Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

Bakalářská práce

Přípravek pro praktickou výuku mechatroniky

Autor práce: Lukáš Svatoš
Vedoucí práce: Ing. Jiří Lahoda, Ph.D.

Plzeň 2018

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2017/2018**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Lukáš SVATOŠ**
Osobní číslo:	**E15B0028P**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Elektronika a telekomunikace**
Název tématu:	**Přípravek pro praktickou výuku Mechatroniky**
Zadávající katedra:	**Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Stručně popište programovatelný logický automat Siemens Simatic S7-1500 a základní rozšiřující moduly.
2. Stručně popište vývojové prostředí TIA Portal.
3. Navrhněte a realizujte přípravek pro výuku logického řízení, který bude připojen k PLC Siemens Simatic S7-1500.
4. Navrhněte a realizujte alespoň dvě vzorové úlohy pro PLC Siemens Simatic S7-1500 s využitím navrženého přípravku.
5. Zhodnoťte možnosti použití přípravku v praktické výuce Mechatroniky.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **30 - 40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: **Ing. Jiří Lahoda, Ph.D.**
Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **10. října 2017**

Termín odevzdání bakalářské práce: **7. června 2018**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan

L.S.

Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 10. října 2017

Abstrakt

V úvodu můžete nalézt stručný přehled rozdělení PLC podle konstrukce a bližší popis programovatelného logického automatu Siemens Simatic S7-1500, ke kterému je navržen přípravek pro výuku mechatroniky. Další část obsahuje přehled používaných programovacích jazyků pro PLC. Stěžejním úsekem jsou následující kapitoly o návrhu a realizaci přípravku. Společně s vytvořením výukových programů pro tento přípravek je na konci této práce shrnutí o použití v praxi.

Klíčová slova

paritní bity, modulární PLC, mikrořadič, LAD, SPI, TIA Portal

Abstract

Svatoš, Lukáš. *Model for Practical Teaching of Mechatronics* [*Přípravek pro praktickou výuku mechatroniky*]. Pilsen, 2018. Bachelor thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Jiří Lahoda

---

In the introduction, you can find a brief overview of the PLC according to the design and a more detailed description of the Siemens Simatic S7-1500 programmable logic controller, which co-operated with the manufactured product. The next section contains an overview of the PLC programming languages used. The central section of the following chapters on the design and implementation of the product. Together with the creation of training programs for this product, there is a summary of the use in practice at the end of this work.

Keywords

parity bits, modular PLC, microcontroller, LAD, SPI, TIA Portal

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

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

V Plzni dne 6. června 2018

Lukáš Svatoš

.....

Podpis

Poděkování

Tato práce vznikla s finanční podporou Vyšší odborné školy a Střední průmyslové školy elektrotechnické v Plzni.

Obsah

Seznam obrázků

Seznam tabulek

Seznam symbolů a zkratek

PLC	Programmable Logic Controlers. Programovatelný logický automat.
DI	Digital inputs. Digitální vstupy.
DO	Digital outputs. Digitální výstupy.
AI	Analog inputs. Analogové vstupy.
AO	Analog outputs. Analogové výstupy.
HW	Hardware.
PTO	Differential Pulse Train Output. Výstup diferenciálního pulzního průběhu.
PWM	Pulse Width Modulation. Pulzně šířková modulace.
CPU	Central processing unit. Centrální procesorová jednotka.
SCADA	Supervisory Control And Data Acquisition. Dispečerské řízení a sběr dat.
HMI	Human Machine Interface. Rozhraní mezi člověkem a strojem.
STL	Statement List. Seznam výpisů.
FBD	Function Block Diagram. Diagram funkčních bloků.
LAD	Ladder Diagram. Liniové schéma.
SCL	Structured Control Language. Stukturovaný programovací jazyk.
SPI	Serial Peripheral Interface. Sériové periferní rozhraní.
MSB	Most Significant Bit. Nejvýznamnější bit.
LSB	Least significant bit. Nejméně významný bit.

1

Úvod

Tato práce se zabývá návrhem a realizací přípravku pro výuku mechatroniky. Tento model je kvůli demonstraci využit k tvorbě výukových programů pro Siemens Simatic S7-1500.

První část je zaměřena na základní popis PLC a řad logických automatů Siemens Simatic, jakožto jednoho z největších výrobců těchto zařízení. Hlavní specifikace je zaměřena pouze na jeden model a to Siemens Simatic S7-1500. Tvorba přípravku pro školu (VOŠ a SPŠE Plzeň), pro kterou byl od začátku tento model určen, využívá právě tento modulární systémem PLC v nově vytvořené učebně.

Do tohoto popisu samozřejmě spadá i software, který se pro programování PLC od firmy Siemens využívá. Tím softwarem je TIA Portal, jenž má v sobě implementované dříve samostatné aplikace jako Step7 a WinCC. Ty umožňují využití mnoha programovacích jazyků a spolupráci s mnoha samostatnými systémy u automatizovaného řízení např. HMI i celkové SCADA. To vše je v této práci také stručně shrnuto.

Stěžejním úsekem mé práce byl samotný návrh přípravku a vytvoření výukových programů pro daný model. Jsou zde popsány jednotlivé kroky při návrhu přípravku, co k němu vedlo, a také blíže vysvětlené vytvořené úlohy pro demonstrativní účely. Jádrem přípravku je mikrokontrolér, který vytváří funkce pro jednotlivé řízení prvků na vyrobené desce. Proto je zde podrobně popsána komunikace s rozšiřující deskou a jejími integrovanými obvody, které způsobují převod z napěťové hladiny PLC na akceptovatelnou hladinu pro mikrořadič. Komunikace zde probíhá po sériovém rozhraní SPI.

2

Siemens Simatic S7-1500 a základní rozšiřující moduly

2.1 Význam PLC a princip jeho činnosti

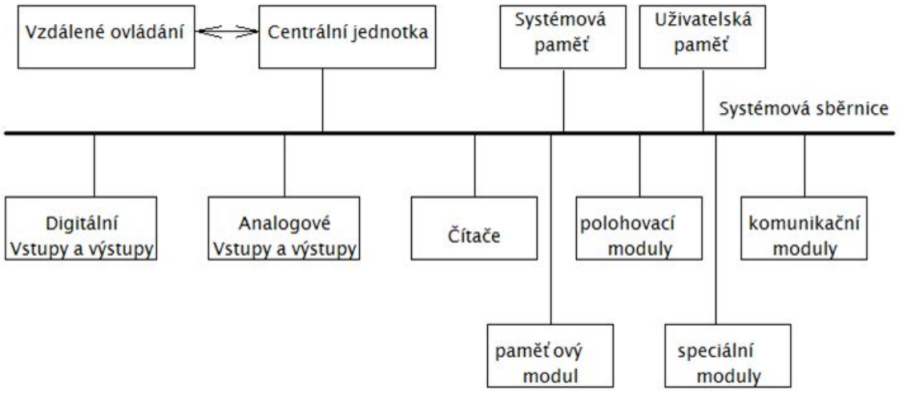
Programovatelné automaty PLC jsou v dnešní době nejvýznamnější řídicí prostředky v oblasti automatizace. Vznikly jako odezva na vývoj centralizovaného řízení řídicími počítači a mikropočítači. Jejich výhodou oproti mikropočítačové technice je vysoká spolehlivost, robustnost, práce s vyšším napětím a rozdělení různých modulů na samostatné celky s danými rozhraními. Díky tomu můžeme rychleji uvést PLC do provozu, zbavujeme se vysokých nároků na kvalifikaci obsluhy/programátorů a tím i optimalizujeme cenu výsledného projektu. Právě jednoduché programovací jazyky jazyk, které mohou mít u PLC programování mnoho podob, stály za velkým rozmachem průmyslových automatů. Jednotlivé možnosti používání různých programovacích jazyků PLC jsou blíže popsány v následující kapitole. Komunikace s řízeným hardwarem se provádí pomocí vstupních a nebo výstupních modulů, které mohou být analogové nebo digitální. Samotnou sestavu PLC můžeme dělit podle typu na: [2]

1. Kompaktní PLC
2. Modulární PLC
3. Micro PLC

2.1.1 Hardware PLC

Na začátku vývoje programovatelných logických automatů bylo cílem vývoje nahradit efektivnějším způsobem reléovou a bezkontaktní logiku, a jelikož v té době pouze 16-bitové procesory nevyhovovaly nárokům na rychlost, tak řešením bylo použití bitových procesorů. S tím se kladly i nároky na odlišné vlastnosti celé struktury PLC, jako je např. bitově orientovaná paměť dat a programu.

V dnešní době rychlost mikroprocesorů je již několikanásobně vyšší než u původních 16-bitových a díky tomu můžeme dosahovat menších výrobních nákladů, což způsobilo změnu architektury oproti původním návrhům. Blokové schéma dnešního standardního modulárního PLC se velmi podobá architektuře mikropočítače. V dnešní době už častěji používaná 32-bitová sběrnice je základem moderních programovatelných automatů a na ní je automat postaven modulárně celý. Používá se stejně jako u normálních PC jedna operační paměť, která má rozdělené prostory pro vstupní a výstupní data, vnitřní program a např. i prostor pro vnitřní proměnné. Zároveň vnitřní operační systém PLC zůstává nadále velmi jednoduchý.[2]



Obr. 2.1: Struktura PLC |Převzato z [1]|

Nejčastější a hlavní využití PLC je při řízení různých výrobních procesů. Výrobní procesy se již nevidí jako jednotlivé dílčí procesy, ale spíše jako integrální součásti celého výrobního procesu. Kompletní integrace celého prostředí automatizace je dnes dosažena pomocí:

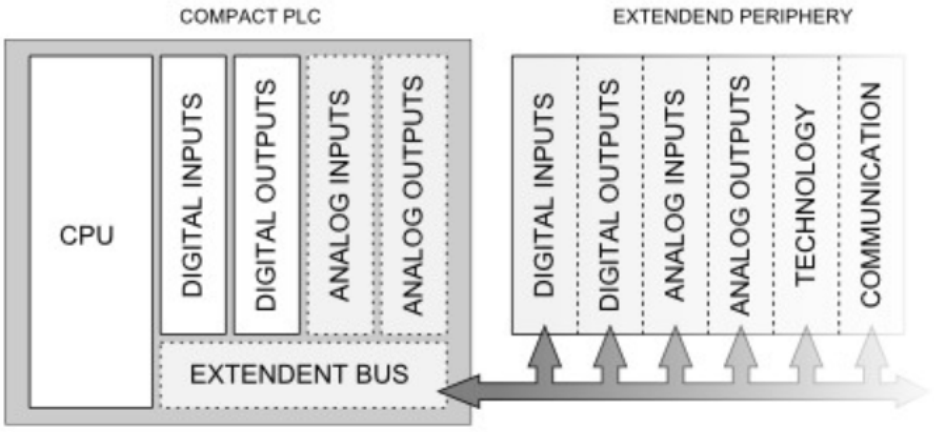
- jednoho běžného prostředí softwaru, který integruje všechny součásti a úkoly do jednotného snadno použitelného systému
- společné správy dat (centrální databáze)
- společné komunikace mezi všemi zúčastněnými komponenty automatizace

2.1.2 Kompaktní PLC

Je to způsob provedení PLC, kdy v jednom modulu je integrovaný nejen procesor, ale rovnou DI, DO, AI, AO. Z digitálních vstupů mohou být některé použity pro zpracování procesních hardwarových přerušení nebo pro vstupy rychlých čítačů, měření frekvence, apod. U výstupů může být vložen výstup PWM a u některých CPU i mnoho dalších komunikačních rozhraní.

Díky implementaci periferií v kompaktním celku má tato struktura daleko rychlejší přístup k periferiím, jelikož signály nemusí procházet řadičem sběrnice.

Kompaktní PLC jsou především určena pro menší řídicí systémy, protože uživatel může k základnímu modulu připojit pouze několik přídavných modulů, které navíc mají pevně danou kombinaci vstupů/výstupů.[4]



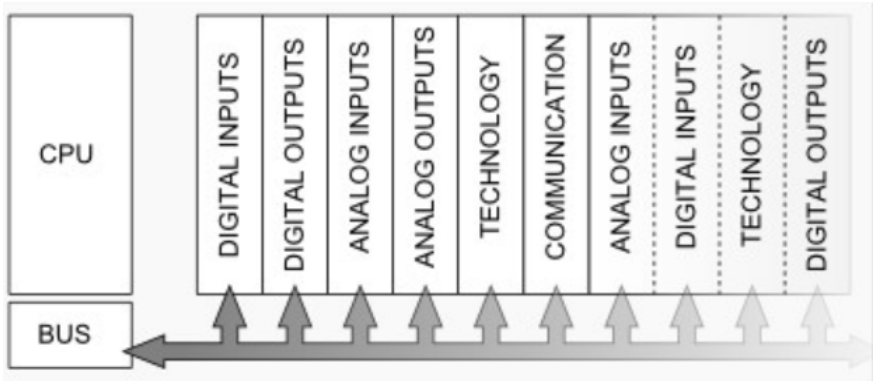
Obr. 2.2: Principiální nákres kompaktního PLC s možným rozšíření periferií |Převzato z [4]|

2.1.3 Modulární PLC

Modulární PLC (jak už sám název naznačuje) jsou sestavena z několika modulů, které jsou vložené do společné sestavy. To umožňuje daleko větší volnost při volbě konfigurace. Základním modulem je CPU a k němu dle volby připojujeme další periferní moduly. Ty dávají při výběr bohatou škálu možností připojených DI, DO, AI, AO.

Rozšiřující moduly mohou být připojeny na vzdálenost až stovek metrů. To je také jednou z velkých výhod modulárních PLC. Řídící CPU může být zcela mimo oblast obsluhovaného prostředí.

Modulární zapojení programovatelných automatů musí být již dopředu vytvořené pro větší množství signálů, se kterými pracuje, a proto je potřeba daleko větší kapacita paměti pro program i data.[4]



Obr. 2.3: Principiální nákres modulárního PLC s periferiemi |Převzato z [4]|

2.1.4 Mikro PLC

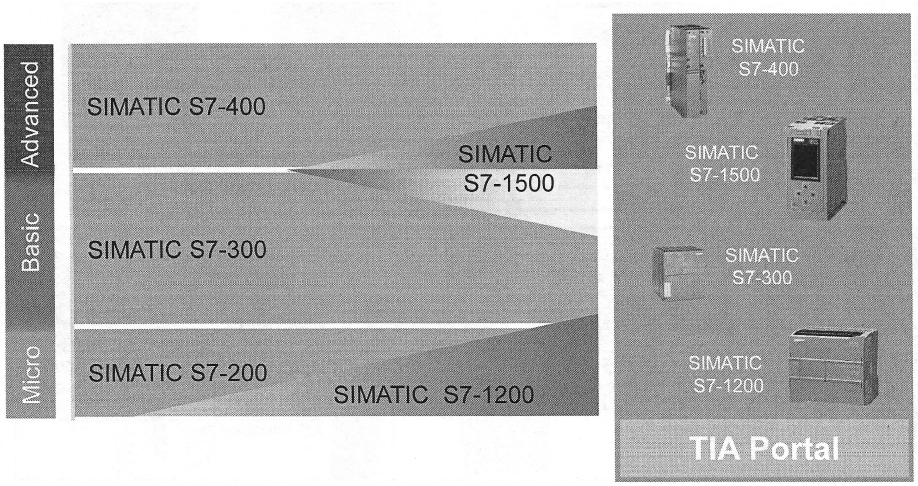
Je to jedno z možností stavby kompaktního PLC. Nabízejí koncovému uživateli pevnou sestavu periferií, kterou již nemůžeme dodatečně rozšiřovat. Při volbě se tedy již na začátku musí uživatel dle požadavků rozhodnout pro jeden typ systému.

Použití takového typu PLC může být nejčastěji jako ovládání jednoduchých strojů a mechanismů, které se dříve musely řešit kombinační nebo reléovou logikou.

2.2 Řada Siemens Simatic S7

„Řídicí systémy SIMATIC jsou známy především svojí spolehlivostí a robustností. Již řadu let jsou stabilním prvkem nejrůznějších technologií. Své renomé si získala dnes už výběhová řada SIMATIC S5. Na ni úspěšně navázala řada SIMATIC S7, která dodnes nabízí nejmodernější způsoby řešení technologických aplikací a je často nositelem inovací v celém oboru průmyslové automatizace. Tak jak se mění požadavky řešených úloh, jsou neustále vyvíjeny i nové řídicí prvky tak, aby co nejlépe vyhovovaly potřebám technologie, splňovaly náročné podmínky efektivního projektování a inženýringu a přitom respektovaly kontinuitu a pracovaly v souladu s již osvědčenými postupy a principy.“[3]

V dnešní době právě nastupující řada S7 nabízí mnoho modulárních modelů PLC a můžeme si při výběru řízení vybrat dle uživatelských nároků. Musíme brát v úvahu velikost výrobního procesu, míru možného rozšíření a prostředí instalování jednotky (viz. Obr.2.4). V této práci je popsán především model S7-1500, protože implementace programu a celková spolupráce s vyrobeným přípravkem probíhala právě na tomto modelu. Nejpoužívanější produkty z řady S7 můžeme vidět na dole na obrázku.



Obr. 2.4: Ukázka modulárních PLC řady S7 |Převzato z [5]|

2.3 Siemens Simatic S7-1500

Siemens S7-1500 je jedním z posledních modelů vycházející z vývoje v řadě S7 a skýtá mnoho nových funkcí a vlastností, které vychází z požadavků poptávky a maximální usnadnění práce pro koncového uživatele. Nový model představuje především zvýšení výkonu, funkční bezpečnosti a vylepšení použitelnosti v praxi. Díky perfektní integraci do TIA Portal software lze i vývoj uživatelských aplikačních programů maximálně zefektivnit. Nabízí velmi rychlou systémovou sběrnici s velkou rychlostí přenosu dat, proto umožňuje dosáhnout rychlého zpracování signálů a kvality výsledného řízení. Samozřejmostí je i zvládnutí běžných provozních aplikací, jako jsou řízení polohy a pohybu. To vše zvládne již bez přídavných modulů. Další integrovanou funkcí je PID regulace, což nám umožňuje jednodušší práci při regulovaném procesu. Optimální hodnoty parametrů PID regulátorů se nastavují automaticky v jednotce CPU.

Obr. 2.5: Model S7-1500 |Převzato z [3]|

Bezpečnost je řešena chráněným přístupem. Tím zabráníme před neoprávněnými změnami konfigurace programu. Jednotlivé bloky PLC můžeme svázat sériovým číslem použité paměťové karty. S7-1500 navíc umožňuje nastavovat oprávnění různých úrovní a tím pádem i nastavit různým uživatelům rozdílná práva přístupu.

Funkční bezpečnost je chráněna mnoha způsoby od základních hardwarových nastavení až po možnosti vývojového prostředí TIA Portal, který kontroluje funkčnost našeho programu.

Každou jednotku CPU řady S7-1500 lze snadno centrálně rozšířit až o 32 přídavných modulů. Ty se mohou instalovat na DIN lištu, což vede k dalšímu usnadnění. Řídící jednotky jsou navíc již v základu vybaveny displejem, a díky tomu můžeme mít rychlé informování o textových hlášeních z PLC. To je v praxi velikou výhodou, jelikož obsluha PLC nemusí mít potřebné technické znalosti a viditelné chybové hlášení je mohou posunout dále při řešení problému.[3]

2.4 DI, DO, AI, AO

Jak už je v krátkosti z předchozích popisů naznačeno, tak i model S7-1500 umožňuje k základnímu CPU připojit velkou řadu modulů DI, DO, AI, AO. Propracovaný podsys-

tém vstupů a výstupů umožňuje velikou variabilitu možného nastavení PLC. To všechno je jednou z příčin stále rostoucí popularity programovatelných automatů. Již nákupem daných modulů máme vyřešen všechen podsystém vstupů a výstupů, se kterými můžeme rovnou pracovat. V případě řízení mikropočítačem se musí případný podsystém vstupů a výstupů dále řešit.

Digitální vstupy (DI) jsou nejčastěji pro univerzální použití (možnost střídavého nebo stejnosměrného napětí). Kvůli ochraně jsou vybaveny galvanickým oddělením realizovaným optronem. Obsahují také filtr pro minimalizaci šumu a poruchových signálů. Kromě toho používají ještě diodový můstek jako ochranu proti přepólování a napěťovým špičkám.

Digitální výstupy (DO) mohou být řešeny tranzistory (pro malé výkony), tyristory (pro větší výkony) a nebo reléovým spínáním, také pro větší výkony. DI i DO využíváme v praxi jako propojení se snímači nebo akčními členy.

Typické pro podsystémy modulárních PLC jsou analogové vstupy a výstupy (AI a AO). Vstupy mohou být řešeny pro stejnosměrné napětí nebo proud. Daná hodnota je poté přes A/D převodníky převedena na číslicovou (typicky 12-bitovou) hodnotu. Před samotným navzrokováním zesilujeme signál, abychom neztratili jeho integritu a mohli s ním co nejlépe pracovat.

Výstupy se často realizují formou signálu PWM z D/A převodníku, kdy změnou střídy pulsů měníme střední hodnotu.

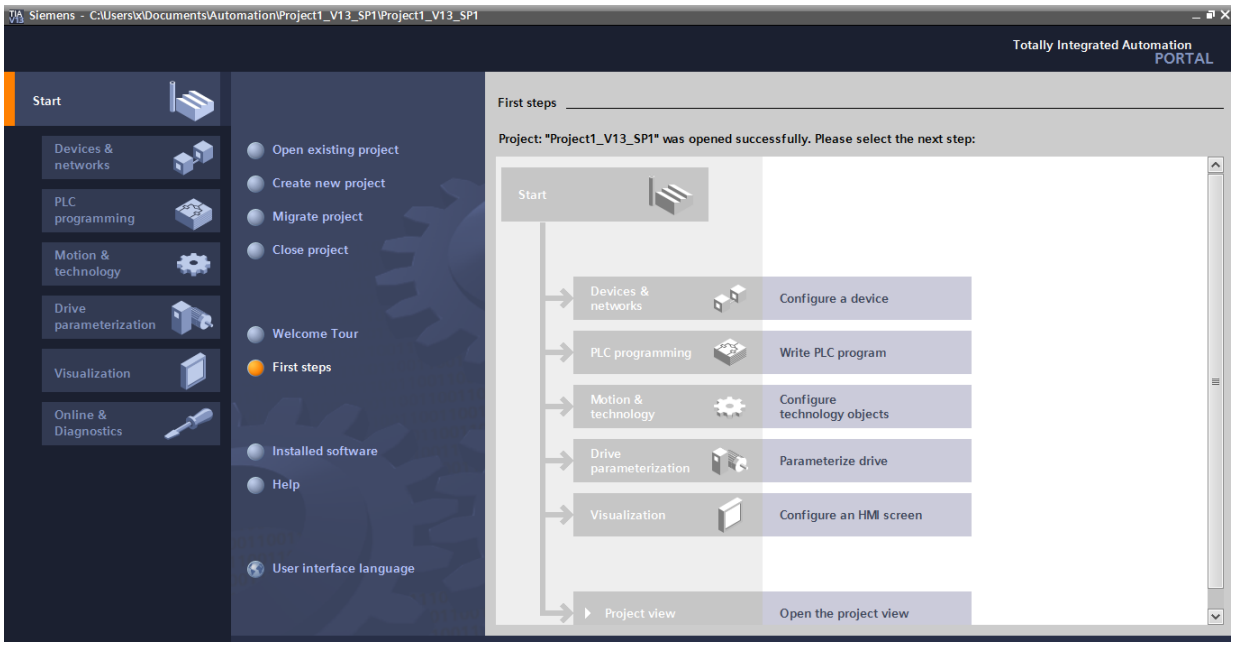
Parametry	Model S7-1500 C
Rozhraní	PROFINET
Programová/datová paměť	250 KB / 1 MB
Bitová rychlost	48 [ns]
Připojení DI/DO	32 / 32
Připojení AI/AO	4+1 / 2
Počet vysokorychlostních čítačů HSC	6
PTO/PWM	4 / 4
Šířka	110 [mm]

Tab. 2.1: Hardwarové parametry modelu Siemens Simatic S7-1500 C

3

Možnosti prostředí TIA Portal

Totally Integrated Automation Portal, ve zkratce TIA Portal, je software vytvořený především pro vývoj aplikací k PLC produktům značky Siemens. Dává ale mnoho dalších možností při navrhování řízení automatizační technikou. K samotnému vývoji patří i projektování panelů HMI, rozsáhlých vizualizací SCADA, síťových komponent a komunikačních prvků - to vše je integrováno do společného softwaru, který působí pro uživatele jednoduše a přívětivě.



Obr. 3.1: Ukázka z vývojového prostředí TIA Portal

Toto komplexní vývojové prostředí sjednocuje vývoj aplikací z oblasti řídicí techniky, decentralizovaných periferií i pro vizualizaci v blízkosti stroje až po rozsáhlá řešení SCADA. Tedy když software z centrálního pracoviště monitoruje/ovládá průmyslová a jiná technická zařízení a procesy.

Velkou pomocí při návrhu jsou inteligentní editory. Ty citlivě s ohledem na kontext zdůrazní přesně to, co vývojář či servisní technik právě potřebuje pro konkrétní úlohu:

funkce, vlastnosti, knihovny. „Společná konfigurace přístrojů a sítě umožňuje plně grafické projektování celého hardwaru a systému sběrnic.“[3] Díky tomu je vše navenek velmi prosté a intuitivní.

Data je možné také zadat pouze jednou a následně je používat při rozdílných konfiguracích všech editorů, např. pro programování PLC nebo projektování na HMI.

Aktuální verze Step 7 V11 a WinCC V11 (dříve samostatné aplikace) jsou dnes součástí prostředí TIA Portal. Step 7 je programovací nástroj pro PLC (především pro modely Siemens Simatic). Díky WinCC V11 můžeme zase v prostředí TIA Portal realizovat všechny úlohy pro HMI. „WinCC V11 nabízí sjednocenou platformu pro vývoj uživatelských řešení od jednoduchých úloh HMI u stroje (panely) až po rozsáhlé SCADA aplikace v podnikových velínech.“[3]

3.1 Programovací prostředky v TIA Portal

TIA umožňuje využívat dvě pracovní rozhraní. Portal view a Project view. První obsahuje základní funkce, prostředky pro založení nového projektu, správu sítě, či identifikaci připojených zařízení. Z Portal view můžeme následně přejít do Project view, které již přímo nabízí prostředky pro tvorbu, správu jednotlivých programů/aplikací a to jak pro PLC tak i HMI.

Vzhledem k jednoduchosti a intuitivnosti vývojového softwaru TIA Portal, se nebudu v této práci detailněji zabývat jednotlivými kroky nastavování IP adres pro jednotlivé porty připojených modulů, založení projektu, nahrání knihoven, výběru programovacího jazyka, apod. Většina těchto věcí je automaticky, a nebo nás velmi lehce navede softwarová nápověda. Důležitější pro cíl mé práce jsou možnosti programování. Samotný TIA Portal pro model S7-1500 umožňuje využít 5 různých programovacích jazyků, které můžeme mezi sebou i různě kombinovat, nebo vytvořit funkční bloky a ty využít zase v jiném jazyce. Tyto programovací prostředky jsou:

1. STL (Statement List)
2. SCL (Structured Control Language)
3. LAD (Ladder Diagram)
4. FBD (Function Block Diagram)
5. GRAPH

Abych dosáhl co největší názornosti, je u popisu každého z těchto jazyků naprogramovaná stejná jednoduchá funkce. Touto funkcí je obyčejný samodržný kontakt.

3.1.1 STL (Statement List)

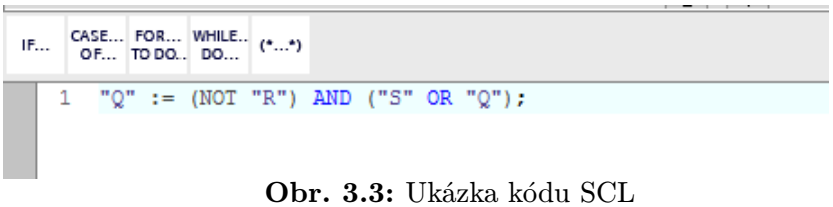
Základní způsob pro programování PLC je již dlouhou dobu využívaný jazyk STL. Je to textový zápis programu, jenž je tvořen v mnemotechnických instrukcích CPU. Proto je velmi blízký instrukční sadě mikroprocesoru. Využívají ho spíše zkušení PLC programátoři, kvůli náročnosti vycházející i z nedostatečné vizuální podoby. Díky větší podobnosti se samotným programem CPU („assembler“ pro PLC) jsou ale rychlosti programu při použití STL největší, a ve výsledku se ostatní programovací prostředky stejně konvertují právě do STL. Program v STL lze převést v softwaru po napsání na jazyky FBD a LAD.

```
1     A     "R"                    %M0.0
2     NOT
3     A(
4     O     "S"                    %M0.1
5     O     "Q"                    %M0.2
6     )
7     =     "Q"                    %M0.2
```

Obr. 3.2: Ukázka kódu STL

3.1.2 SCL (Structured Control Language)

Jak už samotný název napovídá, jedná se o vyšší strukturovaný jazyk. Nápadně připomíná moderní jazyky typu Pascal, C++, apod. Je určen pro složitější algoritmy, kde musíme využívat dlouhé výpočty. Musíme ale zároveň počítat s větším nárokem na paměť dat a programu a tomu uzpůsobit např. i druh paměti programu (FLASH/RAM). Tento programovací jazyk celkem názorně vystihuje moderní trend vývoje PLC a jeho přiblížení se normálnímu programování. Kromě prvků jazyka na vysoké úrovni obsahuje SCL také prvky typické pro PLC, jako jsou vstupy, výstupy, časovače, bitová paměť, blok volání nějaké funkce, atd.



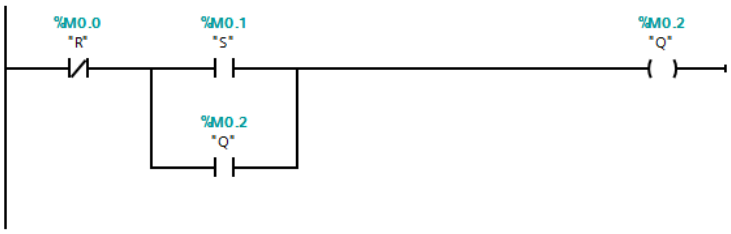
Obr. 3.3: Ukázka kódu SCL

3.1.3 LAD (Ladder Diagram)

Často používaná metoda programování PLC je založena na použití diagramu v liniovém provedení. Psaní programu pak odpovídá kreslení spínacího obvodu. Vychází z logických sítí a snaží se tedy přejímat jejich symboly (relé, tlačítka, časová relé, apod.). Schéma „žebříku“, jak se jinak tomuto jazyku také říká, se skládá ze dvou svislých čar představujících silové lišty. Obvody jsou spojené vodorovnými čárami (jako příčky žebříku)

mezi těmito dvěma silovými lištami. Při kreslení schématu podle LAD jsou přijaty určité konvence které musíme dodržet:

- Vertikální čáry diagramu představují silové lišty mezi kterými jsou obvody připojeny. Směr toku proudu se uvažuje z levé vertikální lišty přes příčku.
- Každá příčka v „žebříku“ definuje jednu operaci v řídicím procesu.
- Schéma žebříku je čteno zleva doprava a shora dolů. Horní příčka se přečte zleva doprava. Pak se přečte druhý krok pod ním, taktéž zleva doprava a tak dále.

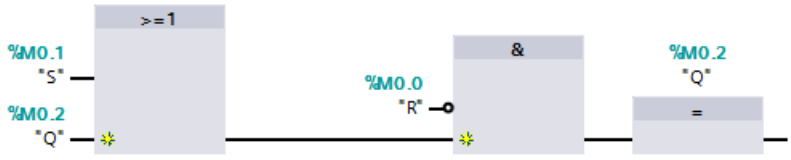


Obr. 3.4: Ukázka programování v LAD

3.1.4 FBD (Function Block Diagram)

Funkční blokový diagram (FBD) je oficiální a široce požívaný druh grafického jazyka pro návrh automatického řízení. Může popisovat funkci mezi vstupními proměnnými a výstupními proměnnými. Diagram funkčních bloků se snadno učí a poskytuje mnoho možností. Funkce je popsána jako sada elementárních bloků. Vstupní a výstupní proměnné jsou připojeny k blokům pomocí spojovacích linek.

FBD je zásadní jazykem pro všechny PLC programátory. Je to skvělý způsob, jak implementovat vše od logiky po časovače, PID regulátory atd. V softwarovém inženýrství jsou dobře známy blokové diagramy, které znázorňují funkci algoritmu. FBD se trochu od tohoto způsobu liší. Nabízí způsob, jak vkládat funkce psané s mnoha řádky kódu do jednotlivých a dále použitelných bloků. Tím je poté můžeme jednoduše propojit a vytvořit větší PLC program. Určité části většiny PLC programů jsou napsány v jazyce FBD, protože i když bychom psali funkce ve strukturovaném textu, tak stále musíme většinu času následně spojovat vytvořené funkce.



Obr. 3.5: Ukázka programování v FBD

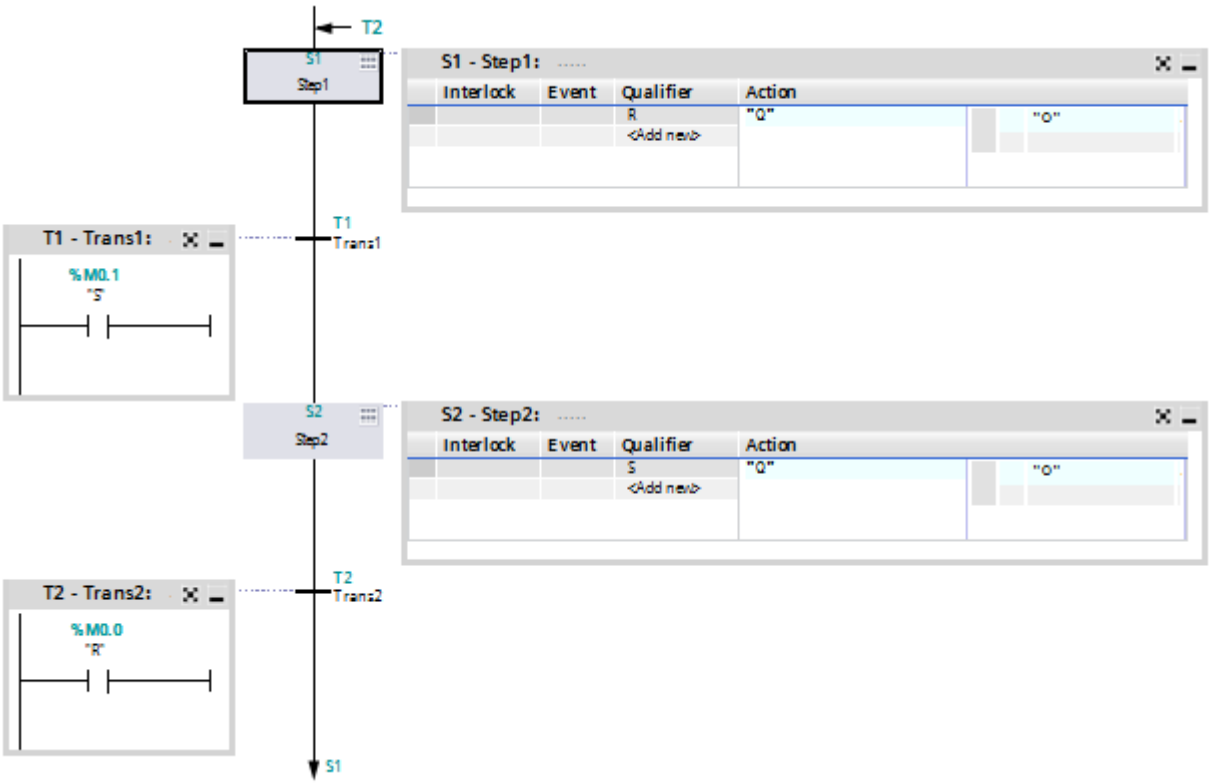
3.1.5 GRAPH

Programovací jazyk GRAPH je jazyk založený na vývojovém diagramu používaném pro programování a provádění sekvenčních řídicích aplikací pro procesory PLC Siemens. Při sekvenčním řízení provádí stroj řadu kroků, jako jsou přechody, podmínky a akce. Dobře známé prvky např. u vizualizace algoritmu při programování ve vyšších jazycích.

Tento programovací jazyk se stává stále populárnější díky jeho přehlednosti. Výsledný projekt vytvořený v GRAPH by měl být snadno přehledný a srozumitelný technologovi, který má na starosti nějaký výrobní proces, a to takové vyjádření kódu umožňuje.

GRAPH je založen na pravidlech a využívá stochastický algoritmus. Ten má rozdíl oproti běžně používaném deterministickém algoritmu v tom, že v některých krocích může volit z několika možností následujících kroků. To nám pomáhá pracovat s jazykem velké abstrakce. Jádro tohoto jazyka se skládá ze čtyř po sobě následujících kroků:[7]

- Jednostupňové složení aplikace a vytvoření množiny podmíněných pravidel GRAPH.
- Postupné skládání jednotlivých aplikací.
- Větvení námi vytvořeného vývojového diagramu.
- Zavedení iterace, tedy cyklu našeho programu.



Obr. 3.6: Ukázka programování GRAPH

3.2 HMI

HMI (Human Machine Interface), jak už samotný překlad napovídá, představuje prostředky pro zobrazení a předání informace o stavu zařízení obsluze nebo operátorovi a zároveň umožňuje uživatelům ovládání stroje a zadávání hodnot. Toto grafické rozhraní tedy obsahuje integrovaný hardware a software určený k monitorování a řízení provozu strojů a souvisejících zařízení v průmyslových prostředích. HMI obsahuje elektronické komponenty pro signalizaci a řízení automatizačních systémů.

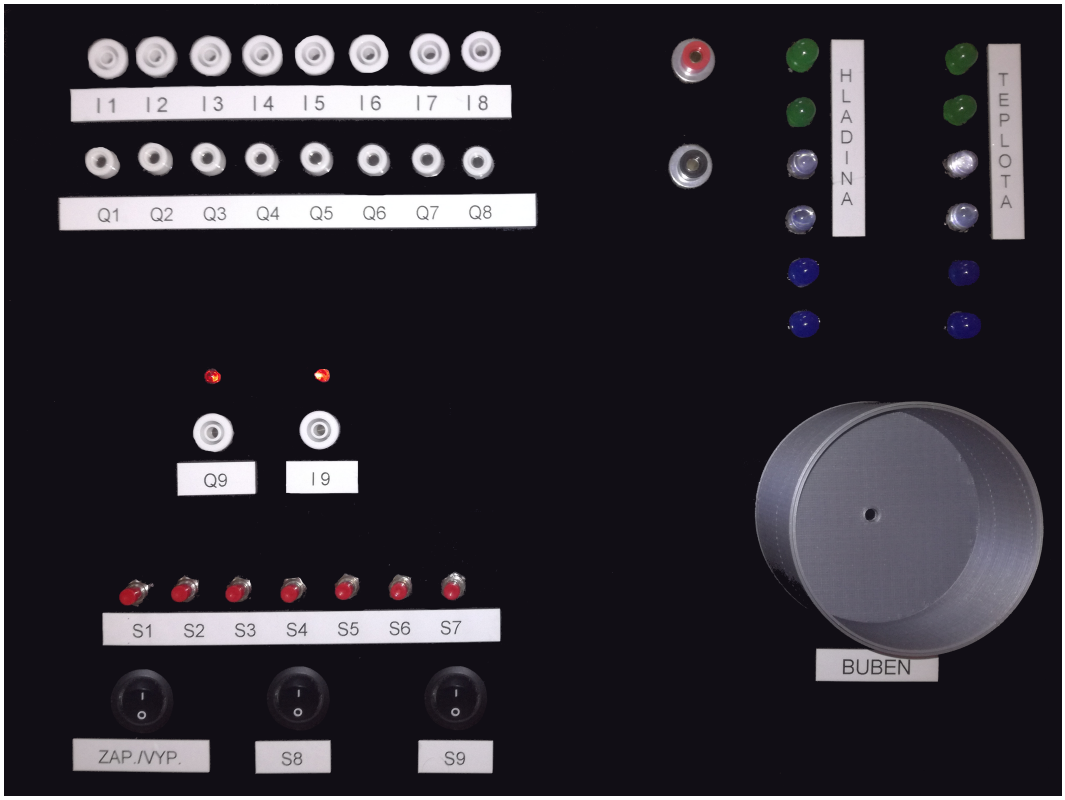
Většina používaných HMI připojených k PLC převádějí data z průmyslových řídicích systémů do vizuální reprezentace čitelné pro běžnou obsluhu. To vše je použito na dotykové obrazovce. Prostřednictvím HMI může obsluha vidět schémata systémů a zapnout nebo vypnout například spínače, čerpadla, nebo zvýšit či snížit teplotu. Možnosti jsou minimálně limitovány. HMI jsou obvykle nasazovány na počítačích se systémem Windows a komunikují s programovatelnými logickými řadiči. TIA Portal umožňuje navrhnout vzhled výsledné plochy a zároveň ho spojit s určitými následujícími procesy, ať už v samotném HMI a nebo následně v řízení procesu PLC. Po tomto návrhu stačí výsledný projekt nahrát, nejčastěji přes komunikační rozhraní Ethernet/Profinet. Takový způsob propojení je použit i u panelů, které jsem v této práci používal.

Obr. 3.7: Jedna z možností nastavení pracovní plochy HMI |Převzato z [3]|

4

Návrh a realizace přípravku pro výuku logického řízení

Cílem mé práce byl návrh a realizace přípravku pro mechatroniku, a jeho následného využití v praxi, resp. při výuce automatizace na střední škole. Názorné přípravky, kde studenti mají možnost správně pochopit chod logického automatu a jeho následného využití v praxi, je v komerční sféře nedostatek, a když už se nějaká firma zabývá jejich vývojem a prodejem, tak se jedná ve výsledku o nesmyslně velké částky, které jsou pro skromný rozpočet dnešních středních škol velmi bolestné.



Obr. 4.1: Přípravek pro výuku mechatroniky(simulátor pračky)

4.1 Návrh typu přípravku

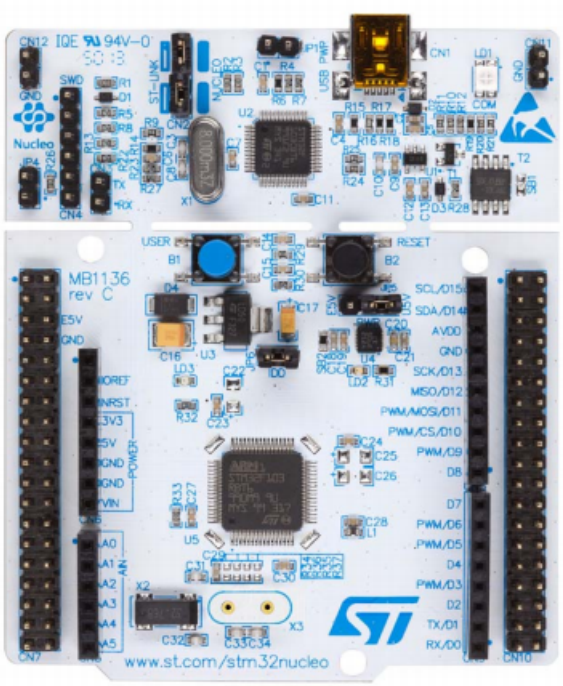
Při návrhu jsem se držel představy, aby výsledný přípravek simuloval chod nějakého skutečného systému, který by mohl být ovládán programovatelným logickým automatem. Pro dobrou přehlednost, vizuálně jednoduché zprostředkování pro studenty a následně možné využití v mnoha nastavení a módech, jsem po úvaze zvolil návrh simulátoru pračky. Jelikož se jedná pouze o simulátor a je důležitá názornost pro výuku, tak zde nebylo nutné uvažovat při návrhu všechny komponenty, které skutečná pračka obsahuje, a proto jsem je v mnoha případech pouze znázornil ekvivalentní grafickou reprezentací. Pro napouštění a ohřev vody jsem např. zvolil řadu LED, které znázorňují stav vodní hladiny nebo teploty vody. Pro různé druhy čidel jsem vzhledem k možnosti variabilnosti zvolil řadu kolébkových spínačů nebo tlačítek (OFF-ON-OFF), které vyžadují využití přídržných relé v jejich programu. Škála variant možných čidel (tlakové čidlo pro kontrolu zavření dvířek, tlakové čidlo pro kontrolu váhy prádla, apod.) naráží tedy pouze na představivost studentů, kteří daný přípravek budou využívat.

Tlačítka, která deska obsahuje, jsou značena symboly S1-S7 a možnost jejich využití je pouze na volbě uživatele. Kolébkové spínače jsou na desce tři. Dva pro různé možnosti definované uživatelem (S8, S9) a jeden pro zapnutí/vypnutí přípravku. Zdroje se využívá, abychom dosáhli ještě menších výrobních nákladů, společný s PLC Simatic. Napěťový zdroj PLC s hladinou 24 [V] a výkonem 192 [W]. Jelikož chyby jsou při výuce možné, je chráněn přípravek proti přepólování za zdířkami vstupu napájení. Tato ochrana je realizována obyčejným diodovým můstkem. Kvůli němu sice v přípravku přijdeme (kvůli napětí na diodě) o 0.7 [V]a tedy výsledné napětí v rozvodech přípravku je 23.3 [V] . Což není sice ideální stav, ale pro splnění logických napěťových hladin v použitém mikropočítači a následnými výstupy pro PLC, stačí více než dost, a proto pro ochranu proti přepólování, diodový můstek dobře poslouží.

4.2 Použití mikropočítače

Přípravek obsahuje mikropočítač, který je jádrem daného výrobku. Využil jsem programovatelnou desku STM32 Nucleo F411RE. Ta pracuje s procesorem o cortexovém jádru, resp. ARM Cortex-M4, což je druh procesoru RISC pracující na harvardské koncepci. Samotná deska obsahuje již také v základu několik časovačů, možné analogové nebo digitální vstupy/výstupy, a to je vše připojeno k paralelním branám. K těm jsou připojeny také externí konektory (Arduino Uno V3 a záhlaví ST morpho) pro rozšíření základní desky o různé shieldy s velkou škálou dalších možností. Podpora těchto připojení je v této práci využita. Deska zároveň nevyžaduje další samostatnou sondu pro programování a využívá již integrovanou sériovou sběrnici UART pro komunikaci s PC. ST-LINK / V2-1 debugger a zároveň paměť programu a dat již samotná deska také obsahuje.

Hlavní důvodem pro umístění mikropočítače v přípravku bylo jednoduché ovládání

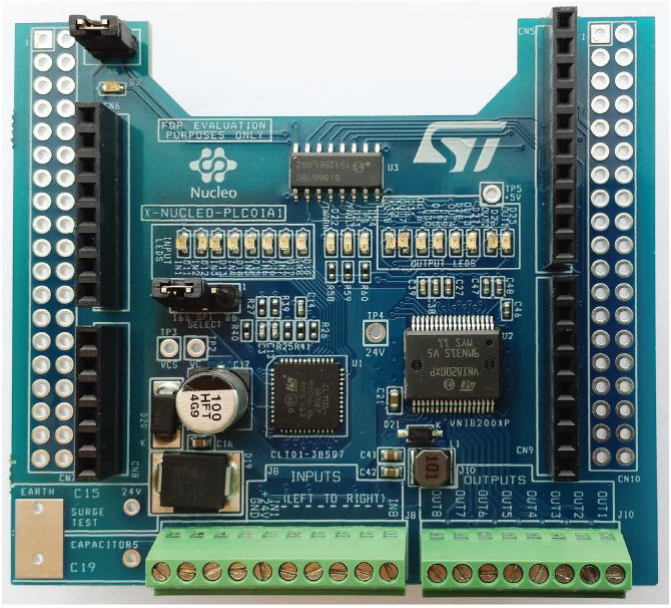


Obr. 4.2: STM32 Nucleo F411RE |Převzato z [8]|

navenek z PLC. Mým cílem bylo, aby po přivedení signálu na vstupy přípravku z výstupů PLC byla umožněna rovnou funkce daných komponent pračky. Např. po přivedení impulsu na vstup I1 by se začala zvedat vodní hladina, impuls na vstup I4 by roztočil motor pomalu po směru hodinových ručiček atd. Všechny tyto možné procesy, které jsou popsány blíže v následujících kapitolách, ovládá právě mikropočítač, který po přivedení impulsů na jeho dopředu definované vstupní porty spustí daný proces.

Jelikož ale napěťové hladiny PLC a mikropočitače jsou rozdílné (24 [V] pro PLC a 5[V] pro mikropočítač), tak prvním problémem bylo vyřešení této nesymetrie, aby mohly mezi sebou dané systémy komunikovat. Jednou z možností by bylo na každý vstup nainstalovat DC-DC měnič a každý vstup takto samostatně měnit. To by ale bylo velmi drahé, pracné a vznikaly by obrovské ztráty, které by mohly vést až ke zničení přípravku. Řešením nakonec bylo nalezení integrovaného obvodu, který zvládá převést hodnotu napětí z PLC na hodnotu, kterou již mikropočítač akceptuje a nevede k jeho zničení. Tím obvodem je VNI8200XP, vyroben taktéž jako programovatelná deska od firmy ST. Následná komunikace z mikropočítače do PLC byla potřeba taky vyřešit. To obstaral zase integrovaný obvod CLT01-38SQ7 od ST. Oba tyto integrované obvody zároveň firma ST vyrábí na společné desce v kitu s Arduino konektory, kterými se můžeme připojit k základní vývojové desce F411RE.

Tento shield pracuje již s 24 [V] a to jak pro vstupy nebo výstupy, které můžeme přivádět rovnou z PLC, a následně signál, převedený na napěťovou úroveň mikropočítače, využít pro početní výkony programu.

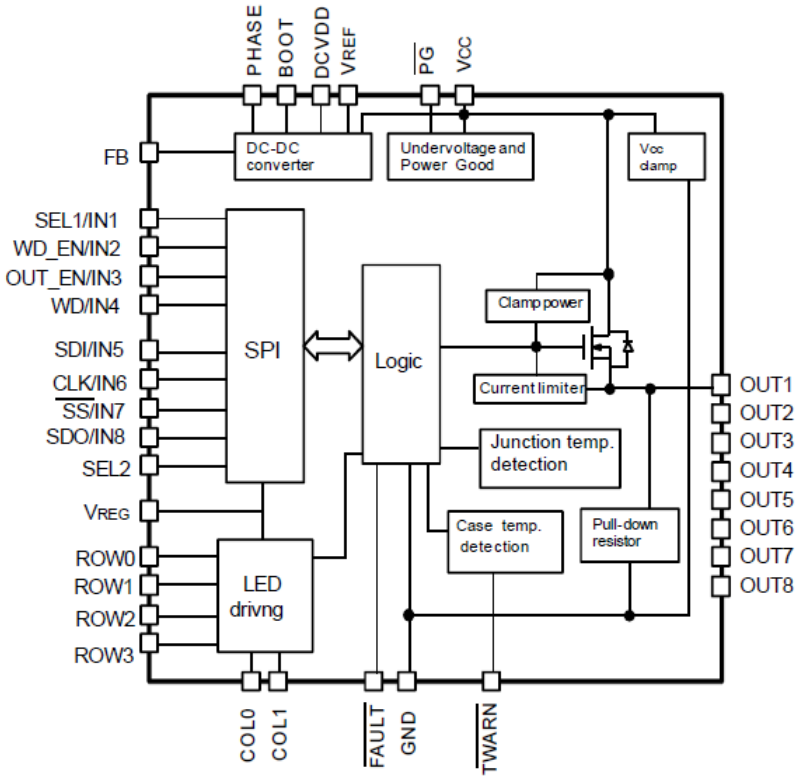


Obr. 4.3: Shield X-NUCLEO-PLC01A1 |Převzato z [8]|

4.2.1 Komunikace SPI a její realizace

Komunikace s VNI8200XP nebo s CLT01-38SQ7 může probíhat paralelně nebo po sériové sběrnici, ale zapojení na desce, která tyto integrované obvody obsahuje, je uzpůsobeno rovnou pro přenos sériovým kanálem. Používá se zde komunikace po sériové periferii SPI. Ta je velmi jednoduchá (obsahuje ji většina mikrořadičů) a spolehlivá. Využívá se především pro spojení s blízkými obvody, převážně na vzdálenosti centimetrů, což tato deska vzhledem ke vzdálenosti od procesoru splňuje. Nevýhodou u delších vzdáleností je potřeba většího množství vodičů a tedy z toho vycházející vyšší cena. Jednoduchost vyplývá především z jednoduché synchronizace. Není zde potřeba bitové nebo znakové synchronizace (jako např. u UART), ale hodinové impulsy přímo posíláme po samostatném vodiči ke všem podřízeným uzlům. Komunikace přes SPI se provádí v základu po třech společných vodičích. Jeden přenáší synchronizační impulsy SCK, po druhém se přenáší sériová vstupní data SDI a po třetím sériová výstupní data SDO. Ke generování synchronizačních impulsů můžeme využít jednotlivé bity zvolené paralelní brány. Samozřejmě programově ovládané.

Náš obvod obsahuje ještě další tři ovládací vodiče. Těmi se nesou signály OE (Output enable), WD (Watchdog) a SS (Slave select). Těmi nastavujeme umožnění vstupů nebo výstupů přenosu přes SPI. OE musí být neustále nastaveno na jedničku, abychom přenos na výstup VNI8200XP měli povolen. Signálem WD ovládáme obvod dané periferie nebo mikrořadiče, který ochraňuje před nechtěným zacyklením programu. Tento obvod nám dohlíží na správný chod našeho programu. Watchdog obsahuje nezávislý čítač, který v případě přetečení vyvolá reset periferie. Proto jsem při návrhu programu musel brát v úvahu rychlost tohoto čítače a nastavovat tento signál ve správných rozmezích, a to před a po vysílání signálu přes vodič pro přenášené data.



Obr. 4.4: Blokové schéma obvodu VNI8200XP |Převzato z [8]|

Signálem na SS vybíráme zpětnou komunikaci přes SPI a to buď s CLT01-38sQ7 a nebo VNI8200XP. Tyto dva obvody jsou totiž připojeny na stejnou sběrnici SDI a můžeme tedy volit mezi kontrolou výstupních bitů a nebo vstupem z externího zdroje (v našem případě z PLC portů).

Na Obr.4.4 můžeme vidět blokové schéma obvodu VNI8200XP, které dosti napoví o funkci komunikace přes SPI, transformace z 5 [V] na 24 [V] a také o různých ochranách, ať už samotného obvodu nebo ochran výstupů, jelikož se počítá s použitím tohoto obvodu třeba u výrobních linek, a tam by chyba např. vlivem hazardů mohla stát nemalé peníze. Tyto funkce jsou již v obvodu pevně nastavené a musel jsem s nimi počítat při návrhu programu pro daný mikrokontrolér. Tou první byla proudová ochrana, která je v obrázku značena blokem „current limiter“. Proudový omezovač je v praxi u elektrických nebo elektronických obvodů důležitý kvůli ochraně zátěže a i samotného obvodu před škodlivými účinky v důsledku zkratu nebo jiného problému, který by způsobil nadměrný proud.

Vůči přepětí je obvod na desce chráněn výborným způsobem, což můžeme vidět v datasheetu na CD přiloženém k této práci. Špičkovou hodnotu napětí zvládne bez zničení obvodu až 15 [kV], což se v průmyslu velice hodí, protože v důsledku přechodných jevů při regulování prvků s velkou indukčností, může k takovému chvilkovému výboji dojít velice snadno.

4.2.1.1 Ochrana výstupních dat

Ochrana správného přesunu dat na výstupy je zde řešena paritními bity. Výstupů na shieldu máme pouze osm, a proto by nám teoreticky stačila komunikace přes SPI pouze 8-bitová, ale kvůli ověření správnosti výsledků musíme do obvodu posílat 16-bitová data, která poslední čtyři bity používá jako už před chvilkou zmíněné paritní bity. Samotný obvod umožňuje komunikaci přes 8 nebo 16-bitů, a jaký mód zvolíme, se nastavuje registrem s přicházejícím signálem SEL1, který můžeme vidět v blokovém schématu. Jelikož ale zapojení tohoto obvodu na zkompletované desce již přivádí rovnou trvalou úroveň „1“ bez dalšího možného výběru, tak zde musíme tuto ochranu taktéž počítat v návrhu. Jinak bychom komunikaci nemohli nijak provést. Zde je výpočet těchto posledních paritních 4 bitů:

$$P0 = IN0 + IN1 + IN2 + IN3 + IN4 + IN5 + IN6 + IN7 \tag{4.1}$$

$$P1 = IN1 + IN3 + IN5 + IN7 \tag{4.2}$$

$$P2 = IN0 + IN2 + IN4 + IN6 \tag{4.3}$$

$$nP0 = \overline{P0} \tag{4.4}$$

MSB															LSB
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0					P2	P1	P0	nP0

Tab. 4.1: Pořadí a význam jednotlivých bitů při přenosu

Tyto paritní bity se musely nastavit funkcí v programu a pouze po jejich splnění obvod propustil vysílaný signál na výstupy. Pokud detekoval chybu při vysílání, tak rozsvítil signalizační LED, která má informovat, že v obvodu nebo samotné funkci došlo k chybě.

Paritní bity jsou redundantní bity, které přidáváme k nějakému vysílanému slovu, zde k původnímu 8-bitovému a nesou informaci o počtu jednotkových bitů ve slově. Paritními bity se dá pouze zjistit lichý počet chyb v přenášeném slově a sudý počet chyb přejde bez detekce, proto je v obvodě nastavena kontrola čtyřmi paritními bity, které nám mohou detekovat větší počet chyb v našem přenášeném původním slově.

Podle rovnic a tabulky umístění paritních bitů v přenášeném slově, jsem vytvořil v programu, který je nahraný v mikropočítači, funkci na tvorbu paritních bitů. Jelikož celý program je velice obsáhlý a není stěžejním zadáním mé práce, tak zde přikládám pouze ukázku této funkce a celý program lze nalézt na přiloženém CD.

Samotný program využívá spodní čtyři vysílané bity ke generování signálu, který se přivádí na LED znázorňující vodní hladinu a horní čtyři bity na řadu LED, jenž zase nahrazují termostat, kontrolující teplotu napuštěné vody. Tyto bity musejí tedy v nadcházejícím přenosu a úpravě patřičnou funkcí, pracovat nezávisle na sobě.

```
1   void NastaveniParitnichBitu(void)
2   {
3   % uložení generovaných signálů do proměnné udata
4   % vypočtení jednotlivých paritních bitů podle zadaných rovnic
5   % složení vypočtených paritních bitů a našeho vysílaného signálu
6   %-----------------------------------------------------------------
7
8       uint8_t udata = (j<<4)|n;
9       uint8_t P0, P1, P2, nP0, parityData, mojeData;
10
11      P0 = udata^(udata>>1);
12      P0 = P0 ^(P0 >> 2);
13      P0 = P0 ^(P0 >> 4);
14      P0 = P0 & 0x01;
15
16      P1 = udata^(udata>>2);
17      P1 = P1 ^ (P1 >> 4);
18
19      P2 = P1 & 0x01;
20
21      P1 = P1 & 0x02;
22      P1 = P1 >> 1;
23
24      nP0 = (~P0) & 0x01;
25
26      parityData = (P2<<3)|(P1<<2)|(P0<<1)|nP0;
27
28      n = n  & 0x0f;
29      j = j & 0x0f;
30      mojeData = (j<<4)|n;
31      b = (mojeData<<8)|parityData;
32  }
```

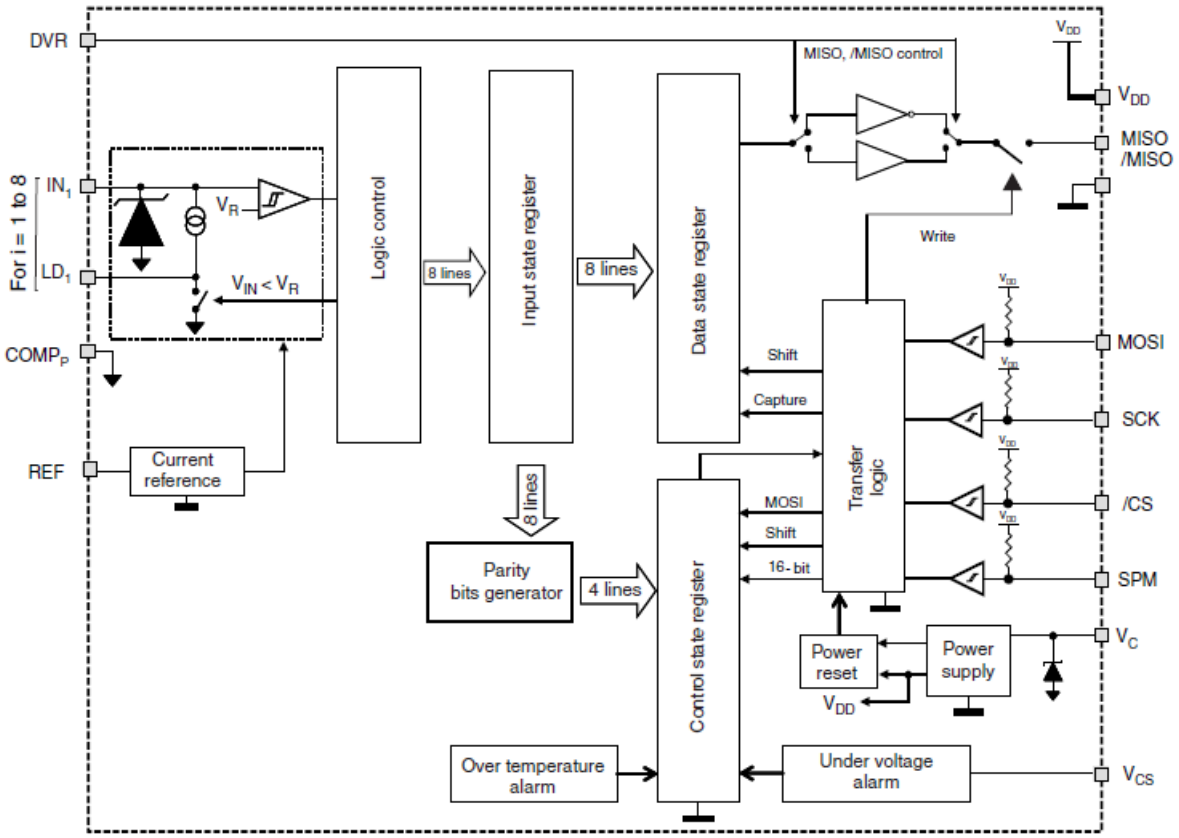
První část (spodní čtyři bity) je ve funkci nahrazena proměnnou „n“ a druhá proměnnou „j“. Tyto vytvořené proměnné z předcházejících funkcí v programu, jsou spojeny do proměnné „udata“. Následně podle předepsaných rovnic z těchto dat složíme všechny čtyři paritní bity. Ze spojení „n“ a „j“ a vzniklých paritních bitů vytváříme novou 16-bitovou proměnnou, kterou už můžeme posílat do obvodu. Zde hardwarové ochrany zabudované v VNI8200XP již propustí bez dalších problémů data z mikrořadiče na výstup této periferie.

4.2.1.2 Zpracování vstupních dat

Vstupní data do mikrokontroléru vstupují také v napěťových úrovních stanovených pro PLC. Pro správnou transformaci úrovní na možnosti mikrořadiče a posílání těchto signálů přes sériovou sběrnici, máme na desce již dříve zmíněný obvod CLT01-38sQ7.

Blokové schéma může opět nastínit práci jako u popisu předchozího obvodu, ale v opačném pořadí. Měníme z většího na menší napětí a zároveň paralelní vstupy převádíme na sériovou komunikaci.

Zde máme pro obvod také určité časové omezení a při synchronizaci musíme brát ohled na maximální možnou frekvenci pro VNI8200XP a CLT01-38sQ7. Pro první z jmenovaných obvodů je tato frekvence stanovena na 5 [MHz], kdežto u CLT je tato frekvence rovna 6.25 [MHz], a proto se i přes případnou větší rychlost u vstupního obvodu musíme řídit pomalejším systémem, jelikož rozvod synchronizačních hodin musí být kvůli synchronizaci obou obvodů stejný.



Obr. 4.5: Blokové schéma obvodu CLT01-38sQ7 |Převzato z [8]|

Vstupy jdou přes komparátor s hysterezí, který je zařazen kvůli lepší strmosti hran a tím nemohlo docházet k rozkmitání obvodu. Poté se tyto signály pošlou do výrobcem nezveřejněných logických obvodů, které zapíší data do vstupního stavového registru. Z toho se v paralelně umístěném obvodu, podle úrovní, vytvoří paritní bity a to ve formátu stejném, jako je popsáno u obvodu VNI8200XP. Ze vstupního registru se jednotlivé bity přesunou do stavového registru a ten nám následně podle synchronizačních pulsů posílá postupně data po sériovém kanálu. Samotný obvod má více funkcí, ale kvůli zapojení na desce, zde využíváme pouze tento mód. Jediné, co můžeme měnit v nastavení našeho obvodu, je přenos 8 nebo 16-bitů, ve kterých se může posílat ochrana, znovu v podobě paritních bitů. Tato možnost není v této práci využita, jelikož se jedná o učební přípravek, ve kterém již z důvodu dalších ochran nemůže dojít k závadě. Tyto ochrany jsou nastaveny hlavně programově a to především, aby nedošlo ke kolizi při vysílání protichůdných výstupních dat.

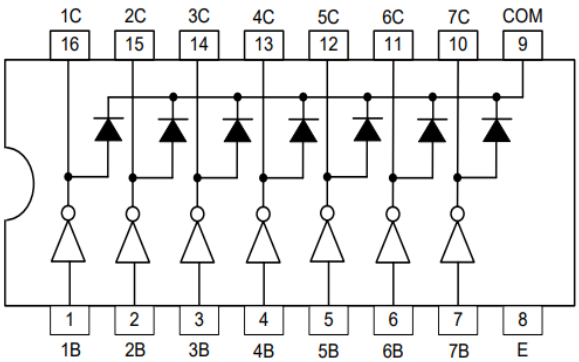
4.3 Ovládané prvky

Pro lepší názornost jsem zvolil řady LED na vyrobeném panelu, o kterých se zmiňuji v předchozím textu. Jsou ve větších velikostech (ø LED 10[mm]) a různých barvách. První dvě LED v obou řadách, ať už signalizující výšku hladiny nebo ohřev vody, jsou

modré barvy s napětím v propustném směru 3.5 [V]. Další dvě LED jsou oranžové barvy s napětím v propustném směru 1.9 [V] a poslední dvě mají zelenou barvu s prahovým napětím 3.2 [V]. Samotný mikropočítač má nastaven maximální proud z jednotlivých portů paralelních bran na 20 [mA], a to je pro všechny tři typy diod hodnota max. propustného proudu, proto bychom teoreticky nemuseli vůbec ochraňovat diody rezistory na omezení proud. Jelikož ale diody jsou připojeny pro ochranu a možnosti větších výkonů na výstupy našeho obvodu VNI8200XP, který dokáže dodávat do zátěže na jednotlivých výstupech až 100 [mA], tak byla potřeba zvolit vhodné rezistory k jednotlivým výstupům podle Ohmova zákona. Samozřejmě po odečtení prahových napětí z 24 [V] podle typů diod.

4.3.1 Krokový motor + driver

Důležité u našeho přípravku, pro alespoň malé autentické přiblížení skutečné pračce, je motor s bubnem pračky. Motor je krokový, ovládaný přes driver mikropočítačem. Je použit krokový motor 28-BYJ48. Ten můžeme nalézt v CD/DVD mechanikách, fotopastích a na mnoha dalších místech. Motor má 4 cívky s unipolárním uspořádáním a každá cívka je napájena napětím 5 [V], proto je relativně snadné ovládat ho základními mikroprocesory. Pro ochranu před možným zkratem, nadproudem nebo jiným nebezpečným stavem pro mikropočítač, je ovládání vedeno ještě přes driver připojený k motoru. Tento motor má krokový úhel 5,625 ° / 64, to znamená, že motor bude muset udělat 64 kroků k dokončení jedné otáčky a každý krok bude pokrývat 5,625 °, proto je úroveň řízení vysoká. Z napájení jednotlivých cívek (pouze 5 [V]) můžeme usoudit, že motor nezajistí vysoký točivý moment, což pro simulační účely bez výrazné zátěže na hřídeli, bohatě postačuje. To je tedy ve výsledku kompaktní snadno použitelný krokový motor s možností jemného nastavování.



Obr. 4.6: Schéma zapojení obvodu ULN2003A |Převzato z [9]|

Jednotlivé buzení cívek při každém kroku je nastaveno programově a vyvedeno z volných portů paralelní brány našeho mikropočítače. Tyto signály kvůli ochraně jsou vedeny nejdříve do driveru s použitým čipem ULN2003A. Jedná se o jednoduchý integrovaný obvod využívající Darlingtonovo zapojení tranzistorů pro zesílení výstupu do zátěže. Každý

výstup umožňuje proud až 500 [mA]. Diody na výstupech připojeny v závěrném směru (viz Obr. 4.6) jsou zahrnuty kvůli řízení induktivní zátěže, což motor je. Tento driver je schopen řídit širokou škálu zátěží včetně cívek, relé, stejnosměrných motorů, LED displeje, světel, apod.

V programu musíme řídit stanovené buzení cívek krokového motoru. Toto buzení obsahuje 8 kroků a vychází z datasheetu motoru. Jednotlivé kroky nastavení v obsluze buzení jsou vytvořeny v podprogramu, který voláme. Zpoždění je v programu vytvořeno integrovaným časovačem. Zde je krátká funkce jednoho cyklu buzení do driveru, které již využívá předem vytvořené kroky v podprogramu a zpoždění vytvořené časovačem.

```
1  void rotacePoSmeru(uint32_t pom6) {
2    krok1();
3    zpozdeniKroku(pom6);
4    krok2();
5    zpozdeniKroku(pom6);
6    krok3();
7    zpozdeniKroku(pom6);
8    krok4();
9    zpozdeniKroku(pom6);
10   krok5();
11   zpozdeniKroku(pom6);
12   krok6();
13   zpozdeniKroku(pom6);
14   krok7();
15   zpozdeniKroku(pom6);
16   krok8();
17   zpozdeniKroku(pom6);
18 }
```

Touto funkcí způsobíme směr otáčení našeho motoru po směru hodinových ručiček. Pokud chceme (v programu je též využito) točení motoru proti směru hodinových ručiček, stačí přehodit pořadí námi definovaných kroků od 8 a postupně snižovat. Vstupní proměnná „pom6“ je zavedena pro změnu rychlosti jednoho kroku motoru. Tato změna se ovládá v hlavním programu.

4.4 Výsledný přípravek

Přípravek umožňuje dobrou názornost a přehlednost i pro využití PLC programů od naprostých začátečníků, což bylo i cílem mé práce. Deska má 8 vstupů značených I1-I8, které vedou na porty mikropočítače. Jednotlivé funkce, které ovládají signály na přivedené tyto porty, jsou definovány v Tab. 4.2 . Dále máme ještě jeden vstup I9, jenž je zapojen na diodu s rezistorem k zemi, pro možné využití jako např. signalizace začátku programu. Výstupů spojených s napájením a některým ze zmíněných spínačů je na naší desce 9 a jsou značeny S1-S9. Poslední S9 má ještě taktéž pro možnou signalizaci zapojenou diodu v sérii.

Při změně ohřevu nebo výšky hladiny (signalizováno LED), je nastavená změna vždy o jednu úroveň výše nebo níže spuštěna po přibližně 6.875 [s]. Toto číslo vychází z dostatečně pomalého náběhu řad LED a snadného ovládání časovačem v obvodu.

Důležitým prvkem na desce jsou dva barevné výstupy (červený, černý), které jsou zapojeny paralelně s polovinou vodní hladiny a max. výškou hladiny. To může mít využití,

Vstup na desce	Vyvolaná funkce v mikropočítači
I1	Zvyšování vodní hladiny
I2	Snižování vodní hladiny
I3	Ohřev vody
I4	Snižování teploty vody
I5	Otáčení motoru po směru hodinových ručiček(pomalu)
I6	Otáčení motoru po směru hodinových ručiček na maximální rychlost
I7	Otáčení motoru proti směru hodinových ručiček(pomalu)
I8	Otáčení motoru proti směru hodinových ručiček na maximální rychlost

Tab. 4.2: Funkce při vstupu na jednotlivé porty

a snažím se to v další kapitole ukázat, při dvoustavovém ovládání. Možnost si vytvořit dvoustavovou regulaci v programu PLC ze zpětné vazby z mikrokontroléru, je veliké přiblížení řízení v praxi.

Samotné napájení našeho mikropočítače je vedeno ze svorek přicházejících ze zdroje společného i pro PLC a měněno přes Stepdown DC-DC měnič, který je nastaven na výstupní napětí 5 [V] pro správný chod CMOS součástek v našem mikrořadiči. Změna napětí z 24 [V] na 5 [V] by byla možná i stabilizátorem napětí např. typicky používaným obvodem 7805. Po spočítání ztrátových výkonů, které by se tímto stabilizátorem promarnily (řádově jednotky Watt), jsem i přes nevýhodu vyšší ceny, zvolil DC-DC menič. Tento měnič využívá integrovaný obvod LM2596 pro regulaci napětí s vhodně připojenými dalšími součástkami jako kondenzátory, nebo trimrem pro jemné nastavení výstupní hodnoty napětí. Důležité hodnoty pro napěťový měnič můžeme vidět v Tab. 4.3 .

Parametry	Hodnoty
Vstupní napětí	4,5 [V] - 53 [V]
Výstupní napětí	3 [V] - 35 [V]
Výstupní proud	3 [A] (max)
Účinnost přeměny	92 procent
Výstupní zvlnění	max. 30 [mV]
Spínací frekvence	150 [kHz]
Provozní teplota	-45 [°C] - 85 [°C]
Velikost	43 [mm] * 21 [mm] * 14 [mm] (L * W * H)

Tab. 4.3: Parametry Stepdown měniče s obvodem LM2596

5

Vzorové úlohy v TIA Portal na realizovaném přípravku

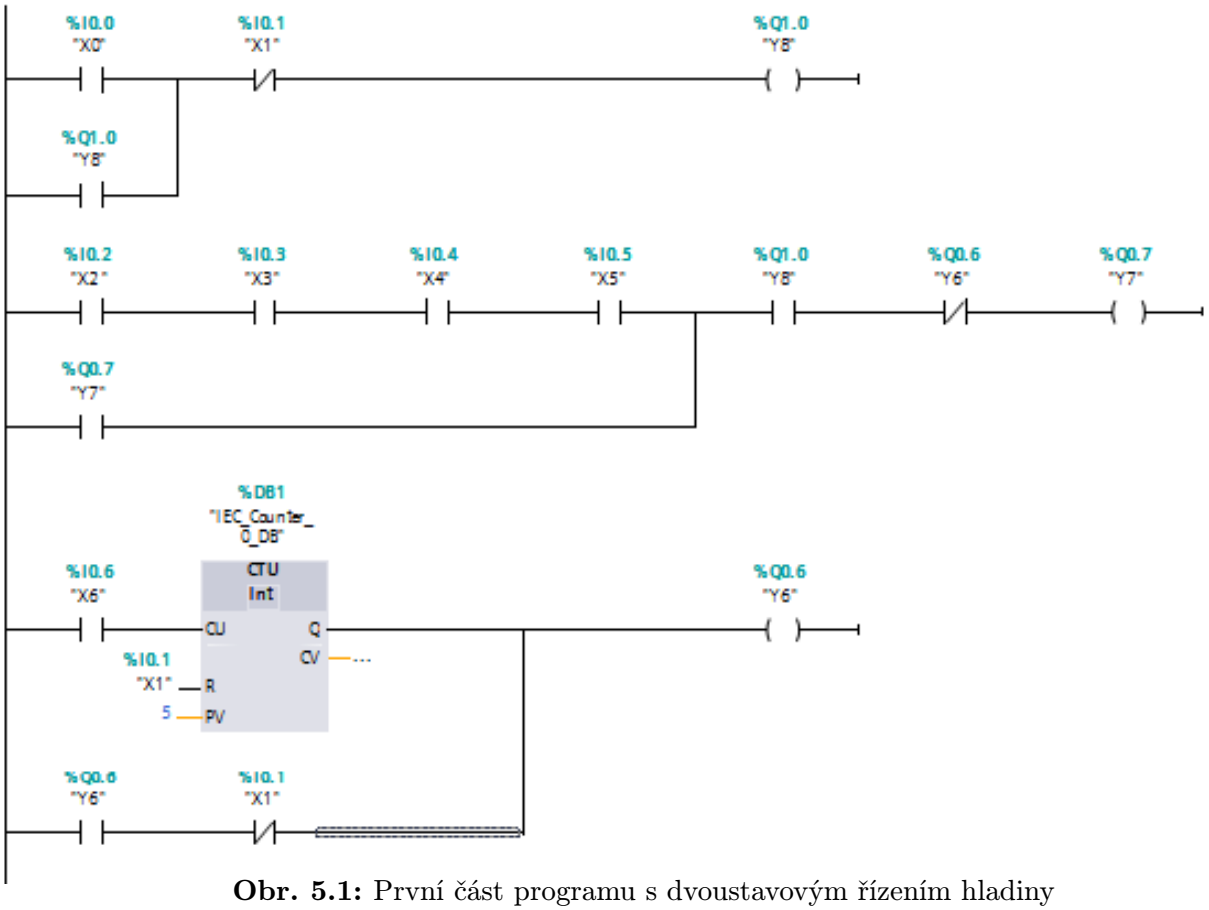
Kvůli demonstraci práce na výsledném přípravku, jsem vytvořil dvě vzorové úlohy v TIA Portal. Tyto programy jsou v programovacím jazyce LAD. Samotný přípravek umožňuje nespočet dalších možných nastavení oproti zde popsaným úlohám a výběr jednotlivých vstupů z tlačítek nebo pořadí výstupů z PLC jsou vybrány pouze pro demonstrativní účely. Při výuce je toto všechno na volbě uživatele. Tabulku 5.1 bereme v úvahu u obou vzorových úloh.

Vstupy PLC	Výstupy přípravku	Simulační úloha daného vstupu
X0	S1	START
X1	S2	STOP
X2	S3	Zavření dvířek
X3	S4	Kontrola povolené váhy
X4	S8	Nasypaný prášek v nádobce
X5	S9	Nalitá aviváž v nádobce
X6	Červený výstup	Plná hladina
X7	Černý výstup	Polovina hladiny

Tab. 5.1: Spojení výstupů na desce se vstupy PLC a jejich možná simulace skutečných čidel

5.1 Prací cyklus s dvoustavovým řízením hladiny

V této úloze jsem se zaměřil na možnost využití zpětné vazby z mikrokontroléru. Využíváme uv{čidla, která jsme si definovali v Tab. 5.1 . Se začátkem jejich sepnutí (samozřejmě kromě STOP) se spustí prací cyklus. Začne se pomalu zvyšovat vodní hladina, zvyšovat teplota a zároveň otáčet buben pračky po směru hodinových ručiček. V Obr. 5.1 můžeme vidět prvotní nastavení našeho programu.



Obr. 5.1: První část programu s dvoustavovým řízením hladiny

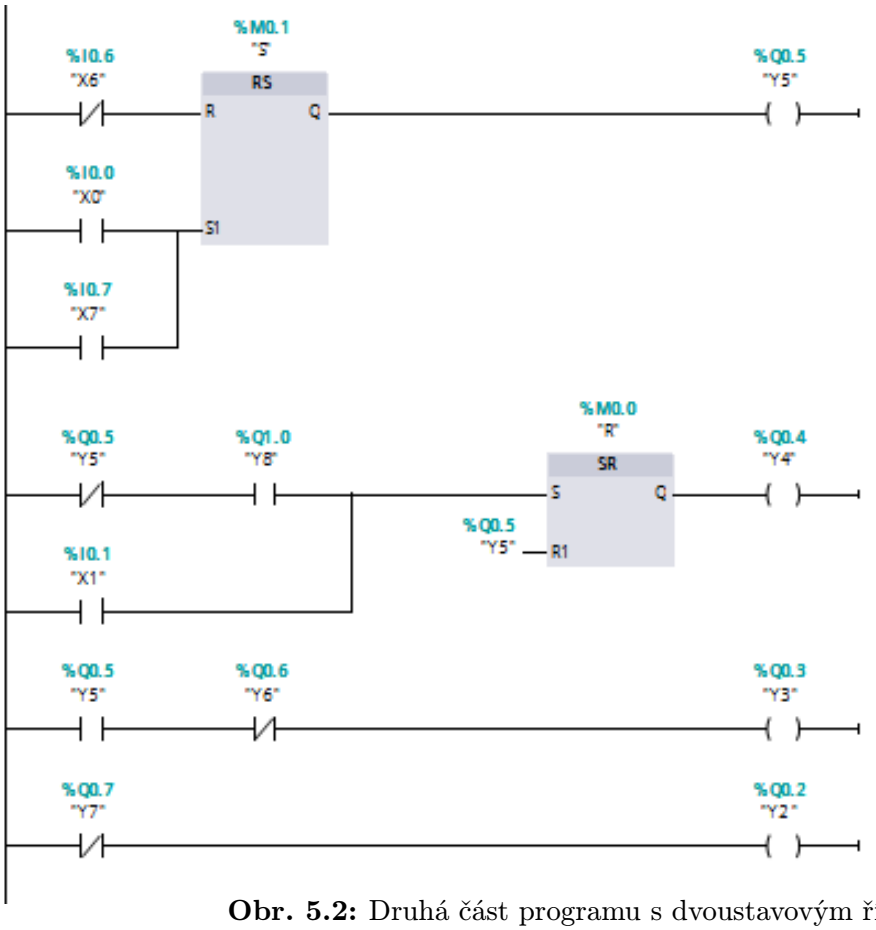
Je zde vytvořeno přídržné relé pro start. To se rozpojí při zmáčknutí tlačítka STOP, které je negované v sérii. Dále jsem program rozšířil o další přídržné relé, které se zapne po souběžném nastavení do jedničky všech čidel potřebných pro začátek programu (viz. Tab. 5.1) a vypne se při přetečení čítače nebo stlačením tlačítka STOP.

Čítač nám nastavuje délku našeho pracího cyklu podle toho, jakou zvolíme v programu jeho hodnotu pro přetečení. Počítá hrany z čidla upozorňujícího na max. vodní hladiny a po přetečení čítače se celý program vypne a vodní hladina a teplota klesnou do nuly. Použil jsem čítač čítající nahoru, reagující na náběžnou hranu.

Na Obr. 5.2 je zbytek obslužného programu pro PLC. Jsou zde klopné obvody RS a SR. Odlišnosti těchto dvou klopných obvodů jsou pouze v tom, jestli je prioritní SET nebo RESET. Ty jsou zde umístěny pro nastavení výstupů z PLC pro ovládání výšky vodní hladiny a tím mohly lépe simulovat skutečný chod pračky. Priority jejich vstupů jsou nastaveny s ohledem na to, že havarijní tlačítko STOP musí mít vždy přednost před jakýmkoliv jiným vstupem.

Při prvním náběhu, vodní hladina stoupne až na max. výšku. Při dosažení této hodnoty, začne hladina klesat na poloviční výšku, a po dosažení této výšky hladiny zase stoupat do maxima. Tento cyklus se opakuje pětkrát a po skončení toho cyklu se postupně začne řada LED, určených pro simulaci vodní hladiny, zhasínat. V celém běhu

programu se motor neustále točí jedním směrem za konstantní rychlosti a teplota vody se zastaví na maximu. Po skončení se otáčky motoru během minimálního možného času sníží až na nulu a teplota vody se s nastaveným časovým krokem též začne snižovat až k nule.



Obr. 5.2: Druhá část programu s dvoustavovým řízením hladiny

Pro funkčnost simulace s daným přípravkem, je důležité správné propojení PLC výstupů se vstupy desky, které vedou na porty mikrořadiče. Podle Tab. 4.2 uvedené v předchozí kapitole, můžeme udělat správné propojení těchto vývodů. Toto propojení máme uvedeno v následující tabulce.

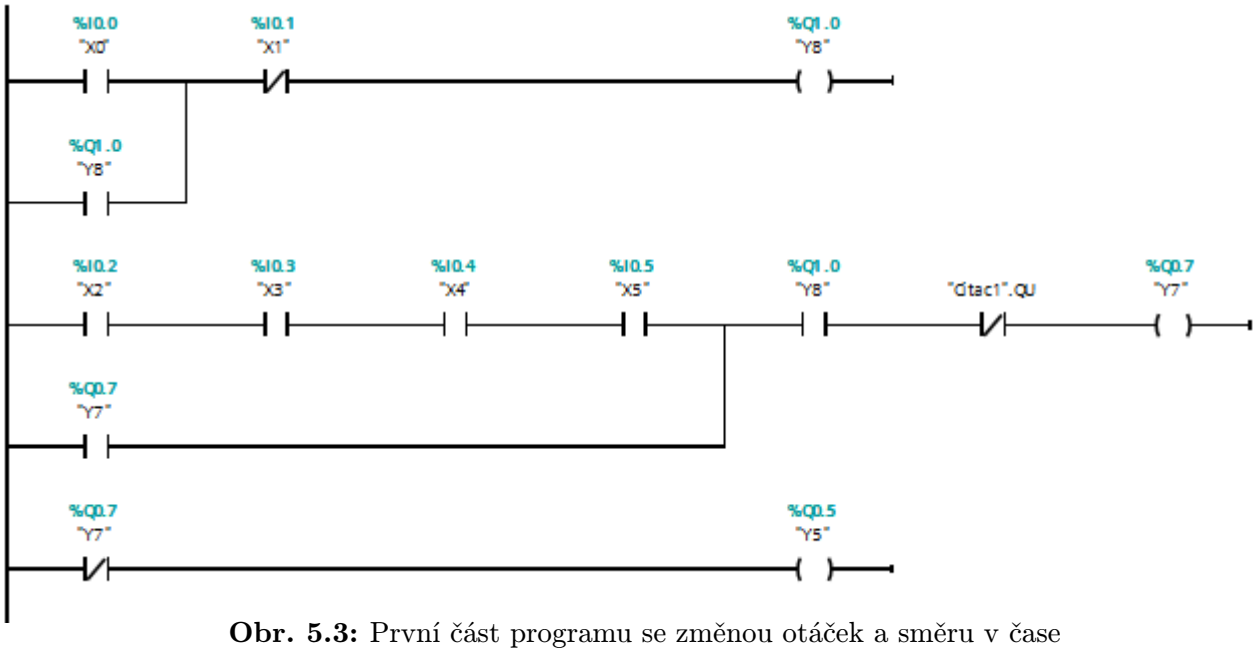
Vstup na desce	Značení daného výstup u PLC v programu
I6 (otáčení motoru)	Y7
I3 (ohřev)	Y7
I4 (snižování teploty)	Y2
I1 (napouštění vody)	Y3
I2 (vypouštění vody)	Y4

Tab. 5.2: Spojení jednotlivých výstupů PLC se vstupy mikrořadiče(1.úloha)

5.2 Prací cyklus se změnou otáček a směru v čase

Ve druhé úloze jsem si kladl za cíl využít v programu časovače. V této úloze časovač TON, který nahrazuje v programu funkci časového relé se zpožděným přítahem. Ty se v praxi společně s TOF, velmi hojně využívají.

Tato úloha se ještě více přibližuje skutečnému chodu pračky a dala by se ještě rozšířit o zpětnou vazbu a jejího následného vyhodnocování pro tvorbu dvoustavové regulace, využité v předchozí úloze. Použijeme navíc všechny vstupy do mikrořadiče.



Obr. 5.3: První část programu se změnou otáček a směru v čase

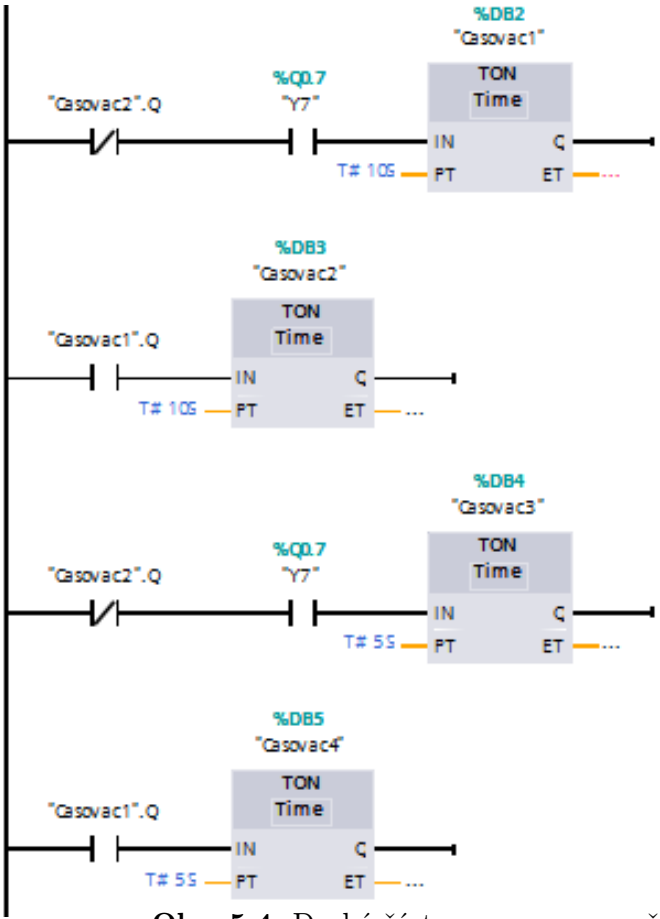
Začátek programu je téměř totožný s minulou úlohou. Musíme si nastavit přídržné relé pro START (zapnutí jedním tlačítkem a po rozepnutí, si svůj stav nadále relé drží), které rozepne pouze STOP zapojený sériově v negaci. Toto přídržné relé následně využijeme pro tvorbu dalšího přídržného relé a to tím, že k němu v sérii přidáme námi definované „čidla“. V této části využíváme stejně jako v předchozím programu výstup z čítače, kdy po přetečení se rozpojí „relé“ Y7.

Čítač zde má svůj význam i při použití časovačů v další části programu. Zde jsem vytvořil cykly pro změny chodu motoru, a tak po dokončení tohoto jednoho cyklu se nám přičte, náběžnou hranou signálu, hodnota v našem čítači a to do doby, než čítač dosáhne námi definovaného čísla. Touto hodnotou můžeme, stejně jako v předchozí úloze, určit délku celého pracího cyklu.

Dále vytvoříme výstupy pro zvyšování teploty a vodní hladiny. Zde, jak jsem již před chvilkou avizoval, nepoužíváme dvoustavou regulaci pro změnu hladiny, a proto přirozeně dojede v našem programu hodnota na řadě LED, graficky znázorňující vodní hladinu, do maxima a to samé bude i pro signalizaci ohřevu vody. Zde setrvá do konce programu, kde vyšleme signály na porty, které nám zapnou funkci pro snížení těchto řad v mikrořadiči.

Časovače, které jsou v této úloze hojně využity, představují významné instrukce pro načasování sledu vykonávaných částí programu. V našem případě změn chodu motoru.

První časovač se spustí již při náběhu programu a motor se tehdy začne točit malou rychlostí po směru hodinových ručiček. Třetí časovač, který se spustí také po začátku programu, „jedničkovou“ hodnotu na výstup vydá po pěti sekundách. Tím se přepne signál z předchozího portu na port, který spustí rychlé otáčky motoru ve stejném směru. Po deseti sekundách první čítač přesune jedničku na jeho výstup a tím zapne chod druhého a čtvrtého časovače. Ty jsou nastavené stejným způsobem jako první dva zmiňované. Tím dosáhneme nejdříve změny směru otáčení při pomalé rychlosti, a po 15 sekundách od začátku tohoto cyklu se motor ještě zrychlí ve stejném směru otáčení. Po dopočítání časovače 2, ke kterému dojde po 20 sekundách od začátku tohoto cyklu, se přičte hodnota na čítač a všechny 4 časovače se v tento moment resetují. Díky tomu cyklus může začít zase od začátku.



Obr. 5.4: Druhá část programu se změnou otáček a směru v čase

Stejně jako ve skutečné pračce, kdy se při různých fázích praní mění směr otáčení a rychlost motoru, tak pro demonstraci a přiblížení se věrohodně skutečnému systému, jsem zvolil návrh této úlohy.

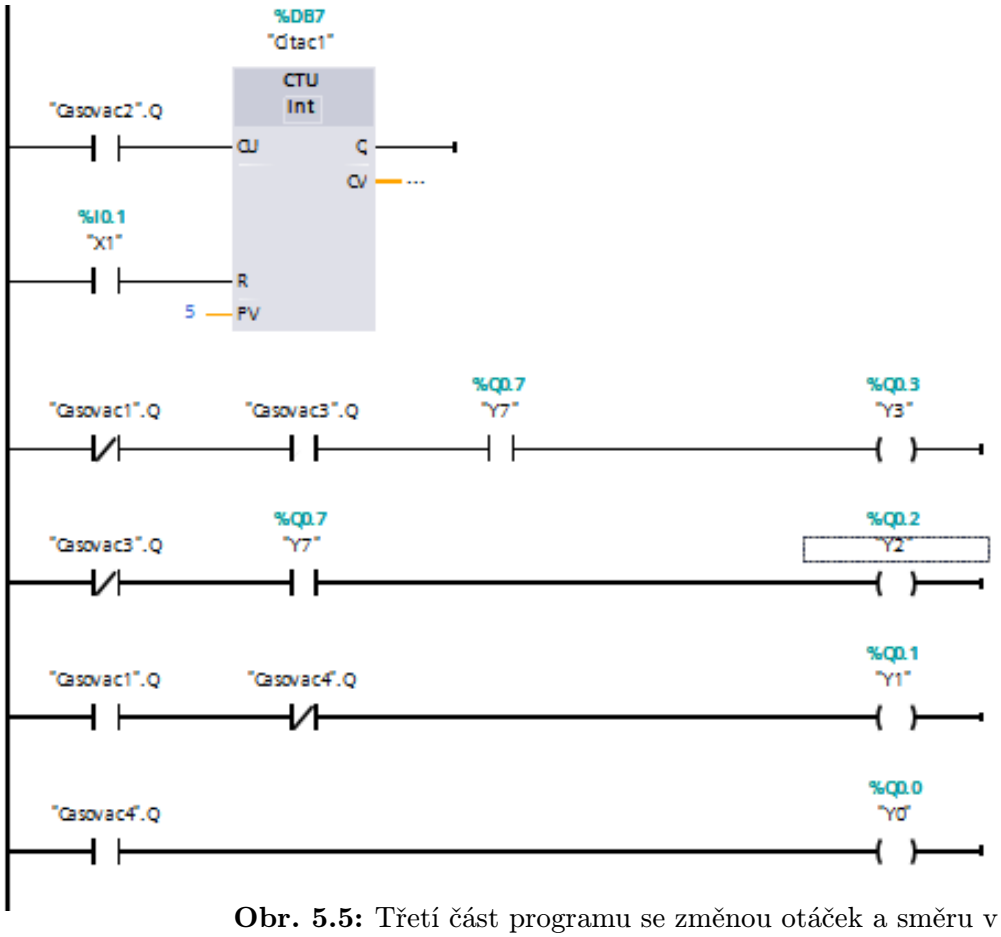
Dalším důležitým bodem by bylo správné zapojení vstupů/výstupů vytvořeného přípravku s PLC Simatic. Nastavení vstupů PLC se spínači na desce, které mají znázorňovat

čidla ve skutečném systému, je shodné jako v předchozí úloze a můžeme tedy toto zapojení provést podle Tab. 5.1 .

Vstup na desce	Značení daného výstup z PLC v programu
I5 (pomalé otáčení motoru vpravo)	Y2
I5 (rychlé otáčení motoru vpravo)	Y3
I5 (pomalé otáčení motoru vlevo)	Y1
I5 (rychlé otáčení motoru vlevo)	Y0
I3 (ohřev)	Y7
I4 (snižování teploty)	Y5
I1 (napouštění vody)	Y7
I2 (vypouštění vody)	Y5

Tab. 5.3: Spojení jednotlivých výstupů PLC se vstupy mikrořadiče(2.úloha)

Spojení výstupů PLC se vstupy I1-I8 vedoucími do mikrořadiče je ale v programu rozdílné, protože využíváme všechny funkce v mikropočítači. V Tab. 5.3 je správné zapojení konektorů obou systémů pro danou úlohu.



Obr. 5.5: Třetí část programu se změnou otáček a směru v čase

6

Možnosti využití v praxi

Už v mnoha částech této prácé je avizováno, že výsledek by měl být využit v praxi, resp. při výuce automatizace. Automatizace se v dnešní moderní době rozpíná do všech koutů lidského žití a řízení systémů programovatelnými logickými automaty, se objevuje i v běžných aplikacích, kde by nás to před pár lety ani nenapadlo. Dříve velice drahé programovatelné automaty, které se využívaly pouze pro řízení velkých výrobních linek, dnes můžeme najít i v aplikacích, kde se původně s jejich využitím vůbec nepočítalo, např. i chytrých domácnostech. Typické konstrukce, jenž byly dříve tvořené jen mikropočítači. K velkému rozmachu vedl rychlý vývoj a zároveň zaměření se i na mikro PLC, které se díky jednodušší výrobě dostávají na nižší náklady.

Právě kvůli velkému rozmachu v dnešní době, je zapotřebí více a více lidí, jenž zvládají tyto logické automaty programovat, nastavovat a uvádět do výsledného procesu. To se i přes velkou snahu firem, které hledají takto kvalifikované zaměstnance, nedaří najít. Je potřeba více rozvíjet logické myšlení a s ním následně alespoň částečně úvody do automatizace na většině škol zaměřených na elektrotechniku a elektroniku.

Již v první kapitole jsem se zmiňoval, že najít učební přípravky, na kterých si studenti mohou vyzkoušet chod skutečného PLC, je málo, a když už se na trhu najdou, tak se jedná o velice přemrštěné částky. V této práci jsem si kladl za cíl, vytvořit jednoduchý, transparentní a levný přípravek, který studentům dá dostatečnou představu o chodu systému řízeném PLC, a zároveň si mohli jeho funkce, možnosti a nastavení volit oni sami. Simulátor pračky byl v tomto ohledu dobrou volbou a věřím, že bude v praxi při výuce mít trvalé využití. To se vše samozřejmě ukáže časem, kam se v následujících letech bude směřovat vývoj právě programovatelných logických automatů. Přesto všechno základy logického řízení budou i nadále stejné, proto bude využití tohoto přípravku možné. Nastavování a vymýšlení možností přidání dalších čidel, které by mohly spouštět chod programu, je nespočet, a proto může ve studentech podporovat představivost.

7

Závěr

Práce se zabývala vytvořením přípravku pro účely při demonstrativním řízení výuky programování PLC Simatic. Ty jsou v dnešní době ve velmi hojném počtu využívány a zaměření na vychování nových programátorů je pro mnohá odvětví velkou prioritou. Domnívám se, že výsledný přípravek by při výuce mohl být velkou pomocí.

Při návrhu mi záleželo na jednoduchosti a transparentnosti při daném řízení studenty. To se v základu povedlo a vyrobený model je pro stěžejní prvky potřebné v programování logických automatů postačující.

Kvůli ceně byla zvolena mikropočítačová deska s rozšiřujícím shieldem pro pro práci s vyššími napěťovými hladinami, než běžný mikrořadič může pracovat. Ve větší výrobě by se samozřejmě daly náklady snížit použitím jednoduchých mikroprocesorů (bohatě by postačil 8051) a tvorbě vlastní desky s integrovanými obvody pro převody napěťových hladin.

Komunikace přes SPI, která je v této práci využita, sice nedosahuje takových rychlostí jako paralelní přenos, ale vzhledem k rychlostem celého systému a možných přidaných integrovaných obvodů, nám zcela postačuje.

Výsledný přípravek by se dal také rozšířit o další paralelní výstupy k řadám LED znázorňujícím vodní sloupce a teplotu vody. Tím bychom dostali v možném návrhu programu další možnosti pro regulaci.

Poslední části mé práce, zaměřující se na tvorbu výukových programů jsou vhodně zvolené pro ukázku možných využití modelu pračky při řízení. Jistě by se ale dalo využít tyto jednotlivé části komplexněji a v jiném nastavení. To už vše záleží na fantazii obsluhy.

Literatura

[1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace.* Praha: BEN - technická literatura, 1999. ISBN 978-80-86056-58-6.

[2] ZEZULKA, František. *Prostředky průmyslové automatizace.* Brno: VUTIUM, 2004. ISBN 80-214-2610-1.

[3] *Automatizační systémy - Digital Factory Process Industries and Drives - Siemens.302 Found [online].* Copyright © [cit. 11.05.2018]. Dostupné z: http://stest1.etnetera.cz/ad/current/index.php?ctxnh=1d37de2332ctxp=home.

[4] *PLC - sestava.* PLC AUTOMATIZACE [online]. Dostupné z: http://www.plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm

[5] SITRAIN - Training for Industry - index, Siemens Training. *SITRAIN - Training for Industry - Training for Industry worldwide, Siemens Training [online].* Copyright © Siemens AG 2009 [cit. 11.05.2018]. Dostupné z: https://www.sitrain-learning.siemens.com/en/search/index.do?query=Simatic+TIA+Portal+programming+1tResultLength=10tPage=1

[6] *[online].* Dostupné z: https://www.foxon.cz/blogs/category/38-kurzy-programovani-v-tia-portal-s7-1500.html

[7] BOZAPALIDIS, Symeon a George RAHONIS*ed. Algebraic Informatics: second International Conference, CAI 2007 : Thessaloniki, Greece, May 21-25, 2007 : revised selected and invited papers.* Berlin: Springer, c2007. Lecture notes in computer science, 4728. ISBN 978-3-540-75413-8.

[8] 301 Moved Permanently [online]. Copyright © 2018 STMicroelectronics [cit. 04.06.2018]. Dostupné z: http://www.st.com/en/ecosystems/x-nucleo-plc01a1.html

[9] Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com [online]. Copyright © [cit. 04.06.2018]. Dostupné z: http://www.ti.com/lit/ds/symlink/uln2003a.pdf