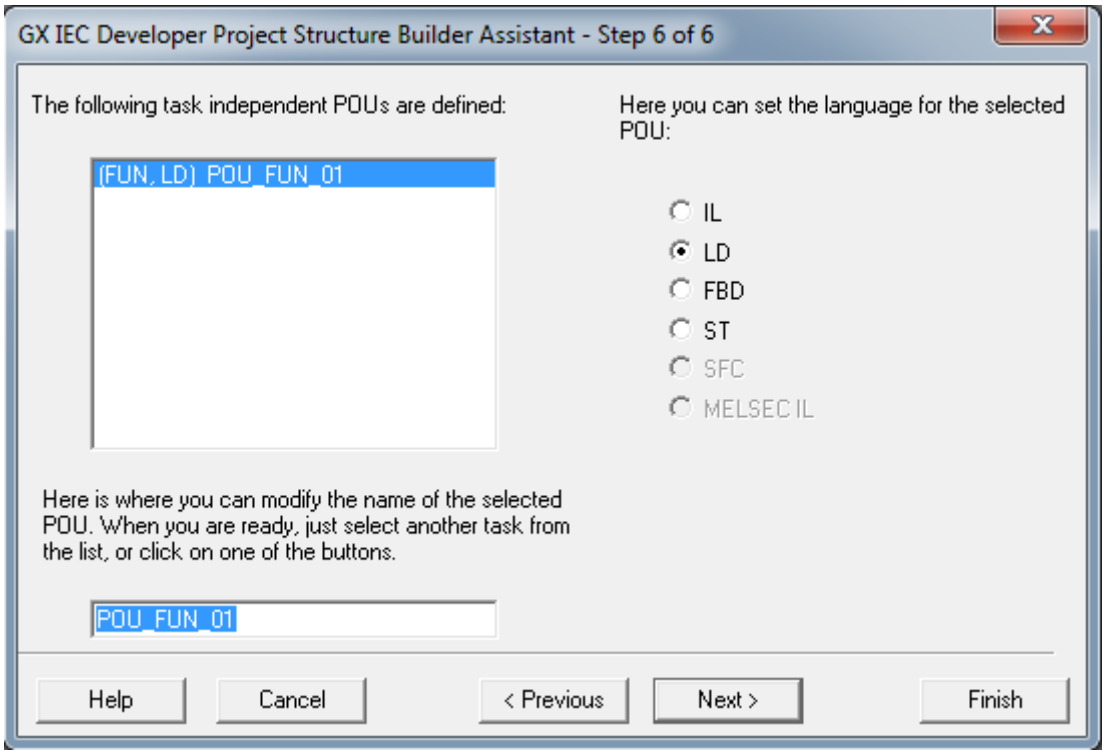
Obr.10: Výběr programovacího jazyka pro POU

Zde se vytváří POU nezávislé na v předešlých krocích vytvořených Úlohách. Vybíráme mezi funkcí, funkčním blokem a programem. Po zaškrtnutí některé z nich ještě vybereme počet těchto POU a nakonec, zda je chceme definovat.



Obr.11: Nezávislé POU

Zde definujeme POU, které jsme v předchozím kroku vybrali a nastavujeme jim jazyk, který budou používat. Vše dokončíme pomocí tlačítka Finish.



Obr.12: Definice nezávislých POU

2.2 Popis projektu

V následujících kapitolách si popíšeme vlastnosti GX IEC Developeru. Dále si ukážeme nastavení připojení k PC a syntaxi zápisu logických funkcí ve čtyřech programovacích jazycích.

2.2.1 Task Pool

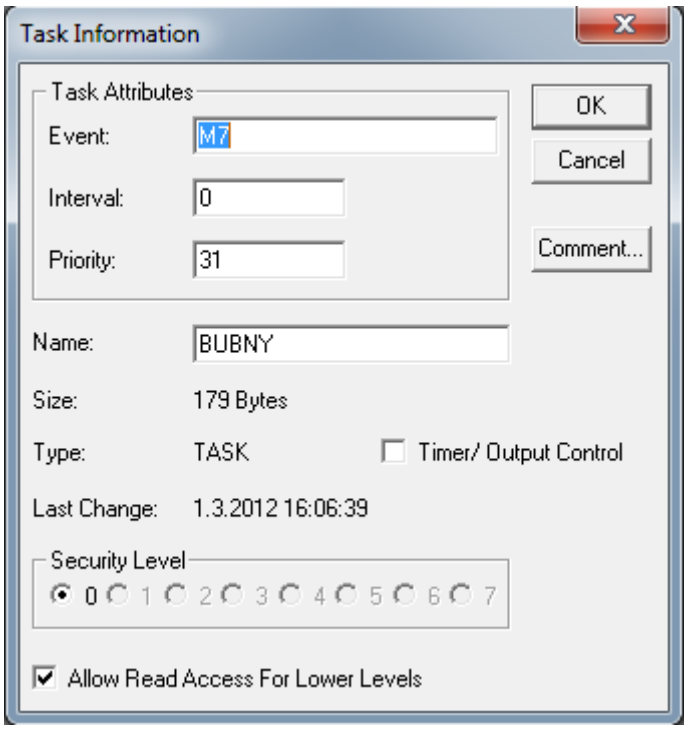
Task řídí jednotlivé programové moduly (POU) pomocí podmínky, kterou v úloze definujeme.
Do nastavení podmínky se dostaneme přes Properties, které nalezneme po stisknutí pravého tlačítka myši na Task, nebo ho pustíme pomocí ALT+ Enter. Přiřazení POU Tasku se provádí dvojklikem na daný Task a výběrem POU. K vytvoření nového Tasku slouží ikona 

, nebo pravým tlačítkem na Task Pool a vybrat New Task. Nyní si popíšeme vlastnosti Tasku.

Event: Pokud je TRUE, tak je úloha aktivovaná vždy, pokud je zvolena proměnná, tak při její náběžné hraně. Jestliže je zvoleno FALSE, znamená to, že je nastaveno časové zpracování.

Interval: Interval v jakém se úloha spouští. Pouze v případě, že v Event je zvoleno FALSE.

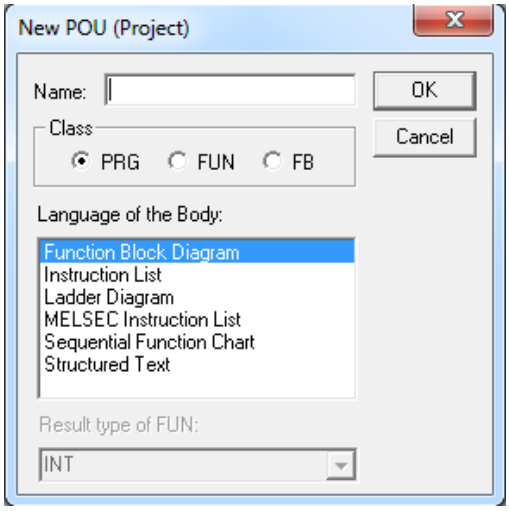
Priority: Při požadavku na aktivaci více úloh závisí na prioritě. 31 je priorita nejnižší, 1 nejvyšší.



Obr.13: Definice podmínky POU

2.2.2 POU Pool

POU se skládá ze dvou částí. Body a Header. V Body je samotný program a v Header jsou proměnné, které použijeme v daném POU. Pro proměnné použitelné ve všech POU slouží Global Vars. Pro vytvoření nového POU použijeme ikonu  , nebo pravým tlačítkem na POU Pool a zvolíme New POU. Při vytváření nové POU zvolíme název, jazyk, kterým budeme danou POU programovat a zda bude vytvořen jako program, funkce, nebo funkční blok.



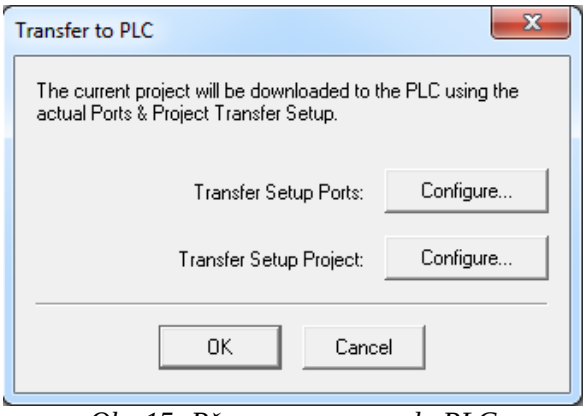
Obr.14: Vytváření nové POU

2.3 Kompilace

Dříve, nežli nahrajeme program do PLC, musíme ho zkompilovat. Kompilace zajišťuje jak hledání chyb v programu, tak překlad programu do podoby srozumitelné PLC. Kompilace probíhá na třech úrovních. Check, Build a Rebuild All.

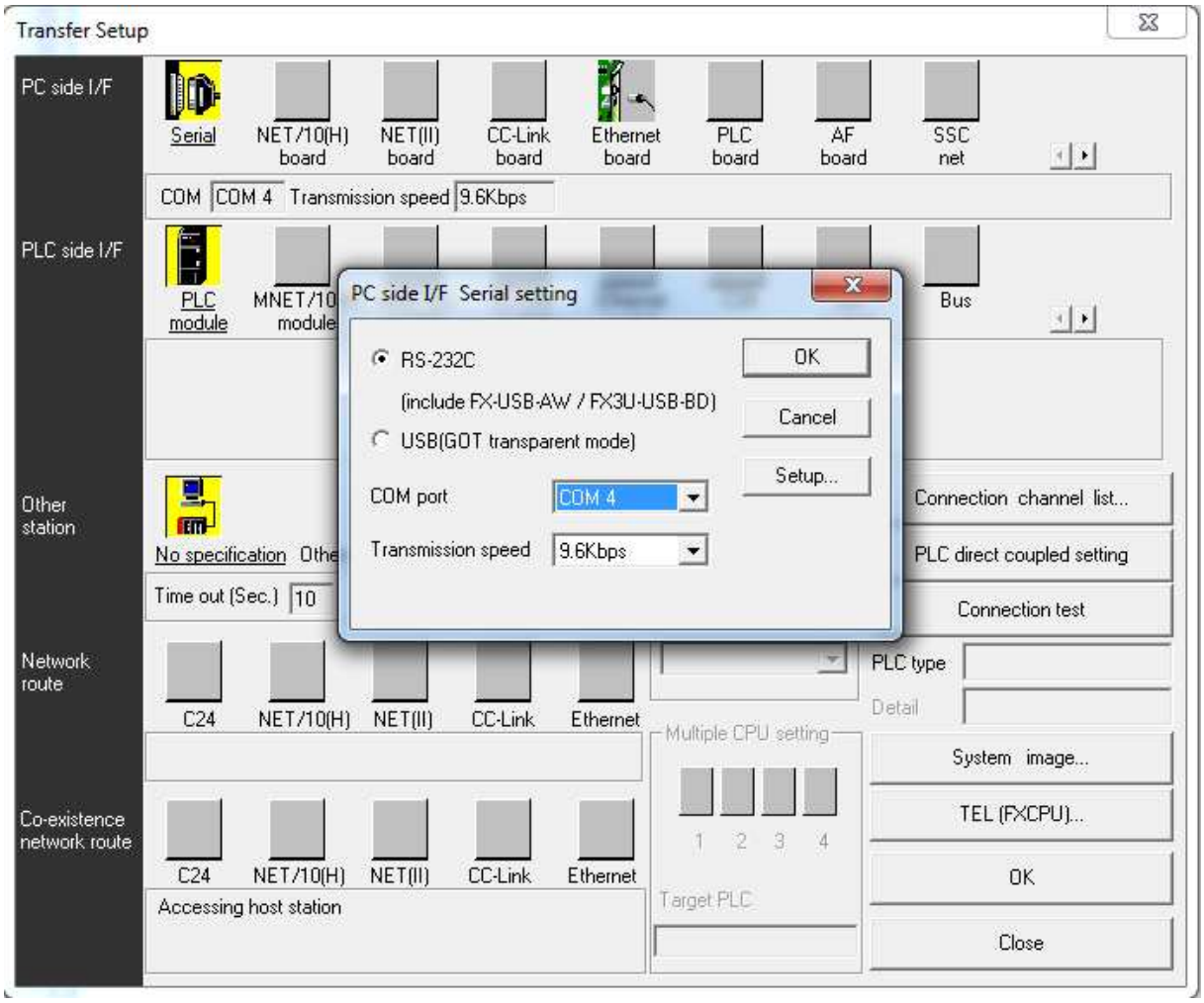
2.4 Download Project

Pomocí Download Project nahrajeme náš program do PLC. Spustí se pomocí ikony  . Po spouštění Download Project si vybereme mezi nastavením portů pro přenos(Transfer Setup Ports), nastavením přenosu projektu(Transfer Setup Project) a posláním projektu do PLC(OK).



Obr.15: Přenos programu do PLC

Po stisknutí Configure u Transfer Setup Ports a následně Serial máme možnost nastavit, zda chceme data přenášet přes USB nebo RS-232C. U RS-232C nastavujeme port COM a rychlost přenosu dat. Přes Setup se nastavuje parita, počet datových bitů a počet stop bitů. Přes Connection Test si vyzkoušíme, zda nám PC komunikuje s PLC.



Obr.16: Nastavení komunikace

2.5 Upload Project

Upload Projekt je opakem Download Projektu. Stahuje nám nahraný program do PC.

2.6 Režimy PLC

PLC může být ve dvou režimech. RUN a STOP. Pokud je PLC v režimu RUN, je program v chodu. Pokud je v režimu STOP, program se přeruší. Ke změně režimu PLC slouží Start/Stop PLC, které je ukryto v záložce Online. Zkratka pro spuštění je ALT+S. Tlačítko Refresh slouží ke zjištění režimu, v jakém se PLC momentálně nachází. Pomocí Execute změníme režim PLC na zvolený.

Obr.17: Změna režimu PLC

2.7 Použité elementy

V této kapitole si ukážeme zápis použitých elementů normy IEC 61 131-3. Každý prvek bude v jazyce IL, ST, LD a FBD.

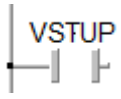
2.7.1 LD

Načte operand.

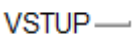
IL:
LD VSTUP

ST:
IF VSTUP

LD:



FBD:



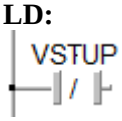
2.7.2 LDN

Načte negovaný operand.

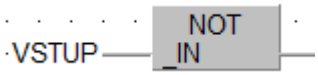
IL:
LDN VSTUP

ST:
IF NOT VSTUP

LD:



FBD:



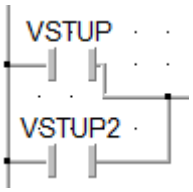
2.7.3 OR

Logický součet.

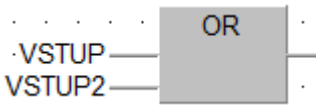
IL:
LD VSTUP
OR VSTUP2

ST:
IF VSTUP OR VSTUP2

LD:



FBD:



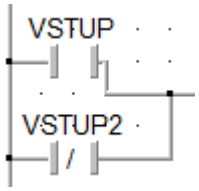
2.7.4 ORN

Logický negovaný součet.

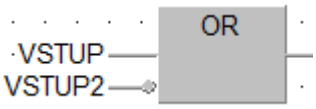
IL:
LD VSTUP
ORN VSTUP2

ST:
IF VSTUP OR NOT VSTUP2

LD:



FBD:



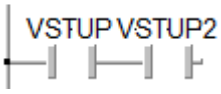
2.7.5 AND

Logický součin.

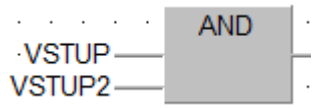
IL:
LD VSTUP
AND VSTUP2

ST:
IF VSTUP AND VSTUP2

LD:



FBD:



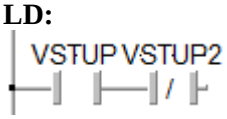
2.7.6 ANDN

Logický negovaný součin.

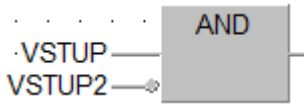
IL:
LD VSTUP
AND VSTUP2

ST:
IF VSTUP AND VSTUP2

LD:



FBD:



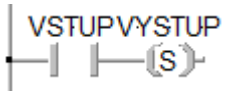
2.7.7 SET

Nastaví na výstupu logickou 1.

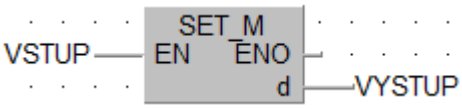
IL:
LD VSTUP
S VYSTUP

ST:
SET_M(VSTUP,VYSTUP);

LD:



FBD:



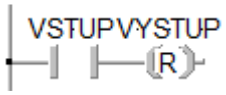
2.7.8 RESET

Nastaví na výstupu logickou 0.

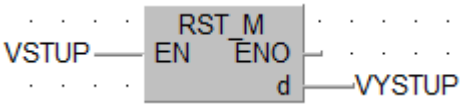
IL:
LD VSTUP
R VYSTUP

ST:
RST_M(VSTUP,VYSTUP);

LD:



FBD:

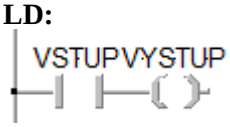


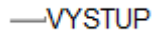
2.7.9 ST

Uložení výsledku na výstup.

IL:
LD VSTUP
ST VYSTUP

ST:
VYSTUP:=VSTUP;

LD:


FBD:


2.7.10 TON (Timer On Delay)

Zpožděné sepnutí po náběžné hraně. Na vstupu musí být po celou dobu nastaveného PT logická 1.

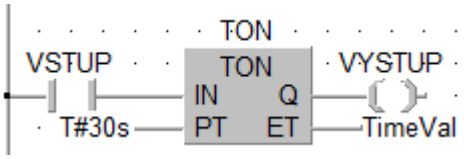
IN- Reakce náběžné hrany
PT- Přednastavený čas
ET- Aktuální hodnota uplynulého času
Q- Výstup

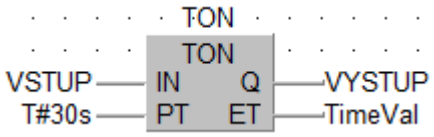
IL:
CAL TON(PT:=T#30s)

LD VSTUP
ST TON.IN

LD TON.Q
ST VYSTUP

ST:
TON(PT:=T#30s);
TON.IN:=VSTUP;
VYSTUP:=TON.Q;

LD:


FBD:


2.7.11 TOF (Timer Off Delay)

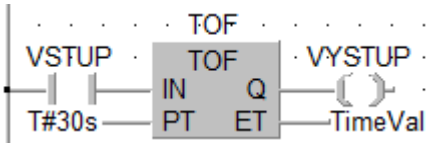
Zpožděné vypnutí po sestupné hraně.

IN- Reakce na sestupnou hranu
PT- Přednastavený čas
ET- Aktuální hodnota uplynulého času
Q- Výstup

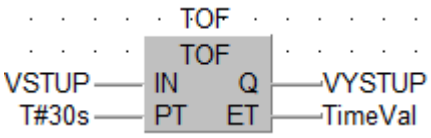
IL:
```
CAL TOF(PT:=T#30s)

LD VSTUP
ST TOF.IN

LD TOF.Q
ST VYSTUP
```

ST:
```
TOF(PT:=T#30s);
TOF.IN:=VSTUP;
VYSTUP:=TOF.Q;
```

LD:



FBD:



2.7.12 TP (Timer Pulse)

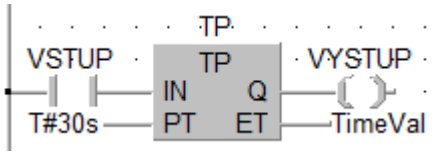
Po náběžné hraně nastaví na přednastavený čas logickou 1 na výstup.

IN- Reakce na náběžnou hranu
PT- Přednastavený čas
ET- Aktuální hodnota uplynulého času
Q- Výstup

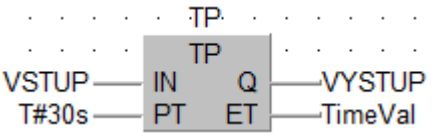
IL:
```
CAL TP(PT:=T#30s)

LD VSTUP
ST TP.IN

LD TP.Q
ST VYSTUP
```

ST:
TP(PT:=T#30s);
TP.IN:=VSTUP;
VYSTUP:=TP.Q;

LD:



FBD:



2.7.13 Paměti

M8000: Relé, které má po celou dobu běhu programu hodnotu 1.

M8001: Relé, které je po celou dobu nastaveno na 0.

M8002: Relé, které má hodnotu 1 po dobu jednoho cyklu programu.

M0- M499: Nezamčená relé (Po vypnutí napájení jsou resetována na 0). Možno nakonfigurovat jako zamčené relé.

M500- M1023: Zamčená relé (Zachovávají si svůj původní stav). Možno nakonfigurovat jako nezamčené relé.

M1024- M7679: Zamčená relé.

3 MITSUBISHI FX3U-32M

Základní informace o zvoleném plc jsou vidět v *Tab.3*. Tato základní jednotka může být doplněna o rozšiřovací vstupní/výstupní moduly, nebo o speciální funkční moduly. Jaká rozšíření jsou možná se dozvíme v kapitole 3.1.[4]

Obr.18: Mitsubishi FX3U-32M[4]

Zdroj napájení	Spotřeba energie	Hmotnost	Rozměry (mm) šířka/výška/hloubka
24V DC	30 W	0,65	150/90/86
Integrované (I/O)	**Integrované vstupy (I)**	**Integrované výstupy (O)**	**Typ výstupu**
32	16	16	Tranzistor

Tab.3: Specifikace Mitsubishi FX3U-32M

3.1 Možná rozšíření:

Vnitřní rozšíření základní jednotky: Digitální, Analogové.

Rozšiřovací moduly: Digitální, Analogové, Teplota.

Síťové moduly: AS-Interface, CC-Link, CAN open, Ethernet, Profibus DP, DeviceNet, Modbus RTU/ASCII, SSCNET.

Komunikační karty: RS232, RS422, RS485, USB.

Komunikační moduly: RS232, RS485.

Vyhrazené funkční moduly: Vysokorychlostní čítače, Polohování.

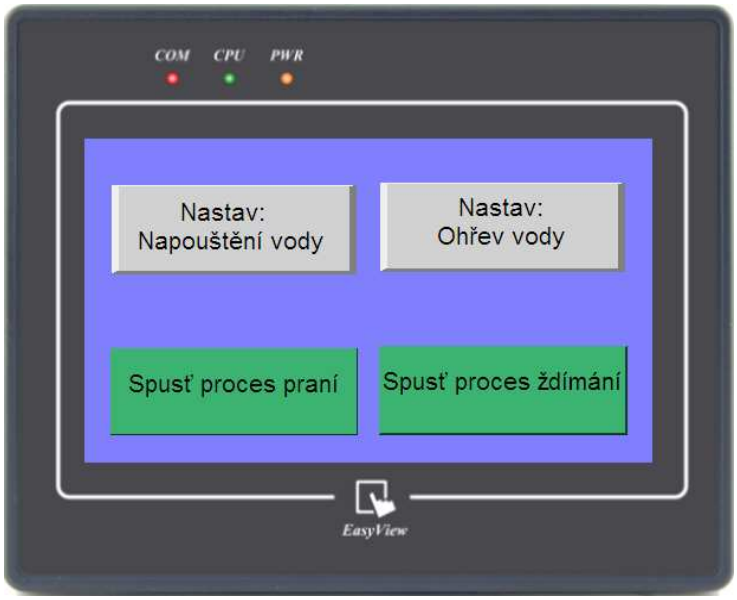
Další: Paměťové kazety, Externí displej.

3.2 Bližší specifikace:

- Paměť až pro 6400 kroků
- 6 čítačů pro 100kHz a 2 čítače pro 10kHz pro zpracovávání vstupních pulzů
- Až 384 vstupů a výstupů
- Hodiny reálného času
- Integrovaný spínač RUN/STOP
- Integrované sériové rozhraní

4 GRAFICKÝ PANEL

Jedná se o panel EasyView série 6000 s dotykovou obrazovkou. Je vybaven třemi kontrolkami umístěnými v horní části. Červená kontrolka s nápisem COM signalizuje probíhající komunikaci panelu s připojeným zařízením. Zelená kontrolka CPU signalizuje běh programu v panelu a oranžová kontrolka PWR signalizuje napájení panelu. Při nečinnosti je osvětlení displeje automaticky vypnuto a displej je přepnut do úsporného režimu (spořič obrazovky). Osvětlení se zapne automaticky po prvním dotyku na stínítko dotykové obrazovky. Funkce signalizačních LED není ovlivněna přechodem panelu do úsporného režimu. Pro komunikaci mezi panelem a automatem nepotřebujeme žádný speciální program v PLC, jedná se o tzv. přímou komunikaci. K panelu lze připojit např. Allen Bradley, Siemens, Mitsubishi a další. Navíc podporuje protokoly viz. Modbus RTU/ASCII atd.. Programuje se pomocí softwaru EasyBuilder 8000.[5]



Obr.19:Panel EasyView MT6050i [5]

4.1 Specifikace

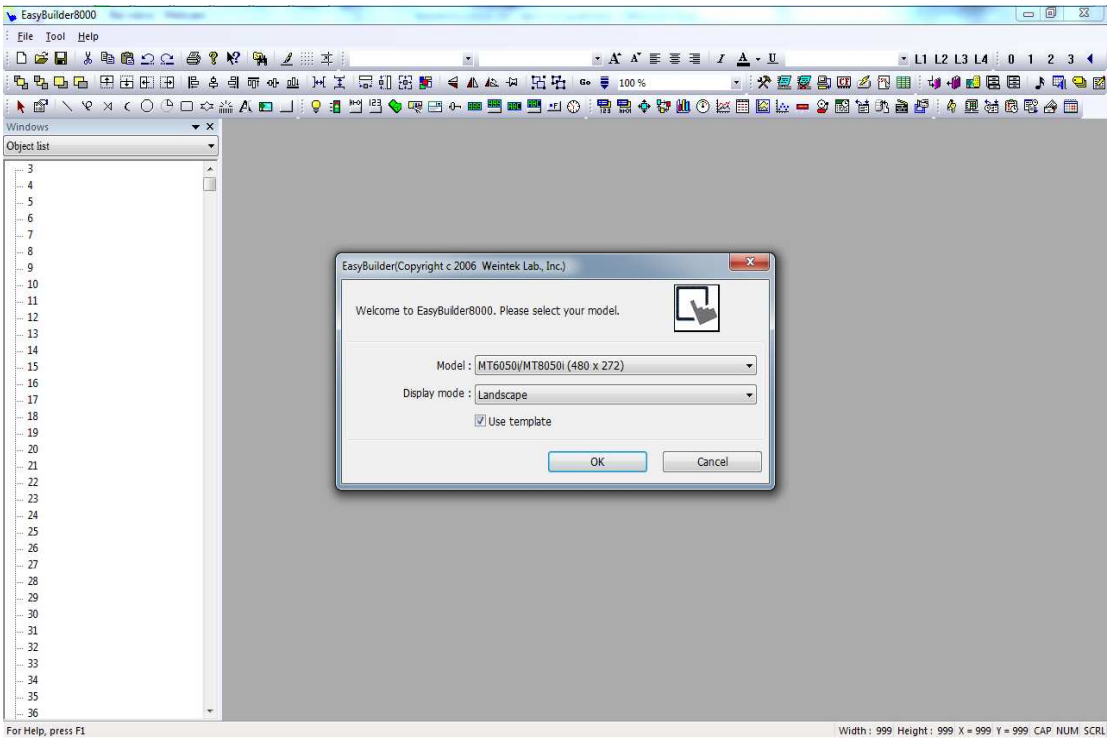
- Napájení 24V DC
- 4,3“ TFT LCD Displej
- 65536 barev
- Rozlišení 480 x 272
- 400 MHz
- 64 MB DDR2
- 128 MB Flash paměť
- Serial port vstup (I) COM1: RS-232 RS-485/ 2w/4w
- USB

5 EASYBUILDER 8000

EasyBuilder je konfigurační software určený k programování dotykových panelů od firmy Weintek. Jedna z jeho výhod je online a offline simulace na PC. Díky tomu není zapotřebí při každé menší změně v softwaru přehrávat program do panelu. Má široký výběr mezi připojitelnými PLC.[6]

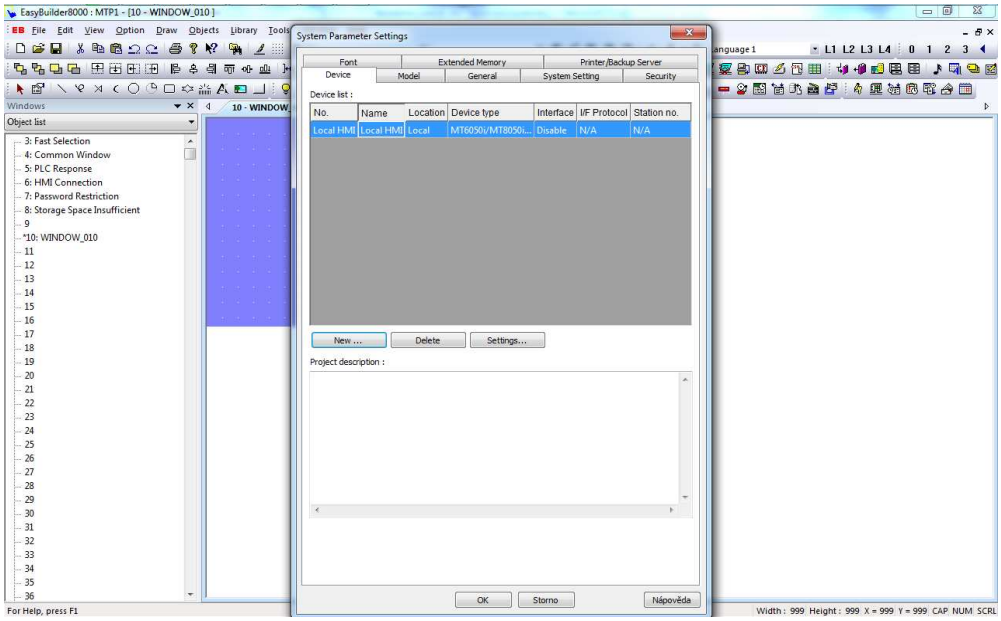
5.1 Vytvoření projektu

Po spuštění programu vybereme druh panelu, režim zobrazení a zda použijeme šablonu.



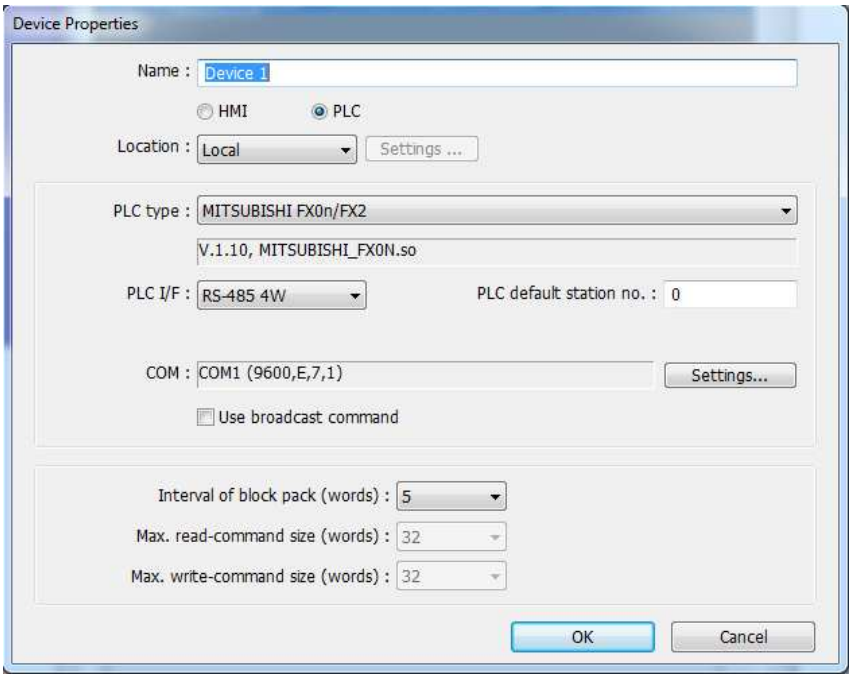
Obr.20: Otevření EasyBuilderu

Poté, co vybereme panel, se zobrazí okno s nastavením parametrů systému, viz *Obr.21*. Pomocí tlačítka New zvolíme nové zařízení, se kterým bude panel komunikovat. Tlačítkem Settings zobrazíme vlastnosti již vytvořeného zařízení a tlačítkem Delete toto zařízení odstraníme.



Obr.21: Nastavení parametrů systému

V položce Name vyplníme zvolený název zařízení. Další je výběr mezi HMI a PLC. Jestliže zvolíme HMI, panel nám bude komunikovat, řídit, nebo bude řízen jiným panelem. V tomto případě je potřeba nastavit pouze umístění, IP adresa, port a interval of block pack. Zvolíme-li PLC, nastavíme umístění (místní, dálkový), typ PLC, komunikaci (kapitola 5.1), port COM, možnost použít ovládání přenosu, Interval of block pack.



Obr. 22: Nastavení komunikace

5.2 Použité objekty

V této kapitole si ukážeme prvky, které jsme použili pro vytvoření projektu, kompilaci a poslání do panelu.

			
Bit Lamp	Word Lamp	Set Bit	Set Word
			
Toggle Switch	Slider	Numeric Display	Numeric Input
			
ASCII Display	ASCII Input	Function Key	Timer
			
Compile	On_line Simulation	Off-line Simulation	Download

Tab.4: Elementy EasyBuilder

Bit Lamp- Tento objekt nabývá stavu ON/OFF podle stavu čtené adresy. Tento stav nabývá hodnot 0/1

Word Lamp- Zobrazuje se v závislosti na hodnotě čtené adresy, jejíž maximum je 256.

Set Bit- Nabízí dvě varianty ovládání. Ruční a automatický. V automatickém režimu definujeme podmínky, za kterých bude automaticky zapisována na vybranou adresu 0 nebo 1. V ručním toto nastavujeme dotykovou plochou.
Set Word- Stejně jako Set Bit nabízí ruční a automatické ovládání. V ručním režimu pomocí dotykové plochy přičítáme, odečítáme, nebo nastavujeme hodnotu. Automatickým režimem se tyto akce provádějí pomocí předem definovanými podmínkami.

Toggle Switch- Jedná se o kombinaci Bit Lampy a Set Bitu. Objekt nejen pomocí dotykové plochy nabývá hodnot 0/1, ale může být použit k jeho zobrazení.

Slider - Pomocí přetažení myší mění hodnotu zapisované adresy. Hodnota je typu word.

Numeric Display - Zobrazuje hodnotu čtené adresy.

Numeric Input - Zobrazuje a zapisuje hodnotu na zvolenou adresu.

ASCII Display - Zobrazuje hodnotu čtené adresy v ASCII znacích.

ASCII Input - Zobrazuje a zapisuje hodnotu v ASCII znacích na vybranou adresu.

Function Key - Používá se k přepínání mezi okny, nebo jejich zavření. Lze použít i k návrhu tlačítek klávesnice.

Timer - Provádí akce se zpožděním (zpožděné zapnutí, vypnutí, atd.).

Compile - Převede program do tvaru, který je pro panel srozumitelný.

On-line Simulation - Slouží k simulaci změn v programu, aniž by se musel nahrávat do panelu. Např. změny v komunikaci. Musí být připojen panel.

Off-line Simulation - Slouží k simulaci chování programu. Nemusí být připojen panel.

Download- Odešle program do panelu.

6 VYTVOŘENÉ ÚLOHY

Ke každému EDU-modelu byly vytvořeny tři úlohy, přičemž každá z nich byla naprogramována ve čtyřech programovacích jazycích.

6.1 Křižovatka

Protože jsem hardwarovou stránku křižovatky popsal již v kapitole 1.1, budu se zde zabývat spíše softwarovou stránkou. Zdrojové kódy jsou v příloze, z toho důvodu uvedu v následujících kapitolách použité proměnné, časovače, rozložení podprogramů a vůbec celkový průběh řešení v GX IEC Developeru. Místo zdrojových kódů jsou zde vývojové diagramy. *Tab.5 a Obr.23* znázorňují rozdíl mezi označením adres u modelu a u automatu.

Význam	Proměnná	Adresa
Červená hlavní	CE_HL	Y0
Oranžová hlavní	OR_HL	Y1
Zelená hlavní	ZE_HL	Y2
Červená vedlejší	CE_VE	Y3
Oranžová vedlejší	OR_VE	Y4
Zelená vedlejší	ZE_VE	Y5
Červená přechod	CE_PR	Y6
Zelená přechod	ZE_PR	Y7

Tab.5: Globální proměnné 1. Úloha křižovatka

6.1.1 Zadání 1. úlohy

V první úloze bylo za úkol naprogramovat simulaci chování křižovatky ve dne. Diody se na křižovatce střídají ve stejném sledu jako ve skutečnosti. Pro urychlení simulace jsem naprogramoval střídání KROKŮ po 3 sekundách. KROKY znázorňují, které diody se mají v jakou chvíli rozsvítit. Vše je zobrazeno v *Tab.6.*

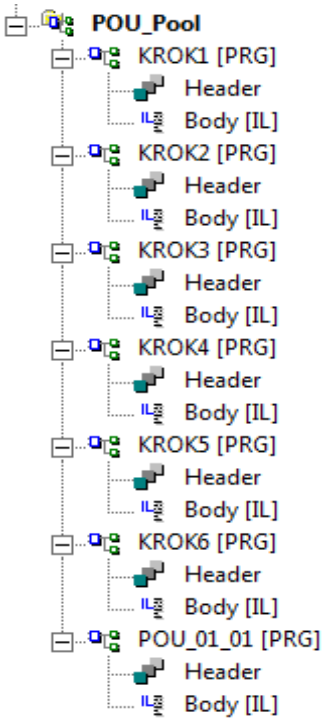
6.1.2 Řešení 1.úlohy

Nyní si tedy ukážeme vytvořený program v GX IEC Developeru. Přiřazení proměnných použitelných v celém projektu vidíme v Global Vars. Jak je vidět na *Obr.24*, tak jsem úlohy řídil pomocí pamětí. Podle toho, která paměť byla načtená, byl použit přiřazený KROK. Task_01 řídí POU_01_01, které ovládá přepínání mezi KROKY, proto je jeho hodnota TRUE. Protože se jedná o jedinou úlohu mající hodnotu TRUE, je jasné, že se jedná o hlavní program ovládající zbylé podprogramy.

	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment	Remark
0	CE_HL	Y0	%QX0	BOOL	...	FALSE	Červená hlavní	
1	OR_HL	Y1	%QX1	BOOL	...	FALSE	Oranžová hlavní	
2	ZE_HL	Y2	%QX2	BOOL	...	FALSE	Zelená hlavní	
3	CE_VE	Y3	%QX3	BOOL	...	FALSE	Červená vedlejší	
4	OR_VE	Y4	%QX4	BOOL	...	FALSE	Oranžová vedlejší	
5	ZE_VE	Y5	%QX5	BOOL	...	FALSE	Zelená vedlejší	
6	CE_PR	Y6	%QX6	BOOL	...	FALSE	Červená přechod	
7	ZE_PR	Y7	%QX7	BOOL	...	FALSE	Zelená přechod	

Obr.23: Global Vars denního režim křižovatky

Obr.24: Task Pool denního režimu křižovatky



Obr.25: POU Pool denního režimu křižovatky

	Class	Identifier	Type		Initial	Comment
0	VAR	KROK2	TON	...		Zpožděné zapnutí druhého kroku
1	VAR	KROK3	TON	...		Zpožděné zapnutí třetího kroku
2	VAR	KROK4	TON	...		Zpožděné zapnutí čtvrtého kroku
3	VAR	KROK5	TON	...		Zpožděné zapnutí pátého kroku
4	VAR	KROK6	TON	...		Zpožděné zapnutí šestého kroku
5	VAR	KROK1	TON	...		Zpožděné zapnutí prvního kroku
6	VAR	TimeVal1	TIME	...	T#0s	Ukládání uběhlého času kroku 2
7	VAR	TimeVal2	TIME	...	T#0s	Ukládání uběhlého času kroku 3
8	VAR	TimeVal3	TIME	...	T#0s	Ukládání uběhlého času kroku 4
9	VAR	TimeVal4	TIME	...	T#0s	Ukládání uběhlého času kroku 5
10	VAR	TimeVal5	TIME	...	T#0s	Ukládání uběhlého času kroku 6
11	VAR	TimeVal6	TIME	...	T#0s	Ukládání uběhlého času kroku 1

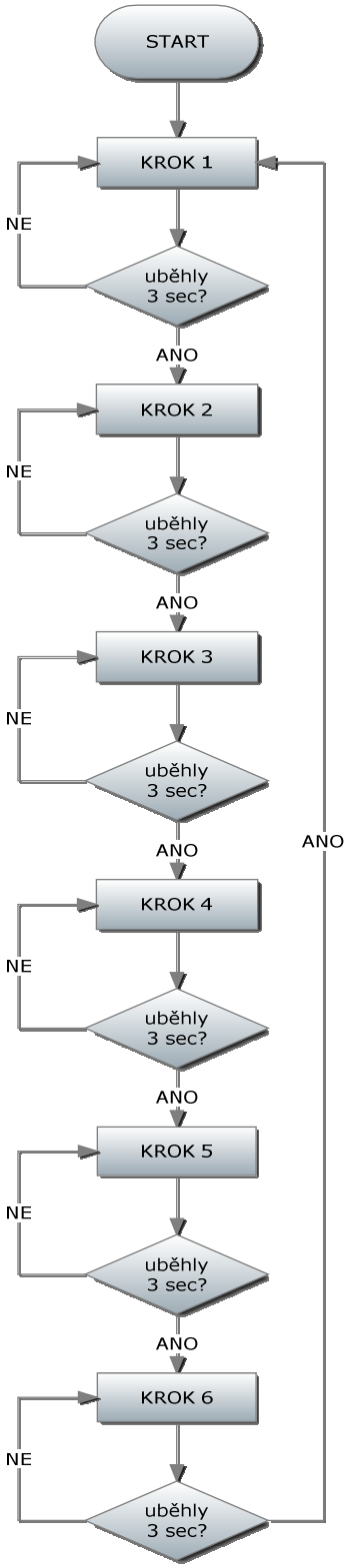
Obr.26: Header POU_01_01denního režimu křižovatky

Ve vývojovém diagramu je vidět pořadí KROKŮ, jak jdou za sebou. V *Tab.6* je vidět, které diody se v nich aktivují.

	KROK1	KROK2	KROK3	KROK4	KROK5	KROK6
Červená hlavní	0	0	1	1	1	1
Oranžová hlavní	0	1	0	0	0	1
Zelená hlavní	1	0	0	0	0	0
Červená vedlejší	1	1	1	0	0	1
Oranžová vedlejší	0	0	1	0	1	0
Zelená vedlejší	0	0	0	1	0	0
Červená přechod	1	1	0	1	1	1
Zelená přechod	0	0	1	0	0	0

Tab.6: Simulace křižovatky ve dne

Nyní si ukážeme průběh vývojového diagramu. START je zde chápán jako speciální paměť M8002, která vyšle impuls při prvním běhu programu. Podmínky určují kdy se přejde na další KROK.



Obr.27: Diagram hlavního programu křižovatky denní režim

6.1.3 Zadání 2. Úlohy

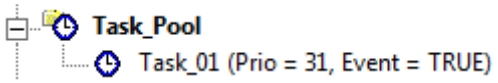
Ve druhé úloze jsem naprogramoval chování křižovatky v noci. Defaultně je nastaveno blikání oranžových diod na hlavním a vedlejším semaforu (0,5sec svítí a na 0,5sec zhasínají) a na přechodu svítí červená. Po stisknutí tlačítka na přechodu se na pět vteřin na hlavním semaforu objeví červená a na vedlejším zelená. Dále se po stisknutí tlačítka se zpožděním dvou vteřin rozsvítí na přechodu zelená na dobu tří vteřin. Po uplynutí této doby se vrátí hlavní a vedlejší semafor zpět na blikání a přechod na červenou.

6.1.4 Řešení 2. Úlohy

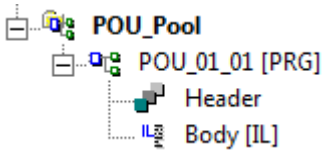
Na tuto úlohu jsme nepotřebovali použít podprogramy, proto je v Task_Pool pouze jedna úloha a v POU_Pool jen jeden program. Header je v tomto případě použit z jediného důvodu. V Global Vars se nedají editovat proměnné typu TON, TIME, atd..

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment
0	VAR_GLOBAL	CE_HL	Y0	%QX0	BOOL		FALSE	Červená hlavní
1	VAR_GLOBAL	OR_HL	Y1	%QX1	BOOL		FALSE	Oranžová hlavní
2	VAR_GLOBAL	OR_VE	Y4	%QX4	BOOL		FALSE	Oranžová vedlejší
3	VAR_GLOBAL	ZE_VE	Y5	%QX5	BOOL		FALSE	Zelená vedlejší
4	VAR_GLOBAL	CE_PR	Y6	%QX6	BOOL		FALSE	Červená přechod
5	VAR_GLOBAL	ZE_PR	Y7	%QX7	BOOL		FALSE	Zelená přechod
6	VAR_GLOBAL	TL	X10	%IX8	BOOL		FALSE	Tlačítko pro chodce

Obr.28: Global Vars nočního režimu křižovatky



Obr.29: Task Pool nočního režimu křižovatky

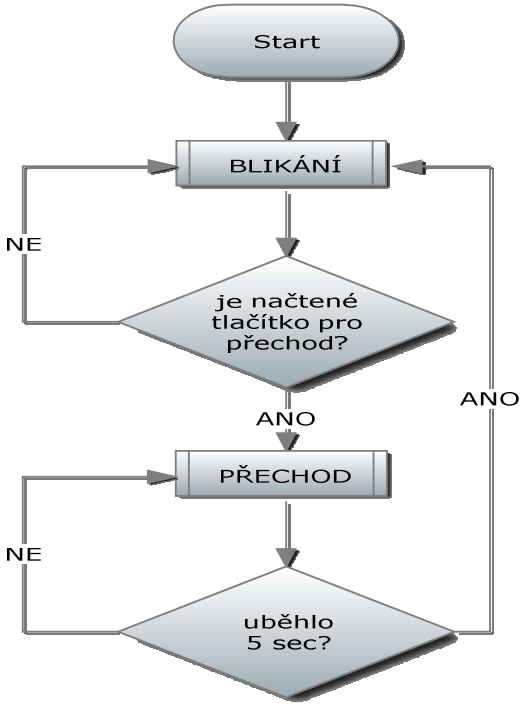


Obr.30: POU Pool nočního režimu křižovatky

	Class	Identifier	Type		Initial	Comment
0	VAR	TimeVal	TIME		T#0s	Ukládání uběhlého času počkej_blikání
1	VAR	TimeVal1	TIME		T#0s	Ukládání uběhlého času blikání
2	VAR	TimeVal2	TIME		T#0s	Ukládání uběhlého času přechod
3	VAR	TimeVal3	TIME		T#0s	Ukládání uběhlého času počkej_přechod_zelená
4	VAR	TimeVal4	TIME		T#0s	Ukládání uběhlého času přechod_zelená
5	VAR	POCKEJ_BLIKANI	TON			Čas mezi blikáním
6	VAR	POCKEJ_PRECHOD_ZELENA	TON			Zpožděně zapni zelenou po akci přechod
7	VAR	BLIKANI	TP			podrž blikání
8	VAR	PRECHOD	TP			podrž akci přechod
9	VAR	PRECHOD_ZELENA	TP			podrž zelenou na přechodu

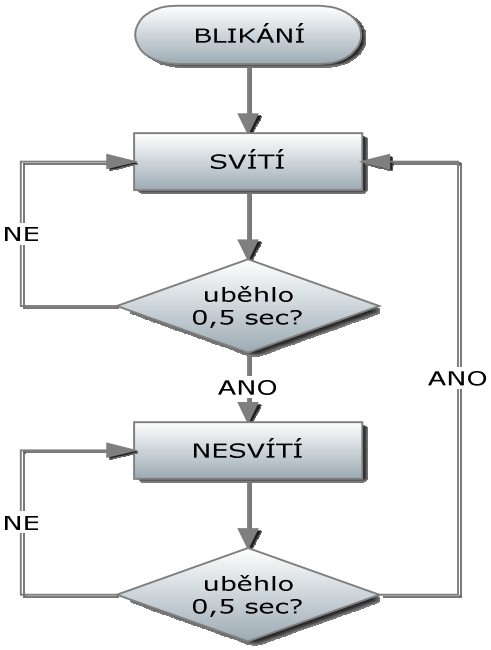
Obr.31: Header POU_01_01 nočního režimu křižovatky

Vývojový diagram je v tomto případě kvůli přehlednosti rozdělen na tři části. První část řeší, zda bylo stisknuto tlačítko na přechodu pro chodce. Pokud není, je stav klasický jak ho v noci známe a jak je popsán v textu zadání. Zde ho máme uveden jako BLIKÁNÍ. Pokud tlačítko stlačíme, proběhne situace uvedená v diagramu PŘECHOD. Po 5 vteřinách se vrátí do předchozího stavu. START opět značí speciální paměť M8002.



Obr.32:Diagram hlavního programu křižovatky nočního režimu

Význam SVÍTÍ/NESVÍTÍ je zobrazen v *Tab.7.* Podmínky určují frekvenci blikání.



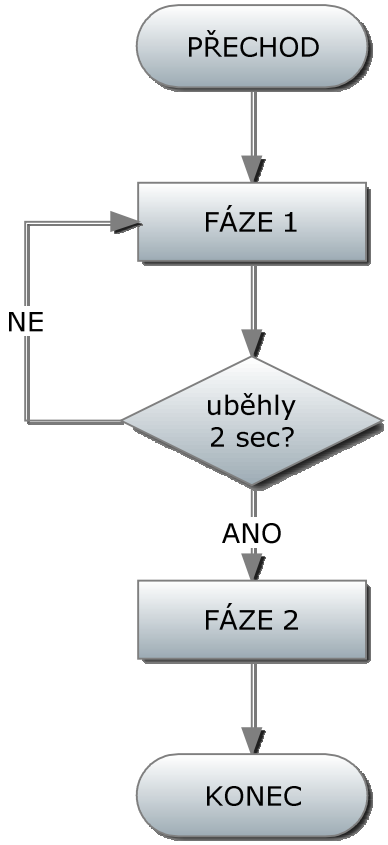
Obr.33:Diagram podprogramu blikání křižovatky nočního režimu

Při porovnání předchozího diagramu s následující tabulkou je vidět, že BLIKÁNÍ se týká pouze hlavního a vedlejšího semaforu. Pro semafor na přechodu je jako výchozí dioda nastavená červená. Pouze po stisknutí tlačítka na přechodu se změní na zelenou. Kdy a jak je ukázáno v diagramu a tabulce.

	NESVÍTÍ	SVÍTÍ	NESVÍTÍ	SVÍTÍ	…
oranžová hl	0	1	0	1	…
oranžová vedl	0	1	0	1	…
červená př	1	1	1	1	1

Tab.7: Blikání v nočním režimu křižovatky

Přechod se rozdělil na dvě fáze. FÁZE 1 znázorňuje chování prvních dvou sekund po stisknutí tlačítka na přechodu. FÁZE 2 ukazuje chování diod po další tři vteřiny, než se vše vrátí na blikající stav.



Obr.34:Diagram přechodu křižovatky v nočním režimu

FÁZE 1 značí bezpečné zastavení na možnost přechodu přes přechod ve FÁZI 2.

	FÁZE 1	FÁZE 2
červená hl	1	1
zelená vedl	1	1
červená př	1	0
zelená př	0	1

Tab.8: Fáze přechodu na křižovatce v nočním režimu

6.1.5 Zadání 3. Úlohy

Třetí úloha je vlastně kombinací předchozích dvou úloh. Defaultně se chová, jako křižovatka ve dne, ale po stisknutí tlačítka se přenese na 30 vteřin do nočního režimu. Po uplynutí 30 vteřin se vrací zpět na denní režim.

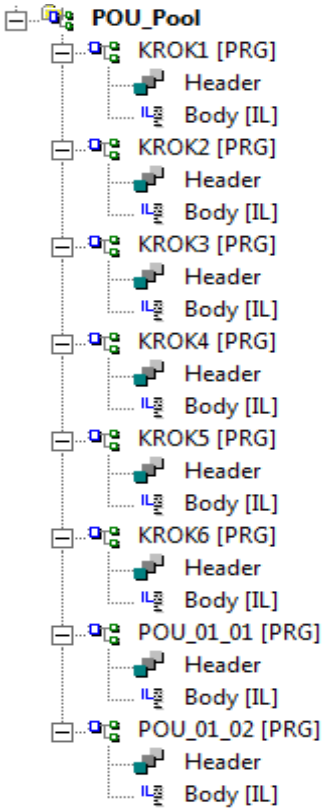
6.1.6 Řešení 3. Úlohy

Jak můžeme vidět na *Obr.36 a Obr.37*, použili jsme v projektu jeden hlavní program a sedm podprogramů. Hlavní program neřeší pouze posouvání KROKŮ jako je tomu u denního režimu, ale navíc má za úkol sledovat, zda se neaktivovalo tlačítko pro noční režim. V praxi je to řešeno pomocí reálného času. Bohužel pro simulaci je nemožné reálný čas použít, protože by se pro ověřování funkčnosti musel čas stále upravovat. V nočním režimu je opět možné možné použít tlačítko na přechodu pro chodce. V denním režimu je přechod řešen automaticky v KROCÍCH.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment	Remark
0	VAR_GLOBAL	CE_HL	Y0	%QX0	BOOL		FALSE	Červená hlavní	
1	VAR_GLOBAL	OR_HL	Y1	%QX1	BOOL		FALSE	Oranžová hlavní	
2	VAR_GLOBAL	ZE_HL	Y2	%QX2	BOOL		FALSE	Zelená hlavní	
3	VAR_GLOBAL	CE_VE	Y3	%QX3	BOOL		FALSE	Červená vedlejší	
4	VAR_GLOBAL	OR_VE	Y4	%QX4	BOOL		FALSE	Oranžová vedlejší	
5	VAR_GLOBAL	ZE_VE	Y5	%QX5	BOOL		FALSE	Zelená vedlejší	
6	VAR_GLOBAL	CE_PR	Y6	%QX6	BOOL		FALSE	Červená přechod	
7	VAR_GLOBAL	ZE_PR	Y7	%QX7	BOOL		FALSE	Zelená přechod	
8	VAR_GLOBAL	TL	X10	%IX8	BOOL		FALSE	Tlačítko přechod	
9	VAR_GLOBAL	NOC	X11	%IX9	BOOL		FALSE	Noční chod	

Obr.35: Global Vars celodenního režimu křižovatky

Obr.36: Task Pool celodenního režimu křižovatky



Obr.37: POU Pool celodenního režimu křižovatky

Na *Obr. 38* a *Obr.39* jsou vidět použité časovače zpožděného zapnutí a vypnutí a časovač pro podržení signálu TP.

	Class	Identifier	Type		Initial	Comment
0	VAR	TimeVal1	TIME	...	T#0s	Ukládání uběhlého času kroku 2
1	VAR	TimeVal2	TIME	...	T#0s	Ukládání uběhlého času kroku 3
2	VAR	TimeVal3	TIME	...	T#0s	Ukládání uběhlého času kroku 4
3	VAR	TimeVal4	TIME	...	T#0s	Ukládání uběhlého času kroku 5
4	VAR	TimeVal5	TIME	...	T#0s	Ukládání uběhlého času kroku 6
5	VAR	TimeVal6	TIME	...	T#0s	Ukládání uběhlého času kroku 1
6	VAR	TimeVal11	TIME	...	T#0s	Ukládání uběhlého času zapni_den
7	VAR	TimeVal10	TIME	...	T#0s	Ukládání uběhlého času vypni_noc
8	VAR	VYPNI_NOC	TOF	...		zpozdene vypnuti nocniho rezimu
9	VAR	KROK2	TON	...		Zpožděné zapnutí druhého kroku
10	VAR	KROK3	TON	...		Zpožděné zapnutí třetího kroku
11	VAR	KROK4	TON	...		Zpožděné zapnutí čtvrtého kroku
12	VAR	KROK5	TON	...		Zpožděné zapnutí pátého kroku
13	VAR	KROK6	TON	...		Zpožděné zapnutí šestého kroku
14	VAR	KROK1	TON	...		Zpožděné zapnutí prvního kroku
15	VAR	ZAPNI_DEN	TON	...		zpozdene zapnuti denniho rezimu po noci
16	VAR	TP	TP	...		

Obr.38: Header denní části (POU_01_01) celodenního režimu křižovatky

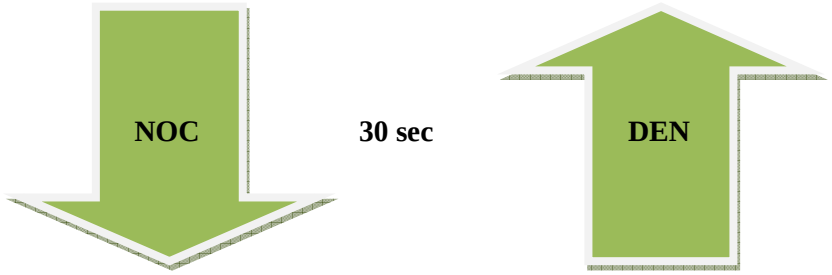
	Class	Identifier	Type		Initial	Comment
0	VAR	POCKEJ_BLIKANI	TON	...		Čas mezi blikáním
1	VAR	POCKEJ_PRECHOD_ZELENA	TON	...		Zpožděně zapni zelenou po akci přechod
2	VAR	BLIKANI	TP	...		podrž blikání
3	VAR	PRECHOD	TP	...		podrž akci přechod
4	VAR	PRECHOD_ZELENA	TP	...		podrž zelenou na přechodu
5	VAR	TimeVal20	TIME	...	T#0s	Ukládání uběhlého času počkej_blikání
6	VAR	TimeVal21	TIME	...	T#0s	Ukládání uběhlého času blikání
7	VAR	TimeVal22	TIME	...	T#0s	Ukládání uběhlého času přechod
8	VAR	TimeVal23	TIME	...	T#0s	Ukládání uběhlého času
9	VAR	TimeVal24	TIME	...	T#0s	Ukládání uběhlého času přechod_zelená

Obr.39: Header noční části (POU_01_02) celodenního režimu křižovatky

V *Tab.9* je vidět časový průběh střídání diod křižovatky v denním režimu, který se po aktivaci tlačítka přenese na 30 sekund do nočního režimu, který zobrazuje *Tab.10*. Pokud je křižovatka v nočním režimu, může se použít tlačítko na přechodu pro chodce. Průběh přechodového režimu zobrazuje *Tab.11*. Tlačítko na přechodu v denním režimu nefunguje.

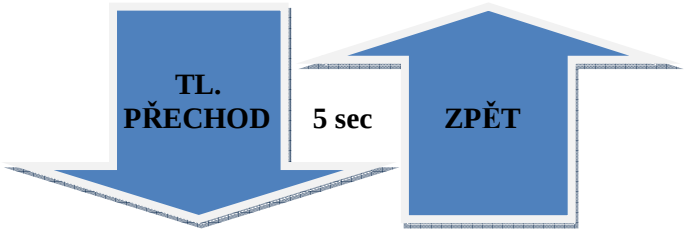
	KROK1	KROK2	KORK3	KROK4	KROK5	KROK6
zelená hl	1	0	0	0	0	0
oranžová hl	0	1	0	0	0	1
červená hl	0	0	1	1	1	1
zelená vedl	0	0	0	1	0	0
oranžová vedl	0	0	1	0	1	0
červená vedl	1	1	1	0	0	1
zelná př	0	0	1	0	0	0
červená př	1	1	0	1	1	1

Tab.9:



	NESVÍTÍ	SVÍTÍ	NESVÍTÍ	SVÍTÍ	...
oranžová hl	0	1	0	1	...
oranžová vedl	0	1	0	1	...
červená př	1	1	1	1	1

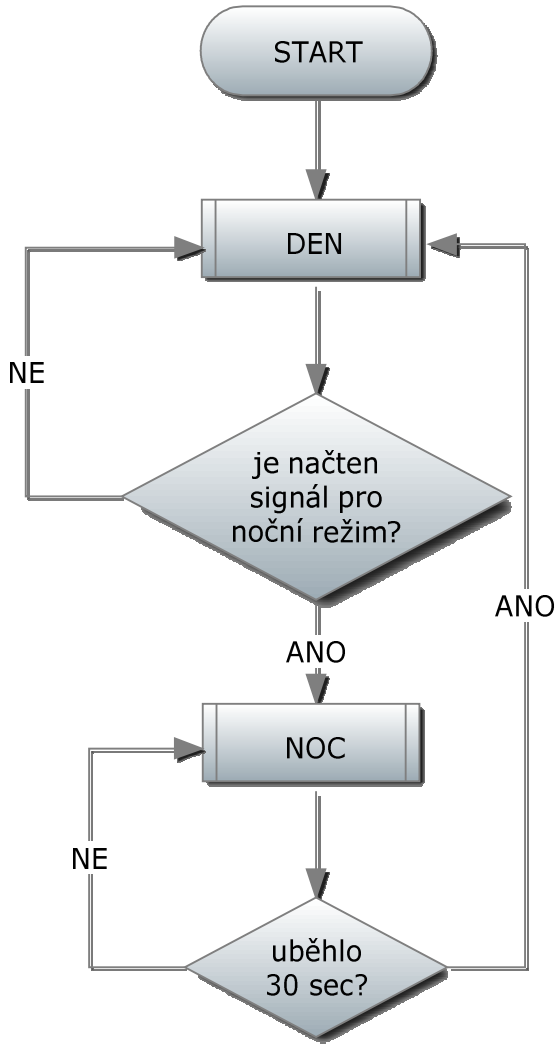
Tab.10:



	FÁZE 1	FÁZE 2
červená hl	1	1
zelená vedl	1	1
červená př	1	0
zelená př	0	1

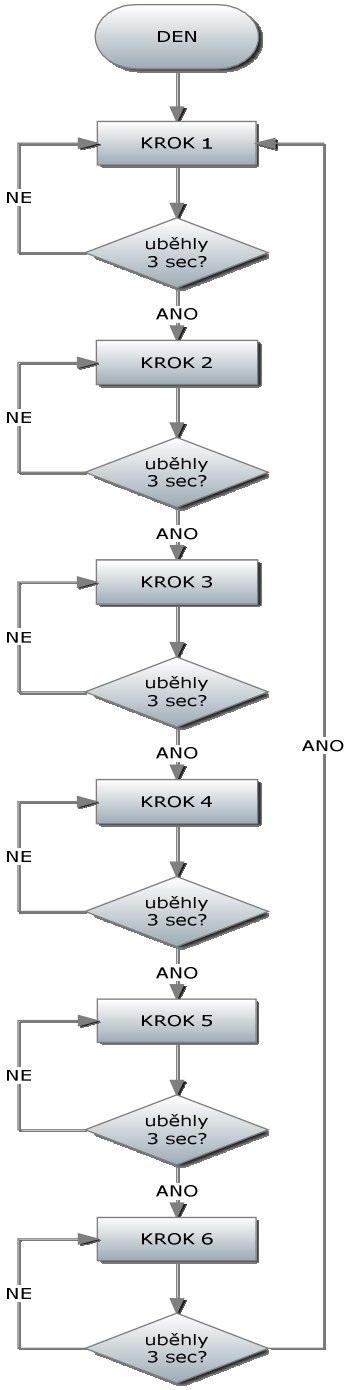
Tab.11:

Jako výchozí stav po zapnutí je denní režim, který neustále koluje. Během běhu denního režimu neustále kontroluje, zda není aktivováno tlačítko k přechodu do nočního režimu.



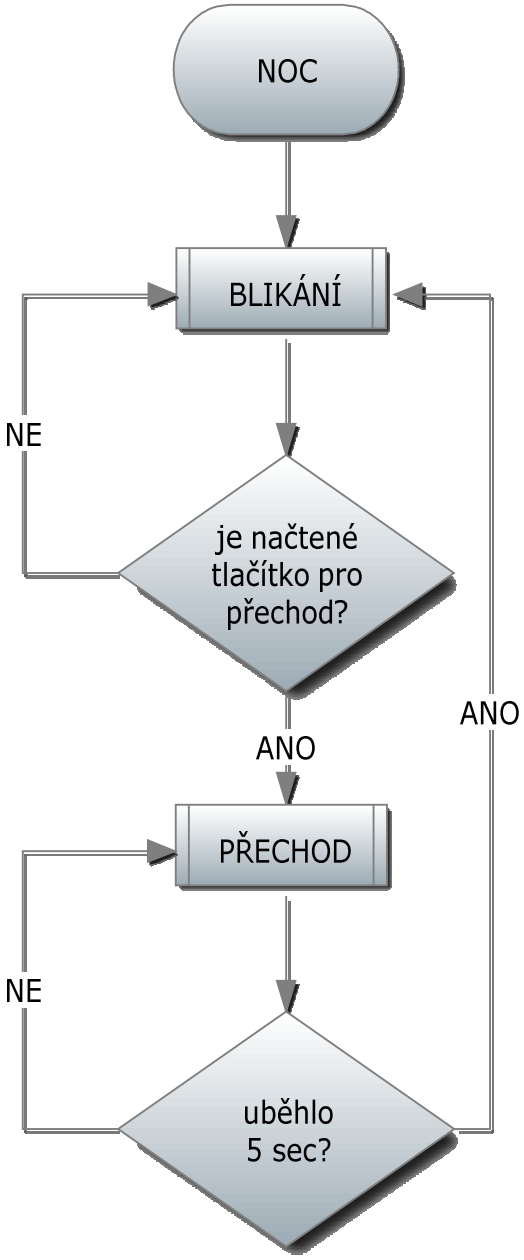
Obr.40: Diagram hlavního programu celodenního režimu křižovatky

Výchozí stav střídání kroků je stejný jako u denního režimu. Jediný rozdíl je, že u diagramu není na začátku START, ale DEN. Rozdíl mezi nimi je ten, že START značí hlavní program, kdežto v našem případě je denní režim řešen jako podprogram. Protože je denní režim označen jako DEN v hlavním programu, musí tak stejným označením začínat i podprogram.



Obr.41: Diagram podprogramu pro den v celodenním režimu křižovatky

Stejně jako je denní režim označen DEN, podprogram nočního režimu jsme označili NOC. Výchozím stavem je blikání, přičemž neustále kontroluje, zda nebylo stisknuto tlačítko na přechodu. Pokud ano, podprogram PŘECHOD je po 5 vteřinách přepnut zpět na výchozí stav. PŘECHODU a BLIKÁNÍ náleží *Tab. 11* a *Tab.10.*



Obr.42:Diagram podprogramu noc v celodenním režimu křižovatky

Diagramy BLIKÁNÍ a PŘECHOD jsou stejné jako BLIKÁNÍ a PŘECHOD v kapitole 6.1.4.

6.2 Pračka

V nadcházející kapitole si ukážeme úlohy týkající se simulace chování automatické pračky. Stejně jako u křižovatky je zde vidět zpracovávání programů a podprogramů v GX IEC Developeru. Vidět jsou také použité proměnné v Global Vars a Header.

6.2.1 Zadání první úlohy

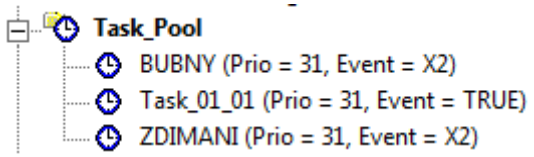
První úloha má znázornit chování pračky v automatickém stavu. Napustíme vodu do 100%. Při 50% se začne ohřívat voda až na 90°C. Po ohřevu vody se začne střídavě otáčet bubnem doprava a doleva po 5 sekundách, přičemž je při změně směru sekundová pauza. Po 30 sekundách střídavého otáčení se buben začne otáčet pouze na jednu a to levou stranu, přičemž se aktivuje zrychlení otáčení bubnu. Po dalších 3 sekundách se zapne vypouštění vody. Následuje 27 vteřin, po kterých se pračka vypne.

6.2.2 Řešení první úlohy

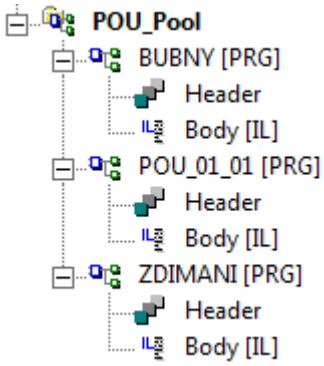
Hlavní program řeší napouštění, ohřev vody a dobu ždímání. V programu Ždímání se kontroluje, zda se pere již 30 sekund, aby se mohlo aktivovat vypouštění, zrychlené otáčení bubnem, atd.. V programu BUBNY se řeší střídavé otáčení bubnem, což simuluje proces praní.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment	Remark
0	VAR_GLOBAL	VODA50	X0	%IX0	BOOL		FALSE	Napuštění do 50%	
1	VAR_GLOBAL	VODA100	X1	%IX1	BOOL		FALSE	Napuštění 100%	
2	VAR_GLOBAL	TEPLOTA90	X2	%IX2	BOOL		FALSE	Ohřev vody na 90°	
3	VAR_GLOBAL	TEPLOTA60	X3	%IX3	BOOL		FALSE	Ohřev vody na 60°	
4	VAR_GLOBAL	TEPLOTA40	X4	%IX4	BOOL		FALSE	Ohřev vody na 40°	
5	VAR_GLOBAL	TEPLOTA30	X5	%IX5	BOOL		FALSE	Ohřev vody na 30°	
6	VAR_GLOBAL	NAPOUSTENI	Y5	%QX5	BOOL		FALSE	Aktivace napouštění vody	
7	VAR_GLOBAL	TOPENI	Y4	%QX4	BOOL		FALSE	Zapnutí ohřevu vody	
8	VAR_GLOBAL	BUBEN_VPRAVO	Y0	%QX0	BOOL		FALSE	Otáčení bubnu vpravo	
9	VAR_GLOBAL	BUBEN_VLEVO	Y1	%QX1	BOOL		FALSE	Otáčení bubnu vlevo	
10	VAR_GLOBAL	ZRYCHLENI	Y2	%QX2	BOOL		FALSE	Zrychlení otáčení bubnu	
11	VAR_GLOBAL	VYPOUSTENI	Y6	%QX6	BOOL		FALSE	Vypouštění vody	

Obr.43: Global Vars automatického režimu pračky



Obr.44: Task Pool automatického režimu pračky



Obr.45: POU Pool automatického režimu pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	DOBA_ZDIMANI	TON	...		Doba ždímání
1	VAR	TimeVal2	TIME	...	T#0s	Ukládání uběhlého času

Obr.46: Header hlavního programu (POU_01_01) automatického režimu pračky

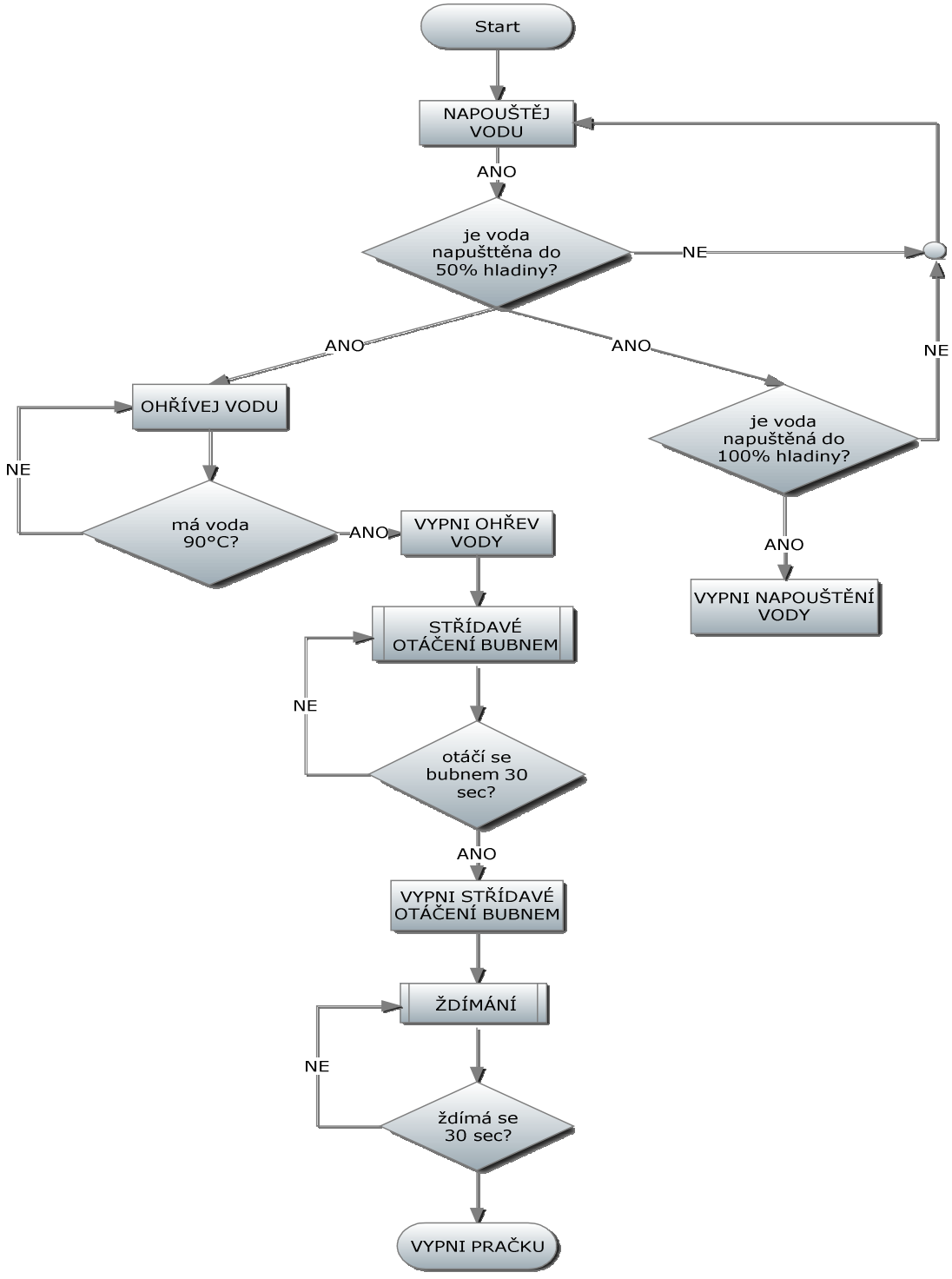
	Class	Identifier	Type		Initial	Comment
0	VAR	TimeVal	TIME	...	T#0s	Ukládání uběhlého času
1	VAR	TimeVal1	TIME	...	T#0s	Ukládání uběhlého času
2	VAR	TimeVal01	TIME	...	T#0s	Ukládání uběhlého času
3	VAR	TimeVal11	TIME	...	T#0s	Ukládání uběhlého času
4	VAR	TimeVal21	TIME	...	T#0s	Ukládání uběhlého času
5	VAR	POCKEJ_VPRAVO	TON	...		
6	VAR	POCKEJ_VLEVO	TON	...		
7	VAR	OTACEJ_VPRAVO_PRVNI_CYKLUS	TP	...		
8	VAR	OTACEJ_VPRAVO	TP	...		
9	VAR	OTACEJ_VLEVO	TP	...		

Obr.47: Header bubny automatického režimu pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	PRANI	TON	...		Doba střídání otáčení bubnu
1	VAR	POCKEJ_VYPOUSTENI	TON	...		Zpožděné zapnutí vypouštění vody
2	VAR	TimeVal4	TIME	...	T#0s	Ukládání uběhlého času počkej_vypouštění
3	VAR	TimeVal3	TIME	...	T#0s	Ukládání uběhlého času praní

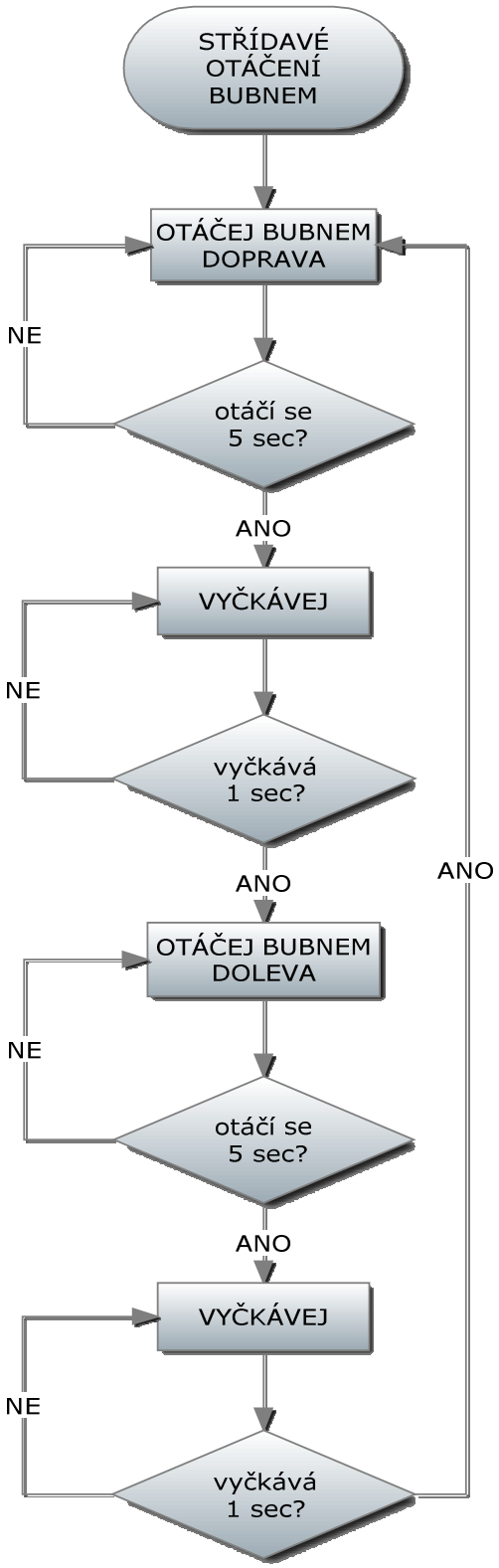
Obr.48: Header ždímání automatického režimu pračky

V následujícím diagramu vidíme řešení napouštění a ohřevu vody. Poté se zapne střídavé otáčení bubnem na nastavený čas a po něm se začne ždímat.



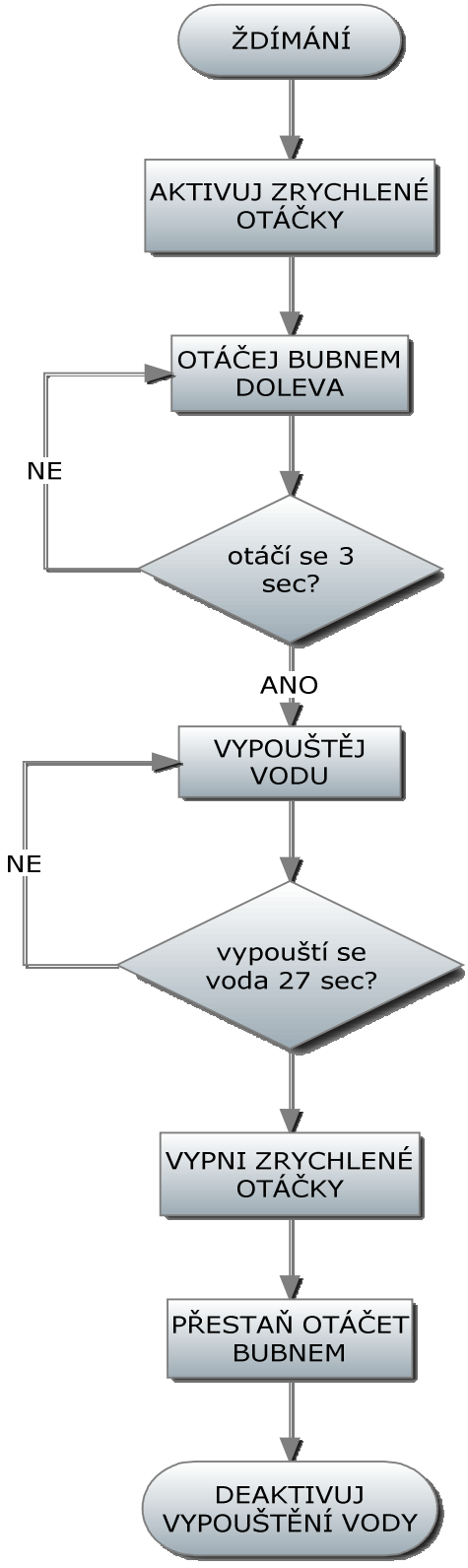
Obr.49: Diagram hlavního programu automatické pračky

V předchozím digramu jsme viděli, jak jsou označeny příslušné podprogramy. Nyní si ukážeme podprogram pro střídavé otáčení bubnů. Vždy se točí 5 sekund na každou stranu, přičemž je při změně smyslu otáčení sekundová pauza. Takto celý cyklus rotuje po celou dobu platnosti tohoto podprogramu.



Obr.50: Diagram střídavého otáčení bubnem

U podprogramu ŽDÍMÁNÍ vidíme, že se po spuštění tohoto cyklu zapnou zrychlené otáčky a smysl otáčení bubnu je pouze doleva. Se zpožděním se aktivuje vypouštění vody.



Obr.51:Diagram ždímání automatické pračky

6.2.3 Zadání 2. Úlohy

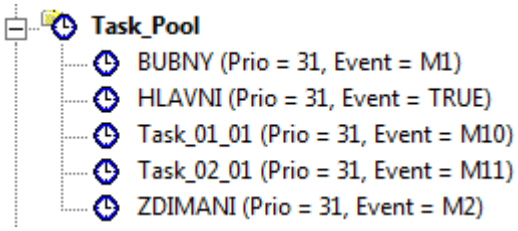
Naprogramujte chování pračky tak, aby si uživatel mohl dvěma tlačítky určit, zda chce vybrat mezi celým cyklem praní, nebo mácháním. Pokud si vybereme praní, chová se pračka stejně, jako v první úloze. Vybereme-li si máchání, tak se po napuštění vody do 100% a ohřevu na 90°C aktivují pouze bubny, které se střídavě otáčejí 30 sekund. Opět je mezi otáčeními sekundová pauza. Po těchto 30 sekundách se nezapne ždímání, ale začne se voda vypouštět po dobu 30 sekund. Po uplynutí 30 sekund se pračka vypne.

6.2.4 Řešení 2. Úlohy

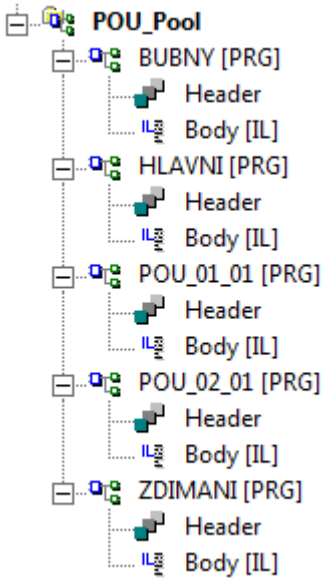
Program HLAVNÍ z *Obr. 53* a *54* obsahuje pouze dotaz na výběr mezi cyklem praní a mácháním. Podle výběru si zvolí buď POU_01_01 nebo POU_02_01, které si pak volají BUBNY a ŽDÍMÁNÍ dle potřeby. BUBNY představují střídavé otáčení bubnem.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type		Initial	Comment	Remark
0	VAR_GLOBAL	VODA50	X0	%IX0	BOOL		FALSE	Napuštění do 50%	
1	VAR_GLOBAL	VODA100	X1	%IX1	BOOL		FALSE	Napuštění 100%	
2	VAR_GLOBAL	TEPLOTA90	X2	%IX2	BOOL		FALSE	Ohřev vody na 90°	
3	VAR_GLOBAL	TEPLOTA60	X3	%IX3	BOOL		FALSE	Ohřev vody na 60°	
4	VAR_GLOBAL	TEPLOTA40	X4	%IX4	BOOL		FALSE	Ohřev vody na 40°	
5	VAR_GLOBAL	TEPLOTA30	X5	%IX5	BOOL		FALSE	Ohřev vody na 30°	
6	VAR_GLOBAL	NAPOUSTENI	Y5	%QX5	BOOL		FALSE	Aktivace napouštění	
7	VAR_GLOBAL	TOPENI	Y4	%QX4	BOOL		FALSE	Zapnutí ohřevu vody	
8	VAR_GLOBAL	BUBEN_VPRAVO	Y0	%QX0	BOOL		FALSE	Otáčení bubnu vpravo	
9	VAR_GLOBAL	BUBEN_VLEVO	Y1	%QX1	BOOL		FALSE	Otáčení bubnu vlevo	
10	VAR_GLOBAL	ZRYCHLENI	Y2	%QX2	BOOL		FALSE	Zrychlení otáčení bubnu	
11	VAR_GLOBAL	VYPOUSTENI	Y6	%QX6	BOOL		FALSE	Vypouštění vody	
12	VAR_GLOBAL	PRANI	X10	%IX8	BOOL		FALSE	Vypouštění vody	
13	VAR_GLOBAL	MACHANI	X11	%IX9	BOOL		FALSE	Vypouštění vody	

Obr.52: Global Vars výběrového režimu pračky



Obr.53: Task Pool výběrového režimu pračky



Obr.54: POU Pool výběrového režimu pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	DOBA_ZDIMANI	TON	...		Doba ždímání
1	VAR	TimeVal11	TIME	...	T#0s	Ukládání uběhlého času doba_ždímání

Obr.55: Header Praní (POU_01_01) výběrového režimu pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	CELY_CYKLUS	TON	...		Doba otáčení bubnu a vypouštění
1	VAR	STRIDANI_BUBNU	TON	...		Doba střídání otáčení bubnu
2	VAR	TimeVal21	TIME	...	T#0s	Ukládání uběhlého času celý_cyklus
3	VAR	TimeVal22	TIME	...	T#0s	Ukládání uběhlého času střídání_bubnu

Obr.56: Header Máchání (POU_02_01) výběrového režimu pračky

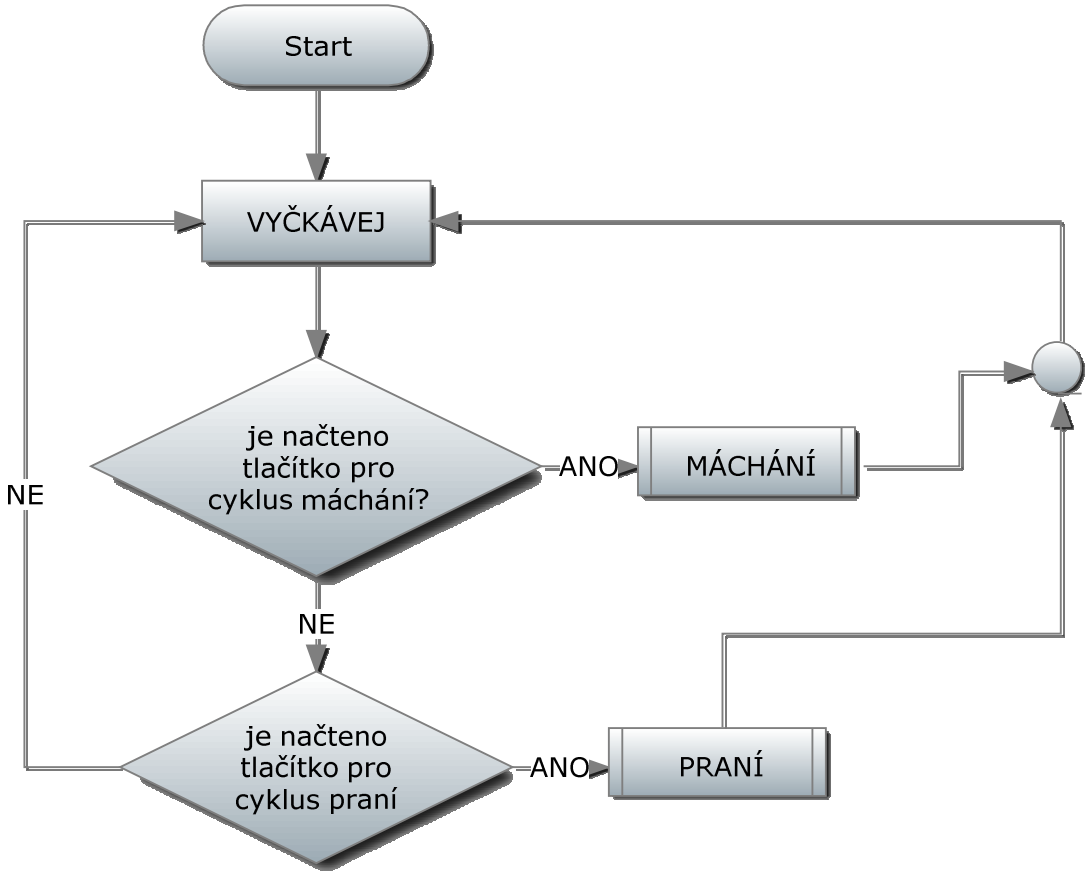
	Class	Identifier	Type		Initial	Comment
0	VAR	TimeValB	TIME	...	T#0s	Ukládání uběhlého času
1	VAR	TimeValB1	TIME	...	T#0s	Ukládání uběhlého času otáčení vpravo
2	VAR	TimeValB2	TIME	...	T#0s	Ukládání uběhlého času otáčení_vlevo
3	VAR	TimeValNB	TIME	...	T#0s	Ukládání uběhlého času počkej_vpravo
4	VAR	TimeValNB1	TIME	...	T#0s	Ukládání uběhlého času počkej_vlevo
5	VAR	POCKEJ_VPRAVO	TON	...		Počkej na otáčení bubnu vpravo
6	VAR	POCKEJ_VLEVO	TON	...		Počkej na otáčení bubnu vlevo
7	VAR	OTACENI_VPRAVO_PRVNI_CYKLUS	TP	...		Otáčení bubnu doprava při prvním cyklu
8	VAR	OTACENI_VPRAVO	TP	...		Otáčení bubnu doprava
9	VAR	OTACENI_VLEVO	TP	...		Otáčení bubnu doleva

Obr.57: Header bubny výběrového režimu pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	STRIDANI_BUBNU	TON	...		Doba střídání otáčení bubnu
1	VAR	POCKEJ_VYPOUSTENI	TON	...		Zpožděné zapnutí vypouštění
2	VAR	TimeValZ	TIME	...	T#0s	Ukládání uběhlého času střídání_bubnu
3	VAR	TimeValZ2	TIME	...	T#0s	Ukládání uběhlého času počkej_vypouštění

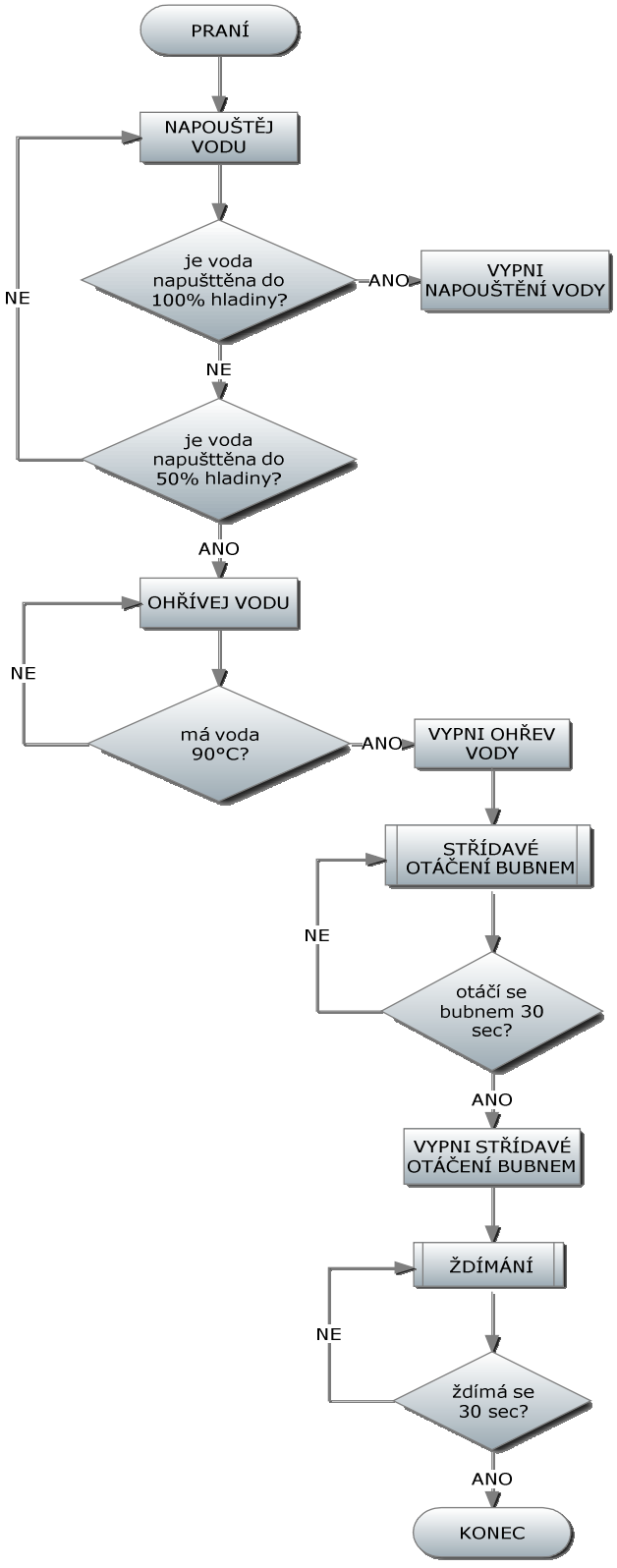
Obr.58: Header ždímání výběrového režimu pračky

Zde vidíme hlavní program s dotazem na výběr mezi cyklem PRANI a MÁCHÁNÍ. Pokud není vybrán ani jeden cyklus program vyčkává.



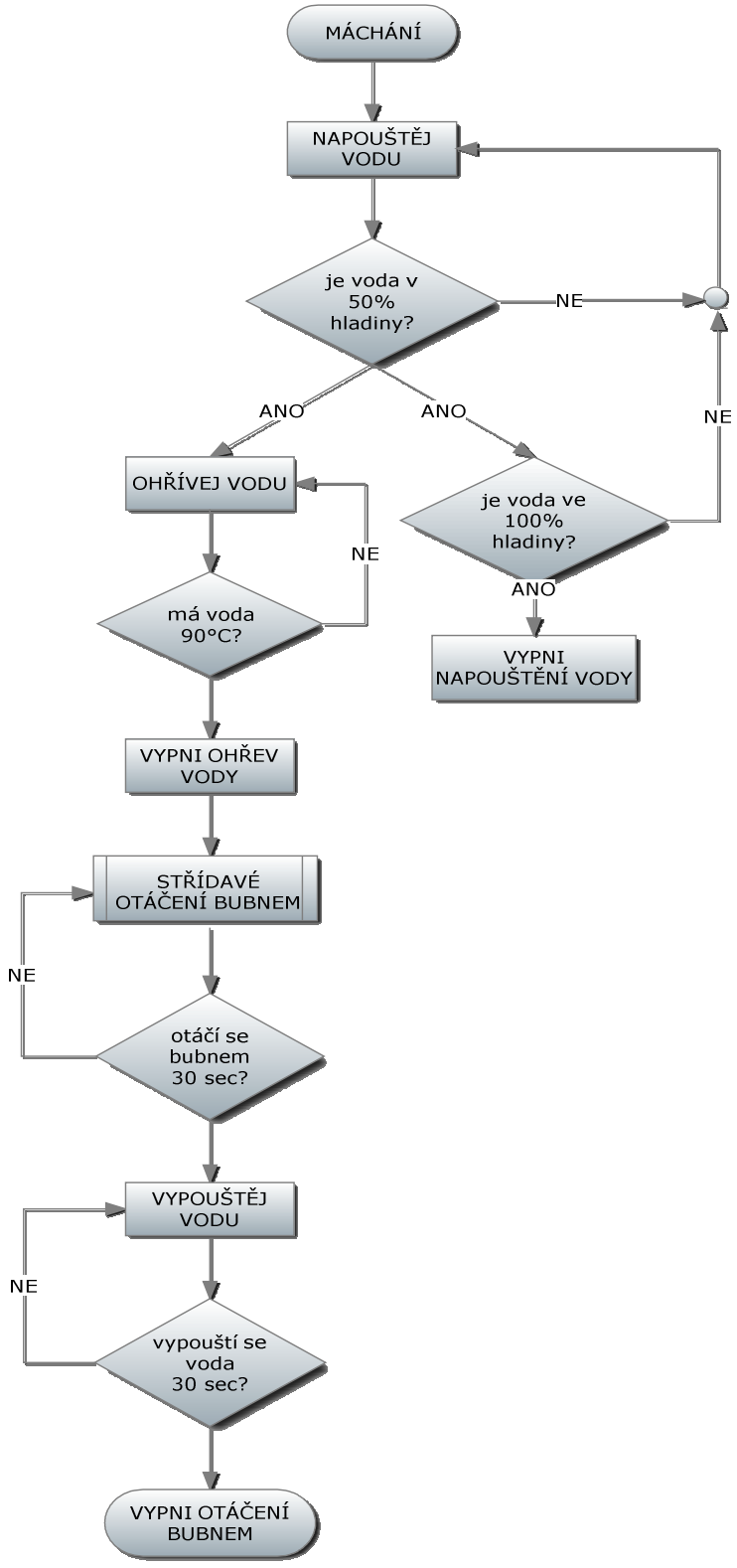
Obr.59:Diagram výběru mezi praním a mácháním

Takto vypadá program po výběru cyklu PRANÍ. Volá podprogramy mající za úkol střídavě otáčet bubnem, nebo ždímat. Protože jsou stejné jako u první úlohy, nebudu je uvádět.



Obr.60:Diagram praní ve výběrovém režimu

Při výběru MÁCHÁNÍ se po napuštění a ohřevu vody volá pouze STŘÍDAVÉ OTÁČENÍ BUBNEM, které je spuštěno po celý zbytek procesu. V polovině zvoleného času pro otáčení bubnem se začne vypouštět voda. Vypouštění trvá 30 sekund a poté se buben přestane otáčet.



Obr.61:Diagram máchání ve výběrovém režimu

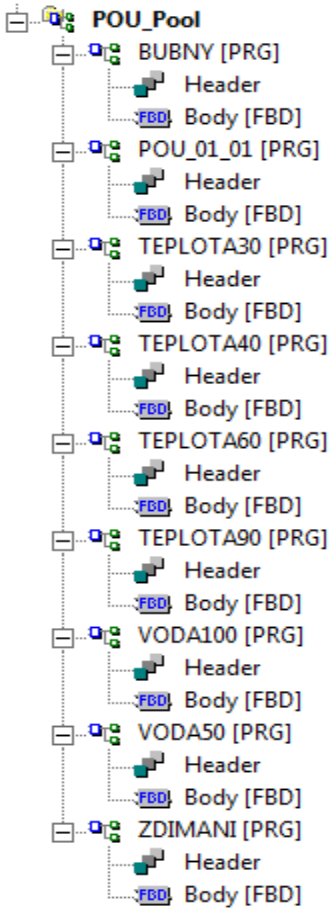
6.2.5 Zadání 3.úlohy

Tento program slouží ke zjištění příčiny poruchy pračky, kde si postupně volíme, zda chceme napouštět vodu a do jaké hladiny. Dále si zvolíme na kolik stupňů chceme ohřát vodu. Rotování bubnem, zrychlené otáčení a vypouštění vody jsou řešeny přes cykly, aby zákazník mohl vyprat/vyždímat i při poruše. Program je možné ovládat přes dotykový panel, viz kapitola 6.3.4.

6.2.6 Řešení 3. Úlohy

Záměrně jsem ve zdrojových kódech nepoužíval proměnné vstupů a výstupů, aby byla na první pohled vidět souvislost s programem v EasyBuilder v kapitole 6.3.2. Z toho důvodu chybí Global Vars.

Obr.62: Task Pool manuálního ovládání pračky



Obr.63: POU Pool manuálního ovládání pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	TimeValM8	TIME		T#0s	
1	VAR	TimeValM7	TIME		T#0s	
2	VAR	TONM7	TON			
3	VAR	TONM8	TON			

Obr.64: Header hlavního programu manuálního ovládání pračky

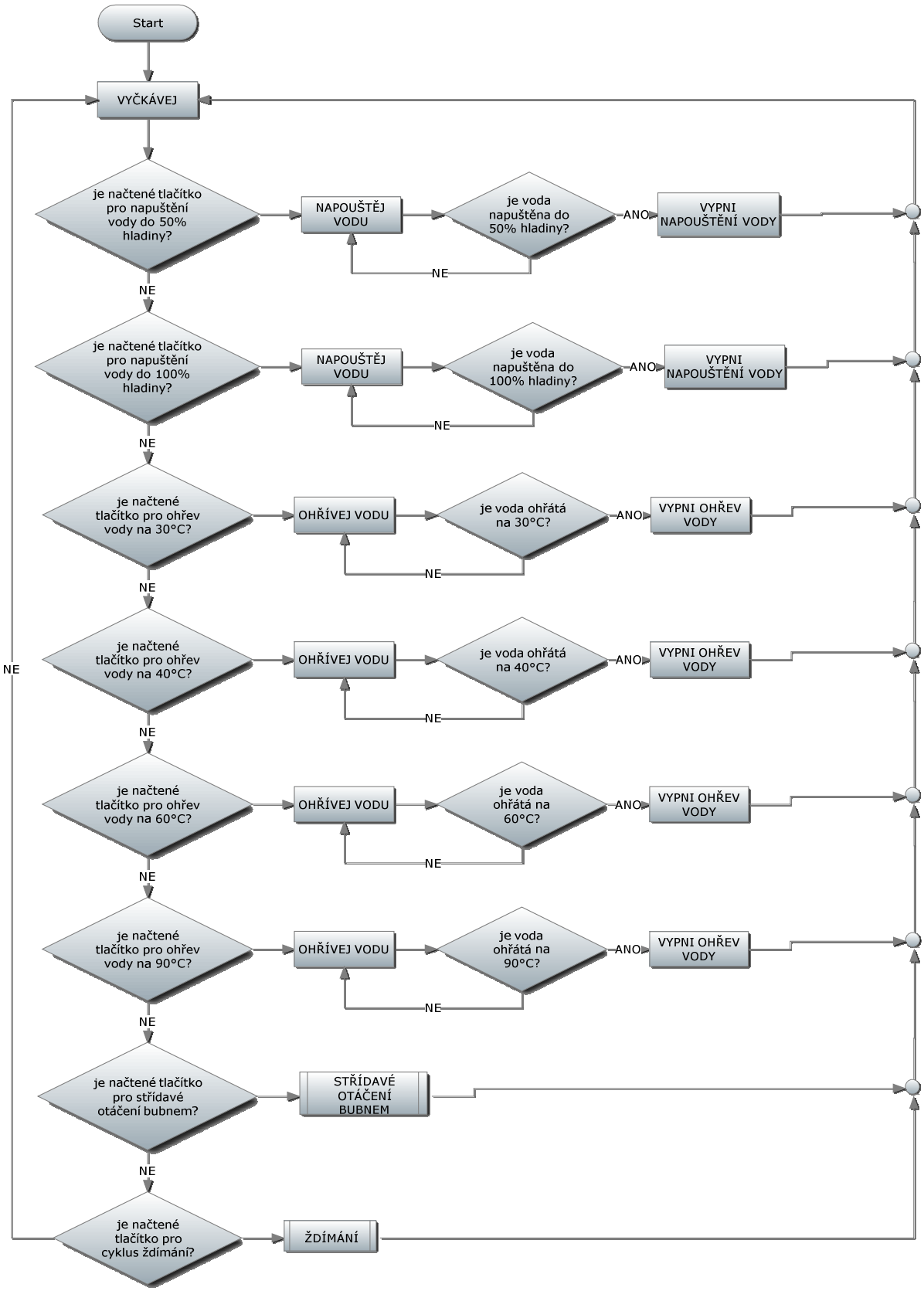
	Class	Identifier	Type		Initial	Comment
0	VAR	TP2	TP			
1	VAR	TP1	TP			
2	VAR	TP	TP			
3	VAR	TON1	TON			
4	VAR	TON	TON			
5	VAR	TOF	TOF			
6	VAR	TimeVal21	TIME		T#0s	
7	VAR	TimeVal1	TIME		T#0s	
8	VAR	TimeVal11	TIME		T#0s	
9	VAR	TimeVal	TIME		T#0s	
10	VAR	TimeVal01	TIME		T#0s	
11	VAR	TimeValF	TIME		T#0s	

Obr.65: Header otáčení bubnů manuálního ovládání pračky

	Class	Identifier	Type		Initial	Comment
0	VAR	TONZ	TON			
1	VAR	TimeValZ	TIME		T#0s	

Obr.66: Header ždímání manuálního ovládání pračky

Jak je vidět na *Obr.67*, program vyčkává, dokud nezvolíme námi vybranou akci. Po např. napuštění vody do 50% hladiny se vrátí zpět na vyčkávání. Tento program je výhodný v tom, že když máme např. poruchu na čidle ke zjištění 100% hladiny, program nehodí chybu a nezastaví se, ale můžeme vyprat s 50% vody. Což u automatického přednastaveného programu nelze.



Obr.67:Diagram manuálního režimu pračky

6.3 Rozšíření 3. úlohy pračky dotykovým displejem

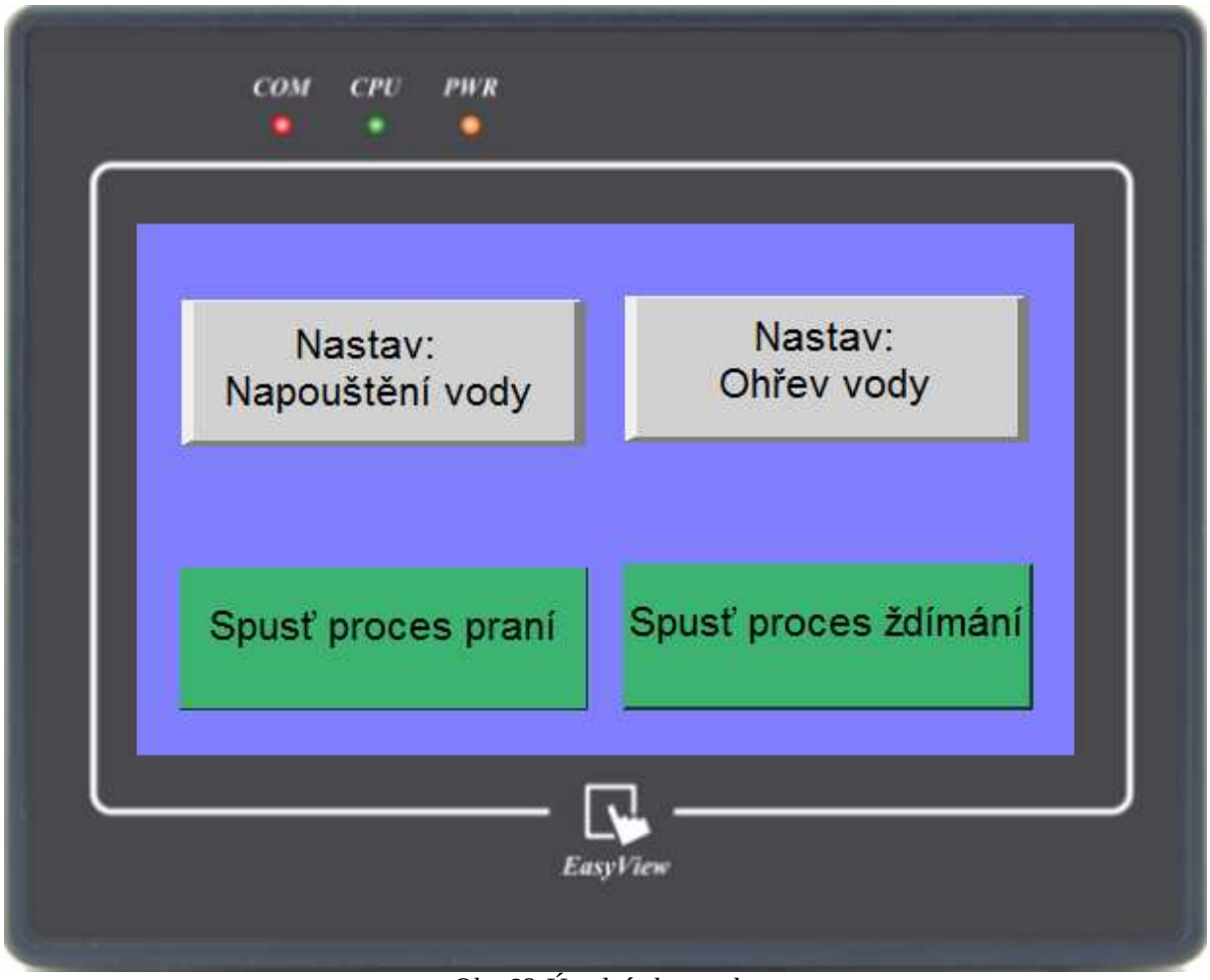
3. úlohu u automatické pračky jsem rozšířil o možnost ovládání pomocí dotykového panelu. Vlastnosti všech použitých tlačítek jsou popsány v sekci nastavení objektů.

6.3.1 Zadání 1. Úlohy

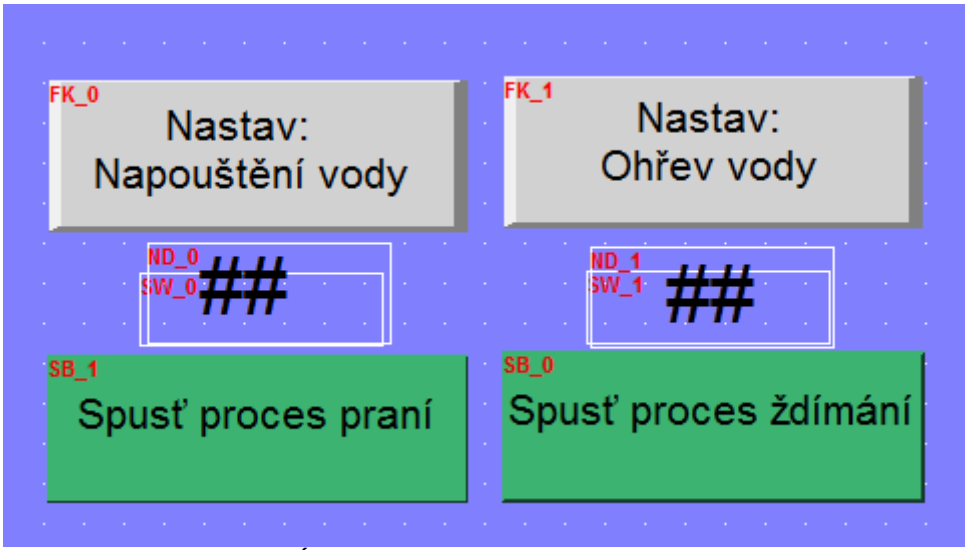
Naprogramujte dotykový panel tak, abychom mohli ovládat napouštění, teplotu a mohli si vybrat mezi zkouškou otáčení bubnem a ždímáním.

6.3.2 Řešení 1. úlohy

Na úvodní obrazovce si uživatel může vybrat, zda chce přepnout na obrazovku s výběrem napouštění vody, výběrem ohřevu vody, spuštěním praní, nebo spuštění procesu ždímání.



Obr.68:Úvodní obrazovka



Obr.69:Úvodní obrazovka programátorský režim

Nastavení objektů

FK_0: Jedná se o Function key object

Záložky:

General - nastavíme Change full-screen window a vybereme Window no.:11, což je přepnutí na obrazovku s nastavením napouštění vody.

Shape - Zatrhneme možnost Use shape a pomocí Shape Library vybereme požadovanou podobu tlačítka.

Label - zadáme možnost Use label a v okně Content napíšeme text, který se zobrazí na tlačítku.

FK_1: Opět se jedná o Funtion key object

Záložky:
General - stejné nastavení jako u FK_0, pouze Window no. vybereme 12.

Shape - Totožné s FK_0.

Label - Opět stejné jako FK_0, jen text se liší.

SB_1: Set Bit Object

Záložky:
General - Write adress (PLC Name: MITSUBISHI FX3u, Adress: X16- jak je napsáno výše, tato úloha je spjatá s úlohou automatické pračky č.3, tzn., že význam adresy X16 můžeme vyčíst z kódu)

- Attribute (Set style: Momentary - vyšle impuls na danou adresu)

Shape - Zaškrtneme Use shape, poté v Shape Library vybereme požadovaný tvar tlačítka a nakonec barvu.

Label- Vybereme Use label a do Content vložíme text.

SB_0: Set Bit Object

Záložky:
General - Stejné nastavení jako SB_1, jen změníme adresu na X17.

Shape - Stejné jako SB_1.

Label - V Content je změněn text.

ND_0: Numeric Display Object

Záložky:
General - Read adress: D1

Numeric Format - Data format: 16-bit Unsigned
- Left of decimal Pt.: Zobrazení počtu celých čísel
- Right of decimal Pt.: Zobrazení počtu míst za desetinnou čárkou

Limits - Direct: nastavili jsme 99, jako maximální zobrazovací číslo.

Security - Zaškrtneme Use interlock function a Enable when Bit is ON a adresu vybereme M7.

Shape - Jelikož nechceme, aby se nám odpočet zobrazoval stále (Ne jen u nastaveného M7), nenastavujeme Shape.

ND_1: Numeric Display Object

Záložky:
General - Read adress: D2

Numeric Format - stejné jako u ND_0.

Security - adresu nastavíme na M8.

SW_0:

General - Write adress: D1
Attribute: Set style: Periodic step down
Low limit: 0, High limit: 30 (jelikož jsme v kódu nastavili 30s)
Dec. Value: 1 (hodnota, po které se bude odečítat)
Time interval: 1s (hodnota se bude odečítat po 1s)

Security - Zaškrtneme Use interlock function, Enable when Bit is ON, adress M7 a nakonec Hide when disabled, aby se nám číslo zobrazovalo pouze při funkci.

Shape, Label - nevyplňujeme.

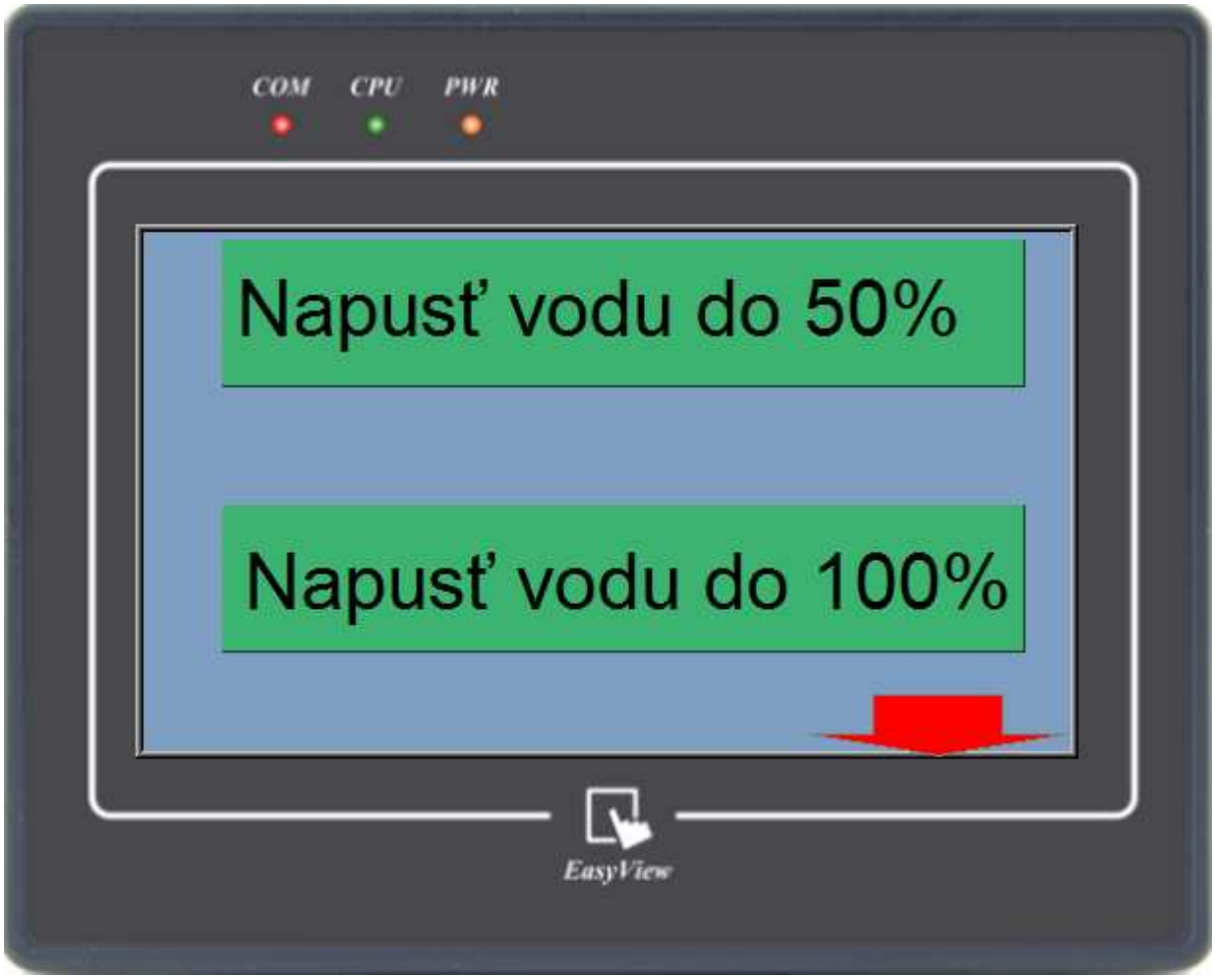
SW_1:

General - Write adress: D2
Attribute: Stejné jako u SW_0

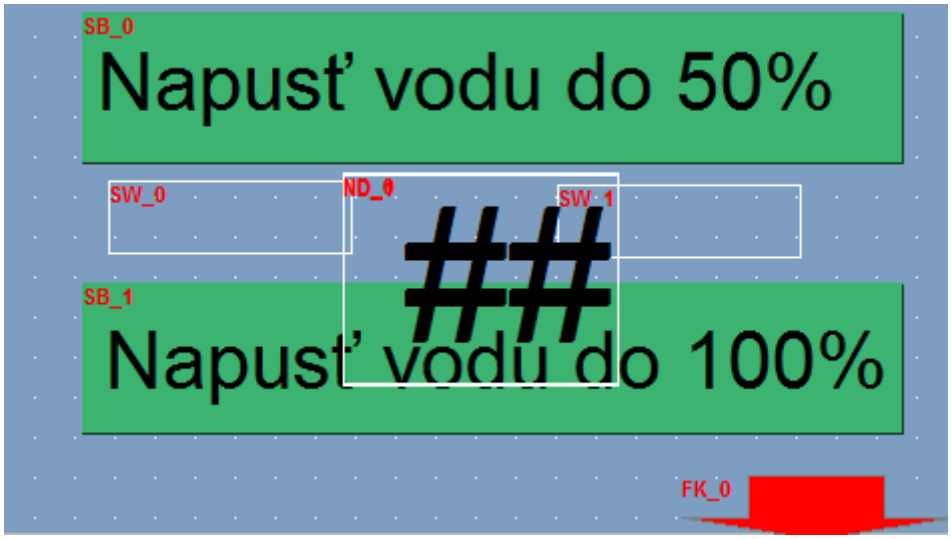
Security - Stejné jako u SW_0, pouze adress M8

Shape, Label - nevyplňujeme

Na *obr*. Můžeme vidět obrazovku s výběrem možnosti napuštění vody do 50% nebo do 100%. Při výběru jedné z možností tlačítka zmizí a zobrazí se zbývající čas k dokončení operace. Červená šipka znázorňuje možnost přepnutí na Úvodní obrazovku.



Obr.70: Výběr napouštění vody



Obr.71:Napouštění vody programátorský režim

SB_0: Set Bit Object
Záložky:

General - Write adress: X10
Attribute: Set style: Momentary

Security - Zaškrtneme Use interlock function, Hide when disabled, Enable when Bit is OFF, adresu nastavíme na Y5.

Shape - Z předchozích objektů již víme, jak nastavit na požadovanou podobu

Label - Zde nastavíme text

SB_1: Set Bit Object

Záložky: Kromě General, kde nastavíme adresu na X11 a v Label změníme text, tak jsou všechny záložky totožné s SB_0.

ND_0: Numeric Display Object

Záložky: Nastavení stejné, jako u předchozích ND, jen v General a Security změníme adresy na D3 a M10.

ND_1: Numeric Display Object

Záložky: Nastavení je stejné, jako u předchozích ND, jen v General a Security změníme adresy na D4 a M11.

SW_0: Set Word Object

Záložky:
General: Write adress: D3
Attribute - Set style: Periodic step down
Low limit: 0, High limit: 12
Dec. Value: 1
Timer interval: 1

Security: Zaškrtneme Use interlock function, Hide when disabled, Enable when Bit is ON, adresu nastavíme na M10.

Shape, Lable - nenastavujeme

SW_1: Set Word Object

Záložky: stejné jako u SW_0, pouze v General nastavíme adresu na D4, High limit na 24 a v Security adresu nastavíme na M11.

FK_0: Function key object

Záložky:
General - Change full-screen window, Window no.: 10

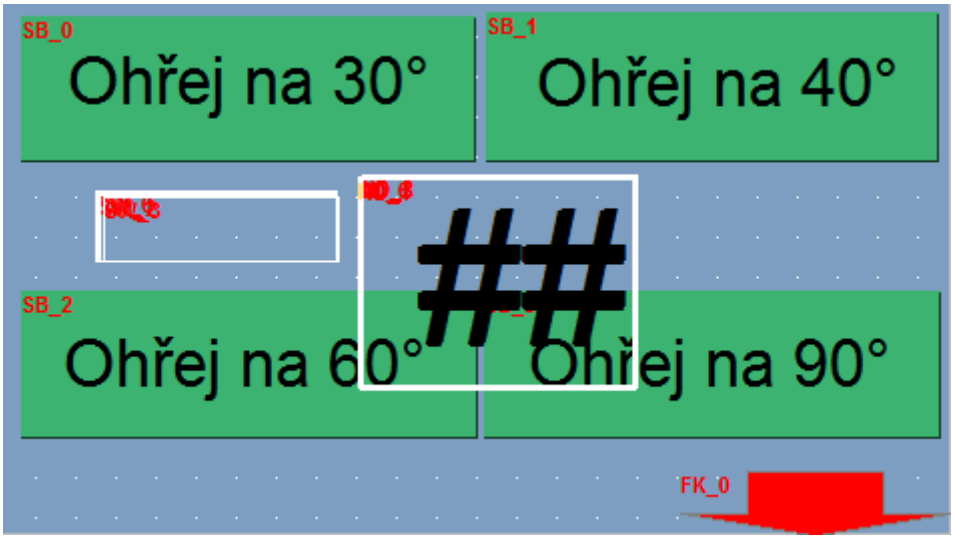
Security - Zaškrtneme Use interlock function, Hide when disabled, Enable when Bit is OFF a adresu nastavíme na Y5.

Shape - Vybereme Use shape a v Shape Library vybereme požadovaný obr..

Label - nenastavujeme

Na *obr.* vidíme obrazovku s možností výběru ohřevu vody. Opět se nám při výběru jedné z možností zobrazí zbývající čas k ukončení úkolu a tlačítka zmizí. Šipka znovu zastupuje vrácení na úvodní obrazovku.

Obr.72: Výběr ohřevu vody



Obr.73:Ohřev vody programátorský režim

SB_0: Set Bit Object

Záložky:
General - Write address: X12
Attribute: Set style: Momentary

Security - Zaškrtneme Use interlock function, Hide when disabled, Enable when Bit is OFF a adresu nastavíme na Y4.

Shape - Zaškrtneme Use shape a vybereme požadovaný tvar a barvu.

Label - Zaškrtneme Use label, do Content napíšeme požadovaný text

SB_1, SB_2, SB_3:
V General nastavíme adresy X13, X14, X15

V Label v políčku Content změníme texty

Ostatní záložky jsou totožné s SB_0.

ND_0: Numeric Display Object

Záložky:
General - Read address: D5

Numeric format - Data format: 16-bit Unsigned
- Left of decimal Pt.: 2, Right of decimal Pt.: 0
- Limits: Direct
- Input low: 0, Input high: 99

Security - Zaškrtnout Use interlock function, Enable when Bit is ON a adresu nastavíme na M12.

Shape- nevyplňujeme.

ND_1, ND_2, ND_3: V General nastavíme adresy na D6, D7, D8 a v Security adresy M13, M14, M15, jinak je nastavení stejné jako u ND_0.

SW_0: Set Word Object

Záložky:
General - Write address D5
Attribute: Set style: Periodic step down
Low limit: 0, High limit: 6
Dec. Value, Time interval: 1

Security - Zaškrtneme Use interlock function, Hide when disabled, Enable when Bit is ON a adresu nastavíme na M12.

Shape, Label - nevyplňujeme.

SW_1, SW_2, SW_3:

V General nastavíme adresy na D6, D7, D8 a High limity na: 9, 14, 26.

V Security nastavíme adresy na M13, M14, M15, jinak jsou všechna nastavení totožná s SW_0.

FK_0: Function Key Object

Nastavení je stejné jako FK_0 u předchozí obrazovky, pouze v Security je adresa Y4.

ZÁVĚR

Cílem této bakalářské práce bylo seznámit se s modely EDU-mod, programovatelným automatem Mitsubishi FX3U-32M a vývojovým prostředím GX IEC Developer. Modely byly křižovatka a automatická pračka. Pro každý model vytvořit 3 úlohy, které jsem měl zrealizovat ve čtyřech programovacích jazycích. Tyto jazyky jsou instruction list(IL), structured text(ST), ladder diagram (LD) a function block diagram (FBD).
V první kapitole jsem představil hardwarovou stránku modulů.
V druhé kapitole jsem popsal jak se vytváří nový projekt v GX IEC Developeru. Dále jsem popsal vlastnosti tohoto vývojového prostředí. Znázornil jsem syntaxi zápisu základních příkazů a funkčních bloků.
Ve třetí kapitole byl ukázán programovatelný automat Mitsubishi FX3U-32M, jeho vlastnosti, specifikace a možná rozšíření.
Ve čtvrté kapitole jsem popsal dotykový panel EasyView MT6050i a jeho specifikace.
Pátá kapitola slouží k popsání konfiguračního softwaru EasyBuilder. Převedl jsem vytvoření nového projektu, nastavení připojení s PLC, nebo s dalším panelem. Vyjmenoval jsem použité objekty a ukázal jejich funkci.
Šestá kapitola má za úkol popsat vytvořené úlohy, jejich nastavení v GX IEC Developeru. Každá úloha je popsána vývojovým diagramem. Zdrojové kódy jsou v příloze. Poslední kapitola je rozšíření třetí úlohy automatické pračky o ovládání dotykovým panelem. Byl vytvořen program v EasyBuilderu a popsány vlastnosti všech použitých prvků. Odzkoušení všech programů pomocí PLC, případně pomocí dotykového panelu a proběhlo v pořádku.

SEZNAM POUŽITÉ LITERATURY

[1] Ing. Kohout, Luděk. Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří [online]. [cit. 2012-01-13]
Dostupné z: <http://www.edumat.cz/produkty.php>

[2] Ing. Kohout, Luděk. Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří [online]. [cit. 2012-01-13]
Dostupné z: < http://www.edumat.cz/produkty.php?produkt=krizovatka >

[3] Ing. Kohout, Luděk. Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří [online]. [cit. 2012-01-13]
Dostupné z: <http://www.edumat.cz/produkty.php?produkt=pracka >

[4] Mitsubishi. Mitsubishi Electric [online]. [cit. 2012-01-25]

[5] Tecon. EasyView MT6000 grafické operátorské panely | TECON s.r.o [online]. [cit. 2012-02-15]
Dostupné z: <http://www.tecon.cz/prod_panely_easy6.php>

[6] Tecon. Manuály a technické specifikace | TECON s.r.o. [online]. [citace 2012-02-27]
Dostupné z: <http://www.tecon.cz/pdf/EB8000_User_Manual.pdf>

SEZNAM PŘÍLOH

Příloha č. 1 – Tento dokument v elektronické podobě.
Příloha č. 2 – Zdrojové kódy na cd