



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**APLIKACE ŘÍDICÍCH SYSTÉMŮ OD SPOLEČNOSTI  
AMIT**

APPLICATION OF CONTROL SYSTEMES FROM AMIT

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Jiří Hula

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. et Ing. Stanislav Lang, Ph.D.

BRNO 2023





## Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Jiří Hula</b>
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	<b>Ing. et Ing. Stanislav Lang, Ph.D.</b>
Akademický rok:	2022/23

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

### Aplikace řídicích systémů od společnosti AMiT

#### Stručná charakteristika problematiky úkolu:

Práce je věnována problematice využití programovatelných řídicích systémů v oblasti průmyslové automatizace. Vzhledem ke skutečnosti, že jsou v České republice vyvíjeny a vyráběny kvalitní řídicí systémy, bude se práce věnovat přednostně řídicím systémům vyvíjených tuzemskými společnostmi. V praktické části práce se student seznámí s konkrétními řídicími prostředky od firmy AMiT, spol. s r.o. a pro vybraný řídicí systém vytvoří jednoduchý demonstrační program řízení reálného či fiktivního zařízení.

#### Cíle bakalářské práce:

Proveďte stručnou rešerši v oblasti řídicích systémů.  
Proveďte základní průzkum trhu v oblasti řídicích systémů (zaměřte se na tuzemské výrobce).  
Nastudujte řídicí systémy od společnosti AMiT.  
Vytvořte demonstrační řídicí program pro modul AMiRiS99.  
Realizujte propojení řídicího modulu AMiRiS99 s dotykovým ovládacím panelem AMR-OP83.  
Vytvořte jednoduchou uživatelskou vizualizaci pro modul AMR-OP83.  
Vytvořte jednoduchý návod pro práci s modulem AMiRiS99. (volitelné)  
Stručně zhodnoťte dosažené výsledky a poznatky.

#### Seznam doporučené literatury:

ZEZULKA, František. Průmyslová automatizace: (teze přednášky k profesorskému jmenovacímu řízení). Brno: VUTIUM, 2000. Vědecké spisy Vysokého učení technického v Brně. ISBN 80-214-1634-3.

ZEZULKA, František, Jan PÁSEK, a Milan FINDURA a Jaroslav PREČAN. Automatizace procesů I: Úroveň procesního řízení [online]. Skriptum. VUT FEKT, 2014 [cit. 2022-10-10]. Dostupné z: <https://www.vutbr.cz/studenti/predmety/detail/176270>.

AMiT Automation [online]. [cit. 2022-10-10]. Dostupné z: <https://amitotation.cz/>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

---

doc. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Práce je věnována problematice programování řídicích systémů, zejména PLC, a to s akcentem na možnost využití produktů české výroby. Po stručném úvodu do tématu řídicích systémů se práce v teoretické části věnuje průzkumu českého trhu. Detailnější výklad je proveden u produktů společnosti AMiT. Vybrané produkty výše zmíněného výrobce jsou využity v rámci praktické části, která demonstruje využití systémů AMiRiS99 a AMR-OP83 na jednoduché úloze řízení světelné signalizace.

## **ABSTRACT**

The work is devoted to the problem of programming control systems, especially PLC, with an emphasis on the possibility of using products of Czech manufacture. After a brief introduction to the topic of control systems, the theoretical part of the thesis is devoted to a survey of the Czech market. A more detailed explanation is made of AMiT products. Selected products of the above mentioned manufacturer are used in the practical part, which demonstrates the use of AMiRiS99 and AMR-OP83 systems on a simple task of controlling traffic lights.

## **KLÍČOVÁ SLOVA**

řídicí systém, programovatelný logický automat, manuál, postup, návod, programování, průzkum českého trhu, PLC, AMiT, Unipi, TECO, Orbit Merret, AMiRiS99, AMR-OP83, Foxtrot, TC800, OMC8000, Patron, Neuron, DetStudio

## **KEYWORDS**

control system, programmable logic controller, manual, procedure, guide, programming, Czech market research, PLC, AMiT, Unipi, TECO, Orbit Merret, AMiRiS99, AMR-OP83, Foxtrot, TC800, OMC8000, Patron, Neuron, DetStudio





ÚSTAV AUTOMATIZACE  
A INFORMATIKY



2023

## **BIBLIOGRAFICKÁ CITACE**

HULA, Jiří. *Aplikace řídicích systémů od společnosti AMiT*. Brno, 2023. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/149375>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Stanislav Lang.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 22. 5. 2023

.....

Jiří Hula





# OBSAH

<b>1</b>	<b>ÚVOD.....</b>	<b>13</b>
<b>2</b>	<b>ŘÍDICÍ SYSTÉM.....</b>	<b>15</b>
2.1	Funkce PLC .....	16
2.1.1	Struktura PLC .....	17
2.1.2	Vstupy a výstupy .....	17
2.1.3	CPU a paměť .....	18
2.1.4	Komunikace.....	18
2.2	Dělení PLC .....	19
2.2.1	Modulární provedení .....	19
2.2.2	Kompaktní provedení .....	19
2.3	Programování.....	19
<b>3</b>	<b>ŘÍDICÍ SYSTÉMY SPOLEČNOSTI AMIT .....</b>	<b>21</b>
3.1	ADiR.....	21
3.2	AMiNi5D .....	22
3.3	AMiRiS99W3 .....	23
3.4	ACOS200.....	24
3.5	ADiS .....	25
3.6	APT1000G .....	27
3.7	AMR-OP87/V .....	27
<b>4</b>	<b>ŘÍDICÍ SYSTÉMY UNIPI TECHNOLOGY .....</b>	<b>29</b>
4.1	Unipi Neuron .....	29
4.2	Unipi Patron.....	30
<b>5</b>	<b>ŘÍDICÍ SYSTÉMY ORBIT MERRET .....</b>	<b>33</b>
5.1	OMC 8000 .....	33
<b>6</b>	<b>ŘÍDICÍ SYSTÉMY SPOLEČNOSTI TECO .....</b>	<b>35</b>
6.1	Tecomat Foxtrot 2 .....	35
6.2	Tecomat TC800 .....	36
<b>7</b>	<b>APLIKACE ŘÍDICÍCH SYSTÉMŮ SPOLEČNOSTI AMIT .....</b>	<b>39</b>
7.1	Použité moduly .....	39
7.1.1	AMiRiS99.....	39
7.1.2	AMR-OP83.....	39
7.2	Použitý software DetStudio .....	39
7.3	Výběr aplikace a rozvaha .....	40
7.3.1	Popis vybrané aplikace .....	40
7.3.2	Potřebný rozsah hardwaru a jeho volba.....	40
7.4	Tvorba uživatelského programu pro AMiRiS99 (řídící část).....	41

7.4.1	Založení projektu .....	41
7.4.2	Rozvržení programu - volba jazyku, procesů, cyklické třídy .....	43
7.4.3	Definování proměnných a aliasů.....	45
7.4.4	Programování procesů.....	47
7.4.5	Přiřazení proměnných k I/O .....	51
7.4.6	Nastavení komunikace .....	53
7.4.7	Zavádění programu do jednotky AMiRiS99.....	54
7.5	Tvorba uživatelského programu pro AMR-OP83 (vizualizační část) .....	56
7.5.1	Založení projektu .....	56
7.5.2	Rozvržení programu - obrazovek.....	57
7.5.3	Nastavení komunikace .....	57
7.5.4	Tvorba obrazovky .....	58
7.5.5	Zavádění programu do jednotky .....	62
7.6	Ladění a testování .....	63
<b>8</b>	<b>ZÁVĚR.....</b>	<b>67</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>69</b>
	<b>SEZNAM OBRÁZKŮ A TABULEK.....</b>	<b>73</b>
	<b>SEZNAM ZKRATEK.....</b>	<b>75</b>
	<b>PŘÍLOHY .....</b>	<b>77</b>

# 1 ÚVOD

S řídicími systémy se v dnešní době plné automatizace a chytrých řešení setkáváme prakticky neustále, jen nejsou vždy na očích. Na rozdíl od aktivních prvků jsou totiž v pozadí celých řízených systémů. Většinou čtenářů budou známy nadnárodní zahraniční společnosti, které mají vybudovaná svá jména ve světě automatizace. Existují i výrobci českých řídicích systémů, kteří nabízejí velmi zajímavá řešení. Každý z nich má svůj specifický přístup k vývoji a produkty jsou tím velmi rozličné. Buď jde o univerzální programovatelné logické automaty (PLC), kterým se tato práce věnuje. Každý výrobce jim dává do vínku jistá specifika, která je odlišují od ostatních. Tím je může směřovat jistým směrem, tedy do oblastí použití, kde mohou právě jeho systémy nabídnout větší přidanou hodnotu. V průzkumu trhu českých výrobců PLC jsou popsány a vyobrazeny vybrané produkty a jejich specifikace. V průzkumu jsou zmíněni pouze čtyři výrobci, ačkoli se nejedná o výrobce všechny. Bohužel je tento trh opravdu široký a není reálné do této práce vměstnat všechny výrobce a zároveň udržet práci přehlednou a informačně neosízenou. Byla snaha obsáhnout alespoň ty nejvýznamnější a prezentovat jejich portfolio. Protože se druhá část práce věnuje aplikování řídicích systémů společnosti AMiT je portfolio tohoto výrobce v průzkumu detailnější.

Aplikace řídicích systémů je postavena do edukační roviny. A to zejména na přání vedoucího mé bakalářské práce. Došlo totiž ke shodě, že tímto práce získá přidanou hodnotu a třeba naleznou další využití jako případný “návod” pro další studenty nebo zájemce o programování a používání produktů společnosti AMiT. Aplikace je tedy směřována do postupu vysvětlujícího tvorbu řídicího programu pro systémy této společnosti. Postup je prezentován na jednoduchém případu tak, aby byl maximálně přehledný a přesto konkrétní.

Jedná se tedy o praktickou ukázkou celkového postupu od definování konkrétního problému, přes programování a zavedení řídicích programů do řídicích jednotek až po samotné fyzické testování. Pro aplikování jsou fyzicky použity jednotky AMiRiS99 a AMR-OP83. Aplikace zahrnuje i zpestření v podobě použití generačně odlišných modulů. Zpestření představuje programování ve dvou odlišných editorech a například odlišný přístup k nastavení vzájemné komunikace. Pro jednotku AMiRiS99 je v rámci ukázkového postupu vytvořen řídicí program prezentující řízení světelného signalizačního zařízení (SSZ) s radarovou kontrolou a přechodem pro chodce. Toto řízení je dále vizualizováno přímo na displeji jednotky AMR-OP83, která zároveň umožňuje pomocí změn proměnných ovlivňovat běh řídicího programu jednotky AMiRiS99. Toto ovlivňování reprezentuje například simulace tlačítka chodce vytvořeného na dotykovém panelu jednotky. Vzájemná komunikace použitých modulů je zprostředkována použitím linky RS485 a protokolem MODBUS RTU. Autor pro poskytnutí maximálně relevantních informací také absolvoval oficiální školení výrobce AMiT. Pro samotný vývoj a ladění řídicích programů je použit vlastní software výrobce, DetStudio.



## 2 ŘÍDICÍ SYSTÉM

Řídicí systémy se používají od prvopočátku automatizace technologických procesů a průmyslových strojů. Slouží k řízení daného procesu tak, aby nepotřeboval nebo alespoň maximálně nahradil lidskou obsluhu. Umožňují například řídit i tak sofistikované nebo přesnost vyžadující systémy či technologické procesy, které by jiným způsobem řídit ani nebylo možné.

Do řízení vstupují také regulátory, které se mohou označovat jako “malá automatizace”. Jejich nasazení bylo základem celé automatizace. Umožnili regulovat konkrétní veličinu v procesu podle nastavených parametrů. A to asi nejčastěji teplotu, tlak, průtok nebo třeba řídit poměry několika veličin.

Dříve řídicí systémy představovaly zejména mechanické prvky, které svým chováním definovaly potřebnou logiku nebo funkci. Velmi se využívalo reléových zapojení pro vytvoření řídicí logiky. Takovéto systémy byly tvořeny celými rozvaděči obsahujícími velké množství mechanických prvků, které nebylo jednoduché přestavět na jinou logiku. Proto byly vyvinuty řídicí systémy dnes známé jako programovatelné automaty (PLC). Ty již od počátku 70. let umožnily měnit logiku programově, a tím umožnily zrychlit úpravy nebo celé přestavby výrobních linek.

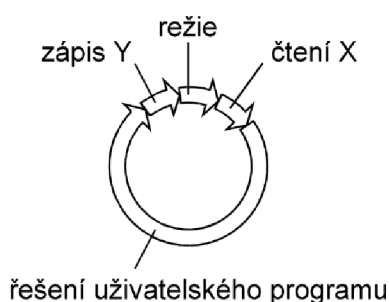
Tyto řídicí systémy jsou již elektronické, a to zejména díky pokroku v oblasti mikroprocesorové techniky a elektronických výkonových prvků. S jejich příchodem začaly vznikat také průmyslové regulátory vybavené mikroprocesory a začaly si opět získávat své místo na trhu, které pouze na okamžik zastínily PLC. Tyto průmyslové regulátory slouží k řízení jednodušších procesů a strojů, které nevyžadují příliš velkou komplexitu řízení a provádějí zejména měření sledovaných veličin a vlastní zásah provádějí nejčastěji jedním akčním členem. Tím jsou dostupnější a při použití na daný účel i aplikačně jednodušší. Jako programovatelné automaty (PLC) jsou označovány ty řídicí systémy, které mimo jiné reprezentují taktéž vysokou spolehlivost řízení a determinističnost v reálném čase. Taky jsou svými možnostmi nesrovnatelně více univerzální než regulátory. Právě zejména programovatelným logickým automatů se věnuje předložená práce.

Další variantou řídicího systému je tzv. IPC, průmyslový počítač. Což je vlastně osobní počítač (PC) provedený v souladu s průmyslovými standardy a splňující příslušné normy. Snaha konkurovat PLC a nabídnout podobný komfort programování jako mělo PC nebyla masově přijata. Avšak existují výrobci, kteří se touto cestou řízení úspěšně vydali. [1, 2]

## 2.1 Funkce PLC

Klíčem ke správné funkčnosti PLC je operační systém spouštějící úlohy na základě času, priority a události. Takový přístup umožní systému reagovat na požadavky prakticky v reálném čase. Taktéž je využíváno vnitřních časovačů a čítačů, které umožňují využít přírůstkového času a držet počty opakování konkrétní události.

PLC pracuje s daty, která získává z řízeného procesu prostřednictvím digitálních, analogových nebo jiných specializovaných vstupů, data jsou dále zpracovávána a vyhodnocována spolu s nastavenými vnitřními parametry v řídicím programu. Systém poté provede reakci na analogových nebo digitálních výstupech, aby dosáhl požadované reakce v procesu. Díky jednorázovému čtení vstupních proměnných v rámci jednoho cyklu je zamezeno vzniku hazardních stavů při řešení algoritmu uživatelského programu. To znamená, že během probíhajícího řešení nemůže dojít ke změně vstupních dat. [33]



Obrázek 1: Cyklus řešení uživatelského programu, převzato z [33]

*Popis obrázku:*

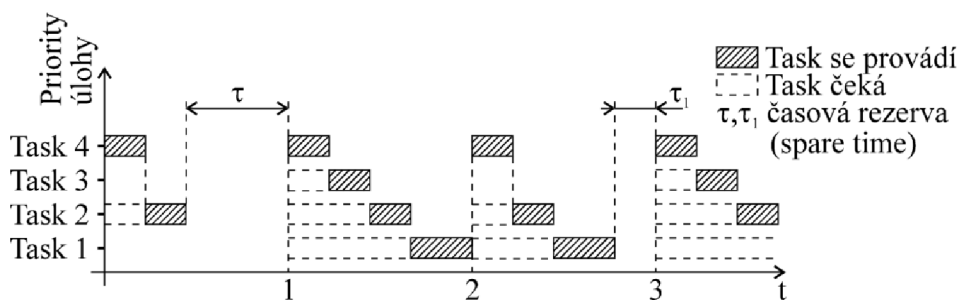
*čtení X - načtení vstupních dat do paměti*

*zápis Y - zápis dat z paměti na výstupní moduly*

*řešení - příprava centrální jednotky PLC k řešení dalšího cyklu programu*

Řídicí systémy, zastoupené PLC fungují na principu cyklického vykonávání řídicího programu, tento cyklus se stále opakuje a to s požadavkem na vysokou míru determiničnosti chování v čase. Dodržení tohoto kritéria je dosaženo využitím prioritního řízení úloh. Ty části programu, které nemusí být vykonány v reálném čase dostávají nižší priority tak, aby se důležité řídicí části programu prakticky nezpožďovaly. Jako příklad úloh s nižší prioritou je znázornění dat na HMI panelu nebo webový server. Pro zabezpečení opravdu okamžité reakce může být použito spuštění úlohy na základě události vyvolající tzv. přerušení. Příkladem může být výpadek napájení, nebo použití tlačítka central-stopu. V tento moment je systém schopen okamžitě reagovat předem naprogramovanou reakcí například k bezpečnému nouzovému zastavení procesu. A to díky vyvolanému přerušení, které pozastaví právě vykonávané úlohy a spustí příslušnou úlohu. [1, 2, 33]

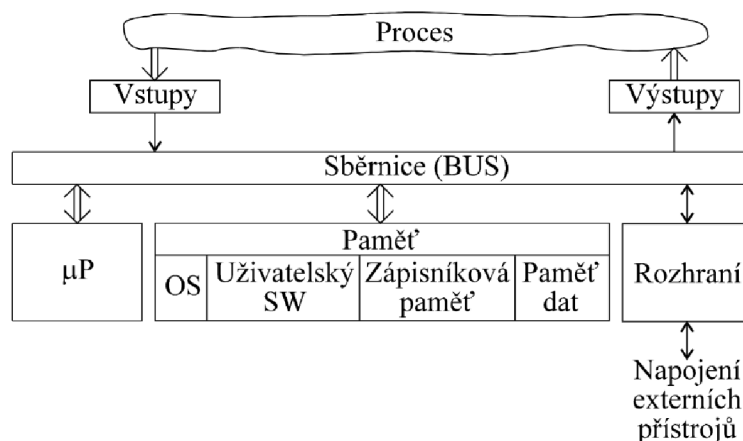




Obrázek 2: Funkce časového řízení úloh a přerušení, převzato z [1]

### 2.1.1 Struktura PLC

Jako příklad uvádím na obrázku obecnou strukturu PLC, která odpovídá klasickým modulárním konstrukcím. Konkrétní struktury budou velmi podobné, a v reálných zařízeních se budou lišit pouze odlišnou konfigurací a použitými periferiemi. [1]



Obrázek 3: Blokové schéma standardního modulárního PLC, převzato z [1]

### 2.1.2 Vstupy a výstupy

Celý podsystém vstupů/výstupů je charakteristický právě pro PLC. Je perfektně zpracovaný a podtrhuje celkovou robustnost a spolehlivost těchto systémů. Vstupy slouží systému pro zjištění hodnot sledovaných veličin řízeného procesu a výstupy jsou naopak jedinou přímou cestou jak do procesu zasahovat a řídit jej. Počty vstupů a výstupů mohou být v řádu jednotek nebo až v řádu tisíců. Jedná se o “terminály” pro připojení, které oddělují vnější signálové napětí od vnitřního napětí procesorové jednotky. V případě digitálních vstupů je rozhraním mezi obvody řídicí a výkonové části elektroniky optočlen či jiný oddělovací prvek. Tyto mechanismy galvanického oddělení poskytují vysokou odolnost citlivé části elektroniky PLC proti poškození vnějším vlivem. Vybaveny jsou i RC filtrem pro odfiltrování rušivých signálů. Při použití analogových signálů se využívá AD/DA převodníků, které jsou součástí elektroniky vstupů i výstupů.

Analogové signály přicházejí ze senzorů spojených veličin jako teplota, tlak apod. Snímače pro takové veličiny používají převod na změnu napětí případně proudu, který už PLC dokáže měřit. Analogovým signálem může PLC také řídit akční členy, které to vyžadují.

Digitální signály se používají pro zjištění nastavení stavu dvoustavových prvků. Takovými prvky jsou tlačítka, koncové spínače, zámky dveří, ventily, kontrolky, stykače, relé a další. Zvláštní případ digitálního výstupu je výstup reléový. Toto relé je přímo součástí výstupní části, využívají jej zejména kompaktní systémy.

Pro signál na digitálních vstupech a výstupech se používá většinou normalizované napětí 24 V, které je v rámci průmyslem standardizovaných napětí v oblasti logického řízení nejpoužívanější. Logické napětí nicméně může být i střídavé, nebo jiných napěťových hodnot. Pro analogové vstupy a výstupy se používá typicky napětí 0-10 V, případně proud v rozsahu 0-20 mA nebo 4-20 mA. I zde napěťový rozsah může být jiný, případně i záporný. [1, 2, 3]

### 2.1.3 CPU a paměť

Processorová jednotka je jádrem celého řízení. Tvoří majoritní část výpočetního výkonu, který má PLC k dispozici. V případě některých konstrukcí, zejména modulárních, jednotlivé segmenty I/O odbavují další mikroprocesory. Výkon CPU by měl být vhodně dimenzován vzhledem k použitému počtu I/O v systému, složitosti použité logiky a výpočtů a náročnosti sekundárních funkcí jako může být webový server nebo grafické zobrazení. Výkon CPU přímo ovlivňuje délku vykonání jednoho cyklu programu a tím i jeho rychlost odezvy a práci v reálném čase.

Využívá se nejčastěji dvou paměťových lokací. Jedna je tvořena nejčastěji již paměťmi typu FLASH nebo EEPROM a ukládá operační systém, a může umožňovat zálohování dat z RAM. Druhá paměť je systémová a využívá typu RAM. V této paměti systém neustále pracuje, paměť v sobě udržuje hodnoty registrů, hodnot I/O, čítačů a časovačů. Tato paměť může být zálohovaná baterií. [1, 3]

### 2.1.4 Komunikace

PLC zákonitě musí být vybaveno jak vnitřními, tak vnějšími komunikačními kanály. Vnitřní tvoří nejčastěji systémová sběrnice, po které komunikují všechny části PLC systému, zejména je to patrné u modulárního provedení. Pro vzájemnou komunikaci více řídicích PLC, propojení PLC se vzdálenými periferiemi, které mohou tvořit i specializovaná čidla a snímače, nebo pro spojení s nadřazenými systémy existují různé možnosti propojení. Nejčastěji k tomu slouží standardizovaná průmyslová sériová rozhraní jako RS232 či RS485, komunikační protokoly jako MODBUS nebo sběrnice PROFIBUS a PROFINET. Nejčastěji se používají mezinárodně standardizovaná řešení.

Někteří výrobci používají jiná (jejich vlastní) proprietární řešení pro vzájemnou komunikaci periferií. Pro připojení k internetu, vnitropodnikové síti nebo PC slouží klasický Ethernet IP. [1, 2]



## 2.2 Dělení PLC

PLC se dělí podle počtu vstupů a výstupů, takové rozdělení koreluje přímo s výrobními provedeními. Menší provedení s nižšími počty vstupů nebo výstupů jsou vyráběna jako kompaktní. Větší PLC s jejich velkými počty I/O mají provedení modulární. Existuje i označení jako programovatelné relé, kterým je na mysli velmi malé PLC, menší než kompaktní provedení a jeho rozsah ovládání vstupů a výstupů je pouze v řádu jednotek. [1, 2]

### 2.2.1 Modulární provedení

Modulární PLC jsou velké systémy, které se sestávají z jednotlivých modulů. Základním modulem je výpočetní CPU modul, který lze rozšiřovat o moduly s konkrétními konfiguracemi a počtem I/O. Takový systém je kdykoli plně rozšiřitelný a může disponovat i tisíci vstupy nebo výstupy. Procesor je dostatečně dimenzovaný a zabezpečuje rychlé reakce i při vysokém počtu obsluhovaných vstupů a výstupů. Takové systémy obsluhují zejména velké a složité stroje a procesy. Jsou také podstatně dražší než kompaktní systémy. [1, 2]

### 2.2.2 Kompaktní provedení

Kompaktní provedení má daný počet a konfiguraci I/O, které jsou umístěny přímo v jednom bloku i s CPU. I tato PLC je většinou možné rozšířit o další moduly I/O. Takové moduly mohou přidávat řádově desítky vstupů/výstupů navíc oproti hlavní kompaktní jednotce. Výpočetní výkon kompaktního systému je obvykle sice poměrně vysoký, ale nehodí se pro použití na velké aplikace, které by si vyžádaly nepřiměřené počty rozšiřujících modulů. [1, 2]

## 2.3 Programování

Program může být psán normalizovanými jazyky, ale může se jednat i o jejich různé deriváty používané konkrétními výrobci. Jazyky je již možné často i vzájemně kombinovat a každou část PLC programu psát jiným způsobem. Je tedy například možné textový jazyk prokládat grafickými funkčními bloky. To umožní programování maximálně přizpůsobit a zefektivnit. Norma ČSN EN 61131-3 vycházející z evropské normy EN 61131-3:2013 definuje následující normalizované programovací jazyky pro použití v PLC.

Textové jazyky reprezentuje jazyk *Instruction list (IL)*, jehož každý řádek kódu odpovídá konkrétní instrukci a je obdobou assembleru. *Structured text (ST)* je textový jazyk řazen do vyšších programovacích jazyků. Podobá se například jazyku Pascal.

Grafickým jazykem je *Ladder diagram (LD)*, který se podobá kontaktním (reléovým) schémátům a je velmi dobře čitelný elektrikáři. Dalším jazykem je *Function block diagram (FBD)*, jehož programování probíhá spojováním logických a funkčních bloků. Třetí grafický jazyk *Sequential function chart (SFC)* může být ostatním jazykům

nadřazen, protože umožňuje formulaci instrukcí v kterémkoli ze zbývajících jazyků. Reprezentují jej sekvenčně navazující bloky, které jsou odděleny vodorovnou příčkou. [3, 4, 5]

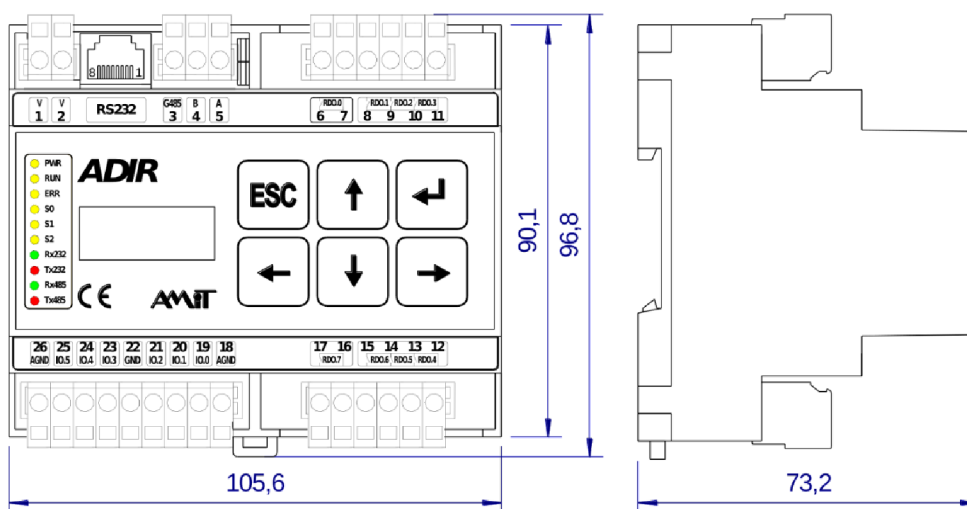
### 3 ŘÍDICÍ SYSTÉMY SPOLEČNOSTI AMIT

“Volně programovatelné řídicí systémy firmy AMiT jsou vhodné pro všechny aplikace tzv. malé a střední automatizace zejména v oblasti automatizace budov, řízení technologických celků, energetice nebo vytápění měst a obcí. Díky svým mnohdy unikátním vlastnostem představují tyto řídicí systémy ideální volbu z hlediska poměru cena/výkon.” [6]

Programování řídicích systému probíhá ve volně dostupném vývojovém software DetStudio. I ostatní software pro ladění a servisní použití firma nabízí volně na svých stránkách. Taktéž volně nabízí na svých stránkách typová řešení a aplikační poznámky, které obsahují důležité tipy pro řešení konkrétních úloh a taky obsahují funkční kódy pro jejich řídicí systémy, které může uživatel volně upravit svému požadavku. Firma poskytuje na veškerou svou produkci záruku 36 měsíců, rozšířitelnou na 5 let. [7]

#### 3.1 ADiR

ADiR je nejmenší programovatelný automat v nabídce firmy AMiT. I přes jeho velmi malé rozměry se jedná o plnohodnotné PLC, které ovšem má specificky řešené vstupy a výstupy. Vstupy a výstupy, kterých je zde 6, jsou univerzální a mohou fungovat v režimu digitálních bezpotenciálních vstupů, digitálních výstupů s napětím 15 VDC a analogových vstupů fungujících ve třech režimech. Režim první je pro připojení teplotního čidla Ni1000 nebo Pt1000, další režim je napěťový v rozsahu 0-5 V a třetí režim je standardní proudový v rozsahu 0-20 mA. Napěťový a proudový režim má nicméně omezenou přesnost měření a to 5 %. Dále automat obsahuje 8 digitálních reléových výstupů.

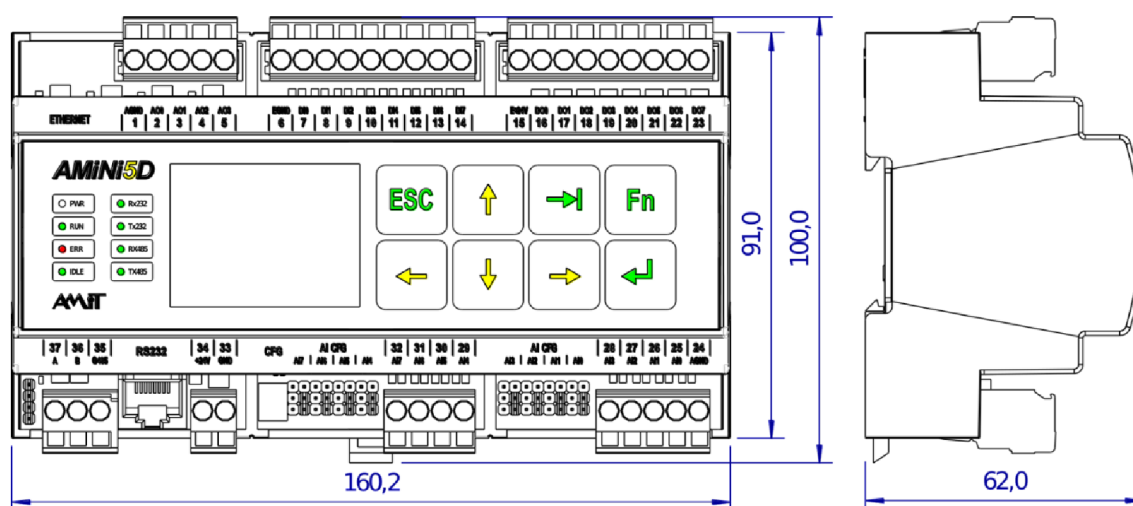


Obrázek 4: Rozměry produktu ADiR, převzato z [9]

Stanice je vybavena displejem o velikosti dvou řádků po osmi znacích, pro ovládání slouží 6 tlačítek. Základní činnosti a stavy jsou zobrazeny pomocí LED indikátorů. Disponuje pamětí RAM o velikosti 1 MB, kterou běžně disponují i mnohem výkonnější PLC. Pro programování a připojení GSM modemu zde slouží linka RS232, pro připojení dalších stanic (až 32) je jednotka vybavena rozhraním RS485. Rozměrově je velmi kompaktní a upevnění se provádí na 35 mm DIN lištu. [8, 9]

### 3.2 AMiNi5D

Tento řídicí systém je nejuniverzálnějším z nabídky společnosti. Zařízení má již vstupy standardně rozdělené na analogové (8) a digitální (8) přímo na oddělených svorkách. Digitální vstupy jsou standardních parametrů, tj. 24 VDC a čtyři z nich mají navíc možnost použití jako čítače do frekvence 200kHz. Analogové vstupy mají plný rozsah napětí 0-10V a proudu 0-20mA s možným přímým zapojením čidel Ni1000/Pt1000. Specialitou je možnost přímého zapojení teplotních čidel NTC. Model disponuje dále skupinou 4 analogových výstupů (0-10 V) s 12bitovým rozlišením a 8 běžných digitálních výstupů.



Obrázek 5: Rozměry produktu AMiNi5D, převzato z [11]

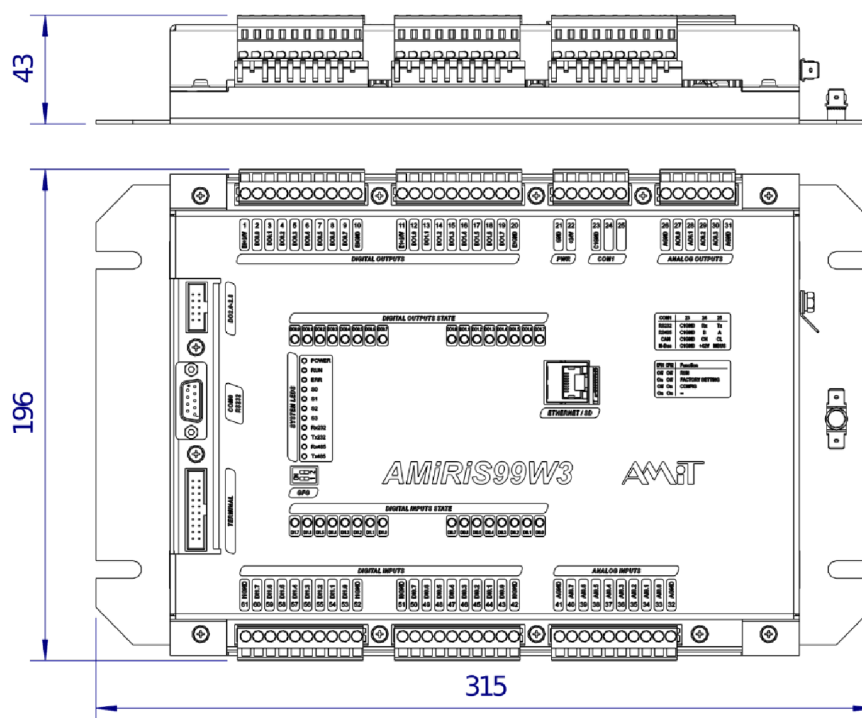
Pro obsluhu je zde 2" barevný IPS displej s rozlišením 320x240 pixelů a 8 kláves. Vedle displeje je 8 LED indikátorů stavů. Systém je vybaven také volně parametrizovatelným webovým serverem. Z toho nepřímo vyplývá možnost připojení do sítě a programování přes rozhraní Ethernet. Jako další komunikační rozhraní je k dispozici RS232 nebo RS485 pro komunikaci s dalšími jednotkami. Tudíž je možné rozšířit zařízení o vzdálené vstupy/výstupy pomocí speciálních modulů. Systém se rovněž montuje na 35mm DIN lištu. Provozní teploty jsou v rozmezí -20 až 70°C. [10, 11]

### 3.3 AMiRiS99W3

Tělo systému je narozdíl od dříve zmíněných kompaktních systémů tvořeno velmi robustním plechovým provedením o značné síle. Svědčí o tom i jeho váha, která čítá 2,2 kg. Montáž je řešena již přímo na základní desku rozvaděče. Nabízí již značný počet vstupů/výstupů. A to 16 digitálních výstupů a stejný počet digitálních vstupů. Všechny jsou samozřejmě galvanicky oddělené.

Dále je pak k dispozici 8 analogových vstupů s volitelným napěťovým rozsahem 0-5 V nebo 0-10 V, klasickým proudovým rozsahem 0-20 mA a nechybí již známá podpora pro čidla Ni1000/Pt1000.

Tento produkt nabízí možnost použít rozšiřující interní moduly, které mohou nabídnout až 4 analogové výstupy. Takový modul má dvě konfigurace, a to napětíovou nebo proudovou. Jeho převodní rozlišení je 10 bitů.

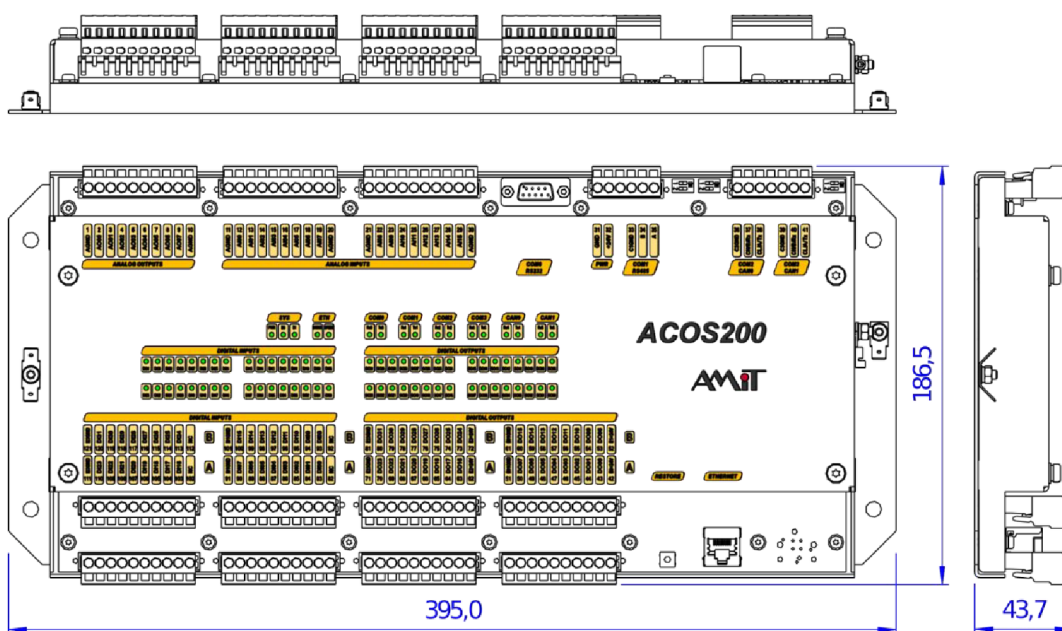


Obrázek 6: Rozměry produktu AMiRiS99W3, převzato z [13]

AMiRiS99W3 komunikuje po síti pomocí Ethernetu 10/100 Mbps, RS232 a umožňuje ve volitelném přídatném slotu přidat podporu další RS232, RS485, M-BUS nebo CAN sběrnice. Kromě toho má integrovaný webový server, kde může být umístěna vizualizace řízeného procesu. Slot pro SD kartu lze použít v případě potřeby úložiště pro dlouhodobé ukládání hodnot. Pro zjištění aktuálních stavů digitálních vstupů/výstupů je na čelní desce umístěna řada LED. Teplotní pracovní rozsah je -40 až 70°C. [12, 13]

### 3.4 ACOS200

Největší kompaktní řídicí systém společnosti AMiT, přesně to je ACOS200. Jeho konstrukce vychází z AMiRiS99. Je taktéž v robustním silnostěnném provedení a je ještě delší. To výrobci umožnilo na zařízení umístit celkově 88 vstupů/výstupů. Přesněji jde o konfiguraci 32 digitálních vstupů s 32 digitálními výstupy a 16 analogových vstupů organizovaných po osmi. Ty jsou schopny fungovat v již známých běžných rozsazích 0-10 V, 0-20 mA a samozřejmě mají podporu pro Ni1000/Pt1000 čidla. Systém také má 8 analogových výstupů se společným záporným pólem. Analogové vstupy/výstupy zde disponují 12bitovými převodníky.



Obrázek 7: Rozměry produktu AMiRiS99W3, převzato z [15]

Jsou zde dvě volitelná rozhraní pro komunikační moduly, z těch můžeme vybrat podporu další RS232, RS485 nebo přidat CAN. Systém již totiž disponuje Ethernetem, RS232 a RS485. Jako jeho menší varianta umožňuje ovládání a vizualizaci pomocí integrovaného webového serveru, nebo ukládání logovaných dat na SD kartu. Taktéž má indikační LED na čelní straně krytu. [14, 15]



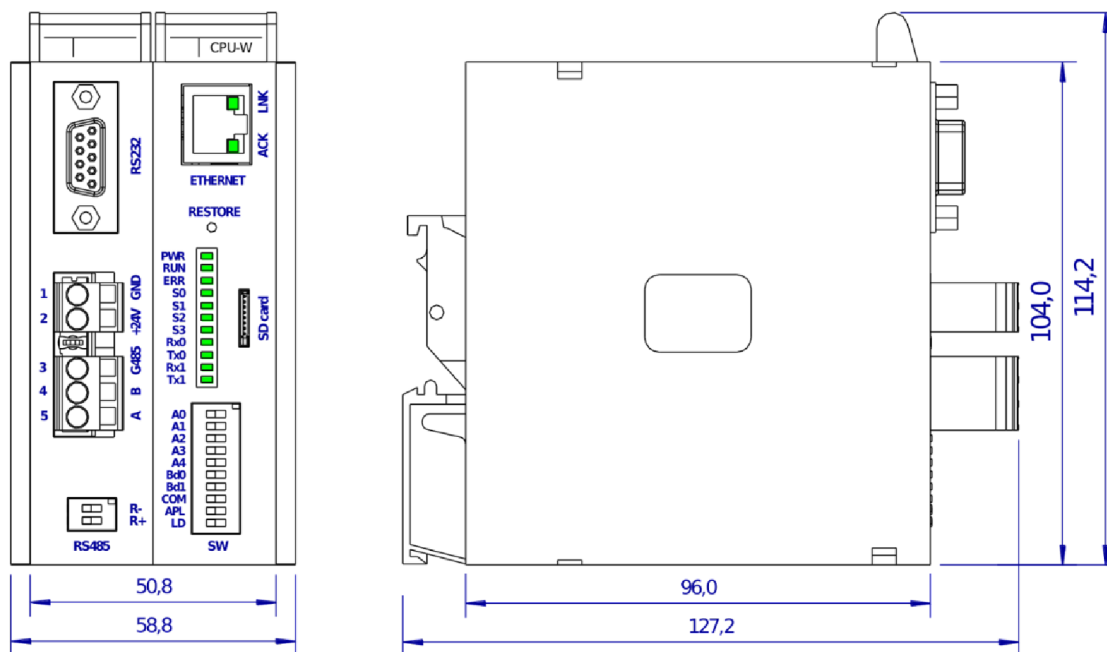
### 3.5 ADiS

System zvaný ADiS je jediný zástupce modulárních řídicích systémů firmy AMiT. Historicky ho tvořily dva možné hlavní CPU moduly. Dnes již pouze jeden, který je nástupcem staršího typu. Tento typ je označen jako AD-CPUW2 a mimo jiné na rozdíl od předchůdce již má rozhraní Ethernet. Tady již disponuje vlastním webovým serverem.

Základní vlastností modulárního systému ADiS je možnost poskládat dohromady až 16 vstupně-výstupních rozšiřujících modulů využívajících pouze interní komunikační sběrnici. To znamená dostupných až 256 digitálních vstupů/výstupů. System umožňuje s využitím modulu ADC-CAN a expanzními moduly systému DIOCAN připojit celkově až 1024 digitálních vstupů/výstupů. Komunikace CPU jednotky s dalšími až 32 řídicími systémy umožňuje rozhraní RS485. Dříve zmiňovaný Ethernet samozřejmě prakticky omezení počtu nemá. Zařízení nabízí i rozhraní RS232, které je určeno pro přímé propojení dvou zařízení, například propojení s PC.

Procesorové jádro je řešeno formou DualCPU. To znamená, že systém obsahuje dvě procesorové jednotky. Jedna je procesní a druhá je vyhrazena pro komunikaci. Navzájem jsou schopny komunikovat po SPI sběrnici.

Provedení tohoto systému jsou plastové modulové krabičky jejichž montáž je výhradně na 35mm DIN lištu v rozvaděči umožňující skládat moduly vedle sebe pro jejich vzájemné propojení sběrnice. [16, 17, 18]



Obrázek 8: Rozměry produktu AD-CPUW2 z řady ADiS, převzato z [17]

Rozšiřující moduly představují například CPU moduly, vstupně výstupní moduly a komunikační moduly.

Nabízeným CPU modulem je kromě výše zmiňovaného hlavního procesorového modulu AD-CPUW2 i jeho starší typ AD-CPU167. Do této skupiny se řadí také modul ADC-CAN obsahující procesor výhradně pro rozšíření počtu vstupů/výstupů využitím rozhraní CAN, ke kterému lze připojit expanzní moduly využívající tuto sběrnici.

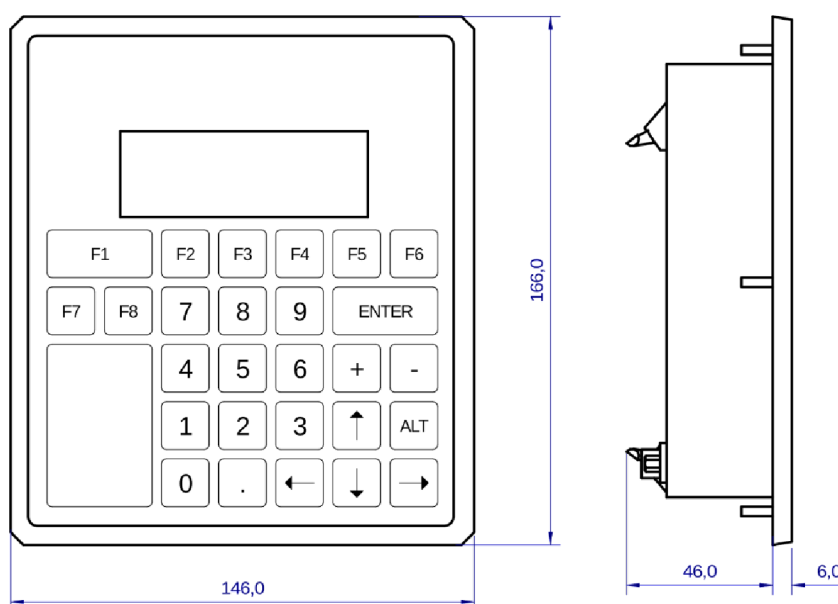
Vstupně výstupní moduly mohou být v různých konfiguracích počtů vstupů/výstupů, jak napovídá jejich název. V běžném analogovém či digitálním provedení ve standardu 24 VDC. Například modul AD-GAI8 obsahuje 8 analogových vstupů využívající 15bitové převodníky a pracuje i se zápornou hodnotou napětí a proudu. Rozsahy analogových signálů jsou  $\pm 10$  V,  $\pm 20$  mA. Další zajímavostí jsou speciální moduly s tenzometrickými nebo IRC vstupy.

Komunikační moduly rozšiřují, případně přidávají podporu pro další komunikační rozhraní, například RS232, RS485 / RS422, Ethernet. Rozhraní CAN podporuje modul AD-CAN, který sám o sobě neumožňuje rozšíření počtu vstupů/výstupů. [16,17,18]



### 3.6 APT1000G

APT1000G je jednoduchý textový operátorský panel. Pro obsluhu obsahuje 27 membránových tlačítek. K zobrazování využívá podsvícený LCD displej o rozsahu 4x20 znaků s výškou 4 mm. Montáž probíhá do panelu, kdy z přední strany zařízení následně poskytne krytí IP55. Napájení je 24 VDC při odběru do 0,2 A. Pro komunikaci využívá rozhraní RS232 nebo RS485/RS422. Je možné připojit jej k jakémukoliv řídicímu systému z nabídky výrobce. [19]

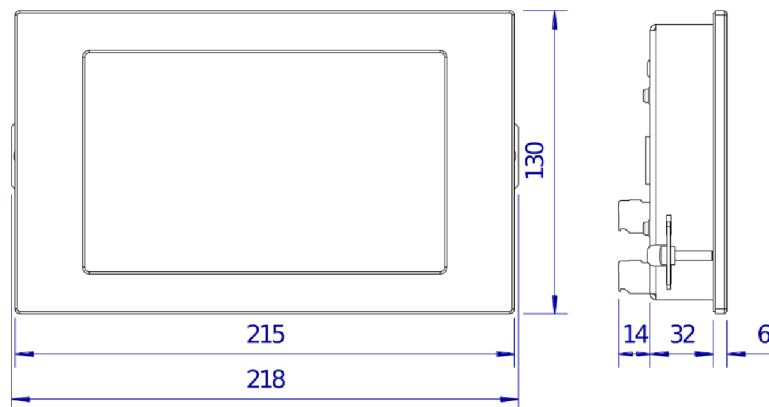


Obrázek 9: Rozměry produktu APT1000G, převzato z [19]

### 3.7 AMR-OP87/V

Technicky vzato jde o volně programovatelný plnohodnotný řídicí systém, který je vybaven barevným 7" TFT LCD displejem s rozlišením 800x480 bodů. Displej je vybaven odporovou dotykovou vrstvou pro ovládání. Takovéto specifikace z něj tvoří zástupce HMI panelů. Sice nemá žádné vlastní vstupy a výstupy, ale disponuje rozhraním RS485 a mohou být přes něj připojeny moduly vzdálených vstupů a výstupů řady AMRIO využívající protokolu ARION a MODBUS RTU. Tyto moduly spolu s dostatečně výkonným procesním CPU jednotky a 4 MB RAM mohou tvořit solidní řídicí systém. Montážně je určen pro vsazení do dveří rozvaděčů. Čelní panel poskytuje krytí IP65 a rozsah pracovních teplot zařízení je -20 až 70°C.

Nechybí ani rozhraní Ethernet pro připojení do sítě a integrovaný webový server. Uvnitř systém ukrývá jeden slot pro volitelné přídatné moduly řady EM-xx, což jsou komunikační moduly pro rozhraní RS232, RS485 či CAN. Také se v něm nachází slot pro volitelný modul EMW-GSM, umožňující bezdrátovou komunikaci v GSM síti. Grafické obrazovky se programují stejně jako řídicí systémy v software DetStudio nebo EsiDet. [20]



Obrázek 10: Rozměry produktu OP87/V, převzato z [20]

## 4 ŘÍDICÍ SYSTÉMY UNIPI TECHNOLOGY

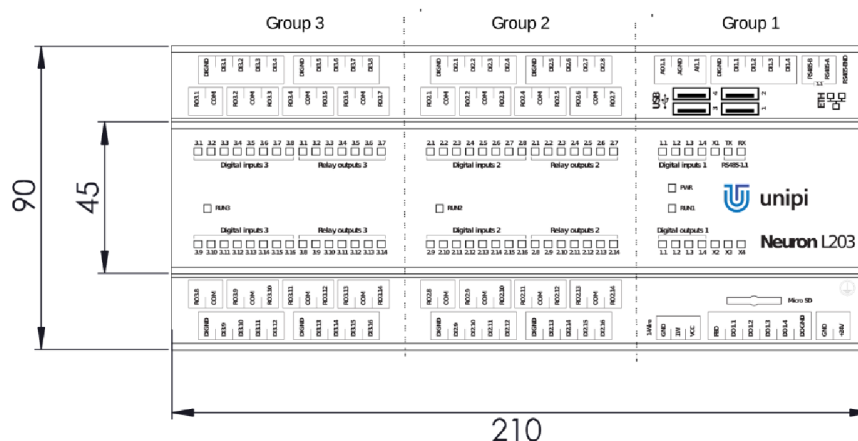
"Programovatelné logické kontroléry (PLC) a převodníky Unipi jsou navrženy pro automatizaci, ovládání, monitoring či měření a regulaci (MaR). Vysoký výkon, softwarová otevřenost a možnosti zakázkových úprav umožňují širokou škálu nasazení od technických zařízení budov a průmyslu, přes řízení systémů ventilace, vytápění, klimatizace a SCADA až po chytré domácnosti a další automatizační projekty." [21]

Společnosti Unipi staví své řídicí systémy softwarově otevřené jako Open-source. Kromě oficiálně podporované platformy Mervis tak lze používat libovolný software třetích stran, či vlastní software. Obecně jsou produkty Unipi stavěny na operačním systému Linux a k produktům je k dispozici API pro jednodušší aplikaci vlastních softwarových řešení. [21]

### 4.1 Unipi Neuron

Neuron je řada modulárního řídicího systému (PLC) vhodná pro použití v průmyslu i domácí automatizaci. Využívá se pro monitorování, řízení a regulaci systémů budov a HVAC systémů. Hardware je postaven na využití jednodeskového počítače RaspberryPi, aktuálně ve verzi 3 a 4.

Takový hardware dodá PLC vysoký výkon. V případě výkonnějšího modelu obsahuje čtyřjádrový procesor o taktu 1,5 GHz a až 8 GB RAM. RaspberryPi přináší i podporu Bluetooth, Wifi a 1Gbit Ethernet portu. Pro vizualizaci a ovládání má zařízení vestavěno vlastní webový server.



Obrázek 11: Rozměry produktu Neuron L, převzato z [23] a upraveno

PLC má modulární kompaktní architekturu stavící na sekčním rozdělení. Každou sekci řídí samostatný procesor komunikující s hlavním CPU. Zařízení je nabízeno ve třech velikostech. Nejmenší velikost "S" zastupuje například Neuron S103, tvoří jej pouze jedna sekce. Naopak největší velikostí "L" je například Neuron L533, ten má již 3 sekce.

Každá sekce obsazuje 4 DIN moduly na 35 mm DIN liště, na kterou se tyto systémy montují. Každá velikost (S, M, L) je k dostání v různých hardwarových konfiguracích. V závislosti na zvolené konfiguraci lze dosáhnout maximálně těchto parametrů (bez případného rozšiřování):

- 36 digitálních vstupů
- 4 digitálních výstupů
- 56 relé výstupů
- 9 analogových vstupů
- 9 analogových výstupů

Parametry analogových vstupů a výstupů jsou v běžném rozsahu 0-10 V a 0-20 mA s přímou podporou čidel Ni1000 a Pt1000. Napětí digitálních vstupů/výstupů je standardních 24 VDC, totéž platí pro napětí napájecí.

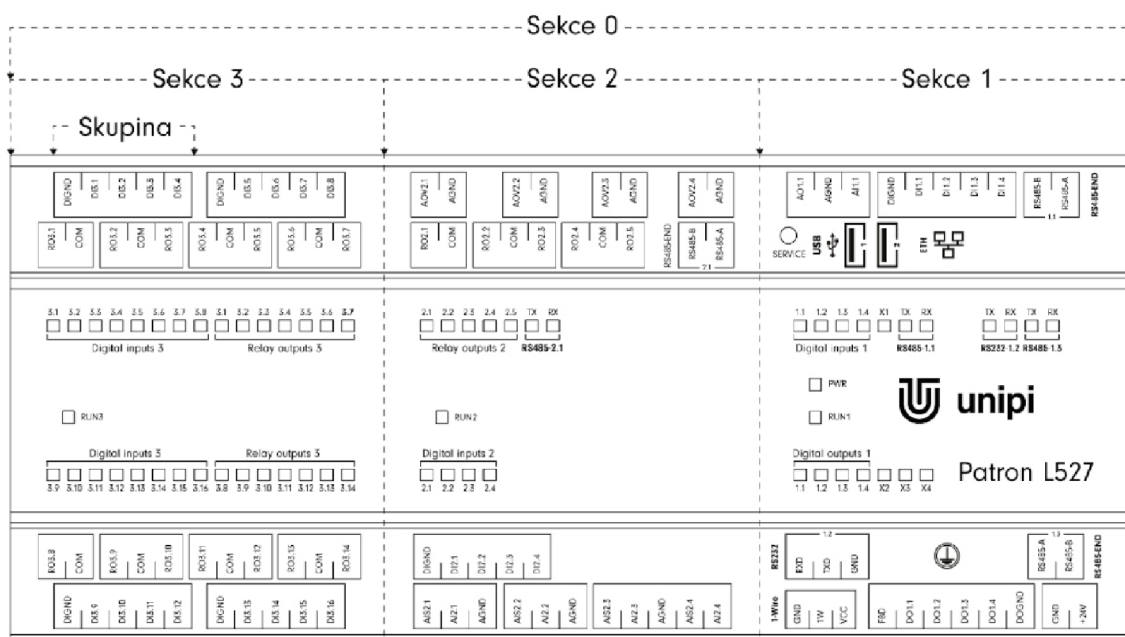
V základní výbavě disponuje každý Neuron třemi linkami RS485 sloužícími komunikaci prostřednictvím protokolu Modbus RTU nebo pro připojení externích zařízení jako například HMI panelů, měřičů energie nebo rozšiřujících modulů. Přídavné moduly tak umožní ještě navýšit počet vstupů/výstupů. Na jedné sběrnici může komunikovat až 32 zařízení.

Zajímavostí tohoto produktu je možnost využití 1-Wire sběrnice, na níž může probíhat čtení dat až z 15 senzorů podporujících tento standard. Méně častou výbavou PLC jsou 4 USB porty umožňující připojení externích USB převodníků či paměťových zařízení. [22, 23, 24]

## 4.2 Unipi Patron

Patron je další kompaktní produktovou řadou PLC výrobce Unipi. Je navržen pro regulaci, měření, ovládání, automatizaci technických zařízení budov a pro nasazení jako SCADA. Výrobce na něj poskytuje prodlouženou záruku 4 roky. Slovy výrobce se jedná o nejvýkonnější kompaktní PLC na trhu. To má dokládat použití vlastního výpočetního modulu osazeného procesorem i.MX 8M Mini s taktem 1,8 GHz na čtyřech fyzických jádrech. Procesor disponuje 1 GB RAM a 8 GB eMMC pamětí a 100 Mbit Ethernetem.

Další konektivitu zde zajišťuje RS485, a to dokonce v maximálním počtu 4 nezávislých linek. Dále pak RS232, 1-Wire sběrnice a dva USB 2.0 porty. USB porty lze využít pro adaptéry, externí úložiště a podobně. Některé modely jsou vybaveny LTE modemem pro datovou komunikaci přes mobilní síť.



Obrázek 12: Náhled produktu Patron L527, převzato z [26]

Řada patron je navržena pro maximální možné individuální zakázkové přizpůsobení. Opět jako řada Neuron využívá sekční konstrukce o třech velikostech (S, M, L) a každou sekci řídí samostatný procesor komunikující s hlavní CPU. Je i rozměrově v podstatně totožná, jedna sekce zabírá opět 4 DIN pozice na montážní 35 mm DIN liště.

Jednotka může disponovat velmi různými počty analogových a digitálních vstupů a výstupů, reléových výstupů i komunikačních rozhraní. Počty i uspořádání se liší v každé konfiguraci i v závislosti na velikosti zvoleného modulu. Dále je možné i jejich rozšíření použitím rozšiřujících modulů Unipi Extension.

Kdyby nebyly dostačující nabízené konfigurace jednotek, je možnost jednotky zakázkově přizpůsobit. Příkladem zakázkových úprav může být napájení přes PoE, SecureBoot, podpora PCIe, vlastní konfigurace vstupů/výstupů nebo přidání podpory bezdrátových komunikačních protokolů jako například ZigBee. Výrobce nabízí přizpůsobení vnějšího designu, vlastního loga, nebo také zakázkově upravený obraz operačního systému. [25, 26, 27]



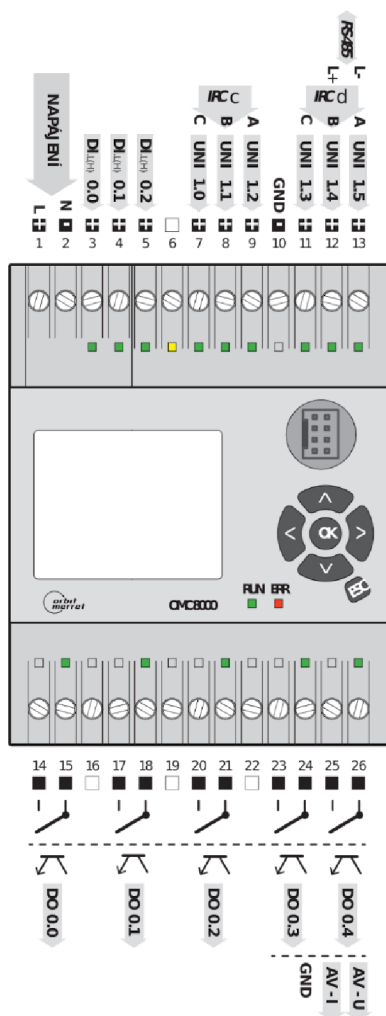
## 5 ŘÍDICÍ SYSTÉMY ORBIT MERRET

Orbit Merret nabízí pouze jedno řešení PLC, avšak je natolik univerzální, že pokrývá velmi široké spektrum aplikací. Společnost poskytuje na své výrobky rozšířenou záruku trvající 5 let. Cyklicky testuje 48 hodin zahořování na každém přístroji. Měřicí protokoly přikládá ke každému výrobku. Specialitou je aplikace jejich řídicích systémů v energetice a zejména jaderných elektrárnách. [28]

### 5.1 OMC 8000

Konstrukce PLC OMC 8000 je modulová a zároveň se jedná o kompaktní systém umístěný na 35mm DIN lištu s vlastnostmi velkých řídicích systémů. K hlavní jednotce lze připojit až 31 rozšiřujících modulů, se kterými jednotka komunikuje po sběrnici CAN. Pro získání ještě většího rozsahu, nebo výpočetního výkonu, lze řídicí program rozdělit mezi více hlavních jednotek OMC 8000 a propojit je rozhraním Ethernet. Navíc takovéto propojení může být prakticky libovolně vzdálené. Může být napájen 24 V nebo 230 V a přímo na jednotce jsou 3 digitální vstupy. Kromě zmíněných digitálních se zde vyskytuje 6 naprosto univerzálních vstupů. Tyto vstupy umí zpracovat konkrétně následující signály: impulzní signál do 30 V, analogový napěťový do 30 V, analogový proudový do 20 mA, analogový odporový do 3900  $\Omega$ , analogový signál z Pt100, Pt1000 a Ni1000, analogový z KTY81-2xx, IRC do 0,5 Mhz, frekvenční vstup do 0,5 kHz a dva vstupy umí tvořit linku RS485. Navíc jsou i galvanicky oddělené.

Výstupy má řešené v počtu pěti reléových, OC nebo i v kombinaci (3+1) s jedním analogovým výstupem s 16 bit převodníkem. Výstupy a napájecí rozsah jsou konfigurovatelné při nákupu.



Obrázek 13: Náhled produktu OMC 8000, převzato z [29]

Pro ovládání a zobrazování je na čelní straně vybaven 6 tlačítky a barevným TFT displejem o rozlišení 160x128 bodů. Pro vzdálenou správu PLC nabízí možnost připojení pomocí VNC, kde se zobrazí displej s jeho hardwarovým rozlišením. Zařízení obsahuje také FTP server pro připojení k interní flash paměti nebo i k vložené SD kartě. Interní paměť je typu NAND o velikosti 512 MB, SD karta může mít kapacitu až 32 GB. Další vizualizaci a nastavení parametrů lze provést přes webový server.

Rozšiřujících modulů je v nabídce výrobce celkem 20 typů. Každý z nich má jinou konfiguraci vstupů/výstupů, případně nabízí komunikační rozšíření. Modulem s nejmenším počtem vstupů/výstupů může být například OMC 81114DU, který je specializovaný na 4 analogové vstupy lineárních potenciometrů. Naopak největším je i OMC 800136DI, který disponuje 36 digitálními vstupy. [28, 29, 30]



## 6 ŘÍDICÍ SYSTÉMY SPOLEČNOSTI TECO

Společnost vyvíjí dva typy PLC systémů, jeden je kompaktní Foxtrot a druhý modulární TC800, avšak stále velmi kompaktních rozměrů. Využití těchto systémů je zejména v energetice, automatizaci inteligentních budov, řízení lodí/jachet, vodohospodářství a existují i aplikace pro řízení dopravy, například řízení proměnlivého značení dopravních tunelů. Společnost poskytuje nepřetržitou technickou podporu a garantuje plnou podporu pro zařízení (generaci) po ukončené výrobě 10 let. [32, 33]

“Orientaci na mezinárodní technické standardy podtrhuje Teco a.s. svým členstvím v organizaci PLCopen, která má za cíl rozvíjet a prosazovat do života normu IEC 61131-3. Teco a.s. má pro vývoj PLC a dalších automatizačních systémů k dispozici vlastní výrobní a vývojové kapacity a zajišťuje průběžné zvyšování kvalifikace všech svých zaměstnanců, techniků a vývojových inženýrů.” [31]

### 6.1 Tecomat Foxtrot 2

Tecomat Foxtrot 2 představuje nástupce předchozí generace Foxtrot a reprezentuje kompaktní třídu PLC, umísťuje se již standardně na montážní 35 mm DIN lištu. Foxtrot 2 aktuálně představuje pouze základní moduly, tyto moduly jsou však kompatibilní s předchozí generací. Tato generace čítá 9 různých konfigurací základního modulu. Základní moduly jsou kompaktní PLC avšak v různých velikostech v závislosti na konfiguraci. Společný mají všechny konfigurace procesor ARMv7 792 MHz, dále 2x Ethernet, TCL2, min. jednu CIB sběrnici, 2 USB porty (1 host + 1 device), OLED displej o různé velikosti a tlačítka pro obsluhu. Dále pak velikost paměti RAM 16 MB, paměť pro uživatelský program o velikosti 1 MB, programování jazyky ST, LD, FBD, SFC, CFC. Obsahuje integrovaný datalogger a integrovaný webový server, přes web API je možný přístup k programovým proměnným. Programování probíhá ve vývojovém prostředí Mosaic. Doba jednoho cyklu je při 1000 logických operacích 0,036 ms. TCL2 je v konfiguraci RS485 a sběrnice CIB umožňuje připojení až 32 periférií typu CFox. Tyto periferie mohou být speciální čidla, servo hlavice ale zejména potom moduly rozšiřující počty vstupů a výstupů v různých počtech a konfiguracích. Řada Foxtrot obsahuje i speciální rozšiřující periferie využívající rozhraní TCL2 a to například polohovací modul GT-1753 určený pro řízení 4 os.



Obrázek 14: Náhled produktu Foxtrot 2 CP-2007, převzato z [34]

Konkrétně konfigurace CP-2007 disponuje 14 volitelnými vstupy, tedy digitálními, bezpotencionálními nebo analogovými. V analogovém režimu umí samozřejmě napětové, proudové nebo odporové snímání v rozlišení 12 bitů a čtyři z těchto vstupů je možné použít i jako čítače. Dále na modulu nalezneme 1 digitální vstup pro 230 VAC, 11 reléových výstupů (230 V / 3 A), 2 volitelné analogové napětové výstupy v rozsahu 0-10 V s rozlišením 12 bit použitelné také jako PWM, také je možné získat další 2 analogové napětové výstupy místo 2 vstupů a to pomocí propojky. Modul obsahuje volné sloty pro vložení rozšiřujících karet, kterými lze získat rozhraní WLAN a LTE. Jiné varianty neobsahující možnost rozšíření WLAN disponují naopak možností SD karty pro získání většího úložiště. [32, 34, 35, 36]

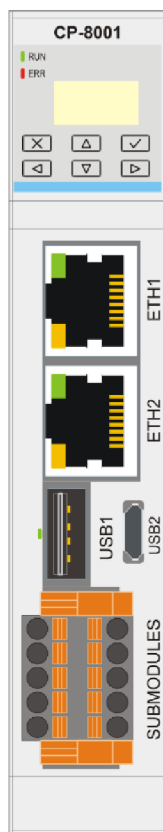
## 6.2 Tecomat TC800

Produkt Tecomat TC800 je další generací modulárních řídicích systému společnosti Teco. Konkrétně se jednotlivé moduly zavěšují na výrobcem speciálně upravenou (vyvinutou) 35 mm DIN lištu, která v sobě ukrývá navíc systémovou sběrnici, po které moduly komunikují, a přes kterou jsou napájeny. Tuto lištu výrobce nabízí v různých délkách podle potřeby počtu modulů. Základní CPU modul je reprezentován jedinou konfigurací a to CP-8001. Obsahuje totožný procesor ARMv7 792 MHz jako kompaktní verze Foxtrot 2. Mimo to mají stejné i jiné hardwarové parametry. Velikost paměti RAM 16 MB, paměti pro program 1 MB, 2 Ethernet porty, 2 USB porty (1 host + 1 device), OLED displej a 6 tlačítek. Navíc má potom 2 sloty pro submoduly (MR-013x) a interní sběrnici TCL3, pomocí které lze připojit až 64 vstupně-výstupních modulů. Tuto sběrnici obsahuje dříve zmíněná modifikovaná "DIN" lišta s označením RM-894x.



Obrázek 15: Náhled produktu TC800 IT-8601, převzato z [38]

Další komunikační rozhraní je možné získat připojením submodulů, které jsou určeny právě k tomuto účelu. Submoduly mohou rozšířit možnost komunikace o RS232, RS485 a CAN. Výrobce nenabízí komunikační moduly na interní TCL3 sběrnici. Systém podobně jako předchozí Foxtrot má integrovaný webový server a datalogger. Taktéž disponuje možností přístupu k proměnným přes web API. Jak Foxtrot, tak TC800 umožňují online úpravu uživatelského programu bez přerušení řízení. K hlavní CPU jednotce se připojuje napájecí modul (PW-8901), který umožňuje oddělit napájení vstupů a výstupů od interního napájení PLC sestavy. Potom už výrobce nabízí pouze vstupně-výstupní moduly v různých konfiguracích. [32, 37]



Obrázek 16: Náhled produktu TC800 CP-8001, převzato z [37]

Jedním z těchto modulů je IT-8601 rozšiřující systém o 8 analogových vstupů. Pro propojení s hlavním modulem využívá sběrnice TCL3. Rozlišení AD převodníku je 24 bitů. Modul má nastavitelné napěťové a proudové rozsahy s příslušným rozlišením viz. Tabulka 1. [32, 38]

*Tabulka 1: Napěťové a proudové rozsahy AD převodu modulu IT-8601 [38]*

Napěťové rozsahy a rozlišovací schopnost		Proudové rozsahy a rozlišovací schopnost	
0 až 10 V	0,84 $\mu$ V	0 až 20 mA	0,002 $\mu$ A
0 až 5 V	0,42 $\mu$ V	4 až 20 mA	0,002 $\mu$ A
0 až 2 V	0,21 $\mu$ V		
0 až 1 V	0,08 $\mu$ V		
0 až 0,5 V	0,04 $\mu$ V		

## 7 APLIKACE ŘÍDICÍCH SYSTÉMŮ SPOLEČNOSTI AMIT

### 7.1 Použité moduly

Pro aplikaci byly použity řídicí systémy společnosti AMiT. Jednak byl využit modul AMiRiS99, který má ukončenou výrobu, tudíž mu chybí některé aktuální technologie a funkce. Tyto nedostatky jsou kompenzovány použitím operačního panelu (AMR-OP83) současné generace, který chybějící funkce doplňuje a zároveň umožní vizualizaci a ovládání dotykovým displejem. Oba moduly lze propojit rozhraním RS485 a tím umožnit jejich vzájemnou komunikaci. Každý z modulů je programován v jiném editoru v rámci jednoho softwaru.

#### 7.1.1 AMiRiS99

AMiRiS99 je zástupcem řídicích systému starší tzv. 1 generace výrobce AMiT. Využívá starší CPU a starší editor PseDet v rámci software DetStudio. Jedná se o modul, který narozdíl od současné verze (AMiRiS99W3) nedisponuje mimo jiné komunikačním rozhraním Ethernet a ani webovým serverem. Pro komunikaci využívá RS232 nebo RS485. Další parametry má prakticky shodné s modulem AMiRiS99W3, který je detailněji popsán v teoretické části práce.

#### 7.1.2 AMR-OP83

Řídicí systém AMR-OP83 zde plní zejména funkci HMI panelu. HMI panel je postaven na aktuální generaci 2E+, což znamená, že je programován v editoru EsiDet. Pro zobrazování je vybaven 3,2" TFT displejem. Prakticky jde pouze o zmenšenou verzi panelu AMR-OP87/V, který je taktéž popsán detailněji v teoretické části práce.

### 7.2 Použitý software DetStudio

DetStudio je vývojové prostředí společnosti AMiT, které slouží k vytváření uživatelských aplikací pro řídicí systémy a programovatelné regulátory vlastní výroby. Umožňuje tvořit vlastní aplikaci, navrhovat obrazovky, on-line ladit běžící aplikaci i následně vytvářet dokumentaci. Programování může probíhat v jazycích ST/LD/IL/FB, nebo i kombinací ST+LD, podle podpory konkrétního řídicího systému. Samotný software je zdarma a open source. Výrobce poskytuje k tomuto prostředí také zdarma školení. [7]

## 7.3 Výběr aplikace a rozvaha

Celý záměr tvorby programu pro řídicí systém by měl začít rozvahou nad problematikou konkrétní aplikace. Tedy rozvahou nad tím, co budeme řídit a jaký by měl být výsledek. Pro tento konkrétní účel, tedy prezentaci postupu tvorby programu v ekosystému společnosti AMiT je vybrán velmi jednoduchý problém.

Úkol spočívá v řízení světelného signalizačního zařízení (SSZ) s radarovou kontrolou rychlosti projíždějících vozidel. Jednoduše řečeno semafor s radarem a přechodem pro chodce, který se vyskytuje často na okrajích obcí. Aplikace je dostatečně jednoduchá pro celkové pochopení a zároveň na ní lze postup tvorby řízení názorně popsat. Navíc pro lepší demonstraci má být řízení vizualizováno a obsahovat i simulované řízení.

### 7.3.1 Popis vybrané aplikace

Aplikaci je dobré velmi dobře popsat. Popsat tedy veškerou vyžadovanou funkcionalitu a chování, kterého chceme řízením dosáhnout. Řízení semaforu a jeho funkcionalita jsou sice zřejmé, přesto je dobré si chování semaforu sepsat. SSZ je umístěno na pozemní komunikaci a signalizuje jak pro vozidla, tak pro chodce na chodníku. Je vybaveno tlačítkem pro chodce, které zastaví dopravu a umožní bezpečné přejetí přechodu. Pro kontrolu rychlosti je vybaveno radary, které při překročení povolené rychlosti daný směr zastaví.

Po technické stránce tedy systém řídí tři logické celky světelné signalizace, dva z toho pro dopravu a jeden pro chodce (semafory na obou stranách přechodu jsou totiž synchronní). Signalizace definuje požadované výstupy z řízení. Vstupy, které vstupují do řízení jsou také tři. Dva radarové signály o překročení rychlosti a jedno tlačítko pro chodce (opět se na obou stranách chová stejně).

Změny v signalizaci jsou s konkrétním časovým zpožděním, takže musí být použity i časovače s požadovanými parametry.

Vizualizace má ukazovat všechny aktuálně signalizované stavy na SSZ. Pro možnost simulace těchto stavů je třeba simulovat i pokyny k zastavení z radarů a tlačítko pro chodce.

### 7.3.2 Potřebný rozsah hardwaru a jeho volba

Z popisu aplikace by mělo být zřejmé jak velký počet vstupů či výstupů bude zapotřebí. Dále by měly vyplynout případné speciální požadavky nebo vlastnosti řídicího hardwaru. Z těchto informací lze vyvodit požadované parametry hardwaru a lze poté provést jeho výběr. V této situaci je hardware předem stanoven zadáním, každopádně pro řízení je potřeba konfigurace o minimálně 3 digitálních vstupech a 8 výstupech. Tyto požadavky vybraný hardware s velkou rezervou splňuje. Na samostatný výstup je připojeno každé jednotlivé světlo signalizace. Tomu se vymyká pouze signalizace pro chodce, kde je signalizace na obou stranách přechodu synchronní.



Vizualizace může být řešena různými způsoby, v předložené práci je konkrétně řešena na dotykovém panelu, který zároveň umožňuje i zmiňovanou simulaci vstupů.

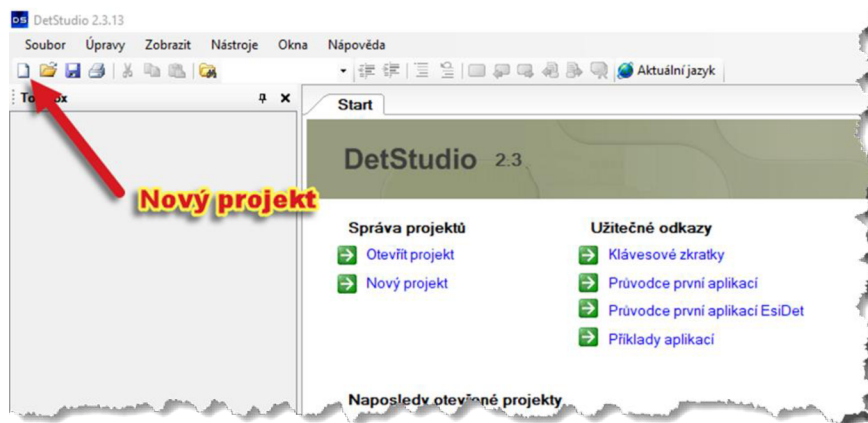
Při použití více řídicích systémů je třeba dbát i na možnosti vzájemné komunikace. AMiRiS99 použitý pro řízení disponuje 16 digitálními vstupy a 16 digitálními výstupy. Z analogových signálů umí zpracovat 8 vstupů. AMR-OP83 k vizualizaci využívá dotykový displej a pro vzájemnou komunikaci jsou oba systémy vybaveny linkou RS485.

## 7.4 Tvorba uživatelského programu pro AMiRiS99 (řídící část)

Instalace softwaru a technická příprava nejsou předmětem tohoto výkladu.

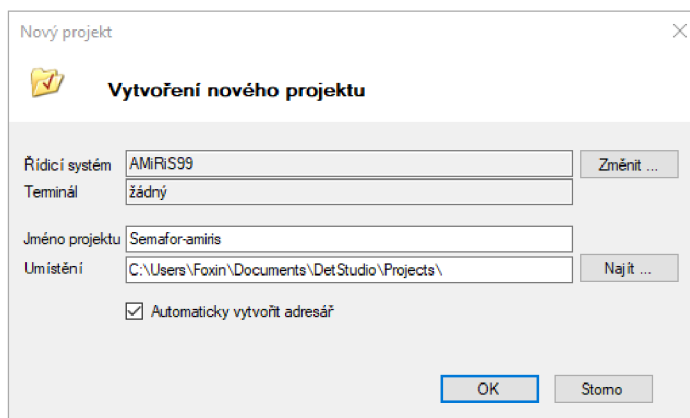
### 7.4.1 Založení projektu

Pro založení projektu slouží první tlačítko v horní liště funkcí. Při zakládání nového projektu je třeba vybrat řídicí systém, pro který chceme vytvořit řídicí program. Prostředí podle volby nakonfiguruje fyzické vstupy a výstupy odpovídající vybavení konkrétního modelu, a to včetně komunikačního rozhraní. Nejdůležitější ale je, že prostředí podle zvoleného modelu vybere vhodný editor. Pro použitý AMiRiS99 se jedná o starší editor PseDet.



Obrázek 17: Založení nového projektu

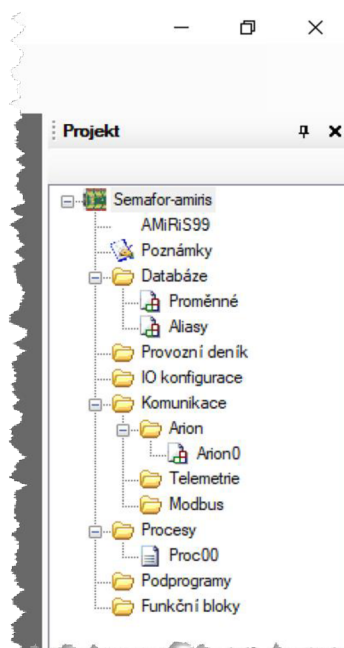
Po uvedení jména projektu, adresáře umístění a potvrzení se zobrazí okno s podrobnostmi o projektu, kde uživatel může definovat číslování verzí, autora, copyright a podobně. Vše je volitelné, není nutné nic měnit.



Obrázek 18: Dialog zakládání projektu

Po založení je v pravé části obrazovky pravděpodobně nejdůležitější část projektu, a to jeho kompletní struktura. V té se nachází vše, co se projektu týká, a lze z tohoto místa ke všemu přistupovat (od definice hardwaru po poslední podprogram). Zásadní části jsou *databáze*, *IO konfigurace*, *procesy* a případně *komunikace*, pokud ji budeme využívat.

V databázi jsou uloženy uživatelem definované proměnné a aliasy. IO konfigurace zobrazuje fyzické adresy vstupů a výstupů, které má zvolený ŘS k dispozici. Složka komunikace skrývá definice všech komunikačních protokolů, kterými může systém komunikovat. Zásadní je složka s procesy, kde lze najít a definovat veškeré procesy, které bude ŘS vykonávat. V provozním deníku je seznam předdefinovaných chyb, nebo hlášení, které může systém jednoduše volat v případě konkrétních událostí, tento seznam lze doplnit nebo úplně změnit. Podprogramy a funkční bloky tvoří přehledný seznam položek, které si uživatel sám definoval. Podprogramy lze volat a funkční bloky přímo vkládat do procesů.



Obrázek 19: Projektová struktura aplikace pro AMiRiS99



### 7.4.2 Rozvržení programu - volba jazyku, procesů, cyklické třídy

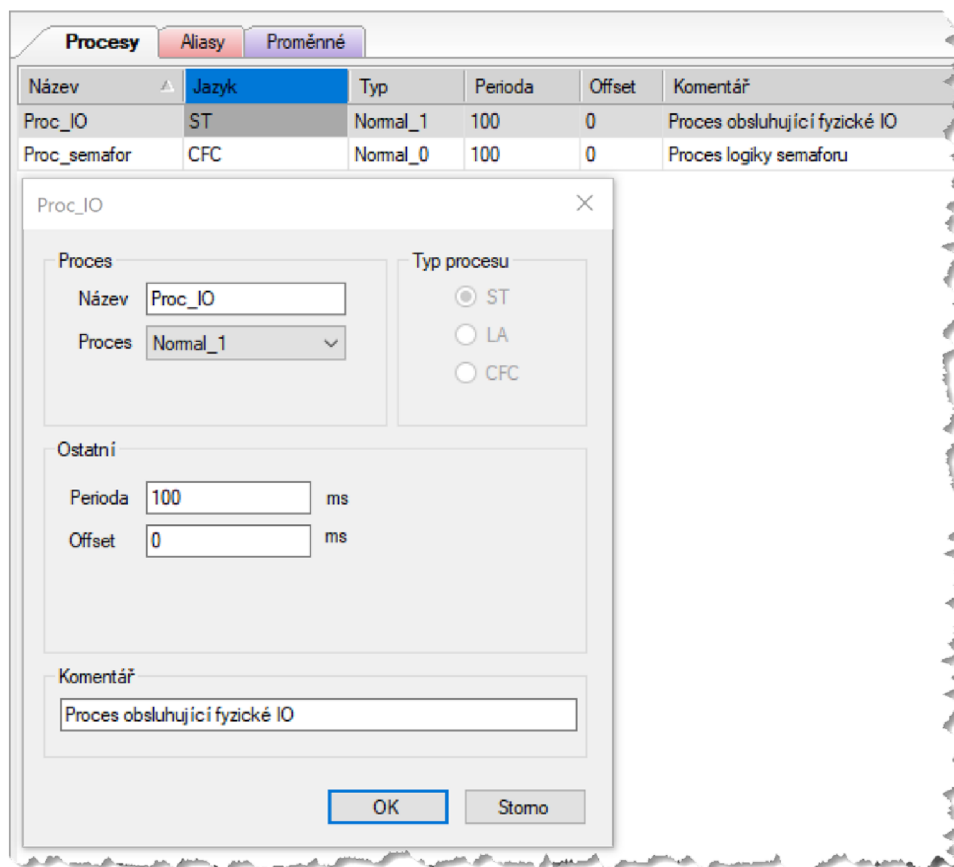
Po založení projektu je třeba rozvrhnout strukturu řídicího programu, respektive jeho procesů. Mělo by jít o logické rozdělení na funkční celky. Rozdělení by mělo mít za následek lepší využití času procesoru a požadovanou rychlost, respektive odezvu ŘS. K tomu se váže i použití programovacích jazyků, které se na rychlosti zpracování rovněž podílí. Zvolený jazyk je víceméně individuální volbou, avšak každý z nich má jistá pozitiva. V editoru PseDet lze programovat pomocí tří jazyků.

#### Programovací jazyky

Prvním je standardní ST, tedy strukturovaný text. Další je jazyk logických adres LA, jedná se ve své podstatě o jazyk známý jako instrukční list IL. Ten nabízí nejširší možnosti. Třetí je však zajímavější a jde o grafickou kombinaci několika jazyků. Jazyk je označen jako CFC, což znamená průběžně vykonávané funkční bloky. Je tvořen kombinací jazyků FBD a LD s volnějším zásadami kreslení schémat. Tento jazyk má největší výhodu v rychlosti vykonávání instrukcí srovnatelnou s LA. Je proto vhodný pro procesy v rychlé cyklické třídě s krátkou periodou, nebo procesy přerušeni. Navíc je obdařen možností vkládání řádků psaných v jazyce ST. Využitím této funkce však ztratí vlastnost rychlého vykonání procesu. Jazyk CFC umí být však velmi názorný a dobře čitelný. Jazyk konkrétního procesu je tedy možné vybrat jak subjektivně, tak podle složitosti řídicí logiky.

#### Procesy a cyklické třídy

U procesu lze zvolit jeho název, typ procesu, periodu spouštění a programovací jazyk, případně komentář. Typ procesu lze zvolit jako *normal*, *quick*, *Hi*, *interrupt* a *init*. *Normal* je běžný proces s prioritou (vyšší číslo procesu = vyšší priorita) s možností nastavení periody (v rozsahu 100 ms až cca 11 dní) a její offset. *Quick* je určen pro obsluhu rychlých dějů s periodou od 5 do 100 ms. Proces *Hi* (HiSpeed) je nejrychlejší proces s periodou od 1 ms, jeho vlastností je, že má nejvyšší prioritu a při jeho přetížení dokáže ochromit i veškerou komunikaci ŘS zahlcením procesoru. Proces typu *interrupt* je proces přerušeni vyvolaný vnější změnou na konkrétním vstupu, ten označuje číslo u procesu. Proces *init* je inicializační, vykonává se pouze jednou (po zapnutí PLC).



Obrázek 20: Přidání nového procesu

Konkrétní případ řízení semaforu lze zvládnout všemi jazyky. Pro názornost a přehlednost řídicího procesu bude vybrán jazyk CFC.

Volba procesů je zde poměrně intuitivní a není třeba jich vytvářet mnoho. Po otevření položky *Procesy* ve struktuře projektu se zobrazí přehledný seznam a pravým tlačítkem myši v něm lze procesy přidávat. Přidají se celkem 2 procesy. Jeden proces bude typu *normal\_0* v jazyce CFC, a proto musí být výchozí proces *Proc00* v jazyce ST nejprve smazán. Ten poté bude obsahovat samotnou logiku řízení semaforu.

Druhý proces, třeba typu *normal\_1*, který bude zapisovat hodnoty proměnných na výstupy, ale také číst vstupy a zapisovat je do příslušných proměnných. Takový proces je nejjednodušší psát v jazyce ST. Procesy lze pro přehlednost doplnit komentářem o jejich funkci. Periody obou procesů jsou nastaveny na 100 ms.

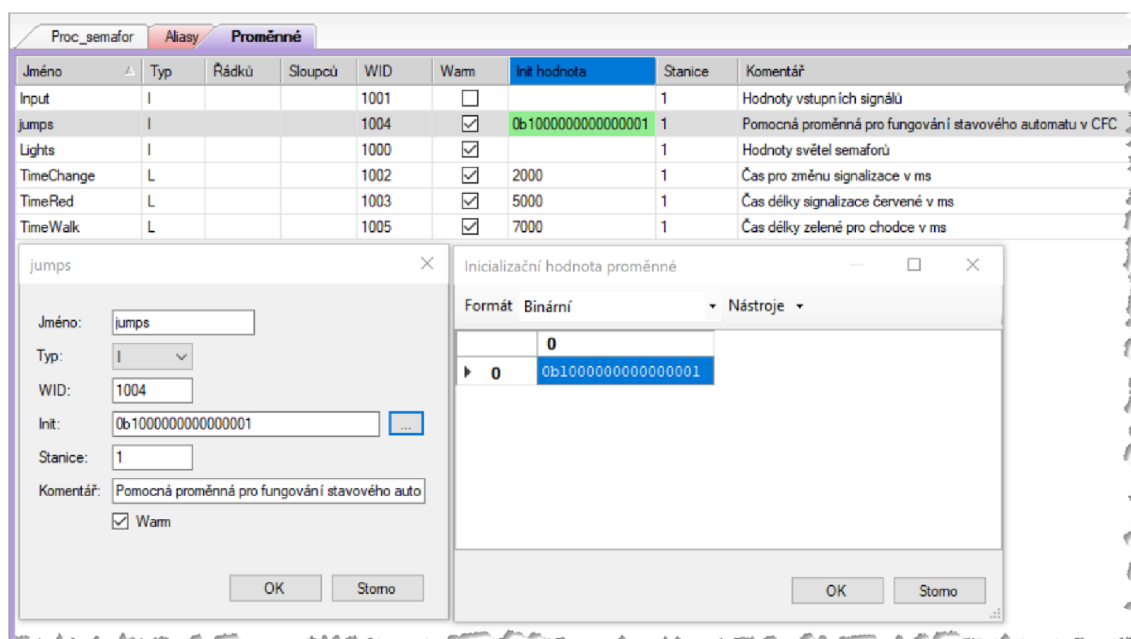
### 7.4.3 Definování proměnných a aliasů

#### Proměnné

Je možné pracovat celkem s 6 typy proměnných. A jsou to *integer*, *long*, *float* a zároveň jejich matice. Jsou označeny *I*, *L*, *F*, *MI*, *ML*, *MF*.

Pravým kliknutím myši v databázi proměnných je možné je vytvořit. Následně lze v seznamu datový typ upravovat, a to i zpětně. V rámci úprav je možné změnit také název, doplnit inicializační hodnotu nebo komentář. Je zde taky zobrazen *WID*, což je unikátní identifikace proměnné v síti DB-net. Je-li zaškrtnuto *warm*, bude do proměnné zapsána inicializační hodnota po každém zapnutí systému. V druhém případě zůstane v proměnné hodnota uložená před vypnutím ŘS.

Pro reprezentaci stavů všech světelných signálů je vytvořena proměnná *Lights* typu *I*, s inicializační hodnotou *0* a lze zapnout parametr *warm*, takže po opětovném zapnutí budou světelné signály ve vypnutém stavu. Druhá vytvořená proměnná představuje získané vstupní signály z tlačítka pro chodce nebo radarové kontroly rychlosti. Je taktéž typu *I* a pojmenovaná *Input*. Za účelem možnosti parametrizování časovačů se založí proměnné, které budou tyto časy definovat v milisekundách (budou typu *L*). Nakonec bude potřeba jedna pomocná proměnná, jejíž bity mohou být použity například pro skoky ve stavovém automatu. Taková proměnná s názvem *jumps* a typu *I* je založena také. Její první bit bude využit zároveň pro inicializaci stavových automatů, proto musí být zaškrtnuta volba *Warm* a *Init* hodnota musí být *0b1000000000000001*. Důvod proč je 16. bit také iniciován na hodnotu 1 je vysvětlen při tvorbě stavového automatu později. Všechny proměnné jsou příslušně okomentované.



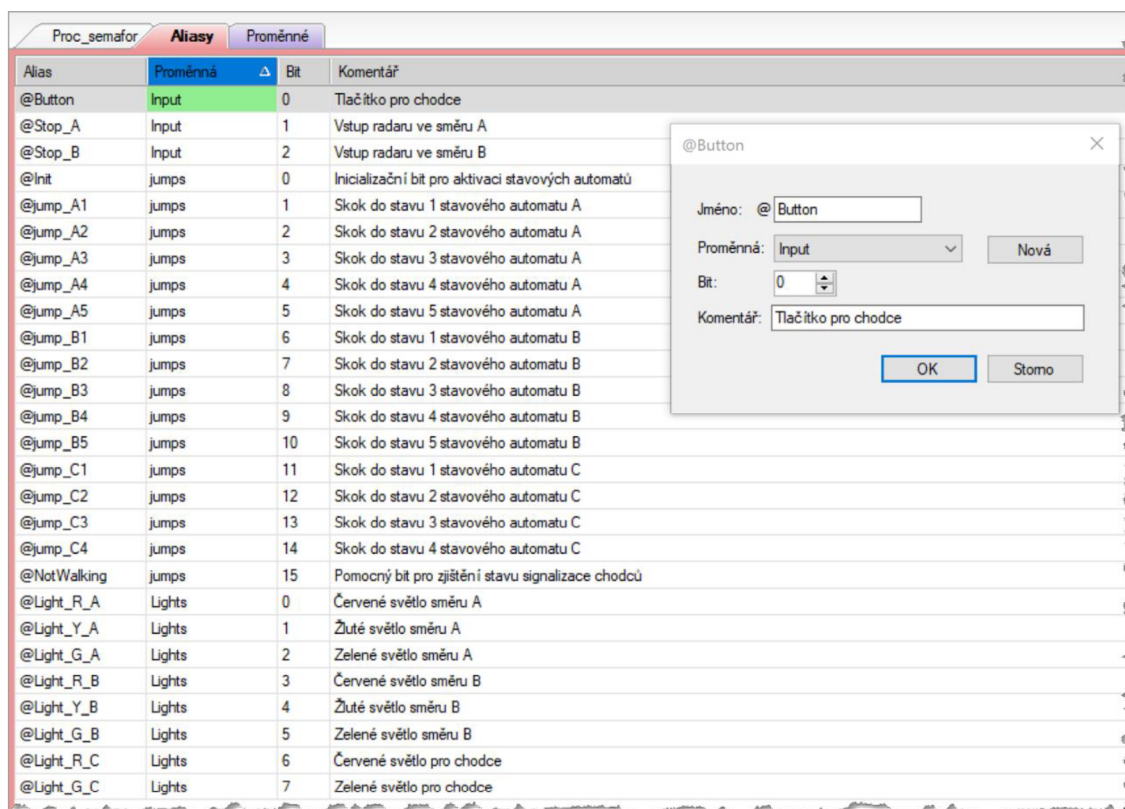
Obrázek 21: Přidání nové proměnné

## Alias

Alias reprezentuje právě jeden konkrétní bit z předem navázané proměnné. Existuje z toho důvodu, protože představuje jediný způsob, jak používat proměnné pravdivostní hodnoty (bool), které jinak v tomto vývojovém prostředí samostatně neexistují. Má tedy pouze binární význam. Pro odlišení od proměnných mají aliasy před vlastním názvem umístěný symbol zavináč (@).

Je vhodné pro každé světlo SSZ nastavit konkrétní alias, který je bude reprezentovat. Je vytvořeno celkem 8 aliasů navázaných na proměnnou *Lights*, respektive postupně na její bity (0-7). Další 3 aliasy jsou vytvořeny pro reprezentaci jednotlivých vstupních signálů a jsou navázány na bity proměnné *Input* (0-2). Je vytvořeno 16 dalších aliasů navázaných na pomocnou proměnnou *jumps* (konkrétně na obrázku 22). Takto pojmenované aliasy usnadňují a hlavně zpřehledňují používání jednotlivých bitů. Všechny jsou jednoznačně pojmenované a příslušně okomentované.

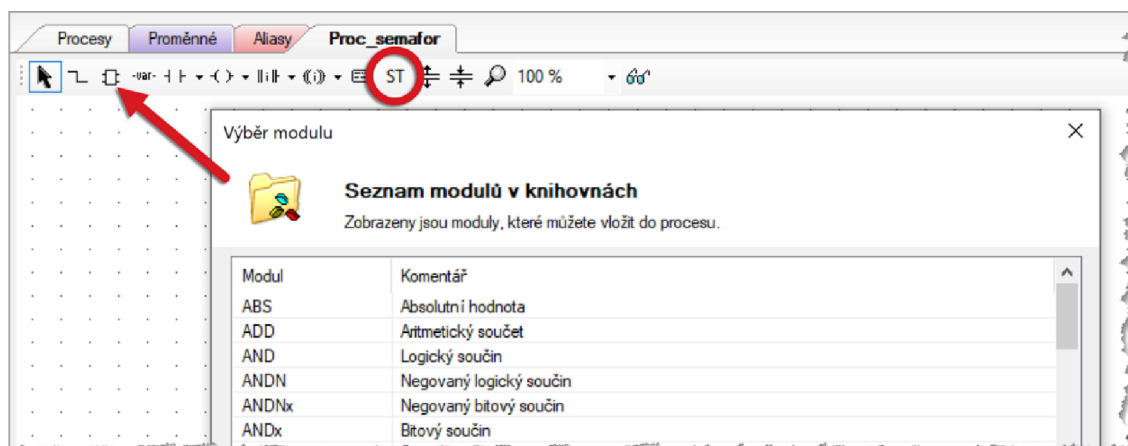
Další aliasy je vždy možné doplnit i během samotné tvorby programu, stejně jako další proměnné.



Obrázek 22: Vytvoření aliasů

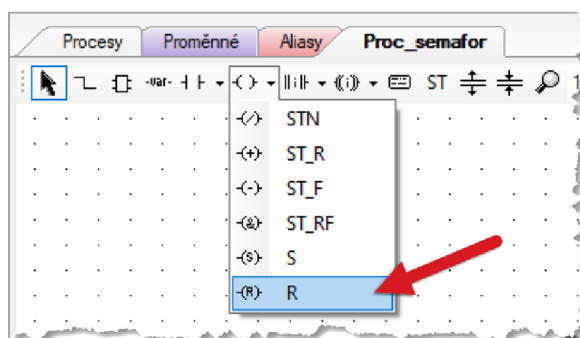
#### 7.4.4 Programování procesů

Po otevření procesu *Proc\_semafor* se zobrazí lišta nástrojů a prostor pro umístění programových bloků a funkcí. Například 3. nástroj v liště umožňuje vkládání modulů (funkčních bloků) do procesu, jak je vidět na obrázku 23. Podobně funguje nástroj ST, ovšem ten vkládá moduly z jazyka ST a ve formě řádku.



Obrázek 23: Programovací prostředí v jazyce CFC

Jiné nástroje z této lišty například vkládají moduly pro čtení nebo zápis hodnoty bitů. Při výběru modulů se zobrazí nabídka umožňující konkretizaci logické funkce. Například na obrázku 24 je označen modul zápisu do bitu, konkretizovaný na funkci reset. V případě pravdivého vstupu s náběžnou hranou dojde k resetu výstupního bitu.



Obrázek 24: Prvky v nástrojové liště

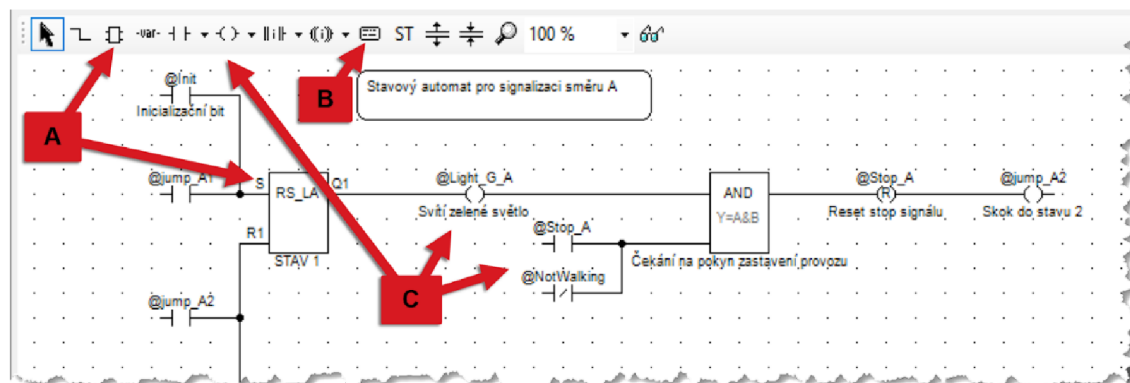
Každý prvek, neboli modul, má v DetStudiu přehlednou a dobře zpracovanou nápovědu, která se po jeho použití automaticky otevře. Nebo lze dvojklikem na prvek opět vyvolat.

Řízení SSZ je možné například stavovým automatem (SA), který je zároveň přehledným způsobem programování. Jelikož signalizace v obou směrech musí fungovat nezávisle, musí být vytvořeno stavových automatů více, zde konkrétně tři.

Pro fungování stavových automatů v jazyce CFC se použije proměnná *jumps*. K počáteční aktivaci slouží první bit, který je nastaven tak, aby nabyl po inicializaci vždy hodnoty 1. Takto je zajištěna inicializace všech stavových automatů, později je



v programu inicializační bit deaktivován. Poté dochází k přepínání stavů podle splnění podmínek. Vytvořen je první stav prvního SA, který je aktivní při nečinnosti, tedy svítí zelená a čeká na pokyn k zastavení provozu. Tento stavový automat řídí signalizaci směru A.

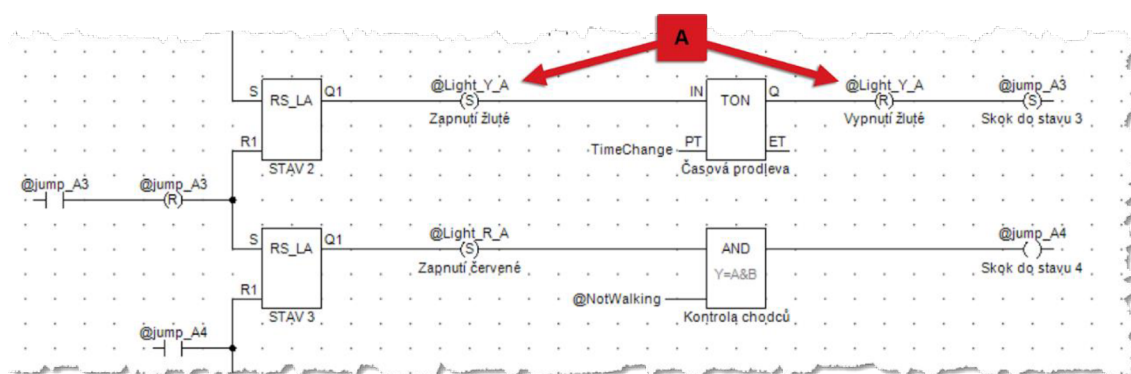


Obrázek 25: Náhled na stav 1 stavového automatu a použití prvků

Ukazatel A na obrázku 25 ukazuje na knihovnu ze které byl vybrán prvek *RS\_LA*. Tento prvek se chová jako RS klopný obvod s dominantním resetem. Ukazatel B směřuje na funkci přidávání volného poznámkového okna, ve kterém je napsán text *Stavový automat pro signalizaci směru A*. Ostatní prvky označené ukazatelem C lze najít přímo na nástrojové liště, nebo po rozbalení modulu a konkretizaci jeho typu. Na liště se nachází prvky jako *LD* (načtení bitu) nebo *ST* (zápis do bitu), dále i jejich alternativy pro jiné druhy proměnných.

Jak je možné vidět na obrázku 25, první stav rozhoduje o stavu zeleného světla signalizace. Stav je ukončen při pravdivém aliasu *@Stop\_A* nebo nepravdivém *@NotWalking*. Jeden reprezentuje překročení rychlosti a druhý stav, kdy chodci nepřechází (nebo nechtějí přejít). Spojují se logikou funkcí OR známou z ladder diagramů a k podmínce *stavu 1* se zapojují logickou funkcí AND, avšak blokem známým z FBD. Poté následuje prvek rozhodující pouze o resetování aliasu *@Stop\_A*. Vysvětlení tohoto provedení je jednoduché. Předpokládá se, že alias bude ovládán i přes komunikaci Modbus, která může mít různě nastavenou periodu zpracování. A tak kvůli zajištění potvrzení tohoto signálu bude signál deaktivován až po reakci podmínky *stavu 1*.

Z důvodu nemožnosti vytvoření uzavřené smyčky v obvodech CFC je nutné stavový automat přepínat mezi stavy pomocí pomocných aliasů (například *@jump\_A2*), které smyčku nahradí. Jeho aktivace vypne RS obvod *stavu 1* a zapne RS obvod *stavu 2*.



Obrázek 26: Náhled stavů 2 a 3, použití prvků "Set" a "Reset" pro ovládní proměnné

Ve *stavu 2* se rozsvěcuje žluté světlo signalizace. Pro jeho rozsvícení je užit prvek *Set*. To z toho důvodu, že nelze použít vícekrát prvek *ST* (použitý v předchozím stavu) pro jeden výstup, protože by se navzájem blokovaly. A jelikož se žluté světlo vyskytuje ve dvou stavech SA (při přechodech zelené a červené), tak by se vzájemně blokovaly. Blokaci způsobuje fakt, že prvek *ST* drží na výstupu zásadně stejnou hodnotu jako na vstupu, a to i případnou hodnotu 0. Protože žluté světlo je aktivované prvkem *Set* je nutné světlo po opuštění stavu zase vypnout prvkem *Reset*, jak upozorňuje ukazatel A na obrázku 26.

Podmínku *stavu 2* tvoří blok *TON*, který má funkci zpoždění náběžné hrany. Tento funkční blok má i druhý vstup pro proměnnou typu *L*, a ta představuje zpoždění v milisekundách. Ke vstupu je připojena proměnná s požadovanou časovou délkou *TimeChange*. Na druhém výstupu funkčního bloku lze sledovat aktuální hodnotu časovače, je-li to třeba.

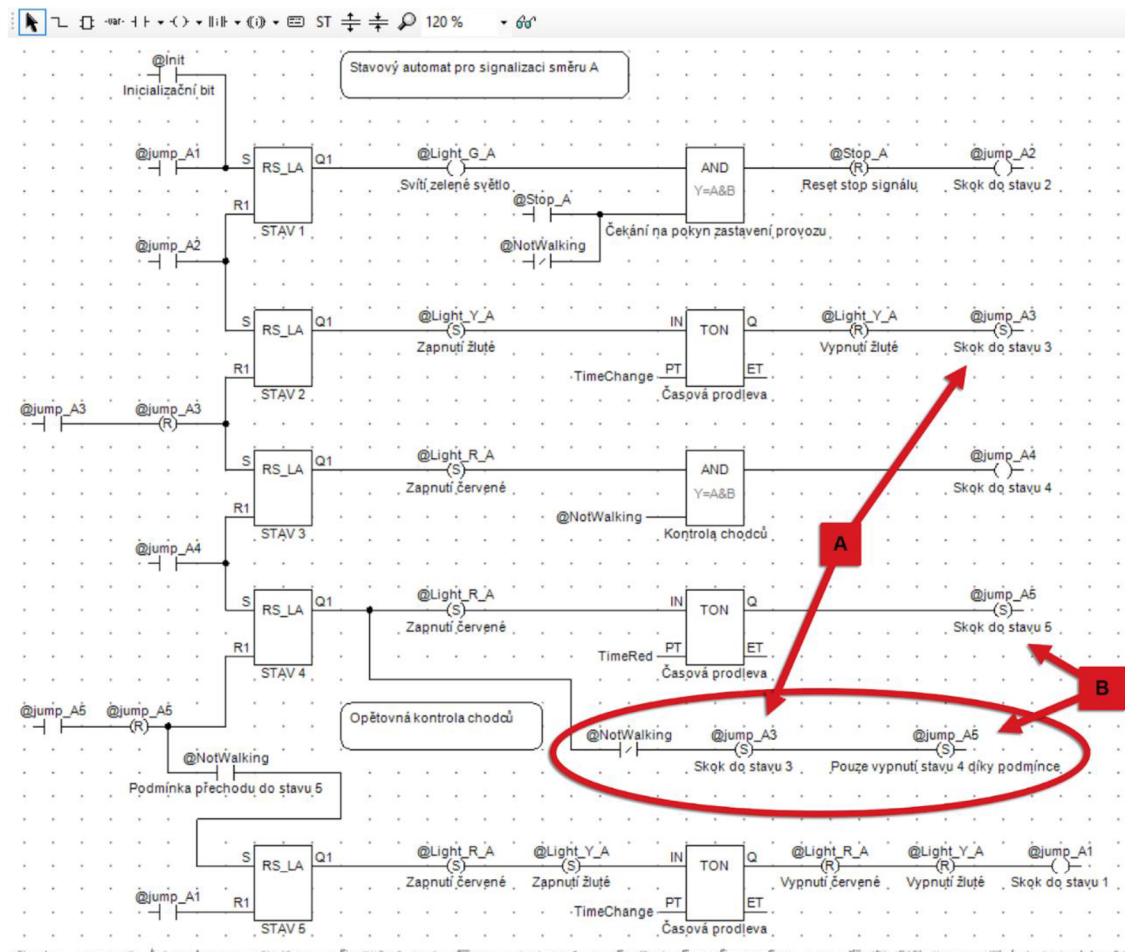
Podobný problém s blokací prvkem *ST* trpí i skok do *stavu 3*, jelikož z důvodu dvojitě kontroly chodců je třeba se do toho stavu umět vracet. Proto je použit rovněž prvek *Set* s následným resetem po provedení skoku (viz. ukazatel A na obrázku 27).

Problém s blokováním lze řešit různými způsoby. Použitím několika dalších pomocných aliasů aktivující jediný společný výstup, nebo "prodrátováním" na jediný použitý výstup. Prodrátování by ale snížilo přehlednost kvůli faktu, že signálové spoje zde nejde odkazovat nebo jinak vizuálně přerušit.

Na obrázku 27 je zobrazen kompletní stavový automat směru A. Ukazatel B na něm zvýrazňuje 2 skoky do stavu 5. V druhém případě však ke skoku prakticky nedojde a je zde umístěn z důvodu potřeby resetování stavu 4. Aby nedošlo zároveň k aktivaci stavu 5 je před jeho RS obvodem podmínka, která přechod do stavu blokuje, pokud nemají chodci trvalou červenou signalizaci. Na obrázku je zvýrazněna *Opětovná kontrola chodců*. Tento obvod kontroluje, zdali nedošlo k opuštění stavu trvalé červené signalizace chodců během časové prodlevy udržující červenou signalizaci směru A po danou dobu. Bez této kontroly by mohlo za jistých podmínek dojít k nebezpečné situaci, v níž by signalizace pro vozidla i chodce mohla svítit zeleně.

Funkce všech stavů je patrná z předchozího obrázku, kde jsou stavy a jejich jednotlivé podmínky náležitě okomentované.

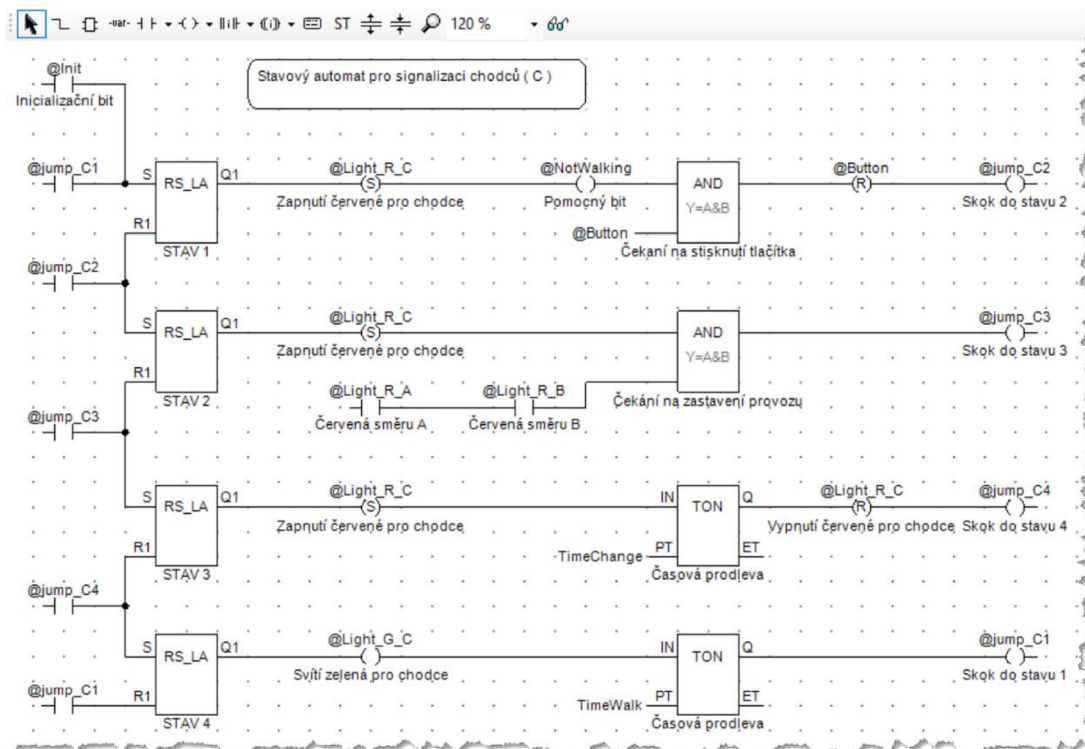
Stejným způsobem je vytvořen stavový automat pro směr B, kde jsou pouze změněny aliasy, které se týkají směru B.



Obrázek 27: Náhled stavového automatu signalizace směru A, řešení skoků mezi stavy

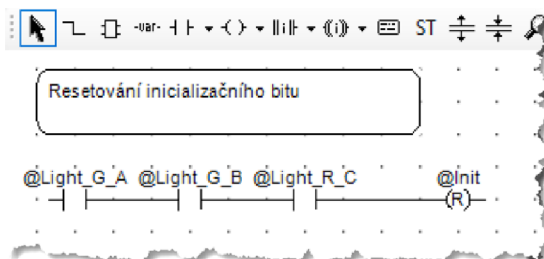
Podobným způsobem, jak bylo uvedeno, je vytvořen stavový automat signalizace pro chodce uvedený na obrázku 28. Opět je jeho funkce včetně stavových podmínek nejnázornější právě z odkazovaného obrázku.





Obrázek 28: Náhled stavového automatu řídicího signalizaci pro chodce

Nakonec musí být proces doplněn o důležitý obvod. Tento obvod je podstatný pouze po zapnutí řídicího systému. Po splnění podmínek, až inicializační alias *@Init* aktivuje první stavy všech tří stavových automatů, dojde k jeho resetování. Poté již není tento alias nikdy aktivní. (Obrázek 29.)

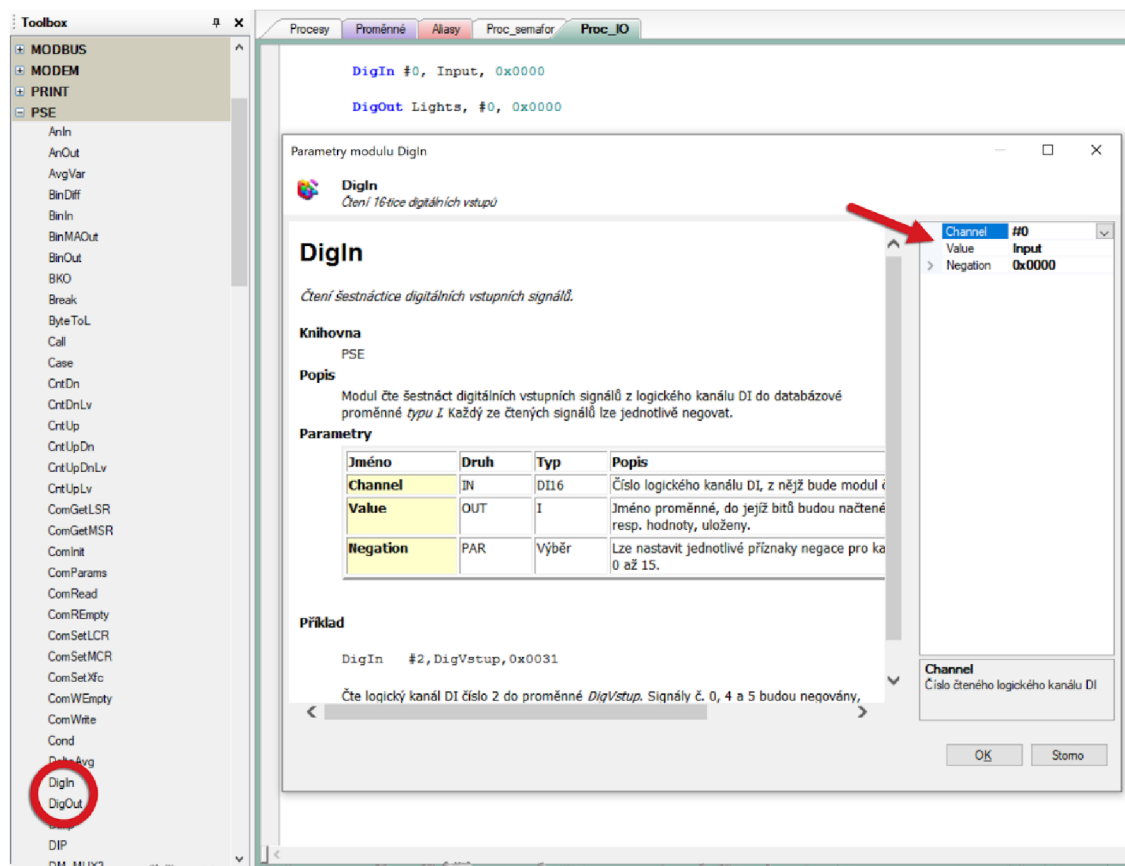


Obrázek 29: Provedení obvodu řešícího inicializaci SA

### 7.4.5 Přirazení proměnných k I/O

Celý dříve vytvořený program pracuje pouze s proměnnými. Právě to mu propůjčuje jistou univerzalitu, nebo flexibilitu. Program tak není vázaný na konkrétní řídicí systém a je možné jej použít prakticky na jakémkoliv jiném ŘS společnosti AMiT bez úprav. Svázání s konkrétním systémem je provedeno několika řádky v ST kódu v jiném procesu, který je určen výhradně pro čtení nebo zápis na vstupy a výstupy.

Po otevření procesu *Proc\_IO* je z toolboxu nacházejícího se na levé straně, z jeho knihovny PSE, použit modul *DigIn* a modul *DigOut*. Modul *DigIn* zajišťuje zápis hodnot fyzických vstupů do přidělené proměnné. Druhý modul *DigOut* naopak zajišťuje zapsání hodnot proměnné na fyzické výstupy. V každém modulu je třeba přiřadit použitý fyzický kanál I/O s proměnnou, která jej reprezentuje v programu. Pro oba je možné použít nulté kanály, ty jsou vždy první svorkovnicí na systému. Pro editaci již vložených modulů do kódu pomocí GUI (nápopovědy k modulu) slouží kl. zkratka *Ctrl+M* se současným umístěním kurzoru na modul.

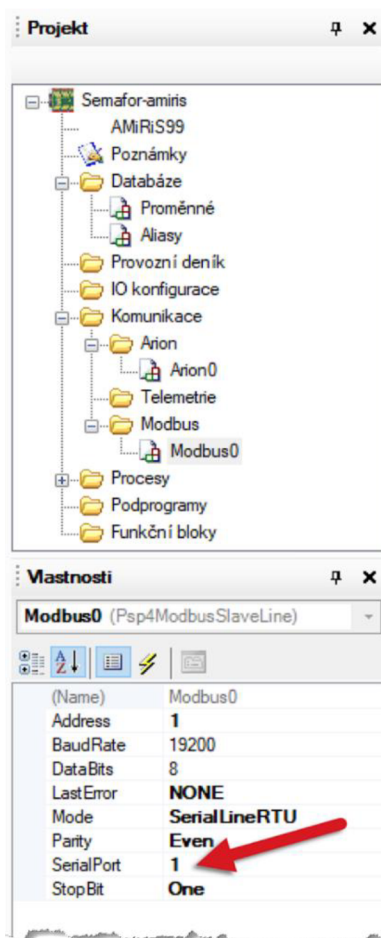


Obrázek 30: Použití GUI při programování v jazyce ST v procesu *Proc\_IO*, použití modulu *DigIn*

Jelikož starší editor *PseDet* neumí simulovat, může být program ověřen pouze na přítomnost syntaktických chyb. V hlavní nástrojové liště programu v záložce *Generace* se vybere *Generuj vše*. *DetStudio* tím vygeneruje aplikaci z otevřeného projektu, v případě výskytu chyb je vypíše.

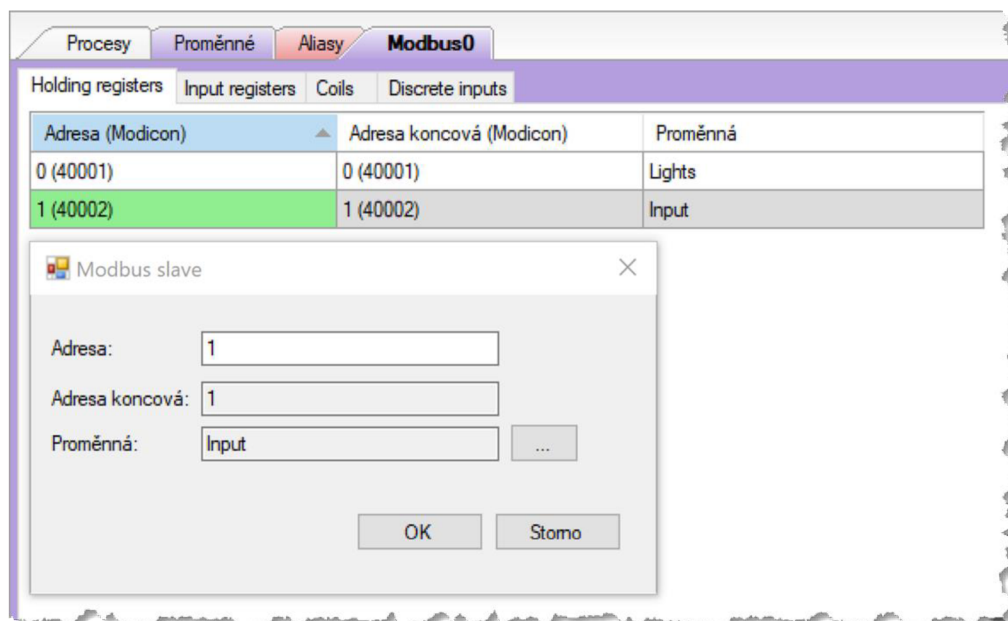
### 7.4.6 Nastavení komunikace

Pro komunikaci s druhým modulem je nutností ji správně nakonfigurovat. Využito může být pouze rozhraní RS485 a protokolu Modbus RTU. Konfigurace komunikace začíná přidáním příslušné položky ve složce komunikace, aktuálně ve složce *Modbus*. Do ní je pravým tlačítkem myši přidáno postavení *Slave*. Tím vznikne položka *Modbus0*, která otevře možnosti konkrétnímu nastavení protokolu. Pouhým jejím označením se pod strukturou projektu objeví vlastnosti této komunikace. Jediná úprava zde spočívá ve změně čísla položky *SerialPort* na *1*. To z důvodu, že u AMiRiS99 je původní port *0* označen RS232. Číslo portu *1* potom představuje RS485.



Obrázek 31: Nastavení Modbus komunikace

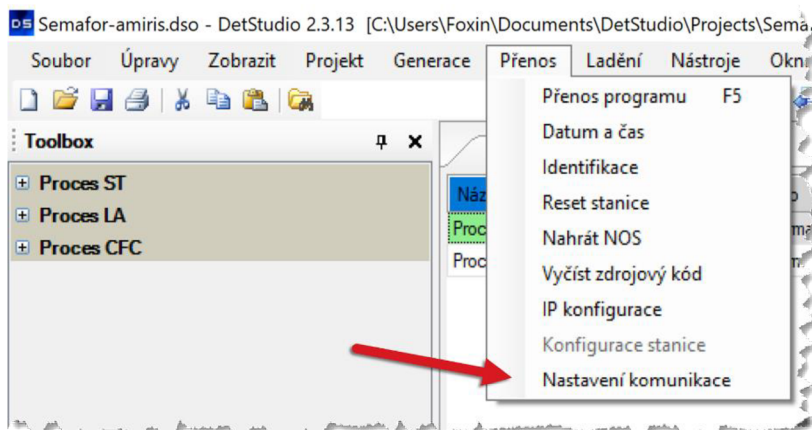
Rozkliknutím položky *Modbus0* ve struktuře projektu se objeví okno s aktivní záložkou  *Holding registers*. Tady se přiřadí ke komunikační adrese (Modicon) proměnná a její hodnota bude komunikována (vyobrazeno na obrázku 32). Takto se nastaví adresy proměnným *Lights* a *Input*. Tím dovolíme modulu AMR-OP83, aby je mohl následně číst a měnit. To umožní vzdálené ovládání a vizualizaci aplikace.



Obrázek 32: Přidávání Holding registrů do Modbus komunikace

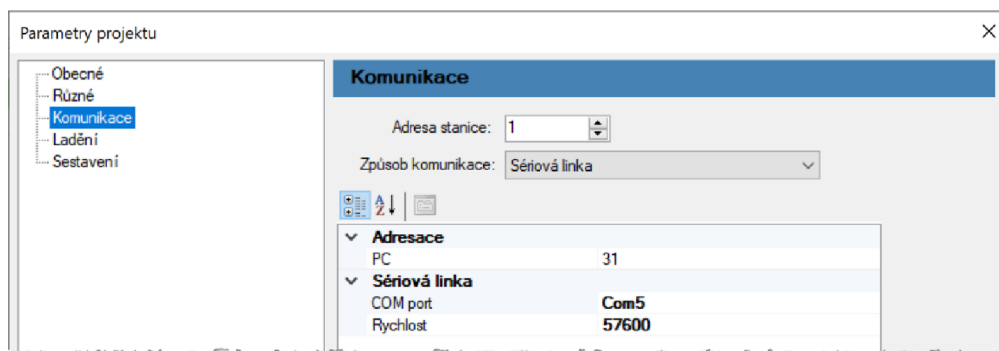
#### 7.4.7 Zavádění programu do jednotky AMiRiS99

Zavádění do jednotky AMiRiS99 se provádí pomocí sériového portu RS232. Pokud takový port počítač nemá, je nutnost využít například USB převodníku.



Obrázek 33: Nastavení komunikace jednotky AMiRiS99 pro zavedení aplikace

V DetStudiosu se musí nastavit konkrétní COM port, který identifikuje sériový port použitý pro komunikaci s AMiRiS99. Po otevření *Nastavení komunikace* (dle obrázku 33) se tento port musí správně nastavit a to i adresa stanice a přenosová rychlost. Tyto dva parametry jsou nastavitelné hardwarovými DIP přepínači na stanici.



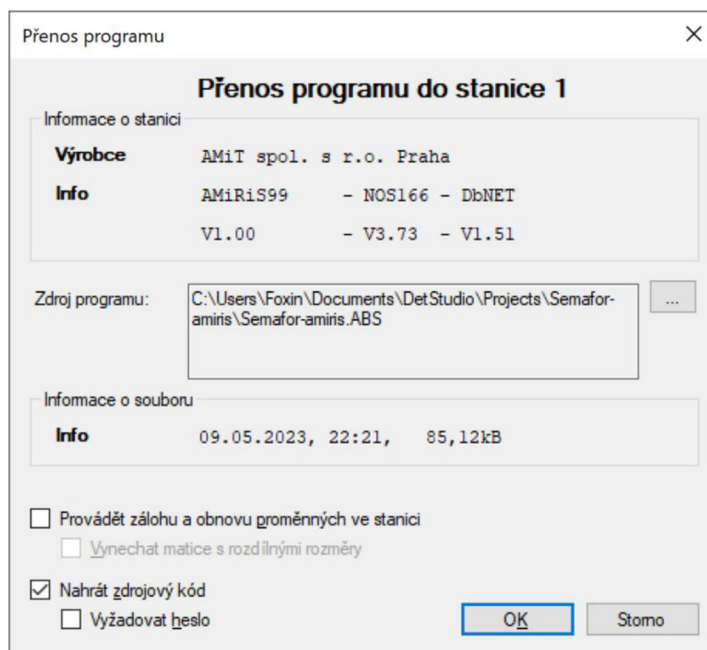
Obrázek 34: Konfigurace sériové linky pro komunikaci s PC

Pro komunikaci se využívá systému DB-Net, v něm musí mít stanice nastavenou unikátní adresu. To právě pomocí DIP přepínačů označovaných SW. Nejvyšší přenosová rychlost je dána konkrétním systémem. Některý umožňuje jen 57600 Bd a některý 115200 Bd.

SW5	SW4	SW3	SW2	SW1	Adresa systému v síti DB-Net	SW6	SW7	Rychlost [Bd]
OFF	OFF	OFF	OFF	ON	0	OFF	OFF	9600
OFF	OFF	ON	ON	OFF	1	ON	OFF	19200
OFF	ON	OFF	ON	OFF	2	OFF	ON	38400
ON	OFF	ON	ON	ON	3	ON	ON	57600 / 115200*)
ON	ON	ON	ON	ON	4			
					5			
					6			
					7			
					8			
					9			
					10			
					11			
					12			
					13			
					14			
					15			
					16			
					17			
					18			
					19			
					20			
					21			
					22			
					23			
					24			
					25			
					26			
					27			
					28			
					29			
					30			
					31			

Obrázek 35: Návod pro význam konfigurace DIP přepínačů

Jakmile je vše nastaveno, přejde se k samotnému zavádění aplikace. V hlavní nástrojové liště programu v záložce *Generace* se vybere *Generuj vše*. DetStudio tím vygeneruje aplikaci z otevřeného projektu a zobrazí se okno s volbou *Přenést do stanice*. Poté se zobrazí okno s informacemi o stanici, a je možné potvrdit přenos (vyobrazeno na obrázku 36). Po dokončení nahrávání je uživatel o úspěchu informován.

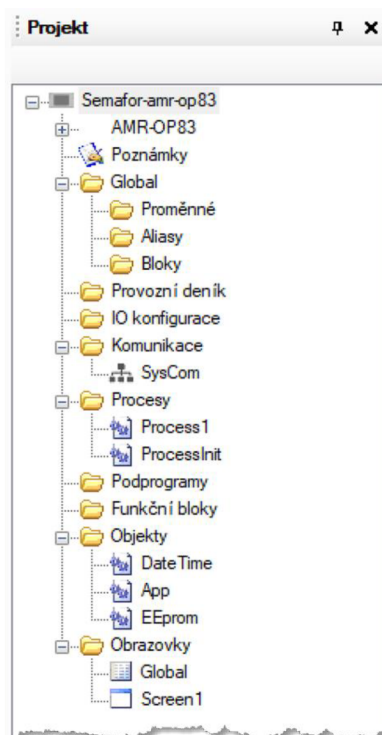


Obrázek 36: Dialogové okno pro zavádění aplikace

## 7.5 Tvorba uživatelského programu pro AMR-OP83 (vizualizační část)

### 7.5.1 Založení projektu

Založení projektu je popsáno v předchozí podkapitole pro řídicí systém AMiRiS99, postup je naprosto totožný. Snad jen s rozdílným výsledkem v tom, že při volbě AMR-OP83 je projekt otevřen v editoru EsiDet. A jelikož jde o odlišné zařízení s vlastním displejem, vygeneruje se jiná struktura projektu.



Obrázek 37: Projektová struktura aplikace pro AMR-OP83



## 7.5.2 Rozvržení programu - obrazovek

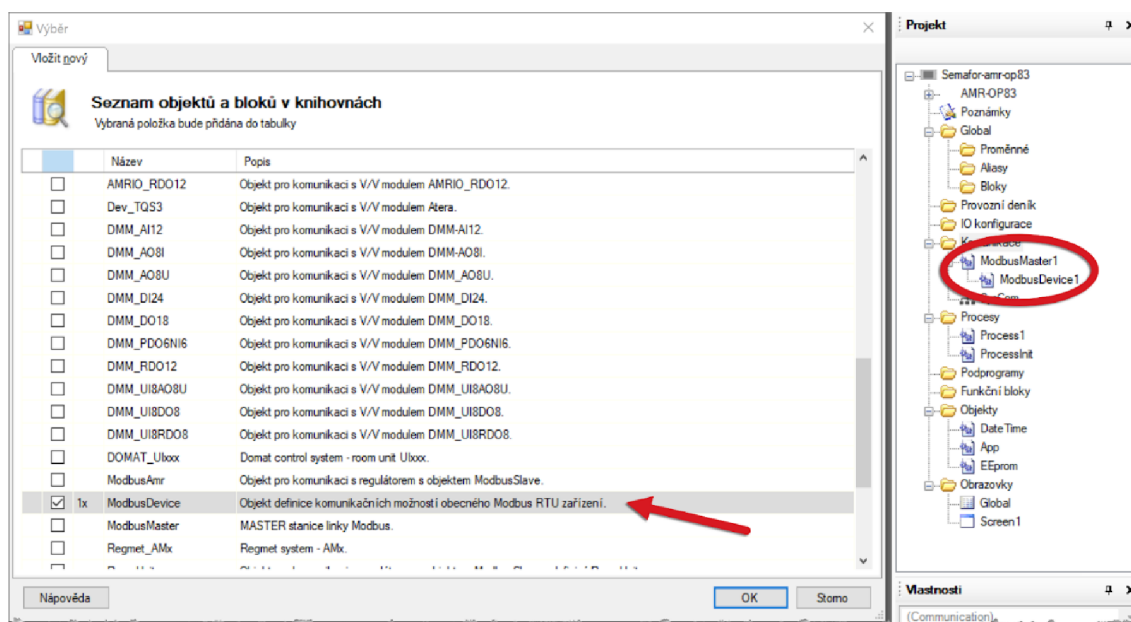
Program pro vizualizaci a simulaci je tvořen odlišným způsobem než řídicí. Je možné se držet pouze grafického editoru obrazovek a nepracovat přímo s procesy.

Vizualizační panel v této aplikaci po zapnutí bez žádného nutného zásahu zobrazí aktuální stavy světelných signalizací. Na displeji se budou zobrazovat i tlačítka umožňující simulaci vstupů. Tlačítka budou 3, pro zastavení dopravy ve směru A, pro zastavení dopravy ve směru B a tlačítko pro chodce. To zastaví dopravu obou směrů a umožní přejít po přechodu. Vystačí zde jedna jednoduchá obrazovka.

Pro správnou funkci je nutné propojení s řídicím systémem AMiRiS99 pomocí RS485, aby bylo možné číst nebo měnit proměnné tohoto systému. Zobrazení stavů světelné signalizace bude sloužit proměnná *Lights* z předchozího projektu. Simulace vstupů bude zajištěna změnou proměnné *Input*, taktéž z předchozího projektu.

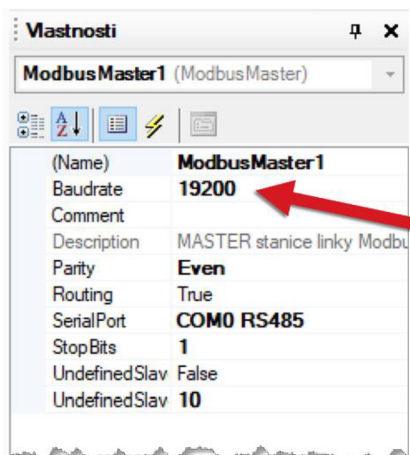
## 7.5.3 Nastavení komunikace

Nastavení komunikace je v tomto případě od předchozího postupu odlišné. Volbou na položce *Komunikace* ve struktuře projektu se zvolí *přidat objekt*. Přidá se objekt s názvem *ModbusDevice* z právě otevřené knihovny. Objekt se zobrazí ve stromové struktuře jako na obrázku 38.



Obrázek 38: Seznam komunikačních objektů

Označením položky *ModbusMaster1* se níže otevřou její vlastnosti a je nutné upravit Baudrate na stejnou hodnotu jako je nastavena v projektu pro AMiRiS99.



Obrázek 39: Nastavení Modbus komunikace pro AMR-OP83

Rozkliknutím položky *ModbusDevice1* se zobrazí seznam Modbus registrů. Volbou pravého tlačítka myši do něj přidáme holding registry odpovídající položkám holding registrů v projektu pro AMiRiS99. A to zejména včetně odpovídajících čísel registrů.

Jméno	Typ	Řádků	Sloupců	Registr	Druh	Priorita	MB Funkce čtení	MB Funkce zápis	Komentář
Lights	INT			0	Holding register	Normal	Function_03	Function_16	Světelné signalizace
Inputs	INT			1	Holding register	Normal	Function_03	Function_16	Vstupní signály

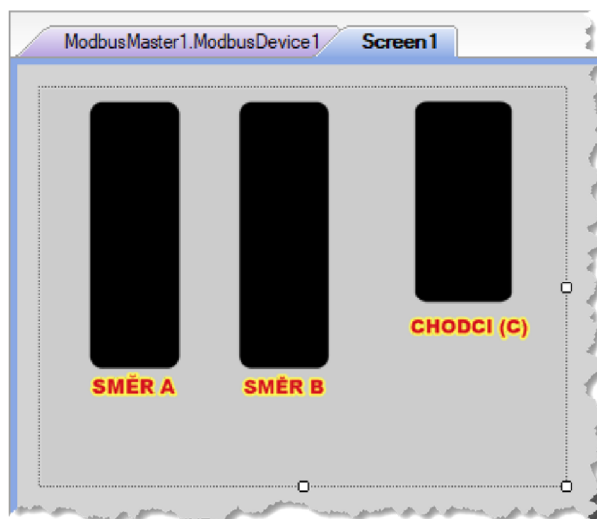
Obrázek 40: Nastavené holding registry Modbus komunikace

Jelikož se vytvořené registry chovají v projektu jako proměnné, není třeba je znovu vytvářet.

#### 7.5.4 Tvorba obrazovky

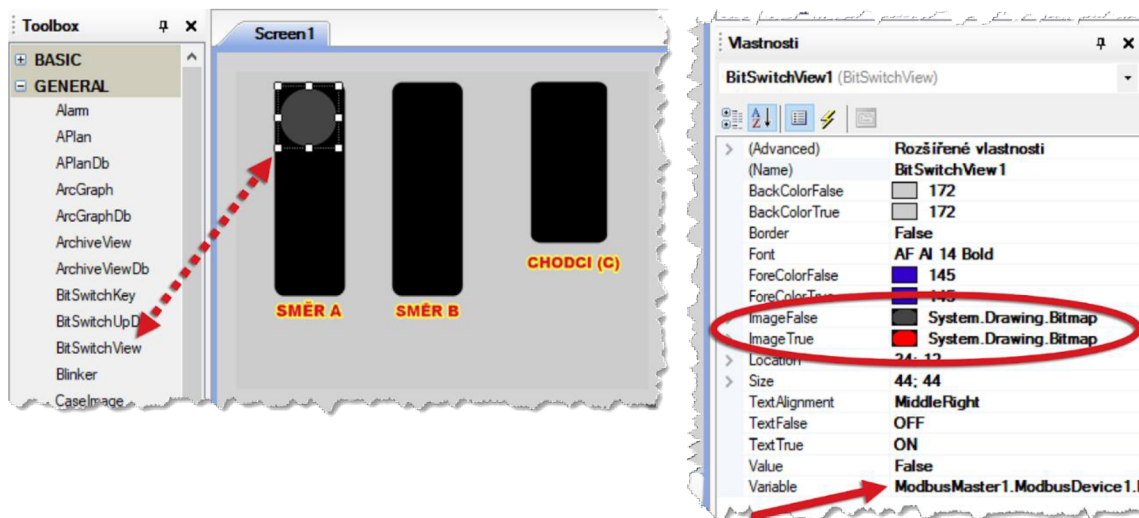
Programování jednotky, která disponuje prakticky pouze displejem je odlišné. Pro vizualizaci a tvorbu ovládacích prvků je zde prostředí graficky přizpůsobené danému displeji. Programování vizuálních prvků probíhá přetahováním modulů z knihovny. U každého prvku je poté v jeho vlastnostech možné nastavit jeho vzhled a chování. Některé prvky je možné nebo přímo nutné skriptovat. Skriptování umožní naprogramovat nejrůznější chování i jednoduchým prvkům. Je zde aplikován princip tvoření obrazovek. To znamená, že je možno definovat několik obrazovek, mezi kterými lze přecházet. Toho může být docíleno přes kontextové menu volby obrazovek, nebo funkcemi některých prvků. Také není problém definovat přístupová hesla a uživatele.





Obrázek 41: Grafické vývojové prostředí obrazovek, příklad použitého pozadí

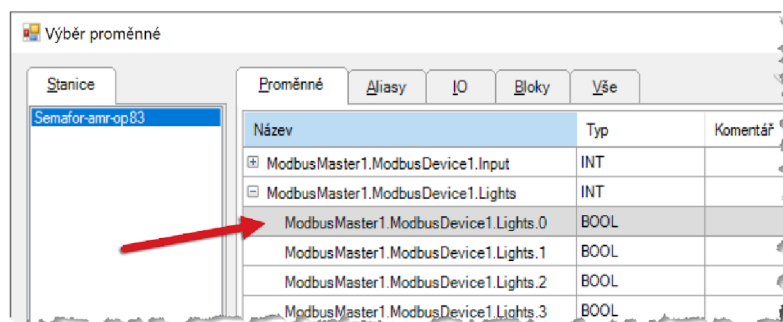
Pro jednoduchou vizualizaci tří semaforů není třeba vytvářet obrazovek více a využije se pouze jedna, výchozí. Prostedí sice umožňuje používat různé grafické objekty, avšak výrobcem je důrazně doporučeno využívat obrázky umístěné na pozadí obrazovky. Na nich mít požadovaný grafický vzhled a přes ně umísťovat jen funkční prvky. Pro nahrání pozadí je třeba otevřít položku `Screen1`, představující onu jedinou obrazovku. Její vlastnosti obsahují položku `BackgroundImage`, to je tedy pozadí dané obrazovky. Pozadí je nutné zhotovit v libovolném grafickém softwaru a v rozlišení displeje (320x240). Konkrétně v naší aplikaci je vytvořeno tak, aby znázorňovalo 3 signalizační zařízení včetně případných popisků (obrázek 41).



Obrázek 42: Umístění prvku `BitSwitchView`, nastavení jeho vzhledu a přiřazení proměnné

Konkrétní světla zde bude tvořit prvek `BitSwitchView`, který je schopen zobrazovat dva stavy. A to i pomocí nahrané grafiky, jako je uskutečněno a zvýrazněno na obrázku 42. Pro zobrazení stavu světelné signalizace je potřeba vytvořit grafiku, která bude stavy názorně zobrazovat. I zde je důležité rozlišení, protože ve vývojovém prostředí nelze

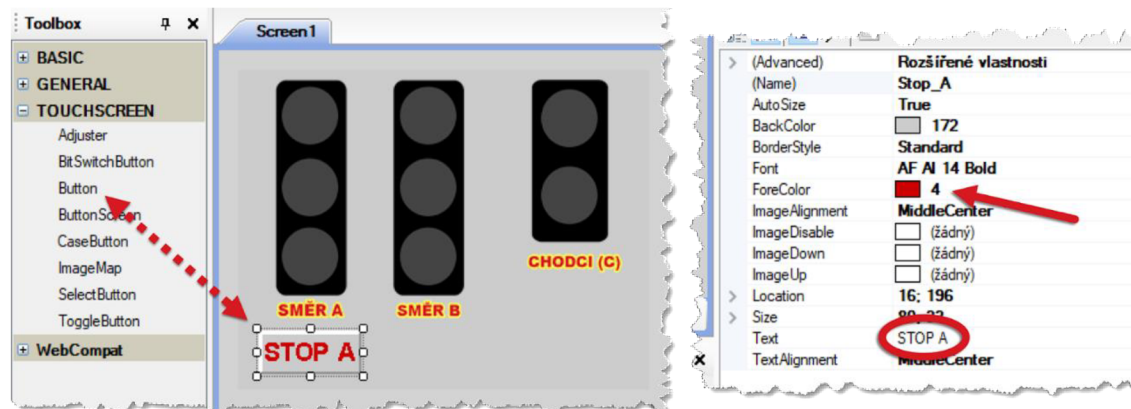
obrázky škálovat. V tomto případě je ideální velikostí rozlišení 44x44 pixelů. Prvek *BitSwitchView* se vloží přetažením z knihovny na konkrétní polohu na obrazovce. V jeho vlastnostech může být pojmenován a je nutné mu přiřadit konkrétní bit z Modbus komunikace, který má ve vizualizaci reprezentovat. Je třeba vycházet ze vzájemně nastavené komunikace, proměnných a adres protokolu Modbus v obou řídicích systémech tak, aby volba byla správná. První odpovídá *ModbusMaster1.ModbusDevice1.Lights.0*, označen je na obrázku 43.



Obrázek 43: Výběr proměnné z Modbus holding registru

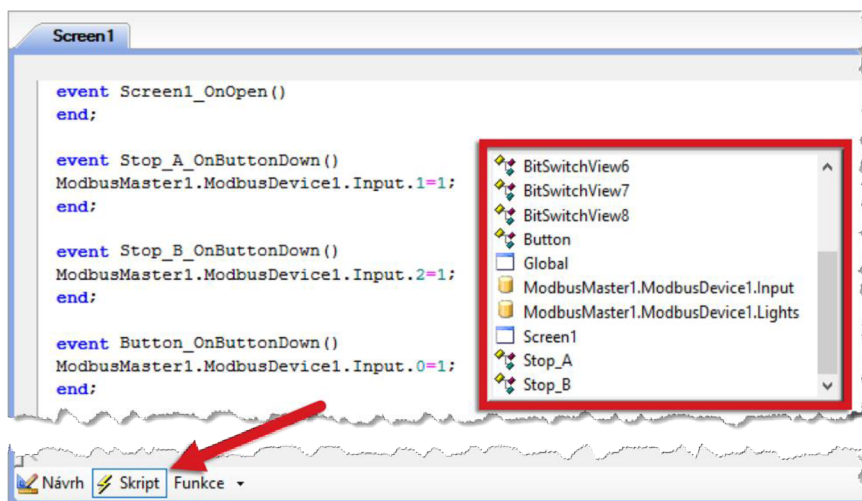
Stejným způsobem je umístěno a nadefinováno zbylých 7 světél.

Podobně je přidán i prvek *Button*. Je pojmenován, má upravený zobrazovaný text na *STOP A* a je barevně přizpůsoben. Tlačítko má simulovat zastavení provozu ve směru A z důvodu překročení povolené rychlosti. Stejným způsobem jsou vytvořena další 2 tlačítka. Jedno pro zastavení směru B, druhé tlačítko je pro chodce.



Obrázek 44: Použití prvku Button a jeho přizpůsobení

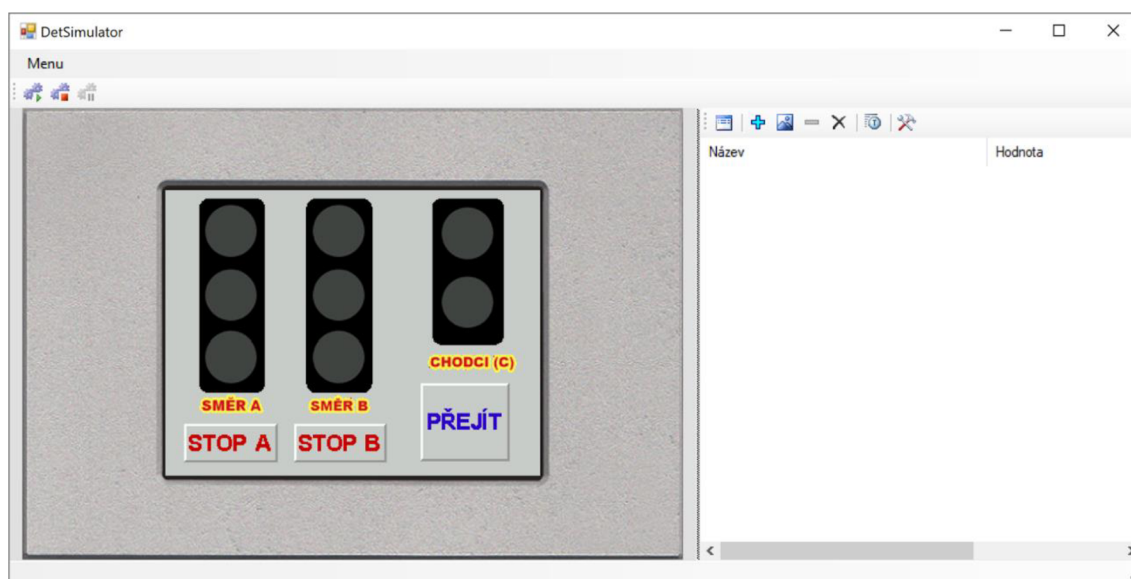
Prvek *Button* je jedním z těch, které se musí skriptovat. Pro přístup do skriptu dané obrazovky slouží tlačítko na spodní straně vývojového okna, jak ukazuje obrázek 45. Vedlejším tlačítkem *Návrh* je uskutečněn návrat ke grafickému náhledu. Na obrázku je zobrazena v červeném rámečku funkce „našeptávače“ otevřeného pomocí kl. zkratky *Ctrl+J*. Ten usnadňuje tvorbu skriptů. Například skript pro tlačítko *Stop\_A* při události *OnButtonDown* (tlačítko dole) zapíše do bitu *ModbusMaster1.ModbusDevice1.Input.1* hodnotu 1. A obdobně u tlačítek ostatních.



Obrázek 45: Použití skriptování, našeptávací funkce

Výsledná vizualizace je vyobrazena na obrázku 46. V případě, kdy jsou proměnné zpracovávány v jiném řídicím systému není simulace bez zásahu do nastavení obrazovek funkční. Respektive funkční je, ale signalizace se provádět nebude, jelikož ji řídí jiný systém.

Do simulace se dá dostat přes záložku *Generace* v hlavní nástrojové liště zvolením *Simulátor* a použitím příkazu *Generuj vše*. V tomto okamžiku dojde k vygenerování vytvořené aplikace a použití simulátoru DetSimulátor. Pokud se v projektu objeví syntaktické chyby, k vygenerování nedojde a uživatel bude na tyto chyby upozorněn.

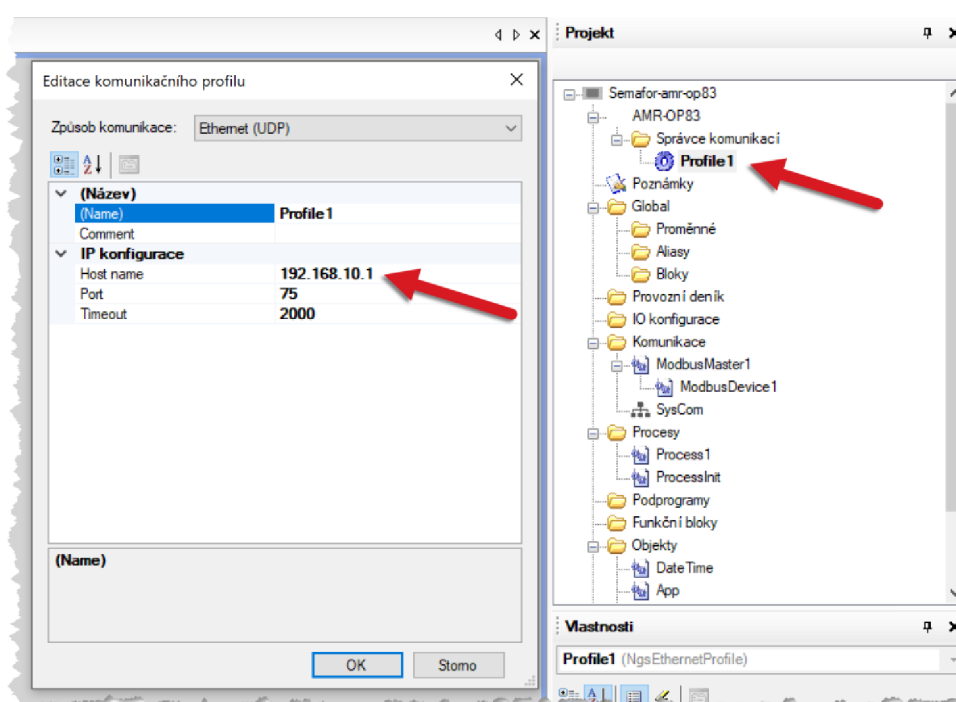


Obrázek 46: Simulátor pro operační panely

Přes marné pokusy “rozsvítit” signalizace nebylo vůbec jasné, proč nefunguje způsob signalizace, který byl použit. Nakonec se ukázalo, že po nahrání aplikace přímo do hardwaru vše fungovalo jak původně mělo. Tímto byla objevena zřejmá chyba v dané verzi softwaru. Z toho důvodu nebylo možné se simulaci podrobněji věnovat.

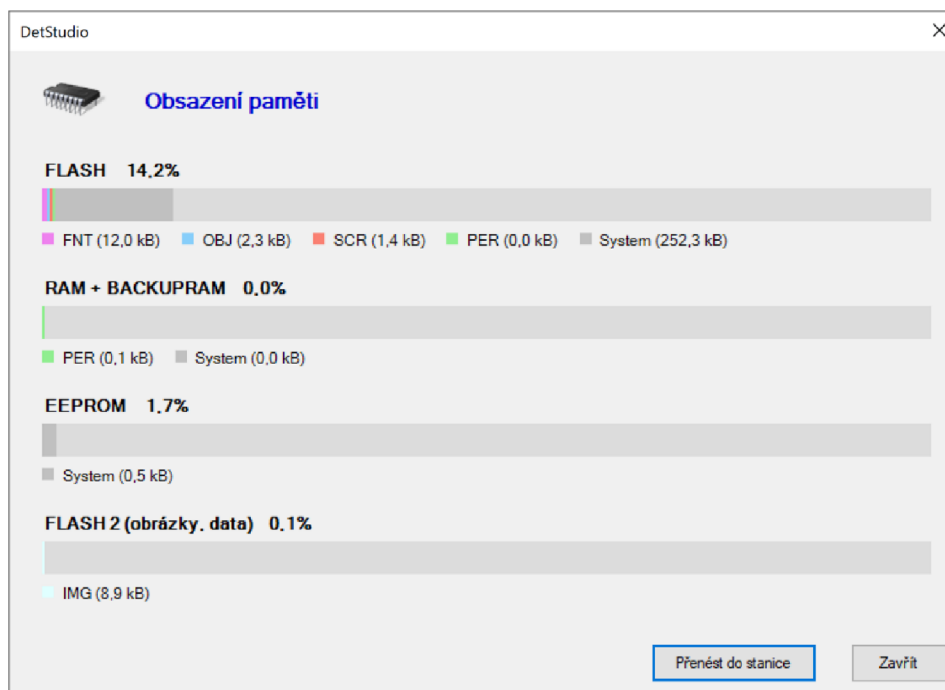
### 7.5.5 Zavádění programu do jednotky

Před samotným nahráním aplikace do systému je nutné správně nastavit IP adresu. Komunikace s PC zde probíhá přes ethernet. Modul AMR-OP83 může být připojen přímo k počítači, nebo do lokální sítě. V případě, že se jedná o první nahrávání aplikace, modul po zapnutí zobrazuje svou aktuální adresu, podle které je možné správně nastavit profil ve Správci komunikací umístěném ve struktuře projektu, jak znázorňuje obrázek 47.



Obrázek 47: Nastavení komunikace pro zavedení aplikace pomocí Ethernetu

Po nastavení komunikačního profilu se v hlavní nástrojové liště v záložce *Generace* navolí *Stanice* a vybere *Generuj vše*. DetStudio tím vygeneruje aplikaci z otevřeného projektu, kterou umožní do stanice nahrát. Zobrazí se okno (obrázek 48) s předpokládaným obsazením paměti dané stanice (zvoleného modulu AMR-OP83). Tlačítkem *Přenést do stanice* se zobrazí ještě okno s dalšími údaji a po jeho potvrzení dojde k zavedení aplikace a stanice se následně restartuje.



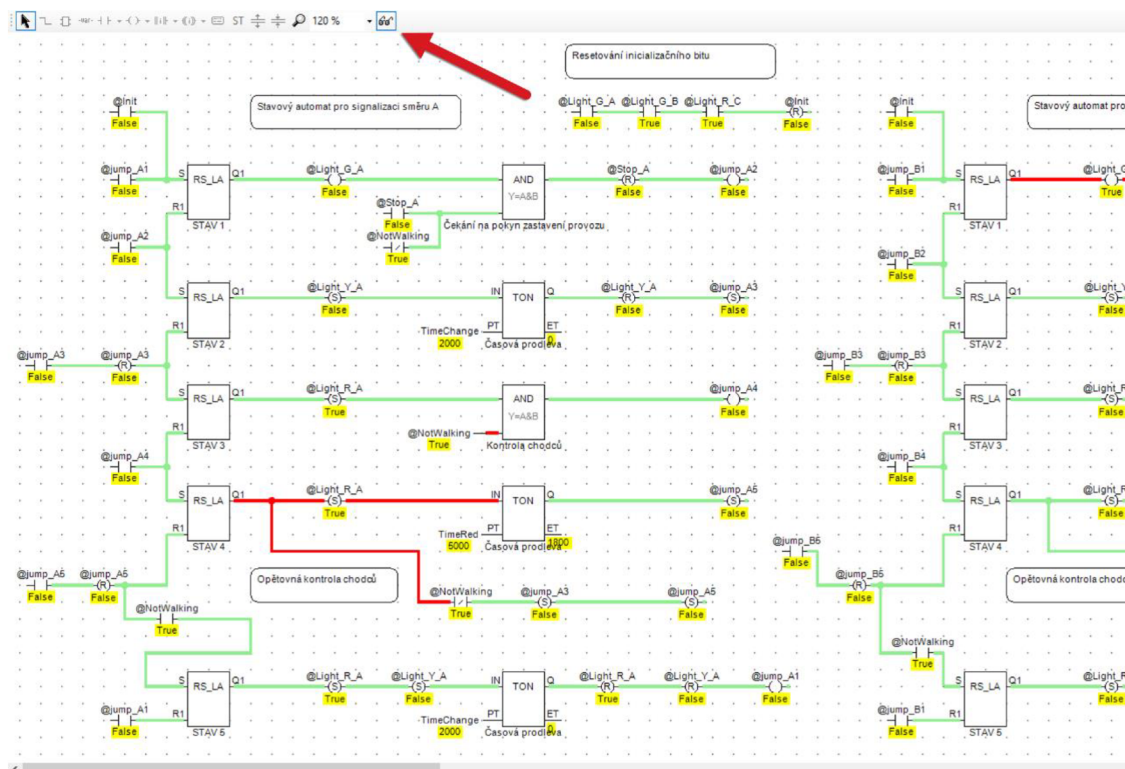
Obrázek 48: Předpokládané obsazení paměti stanice po nahrání aplikace

## 7.6 Ladění a testování

Ladit řídicí program lze simulačně pouze při tvorbě v novějším editoru EsiDet. To je ukázáno v předchozí kapitole při tvorbě obrazovky. V tomto konkrétním případě řízení SSZ je řídicí program psán ve starším PseDet editoru pro starší generaci. Bohužel je proto nutné řídicí program ladit a testovat s připojeným hardwarem k PC. Pro takový postup je třeba mít řídicí systém připojený způsobem popisovaným v kapitole *Zavádění programu do jednotky AMiRiS99* a mít program v jednotce zavedený. Při vzájemném propojení s AMR-OP83 lze ladit chování obou jednotek jako celku.

První možnost ladění je pomocí softwaru DetStudio a spočívá ve spuštění režimu, který v jazyce CFC graficky znázorňuje aktuální stav řídicího programu běžícího v připojené jednotce. V projektu pak lze nad konkrétním procesem spustit tlačítkem *Watch režim*, jak je uvedeno na obrázku 49. Režim ladění způsobí “oživení” procesu ve vývojovém prostředí, což se projeví zobrazováním aktuálních hodnot proměnných a prosvícením aktivních (pravdivých) logických cest červenou barvou (zeleně jsou označeny neaktivní signálové cesty). Lze sledovat i aktuální hodnoty časovačů, ačkoliv ne moc přesně, jelikož je vzájemná komunikace vývojového prostředí se zařízením pomalá.





Obrázek 49: Ladící režim (Watch režim) v jazyce CFC

Na obrázku 49 je vidět stavový automat signalizace směru A aktuálně se nacházející v stavu 4, který tvoří časovou prodlevu pro setrvání svítící červené signalizace. Dále je v tomto stavu například vidět běžící časovač, respektive jeho aktuální hodnota (1800 ms) na výstupu *ET*.

Druhá možnost ladění je využití nástroje *Inspektor*, alternativně jeho varianty jménem *AutoInspektor*. Oba nástroje lze najít na hlavní liště v záložce *Ladění*. *Inspektor* umožňuje sledovat vybrané proměnné, nebo do nich i zapisovat hodnoty dle jejich typu. Současně lze mít předdefinované až 4 inspektory současně. Tlačítkem zvýrazněným na obrázku 50 lze proměnné přidávat a definovat způsob jejich zobrazení. *AutoInspektor* je totožný nástroj, jen umožňuje rychlejší přidávání proměnných a je pouze jeden. Rychlé přidávání probíhá kl. zkratkou *Ctrl+F7* při současném označení požadovaných proměnných v procesu.

Výraz	Typ	WID	Hodnota
◆ Lights_b	Int	1000	00000000 01100100
◆ TimeWalk	Long	1005	7000
◆ TimeRed	Long	1003	5000
○ @Stop_A	Bit	1001	Ne
○ @Light_G_A	Bit	1000	Ano
○ @Button	Bit	1001	Ne
○ @NotWalking	Bit	1004	Ano

Obrázek 50: AutoInspektor s vybranými proměnnými

Pro testování programu (v ladícím režimu) bylo nutné využít fyzické zařízení, protože prostředí DetStudio neumožňuje simulovat běh programu. Na obrázku 51 je zobrazena testovací sestava, která se skládá z jednotek AMiRiS99 a AMR-OP83 propojených UTP kabelem za účelem zajištění jejich spojení pomocí RS485. Dále se zde nalézá zdroj 24 V pro napájení jednotek, převodník RS232 na USB pro propojení s PC a kabelové prodloužení RS232. Nakonec na obrázku figuruje UTP kabel s konektorem RJ45 pro připojení do síťové karty PC nebo síťového switchu.

Více fotek z testování, hardware a jeho zapojení přiloženo v přílohách 1 a 2.



Obrázek 51: Použitý hardware, fyzické ladění nahraných aplikací





## 8 ZÁVĚR

Uvedením řídicích systémů v zastoupení PLC bylo dosaženo základního pochopení principu jejich fungování. Následoval průzkum trhu, který se zaměřil výhradně na české výrobce. Tento průzkum vytvořil představu o schopnostech některých tuzemských výrobců. Jak bylo již zmíněno, výrobci své produkty předurčují pro konkrétní nasazení. Uskutečňují to jak hardwarovou konfigurací výrobků, tak softwarovou podporou knihoven v použitých vývojových prostředích. Z průzkumu je taky patrné, že některé systémy jsou schopné velmi vysoké spolehlivosti. Společnost Orbit Merret má svůj řídicí systém OMC8000 nasazen dokonce v jaderné energetice. Výrobce Unipi Technology směřuje své produkty zejména k řízení budov a použití jako SCADA systémy. Řídicí systémy aplikované v dopravě, energetice nebo vodohospodářství má společnosti TECO.

Na vytápění, řízení kotelen a výměníků se specializuje zase společnost AMiT, která nabízí větší počet různých kompaktních řešení. Z daleka se nejedná o všechny tuzemské výrobce, nicméně věnovat se všem výrobcům by bylo nad rozumný rozsah této práce.

Postup tvorby řídicího programu pro zařízení společnosti AMiT byl demonstrován na jednoduché (a pro edukativní účely vhodné) úloze řízení dopravní světelné signalizace. Hlavní proces řídicího programu byl napsán v grafickém jazyce CFC, který kombinuje prvky ladder diagramu, funkční bloky a navíc umí specifickým způsobem zakomponovat i jazyk ST přímo do grafického editoru. Popis postupu tvorby programu si kladl za cíl zejména základní seznámení čtenáře s vývojovým prostředím, a vysvětlena byla i specifika vývojového prostředí v kombinaci s použitými komponenty. Postup také zahrnuje tvorbu obrazovek pro řídicí systém s displejem, který může sloužit zároveň jako HMI panel. Nastavení vzájemné komunikace dvou generačně odlišných systémů využívajících odlišné editory bylo výzvou, která byla úspěšně zvládnuta. Řídicí systém AMiRiS99 obsahuje zmiňovaný hlavní proces, který řídí světelné signalizační zařízení. Toto zařízení může reagovat na povely k zastavení dopravy například z radarově měřené rychlosti, nebo tlačítka pro bezpečný přechod chodce. Navíc byly řešeny i potenciálně kolizní situace, které by mohly nastat v několika konkrétních přechodných stavech. Následně řídicí systém AMR-OP83 s obrazovkou je schopen zasahovat do probíhajícího řízení na systému AMiRiS99 a tvoří simulační a vizualizační panel. Vizualizuje se signalizace tří semaforů a simulaci vstupů představují taktéž tři tlačítka na dotykovém displeji. Aplikace je uzavřena řádným laděním a testováním na fyzickém hardware.

Práce může posloužit jako vstupní brána k aplikování zařízení společnosti AMiT, což bylo hlavní motivací a cílem práce. Všechny vytvořené programy jsou součástí elektronické přílohy.

Další navazující práce by mohla rozšířit průzkum trhu tuzemských výrobců, nebo je přímo srovnat s uznávanými zahraničními výrobci. Také by bylo možné vytvořit obdobný postup práce s řídicími systémy jiných českých výrobců.



## SEZNAM POUŽITÉ LITERATURY

- [1] ZEZULKA, František. *Prostředky průmyslové automatizace*. Brno: VUTIUM, 2004. ISBN 80-214-2610-1.
- [2] BENEŠ, Pavel. *Automatizace a automatizační technika: systémové pojetí automatizace*. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3628-7.
- [3] BENEŠ, Pavel. *Automatizace a automatizační technika: prostředky automatizační techniky*. 5., rozš. a aktualiz. vyd. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.
- [4] URBAN, Luboš. Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky. *Automa* [online]. 2005, 1 [vid. 2023-01-25]. Dostupné z: [https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005\\_02\\_30310\\_1237/](https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005_02_30310_1237/)
- [5] ČSN EN 61131-3. *Programovatelné řídicí jednotky – Část 3: Programovací jazyky*. Ed. 2. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2013, 238s. Třídící znak 18 7050.
- [6] Řídicí systémy. *AMiT Automation* [online]. [vid. 2023-03-05]. Dostupné z: <https://amitautomation.cz/produkt/ridici-systemy/>
- [7] Homepage. *AMiT Automation* [online]. [vid. 2023-02-26]. Dostupné z: <https://amitautomation.cz/>
- [8] ADiR: Katalogový list. In: *AMiT Automation* [online]. IV/2016 [vid. 2023-02-27]. Dostupné z: <https://amitautomation.cz/?file=3155>
- [9] PODOLÁK, S.; MLEJNEK, J.; NOVOTNÝ, M. ADiR: Návod na obsluhu. In: *AMiT Automation* [online]. 2006 ,rev. 21. 03. 2019 [vid. 2023-02-27]. Dostupné z: <https://amitautomation.cz/?file=3156> [dostupné po přihlášení]
- [10] AMiNi5D: Katalogový list. In: *AMiT Automation* [online]. X/2021 [vid. 2023-02-28]. Dostupné z: <https://amitautomation.cz/?file=13813>
- [11] PODOLÁK, S. AMiNi5D: Návod na obsluhu. In: *AMiT Automation* [online]. 2021, rev. 22. 02. 2022 [vid. 2023-02-28]. Dostupné z: <https://amitautomation.cz/?file=13814> [dostupné po přihlášení]
- [12] AMiRiS99W3: Katalogový list. In: *AMiT Automation* [online]. IV/2021 [vid. 2023-03-01]. Dostupné z: <https://amitautomation.cz/?file=5766>
- [13] ŘÍHA, Z.; BŘEZINA, J. AMiRiS99W3: Návod na obsluhu. In: *AMiT Automation* [online]. 2017, rev. 22. 04. 2021 [vid. 2023-03-01]. Dostupné z: <https://amitautomation.cz/?file=5768> [dostupné po přihlášení]
- [14] ACOS200: Katalogový list. In: *AMiT Automation* [online]. X/2017 [vid. 2023-03-01]. Dostupné z: <https://amitautomation.cz/?file=5376>
- [15] PODOLÁK, S.; MLEJNEK, J.; ŘÍHA, Z. ACOS200: Návod na obsluhu. In: *AMiT Automation* [online]. 2017, rev. 16. 10. 2017 [vid. 2023-03-01]. Dostupné z: <https://amitautomation.cz/?file=5377> [dostupné po přihlášení]

- [16] AD-CPUW2: Katalogový list. In: *AMiT Automation* [online]. V/2017 [vid. 2023-03-02]. Dostupné z: <https://amitautomation.cz/?file=3303>
- [17] PODOLÁK, S.; ŘÍHA, Z., MLEJNEK, J. AD-CPUW2: Návod na obsluhu. In: *AMiT Automation* [online]. 2011, rev. 10. 11. 2016 [vid. 2023-03-02]. Dostupné z: <https://amitautomation.cz/?file=3234> [dostupné po přihlášení]
- [18] ŘÍHA, Z., MLEJNEK, J. ADiS Modulární řídicí systém: Technická příručka. In: *AMiT Automation* [online]. 2012, rev. 10. 11. 2016 [vid. 2023-03-05]. Dostupné z: <https://amitautomation.cz/?file=3232> [dostupné po přihlášení]
- [19] MLEJNEK, J. APT1000G Průmyslový terminál: Návod na obsluhu. In: *AMiT Automation* [online]. 2010 [vid. 2023-03-05]. Dostupné z: <https://amitautomation.cz/?file=3572> [dostupné po přihlášení]
- [20] TAŤÁK, M. AMR-OP87/V Řídicí systém s grafickým displejem: Návod na obsluhu. In: *AMiT Automation* [online]. 2021, 04. 01. 2022 [vid. 2023-03-05]. Dostupné z: <https://amitautomation.cz/?file=13804> [dostupné po přihlášení]
- [21] *Unipi* [online]. [vid. 2023-03-06]. Dostupné z: <https://www.unipi.technology/cs/>
- [22] *Unipi Neuron [Unipi.technology Knowledge Base]* [online]. [vid. 2023-03-06]. Dostupné z: <https://kb.unipi.technology/cs:hw:02-neuron>
- [23] *Unipi Neuron: Manuál.* In: *Unipi* [online]. [vid. 2023-03-06]. Dostupné z: <https://www.unipi.technology/cs/obchod/product/download?fileId=2787>
- [24] *Unipi Neuron: Produktový list.* In: *Unipi* [online]. [vid. 2023-03-06]. Dostupné z: <https://www.unipi.technology/cs/obchod/product/download?fileId=3321>
- [25] *Unipi Patron [Unipi.technology Knowledge Base]* [online]. [vid. 2023-03-07]. Dostupné z: <https://kb.unipi.technology/cs:hw:007-patron>
- [26] *Unipi Patron: Manuál.* In: *Unipi* [online]. [vid. 2023-03-07]. Dostupné z: <https://www.unipi.technology/cs/obchod/product/download?fileId=3415>
- [27] *Unipi Patron: Produktový list.* In: *Unipi* [online]. [vid. 2023-03-07]. Dostupné z: <https://www.unipi.technology/cs/obchod/product/download?fileId=2715>
- [28] *Výrobce digitálních panelových měřicích přístrojů, PLC a převodníků | Orbit Merret* [online]. [vid. 2023-03-07]. Dostupné z: <https://www.merret.cz/>
- [29] OMC 8000: Manuál. In: *Orbit Merret* [online]. 2016 [vid. 2023-03-07]. Dostupné z: [https://www.merret.cz/sites/default/files/obsah/documents/0001CLx001/man01\\_OMC8000\\_2016.2.1\\_cs.pdf](https://www.merret.cz/sites/default/files/obsah/documents/0001CLx001/man01_OMC8000_2016.2.1_cs.pdf)
- [30] OMC 8000: Katalogový list. In: *Orbit Merret* [online]. 2014 [vid. 2023-03-07]. Dostupné z: [https://www.merret.cz/sites/default/files/obsah/documents/0001COt001/ds050\\_Katalog%20PLC\\_2014\\_cs.pdf](https://www.merret.cz/sites/default/files/obsah/documents/0001COt001/ds050_Katalog%20PLC_2014_cs.pdf)
- [31] *O společnosti Teco a.s. - TECO - Automatizace* [online]. [vid. 2023-04-07]. Dostupné z: <https://www.tecomat.cz/about-us/about-company/>

- [32] *TECO - Automatizace - Úvod* [online]. [vid. 2023-04-05]. Dostupné z: <https://www.tecomat.cz/>
- [33] **PŘEDSTAVENÍ SPOLEČNOSTI TECO a. s.:** *TECO - Automatizace* [online]. [vid. 2023-04-07]. Dostupné z: [https://www.tecomat.cz/uploads/files/DOCS/cze/PRINTS/Cat\\_2012\\_03\\_teco%20introduction\\_cz\\_01.pdf](https://www.tecomat.cz/uploads/files/DOCS/cze/PRINTS/Cat_2012_03_teco%20introduction_cz_01.pdf)
- [34] *CP-2007 - Foxtrot základní moduly / Teco Wiki* [online]. [vid. 2023-04-07]. Dostupné z: <https://wiki.tecomat.cz/clanek/cp-2007>
- [35] **PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT 2:** *Příručky HW - TECO - Automatizace* [online]. IV/2021 [vid. 2023-04-07]. Dostupné z: [https://www.tecomat.cz/download/get/txv00450\\_01/164/](https://www.tecomat.cz/download/get/txv00450_01/164/)
- [36] **PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-2007:** *Příručky HW - TECO - Automatizace* [online]. IV/2021 [vid. 2023-04-07]. Dostupné z: [https://www.tecomat.cz/download/get/txv00454\\_01\\_foxtrot2\\_cp-2007\\_u02\\_c/164/](https://www.tecomat.cz/download/get/txv00454_01_foxtrot2_cp-2007_u02_c/164/)
- [37] **PROGRAMOVATELNÉ AUTOMATY TECOMAT TC800:** *Příručky HW - TECO - Automatizace* [online]. III/2023 [vid. 2023-04-07]. Dostupné z: [https://www.tecomat.cz/download/get/txv00470\\_01\\_tc800\\_cp8001\\_cz/164/](https://www.tecomat.cz/download/get/txv00470_01_tc800_cp8001_cz/164/)
- [38] **Základní dokumentace IT-8601:** *IT-8601 - TC800 - Rozšiřující moduly vstupů a výstupů / Teco Catalog* [online]. [vid. 2023-04-07]. Dostupné z: <https://catalog.tecomat.cz/produkt/it-8601#download>



## SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1: Cyklus řešení uživatelského programu, převzato z [33].....	16
Obrázek 2: Funkce časového řízení úloh a přerušení, převzato z [1].....	17
Obrázek 3: Blokové schéma standardního modulárního PLC, převzato z [1].....	17
Obrázek 4: Rozměry produktu ADiR, převzato z [9] .....	21
Obrázek 5: Rozměry produktu AMiNi5D, převzato z [11].....	22
Obrázek 6: Rozměry produktu AMiRiS99W3, převzato z [13].....	23
Obrázek 7: Rozměry produktu AMiRiS99W3, převzato z [15].....	24
Obrázek 8: Rozměry produktu AD-CPUW2 z řady ADiS, převzato z [17] .....	25
Obrázek 9: Rozměry produktu APT1000G, převzato z [19].....	27
Obrázek 10: Rozměry produktu OP87/V, převzato z [20].....	28
Obrázek 11: Rozměry produktu Neuron L, převzato z [23] a upraveno .....	29
Obrázek 12: Náhled produktu Patron L527, převzato z [26] .....	31
Obrázek 13: Náhled produktu OMC 8000, převzato z [29].....	34
Obrázek 14: Náhled produktu Foxtrot 2 CP-2007, převzato z [34].....	36
Obrázek 15: Náhled produktu TC800 IT-8601, převzato z [38].....	37
Obrázek 16: Náhled produktu TC800 CP-8001, převzato z [37].....	37
Obrázek 17: Založení nového projektu .....	41
Obrázek 18: Dialog zakládání projektu.....	42
Obrázek 19: Projektová struktura aplikace pro AMiRiS99.....	42
Obrázek 20: Přidání nového procesu.....	44
Obrázek 21: Přidání nové proměnné .....	45
Obrázek 22: Vytvoření aliasů.....	46
Obrázek 23: Programovací prostředí v jazyce CFC .....	47
Obrázek 24: Prvky v nástrojové liště .....	47
Obrázek 25: Náhled na stav 1 stavového automatu a použití prvků .....	48
Obrázek 26: Náhled stavů 2 a 3, použití prvků "Set" a "Reset" pro ovládání proměnné.....	49
Obrázek 27: Náhled stavového automatu signalizace směru A, řešení skoků mezi stavy.....	50
Obrázek 28: Náhled stavového automatu řídicího signalizaci pro chodce .....	51
Obrázek 29: Provedení obvodu řešícího inicializaci SA.....	51
Obrázek 30: Použití GUI při programování v jazyce ST v procesu Proc_IO, použití modulu DigIn .....	52
Obrázek 31: Nastavení Modbus komunikace.....	53
Obrázek 32: Přidávání Holding registrů do Modbus komunikace .....	54
Obrázek 33: Nastavení komunikace jednotky AMiRiS99 pro zavedení aplikace .....	54
Obrázek 34: Konfigurace sériové linky pro komunikaci s PC.....	55
Obrázek 35: Nápopověda pro význam konfigurace DIP přepínačů.....	55
Obrázek 36: Dialogové okno pro zavádění aplikace.....	56



Obrázek 37: Projektová struktura aplikace pro AMR-OP83 .....	56
Obrázek 38: Seznam komunikačních objektů .....	57
Obrázek 39: Nastavení Modbus komunikace pro AMR-OP83 .....	58
Obrázek 40: Nastavené holding registry Modbus komunikace .....	58
Obrázek 41: Grafické vývojové prostředí obrazovek, příklad použitého pozadí ..	59
Obrázek 42: Umístění prvku BitSwitchView, nastavení jeho vzhledu a přiřazení proměnné .....	59
Obrázek 43: Výběr proměnné z Modbus holding registru .....	60
Obrázek 44: Použití prvku Button a jeho přizpůsobení .....	60
Obrázek 45: Použití skriptování, našeptávací funkce .....	61
Obrázek 46: Simulátor pro operační panely .....	61
Obrázek 47: Nastavení komunikace pro zavedení aplikace pomocí Ethernetu .....	62
Obrázek 48: Předpokládané obsazení paměti stanice po nahrání aplikace .....	63
Obrázek 49: Ladící režim (Watch režim) v jazyce CFC .....	64
Obrázek 50: AutoInspektor s vybranými proměnnými .....	64
Obrázek 51: Použitý hardware, fyzické ladění nahraných aplikací .....	65
Tabulka 1: Napěťové a proudové rozsahy AD převodu modulu IT-8601 [38] .....	38



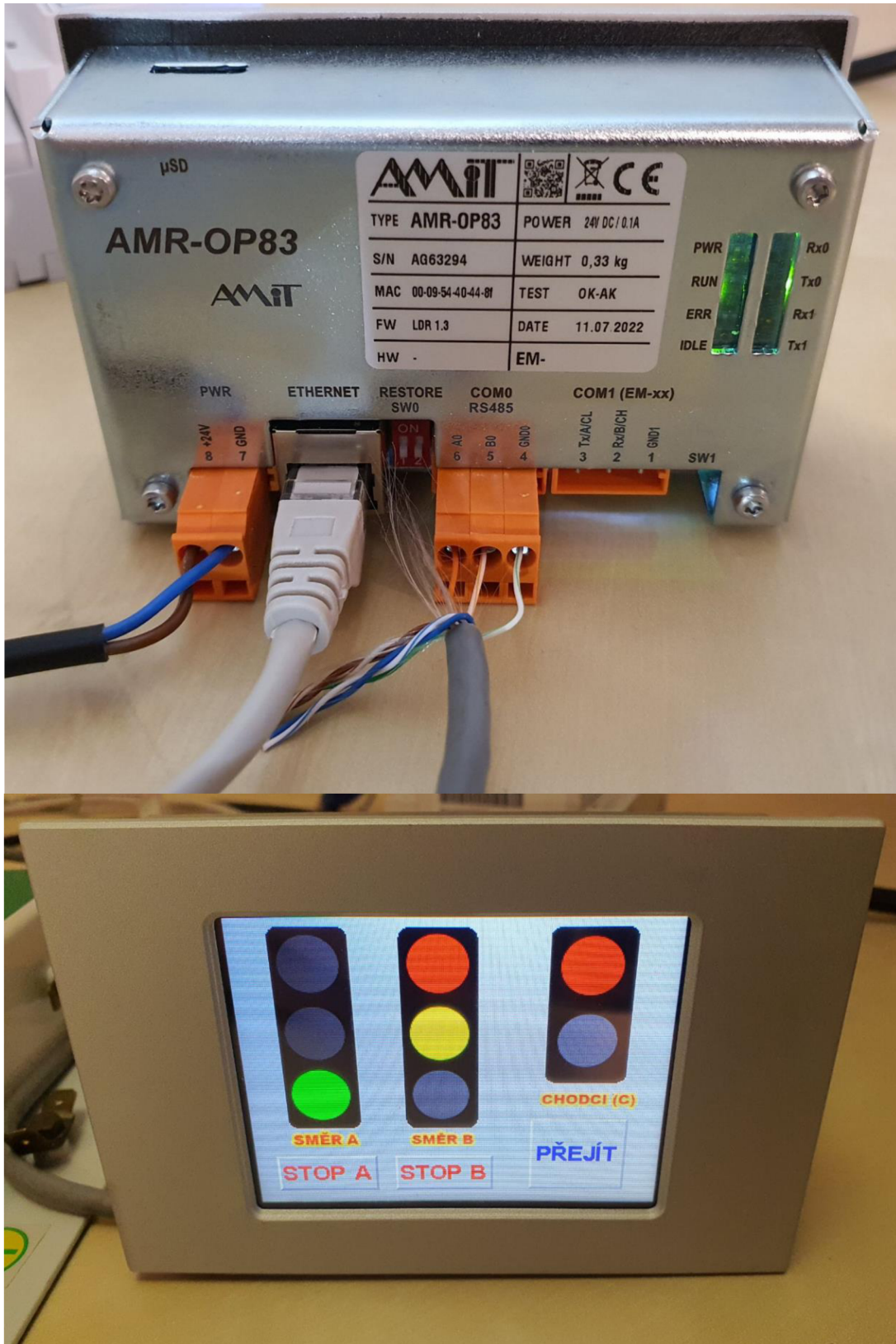
## SEZNAM ZKRATEK

AD	Analog to digital
CAN	Controller Area Network
CFC	Continuous Function Chart
CIB	Common Installation Bus
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central processing unit
DA	Digital to analog
EEPROM	Electrically Erasable Programmable Read-Only Memory
FDB	Function Block Diagram
GSM	Groupe Spécial Mobile
HSCI	HEIDENHAIN Serial Controller Interface
HVAC	Heating, ventilation, and air conditioning
I/O	Input/Output
IL	Instruction List
IP	Ingress Protection
LA	Logical Adress
LD	Ladder Diagram
LED	Light-Emiting Diode
LTE	Long-Term Evolution
OC	Open Collector
PC	Personal Computer
PCIe	Peripheral Component Interconnect Express
PLC	Programmable Logic Controller
PoE	Power over Ethernet
PWM	Pulse Width Modulation
RAM	Random Access Memory
RC	Resistor-Capacitor
SCADA	Supervisory Control And Data Acquisition
SD	Secure Digital
SFC	Sequential function chart
SPI	Serial Peripheral Interface
SSZ	Světelné Signalizační Zařízení
GUI	Graphical User Interface
ST	Structured Text
TFT	Thin Film Transistor
TTL	Transistor-Transistor Logic
WLAN	Wireless Local Area Network



# PŘÍLOHY

## Příloha 1: Testování vizualizace a ovládání simulace na AMR-OP83



**Příloha 2: Testování funkčnosti řízení AMiRis99 a komunikace s AMR-OP83**