

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

DEMONSTRAČNÍ SOUSTAVA PRO LOGICKÉ ŘÍZENÍ

Vratislav Fikar

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vratislav Fikar**
Osobní číslo: **I14039**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Název tématu: **Demonstrační soustava pro logické řízení**
Zadávající katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vytvořit demonstrační soustavu o několika vstupech a několika výstupech, která bude sloužit pro demonstraci logického řízení pomocí programovatelných logických automatů. Při návrhu je třeba dbát na bezpečnost, spolehlivost, časovou stálost a kompaktnost řešení. Předpokládá se výstup ve formě celistvé krabičky s příslušnými vstupy, výstupy a napájením, zároveň musí zařízení vizualizovat svůj stav (diody, displej). Součástí řešení bude také několik demonstračních návrhů řízení.

Teoretická část: Rešerše podobných modelů, teoretický popis algoritmů použitých v praktické části

Praktická část: Tvorba zařízení, návrh řízení různými metodami, vyhodnocení.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

BALÁTĚ, J. Automatické řízení. Praha: BEN, 2003. 654 s. ISBN 80-7300-020-2.

KWASNIEWSKI, J. Programmable Logic Controllers. Cracow: ROMA-POL, 2002. ISBN 83-86320-45-1.

VÁŇA, V. Mikrokontroléry Atmel AVR - assembler, BEN - technická literatura, Praha 2003, ISBN 80-7300-093-8.

Vedoucí bakalářské práce: **Ing. Petr Doležel, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **15. prosince 2016**

Termín odevzdání bakalářské práce: **12. května 2017**



Ing. Zdeněk Němec, Ph.D.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. prosince 2016

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 5. 2017

Vratislav Fikar

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Petrovi Doleželovi, Ph.D., za vstřícnost a ochotu při řešení jakéhokoliv požadavku a problému.

V Pardubicích dne 12. 5. 2017

Vratislav Fikar

ANOTACE

Bakalářská práce je věnována problematice logického řízení, které je následně nastíněno na demonstrační soustavě řízené programovatelným logickým automatem Simatic S7-1200. Je popsána tvorba demonstrační soustavy jak po hardwarové, tak po softwarové stránce v programovacím prostředí Totally Integrated Automation Portal V10.

KLÍČOVÁ SLOVA

logické řízení, programovatelný logický automat, Ladder diagram, TIA portal.

TITLE

EDUCATIONAL SYSTEM FOR LOGIC CONTROL

ANNOTATION

Bachelor thesis deals with the demonstration system for logic control. The system is controlled by a programmable logic controller Simatic S7-1200. In the thesis, the system is properly described (both hardware and software) programmed using an environment of Totally Integrated Automation Portal V10 is demonstrated, as well.

KEYWORDS

Logic control, programmable logic controller, Ladder diagram, TIA portal.

OBSAH

Seznam zkratk a značek.....	9
Seznam symbolů proměnných veličin a funkcí.....	10
Seznam ilustrací.....	11
Seznam tabulek.....	13
ÚVOD	14
1 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY	15
1.1 DĚLENÍ PROGRAMOVATELNÝCH LOGICKÝCH AUTOMATŮ.....	15
1.2 DRUHY PROGRAMOVACÍCH JAZYKŮ PLC	17
1.2.1 Jazyk strukturovaného textu.....	17
1.2.2 Jazyk příčkového diagramu.....	17
1.2.3 Jazyk funkčního blokového schématu	18
1.2.4 Sekvenční funkční diagram.....	18
1.2.5 Jazyk seznamu instrukcí.....	18
2 LOGICKÉ ŘÍZENÍ.....	19
2.1 ZNAČENÍ	19
2.2 FORMY POPISU	19
2.2.1 Pravdivostní tabulka.....	20
2.2.2 Logický výraz	20
2.2.3 Karnaughova mapa	21
2.3 KOMBINAČNÍ LOGICKÉ OBVODY	22
2.3.1 Logické funkce jedné proměnné.....	22
2.3.2 Logické funkce dvou proměnných	24
2.4 SEKVENČNÍ LOGICKÉ OBVODY	26
2.5 DRUHY ŘÍZENÍ.....	32
2.6 REŠERŠE PODOBNÝCH PRACÍ.....	32
2.6.1 Model výtahu.....	32
2.6.2 Model křižovatky.....	33
3 PRAKTICKÁ ČÁST.....	34
3.1 TVORBA DEMONSTRAČNÍ SOUSTAVY.....	34
3.2 VYTVOŘENÍ PROGRAMU DO ARDUINA DUE	36
3.3 SIEMENS SIMATIC S71200	37
3.4 SPOJENÍ DEMONSTRAČNÍ SOUSTAVY S PLC	38

3.5	TOTALY INTEGRATED AUTOMATION PORTAL V10	40
3.6	POUŽITÉ BLOKY LADDER DIAGRAM	42
3.7	FUNKCE PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU.....	43
3.8	PROGRAMOVÁNÍ PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU.....	43
4	ZHODNOCENÍ	55
5	ZÁVĚR	56
	LITERATURA	57
	PŘÍLOHY	59

SEZNAM ZKRATEK A ZNAČEK

IP	Internet Protocol (Internetový protokol)
KLO	Klopný logický obvod
LCD	Liquid Crystal Display (Dislejš z tekutých krystalů)
PLC	Programmable Logic Controller (Programovatelný logický automat)
TIA	Totally integrated Automation (Zcela integrovaná automatizace)
TTL	Tranzistorově tranzistorová logika

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

a	nezávislá binární proměnná logického výrazu
b	nezávislá binární proměnná logického výrazu
c	nezávislá binární proměnná logického výrazu
F	množina výstupních proměnných sekvenčních logických obvodů
J	vstupní nezávislá binární proměnná obvodu JK
K	vstupní nezávislá binární proměnná obvodu JK
n	značí číslo vstupu/výstupu
Q	výstupní proměnná logických obvodů
\overline{Q}	negovaná výstupní proměnná logických obvodů
R	vstupní nezávislá binární proměnná obvodu RS
S	vstupní nezávislá binární proměnná obvodu RS
s	sekunda, základní jednotka času
T	vstupní nezávislá binární proměnná obvodu RST
V	volt, jednotka elektrického napětí
X	množina vstupních proměnných sekvenčních logických obvodů
x	vstupní nezávislá binární proměnná logických funkcí
x_1	vstupní nezávislá binární proměnná logických funkcí
x_2	vstupní nezávislá binární proměnná logických funkcí
Y	množina vnitřních stavů sekvenčních logických obvodů
y	výstupní závislá binární proměnná logických funkcí/výrazů
"	palec, jednotka míry

SEZNAM ILUSTRACÍ

Obr. 1.1 – Kompaktní PLC Siemens 6ED1052	16
Obr. 1.2 – Modulární PLC do firmy Beckhoff	16
Obr. 2.1 – Kombinační logický obvod jedné proměnné	22
Obr. 2.2 – Značka funkce AND	24
Obr. 2.3 – Značka funkce OR	24
Obr. 2.4 – Značka funkce NAND	25
Obr. 2.5 – Značka funkce NOR	26
Obr. 2.6 – Sekvenční logický obvod	26
Obr. 2.7 – Klopný obvod RS složený z hradel	28
Obr. 2.8 – Schématická značka obvodu RS	28
Obr. 2.9 – Klopný obvod RST složený z hradel	29
Obr. 2.10 – Schématická značka RST obvodu	29
Obr. 2.11 – Klopný obvod D složený z hradel	30
Obr. 2.12 – Schématická značka obvodu D	30
Obr. 2.13 – Klopný obvod JK složený z hradel	31
Obr. 2.14 – Schématická značka obvodu JK	31
Obr. 3.1 – Demonstrační soustava	36
Obr. 3.2 – Siemens Simatic s71200	38
Obr. 3.3 – Ikona TIA Portal	40
Obr. 3.4 – Startovní okno	41
Obr. 3.5 – Tvorba projektu	41
Obr. 3.6 – Schématická značka spínacího kontaktu	42
Obr. 3.7 – Schématická značka rozpínacího kontaktu	42
Obr. 3.8 – Schématická značka nastavení cívký	42
Obr. 3.9 – Schématická značka resetování cívký	42
Obr. 3.10 – Schématická značka zpožd'ovače	43
Obr. 3.11 – Definování proměnných	46
Obr. 3.12 – První síť	47
Obr. 3.13 – Druhá síť	48
Obr. 3.14 – Třetí síť	49
Obr. 3.15 – Čtvrtá síť	50
Obr. 3.16 – První verze programu	51

Obr. 3.17 – Druhá verze programu	52
Obr. 3.18 – Třetí verze programu.....	54

SEZNAM TABULEK

Tab. 2.1 – Značení dvoustavových hodnot.....	19
Tab. 2.2 – Pravdivostní tabulka	20
Tab. 2.3 – Karnaughova mapa pro tři proměnné	21
Tab. 2.4 – Pravdivostní tabulka funkce falsum.....	22
Tab. 2.5 – Pravdivostní tabulka funkce verum	23
Tab. 2.6 – Pravdivostní tabulka funkce aserce	23
Tab. 2.7 – Pravdivostní tabulka funkce negace	23
Tab. 2.8 – Pravdivostní tabulka funkce AND.....	24
Tab. 2.9 – Pravdivostní tabulka funkce OR.....	25
Tab. 2.10 – Pravdivostní tabulka funkce NAND	25
Tab. 2.11 – Pravdivostní tabulka funkce NOR.....	26
Tab. 2.12 – Pravdivostní tabulka přechodů klopného obvodu RS složená z NAND hradel....	28
Tab. 2.13 – Pravdivostní tabulka přechodů klopného obvodu RST složená z NAND hradel..	29
Tab. 2.14 – Pravdivostní tabulka přechodů obvodu D sestaveného z hradel AND a OR.....	30
Tab. 2.15 – Pravdivostní tabulka přechodů obvodu JK sestaveného z hradel AND	31
Tab. 3.1 – Spojení demonstrační soustavy s PLC.....	39
Tab. 3.2 – Spojení programovatelného logického automatu a napěťového zdroje.....	39
Tab. 3.3 – Spojení demonstrační soustavy a napěťového zdroje.....	39
Tab. 3.4 – Kombinace výstupů	45

ÚVOD

Práce je zaměřena na logické řízení programovatelnými logickými automaty. Teoretická část je rozdělena na dvě hlavní kapitoly. V první kapitole jsou popsány třídy programovatelných logických automatů. Dále jsou uvedeny některé z programovacích jazyků logických automatů.

V druhé části je popsáno samotné logické řízení, formy popisu logických funkcí a jejich možné zjednodušení. Následuje výpis jednoduchých logických funkcí jedné/dvou proměnných s pravdivostními tabulkami. Dále popis složitějších logických funkcí, které dovedou uchovávat předchozí výstupní stav. V závěru druhé kapitoly teoretické části je popsáno řešení dvou soustav logického řízení.

Cíl práce je popsán v praktické části, která je umístěna ve třetí kapitole. Praktickým výstupem práce je simulátor kávovaru. V úvodu je popsána hardwarová tvorba simulátoru kávovaru, následuje popis vytvoření řídicího programu do Arduina DUE pro zobrazování stavu simulátoru. Soustava zobrazuje stavy jako opravdový kávovar. Dále je popsáno spojení simulátoru kávovaru s programovatelným logickým automatem. Je popsán důvod, proč soustava vyžaduje dvě úrovně napájení. V závěru třetí kapitoly je popsána tvorba programu do programovatelného logického automatu. Práce obsahuje tři rozdílně obtížné programy řízení simulátoru kávovaru. Postup programování je popsán v programovacím prostředí TIA portálu dodávaného k programovatelnému logickému automatu Siemens Simatic s7-1200, který je zde použit pro řízení demonstrační soustavy kávovaru.

1 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

Za jedno z nejrozšířenějších automatických řízení je považováno právě řízení pomocí programovatelných logických automatů. Původně bylo logické řízení realizováno pomocí reléového řízení, které bylo využíváno již na přelomu 19. a 20. století (Kolář, 2006).

Nicméně reléové řízení mělo značné nevýhody, jako například změna řídicího systému byla velice obtížná, dále bylo velice nákladné uvádění do provozu, proto v pozdější době během sedmdesátých let dvacátého století vznikají první programovatelné automaty, které mají za úkol odstranit nevýhody reléového řízení.

Hlavním cílem programovatelných automatů byla jednoduchá možnost změny programu bez změn v zapojení, dále jednoduché naprogramování, nenáročné uvádění do provozu, a v neposlední řadě také jednoduchá a levná údržba (Martinásková, 2001).

Technické vybavení automatů je navrženo tak, aby i v drsných pracovních podmínkách spolehlivě vykonávalo danou funkci. Automaty jsou obvykle vybaveny i vlastními diagnostickými funkcemi, které v průběhu činnosti systému dokáží odhalit případnou závadu. Nezvratnou výhodou je také možnost opakovaně měnit vnitřní program automatu. Program je tak možno měnit v případě některých nedostatků návrhu. Další výhodou je také možnost spojovat logické automaty a vytvářet tak obrovské systémy, které mohou např. řídit celý chod chytrých domů. Jako příklad logického řízení v běžném životě je řízení křižovatky semaforů, dále také řízení výtahu, kde je velice důležité dbát na správnou funkčnost. Jednoduché logické řízení je také velice používané v domácích spotřebičích, jako je pračka, myčka, elektrický sporák.

1.1 DĚLENÍ PROGRAMOVATELNÝCH LOGICKÝCH AUTOMATŮ

Programovatelné automaty můžeme dělit do skupin, podle provedení, podle velikosti (počtu vstupů a výstupů), podle počtu procesorů. Podle provedení se automaty dělí na kompaktní a modulární, dále podle velikosti dělíme automaty na mikro, malou, střední, a nejvyšší výkonnostní třídu a podle počtu procesorů na jednoprocessorové a víceprocesorové automaty.

Kompaktní automat obsahuje celý systém v jednom modulu, digitální a analogové vstupy/výstupy, základní podporu komunikace, zdroj napájení. Výstupní funkce automatu je tak omezena na použití daných prvků v tomto modulu.



Obr. 1.1 – Kompaktní PLC Siemens 6ED1052 (ELUC)

Modulární automat, jak už název napovídá je rozdělen do jednotlivých modulů jako zdroj, CPU, vstupy/výstupy, a dalších funkčních bloků. Je možno si tak složit automat, který bude mít všechny potřebné prvky, pro jednotlivou regulaci.



Obr. 1.2 – Modulární PLC do firmy Beckhoff (ELUC)

Mikro třída PLC automatů, někdy též nazývaná jako třída programovatelných relé, představuje automaty, které mají do 20 vstupů/výstupů, dále disponují malou pamětí, a jsou bez možnosti komunikace. Např. Siemens LOGO!, Mitsubishi ALFA.

Automaty spadající do malé třídy mají do 128 vstupů/výstupů. Tyto automaty jsou již obvykle vybaveny možností komunikace, a mají již výrazně větší výkon. Do této řady patří Festo FEC20, Siemens S7-200.

Do střední výkonnostní třídy spadají automaty, které mají mezi 128 ÷ 512 vstupy/výstupy. Jedná se většinou o automaty v modulárním provedení např. Siemens Simatic S7-300.

Nejvyšší výkonnostní třída se vyznačuje tím, že automaty mohou obsahovat až tisíce vstupů/výstupů. Není tak divu, že se jedná vždy o speciální programovatelné automaty v modulárním provedení, např. Festo FPC 405, Siemens Simatic S7-400 (Šťastný, 2013).

1.2 DRUHY PROGRAMOVACÍCH JAZYKŮ PLC

Zpočátku byly programovatelné logické automaty programovány v jednoduchém jazyku kontaktních schémat. Díky jednoduchosti programování v tohoto jazyce se automaty staly velice oblíbenými a začaly se rychle rozšiřovat do všech oblastí. Díky růstu možností využití těchto automatů začaly vznikat i nové typy programovacích jazyků a starší jazyky se obohacovaly o nové funkce.

Původně každý výrobce měl svůj vlastní programovací jazyk. Pro uživatele automatů od různých firem tak vznikly komplikace při programování. Tyto komplikace se vyřešily až s příchodem mezinárodní normy, která doporučuje základních pět jazyků s jasně danou sémantikou a syntaxí. Jedná se o jazyky Ladder Diagram, Function Block Diagram, Instruction List, Structured Text, Sequential Chart. Nicméně někteří výrobci uvádějí možnost programování i v jiných programovacích jazycích, které mohou přinášet uživatelům určité výhody (Kochaníček, 2010).

1.2.1 Jazyk strukturovaného textu

Jedná se o vyšší programovací jazyk, který má své kořeny v jazycích Pascal a C. Daná syntaxe je dána povolenými výrazy a příkazy. Je zde definováno deset základních typů příkazů (přiřazení, vyvolání funkce, návrat, výběr apod.). Jednotlivé příkazy se mezi sebou oddělují středníkem a může jich tak být i více umístěno na jednom řádku (Kochaníček, 2010).

1.2.2 Jazyk příčkového diagramu

Grafický jazyk Ladder Diagram, který je také nazýván, jako jazyk kontaktních schémat. Prvky daného jazyku tak připomínají reléovou logiku. Programování probíhá jako přetahování prvků do ohraničené sítě, ve které se pak následně provádějí instrukce zleva doprava. Tento jazyk bude dále popsán v praktické části této práce (Kochaníček, 2010).

1.2.3 Jazyk funkčního blokového schématu

Jazyk Function Block Diagram je také grafický, a jak již zde název napovídá, programování probíhá jako spojování jednotlivých grafických bloků mezi sebou. Používají se zde standartní funkční bloky, pro vyjádření logických funkcí, a také čítače a časovače, komunikační bloky a je zde možnost i speciálních bloků. Čtení programu zde také probíhá z levé strany do pravé (Kochaníček, 2010).

1.2.4 Sekvenční funkční diagram

Anglicky nazýván jako Sequential Function Chart je grafickým jazykem odvozeným ze symboliky Petriho sítí. Program se zde skládá z kroků a přechodů, kde každý přechod definuje podmínky, které musejí být splněny, aby došlo k vykonání akcí, které jsou přiřazeny ke každému kroku. Přechody i bloky akcí je možné programovat pomocí libovolného jazyku uvedeného v normě. Jazyk umožňuje i větvení programu a následnou synchronizaci větví (Kochaníček, 2010).

1.2.5 Jazyk seznamu instrukcí

Jako posledním jazykem z mezinárodní normy je Instruction List, který patří do skupiny textových jazyků. Často se označuje jako jazyk pokynů, seznam instrukcí, připomíná stavbou programovací jazyk assembler. Instrukce se zde zapisují na samostatný řádek a je zde možné také přidávat komentáře pro lepší orientaci v programu. Program se zde provádí od shora dolů, popřípadě pomocí skoků (Kochaníček, 2010).

2 LOGICKÉ ŘÍZENÍ

Hlavním cílem logického řízení je zpracovávat vstupní informace tak, aby výstupní informace odpovídaly danému předepsanému cíli. Tomuto zpracování dochází za použití dvouhodnotových veličin. Tyto veličiny mohou vždy nabývat pouze jednoho ze dvou stavů – vypínač je sepnutý/rozepnutý, topení je zapnuto/vypnuto. Stejně tak tomu je i při monitorování stavu systému – rychlost je nad/pod danou hodnotou, teplota je pod/nad danou teplotou (Dočekal, 2010).

2.1 ZNAČENÍ

Tyto dvoustavové hodnoty mohou být různě označovány. Nejčastější označování je popsáno v tab. 2.1.

Tab. 2.1 – Značení dvoustavových hodnot

Formálně	0	1
Fyzikálně	Vypnuto	Zapnuto
Slovně	Ne	Ano
Anglicky	False	True

Dále jsou tyto dvoustavové hodnoty označovány pomocí jasně daných rozsahů napětí. Tyto rozsahy mohou být např. $0\text{ V} \div 0,8\text{ V}$ sloužící pro definici logická nuly, $2\text{ V} \div 5\text{ V}$ zas definuje logickou jedničku, jak je tomu u TTL obvodů.

U programovatelných logických automatů jsou tyto rozsahy posunuty. Pro definování logické 0 je napětí v rozsahu $0\text{ V} \div 5\text{ V}$, a pro logickou 1 je definován rozsah $15\text{ V} \div 30\text{ V}$, například u Siemens Simatic S7-1200. Tato napětí jsou určována napájecím napětím daných zařízení, a také jejich stavbou.

2.2 FORMY POPISU

Pro zápis logických soustav máme více možností. Zápis má za úkol vystihnout všechny stavy systému, které mohou nastat, a zároveň popisuje všechny stavy vstupních proměnných.

2.2.1 Pravdivostní tabulka

Pravdivostní tabulka je nejběžnější forma zápisu pro logické funkce. Jedná se o tabulku, která má 2^n řádků, kde 2 značí počet stavů, kterých může proměnná nabývat, a n je počet vstupních proměnných.

Tab. 2.2 – Pravdivostní tabulka

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Do levé části tabulky se obecně zapisují stavy vstupních proměnných a jejich kombinací tak, aby byly popsány všechny možné kombinace stavů, vpravo se zas zapisují stavy výstupních proměnných. Tento druh zápisu je velice rozšířený, protože velice jednoduše dokáže popsat malé logické soustavy. Nicméně není moc vhodný pro popis rozsáhlých funkcí, kde s druhou mocninou vstupních proměnných navyšují řádky tabulky.

2.2.2 Logický výraz

Jako další způsob zápisu je možný popis pomocí disjunktivní a konjunktivní formy. Jedná se o algebraickou formu vyjádření logických funkcí. Obě tyto formy dokáží vyjádřit danou funkci.

Konjunktivní normálová forma popisující funkci se skládá z logických součtů (OR) proměnných, případně z negací těchto součtů. Jednotlivé součty proměnných jsou pak spojeny logickým součinem. Z pravdivostní tabulky jsou vypisovány pouze řádky, ve kterých má být výstupní proměnná rovna 0. Proměnné, které mají hodnotu 0 jsou zapisovány přímo, zatímco proměnné, které nabývají 1 jsou zapisovány jako negované (Dočekal, 2010).

$$y = (a + b + c) + (a + b + \bar{c}) + (a + \bar{b} + c) + (\bar{a} + b + c) \quad (2.1)$$

kde a – nezávislá binární proměnná
 b – nezávislá binární proměnná
 c – nezávislá binární proměnná

Disjunktivní funkce má opačný postup tvorby, nejprve je vytvořen součin (AND) proměnných, popřípadě negace součinu, a až poté jsou tyto součiny spojeny logickým součtem (OR). Z pravdivostní tabulky se tentokrát vypisují řádky, ve kterých výstupní proměnná nabývá 1. Při tvoření zápisu jsou proměnné, které jsou rovny 1 zapisovány přímo, a opačné stavy zapisovány jako negace proměnných (Dočekal, 2010).

$$y = \bar{a}bc + a\bar{b}c + abc + ab\bar{c} \quad (2.2)$$

kde a – nezávislá binární proměnná
 b – nezávislá binární proměnná
 c – nezávislá binární proměnná

2.2.3 Karnaughova mapa

V neposlední řadě popisu je nutné zmínit Karnaughovu mapu, která slouží nejen pro zápis logické funkce, ale také především k jejímu zjednodušení. Mapa obsahuje tolik políček, kolik je možných kombinací proměnných. Obsahuje tedy 2^n políček, kde n je počet vstupních proměnných.

Tab. 2.3 – Karnaughova mapa pro tři proměnné

	b	b, c	c
a	0	1	0
	0	1	1

Výsledný zjednodušený vztah (2.3), ačkoli vypadá velice jednoduše, popisuje stejnou funkci jako předchozí rovnice (2.1), (2.2).

$$y = ab + bc + ac \quad (2.3)$$

kde a – nezávislá binární proměnná

b – nezávislá binární proměnná

c – nezávislá binární proměnná

Dále by v tabulce následovalo zjednodušování, pokud je vůbec možné, a to tak, že bychom seskupovali 1, které jsou umístěny vedle sebe horizontálně, nebo vertikálně, do skupin o velikosti mocnin 2, to znamená do skupin po 2, 4, 8, a dále. Výsledný vztah je již uveden jako zjednodušený.

2.3 KOMBINAČNÍ LOGICKÉ OBVODY

Mezi nejjednodušší logické obvody patří kombinační obvody. V těchto obvodech je výstupní stav dán jednoznačně stavy vstupních proměnných. K tomuto stavu dojde až po odeznění přechodových dějů. Je potřeba si na tento přechodový děj dát pozor, zejména u složitějších systémů. Kombinační obvody neobsahují žádné paměťové členy, ani zpětnou vazbu. Tyto obvody slouží jako elementární logická zařízení, uskutečňující např. sčítačky, paralelní dekodéry.

2.3.1 Logické funkce jedné proměnné



Obr. 2.1 – Kombinační logický obvod jedné proměnné

Těchto logických funkcí o jedné proměnné máme čtyři druhy, falsum, verum, aserce, negace. Tyto funkce jsou níže nastíněny pomocí pravdivostních tabulek (Navrátil, 2011).

Funkce falsum neboli lež je pro libovolný vstup rovna 0.

Tab. 2.4 – Pravdivostní tabulka funkce falsum

x	y
0	0
1	0
falsum – lež	

$$y = 0$$

(2.4)

Funkce verum neboli pravda je pro libovolný vstup rovna 1.

Tab. 2.5 – Pravdivostní tabulka funkce verum

x	y
0	1
1	1
verum – pravda	

$$y = 1 \quad (2.5)$$

Funkce aserce, jinak nazývaná jako opakování. Hodnotu vstupu zopakuje na výstup.

Tab. 2.6 – Pravdivostní tabulka funkce aserce

x	y
0	0
1	1
aserce – opakování	

$$y = x \quad (2.6)$$

Funkce negace má vždy opačnou hodnotu na výstupu, než je hodnota na vstupu. Negace je neznámější a také nejčastěji používaná unární funkce.

Tab. 2.7 – Pravdivostní tabulka funkce negace

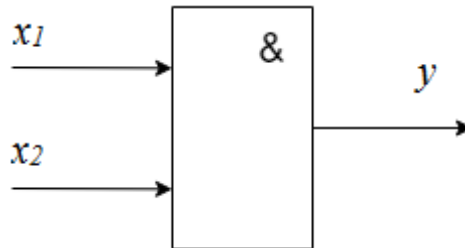
x	y
0	1
1	0
negace – opak	

$$y = \bar{x} \quad (2.7)$$

2.3.2 Logické funkce dvou proměnných

Logických funkcí dvou proměnných je celkem 16, nicméně nejčastěji se používají následující čtyři.

Konjunkce neboli logický součin (násobení), AND. Výstupní proměnná nabývá hodnoty 1, právě když jsou obě vstupní hodnoty rovny také 1.



Obr. 2.2 – Značka funkce AND

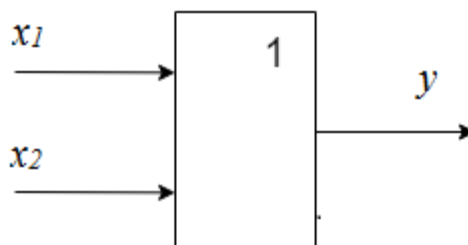
Tab. 2.8 – Pravdivostní tabulka funkce AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$y = x_1 \wedge x_2 \quad (2.8)$$

Logický součin je značen \wedge , čteme ho jako „i“.

Disjunkce neboli logický součet, OR. Výstupní proměnná nabývá hodnoty 1, právě když je alespoň jedna vstupní proměnná rovna 1.



Obr. 2.3 – Značka funkce OR

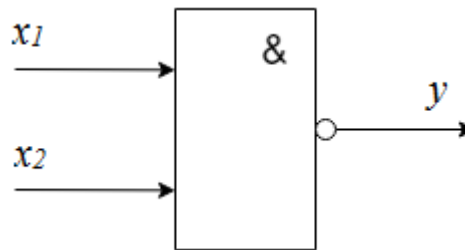
$$y = x_1 \vee x_2 \quad (2.9)$$

Tab. 2.9 – Pravdivostní tabulka funkce OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Logický součet je značen \vee , čteme jako „nebo“.

Negace konjunkce neboli negace logického součinu, NAND. Výstupní proměnná nabývá hodnoty 1, právě když obě vstupní proměnné nejsou rovny 1.



Obr. 2.4 – Značka funkce NAND

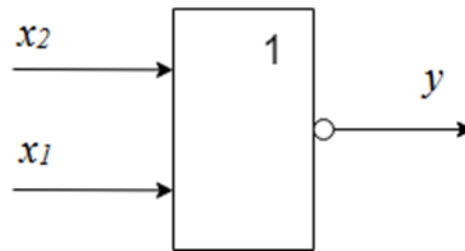
Tab. 2.10 – Pravdivostní tabulka funkce

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

$$y = \overline{x_1 \wedge x_2} \quad (2.10)$$

Negaci logického součinu značíme pomocí logického součinu, a znaku pro negaci, který je umístěn nad celým členem.

Negace disjunkce neboli negace logického součtu, NOR. Výstupní proměnná nabývá hodnoty 1, pouze když jsou obě vstupní hodnoty rovny 0.



Obr. 2.5 – Značka funkce NOR

Tab. 2.11 – Pravdivostní tabulka funkce NOR

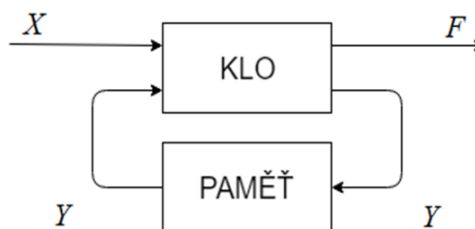
x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

$$y = \overline{x_1 \vee x_2} \quad (2.11)$$

Negaci logického součtu značíme pomocí logického součtu, a znaku pro negaci, který je umístěn nad celým členem.

2.4 SEKVENČNÍ LOGICKÉ OBVODY

Sekvenční obvody disponují již složitější stavbou než obvody kombinační. U těchto obvodů záleží jak na okamžitých kombinacích vstupních proměnných, tak i na jejich předchozím stavu. To znamená, že obvody mají paměť, ve které uchovávají předchozí stav výstupu.



Obr. 2.6 – Sekvenční logický obvod

Kde X značí množinu vstupních proměnných, F značí množinu výstupních proměnných, Y značí množiny vnitřních proměnných, KLO je zkratka kombinačního logického obvodu.

Sekvenční obvody tedy mohou tedy nabývat více různých výstupních stavů pro jednu kombinaci vstupů. Dále se sekvenční obvody dělí na asynchronní a synchronní.

U asynchronních sekvenčních obvodů dochází ke změně výstupů téměř okamžitě při změně vstupních stavů. Doba této změny je dána pouze průchodem přes logické členy. To je velice výhodné pro obvody, kde je dán velký důraz na rychlost zpracování. Nicméně u složitějších zapojení těchto logických obvodů může dojít k takzvanému hazardnímu stavu. Tento stav je způsoben časovými rozdíly zpracování v jednotlivých logických členech (Navrátil, 2011).

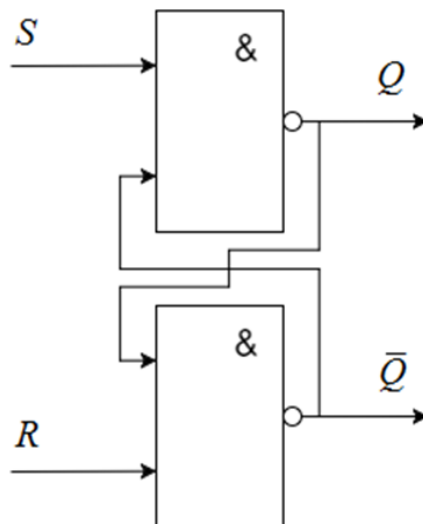
Synchronní sekvenční obvody řeší problém vzniku hazardního stavu, protože nemění svou hodnotu ihned při změně vstupních proměnných. Tyto obvody změni svou stav až při příchodu taktovacího (clock) signálu. Tímto se docílí, že systém mění své hodnoty postupně v definovaných okamžicích, danými hodinovým signálem. Běžně je používána ke změně náběžná hrana taktovacího signálu. Mezi obvyklé sekvenční obvody patří klopné obvody, registry, čítače, mikroprocesory (Navrátil, 2011).

Klopné obvody je dále možné dělit podle stavu výstupu v závislosti na čase.

Monostabilní klopné obvody mají pouze jeden stabilní stav, ve kterém mohou setrvávat libovolně dlouhou dobu. Druhý stav je kvazistabilní, jedná se o stav, ve kterém mohou setrvávat pouze přechodně, po dobu kmitu (Navrátil, 2011).

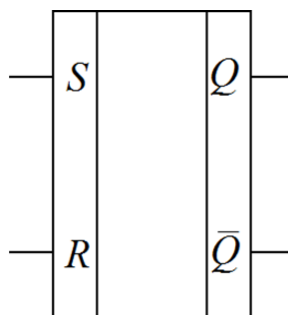
Bistabilní klopné obvody mají již dva stabilní stavy, ve kterých se mohou nacházet. Lze je například použít, jako paměť. Nejčastěji se setkáváme s typy RS, RST, D, JK, které se nacházejí v podobě integrovaného obvodu, nebo v podobě funkčních bloků v programovacích schématech programovatelných automatů (Navrátil, 2011).

Klopný obvod RS má v asynchronním provedení pouze dva vstupy – R který slouží k nulování a S slouží k nastavení. Obvod je možné sestavit ze dvou hradel NAND.



Obr. 2.7 – Klopný obvod RS složený z hradel (Navrátil, 2011)

Je zde nutné brát na vědomí, že při složení z těchto hradel se pro nulování R použije logická 0, a stejně tak pro nastavení pomocí S slouží logická 0.



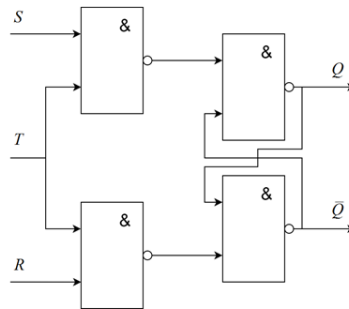
Obr. 2.8 – Schématická značka obvodu RS

Tab. 2.12 – Pravdivostní tabulka přechodů klopného obvodu RS složená z NAND hradel

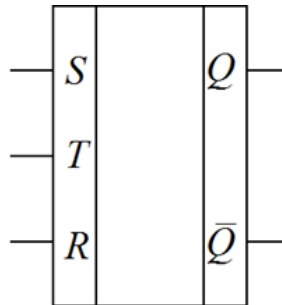
R^n	S^n	Q^{n+1}	$\overline{Q^{n+1}}$
0	0	-	-
0	1	0	1
1	0	1	0
1	1	Q^n	$\overline{Q^n}$

Jak je vidět v tab. 2.12 na prvním řádku nabývá výstupní hodnota log. 1 ale zároveň i její negace. Tomuto stavu dochází, když jsou oba vstupy set, i reset nastaveny, čemuž zde dochází při log. nule. Tento stav je nazýván jako „zakázaný stav“ a nemělo by k němu docházet.

Klopný obvod RST je synchronní klopný obvod o třech vstupech. R , S mají stejnou funkci jako u obvodu RS, a T pak slouží k synchronizaci. Až při příchodu logické jedničky na vstup T dojde k vykonání funkce.



Obr. 2.9 – Klopný obvod RST složený z hradel (Navrátil, 2011)



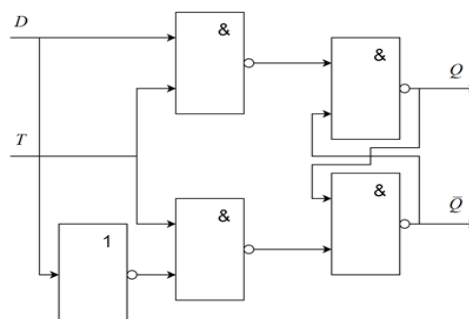
Obr. 2.10 – Schématická značka RST obvodu

Tab. 2.13 – Pravdivostní tabulka přechodů klopného obvodu RST složená z NAND hradel NOR NAND

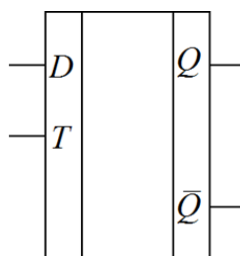
R^n	S^n	Q^{n+1}	\overline{Q}^{n+1}
0	0	Q^n	\overline{Q}^n
0	1	1	0
1	0	0	1
1	1	-	-

Tabulka není zcela kompletní, chybí zde sloupec T . Z celkové tabulky jsou zde vyjádřeny pouze řádky, které mají smysl, to jsou řádky, ve kterých T bylo v logické 1. Také je zde nutné podotknout, že zde již nedochází k otočení logických proměnných, S se nastavuje logickou 1, stejně tak se provádí nulování logickou 1 na vstupu R . Jak již bylo výše zmíněno můžeme si zde také povšimnout na posledním řádku, že dochází opět k zakázanému stavu.

Klopný obvod D vznikne přidáním invertoru mezi vstupy S a R obvodu RST. Tímto se z tabulky přechodů vyloučí první a poslední řádek. Zde již nemůže dojít k zakázanému stavu. Na vstup D se přivádí hodnota, která se má uchovávat. S příchodem log. 1 na vstup T dojde k zápisu hodnoty vstupu na výstup, kde zůstane až do dalšího příchodu log. 1 na vstup T .



Obr. 2.11 – Klopný obvod D složený z hradel (Navrátil, 2011)

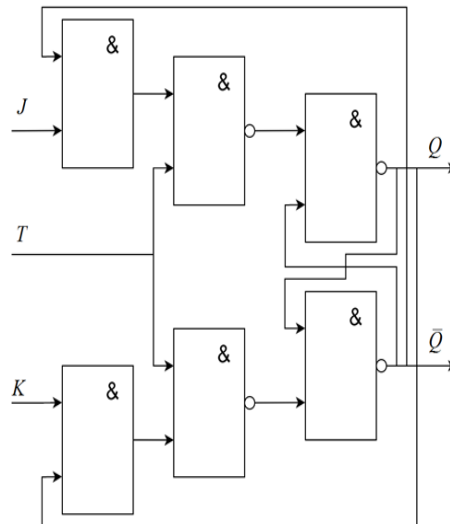


Obr. 2.12 – Schématická značka obvodu D

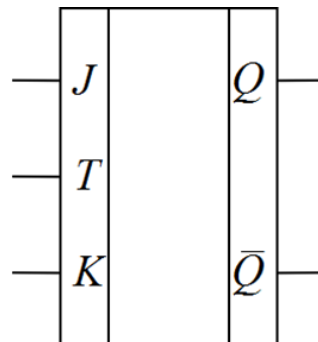
Tab. 2.14 – Pravdivostní tabulka přechodů obvodu D sestaveného z hradel AND a OR

D^n	Q^{n+1}	$\overline{Q^{n+1}}$
0	0	1
1	1	0

Klopný obvod JK je zdokonalením obvodu RS. Nejčastější uplatnění tohoto klopného obvodu je při realizaci čítačů. Tento obvod také odstraňuje zakázaný stav. Další zvláštností tohoto obvodu je, že při přivedení log. 1 na vstup J i K dojde ke kmitání na výstupním pinu. Toto kmitání trvá tak dlouho, jak dlouho je log. 1 na pinu T (časovacím vstupu).



Obr. 2.13 – Klopný obvod JK složený z hradel (Navrátil, 2011)



Obr. 2.14 – Schématická značka obvodu JK

Někdy je používána i značka, na které jsou zobrazeny vývody použitého obvodu RS.

Tab. 2.15 – Pravdivostní tabulka přechodů obvodu JK sestaveného z hradel AND

J^n	K^n	Q^{n+1}	\overline{Q}^{n+1}
0	0	Q^n	\overline{Q}^n
0	1	0	1
1	0	1	1
1	1	x	x

Je zde opět zobrazena jen část tabulky, kde T je v log. 1. Na posledním řádku tabulky můžeme vidět, že dochází ke kmitání na výstupu.

Astabilní klopné obvody nemají ani jeden stabilní stav. Jejich výstup je stále přepínán mezi logickou jedničkou a logickou nulou. Používají se jako generátory obdélníkového signálu, který může sloužit jako zdroj hodinového signálu (Navrátil, 2011).

2.5 DRUHY ŘÍZENÍ

Rozlišujeme základní dva typy řízení, přímé a zpětnovazební.

Přímé neboli dopředné řízení je chod opakujících instrukcí, který je jasně dán programem. U takového řízení se nekontroluje skutečný stav soustavy. Je zde potřeba dokonale znát danou soustavu, například u systému rozvodu páry je nesmírně důležité vědět, kdy zhruba dojde k úbytku tlaku. Soustavy řízené pouze přímým řízením tak nedokáží reagovat na poruchy, proto se toto řízení většinou kombinuje se řízením zpětnovazebním (VanDoren, 2010).

Zpětnovazební řízení již kontroluje skutečný stav regulované soustavy. Toto řízení již dokáže reagovat na poruchy a následně je pak s určitým odstupem odstranit. Porucha je myšlena jako např. pokles tlaku v rozvodu páry. Jak je výše psáno tato reakce je zpožděna, o dobu, než dojde ke změně skutečného stavu, což je nepříjemné. Proto je výhodné použít oba tyto druhy řízení společně. Výsledná soustava pak dokáže rychle a přesně reagovat soustavu (VanDoren, 2010).

2.6 REŠERŠE PODOBNÝCH PRACÍ

V minulosti již byla popsána problematika logického řízení na různých modelech. Zde jsou uvedeny dva nejčastější modely pro logické řízení.

2.6.1 Model výtahu

V úvodu práce je popisována tvorba modelu soustavy, kde se nejprve definují požadavky. Soustava obsahuje čtyři podlaží, mezi kterými se bude výtah pohybovat.

Dále je zde popisováno veškeré ovládání a řeší se zde i nosnost výtahu. V další části práce jsou vyobrazeny grafické modely simulující chování kabiny, cestujícího a následné provázanosti těchto částí, kde jsou popsány veškeré akce v závislosti na vstupech soustavy. Následně je popisována tvorba generátoru fronty cestujících.

Následuje popsání implementace jednotlivých částí. Aplikace vytvořená v prostředí Borland C++ BuilderTM graficky zobrazuje aktuální stav soustavy. Vstupní a výstupní signály jsou přenášeny přes vstupně výstupní kartu PCI-1750.

Na závěr je popsáno jednoduché řešení řízení modelu výtahu, kde jsou použity pouze nejnütnější funkce, ale hlavní funkčnost zůstala zachována (Šišák, 2003).

2.6.2 Model křižovatky

Z kraje práce je vysvětlena problematika logického řízení programovatelnými logickými automaty. Dále je popsán použitý hardware, což je jedná o modulární automat Simatic S7-300, se vstupně/výstupními kartami.

V další části je rozebírána konfigurace hardwaru, což se jedná o definici adres vstupů a výstupů PLC automatu. Adresy jsou dále umístěny v grafickém znázornění, kde je přesně vidět, co jaký vstup/výstup určuje.

Před samotnou realizací demonstrační soustavy byla vytvořena testovací deska za účelem otestování funkčnosti všech součástí obsahující model křižovatky.

V závěru práce je popisována samotná realizace demonstrační soustavy křižovatky. Model obsahuje světelnou signalizaci, propojovací kabely PLC s modelem, schéma modelu a rozmístění součástí. Dále je umístěno foto, kde jsou popsány jednotlivé části soustavy.

Na konec je popsáno testování soustavy a uveden program v programovacím jazyku Ladder diagramu pro řízení modelu PLC automatem (Svoboda, 2012).

3 PRAKTICKÁ ČÁST

Je zde popsána celá tvorba demonstrační soustavy, jak po hardwarové, ta po softwarové stránce, včetně jednotlivých návrhů řízení soustavy.

Vytvořená demonstrační soustava slouží k simulování stavů kávovaru. Pro zobrazení stavů je použit LCD displej, řízený Arduinem DUE. Soustava simuluje stavy zásobníků (vody i kávy) k této simulaci slouží přepínače 10 a 11. Pokud jsou přepínače v logické jedničce, jedná se o stav, kdy jsou oba zásobníky dostatečně plné a zobrazí se nabídka káv.

V nabídce je možnost výběru jedné ze tří typů káv, které se po zvolení začnou připravovat. Na výběr je káva Ristretto, Espresso a Lungo. Pro simulování výběru jsou použity spínací tlačítka 6, 7 a 8. Při přípravě jakékoli kávy se simuluje mletí zrn kávy. Toto mletí trvá tak dlouho, dokud nezměníme stav páčkového přepínače 9. Přepínač slouží k simulování jakési váhy, která by po namletí 7 gramů změnila stav. Dále již následuje protékání vody namletou kávou, zde je čas vytékání různý, podle typu vybrané kávy. Typ přípravy třetí kávy se liší o přidání vody po dokončení vytékání kávy. Po skončení přípravy kávy je na displeji vypsána informace o dokončení a zároveň je zobrazena hláška, o simulování stavu vyhození použité namleté kávy.

Pro řízení soustavy byly vytvořeny tři rozdílně obtížné programy, které slouží k řízení demonstrační soustavy.

3.1 TVORBA DEMONSTRAČNÍ SOUSTAVY

Hlavním prvkem soustavy je elektronická platforma Arduino Due. Arduino Due je rozšířenějším a výkonnějším typem Arduina Megy. Je zde použit výkonnější čip, Atmel SAM3x8E, který běží na 84MHz taktovací frekvenci. Jádro má 32-bitovou architekturu, díky čemuž je zde velký skok oproti ostatním deskám Arduina, které běží na 8-bitové architektuře. Na desce jsou umístěny dva microUSB konektory, kde jeden slouží pro samotné programování čipu, a druhý může sloužit pro připojení zařízení jako může být myš, nebo klávesnice (Voda, 2014).

Pro samotnou demonstrační soustavu jsou zde použity pouze logické piny 8-11, a piny 22-53, které jsou použity ke komunikaci s LCD displejem.

Displej má velikost 3,2", na kterém je zobrazován aktuální stav soustavy. Arduino Due bylo umístěno do elektroinstalační lišty EIP 100×40×2000 mm, která byla zkrácena na 210 mm. Lišta zde slouží jako ochrana pro Arduino Due, a zároveň jako kostra pro uchycení dalších částí demonstrační soustavy.

Skrz víko lišty byl vyříznut otvor na konektor LCD displeje, který bylo tak možné rovnou zapojit do Arduina Due. Aby LCD displej dobře seděl na víku lišty, bylo nutné udělat další otvor o velikosti čtečky na paměťové karty, který je umístěn ze spodní strany displeje. Pomocí vytvořeného klíče z drátu, který byl vložen do čtečky na paměťové karty, byl displej lépe upevněn k víku lišty. Do víka lišty bylo nutné vyvrtat další otvory, a to přesně tři o průměru 15 mm, do kterých byla našroubována spínací tlačítka. Tlačítka budou sloužit pro výběr volby. Do víka bylo nutné vyvrtat ještě další tři otvory o průměru 6 mm, do kterých byly následně usazeny páčkové přepínače. Pomocí těchto přepínačů bude možné simulovat stavy.

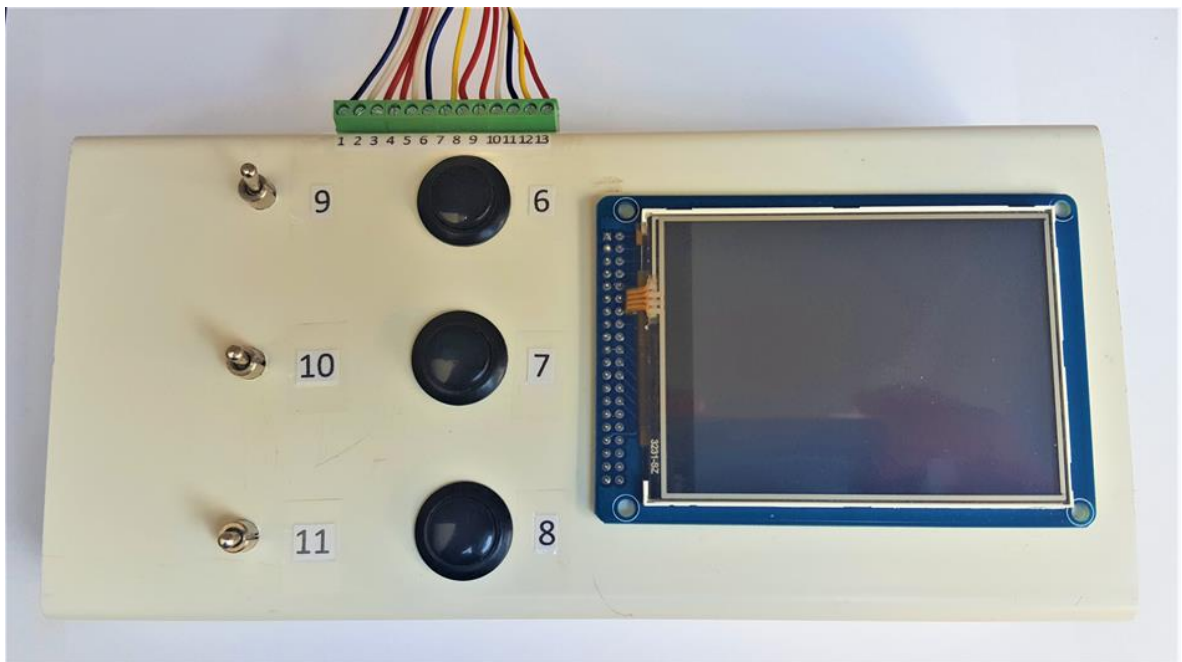
Jak páčkové přepínací, tak spínací tlačítka mají vždy vyvedeny pouze jeden kontakt. Druhé kontakty jak spínacích, tak páčkových přepínačů jsou spojeny. Tento kontakt je také vyveden na svorkovnici, na kterou následně bude přivedeno kladné napětí.

Tak docílíme toho, že páčkovými přepínači budeme moci přivádět kladné napětí. Piny 8-11 Arduina Due jsou spojeny s odporovým děličem, který má za úkol snížit výstupní napětí PLC automatu z 24 V na napětí o velikosti 5 V, které je již možné přivést na piny Arduina Due. Vstupní kontakty děliče jsou spojeny se svorkovnicí. Svorkovnice je umístěna na tištěném spoji spolu s odporovým děličem. Tištěný spoj je zapojen pomocí pinů přímo do Arduina Due. Druhá strana tištěného spoje je umístěna pod spínacími tlačítky, kde je vyvedena svorkovnice.

Celá demonstrační soustava je tak umístěna pouze na víku lišty. Aby bylo možné lištu jednoduše zavřít, bylo nutné do boků lišty vyříznout otvory, na svorkovnici, spínací tlačítka. Strany lišty byly následně zaslepeny za použití zbytku lišty.

Bohužel zde nebylo možné použít k napájení Arduina Due napěťový stabilizátor 7805, který by snížil přiváděné napětí 24 V z napěťového zdroje PLC na napětí o velikosti 5 V. Napěťový stabilizátor se nadměrně zahříval i když byl usazen na chladiči, který by se vešel do lišty. Z tohoto usuzuji, že potřebný proud pro napájení Arduina Due a LCD displeje byl vyšší, než co by mohl daný chladič uchládit. Stabilizátor by byl následně zničen nadměrnou teplotou.

Proto bylo potřeba na levém boku vyvrtat otvor na kabel, který slouží k napájení Arduina Due přímo z počítače. Ke svornici byly připojeny barevně označené kabely, pomocí kterých je již možno spojit demonstrační soustavu s programovatelným logickým automatem. Jelikož je lišta rozdělovací je možné celou soustavu možné jednoduše rozebrat a pět složit v případě poruchy.



Obr. 3.1 – Demonstrační soustava

3.2 VYTVOŘENÍ PROGRAMU DO ARDUINA DUE

Aby Arduino Due zobrazovalo stav programovatelného logického automatu, byl vytvořen jednoduchý program. Pro tvorbu programu byla použita stejnojmenná aplikace Arduino. Program je napsán v programovacím jazyku C. Před samotným programováním bylo nezbytné stáhnout knihovnu UTFT.

Postup programování probíhal následovně. Na úvod programování bylo nutné zahrnout UTFT knihovnu do programu. Dále bylo nutné inicializovat LCD displej, deklarovat fonty, které budou použity. Nyní jsou definovány proměnné, které budou v kódu vystupovat.

Dále je nastavena forma zobrazování LCD displeje na širokoúhlé, následuje definice pinů, přes které bude probíhat komunikace s programovatelným logickým automatem, na vstupní. Toto nastavení je umístěno ve smyčce setup, která probíhá pouze jednou, a to při spuštění zařízení.

Dále následuje smyčka loop, ve které je umístěn program, který se vykonává stále dokola. V úvodu se zde přečtou vstupní proměnné piny 8-11, a následně se jejich hodnoty uloží do proměnných. Poté jsou zde umístěny podmínky, ve kterých se porovnávají proměnné s kombinacemi. Pokud hodnoty proměnných odpovídají dané kombinaci, tak se provede skok do dané smyčky. V dané smyčce se ověřuje hodnota pomocné proměnné, která slouží k tomu, aby se výpis na displej provedl pouze jednou během jedné kombinace, při změně kombinace se opět hodnota pomocné proměnné změní, a tak je zápis opět povolen. Pokud je pomocná

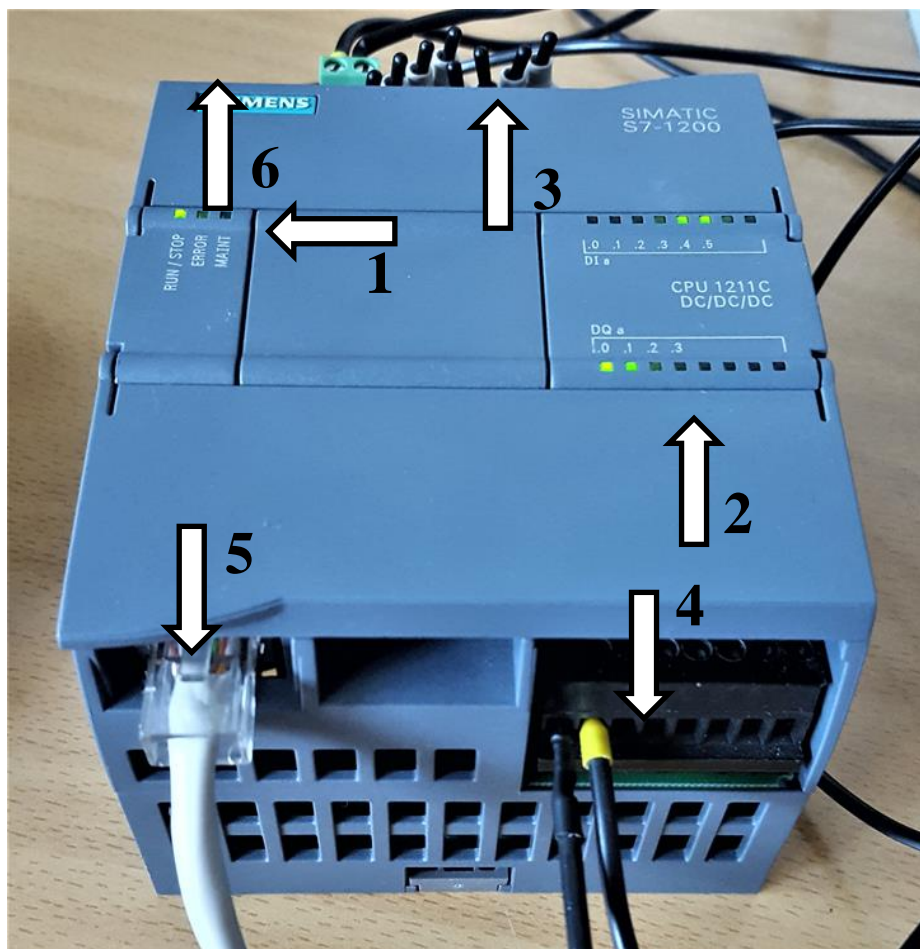
proměnná při ověřování rovna logické nule, je povolen zápis na LCD displej, kde se nejprve smaže předchozí výpis.

Po smazání se zde nastavuje barva, kterou bude mít výplň obdélníku. Výplň obdélníku zde realizuje podbarvení písma, Dále jsou zde umístěny souřadnice obdélníku. Aby byl obdélník výrazný v dalších krocích se zde nastaví barva a vykreslí ohraničení obdélníku. Do takto vykresleného obdélníku se již může přidat text, kde nastavuje velikost fontu, barva a umístění. Po provedení výpisu se již změní hodnota pomocné proměnné na logickou jedničku, tím se zamezí opakovanému vypisování stejného výpisu na displej. Následně je zde proveden slok do porovnávací smyčky, kde se opět ověřují podmínky.

Pomocí takového to principu je zde uvedeno celkem 14 možných kombinací, které zobrazují stav soustavy. Kód programu Arduina Due je uveden v příloze.

3.3 SIEMENS SIMATIC S71200

Na následujícím obrázku jsou popsány komponenty, které byly používány při vytváření návrhů řízení demonstrační soustavy.



Obr. 3.2 – Siemens Simatic s71200

- 1 Diody stavu CPU
- 2 Diody signalizující stav vstupů/výstupů
- 3 Vstupy
- 4 Výstupy
- 5 Programovací rozhraní
- 6 Napájecí konektor

3.4 SPOJENÍ DEMONSTRAČNÍ SOUSTAVY S PLC

Napájení programovatelného logického automatu probíhá pomocí napěťového zdroje VOLCRAFT. FSP 2405, který dodává napětí o velikosti 24 V. Jak již bylo výše zmíněno, komunikace programovatelného logického automatu s demonstrační soustavou zde probíhá pomocí celkem dvanácti vodičů, kde jsou dva vodiče použity na přivedení napájení do soustavy. Kladné napětí slouží k realizování logické jedničky a je tak přivedeno do spínačů. A druhý vodič uzemňovací je zde použit jako společný vodič, který slouží pro spojení demonstrační

soustavy s programovatelným logickým automatem, protože každé zařízení má vlastní zdroj napájení.

Tři vodiče zde slouží pro připojení vstupních kontaktů PLC automatu s páčkovými přepínači. Pomocí následujících tří vodičů jsou spojeny vstupní kontakty PLC automatu se spínacími tlačítky. A poslední čtyři vodiče slouží pro přivedení výstupních hodnot programovatelného logického automatu do demonstrační soustavy, kde se na základě kombinace těchto výstupů vypíše stav soustavy. Do PLC automatu bylo ještě potřeba přivést na výstupní stranu napětí, pomocí kterého se zde budou realizovat kombinace na výstupních kontaktech. Na vstupní stranu bylo nutné přivést také napětí, které se bude porovnávat se vstupními hodnotami, a bude se tak vybírat mezi logickou jedničkou, nebo nulou.

Tab. 3.1 – Spojení demonstrační soustavy s PLC

Siemens Simatic s71200	Demonstrační soustava
Vstup .0	6
Vstup .1	7
Vstup .2	8
Vstup .3	9
Vstup .4	10
Vstup .5	11
Výstup .0	2
Výstup .1	3
Výstup .2	4
Výstup .3	5

Tab. 3.2 – Spojení programovatelného logického automatu a napěťového zdroje

Siemens Simatic s71200	VOLCRAFT. FSP 2405
L+	+
M	-
1M	+

Tab. 3.3 – Spojení demonstrační soustavy a napěťového zdroje

Demonstrační soustava	VOLCRAFT. FSP 2405
1	-
13	+

Takto spojená zařízení jsou již připravena k zobrazování a simulování funkce na demonstrační soustavě.

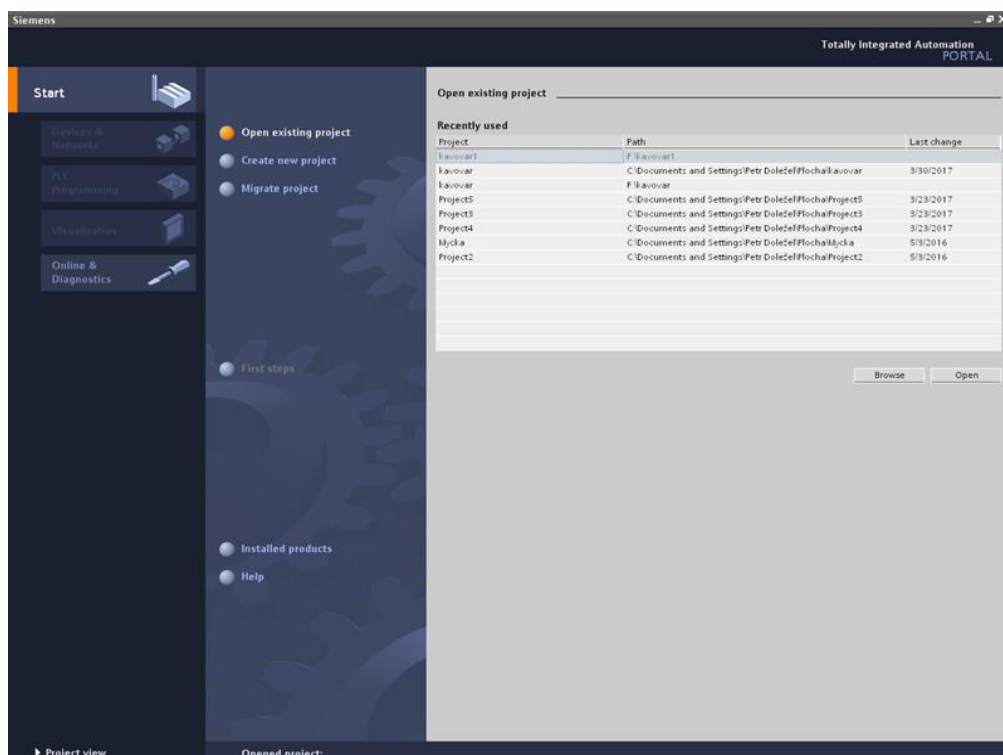
3.5 TOTALY INTEGRATED AUTOMATION PORTAL V10

Programování logického automatu zde probíhalo pomocí UTP kabelu. Po připojení programovatelného logického automatu s počítačem se automaticky nainstaloval ovladač, který zaručoval správnou komunikaci, nebylo tak potřeba nastavovat potřebné IP adresy. Samotné programování probíhalo v aplikaci Totally Integrated Automation Portal V10, která je k tomu určena výrobcem.



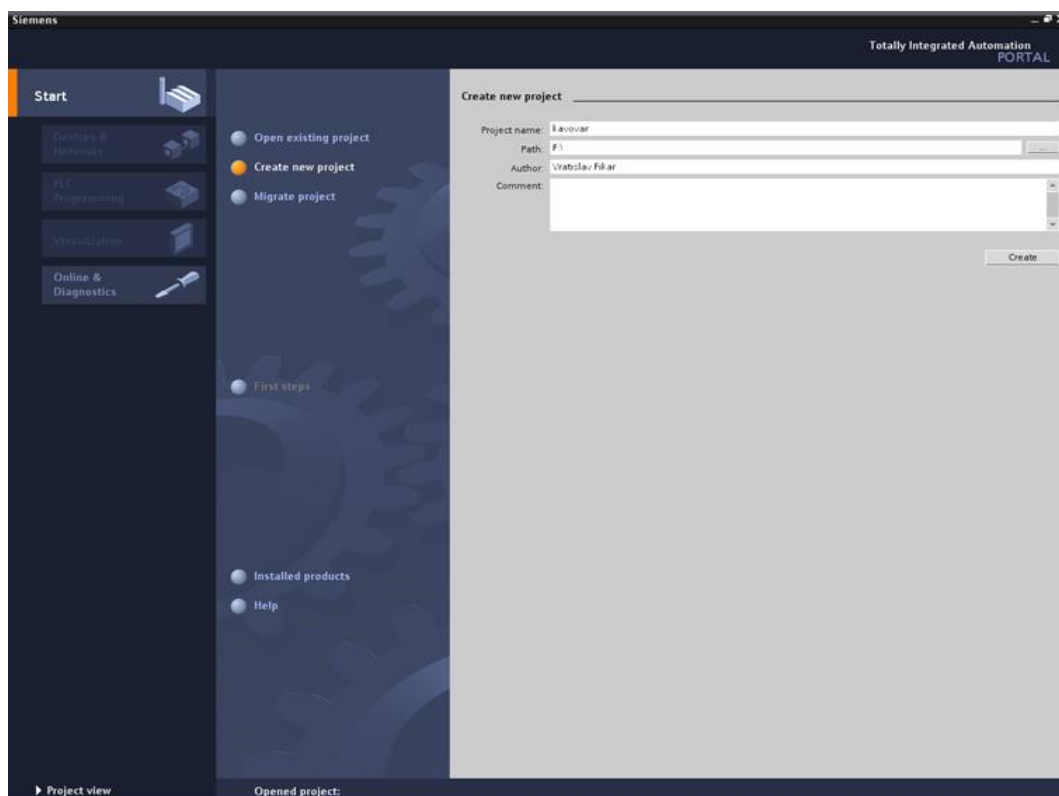
Obr. 3.3 – Ikona TIA Portal

Při spuštění programu se zobrazí startovní okno, ve kterém si můžeme vybrat akci, kterou chceme provádět. Máme na výběr tvorbu nového projektu, otevření již existujícího projektu, nebo možnost přenesení projektu.



Obr. 3.4 – Startovní okno

Při vytváření nového projektu je třeba určit jeho název, a místo kam bude projekt uložen. Dále je možné určit autora projektu.



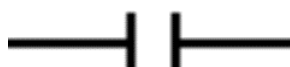
Obr. 3.5 – Tvorba projektu

Jak je vidět na obrázku, já jsem pojmenoval projekt jako kavovar, jelikož vytvořená demonstrační soustavu bude simulovat stavy kávovaru. Jako místo uložení jsem použil flash disk.

V dalším kroku bylo potřeba vybrat z nabídky používaný programovatelný logický automat. V mém případě se jednalo o Siemens Simatic S7-1200. Nyní již bylo vše připraveno k samotnému programování.

3.6 POUŽITÉ BLOKY LADDER DIAGRAM

Při vytváření programu pro demonstrační soustavu byly použity tyto bloky.



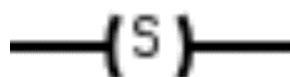
Obr. 3.6 – Schématická značka spínacího kontaktu

Těsně nad každou schématickou značkou se nachází název, podle kterého je jednoduché zjistit o jaký spínací kontakt se jedná. Nad názvem je uvedena adresa kontaktu.



Obr. 3.7 – Schématická značka rozpínacího

V případě, že platí nějaká daná podmínka kontakt se rozepne.



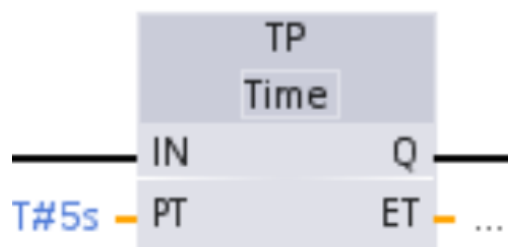
Obr. 3.8 – Schématická značka nastavení cívky

Při přivedení signálu na tento blok je cívka nastavena, a to i když už není signál přiváděn do tohoto bloku.



Obr. 3.9 – Schématická značka resetování cívky

Tento blok je opakem předchozího bloku, slouží pro resetování nastavené cívky.



Obr. 3.10 – Schématická značka zpožďovače

Jako posledním používaným blokem v programu je zpožďovač, který jak je již z názvu jasné, zpozdí vstupní signál o daný čas až po té signál propustí.

Toto jsou všechny potřebné bloky, které byly použity při vytváření programů. Pomocí různých kombinací bloků byly vytvořeny demonstrační programy.

3.7 FUNKCE PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU

Hlavní funkce programu byly, upozornění v případě docházení náplní (vody i kávy). Dále aby bylo na výběr ze tří různých typů káv, zde je na výběr Ristretto, Espresso a Lungo.

Každá káva má trochu jinou přípravu. Ristretto je nejsilnější káva, kde voda protéká namletou kávou 18 s. Espresso je nejoblíbenější kávou, která není tak silná a voda protéká mletou kávou 25 s. A poslední Lungo je nejslabší káva. Má totožnou přípravu s Espressoem, tam je nakonec přidána voda o stejném objemu natečené kávy.

Při výběru z jedné z těchto káv je spuštěn mlýnek, který namele potřebnou váhu kávy, všechny kávy jsou připravovány ze 7 gramů. Po namletí je spuštěn průtok vody kávou, zde záleží na typu kávy. Po natečení kávy je použitá káva vyhozena do výměnného zásobníku.

3.8 PROGRAMOVÁNÍ PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU

Na úvod programu bylo nutné naprogramovat kontrolu stavu náplní. V případě, že dojde voda v kávovaru jsou všechny vstupy nastaveny na logickou nulu a na displeji Arduina se v případě této kombinace vypíše hláška, že není voda. Při docházení kávy se vypíše hláška, která na toto upozorní. Tento stav má kombinaci výstupních proměnných 0001. V případě že jsou obě náplně vyprázdněné, je vypsána hláška „není káva i voda“, tato hláška je vypsána na Arduinu při kombinaci 0010. Další kombinací je 0011, při které se zobrazí nabídka káv.

Při výběru první kávy (Ristretta) kombinace 0101 značí, že byla vybrána tato káva a že byl spuštěn mlýnek. Po ukončení mletí následuje kombinace výstupních proměnných 0110, při které nás Arduino pomocí LCD displeje informuje, že dochází k vytékání kávy. Tato kombinace je po dobu 18 s. Poté je káva dokončena a použitá namletá káva je vyhozena. Tomuto upozornění slouží kombinace 0111. Celá soustava se tak vrátí do stavu výběru kávy.

Při vybrání druhé kávy se výstupy upraví do stavu 1001 a na displej se vypíše že byla vybrána káva Espresso a byl spuštěn mlýnek. Po namletí se po dobu 25 s zobrazuje, že dochází k vytékání druhé kávy. Tomu odpovídá kombinace 1010. Dále následuje kombinace 1011, která informuje o dokončení kávy, a zároveň o vyhození použité kávy.

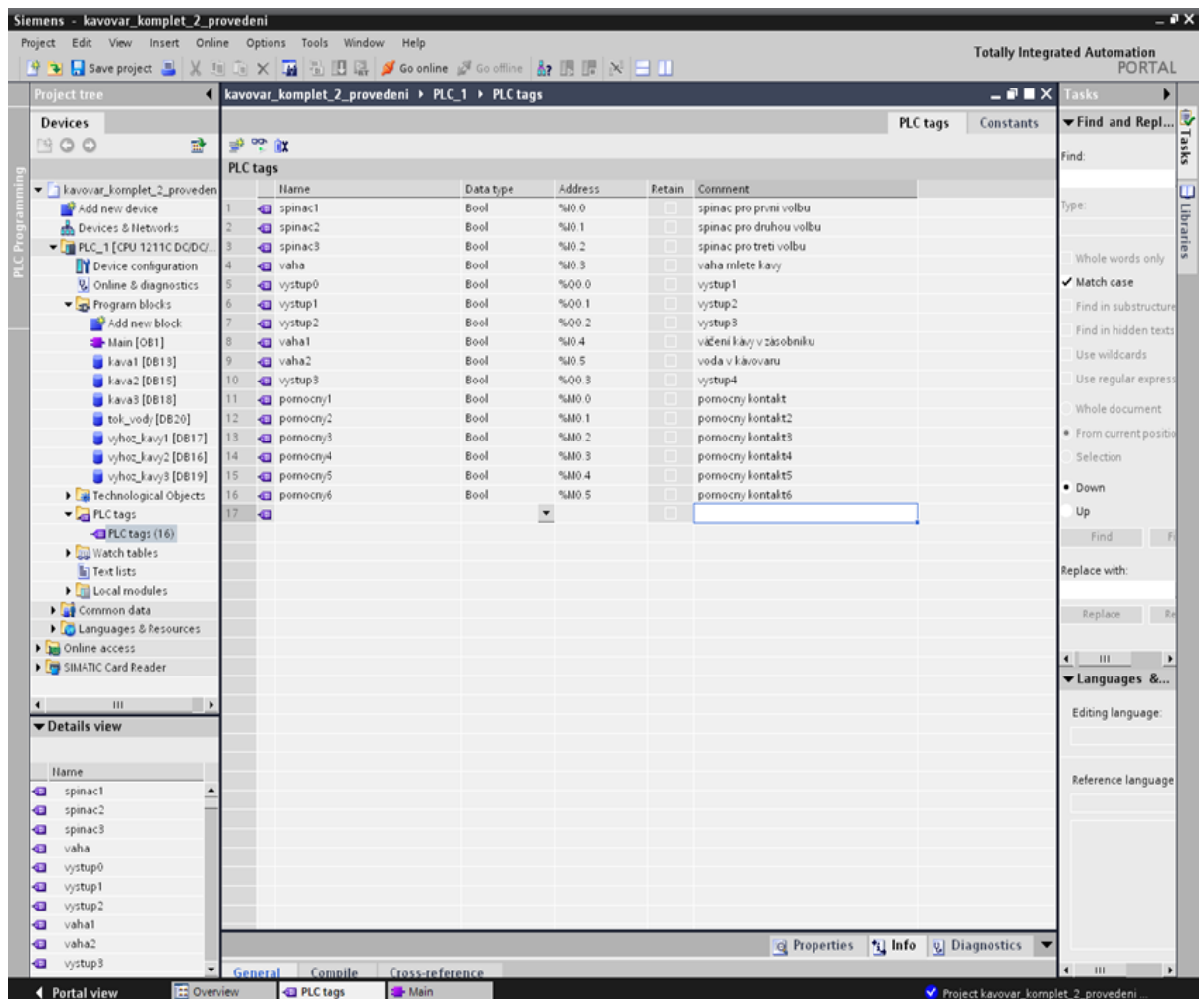
U poslední kávy je tomu obdobně, spuštění mlýnku značí kombinace 1101, o vytékání kávy jsme informováni při kombinaci 1110, kde je následně nastavena kombinace 1100, při níž dochází k vytékání vody. A poslední kombinace 1111 informuje o dokončení kávy. Všechny kombinace zobrazuje níže uvedená tab. 3.4.

Tab. 3.4 – Kombinace výstupů

0000	Není voda
0001	Není káva
0010	Není káva i voda
0011	Nabídka káv
0101	První káva, mletí
0110	První káva, teče
0111	První káva dokončena, výhoz použité kávy
1001	Druhá káva, mletí
1010	Druhá káva, teče
1011	Druhá káva dokončena, výhoz použité kávy
1101	Třetí káva, mletí
1110	Třetí káva, teče
1100	Třetí káva, teče voda
1111	Třetí káva dokončena, výhoz použité kávy
0100	Nevyužitá kombinace
1000	Nevyužitá kombinace

Na základě znalosti těchto kombinací jsem začal vytvářet první verzi programu, která bude obsahovat všechny použité kombinace.

Programování logického automatu bylo provádělo v grafickém programovacím jazyku Ladder Diagram, kde se zvlášť definují potřebné proměnné v záložce PLC tags. Zde jsem si připravil pomocné proměnné, vstupní i výstupní proměnné. Vytváření je možné dvěma způsoby. Prvním způsobem je vyplňování přímo do seznamu, kde je potřebné znát adresy proměnných. Druhým způsobem je vytváření pomocí záložky properties, kde není nutné znát přesně adresu, protože se zde vybírá z nabídky.



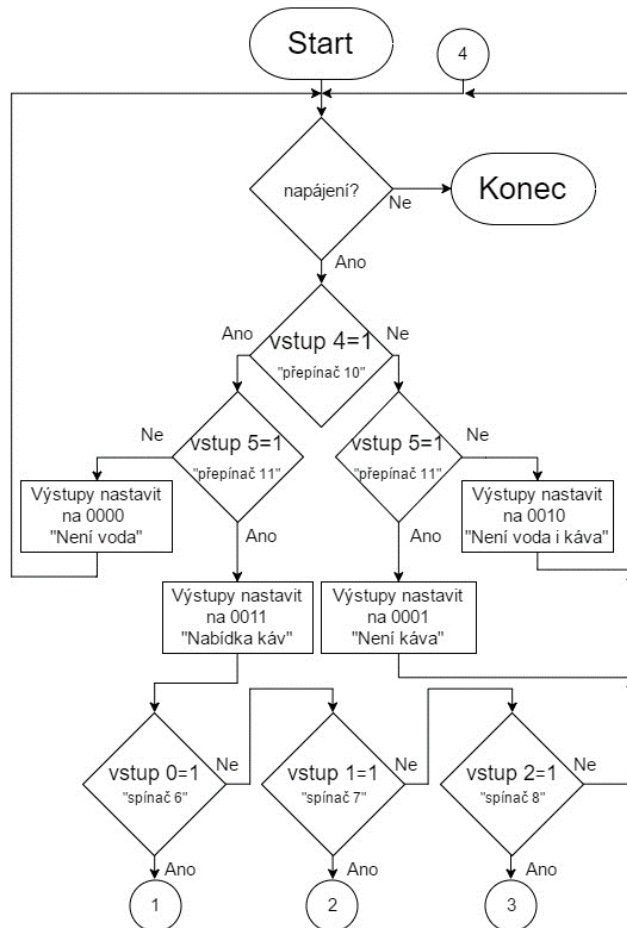
Obr. 3.11 – Definování proměnných

Zde se nastavovalo jméno proměnné, datový typ, kde jsem zvolil, že se jedná o proměnnou druhu Booleovy algebry. Dále bylo třeba nastavit adresu (vstupní proměnné značené písmenem I, výstupní proměnné jsou označovány písmenem Q a pomocné proměnné jsou značeny M). Zbytek adresy odpovídá daným pomocným proměnným, vstupům nebo výstupům, které jsou přímo vyvedené na svorkovnici. Dále je možné si k proměnným přidat krátký komentář.

Při takto připravených proměnných jsem začal vytvářet program pro řízení demonstrační soustavy. Program se vytváří v záložce Program blocks, kde je již předpřipravena smyčka Main. Programování probíhá přetahováním kontaktů do této smyčky, ve které můžeme umisťovat prvky do jednotlivých sítí.

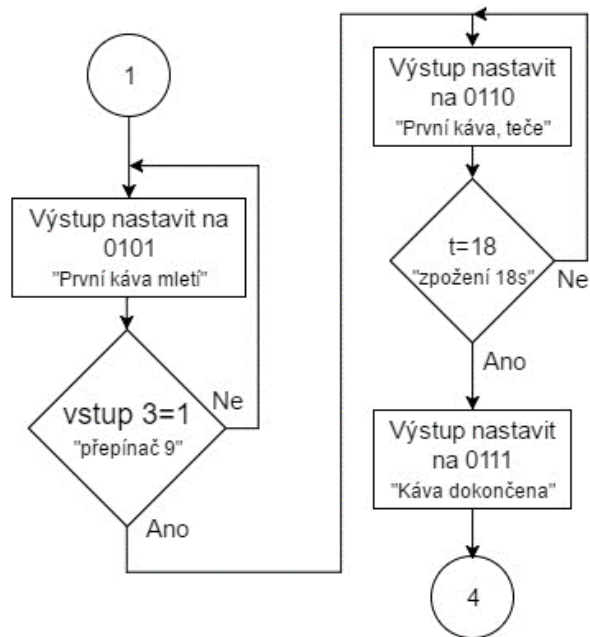
První síť obstarává kontrolu náplní. Tyto náplně jsou simulovány páčkovými přepínači demonstrační soustavy, které jsou přiváděny na vstupní kontakty .4 a .5. V případě, že ani jeden z uvedených vstupních kontaktů není v logické jedničce je nastaven výstup .1 na logickou

jedničku, ostatní výstupy jsou resetovány. V případě logické jedničky na vstupu .4 se nastaví kombinace výstup .0 na logickou jedničku, a ostatní výstupy jsou resetovány. Při logické jedničce na vstupu .5 jsou všechny výstupy resetovány. Když jsou logické jedničky na obou vstupech .4, .5 jsou nastaveny výstupy .0, .1 na logické jedničky, a ostatní jsou resetovány.



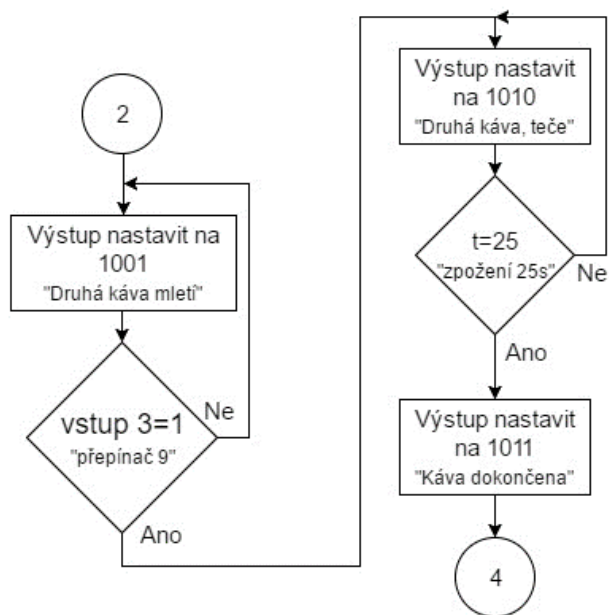
Obr. 3.12 – První síť

V druhé síti pojmenované kaval se při nastavených vstupech .0, .4 a .5 nastaví pomocná cívka, kde její spínací kontakt má funkci přídržného kontaktu. Program tak setrvá v této poloze i když, už nejsou tyto všechny podmínky dodrženy. Výstup .0 a .2 jsou nastaveny na logické jedničky. V tomto stavu soustava setrvá až do příchodu logické jedničky na vstup .3. Ta způsobí to, že se nastaví cívka druhé pomocné a spínací kontakt této proměnné umožní aktivaci třetí větve, ve které se výstupy .1 a .2 nastaví a ostatní se resetují do výchozího nastavení, do logické nuly. Tento výstupní stav je po 18 s změněn stav, při kterém jsou výstupy .0, .1 a .2 nastaveny na logickou jedničku. Po dvou s dojde k nastavení do původního stavu, a to do stavu, kterému odpovídá stav náplní.



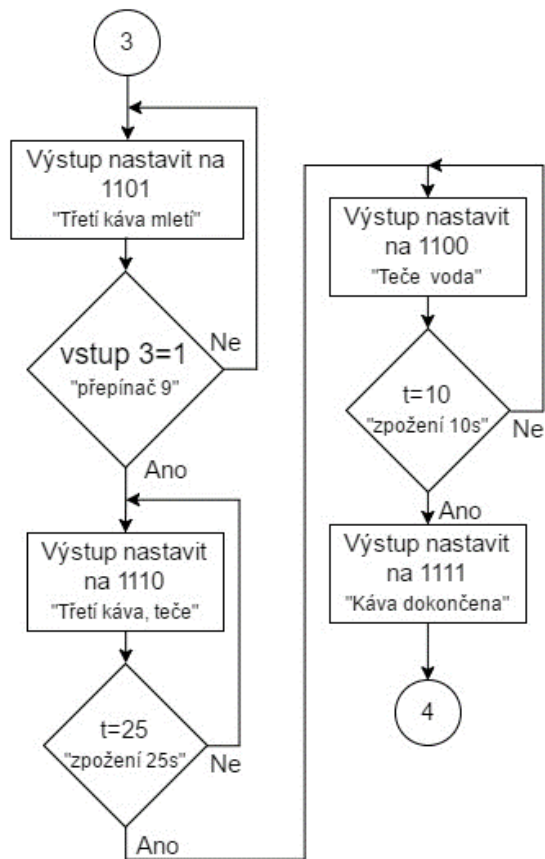
Obr. 3.13 – Druhá síť

Třetí síť je obdobná druhé síti s tím rozdílem, že nastavují jiné výstupní kombinace. Při příchodu logické jedničky na vstup .1 dojde k porovnání, zda jsou i vstupy na kterých se testují naplně a to .4 a .5. Následně je sepnuta cívka třetí pomocné. Spínací kontakt pomocné proměnné opět přidrží větve v sepnutém stavu a následně dojde k nastavení výstupů .0 a .3 do logických jedniček. Ostatní dva výstupy jsou resetovány. Při příchodu logické jedničky na vstup .3 dojde k aktivaci cívky čtvrté proměnné. Spínací kontakt čtvrté proměnné umožní spuštění třetí větve, kde jsou po dobu 25 s výstupy .1 a .3 nastaveny a ostatní resetovány. Po uplynutí doby dojde ke změně výstupů, kde se k předešlé kombinaci nastaví ještě výstup .0 do logické jedničky. V tomto stavu výstupy setrvávají po dobu 2 s a poté jsou opět nastaveny do stavu odpovídajícímu náplni.



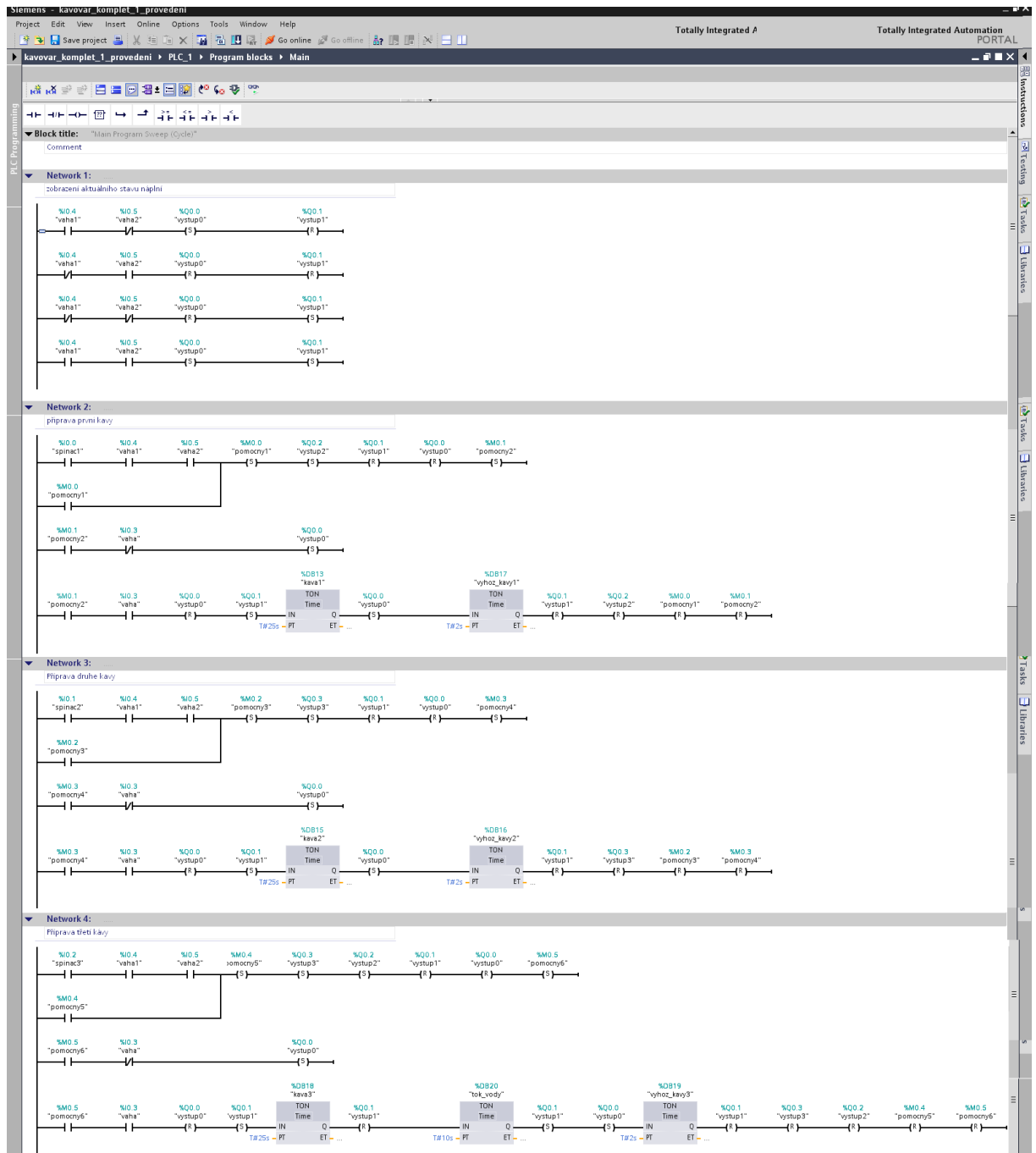
Obr. 3.14 – Třetí síť

Čtvrtá síť je aktivována při příchodu logické jedničky na vstup .2, dojde k nastavení cívky páté pomocné proměnné. Spínací kontakt této pomocné způsobí, že všechny výstupy jsou nastaveny krom výstupu .1, který je v logické nule. S příchodem logické jedničky na vstup .3 dojde k nastavení šesté pomocné cívky, čímž se způsobí aktivace třetí větve v této síti a tím i ke změně výstupů. Výstupy jsou všechny nastaveny krom výstupu .0. Tento stav trvá 25 s, následně je po dobu 10 s nastavena kombinace výstupů, ve které jsou nastaveny pouze výstupy .2 a .3. Následuje 2 s kombinace kdy jsou všechny výstupy nastaveny a následně vrácení do stavu odpovídajícímu náplní.



Obr. 3.15 – Čtvrtá síť

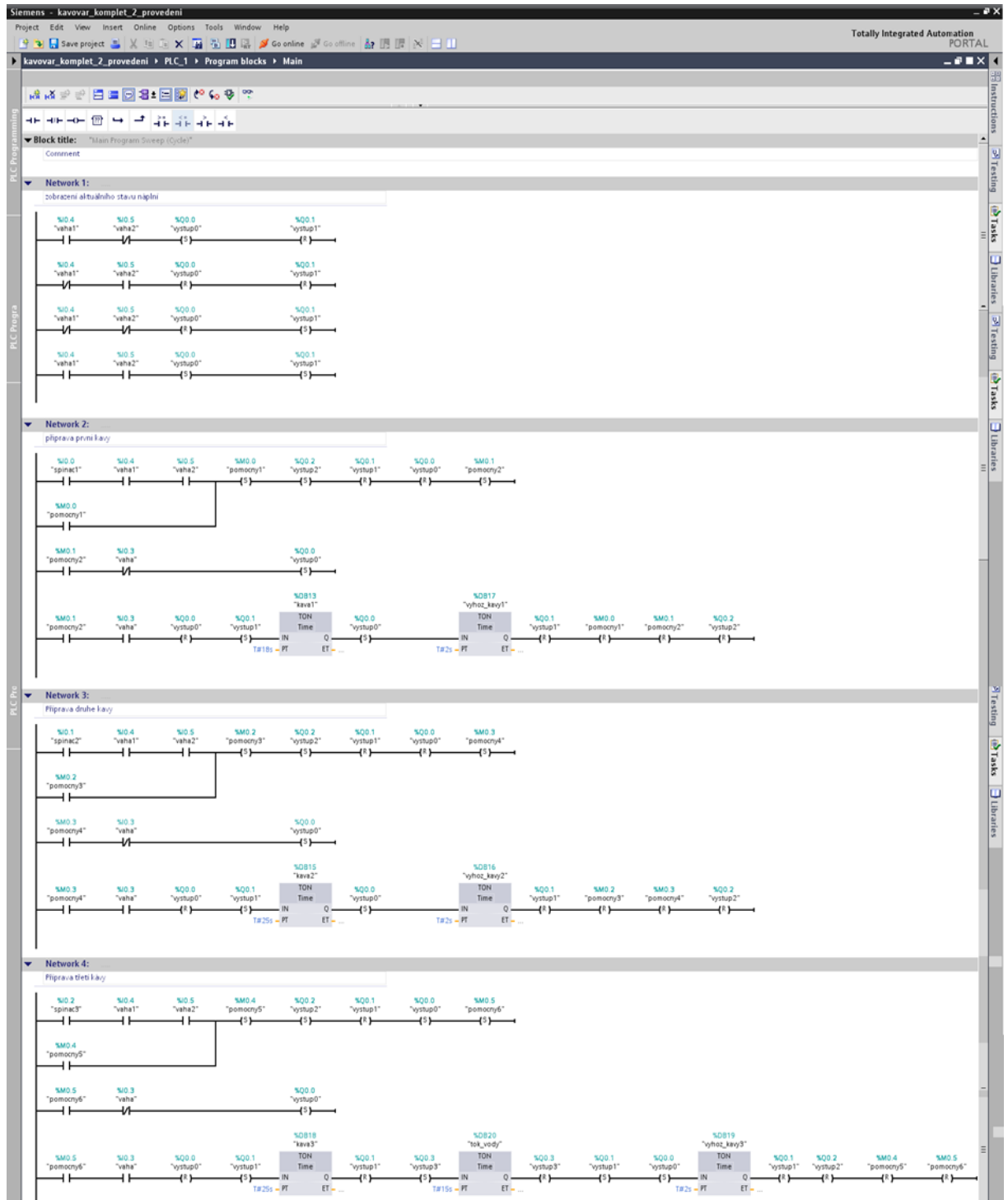
Pro časování jsou zde použity zpoždovače. Takto sestavený kód využívá všechny výstupní kombinace definované v Arduino. Po celou dobu běhu tohoto programu víme, v jaké části programu se soustava nachází.



Obr. 3.16 – První verze programu

Druhá verze možného programu je velice podobná první verzi. Jsou zde jen určitá zjednodušení, a to v části, kdy dochází k mletí kávy. Tento stav je u všech káv stejný, kdy jsou nastaveny pouze výstupy .0 a .2. Nerozlišuje se zde, jaká káva je připravována. Stejně tak tomu je i při vytékání kávy. Zde jsou nastaveny výstupy .1 a .2 pro všechny tři kávy. A poslední zjednodušení je u oznámení že je káva hotová a výhozu použité namleté kávy. Zde je výstupní kombinace pro všechny kávy nastavena na .0, .1 a .2.

Tímto došlo k zjednodušení kódu, protože po vytvoření pro první kávu bylo možné stejný kód rozkopírovat do ostatních dvou. Z původně použitých 14 kombinací se tak kód zjednodušil na 10.



Obr. 3.17 – Druhá verze programu

Třetí verze programu je nejvíce zjednodušená. Může zde nastat pouze pět stavů, prvním stavem je, když není dostatek vody v kávovaru. V tomto případě je nastavena stejná kombinace jako u předchozích programů. Dalším stavem je, že není dostatek kávy. Třetím možným stavem je že jsou náplně v pořádku a na Arduinu se zobrazí nabídka. Čtvrtý stav nastává po výběru kávy, při kterém dochází k mletí kávy. Posledním stavem je tečení kávy. Kombinace výstupů odpovídají kombinacím použitým v předchozích verzích, a to kvůli správnému zobrazování na LCD displej. Dále je zde změna v přípravě třetí kávy, do které se již nepřidává zvlášť voda, ale nechává se déle protékat skrz kávu.

Siemens - kavovar_komplet_3_provedeni

Totally Integrated Automation PORTAL

kavovar_komplet_3_provedeni PLC_1 Program blocks Main

Block title: "Main Program Sweep (Cycle)"

Network 1: zobrazeni aktualního stavu naplni

Network 2: nabídka

Network 3: první káva

Network 4: tok kávy1

Network 5: druhá káva

Network 6: tok kávy2

Network 7: třetí káva

Network 8: tok kávy3

Main

General Compile Cross-reference Syntax

Portal view Overview Main PLC tags zpozdovac1 Loading completed (errors: 0, warnings: 0)

Obr. 3.18 – Třetí verze programu

4 ZHODNOCENÍ

Sestavená demonstrační soustava slouží k simulaci logického řízení. Dále zobrazuje na LCD displeji stav kombinací výstupů programovatelného logického automatu, které odpovídají daným stavům soustavy. Každá z vytvořených verzí programu dokáže řídit demonstrační soustavu.

5 ZÁVĚR

Cílem práce bylo vytvořit demonstrační soustavu, která by sloužila jako názorná ukázka, jak funguje logické řízení za použití programovatelných logických automatů.

Demonstrační soustava obsahuje ovládací část za pomoci spínacích a páčkových přepínačů. Zároveň vizualizuje svůj stav na displeji. Při vytváření soustavy byly použity kvalitní materiály a tím by měla být zaručena časová stálost a spolehlivost. Celá soustava je uzavřená, čímž by měla být zaručena i bezpečnost. Bylo zjištěno, že soustava musí být externě napájena pomocí USB, protože se do pouzdra soustavy nevešel dostatečně velký chladič, který by uchládl stabilizátor, který obstarával snížení napětí z 24 V na 5 V.

Všechny stavy, které může soustava vizualizovat, dostačují k jasnému popisu stavů, ve kterých se demonstrační soustava může nacházet.

Součástí bakalářské práce jsou tři návrhy řízení demonstrační soustavy.

První návrh řízení soustavy obsahuje všechny možné stavy, které mohou nastat během provozu. Zároveň je na displeji zobrazován aktuální stav, přesně v jakém stavu se soustava nachází.

Druhý návrh řízení je zaměřen na to, že lze omezit počet výstupních kombinací, a to za vykonávání stejných akcí jako u předešlé soustavy. Nicméně po výběru již není přesně zobrazováno, ve které síti (jaká káva je připravována) se soustava nachází.

Třetí návrh řízení slouží jako ukázka nejjednoduššího možného řízení demonstrační soustavy. Jsou zde použity jen nejnútnější prvky pro správný chod demonstrační soustavy.

Všechny vytvořené návrhy řízení demonstrační soustavy byly zároveň odzkoušeny na vytvořené soustavě.

Zároveň byla popsána práce v programovacím jazyku Ladder diagramu, ve kterém se použitý programovatelný logický automat Siemens Simatic s71200 programoval.

LITERATURA

- DOČEKAL, A. 2010. *Ukázkové úlohy pro výuku automatizace* [online]. Vysoké učení technické v Brně, Vedoucí: Olga Davidová. [cit. 23. 4. 2017].
Dostupné z: http://autnt.fme.vutbr.cz/szz/2010/BP_Docekal.pdf
- DVOŘÁČKOVÁ, P. 2012. *Booleova algebra* [online]. Masarykova univerzita, Brno, Vedoucí: Jaroslav Beránek, [cit. 26. 4. 2017].
Dostupné z: https://is.muni.cz/th/322248/pedf_b/Booleova_algebra.pdf
- KOCHANÍČEK, L. Programovací jazyky pro PLC. *elektrotechnika* [online]. 8. 4. 2010 [cit. 23. 4. 2017]. Dostupné z: <https://coptkm.cz/portal/?doc=3905>
- KOLÁŘ, V. 2006. *Logické řízení* [online]. Ostrava: Katedra obecné elektroniky. [cit. 23. 4. 2017].
Dostupné z: http://fe1.vsb.cz/kat420/vyuka/hgf/rozvody_lomy/05_logicke_rizeni.pdf
- Kompaktní a modulární PLC. *ELUC* [online]. [cit. 23. 4. 2017]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/970>
- LUJCIK, L. 2013. *Digitální obvody* [online]. [cit. 26. 4. 2017].
Dostupné z: <http://www.umel.feec.vutbr.cz/BDIO/predn/BDIO06.pdf>
- MARTINÁSKOVÁ, M. 2001. PLC-programovatelné automaty. *109 Laboratoř průmyslové automatizace* [online]. Přednáška. [cit. 23. 4. 2017].
Dostupné z: http://iat.fs.cvut.cz/109/files/psar/prednasky_2007/plc_vznik_hw_sw.ppt
- NAVRÁTIL, P. 2011. *AUTOMATIZACE Vybrané statě* [online]. Univerzita Tomáše Bati ve Zlíně, [cit. 11. 5. 2017]. Dostupné z: http://www.utb.cz/file/13883_1_1
- SVOBODA, O. 2012. *Model křižovatky pro výuku Programovatelné automaty* [online]. Vysoká škola polytechnická Jihlava, Vedoucí: František Zezulka, [cit. 29. 4. 2017].
Dostupné z: <https://is.vspj.cz/bp/get-bp/student/27029/thema/2216>
- ŠIŠLÁK, D. 2003. *Řízení výtahu* [online]. České vysoké učení technické v Praze, Vedoucí: Richard Šusta, [cit. 29. 4. 2017]. Dostupné z: https://support.dce.felk.cvut.cz/mediawiki/images/e/e4/Dp_2003_sislak_david.pdf
- ŠŤASTNÝ, M. 2013. *Realizace laboratorní úlohy pro řízení motoru pomocí PLC* [online]. Univerzita Pardubice, Vedoucí: Libor Kupka, [cit. 23. 4. 2017].

Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/52240/StastnyM_RealizaceLaboratorni_LK_2013.pdf?sequence=4&isAllowed=y

VANDOREN, A. 2010. Dopředné řízení. *Control Engineering*. 30. 5. [cit. 23. 4. 2017].

Dostupné z: http://www.controlengcesko.com/index.php?id=47&no_cache=1&tx_ttnews%5Btt_news%5D=3462&cHash=8f43917663&type=98

VODA, Z. 2014. Seznámení s Arduinem. *ARDUINO.CZ* [online]. 11.10. [cit. 26. 4. 2017].

Dostupné z: <https://arduino.cz/seznameni-s-arduinem/>

PŘÍLOHY

A – CD

Příloha k bakalářské práci
Demonstrační soustava pro logické řízení
Vratislav Fikar

CD

Obsah

- 1 Text bakalářské práce ve formátu PDF
- 2 Programy tří možných řešení řízení PLC Siemens Simatic s71200 v prostředí TIA Portal
- 3 Program k zobrazování stavů pro Arduina Due