

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 S7- 1200

EDU-MOD MODEL CONTROL BY PROGRAMMABLE LOGIC CONTROLLER S7-1200

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL POLÁK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D.

BRNO 2013

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Pavel Polák

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 S7-1200

v anglickém jazyce:

EDU-mod model control by programmable logic controller S7-1200

Stručná charakteristika problematiky úkolu:

V laboratoři programovatelných automatů A1/731a jsou čtyři modely soustav EDU-mod. Jedná se o soustavu posuvové jednotky, mísicí jednotky, automatické pračky a křižovatky.

Cílem této BC práce je navrhnout a realizovat řízení modelu automatické pračky a dopravní křižovatky programovatelným automatem Siemens Simatic S7-1200. Dalším cílem je 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 Siemens S7-1200.
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 Siemens pro Simatic S7-1200.

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

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

V Brně, dne 20.11.2012

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

ABSTRAKT

Tato bakalářská práce se zabývá vytvořením několika vzorových řešení k novým programovatelným automatům Siemens Simatic S7-1200. Obsah bakalářské práce bude sloužit jako podpora pro studenty ústavu Automatizace a Informatiky na FSI VUT. Studenti zde mohou čerpat informace ohledně parametrů nových programovatelných automatů, funkcí a vzorových zadání pro samotnou výuku a samostudium. Programování úloh je realizováno pomocí jazyka SCL. Dále je v této bakalářské práci popsáno rozšíření pro HMI dotykové panely a web server.

ABSTRACT

This bachelor thesis deals with creating a few sample solutions to new programmable logic controllers, Siemens Simatic S7-1200. The content of this thesis will serve as support for students of Institute of Automation and Computer Science at the Faculty of Mechanical Engineering. Students can gather information about the performance of new programmable logic, functions and model specified for the education and self-study. Programming tasks are performed using the SCL language. Furthermore, in this thesis describes an extension for HMI touch panels and web server.

KLÍČOVÁ SLOVA

Simatic S7-1200, HMI panel, web server, dopravní křižovatka, automatická pračka.

KEYWORDS

Simatic S7-1200, HMI panel, web server, traffic intersection, automatic washing machine.

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a že všechny použité literární zdroje jsem správně a úplně citoval. Bakalářská práce je z hlediska obsahu majetkem Fakulty strojního inženýrství VUT v Brně a může být využita ke komerčním účelům jen se souhlasem vedoucího práce a děkana FSI VUT.

podpis studenta

Poděkování:

*Rád bych poděkoval Ing. Tomáši Maradovi, Ph.D.
za rady poskytnuté při tvorbě této bakalářské práce.*

BIBLIOGRAFICKÁ CITACE

POLÁK, P. Řízení modelů procesů EDU-mod programovatelným automatem řady S7-1200. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 63 s. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D..

Obsah:

	Zadání závěrečné práce.....	3
	Abstrakt.....	5
	Prohlášení o originalitě.....	7
1	Úvod.....	11
2	Cíl práce.....	13
3	Programovatelný automat Siemens Simatic S7-1200.....	15
4	HMI Panel KTP400 Basic color.....	17
5	Výukové Modely.....	19
5.1	Modely EDU-mod.....	19
5.2	Model automatické pračky.....	19
5.2.1	Popis modelu.....	19
5.2.2	Funkce modelu.....	19
5.2.3	Inicializační stav.....	20
5.2.4	Chybová hlášení.....	20
5.3	Model dopravní křižovatky.....	20
5.3.1	Popis modelu.....	20
6	Desky plošných spojů.....	21
7	Softwarové vybavení.....	23
7.1	Instalace základní verze TIA Portal V11.....	23
7.2	Instalace aktualizace verze 5.....	25
7.3	Rozšíření hardwaru v prostředí TIA Portal.....	25
7.4	Přidání PLC do projektu.....	26
7.5	Přidání HMI do projektu.....	28
7.6	Představení prostředí Tia Portal.....	29
8	Model automatické pračky.....	31
8.1	Použité pomůcky.....	31
8.2	Schéma zapojení.....	31
8.3	Zadání příkladu 1:.....	31
8.3.1	Časovač.....	33
8.3.2	Vypracované řešení příkladu 1.....	34
8.4	Zadání příkladu 2:.....	36
8.4.1	Vypracované řešení příkladu 2.....	36
8.5	Zadání příkladu 3.....	39
8.5.1	Vypracované řešení příkladu 3.....	40
9	Model dopravní křižovatky.....	43
9.1	Použité pomůcky.....	43
9.2	Schéma zapojení.....	43
9.3	Zadání příkladu 1:.....	44
9.3.1	Vypracované řešení příkladu 1.....	47
9.4	Zadání příkladu 2:.....	49
9.4.1	Vypracované řešení příkladu 2.....	50
9.5	Zadání příkladu 3.....	51
9.5.1	Rozšíření s HMI panelem.....	52
9.5.2	Rozšíření s Web Serverem.....	54
10	Závěr.....	61
	Seznam použité literatury.....	63

1 ÚVOD

V této bakalářské práci jsou představeny programovatelné automaty řady Simatic S7-1200 a práce s nimi. Dále je ukázána práce s HMI panely a web serverem. Tato bakalářská práce je koncipována jako návod pro studenta, který prozatím nemá žádné zkušenosti s programováním projektů pomocí jazyka SCL do programovatelných automatů Simatic S7-1200.

Nejprve je student seznámen s hardwarovými specifikacemi programovatelného automatu Simatic S7-1200 a dotykového panelu KTP400. Poté je stručně zobrazena instalace prostředí TiA Portal V11, ve kterém probíhá programování všech úkolů. Toto prostředí je nutné pro nahrávání vytvořeného programu do programovatelného automatu. Dále jsou představeny EDU-mod modely, na kterých probíhá ověřování funkčnosti vytvořených projektů. Jedná se o modely automatické pračky a dopravní křižovatky. Pro každý model jsou vytvořena 3 vzorová zadání. Zadání jsou seřazena od nejjednoduššího po složitější, přičemž u složitějšího vypracování se počítá s úspěšným zvládnutím předešlých úloh.

U modelu automatické pračky si student nejdříve vytvoří jednoduchý prací cyklus. Tento cyklus je následně rozšířen o další cyklus a to je proces předeprání a ovládání procesů pomocí HMI panelu. Na HMI panelu si uživatel zvolí, který prací cyklus má být vykonán. Poslední úkol u modelu automatické pračky je rozšíření o servisní mód, kdy technik může upravit teplotu a čas praní. Přístup do servisního módu je zabezpečen zadáním správného uživatelského jména a hesla.

U modelu dopravní křižovatky si student nejdříve vytvoří tzv. „denní režim“. Tento režim je klasický stav dopravní křižovatky. V dalším úkolu je dopravní křižovatka rozšířena o tzv. „noční režim“. Poslední úkol je rozdělen na 2 pod-úkoly, kdy v prvním je možné časy provozu na dopravní křižovatce nastavit pomocí HMI panelu. Druhý úkol je rozšíření dopravní křižovatky o komunikaci s web serverem, kdy dotyčná osoba může zobrazovat stav křižovatky a upravovat časy provozu na dopravní křižovatce pomocí webového rozhraní.

Poslední, nejsložitější úkoly jsou určeny spíše pro samostatnou práci studenta v jeho volném čase, a to z důvodu časové náročnosti těchto úkolů.

2 CÍL PRÁCE

1. Seznámit se s modely soustav EDU-mod v laboratoři A1/731a.
2. Seznámit se s programovatelným automatem Siemens S7-1200.
3. Vytvořit vzorová zadání s vypracováním řízení úloh.
4. Ověřit funkčnost řízením pomocí PLC.

3 PROGRAMOVATELNÝ AUTOMAT SIEMENS SIMATIC S7-1200



Obr. 1. PLC Siemens Simatic S7-1200 CPU 1214C.

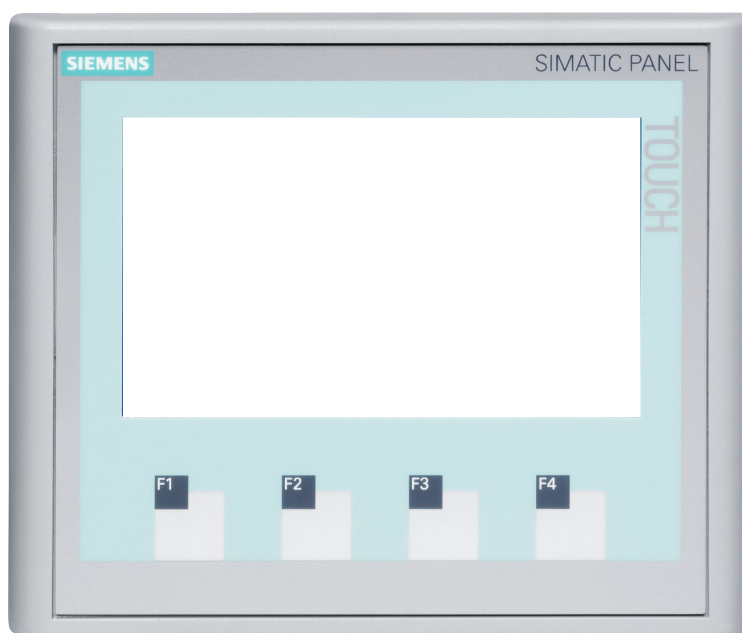
V případě této bakalářské práce se jedná konkrétně o model CPU 1214C. Tento model se vyznačuje následujícími technickými parametry:

Integrované I/O	CPU 1214C
Integrovaní digitální I/O	14 vstupů / 10 výstupů
Integrovaní analogové I/O	2 vstupy / 0 výstupů
Maximální počet I/O – Digitální	284
Maximální počet I/O – Analogové	67
Integrovaná uživ. paměť	
Pracovní paměť	50 KB
Load paměť	2 MB
Zálohovací paměť	2 KB
Bit Memory (M)	8 KB
Výkon	
Rychlost vykonávání bitových instrukcí	0,1 μ s/instrukce
Rychlost přenosu dat typu Word	12 μ s/instrukce
Rychlost mat. operací typu Real	18 μ s/instrukce

Tab. 1. Technické specifikace modelu CPU 1214C [6].

Simatic S7-1200 představuje kompaktní, modulární a moderní řídicí systém. Komunikační rozhraní splňuje ty nejvyšší požadavky na průmyslovou komunikaci. K dispozici je až 50 KB integrované paměti s „plovoucí“ hranicí mezi programem a uživatelskými daty. Profinetové rozhraní slouží pro spojení s programovacím zařízením, komunikaci s dalšími PLC a HMI. Podporuje otevřenou komunikaci na ethernetu TCP/IP native, ISO na TCP a UDP. Dále je na S7-1200 integrováno až 6 vysokorychlostních snímačů. Tři mohou čítat frekvenci 100 KHz a tři 30 KHz. [7]

4 HMI PANEL KTP400 BASIC COLOR



Obr. 2. Panel KTP400 Basic color.

Panel KTP400 se vyznačuje jednoduchým ovládáním a spojením s ostatními zařízeními pomocí rozhraní PROFINET. Tento panel je určen spíše pro jednodušší projekty. Avšak díky svojí velikosti je vhodný pro stísněné prostory přímo ve stroji. Panel lze instalovat jak na šířku, tak i na výšku. Tento model se vyznačuje následujícími technickými parametry:

Display	4,3" TFT širokoúhlý
Počet barev	256
Rozlišení	480 x 272 px
Integrovaná tlačítka	4 programovatelná tlačítka
Vlastní paměť	512 KB
Rozhraní	1 x RJ 45 pro PROFINET
Montážní výřez	123 x 99 mm
Šířka panelu	34 mm

Tab. 2. Technické parametry panelu KTP400 Basic color.

5 VÝUKOVÉ MODELÝ

5.1 Modely EDU-mod

Slovní spojení EDU-mod znamená označení skupiny modelů, které simulují různé technologické procesy. Modely jsou určeny především pro výuku logických systémů, které jsou realizovány především pomocí programovatelných automatů. Společnost, která vyrábí tyto modely je vyrábí ve dvou řadách – logické signály s úrovní 24V ss a 5V ss. V této bakalářské práci jsou použity modely o úrovní 24V ss.

Připojení k programovatelnému automatu je realizováno pomocí 20ti pólového konektoru, který je umístěn na každém modelu. Tento konektor je propojen plochým kabelem s deskou plošných spojů, která byla za tímto účelem vyrobena k programovatelnému automatu. [8]

5.2 Model automatické pračky



Obr. 3. Model automatické pračky.

5.2.1 Popis modelu

Tento model simuluje pracovní režim automatické pračky. Jedná se o aktivní model, což znamená, že model obsahuje procesorovou jednotku. Funkce procesorové jednotky spočívá v tom, že jednotka řídí napouštění, ohřev a chlazení vody. Signalizuje jaké množství vody se právě nachází v prací vaně. Dále zajišťuje ohřev, či chlazení vody. Při ohřívání a chlazení vody se model chová jako soustava 2. řádu. Časové konstanty jsou zkráceny tak, aby se na ohřátí vody nemuselo příliš dlouho čekat. Pokud je vana plná vody (Hladina = 100%), tak ohřev na 90°C trvá cca 1 min. [9]

5.2.2 Funkce modelu

Řízení modelu je zajištěno 6 binárními výstupy programovatelného automatu – akční členy. Tyto výstupy jsou na úrovni 24V logiky (společná GND).

Dva výstupy slouží k otáčení bubnu. První výstup slouží k otáčení bubnu doprava, druhý výstup slouží k otáčení bubnu doleva. Stav otáčení bubnu je signalizován LED diodou u označení na modelu „Buben ->“. Stav otáčení bubnu doleva signalizuje LED u označení „Buben <-“. Dále je pohyb otáčení bubnu signalizován na kruhovém uspořádání LED. Signalizace probíhá tzv. „běžícím světlem“. Běžící světlo znázorňuje jak směr pohybu, tak i zrychlený pohyb při ždímání. Rychlost otáčení bubnu se řídí výstupem s názvem Otáčky. Signalizace otáček je opět znázorněna LED diodou. Pokud na výstup Otáčky přivedeme logickou 1, tak se otáčky zvýší a probíhá ždímání. V opačném případě (logická 0) jsou otáčky nižší a probíhá praní.

Další výstup je nazván jako Voda. Tento výstup zajišťuje napouštění vody do prací vany. Pokud je na výstupu u bitu Voda logická 1, tak svítí LED u tohoto výstupu a dochází k napouštění vody do automatické pračky. Výstup označený jako Topení zajišťuje ohřev vody. Pokud má výstup hodnotu logické 1, tak dochází k ohřevu, v opačném případě dochází k chladnutí vody v prací vaně. Poslední výstup je označen jako Čerpadlo, pokud je na bitový výstup přivedena logická 1, tak dochází k vypouštění vody z automatické pračky.

Model obsahuje 6 logických vstupů programovatelného automatu. První dva se týkají množství vody v prací vaně. Vstupy jsou signalizovány hladinou 50% a 100% objemu prací vany. Další 4 vstupy zajišťují teplotu vody v prací vaně. Signalizovány jsou teploty vody 30°C, 40°C, 60°C a 90°C. Ohřev a chladnutí vody zajišťuje procesorová jednotka modelu. [9]

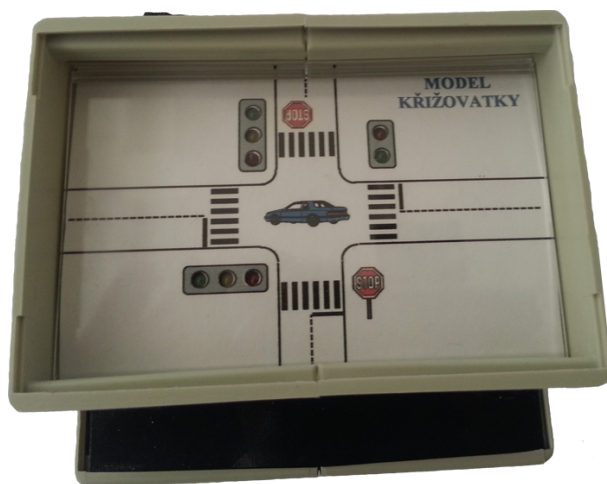
5.2.3 Inicializační stav

Po zapnutí a při stisknutí tlačítka RESET je model uveden do inicializačního stavu. Inicializační stav se vyznačuje tím, že jsou na vstupy nastaveny logické 0, tzn. že je buben prázdný a teplota je nastavena na počáteční. [9]

5.2.4 Chybová hlášení

V modelu jsou generovány 2 druhy chyb. Chyba opravitelná – tato chyba je indikována blikáním LED diody ERR. Chyba nastane v případě, že na výstupy přijde režim „Buben ->“ zároveň s režimem „Buben <-“. Buben se při této chybě přestane otáčet. Chybu můžeme opravit tak, že přivedeme na jeden z výstupů logickou 0. Poté se buben začne otáčet v daném směru a pračka je v normálním režimu. Chyba neopravitelná – tato chyba je indikována rozsvícením LED diody ERR. Chyba může nastat ve dvou případech. Jedním z případů je přetečení vody v prací vaně. Druhým případem je ohřátí vody na vyšší teplotu než 90°C. Stav neopravitelné chyby je možné vyřešit pouze stisknutím tlačítka RESET a uvedením modelu do inicializačního stavu. [9]

5.3 Model dopravní křižovatky



Obr. 4. Model dopravní křižovatky.

5.3.1 Popis modelu

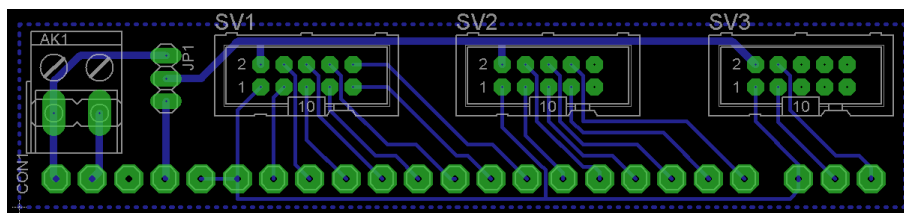
Model dopravní křižovatky neobsahuje žádnou procesorovou jednotku, tzn. model je pasivní. Dopravní křižovatka obsahuje 3 semaforey. Dva semaforey jsou určeny pro vozidla – hlavní a vedlejší komunikace a třetí semafor je určen pro chodce napříč hlavní komunikací. Model obsahuje 8 LED diod – na každém semaforu pro vozidla jsou 3 LED (červená, oranžová a zelená) a u semaforu pro chodce jsou umístěny 2 LED diody (červená a zelená). Model je řízen pouze výstupy PLC.

6 DESKY PLOŠNÝCH SPOJŮ

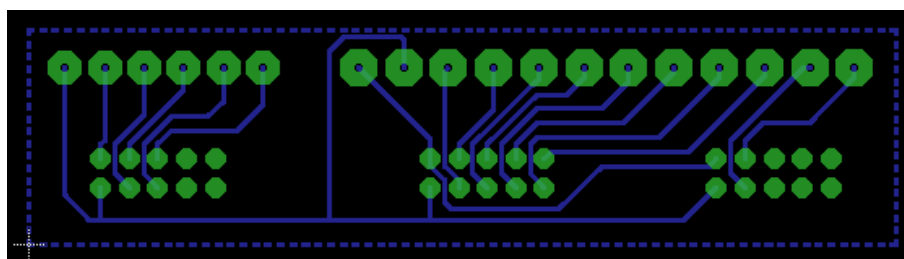
Jelikož programovatelné automaty řady S7-1200 obsahují rozhraní PROFINET, tak nebylo nutné vyrábět speciální propojovací kabely s počítačem. Rozhraní PROFINET obsahují i HMI panely. Takže pouze stačí propojit síťovou kartu počítače se switchem, do kterého je připojen jak programovatelný automat, tak i HMI panel. Jedním krokem konfigurace síťové komunikace bylo přiřazení jedinečné IP adresy v lokální síti. Bez této konfigurace bych nemohl „oslovit“ konkrétní programovatelný automat. Automat, na kterém probíhala tato bakalářská práce, má IP adresu 192.168.0.66. HMI panel má přiřazenou taktéž jedinečnou IP adresu v místní síti, konkrétně v této bakalářské práci se jednalo o IP adresu 192.168.0.67. Dále byly vytvořeny desky plošných spojů.

Abychom mohli lépe propojit modely EDU-mod a programovatelný automat, proto byly vytvořeny desky plošných spojů. Tyto desky byly vyrobeny pro všechny automaty S7-1200, které se v laboratoři nachází. Jelikož tato práce byla časově náročná a složitá na výrobu, tak s touto prací nám ve velké míře pomohl vedoucí této bakalářské práce pan doktor Tomáš Marada.

Nejdříve byly vytvořeny návrhy desek v editoru desek plošných spojů Eagle. Byla vytvořena 2 schémata a to jak pro vstupy programovatelného automatu, tak i pro jeho výstupy. Výroba desek plošných spojů byla společná práce i s dalšími studenty, kteří je taktéž využívají ve své bakalářské, či diplomové práci. Konkrétně se jedná o Jirku Kroupu a Ondřeje Jirčáka.



Obr. 5. Vstupy programovatelného automatu.



Obr. 6. Výstupy programovatelného automatu

Postup prací při výrobě desek plošných spojů:

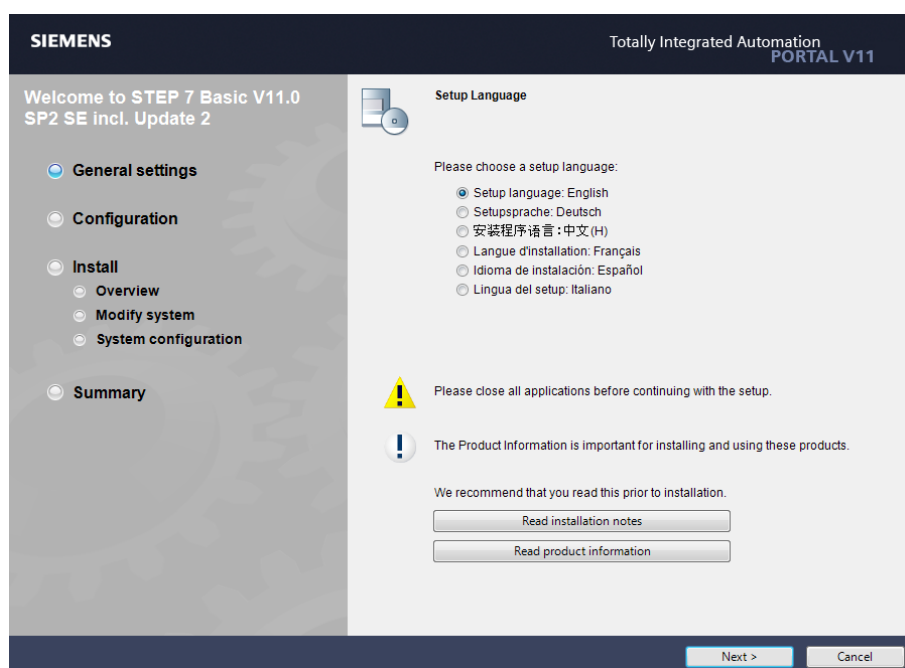
- vytisknutí předlohy z programu Eagle na průhlednou fólii,
- osvit desky,
- leptání v kyselině chlorovodíkové,
- důkladné omytí desky,
- ostřížení přebytečných okrajů a vyvrtání děr,
- obroušení okrajů,
- očištění líhem,
- nanesení pájitelného laku,
- pájení,
- lakování ochranným lakem.

7 SOFTWARE VYBAVENÍ

Nejprve bylo nutné na počítač, kde bylo prováděno psaní aplikace, nainstalovat potřebný software. Jednalo se o program od společnosti Siemens, konkrétně TIA Portal V11. Tento software je poskytovaný společně s programovatelným automatem Simatic S7-1200. Jelikož od vydání verze, která byla přiložena k automatu, došlo k několika opravám programu TIA Portal V11, tak ještě bylo nutné jej aktualizovat na nejnovější verzi.

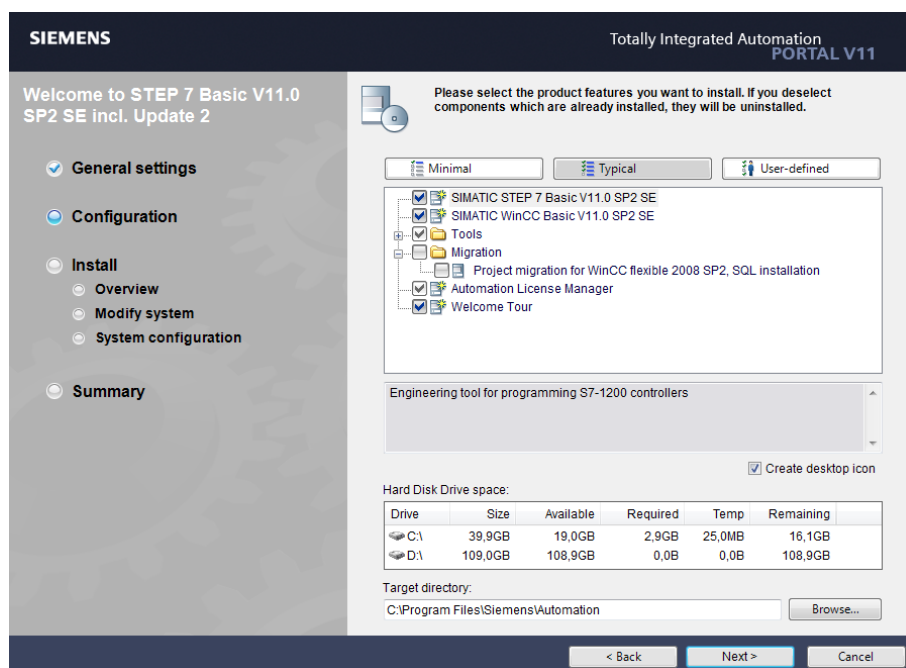
7.1 Instalace základní verze TIA Portal V11

Po vložení instalačního DVD do diskové mechaniky počítače se zobrazila následující obrazovka Obr. 7. Program neobsahuje českou jazykovou mutaci, proto bylo vybráno anglické jazykové prostředí. Po přečtení instalačních instrukcí jsem mohl pokračovat na další krok kliknutím na tlačítko Next.



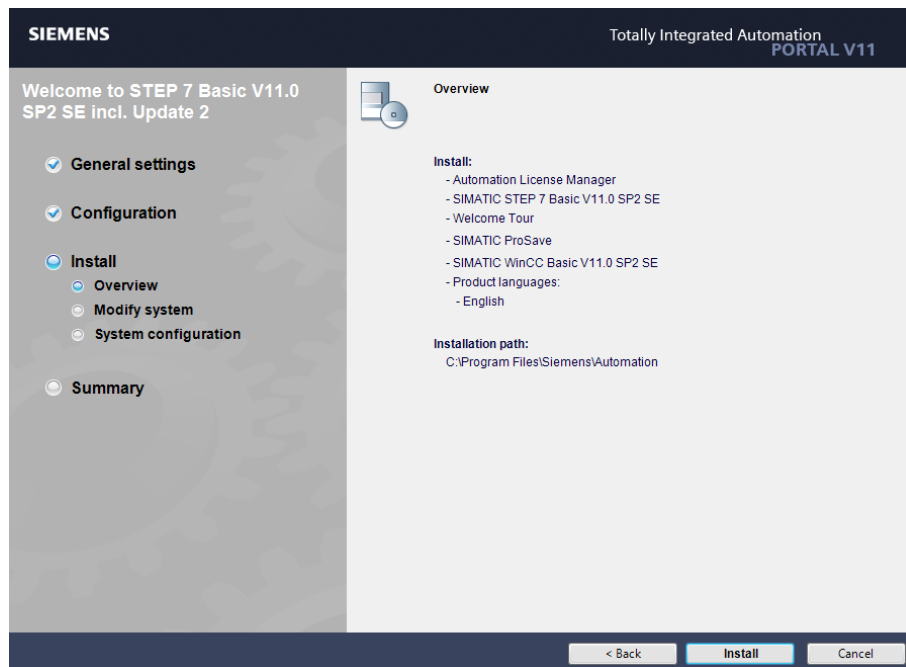
Obr. 7. Výběr jazyka v programu TIA Portal V11.

Na další obrazovce jsem byl vyzván, zda chci do programu přidat i některou další jazykovou mutaci. Čeština opět chyběla, tak jsem zvolil angličtinu. V dalším kroku bylo nutné označit, které moduly chci do TIA Portalu přidat. Vybral jsem takové, jak je vidět na obrázku Obr. 8. Což je STEP 7 – základní program. WinCC – poskytuje nástroje pro vývoj uživatelských aplikací HMI. [1] Tools – zde je např. modul ProSave, který zajišťuje zálohu a obnovu projektů. [2] Automation License Manager – prostředí pro přidávání/odebírání licenčních klíčů. Dále se musela vybrat cesta, kam se má nový software nainstalovat. Po vyplnění všech těchto parametrů jsem opět mohl pokračovat v instalaci kliknutím na tlačítko Next.



Obr. 8. Výběr modulů v programu TIA Portal.

V dalším kroku je nutné si přečíst licenční podmínky a pokud s nimi daný uživatel souhlasí, tak může pokračovat v instalaci. Já jsem s těmito podmínkami souhlasil. Poté se uživateli zobrazí rekapitulace instalovaných atributů. Pokud je vše v pořádku, tak kliknutím na tlačítko Install započne instalace programu.

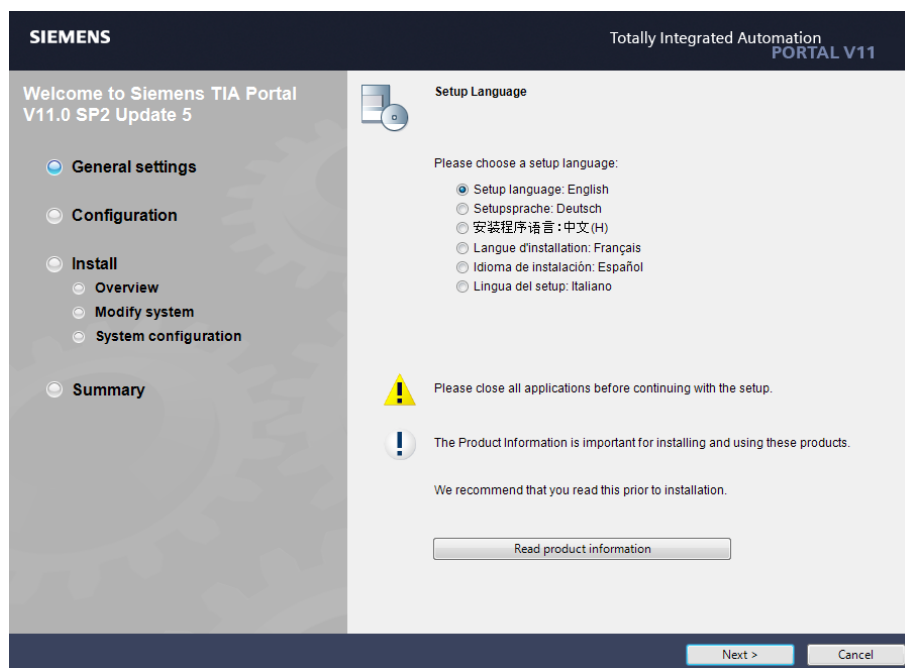


Obr. 9. Výběr modulů v programu TIA Portal.

Po nainstalování všech modulů, které jsem si zvolil, jsem byl vyzván, abych vložil licenční klíč. Tento klíč se nacházel na přiloženém flash disku. Po vložení licenčního klíče byla instalace ukončena a počítač musel být restartován.

7.2 Instalace aktualizace verze 5

Vzhledem k tomu, že výrobce vydal několik opravných balíčků základní verze TIA Portal V11, tak i já jsem nainstaloval nejnovější aktualizaci, která v té době byla aktuální. Aktualizační soubory byly staženy z internetové stránky výrobce. [3] Stejně jako u instalace základního programu jsem nejdříve musel vybrat jazyk prostředí. Výrobce stále neposkytoval český jazyk, proto jsem se rozhodl pro anglický jazyk.



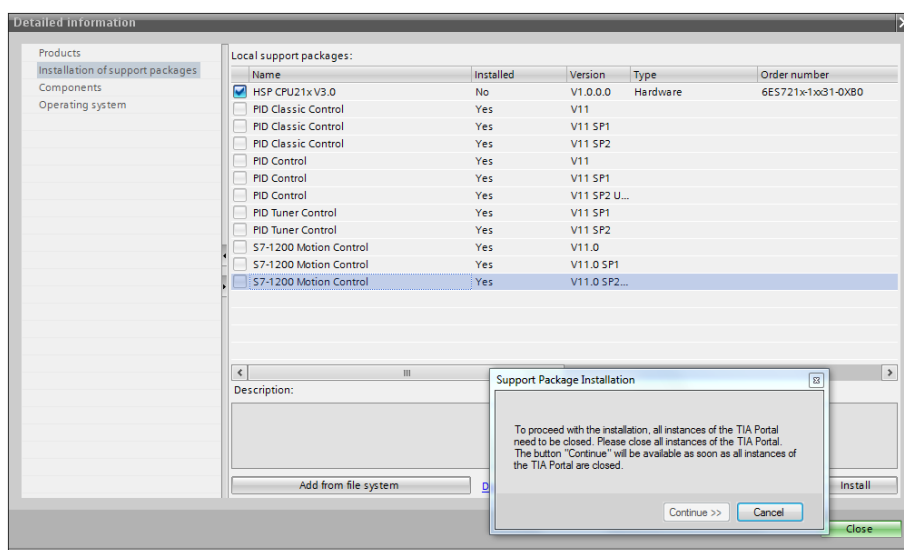
Obr. 10. Instalace aktualizace verze 5.

Na další obrazovce bylo nutné si přečíst licenční podmínky a pokud jsem chtěl v instalaci pokračovat, tak jsem s nimi musel souhlasit.

Po rekapitulaci atributů a krátké době samotné instalace souborů se zobrazilo okno informující o úspěšném dokončení instalace a byl jsem opět vyzván k restartování počítače.

7.3 Rozšíření hardwaru v prostředí TIA Portal

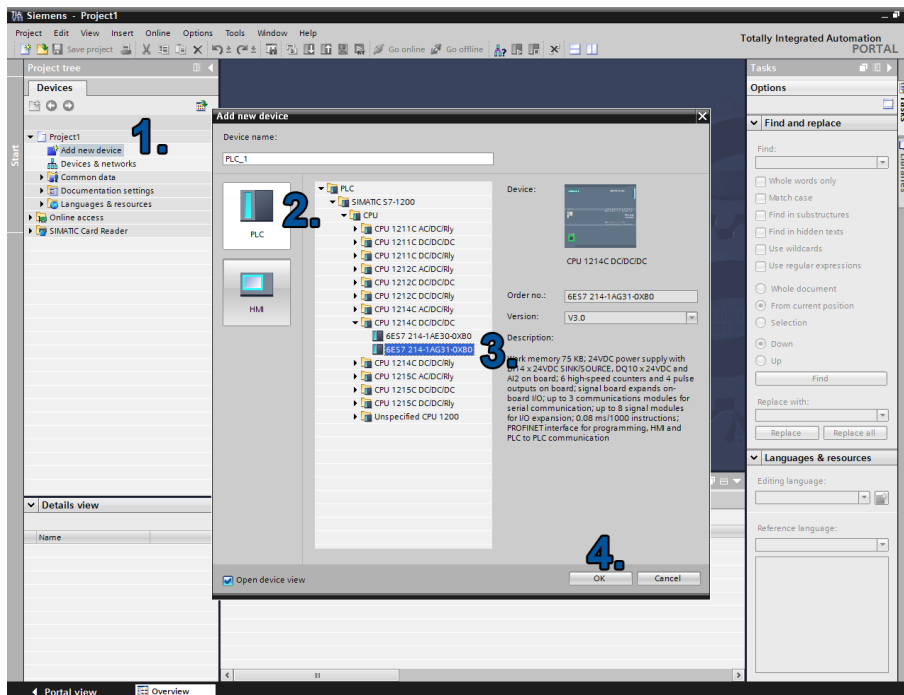
Jak je uvedeno na webu Foxon.cz [4], je ještě nutné aktualizovat hardwarový katalog v prostředí TIA portal. Bez této aktualizace bohužel PLC Simatic S7-1200 v nabídce zařízení nenajdeme. Aktualizační soubor jsem stáhnul ze stránek výrobce. [5] Rozšíření hardwaru byl poslední krok instalace, toto okno je zobrazeno na obrázku Obr. 11. Nyní už bylo na počítači nainstalováno a připraveno veškeré softwarové vybavení pro práci na této bakalářské práci.



Obr. 11. Aktualizace hardwarového katalogu.

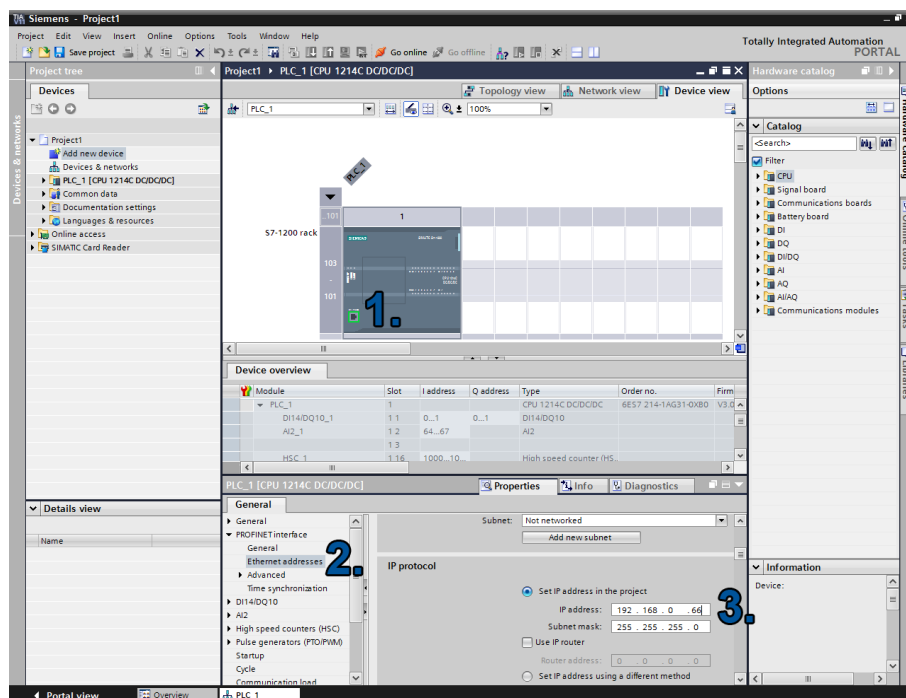
7.4 Přidání PLC do projektu

Před programováním aplikace je nutné do vytvořeného projektu přidat správný PLC. V levém panelu klikneme na položku Add new device. Otevře se okno, ve kterém klikneme na tlačítko PLC. Dále vybereme příslušný programovatelný automat – CPU 1214C DC/DC/DC a vybereme podkategorii 1AG31 a klikneme na tlačítko OK. Vše je zobrazeno na obrázku Obr. 12.



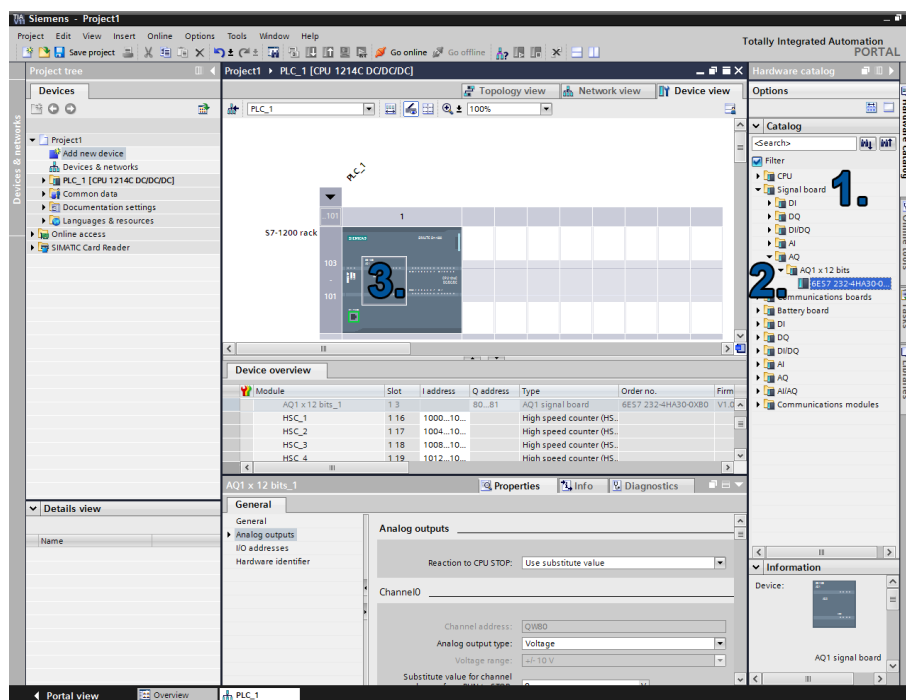
Obr. 12. Výběr typu programovatelného automatu.

V dalším kroku je nutné nastavit IP adresu příslušného automatu. U obrázku PLC klikneme na slot rozhraní PROFINET, otevře se okno s nastavením. V nabídce klikneme na položku PROFINET interface a podkategorii Ethernet addresses. Do pole IP address zadáme IP adresu programovatelného automatu, v mém případě 192.168.0.66.



Obr. 13. Nastavení komunikace s PLC.

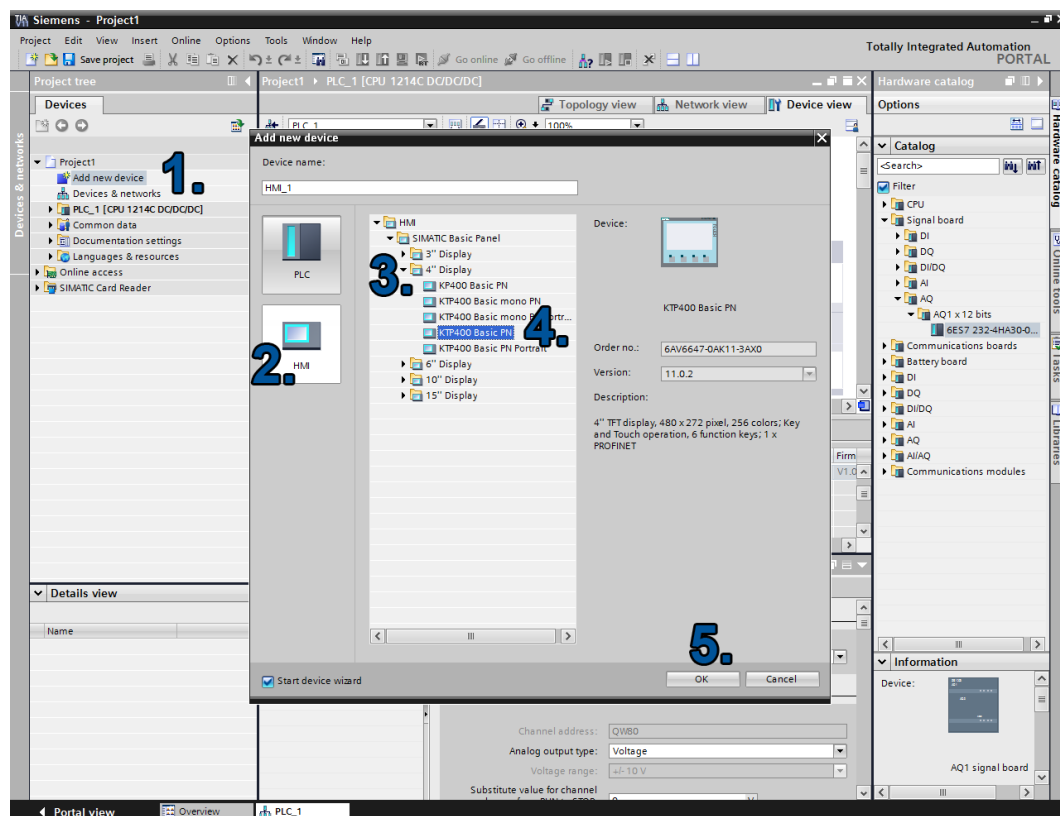
Ještě je nutné nastavit typ analogových výstupů. Toto nastavení provedeme následovně. V pravém panelu Catalog klikneme na nabídku Signal board, poté na položku AQ, dále vybereme položku AQ1 x 12 bits a nakonec jedinou položku, která v této kategorii je 6ES7 232-4HA30... přetáhneme do čtverce, který je znázorněn na obrázku PLC, vše je znázorněno na obrázku Obr. 14. Toto byl poslední krok nastavení přidání PLC do projektu.



Obr. 14. Nastavení analogových výstupů.

7.5 Přidání HMI do projektu

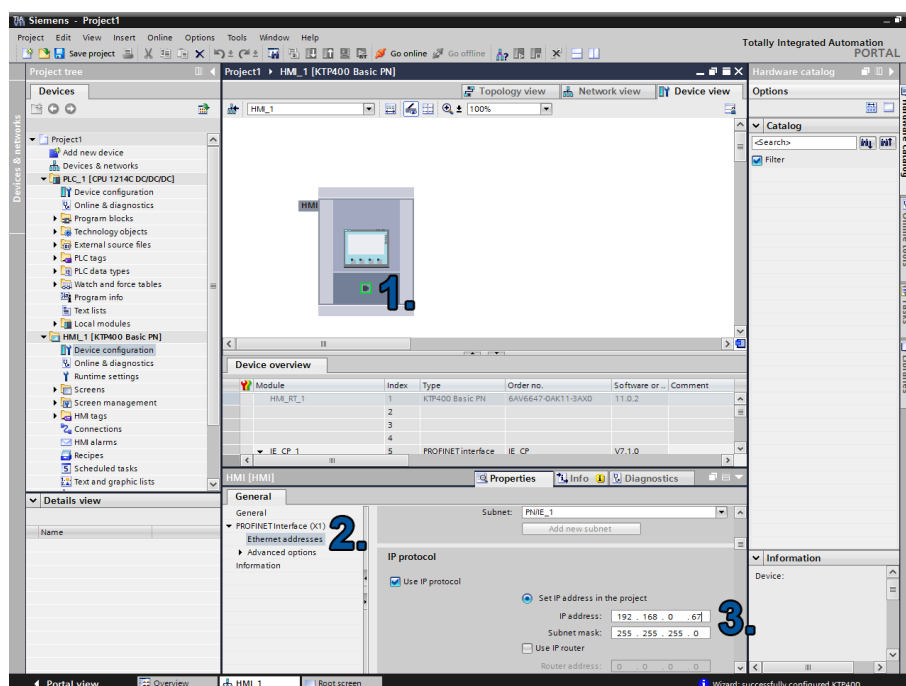
V některých úkolech v této bakalářské práci je využit HMI panel. Dotykový panel musíme do projektu nejdříve přidat obdobně jako samotný programovatelný automat. Nejdříve klikneme na tlačítko Add new device, vybereme tlačítko HMI. Zobrazí se nová nabídka, kde klikneme na položku 4" Display a podkategorii KTP400 Basic PN. A klikneme na tlačítko OK, vše je ukázáno na obrázku Obr. 15.



Obr. 15. Přidání HMI panelu KTP400.

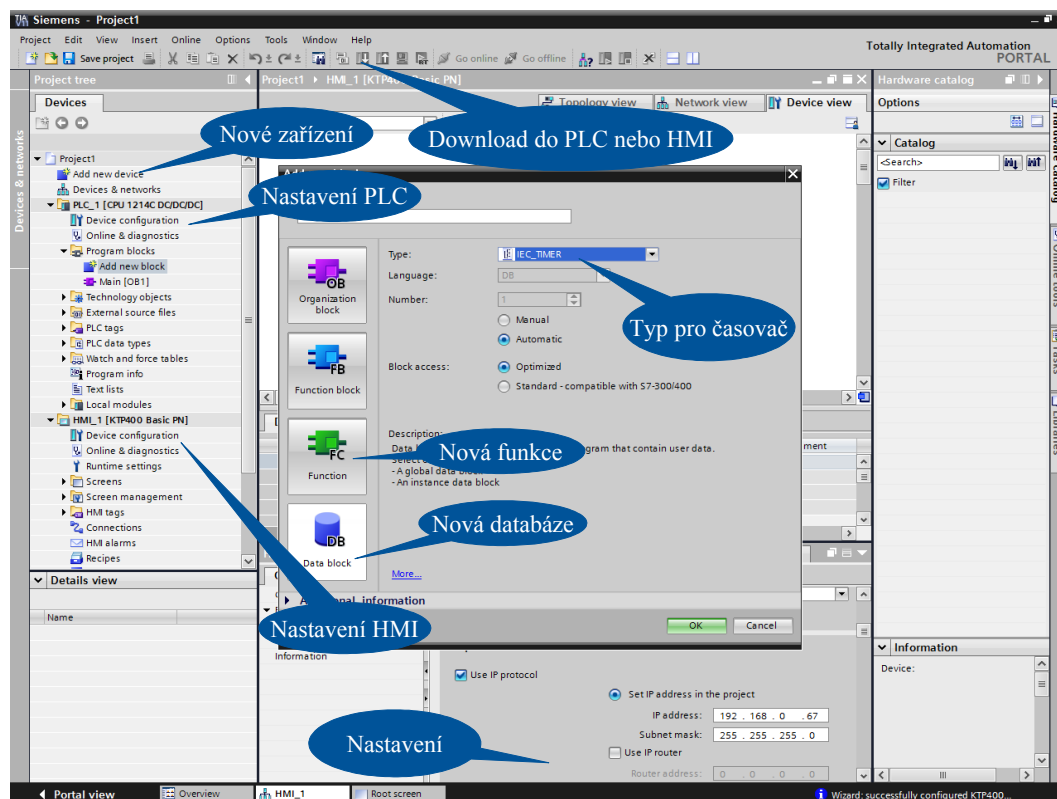
V dalším kroku jsme vyzváni k výběru programovatelného automatu, ke kterému chceme přiřadit HMI panel. Z rozevíracího seznamu vybereme automat, který jsme přidali v kategorii 7.4 a klikneme na tlačítko FINISH, jelikož není nutné nastavovat další parametry.

V posledním kroku je nutné příslušnému HMI panelu přiřadit IP adresu. Stejně jako u nastavení PLC klikneme na slot pro rozhraní PROFINET a zobrazí se nastavení. V nastavení nalezneme položku PROFINET Interface (X1) a v ní podkategorii Ethernet addresses. Do pole IP address napíšeme adresu HMI panelu, v mém případě 192.168.0.67. Pomocí této IP adresy bude automat komunikovat s HMI panelem. Nastavení datové komunikace je zobrazeno na obrázku Obr. 16.

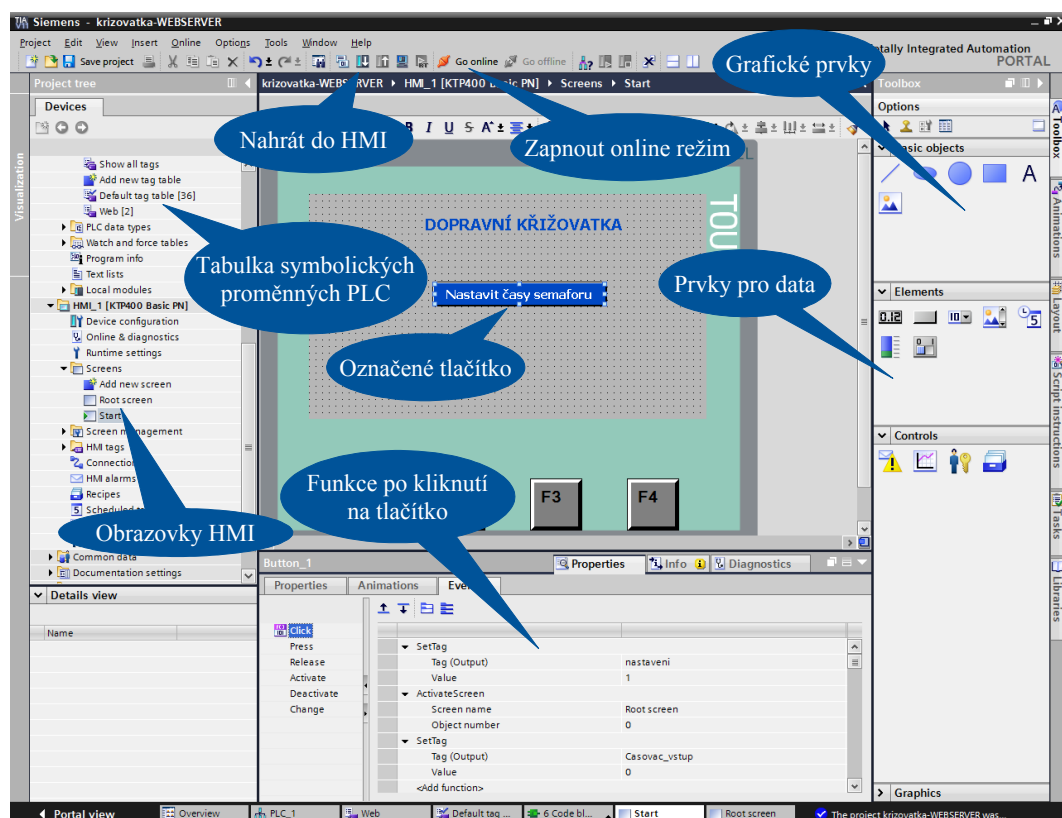


Obr. 16. Nastavení komunikace HMI s PLC.

7.6 Představení prostředí TiA Portal

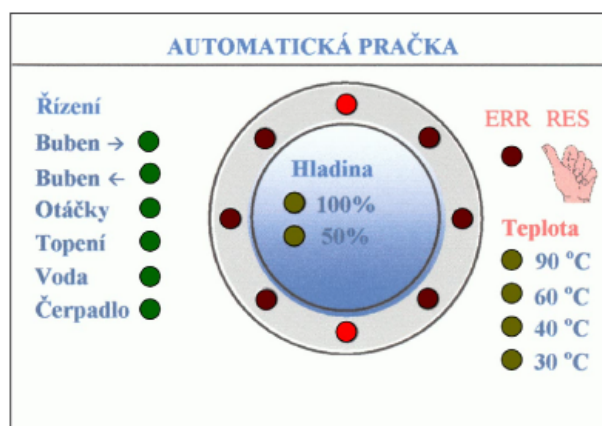


Obr. 17. Prostředí TiA Portal při přidávání nové funkce do PLC.



Obr. 18. Prostředí TIA Portal při vytváření projektu do HMI panelu.

8 MODEL AUTOMATICKÉ PRAČKY

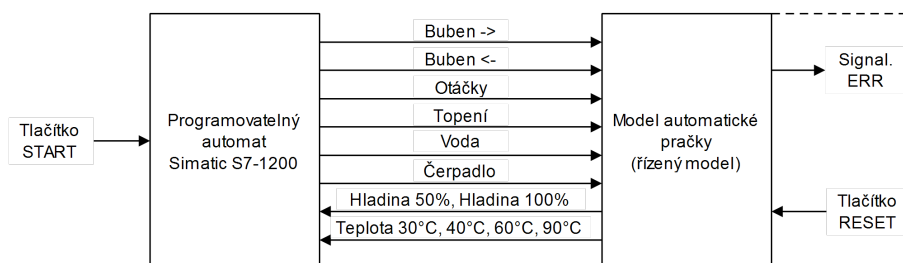


Obr. 19. Znáznornění modelu automatické pračky.

8.1 Použité pomůcky

- Programovatelný automat Simatic S7-1200 CPU 1214C,
- HMI panel KTP400 Basic color,
- model automatické pračky (6 digitálních vstupů, 6 digitálních výstupů),
- modul tlačítek.

8.2 Schéma zapojení



Obr. 20. Schéma zapojení modelu automatické pračky.

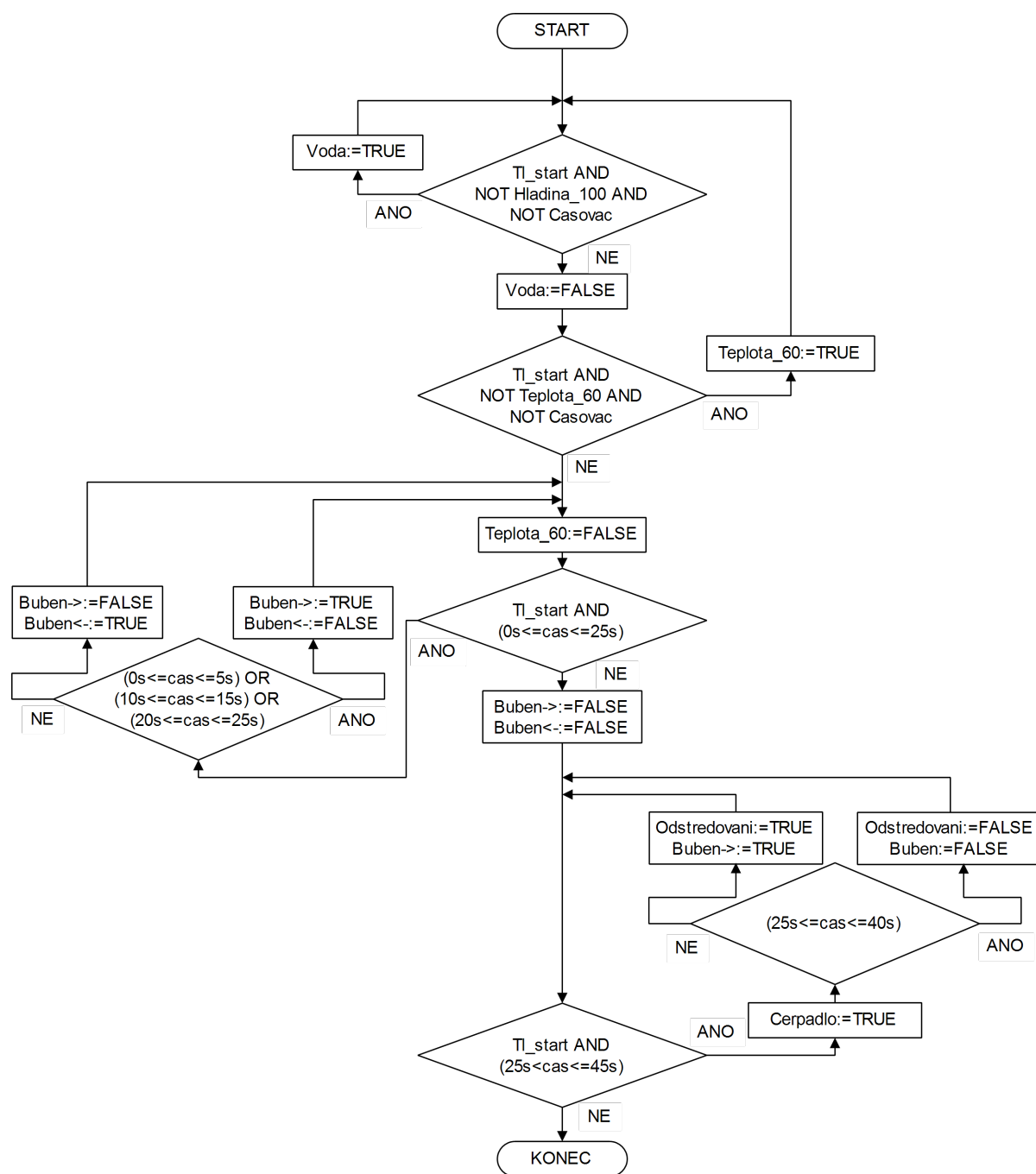
Na obrázku Obr. 20. je zobrazeno schéma propojení modulu spínačů (příp. HMI panelu), programovatelného automatu a modelu automatické pračky. [10]

V tabulce Tab. 3 . jsou přiřazeny symbolické proměnné ke konkrétním adresám programovatelného automatu.

8.3 Zadání příkladu 1:

Vytvořte projekt s názvem PRACKA1 pro řízení pracího procesu. Uvažujte pouze zjednodušenou variantu a to bez předeprání a nastavení teploty od obsluhy. Upřesnění režimů pračky:

- Pračku zapneme tlačítkem Tl_start, pračka pere pouze v případě, že je zapnuto toto tlačítko.
- Voda musí být napuštěna na hodnotu 100%.
- Poté započne ohřev vody a to do 60°C.
- Praní probíhá 25s, kdy každých 5s se střídá otáčení bubnu – vlevo a vpravo.
- Po dokončení praní se začne vypouštět voda.
- Při vypouštění vody dochází k odstředování vody a to po dobu 15s.



Obr. 21. Vývojový diagram příkladu 1.

Nejdříve je nutné si k příslušným názvům proměnných přiřadit adresu vstupu, či výstupu a datový typ. Tuto akci provedeme v nabídce PLC Tags. Nabídku nalezneme v levém panelu v prostředí TIA Portal. Rozevřeme si položku Default tag table. Do tabulky, která se nyní zobrazila v prostřední části prostředí, napíšeme proměnné, datové typy a adresy, jako je uvedeno v tabulce Tab. 3.

Název proměnné	Datový typ	Adresa
Tl_Start	Bool	%I1.0
Buben ->	Bool	%Q0.0
Buben <-	Bool	%Q0.1
Otacky	Bool	%Q0.2
Topeni	Bool	%Q0.4
Voda	Bool	%Q0.5
Cerpadlo	Bool	%Q0.6
Hladina_50	Bool	%I0.2
Hladina_100	Bool	%I0.3
Teplota_90	Bool	%I0.4
Teplota_60	Bool	%I0.5
Teplota_40	Bool	%I0.6
Teplota_30	Bool	%I0.7
Casovac_vystup	Bool	%M0.0
Casovac_vstup	Bool	%M0.1
Cas	Time	%MD4
Cas_nastaveni	Time	%MD8

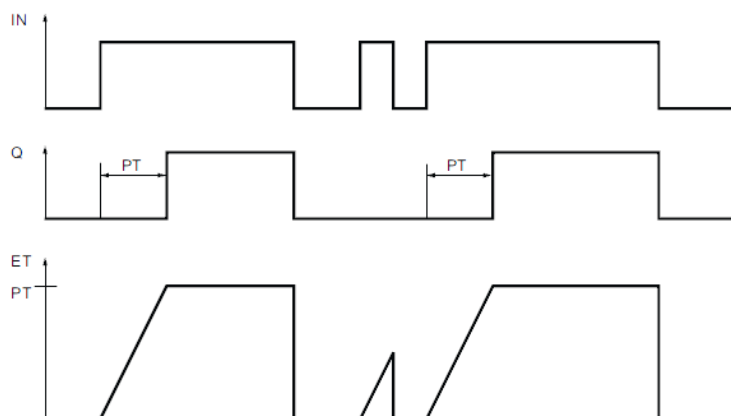
Tab. 3. Tabulka přiřazení názvů proměnných k adresám vstupů/výstupů.

8.3.1 Časovač

Je nezbytně nutné, osvojit si práci s časovačem. V příkladech v této bakalářské práci budu používat časovač typu TON – zpožděné zapnutí.

Nejdříve se musí vytvořit databáze, která bude sloužit pro časovač. V levém panelu prostředí TIA Portal klikneme na položku Add new block. V novém okně zvolíme Data Block, v položce Type vybereme ze seznamu typ IEC_Timer. Nový blok přejmenujeme na název Timer. Číslování bloku necháme automatické. Pro uložení stačí kliknout na tlačítko OK.

Časovač TON má tento časový průběh:



Obr. 22. Časový průběh časovače typu TON. [11]

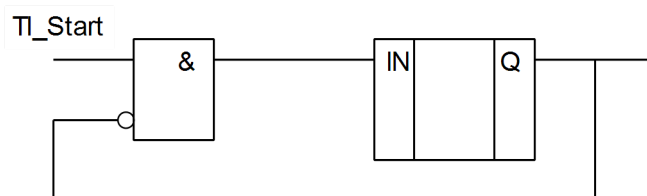
V prostředí TIA Portal se časovač TON zapisuje v jazyce SCL obdobným způsobem, jak je uvedeno ve zdrojovém kódu Zdr. kód 1. Vstupní proměnná IN je typu Bool. Jestliže má tato proměnná hodnotu logické jedničky, tak časovač začne počítat. Hodnotou PT nastavujeme hodnotu, do které má časovač napočítat. Tato hodnota je typu Time a zadáváme ji v milisekundách. Další parametr časovače je proměnná ET, v této proměnné se nachází aktuální čas, který právě v časovači je napočítaný. Až časovač napočítá do času, který je uvedený v proměnné PT, tak na výstup Q je přivedena logická 1. Pokud chceme časovač nastavit do původního stavu (resetovat), je nutné resetovat vstup IN. Tento reset provedeme jednoduše podle kódu Zdr. kód 2., příp. blokového schématu na obrázku Obr. 23.

```
"DB".TON(IN:=__in__, PT:=__val__, ET=>__out__, Q=>__out__);
```

Zdr. kód 1. Časovač v jazyce SCL.

```
"Casovac_vstup" := (NOT "Casovac_vystup") AND "Tl_Start";
```

Zdr. kód 2. Reset časovače.



Obr. 23. Blokové schéma resetu časovače.

8.3.2 Vypracované řešení příkladu 1

Program si rozdělíme na dvě funkce. První funkce bude hlavní, bude se spouštět vždy a její název bude "Main()". Do této funkce napíšeme hlavní podmínky, které nám budou zajišťovat, aby se spouštěl ten program, který právě chceme. Je to z toho důvodu, že později budeme tento úkol rozšiřovat. Tím si tedy ušetříme práci. Samotný program praní napíšeme do funkce s názvem "Rezim_prani()". Novou funkci vytvoříme obdobně jako v kapitole 8.3.1, avšak v nabídce vybereme možnost Function. Poté zadáme název nové funkce, a co je velmi důležité, tak to je výběr typu jazyka. Ve všech případech uvedených v této bakalářské práci budeme vybírat možnost SCL, jelikož všechny projekty budou psány v jazyce SCL. Dále je nutné ošetřit stav, kdy není zadán žádný čas, po jakou dobu se má otáčet buben na jednu nebo druhou stranu. Ve 3. příkladu půjde tato hodnota měnit pomocí HMI panelu.

```
IF NOT "Tl_Start" THEN
    //Pokud není zapnuté tlačítko Start, tak vypneme časovač, pokud běží
    "Casovac_vstup" := FALSE;
ELSE
    //Pokud není nastaven čas praní, tak zde si jej nastavíme, ve třetím úkolu
    //půjde nastavovat pomocí HMI panelu
    IF "Cas_nastaveni" = T#0S THEN
        "Cas_nastaveni" := T#5S;
    END_IF;
END_IF;
//Zapnutí časovače
IF ("Hladina_100" AND "Teplota_60") OR "Casovac_vstup" THEN
    "Timer".TON(IN := "Casovac_vstup",
        PT := (("Cas_nastaveni"*5)+T#20S),
        ET => "Cas",
        Q => "Casovac_vystup");
    //Reset časovače
    "Casovac_vstup" := NOT "Casovac_vystup";
END_IF;
IF "Tl_Start" THEN
```

```
//Pokud je zapnuté tlačítko Start, tak provádíme funkci Režim praní
"Rezim_prani"();
END_IF;
```

Zdr. kód 3. Obsah hlavní funkce "Main()".

```
//Napouštění vody
IF NOT "Hladina_100" AND NOT "Casovac_vstup" AND NOT "Casovac_vystup" THEN
  "Voda" := TRUE;
ELSE
  "Voda" := FALSE;
END_IF;

//Ohřívání vody
IF "Hladina_100" AND NOT "teplota_60" AND NOT "Casovac_vstup" THEN
  "Topeni" := TRUE;
  "Buben ->" := TRUE;
END_IF;

//Jelikož modulo je možné použít jen pro dat. typ integer, tak ho nemůžu využít.
IF "Casovac_vstup" THEN
  //Vypnutí ohřívání vody
  "Topeni" := FALSE;
  IF ("Cas" > T#0S) AND ("Cas" <= "Cas_nastaveni") OR ("Cas" >=
("Cas_nastaveni"*2)) AND ("Cas" <= ("Cas_nastaveni"*3)) OR ("Cas" >=
("Cas_nastaveni"*4)) AND ("Cas" <= ("Cas_nastaveni"*5)) THEN
    //Otáčení pracího bubnu vpravo
    "Buben ->" := TRUE;
    "Buben <-" := FALSE;
  ELSE
    //Otáčení pracího bubnu vlevo
    "Buben ->" := FALSE;
    "Buben <-" := TRUE;
  END_IF;
END_IF;

//Vypouštění + odstředování
IF "Cas" >= ("Cas_nastaveni"*5) AND "Cas" <= ("Cas_nastaveni"*5)+T#20S AND
"Casovac_vstup" THEN
  //Odstředování
  "Buben ->" := TRUE;
  "Buben <-" := FALSE;
  "Otacky" := TRUE;
  IF "Cas" >= ("Cas_nastaveni"*5) AND "Cas" <= ("Cas_nastaveni"*5)+T#15S THEN
    //Vypouštění vody z prací vany
    "Cerpadlo" := TRUE;
  ELSE
    //Zastavení odstředování
    "Otacky" := FALSE;
    "Buben ->" := FALSE;
    "Buben <-" := FALSE;
  END_IF;
END_IF;

//Ukončení činností automatické pračky
IF "Cas" >= ("Cas_nastaveni"*5)+T#20S AND "Casovac_vystup" THEN
  "Buben ->" := FALSE;
  "Buben <-" := FALSE;
  "Otacky" := FALSE;
  "Cerpadlo" := FALSE;
END_IF;
```

Zdr. kód 4. Obsah funkce "Rezim_prani()".

8.4 Zadání příkladu 2:

Vytvořte projekt s názvem PRACKA2 pro řízení pracího procesu. Rozšiřte zadání příkladu 1 tak, že přidáte režim předeprání. Režim praní, či režim předeprání zapnete přes HMI panel. Upřesnění režimů pračky:

- Napustit vodu na 50% prací vany.
- Ohřát vodu na teplotu 40°C, při ohřevu otáčet bubnem doprava.
- Otáčet bubnem po dobu 25s s pauzou mezi reverzacemi otáčení bubnu.
- Po dokončení předeprání pokračovat v režimu praní z prvního zadání.
- Upozorňovat uživatele textem na panelu, který režim právě probíhá.

8.4.1 Vypracované řešení příkladu 2

Nejprve je nutné přidat do tabulky symbolických proměnných proměnné, které budeme využívat v tomto příkladu.

Název proměnné	Datový typ	Adresa
Cas_pauza	Time	%MD12
Rezim	Int	%MW16

Tab. 4. Dodatek tabulky symbolických proměnných.

Vypracování je obdobné jako u předchozího příkladu. Rozdíl je pouze v tom, že do pracího procesu zařadíme pauzu mezi reverzací otáčení bubnu. Dále je nutné, abychom upravili hlavní funkci "Main()"; přidáním podmínky, podle které určíme prací proces. Předeprání bude označeno hodnotou 1 v proměnné Rezim, proces praní bude označen hodnotou 2.

```

IF NOT "Tl_Start" THEN
    //Pokud není zapnuté tlačítko Start, tak vypneme časovač, pokud běží
    "Casovac_vstup" := FALSE;
ELSE
    //Pokud není nastaven čas praní a čas pauzy, tak zde si jej nastavíme, ve
    //třetím úkolu půjde nastavovat pomocí HMI panelu
    IF "Cas_nastaveni" = T#0S THEN
        "Cas_nastaveni" := T#5S;
    END_IF;
    IF "Cas_pauza" = T#0S THEN
        "Cas_pauza" := T#1S;
    END_IF;
END_IF;

//Zapnutí a běh časovače
IF ("Hladina_100" AND "Teplota_60") OR ("Hladina_50" AND "Teplota_40" AND
("Rezim=1") OR "Casovac_vstup" THEN
    "Timer".TON(IN := "Casovac_vstup",
        PT := (("Cas_nastaveni"*5)+T#20S),
        ET => "Cas",
        Q => "Casovac_vystup");
    //Reset časovače
    "Casovac_vstup" := NOT "Casovac_vystup";
END_IF;

//Vybere a spustí takový program, který si uživatel zvolil
CASE "rezim" OF
1:
    "Rezim_predeprani"();
;
2:
    "Rezim_prani"();
;
ELSE

```

```

        "Casovac_vstup" := FALSE;
    ;
    END_CASE;
END_IF;

```

Zdr. kód 5. Obsah hlavní funkce "Main"();

```

//Napouštění vody na hodnotu 50% objemu prací vany
IF NOT "Hladina_50" AND NOT "Casovac_vstup" THEN
    "Voda" := TRUE;
ELSE
    "Voda" := FALSE;
END_IF;

//Ohřívání vody na teplotu 40°C při 50% objemu vody v prací vaně
IF "Hladina_50" AND NOT "Teplota_40" AND NOT "Casovac_vstup" THEN
    "Topeni" := TRUE;
    "Buben ->" := TRUE;
END_IF;

//Praní probíhá v tom případě, že běží časovač
IF "Casovac_vstup" THEN
    "Topeni" := FALSE;
    "Buben ->" := FALSE;
    //Pokud je podmínka splněna, tak dochází k otáčení bubnu vpravo, pokud podmínka
    neplatí, tak dochází k otáčení bubnu vlevo
    IF (("Cas" > T#0S) AND ("Cas" <= "Cas_nastaveni")) OR (("Cas" >=
    ("Cas_nastaveni"*2)) AND ("Cas" <= ("Cas_nastaveni"*3))) THEN
        //Zastavení otáčení na čas, který je nastaven v proměnné Cas_pauza
        IF (("Cas" >= "Cas_nastaveni"- "Cas_pauza") AND ("Cas" <= "Cas_nastaveni")) OR
        (("mereny_cas" >= (("Cas_nastaveni"*3)- "Cas_pauza")) AND ("Cas" <=
        ("Cas_nastaveni"*3))) THEN
            "Buben ->" := FALSE;
        ELSE
            "Buben ->" := TRUE;
        END_IF;
        "Buben <-" := FALSE;
    ELSE
        //Zastavení otáčení na čas, který je nastaven v proměnné Cas_pauza
        IF(("Cas" >= ("Cas_nastaveni"*2)- "Cas_pauza") AND ("Cas" <=
        ("Cas_nastaveni"*2))) THEN
            "Buben <-" := FALSE;
        ELSE
            "Buben <-" := TRUE;
        END_IF;
        "Buben ->" := FALSE;
    END_IF;
END_IF;

//Ukončení režimu předepírání a nastavení proměnné Rezim na hodnotu 2, což znamená,
že se započne režim praní, stejně, jako je napsáno v zadání tohoto příkladu
IF "Cas" = ("Cas_nastaveni"*3) AND "Casovac_vstup" AND ("Rezim" = 1) THEN
    "Buben ->" := FALSE;
    "Buben <-" := FALSE;
    "Rezim" := 2;
    "Casovac_vstup" := FALSE;
END_IF;

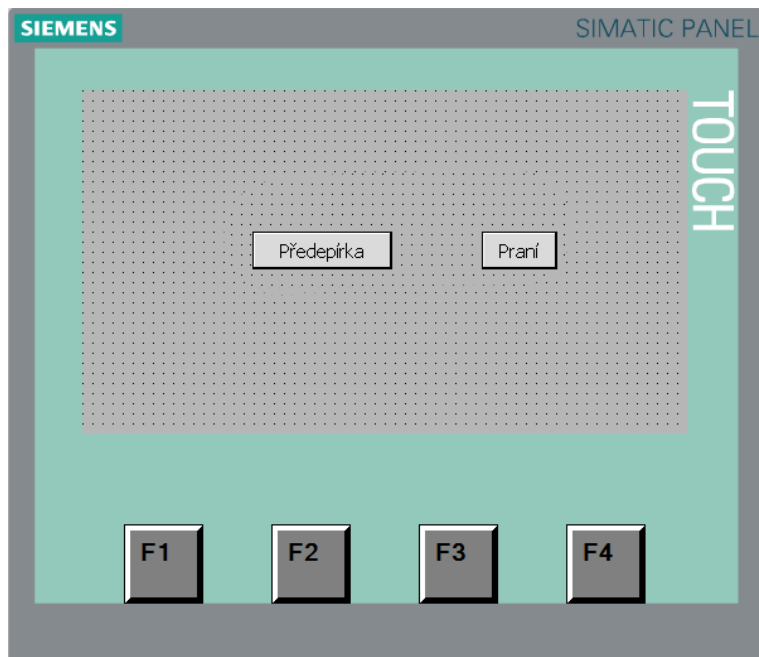
```

Zdr. kód 6. Obsah funkce "Rezim_predeprani"();

V dalším kroku je potřeba vytvořit rozšíření s HMI panelem. Do projektu přidáme HMI panel, jak je popsáno v kapitole 7.5.

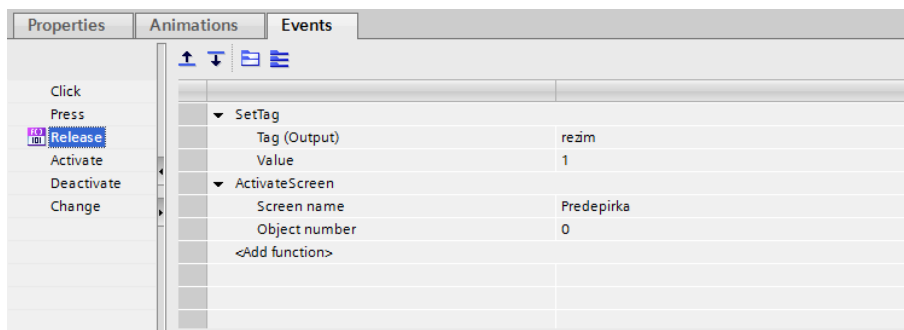
Nejdříve vytvoříme novou obrazovku v HMI panelu a to kliknutím na položku Add new screen v levé části obrazovky. Novou obrazovku pojmenujeme názvem Start. Dvojklikem na tuto položku si otevřeme obrazovku a přidáme do ní 2 tlačítka, přes která budeme moci zapínat daný prací režim (předepírání, praní). Tlačítka do obrazovky přidáme tak, že jednoduše z pravého panelu na

obrazovce, konkrétně z nabídky Elements, přetáhneme ikonku, která vypadá jako tlačítko (šedý obdélník). Dvojklikem na text v tlačítku jej přejmenujeme na text „Předepírka“ a „Praní“.

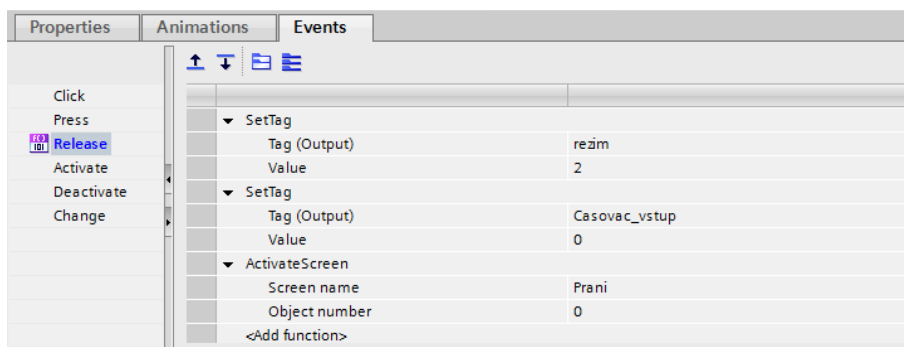


Obr. 24. Rozložení tlačítek na panelu.

Na tlačítko Předepírka nastavíme akce, které jsou uvedeny na obrázku Obr. 25. Akce, které nastavíme u tlačítka Praní, jsou zobrazeny na obrázku Obr. 26.

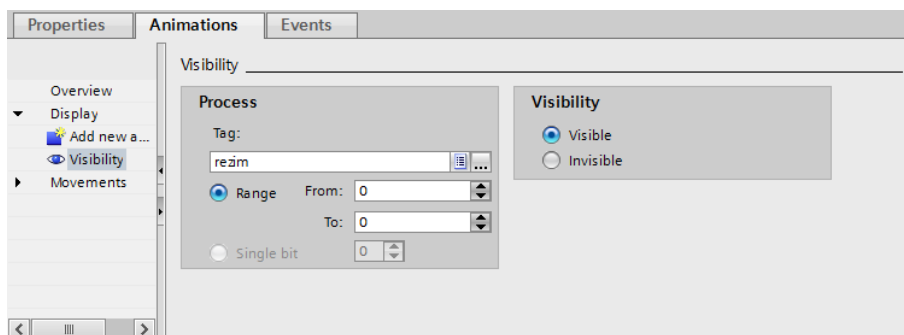


Obr. 25. Akce u tlačítka Předepírka.

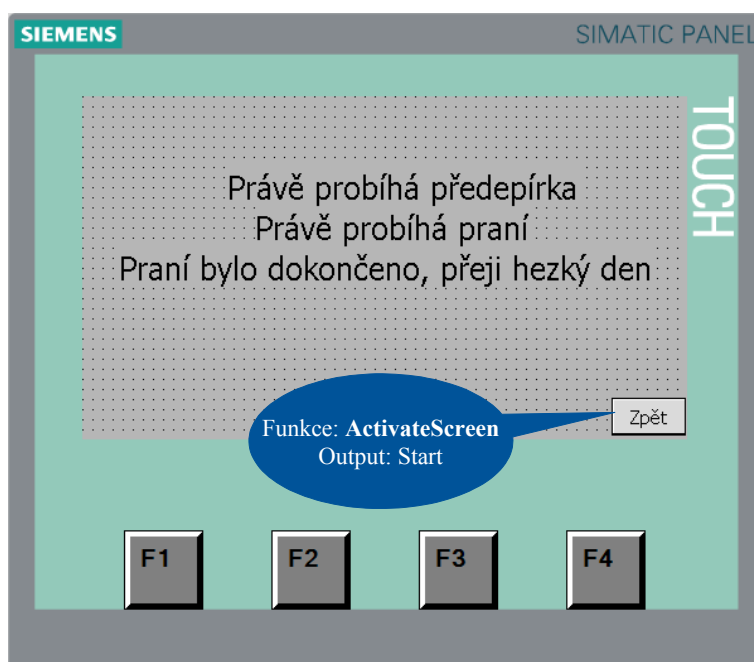


Obr. 26. Akce u tlačítka Praní.

Dále si vytvoříme text, který bude uživatele přes HMI panel upozorňovat, který režim právě probíhá. Vytvoříme si nové obrazovky, které pojmenujeme Předepírka a Praní. Do těchto obrazovek si vložíme textová pole, která jsou znázorněna v obrázku Obr. 28. V jednom textovém poli bude napsaný text „Právě probíhá předepírka“, ve druhém „Právě probíhá praní“ a ve třetím „Praní bylo dokončeno, přeji hezký den“. Zobrazení těchto textových polí nastavíme pomocí viditelnosti v závislosti na hodnotě, kterou obsahuje proměnná Rezim, obdobně, jako je uvedeno na obrázku Obr. 27. V poslední obrazovce přibude ještě tlačítko s názvem Zpět, které uživatele vrátí zpět k výběru režimů. U tlačítka Zpět přidáme funkci ActivateScreen a hodnotu Screen name nastavíme na hodnotu Start – po kliknutí na toto tlačítko se nám zobrazí úvodní obrazovka Start.



Obr. 27. Nastavení viditelnosti u textového pole „Praní bylo dokončeno...“.



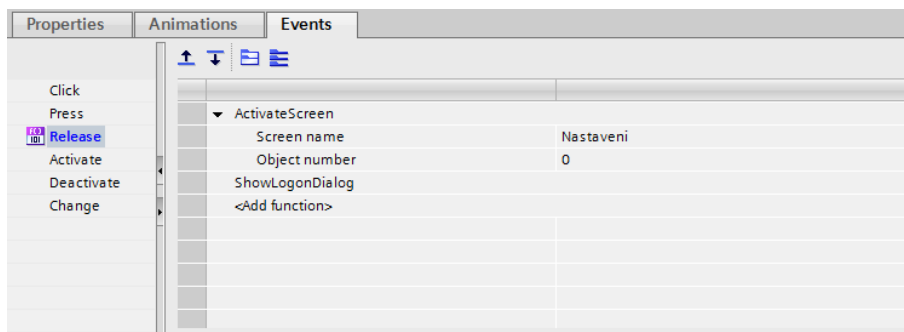
Obr. 28. Textová pole v panelu bez omezení viditelnosti.

8.5 Zadání příkladu 3

Rozšiřte příklad 2 o servisní mód. Na úvodní obrazovce přibude tlačítko Servis. Po kliknutí na toto tlačítko se zobrazí přihlašovací obrazovka, aby servisní parametry nemohl upravovat uživatel. Po zadání přihlašovacího jména a hesla se technik přihlásí a bude moci upravit teplotu ohřevu vody u režimu praní, čas otáčení bubnu a čas pauzy při reverzaci otáčení bubnu.

8.5.1 Vypracované řešení příkladu 3

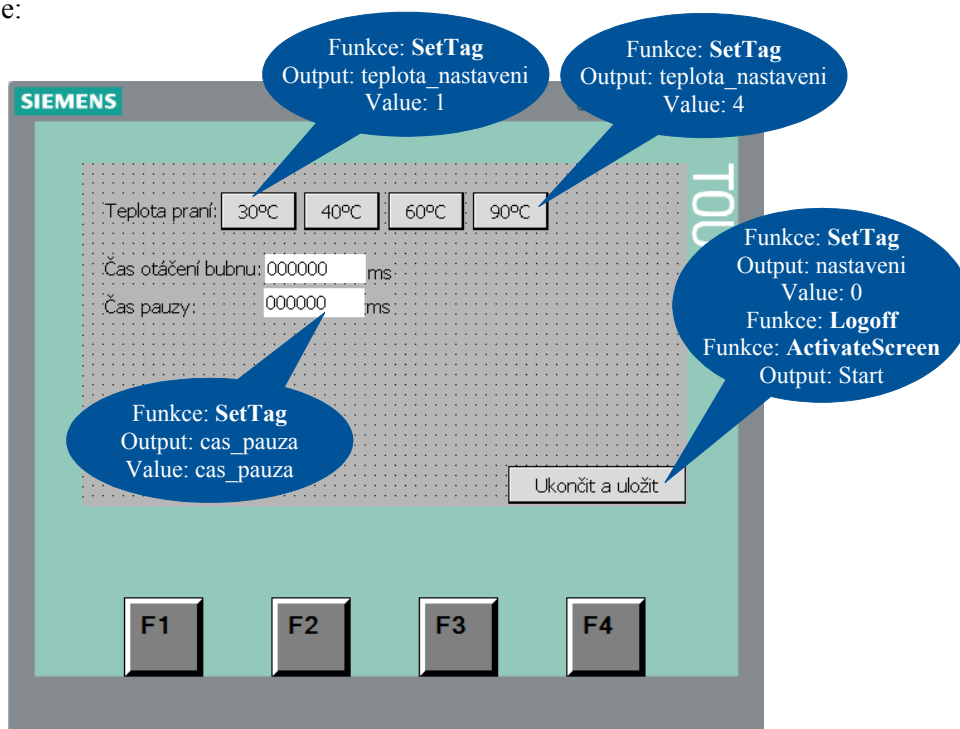
Nejprve si do úvodní obrazovky přidáme tlačítko a pojmenujeme je názvem Servis. Tlačítku nastavíme akce, které jsou znázorněny na obrázku Obr. 29. Dále je nutné přidat do programovatelného automatu uživatele, kteří se mohou přihlásit. V nabídce HMI panelu klikneme na položku User administration. V okně Users přidáme nového uživatele kliknutím na text <Add new>, pojmenujeme jej a zadáme tomuto uživateli heslo. Dále ještě uživateli přiřadíme skupinu, klikneme ve spodní části na položku Administrator group.



Obr. 29. Akce u tlačítka Servis.

Pokud servisní technik zadá správné údaje, bude přesměrován na obrazovku, kterou pojmenujeme například Nastaveni. Pokud technik napíše nesprávné údaje, nedojde k žádnému přesměrování a technik zůstane na úvodní stránce. Tím je zajištěna bezpečnost, aby nepoučená osoba nemohla nastavovat servisní parametry.

Vytvoříme si tedy novou obrazovku s názvem Nastaveni. Do této obrazovky vložíme 5 tlačítek (4 pro teplotu a 1 pro uložení hodnot), 2 políčka pro zápis hodnot časů a 5 textových polí. Rozvržení obrazovky by mělo vypadat podobně, jako je uvedeno na obrázku Obr. 30. Tlačítkům nastavíme tyto akce:



Obr. 30. Servisní obrazovka „Nastaveni“.

Ještě je nutné upravit zdrojový kód hlavní funkce "Main()". A to z důvodu, že nyní máme nastavenou práci teplotu vody v práci vaně na 60°C. Vytvoříme si novou proměnnou s názvem Teplota_ok, do které budeme nastavovat teplotu praní.

Název proměnné	Datový typ	Adresa
Teplota_ok	Bool	%M1.2

Tab. 5. Úprava tabulky symbolických proměnných.

Úpravu ve funkci "Main()"; provedeme následovně. Nalezneme řádek, který je napsán ve zdrojovém kódu Zdr. kód 7. a nahradíme jej kódem, který je napsán ve zdrojovém kódu Zdr. kód 8.

```
IF ("Hladina_100" AND "Teplota_60") OR ("Hladina_50" AND "teplota_40" AND
("Rezim"=1)) OR "Casovac_vstup" THEN
```

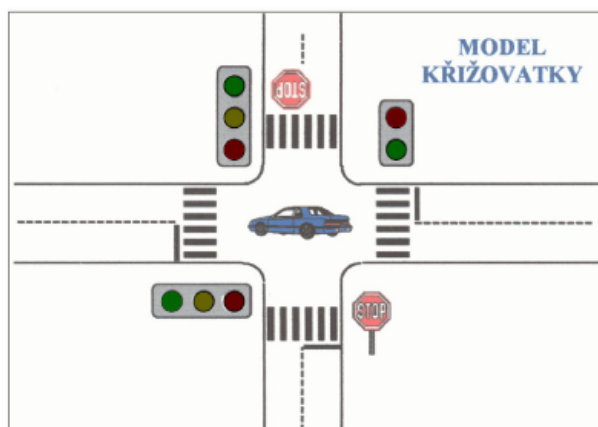
Zdr. kód 7. Stávající obsah hlavní funkce "Main()";.

```
IF ("Hladina_100" AND "Teplota_ok") OR ("Hladina_50" AND "teplota_40" AND
("Rezim"=1)) OR "Casovac_vstup" THEN
```

Zdr. kód 8. Upravený obsah hlavní funkce "Main()";.

Po těchto úpravách bude možné měnit teplotu vody při režimu praní, nastavovat čas praní a čas pauzy. Odhlášení technika je řešeno příslušnou akcí u tlačítka Uložit, jedná se o funkci Logoff.

9 MODEL DOPRAVNÍ KŘÍŽOVATKY

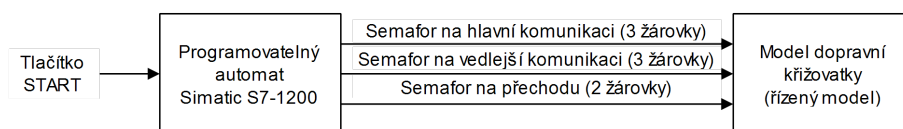


Obr. 31. Znárodnění modelu dopravní křižovatky.

9.1 Použité pomůcky

- Programovatelný automat Simatic S7-1200 CPU 1214C,
- HMI panel KTP400 Basic color,
- model dopravní křižovatky (8 digitálních vstupů, 0 digitálních výstupů),
- modul tlačítek.

9.2 Schéma zapojení



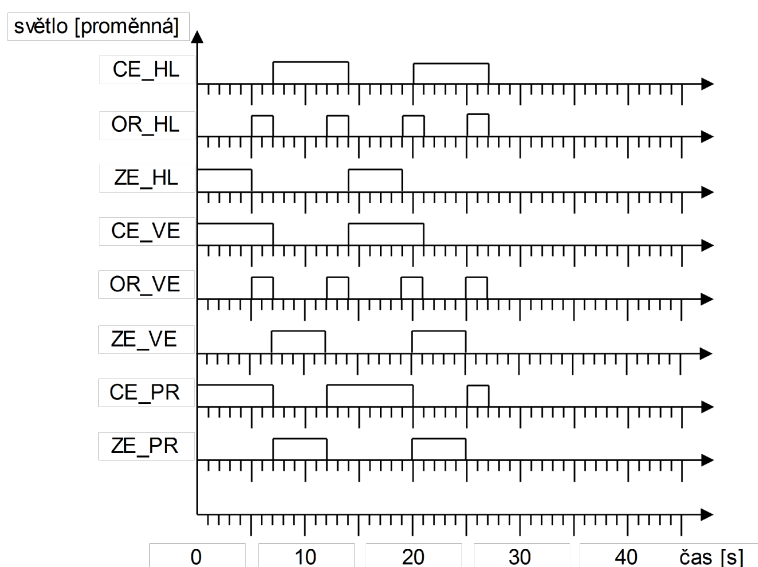
Obr. 32. Schéma zapojení modelu dopravní křižovatky.

Na obrázku Obr. 32. je zobrazeno schéma propojení modulu spínačů (příp. HMI panelu a/nebo web serveru), programovatelného automatu a modelu dopravní křižovatky. [12]

V tabulce Tab. 6 . jsou přiřazeny symbolické proměnné ke konkrétním adresám PLC.

9.3 Zadání příkladu 1:

Vypracujte projekt s názvem Krizovatka-denni-rezim pro řízení dopravní křižovatky dle zadaného časového diagramu.



Obr. 33. Časový diagram funkce "Denni_rezim()".

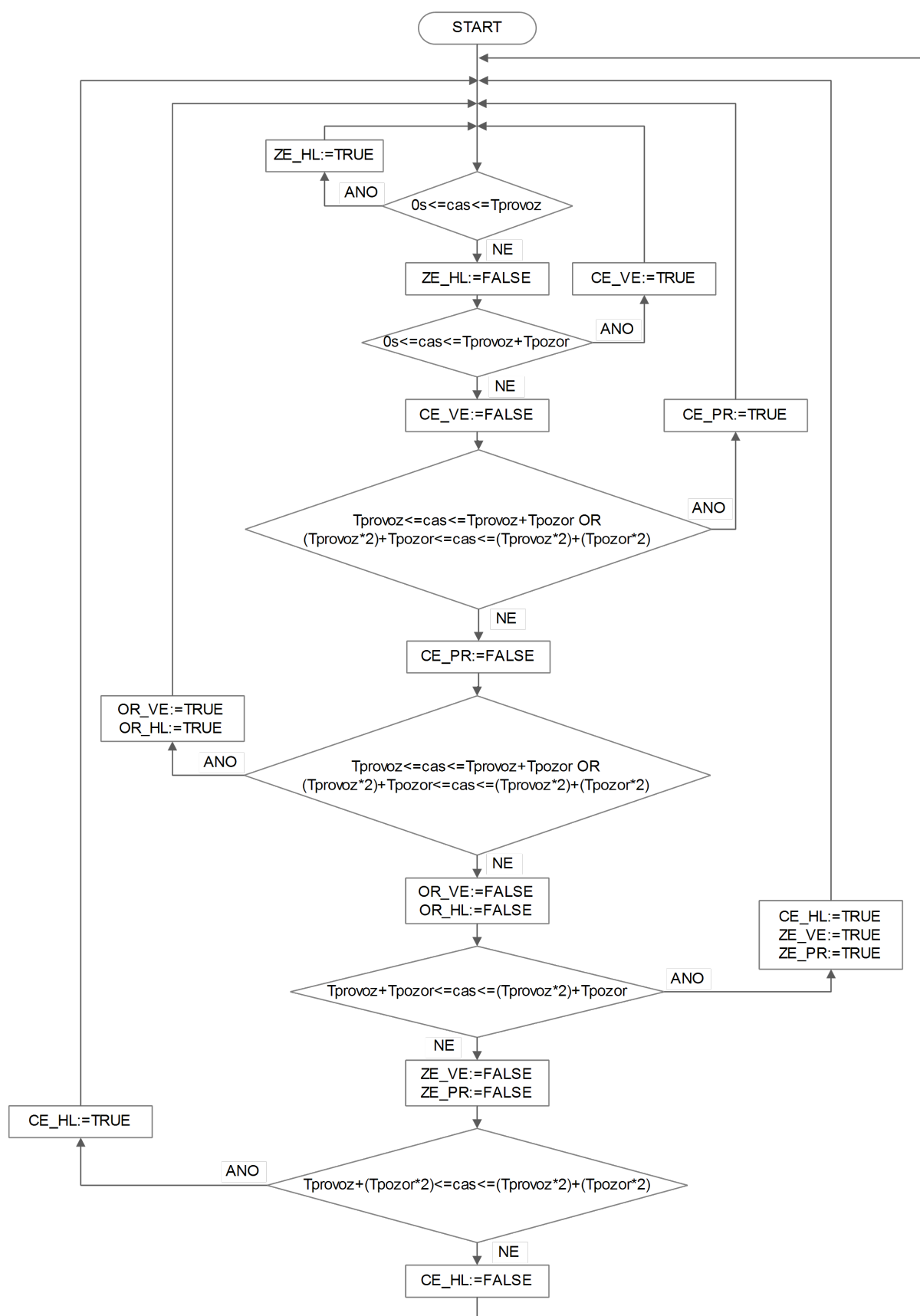
Obdobně jako v předchozích příkladech je nutné vyplnit tabulku názvů proměnných. Čas provozu na křižovatce a čas režimu pozor (oranžová světla) uložíme do paměti programovatelného automatu, abychom tyto hodnoty mohli lépe upravovat a v dalších rozšířeních nastavovat podle obsluhy dopravní křižovatky.

Název proměnné	Datový typ	Adresa
Tl_Start	Bool	%I1.0
CE_HL	Bool	%Q0.0
OR_HL	Bool	%Q0.1
ZE_HL	Bool	%Q0.2
CE_VE	Bool	%Q0.3
OR_VE	Bool	%Q0.4
ZE_VE	Bool	%Q0.5
CE_PR	Bool	%Q0.6
ZE_PR	Bool	%Q0.7
Casovac_vstup	Bool	%M0.0
Casovac_vystup	Bool	%M0.1
Cas	Time	%MD4
Tprovoz	Time	%MD8
Tpozor	Time	%MD12

Tab. 6. Tabulka přiřazení názvů proměnných k adresám vstupů/výstupů.

V těchto zadáních u modelu dopravní křižovatky opět využíváme časovače. Proto doporučuji zopakovat si práci s časovačem, která je popsána v kapitole 8.3.1.

Abychom si tento první projekt připravili i pro rozšíření, tak následující vypracování umístíme do funkce, kterou poté pouze vložíme do hlavní funkce Main. Vytvoříme si novou funkci s názvem "Denni_rezim"(). Klikneme na položku Add new block, vybereme položku Function, pojmenujeme a vytvoříme novou funkci. V ní vytvoříme samotný běh programu, který je popsán v kapitole 9.3.1. Ještě si musíme vytvořit funkci, která bude resetovat veškeré proměnné a nastavovat časové konstanty na inicializační hodnoty v tom případě, že nebudou zadány od uživatele například z webového rozhraní, nebo z HMI panelu. Tato funkce se bude jmenovat "Reset"(). Pro lepší názornost denního režimu je možné čerpat i z vývojového diagramu na obrázku Obr. 34.



Obr. 34. Vývojový diagram funkce "Denni_rezim".

9.3.1 Vypracované řešení příkladu 1

```
//Pokud nejsou nastaveny časy provozu, tak nastavíme na defaultní
IF "Tprovoz"=T#0S AND "Tpozor"=T#0S THEN
    "Tprovoz" := T#5S;           //Čas provozu nastavený na 5s
    "Tpozor" := T#1S;           //Doba, po kterou svítí oranžové světlo
END_IF;

//Zhasnutí všech světel na semaforech
"CE_HL" := FALSE;               //Červená na hlavní komunikaci
"OR_HL" := FALSE;               //Oranžová na hlavní komunikaci
"ZE_HL" := FALSE;               //Zelená na hlavní komunikaci
"CE_VE" := FALSE;               //Červená na vedlejší komunikaci
"OR_VE" := FALSE;               //Oranžová na vedlejší komunikaci
"ZE_VE" := FALSE;               //Zelená na vedlejší komunikaci
"CE_PR" := FALSE;               //Červená na přechodu
"ZE_PR" := FALSE;               //Zelená na přechodu
```

Zdr. kód 9. Obsah funkce "Reset".

```
//Pokud je vypnuté tlačítko Start, tak zastavíme časovač a zhasneme všechna světla
IF NOT "Tl_Start" THEN
    "Reset"();
    "Casovac_vstup" := FALSE;
END_IF;

//Pokud nejsou nastaveny hodnoty časů, tak ve funkci "Reset"; nastavíme
IF "Tprovoz" = T#0S OR "Tpozor" = T#0S THEN
    "Reset"();
END_IF;

//Pokud je zapnuto tlačítko Start, tak je aktivní denní režim
IF "Tl_Start" THEN
    "Denni_rezim"();
END_IF;
```

Zdr. kód 10. Obsah hlavní funkce Main.

```
//Časovač
"Timer".TON(IN := "Casovac_vstup", PT := ("Tprovoz"*2)+("Tpozor"*2), ET => "Cas", Q
=> "Casovac_vystup");

//Reset časovače
"Casovac_vstup" := NOT "Casovac_vystup";

//Provoz na hlavní silnici
IF ("Cas" >= T#0S) AND ("Cas" <= "Tprovoz") THEN
    "ZE_HL" := TRUE;
ELSE
    "ZE_HL" := FALSE;
END_IF;

//Červená svítí společně s oranžovou
IF ("Cas" >= T#0S) AND ("Cas" <= "Tprovoz"+"Tpozor") THEN
    "CE_VE" := TRUE;
ELSE
    "CE_VE" := FALSE;
END_IF;

//Červená pro chodce svítí jen v určité časy
IF (("Cas" >= T#0S) AND ("Cas" <= "Tprovoz"+"Tpozor")) OR (("Cas" >=
("Tprovoz"*2)+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor"*2)) THEN
    "CE_PR" := TRUE;
ELSE
    "CE_PR" := FALSE;
END_IF;

//Nastavení oranžové barvy na obou semaforech
```

```

IF (("Cas" >= "Tprovoz") AND ("Cas" <= "Tprovoz"+"Tpozor")) OR (("Cas" >=
("Tprovoz"*2)+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor"*2)) THEN
    "OR_HL" := TRUE;
    "OR_VE" := TRUE;
ELSE
    "OR_HL" := FALSE;
    "OR_VE" := FALSE;
END_IF;

//Provoz po vedlejší komunikaci a zelená pro chodce
IF ("Cas" >= "Tprovoz"+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor") THEN
    "CE_HL" := TRUE;
    "ZE_VE" := TRUE;
    "ZE_PR" := TRUE;
ELSE
    //Nastavení červené na hlavní komunikaci
    IF (("Cas" >= ("Tprovoz"*2)+"Tpozor")) AND ("Cas" <=
(("Tprovoz"*2)+"Tpozor"*2)) THEN
        "CE_HL" := TRUE;
    ELSE
        "CE_HL" := FALSE;
    END_IF;
    "ZE_VE" := FALSE;
    "ZE_PR" := FALSE;
END_IF;

```

Zdr. kód 11. Vypracované zadání funkce "Denni_rezim"().

Jelikož jsou barvy na semaforech řízeny pouze dvouhodnotovými hodnotami, tak je možno toto řešení značně zjednodušit. Na jedné straně zmenšíme velikost kódu, ale na druhé straně mírně znehledníme zdrojový kód. Ukázka kódu je uvedena ve zdrojovém kódu Zdr. kód 12.

```

//Časovač
"Timer".TON(IN := "Casovac_vstup", PT := ("Tprovoz"*2)+("Tpozor"*2), ET => "Cas", Q
=> "Casovac_vystup");

//Reset časovače
"Casovac_vstup" := (NOT "Casovac_vystup");

//Provoz na hlavní silnici
"ZE_HL" := ("Cas" >= T#0S) AND ("Cas" <= "Tprovoz");

//Červená svítí společně s oranžovou
"CE_VE" := ("Cas" >= T#0S) AND ("Cas" <= "Tprovoz"+"Tpozor");

//Červená pro chodce svítí jen v určité časy
"CE_PR" := (("Cas" >= T#0S) AND ("Cas" <= "Tprovoz"+"Tpozor")) OR (("Cas" >=
("Tprovoz"*2)+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor"*2));

//Nastavení oranžové barvy na všech semaforech
"OR_HL" := (("Cas" >= "Tprovoz") AND ("Cas" <= "Tprovoz"+"Tpozor")) OR (("Cas" >=
("Tprovoz"*2)+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor"*2));
"OR_VE" := "Tl_Start" AND (("Cas" >= "Tprovoz") AND ("Cas" <= "Tprovoz"+"Tpozor"))
OR (("Cas" >= ("Tprovoz"*2)+"Tpozor") AND ("Cas" <= ("Tprovoz"*2)+"Tpozor"*2));

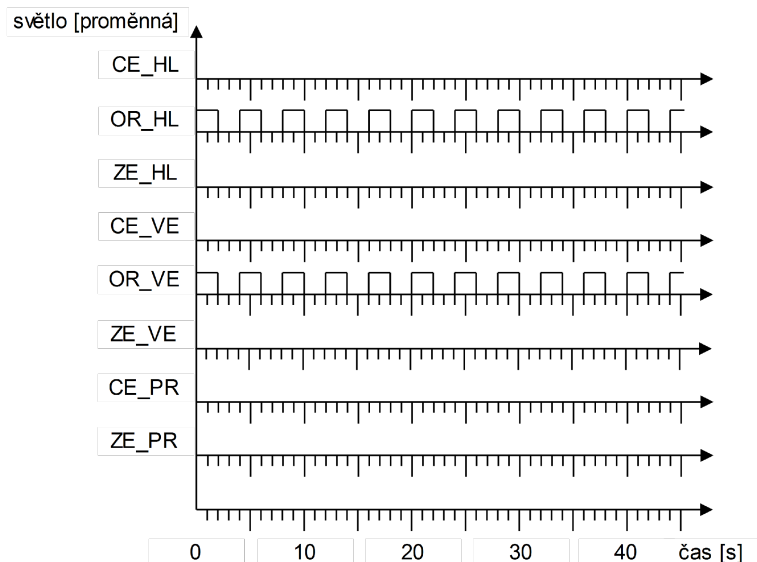
//Provoz na vedlejší komunikaci a zelená na semaforu pro chodce
"CE_HL" := ("Cas" >= "Tprovoz"+"Tpozor") AND ("Cas" <= ("Cas"-"Tpozor") <=
("Tprovoz"*2)+"Tpozor");
"ZE_VE" := "Tl_Start" AND ("Cas" >= "Tprovoz"+"Tpozor") AND ("Cas" <=
("Tprovoz"*2)+"Tpozor");
"ZE_PR" := "Tl_Start" AND ("Cas" >= "Tprovoz"+"Tpozor") AND ("Cas" <=
("Tprovoz"*2)+"Tpozor");

```

Zdr. kód 12. Úspornější kód funkce "Denni_rezim"().

9.4 Zadání příkladu 2:

Vypracujte projekt s názvem Křižovatka-nocni-rezim pro řízení dopravní křižovatky dle zadaného časového diagramu. Noční režim nastavte na tlačítko Tl_noc. Pokud nebude sepnuto tlačítko Tl_noc, tak křižovatka musí fungovat v denním režimu, který byl vytvořen v minulém projektu.

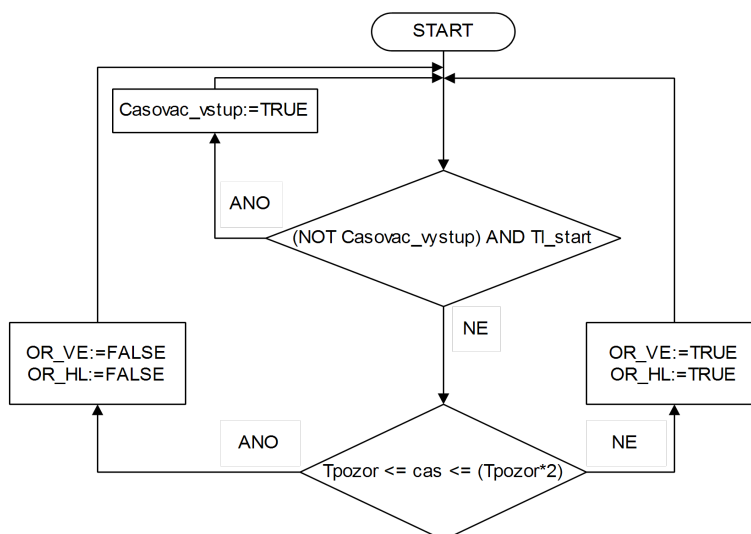


Obr. 35. Časový diagram funkce "Nocni_rezim"();.

Při řešení tohoto úkolu vycházíme z předešlého příkladu. Pouze mírně upravíme hlavní funkci Main. Musíme zajistit, aby při sepnutí tlačítka Tl_noc přešla křižovatka do nočního režimu. Do tabulky symbolických proměnných přidejte následující hodnoty.

Název proměnné	Datový typ	Adresa
Tl_noc	Bool	%I1.1
Kontrola_pruchodu	Bool	%M0.2

Tab. 7. Úprava tabulky symbolických proměnných.



Obr. 36. Vývojový diagram funkce "Nocni_rezim"();.

9.4.1 Vypracované řešení příkladu 2

Z důvodu přepínání režimů (denní a noční) je nutné ošetřit časovač. A to proto, že pokud obsluha přepne režim z denního na noční, tak od této doby je stav semaforu v poloze, kdy svítí světla, která byla aktivní v době přepnutí. Tento stav je až do dosažení hodnoty PT. Proto je nutné při přepnutí režimu restartovat časovač, a to při jednom programovém cyklu. Tuto akci provedeme ve funkci "Reset_casovace"();

```
//Pokud proběhla změna z denního režimu na režim noční
IF NOT "Kontrola_pruchodu" AND "Tl_noc" AND "Tl_Start" THEN
    "Casovac_vstup" := FALSE;
    "Kontrola_pruchodu" := TRUE;
END_IF;

//Pokud proběhla změna z nočního režimu na režim denní
IF "Kontrola_pruchodu" AND NOT "Tl_noc" AND "Tl_Start" THEN
    "Casovac_vstup" := FALSE;
    "Kontrola_pruchodu" := FALSE;
END_IF;
```

Zdr. kód 13. Obsah funkce "Reset_casovace"();

```
//Časovač
"Timer".TON(IN := "Casovac_vstup", PT := "Tpozor"*2, ET => "Cas", Q =>
"Casovac_vystup");

//Reset časovače
"Casovac_vstup" := NOT "Casovac_vystup";

//Zhasneme veškerá světla
"Reset"();

//Rozsvítíme nebo zhasneme oranžová světla
IF ("Cas" >= "Tpozor") AND ("Cas" <= "Tpozor"*2) THEN
    "OR_HL" := FALSE;
    "OR_VE" := FALSE;
ELSE
    "OR_HL" := TRUE;
    "OR_VE" := TRUE;
END_IF;
```

Zdr. kód 14. Obsah funkce "Nocni_rezim"();

```
IF NOT "Tl_Start" THEN
    "Reset"();
    "Casovac_vstup" := FALSE;
END_IF;

//Pokud nejsou nastaveny hodnoty časů, tak ve funkci "Reset"(); nastavíme
IF "Tprovoz" = T#0S OR "Tpozor" = T#0S THEN
    "Reset"();
END_IF;

//Reset časovače v případě, že změníme režim
"Reset_casovace"();

IF "Tl_Start" AND NOT "Tl_noc" THEN
    "Denni_rezim"();
END_IF;
IF "Tl_Start" AND "Tl_noc" THEN
    "Nocni_rezim"();
END_IF;
```

Zdr. kód 15. Obsah hlavní funkce Main.

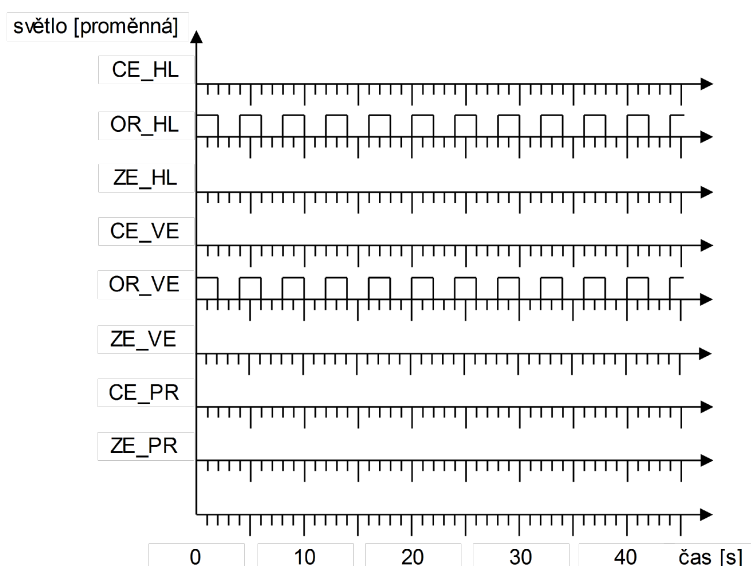
9.5 Zadání příkladu 3

Rozšiřte předchozí příklady, konkrétně denní a noční režim křižovatky, aby bylo možné nastavovat časy provozu na křižovatce pomocí HMI panelu a Web Serveru. U rozšíření s Web Serverem zobrazujte i stav, ve kterém se v dané chvíli dopravní křižovatka nachází.

Při vypracování opět vycházíme ze zadání z kapitol 9.3 a 9.4. Nejdříve musíme přidat proměnnou, pomocí které budeme ověřovat, zda právě dochází k nastavení časů provozu. Pokud se bude provádět nastavení časů, tak křižovatka přejde do speciálního režimu, který je podobný nočnímu režimu. Tento krok je z toho důvodu, abychom zajistili bezpečnost provozu na dopravní křižovatce. Nový režim napíšeme do funkce "Rezim_nastaveni()". Pokud tedy budeme provádět nastavení křižovatky, tak do proměnné Nastavení zapíšeme logickou 1.

Název proměnné	Datový typ	Adresa
Nastavení	Bool	%M0.3

Tab. 8. Úprava tabulky symbolických proměnných.



Obr. 37. Časový diagram funkce "Rezim_nastaveni()".

Zdrojový kód tohoto režimu je stejný, jako u nočního režimu. Rozdíl je pouze v tom, že časy v tomto režimu jsou zadány explicitně, a obsluha tyto časy nemůže měnit.

```
//Časovač
"Timer".TON(IN := "Casovac_vstup", PT := T#4S, ET => "mereny_cas", Q =>
"Casovac_vystup");

//Reset časovače
"Casovac_vstup" := NOT "Casovac_vystup";

"Reset"();

IF ("Cas" >= T#2S) AND ("Cas" <= T#4S) THEN
    "OR_HL" := FALSE;
    "OR_VE" := FALSE;
ELSE
    "OR_HL" := TRUE;
    "OR_VE" := TRUE;
END_IF;
```

Zdr. kód 16. Obsah funkce "Rezim_nastaveni()".

Abychom mohli křižovatku přepnout do režimu nastavení, tak musíme upravit hlavní funkci Main.

```
IF NOT "Tl_Start" THEN
    "Reset"();
    "Casovac_vstup" := FALSE;
END_IF;

//Pokud nejsou nastaveny hodnoty časů, tak ve funkci "Reset"(); nastavíme
IF NOT "Tprovoz" OR NOT "Tpozor" THEN
    "Reset"();
END_IF;

//Reset časovače v případě, že změníme režim
"Reset_casovace"();

IF "Tl_Start" AND NOT "Tl_noc" THEN
    "Denni_rezim"();
END_IF;

IF "Tl_Start" AND "Tl_noc" THEN
    "Nocni_rezim"();
END_IF;

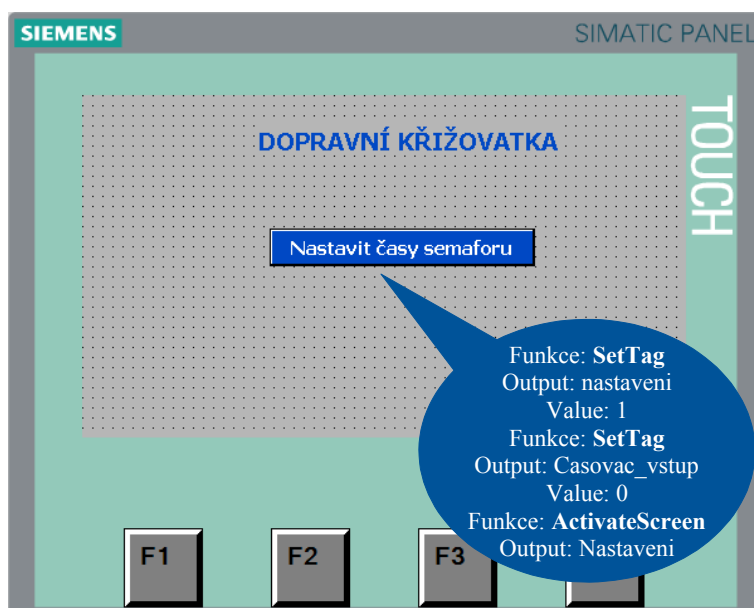
IF "Tl_Start" AND "Nastaveni" THEN
    "Rezim_nastaveni"();
END_IF;
```

Zdr. kód 17. Obsah hlavní funkce "Main"();.

9.5.1 Rozšíření s HMI panelem

V tomto úkolu přidáme do projektu rozšíření, kdy bude možné nastavit časy semaforu pomocí HMI panelu. Pokud bude docházet k nastavování časů, semafor bude ve speciálním režimu, bude to režim "Rezim_nastaveni()". Abychom mohli zapnout tento režim, tak při kliknutí na tlačítko Nastavit časy semaforu se do proměnné "Nastaveni" nastaví logická 1. Semafor tak přejde do příslušného stavu.

Nejdříve vytvoříme novou obrazovku v HMI panelu a to kliknutím na položku Add new screen v levé části obrazovky. Novou obrazovku pojmenujeme názvem Start. Dvojklikem na tuto položku si otevřeme obrazovku a přidáme do ní tlačítko, které nám zajistí přechod semaforu do režimu nastavení a my budeme moci změnit časy provozu křižovatky. Tlačítko do obrazovky přidáme tak, že z pravého panelu na obrazovce, konkrétně z nabídky Elements, přetáhneme ikonku, která vypadá jako tlačítko (šedý obdélník). Dvojklikem na text v tlačítku jej přejmenujeme na text „Nastavit časy semaforu“. Výsledek by měl vypadat podobně, jako je uvedeno na obrázku Obr. 38.



Obr. 38. Rozvržení obrazovky Start.

Dále je nutné nastavit akce, které se provedou při kliknutí. Konkrétně u tohoto tlačítka se jedná o zapsání hodnoty 1 do proměnné Nastavení a přesměrování na další obrazovku. Klikneme tedy pravým tlačítkem myši na tlačítko „Nastavit časy semaforu“ a vybereme položku Properties. Vybereme záložku Events a přidáme funkce, které jsou napsány v bublině v obrázku Obr. 38.

Nyní vytvoříme další obrazovku a pojmenujeme ji například Nastavení. Do této obrazovky vložíme 4 textová pole, která pojmenujeme následovně – Čas provoz, Čas pozor a další 2 s názvem ms. Dále 2 políčka, kam se bude zapisovat nastavovaný čas a jedno tlačítko pro uložení hodnot a návratu na obrazovku Start. Obrazovka bude vypadat podobně, jako je na obrázku Obr. 39. Akce u příslušných tlačítek jsou napsány opět v informačních bublinách.



Obr. 39. Rozvržení obrazovky Nastavení.

Poté obsah nahrajeme do HMI panelu kliknutím na ikonku Download. Funkčnost ověříme nastavením jiných časů, než jsme nastavili v úloze 9.3.1, konkrétně ve funkci "Reset");.

9.5.2 Rozšíření s Web Serverem

Nyní vytvoříme webové rozhraní, pomocí kterého bude možné nastavovat časy provozu v dopravní křižovatce. Na tomto webu bude i schéma křižovatky, toto schéma bude v reálném čase číst hodnoty na výstupech PLC a bude zobrazovat aktuální stav dopravní křižovatky.

Nejdříve vytvoříme web server v programovatelném automatu. V levém postranním panelu klikneme na položku Device configuration, která se nachází v kategorii PLC_1 [CPU 1214C DC/DC/DC]. V nastavení nalezneme položku Web server. Tento modul je ve výchozím stavu neaktivní, proto jej musíme aktivovat a to kliknutím na check box Enable Web server on this module. Dále je ještě možné zabezpečit připojení k Web serveru přes protokol https. Avšak v našem laboratorním testování není toto zabezpečení nutné. Ještě je potřeba nastavit cestu, kde budou umístěny soubory pro web server a nastavit jméno základního souboru. Tento soubor se nám zobrazí jako první, když zobrazíme novou stránku. V této bakalářské práci budu používat složku D:\VAŠE-STUDIJNÍ-ID\root\ a základní soubor bude mít název index.html. Tyto údaje vyplníme do příslušného formuláře. Nové nastavení programovatelného automatu nahrajeme do zařízení. Aby web server fungoval a mohli jsme si zobrazit případné označení chyb, tak je nutné přidat do hlavní funkce "Main()"; na první řádek novou instrukci a přidat novou proměnnou do tabulky symbolických proměnných.

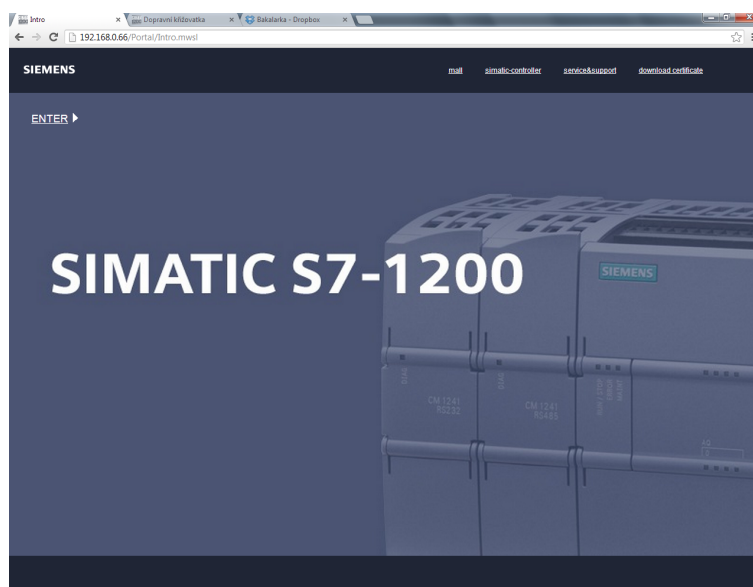
"web_out" := WWW(CTRL_DB:=333); //Číslo 333 je defaultní port webového serveru.

Zdr. kód 18. Úprava funkce Main() pro spuštění webserveru.

Název proměnné	Datový typ	Adresa
web_out	Int	%MW2

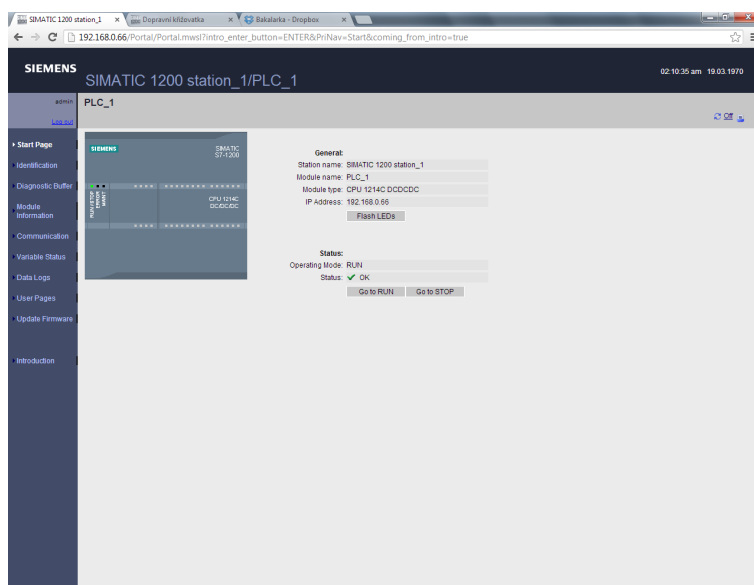
Tab. 9. Úprava tabulky symbolických proměnných.

V dalším kroku ověříme, zda web server funguje. Otevřeme jakýkoliv prohlížeč internetu a do adresy napíšeme IP adresu programovatelného automatu, v našem případě je to adresa <http://192.168.0.66>. Zobrazí se nám úvodní stránka web serveru.



Obr. 40. Úvodní obrazovka webového rozhraní.

Abychom mohli číst a zapisovat data z/do automatu, tak je nutné být přihlášený na tomto serveru. Pro přihlášení zadáme uživatelské jméno admin a heslo nevyplňujeme. Po přihlášení se nám zobrazí následující obrazovka.



Obr. 41. Obrazovka, která se zobrazí po přihlášení do webové administrace.

Nyní si vytvoříme html soubor, který nám bude zobrazovat aktuální stav dopravní křižovatky. Bohužel prostředí TiA Portal nenabízí žádné vývojové prostředí pro programování webových aplikací. K této činnosti nám však plně dostačuje jakýkoliv textový editor. Nově vytvořený soubor pojmenujeme `animace.html` a umístíme do složky `root`. Využijeme značkovacího jazyku HTML5, skriptovací jazyk JavaScript, skriptovou knihovnu jQuery a obrázku Obr. 31.

```
<canvas id="schema" width="390" height="270" style="background-image:
url(./style/img/krizovatka.png);">
```

Váš prohlížeč nepodporuje HTML5, pokud chcete vidět v reálném čase, ve kterém stavu se křižovatka nachází, tak si aktualizujte prohlížeč.

```
</canvas>
```

```
<script type="text/javascript">
```

```
    //Přiřazení proměnných z automatu do JS
```

```
    var CE_HL = ":=CE_HL:";
```

```
    var OR_HL = ":=OR_HL:";
```

```
    var ZE_HL = ":=ZE_HL:";
```

```
    var CE_VE = ":=CE_VE:";
```

```
    var OR_VE = ":=OR_VE:";
```

```
    var ZE_VE = ":=ZE_VE:";
```

```
    var ZE_PR = ":=ZE_PR:";
```

```
    var CE_PR = ":=CE_PR:";
```

```
    //Nastavení barev světel
```

```
    var cervena_barva = "#FF0000";
```

```
    var oranzova_barva = "#FF8000";
```

```
    var zelena_barva = "#00FF00";
```

```
    //Nastavení pro canvas
```

```
    var a=document.getElementById("schema");
```

```
    var b=a.getContext("2d");
```

```

function svetlo(x,y,barva)
{
    b.beginPath();
    b.fillStyle=barva;
    b.arc(x,y,7,0,2*Math.PI);
    b.closePath();
    b.fill();
    b.strokeStyle = "black";
    b.stroke();
}

//Vykreslení barev na semaforech
if(CE_VE == 1) svetlo(147.5,88.5,cervena_barva); //Červená vedlejší
if(OR_VE == 1) svetlo(147.5,68,oranzova_barva); //Oranžová vedlejší
if(ZE_VE == 1) svetlo(147.5,47.5,zelena_barva); //Zelená vedlejší

if(CE_HL == 1) svetlo(147.5,187.5,cervena_barva); //Červená hlavní
if(OR_HL == 1) svetlo(127.5,187.5,oranzova_barva); //Oranžová hlavní
if(ZE_HL == 1) svetlo(107.5,187.5,zelena_barva); //Zelená hlavní

if(CE_PR == 1) svetlo(253.5,68,cervena_barva); //Červená přechod
if(ZE_PR == 1) svetlo(253.5,88.5,zelena_barva); //Zelená přechod
</script>

```

Zdr. kód 19. Obsah souboru animace.html

Dále je nutné vytvořit základní soubor, nazveme ho index.html a umístíme do složky root. Na začátku tohoto souboru musíme napsat všechny proměnné, které budeme upravovat. Poté ve formuláři napíšeme hodnoty, ve skrytém poli, které chceme upravit. Do hodnoty value napíšeme hodnoty, které chceme nastavit do příslušné proměnné, která je pojmenována v parametru name. Dále ještě pomocí jQuery aktualizujeme prvek, který má id semafor, každou vteřinu. Do tohoto prvku vkládáme soubor, který jsme vytvořili dříve. Tzn. Každou vteřinu se tento soubor animace.html znovu otevře, plátno se schématem se tedy překreslí a zobrazí aktuální stav dopravní křižovatky.

```

<!-- Komentáře typu AWP_In_Variable musí být uvedeny pro všechny proměnné, -->
<!-- které jsou ve webové prezentaci použity pro zápis dat do automatu. -->
<!-- AWP_In_Variable Name='"nastaveni"' -->
<!-- AWP_In_Variable Name='"Casovac_vstup"' -->

<!DOCTYPE HTML>
<html style="width: 100%; height: 100%;">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<link rel="stylesheet" href="./style/style.css">
<title>Dopravní křižovatka</title>
<!-- Do hlavičky musíme vložit odkaz pro knihovnu jQuery, kterou používáme pro -->
<!-- aktualizaci stavů proměnných. Více informací v dokumentaci jQuery. -->
<script type="text/javascript" src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

</head>
<body>
<div class="okno">
<h1>Nastavení dopravní křižovatky</h1>
<div id="semafor"></div>
<div class="popisek">Aktuální stav dopravní křižovatky.</div>

```



```

<form method="POST" action="./nastaveni.html">
  <fieldset>
    <p style='text-align: center;'>
      <input type="submit" value="Nastavení časů semaforu" class="submit">
      <input type="hidden" name="'nastaveni'" value="1">
      <input type="hidden" name="'Casovac_vstup'" value="0">
    </p>
  </fieldset>
</form>
</div>

<script type="text/javascript">
  //Aktualizace stavu křižovatky každou 1s
  setInterval(function() {
    $("#semafor").load("animace.html");
  }, 1000);
</script>
</body>
</html>

```

Zdr. kód 20. Obsah souboru index.html.

Ještě musíme vytvořit soubor, na který se odkazujeme v souboru index.html, je to soubor s názvem nastaveni.html. Pokud klikneme na tlačítko, které je ve formuláři, tak proměnné nastaveni se přiřadí logická 1 a v PLC se nastaví "Rezim_nastaveni"(); a proměnné Casovac_vstup se přiřadí logická 0, abychom resetovali časovač.

```

<!-- AWP_In_Variable Name="'Tprovoz'" -->
<!-- AWP_In_Variable Name="'Tpozor'" -->
<!-- AWP_In_Variable Name="'nastaveni'" -->
<!-- AWP_In_Variable Name="'Casovac_vstup'" -->

<!DOCTYPE HTML>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <link rel="stylesheet" href="./style/style.css">
  <title>Nastavení časů dopravní křižovatky</title>
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
</head>
<body>
  <div class="okno">
    <h1>Nastavení časů dopravní křižovatky</h1>
    <div id="semafor"></div>
    <div class="popisek">Aktuální stav dopravní křižovatky.</div>
    <form method="POST">
      <fieldset>
        <p><label id="provoz">Čas provoz:</label><span class="info">Pokud chcete čas
zadat v sekundách, tak jej zadejte ve formátu <strong>T#5S</strong>, kdy číslo
<strong>5</strong> znamená 5s.</span><input type="text" name="'Tprovoz'"
value=":'Tprovoz':" size="5" /></p>
        <p><label id="pozor">Čas pozor:</label><span class="info">Pokud čas zadáte
pouze jako číslo, např. <strong>1000</strong>, tak tato hodnota odpovídá času
v milisekundách. Tedy hodnota <strong>1000</strong> odpovídá

```

```

<strong>ls</strong>.</span><input type="text" name="'Tpozor"' value=':="Tpozor":'
size="5"></p>
    <p><input type="submit" value="Zapsat hodnoty do PLC" class="submit"></p>
</fieldset>
</form>
    <p>
        <form method="POST" action="./index.html">
            <fieldset>
                <input type="submit" value="Uložit a ukončit" class="submit">
                <input type="hidden" name="'nastaveni"' value="0">
                <input type="hidden" name="'Casovac_vstup"' value="0">
            </fieldset>
        </form>
    </p>
</fieldset>
</form>
</div>

<script type="text/javascript">
    setInterval(function() {
        $('#semafor').load("animace.html");
    }, 1000);
</script>
</body>
</html>

```

Zdr. kód 21. Obsah souboru nastaveni.html.

Dále je ještě dobré nastavit grafiku přes CSS3. Vzhled nemá vliv na funkčnost programu, proto si jej uživatel může nastavit podle své fantazie. CSS kód může vypadat podobně, jako v souboru style.css.

```

body{ padding: 0px; margin: 0px; font-family: Arial; color: #0063C6; background-color:
#CCC; }
.okno
{
    margin: 20px auto;
    padding: 0px 5px 10px 5px;
    width: 600px;
    height: 100%;
    background-color: #EEE;
    border-left: 1px solid #c4c4c4;
    border-right: 1px solid #c4c4c4;
    -moz-box-shadow: 0 0 20px 1px #AAA;
    -webkit-box-shadow: 0 0 20px 1px #AAA;
    box-shadow: 0 0 20px 1px #AAA;
}
#semafor
{
    margin: 20px auto 5px;
    border: 1px solid #c4c4c4;
    background-color: #FFF;
    background-image: url(../img/krizovatka.png);
    background-repeat: no-repeat;
    width: 390px;
    height: 270px;
}

```

```

input
{
    font-size: 16px;
    padding: 5px 10px;
    color: #0063C6;
    background-color: #CCC;
    border: 1px solid #0063C6;
}
input.submit
{
    font-size: 18px;
    padding: 5px 10px;
    margin: 0px;
    background-color: #004993;
    border: 1px solid #004993;
    cursor: pointer;
    color: #FFF;
}
input.submit:hover
{
    background-color: #0063C6;
    border: 1px solid #0063C6;
    cursor: pointer;
}
h1{ font-size: 35px; color: #0063C6; font-weight: 300; text-align: center; }
p{ font-weight: 900; }
input.submit:active{ background-color: #0063C6; }
fieldset{ width: 390px; margin: 20px auto; padding: 0px; border: none; }
label{ display: block; padding: 0px; margin: 0px; }
.info{ font-style: italic; font-size: 14px; display: block; font-weight: 300; color:
#004993; }
.popisek{ text-align: center; font-style: italic; }

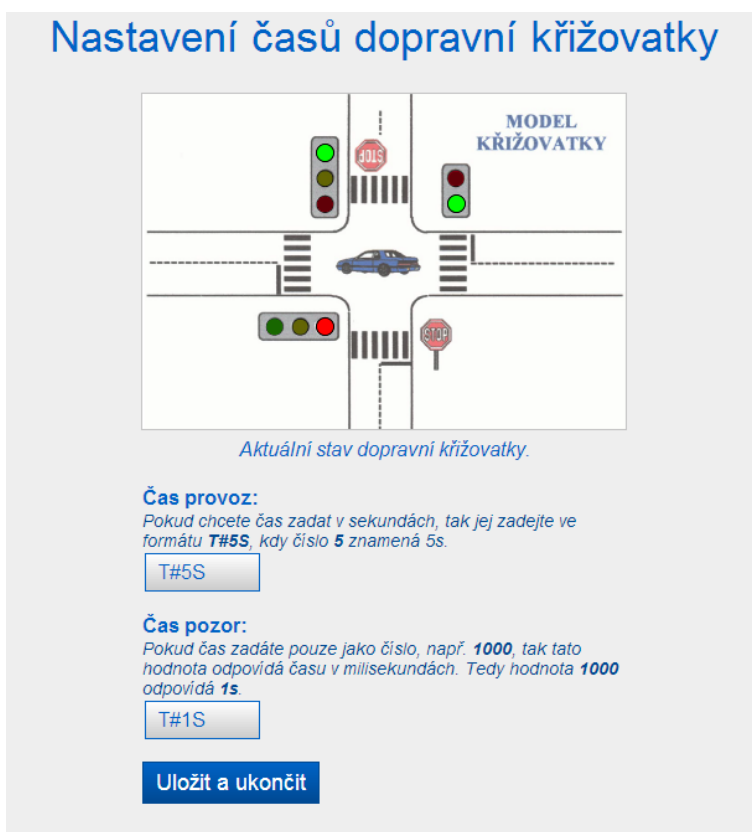
```

Zdr. kód 22. Obsah souboru style.css.

Nyní nastal čas na odzkoušení webserveru. Do adresy prohlížeče napíšeme IP adresu automatu, poté se přihlásíme za administrátora a zobrazí se nám stránka, jako je uvedeno na obrázku Obr. 41. Poté klikneme v levém panelu na položku User Pages a dojde k zobrazení stránky, kterou jsme vytvořili. Stránka by měla vypadat podobně, jako je uvedeno na obrázku Obr. 42. Pokud klikneme na tlačítko Nastavení časů semaforu, zobrazí se nám stránka, která je uvedena na obrázku Obr. 43. a na křižovatce se nastaví režim nastavení. Po nastavení časů a kliknutí na tlačítko Uložit a ukončit dojde ke změně časů a křižovatka se přepne do výchozího režimu (denní, noční). Ve webovém rozhraní se opět zobrazí stránka, která je uvedena na obrázku Obr. 42.



Obr. 42. Základní obrazovka nastavení přes web server.



Obr. 43. Nastavení časů dopravní křižovatky.

10 ZÁVĚR

Nejdříve bylo nutné seznámit se s novým programovatelným automatem Siemens Simatic S7-1200. Programování aplikací do tohoto automatu je možné třemi programovacími jazyky – Ladder diagram, jazyk Funkčních bloků a strukturovaný jazyk SCL. V této bakalářské práci byl využit jazyk SCL. Jelikož v některých úkolech v této bakalářské práci využívám i dotykové panely, tak dále bylo nutné se seznámit s HMI panely KTP400 Basic color. PLC i HMI panel obsahuje komunikační rozhraní PROFINET, proto nebylo nutné vyrábět kabely pro připojení těchto zařízení s PC, ve kterém probíhalo programování aplikací pomocí prostředí TiA Portal. Prostředí TiA Portal je dodáváno společně s programovatelným automatem. Vyrábět jsme museli pouze desky plošných spojů, abychom mohli lépe propojit modely EDU-mod s programovatelným automatem.

Dalším cílem této bakalářské práce bylo se seznámit s modely EDU-mod. V případě této bakalářské práce se jednalo o model automatické pračky a dopravní křižovatky. S těmito modely jsem byl seznámen již dostatečně z předmětu Programovatelné automaty (VPL), kdy na těchto modelech probíhala výuka, avšak programování při výuce probíhalo na starším typu PLC Simatic S7-200 a v jazyce STL.

Dalším krokem při tvorbě bakalářské práce bylo vymezení a rozvržení úkolů. U každého modelu EDU-mod jsou vytvořeny 3 vzorová zadání. První zadání je jednoduché a časově méně náročné. Je určeno k seznámení s programováním pomocí jazyka SCL a novou řadou programovatelných automatů. Na tento jednoduchý úkol navazuje rozšíření. Druhý úkol je již složitější než úkol první. Druhý úkol u modelu automatické pračky obsahuje již rozšíření o dotykový panel, kdy si uživatel/ka zvolí prací režim. U modelu dopravní křižovatky je u druhého úkolu rozšíření o další režim, což je po denním režimu režim noční. U modelu automatické pračky se ve třetím úkolu ke zbylým dvěma úkolům vytvoří servisní mód, kdy technik může upravovat teplotu a čas praní. Vstup do servisního režimu je zabezpečen zadáním správného uživatelského jména a hesla. Třetí úkol u modelu dopravní křižovatky se týká HMI panelu a Web serveru. Konkrétně u rozšíření s dotykovým panelem se ke zbylým dvěma úkolům vytvoří nastavení časů provozu pomocí HMI panelu. U zadání s Web serverem se jedná o indikaci stavu dopravní křižovatky přes webový prohlížeč. V prohlížeči je v reálném čase zobrazován aktuální stav křižovatky. Přes webový prohlížeč je taktéž možné nastavit časy provozu na dopravní křižovatce. Nastavení časů provozu je podmíněno úspěšným přihlášením do administrace webového rozhraní. Webové rozhraní již obsahuje administraci, a proto není nutné ji vytvářet. K projektu s Web serverem je taktéž vytvořen kompletní grafický design pomocí nových technologií, jako je značkovací jazyk HTML5, kaskádové styly CSS3 a knihovna javascriptu jQuery.

Všechny vytvořené projekty byly odzkoušeny v programovatelném automatu řady Simatic S7-1200 a jsou funkční. Typ a způsob řešení úkolů v této bakalářské práci je přizpůsoben studentům, kteří prozatím neměli možnost se seznámit s automaty Simatic S7-1200.

SEZNAM POUŽITÉ LITERATURY

- [1] Siemens, s.r.o. SIMATIC WinCC V11(TIA Portal) [online]. [cit. 2013-04-14]. Dostupné z: <http://www1.siemens.cz/ad/current/index.php?ctxnh=68570cea9e&ctxp=home>
- [2] Siemens, s.r.o. Working with ProSave [online]. [cit. 2013-04-14]. Dostupné z: <https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW&objid=26556941&treeLang=en>
- [3] Siemens, s.r.o. Updates for STEP 7 V11 SP2 and WinCC V11 SP2 [online]. [cit. 2013-02-07]. Dostupné z: <https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=58112582&caller=view>
- [4] BLAŽEK, J. TIA portál V11 pro programování PLC Simatic S7, díl 1. [online]. [cit. 2013-02-07]. Dostupné z: http://www.foxon.cz/index.php?main_page=faq_info&fcPath=30_31&faqs_id=187ca
- [5] Siemens, s.r.o. STEP 7 (TIA Portal) V11 Support Packages for the hardware catalog in the TIA Portal [online]. [cit. 2013-02-07]. Dostupné z: <https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=54164095&caller=view>
- [6] RAKUŠAN, O. TIA na dosah [prezentace]. 2011-04. [cit. 2013-05-08]. Dostupné z: http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/prez_s7-1200-step7-basic-fw2_2011_cz.pdf
- [7] Siemens, s.r.o. Řídicí systém Simatic S7-1200 [online]. [cit. 2013-05-08]. Dostupné z: <http://stest1.etnetera.cz/ad/current/index.php?ctxnh=5dc8474325>
- [8] KOHOUT, L. Modely EDU-mod [online]. [cit. 2013-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=edumod>
- [9] KOHOUT, L. Modely EDU-mod - Pračka [online]. [cit. 2013-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=pracka>
- [10] MARADA, T. Řízení modelu automatické pračky programovatelným automatem Simatic S7-200 CPU 224XP [online]. [cit. 2013-04-27]. Dostupné z: <https://www.vutbr.cz/elearning/mod/resource/view.php?id=200978>
- [11] Siemens, s.r.o. System manual [online]. [cit. 2013-04-28]. Dostupné z: <https://support.automation.siemens.com/CN/llisapi.dll?query=STEP+7+V5.5+SP2+Chinese&func=cslib.cssearch&content=adsearch%2Fadsearch.aspx&lang=en&siteid=cseus&objaction=cssearch&searchinprim=&nodeid99=&x=17&y=3>
- [12] MARADA, T. Řízení modelu dopravní křižovatky programovatelným automatem Simatic S7-200 CPU 224XP [online]. [cit. 2013-04-29]. Dostupné z: <https://www.vutbr.cz/elearning/mod/resource/view.php?id=200979>