

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**AKVIZIČNÍ A KONTROLNÍ JEDNOTKA
PROGRAMOVATELNÉHO AUTOMATU (SCADA)**

Bc. Michal Florián

Diplomová práce

2023

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Michal Florián**
Osobní číslo: **I21302**
Studijní program: **N0714A150005 Automatické řízení**
Téma práce: **Akviziční a kontrolní jednotka programovatelného automatu (SCADA)**
Zadávací katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cílem práce je návrh a realizace konstrukce akviziční a kontrolní jednotky programovatelného automatu. Úkolem jednotky bude sběr vybraných dat pro programovatelný automat. Data budou odesílány s využitím mobilní datové sítě, v protokolu MQTT na webový server. Konstrukčním základem jednotky bude jednočipový mikropočítač ESP32. Firmware mikropočítače bude realizován v jazyce C/C++. Pro komunikaci mezi programovatelným automatem a akviziční jednotkou bude použit vybraný typ průmyslového komunikačního protokolu (např. Modbus RTU). V teoretické části bude provedena rešerše na zadané téma se zaměřením na komerčně vyráběná zařízení určená k akvizici a ovládní programovatelného automatu na dálku pomocí webového serveru. Praktická část bude obsahovat popis konstrukce vlastního návrhu akviziční a kontrolní jednotky. Návrh mechanické konstrukce bude proveden ve formě 3D modelu v libovolném modelovacím prostředí. Nedílnou součástí práce bude výrobní dokumentace, včetně okomentovaných zdrojových kódů mikropočítače.

Rozsah pracovní zprávy: **60**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

ESP-WIFI-MESH <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/apiguides/mesh.html>
ESP-BLE-MESH <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/espble-mesh/ble-mesh-index.html>
802.11s https://www.ieee802.org/802_tutorials/06-November/802.11s_Tutorial_r5.pdf

Vedoucí diplomové práce: **Ing. Libor Havlíček, Ph.D.**
Katedra řízení procesů

Datum zadání diplomové práce: **8. listopadu 2022**
Termín odevzdání diplomové práce: **19. května 2023**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Daniel Honc, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 15. listopadu 2022

Prohlášení

Prohlašuji:

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

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

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

V Pardubicích dne 21. 8. 2023

Bc. Michal Florián

Poděkování

Děkuji svým rodičům, bratrovi, a zvláště pak své manželce za trpělivost a podporu při celé době mého navazujícího studia. Dále děkuji kolegům studentům a zejména pak Filipovi Bellingerovi. Také všem zaměstnancům školy, kteří se jakkoli podíleli na mé výuce, popř. na administrativě spojenou s mým studiem.

Mnohokrát děkuji panu Ing. Liboru Havlíčkovi, Ph.D. za pomoc při výběru tématu mé diplomové práce, za odbornou pomoc při návrhu akviziční a kontrolní jednotky a za nasměrování správným směrem při volbě řešení.

Velmi děkuji panu Ing. Martinovi Varechovi, MBA za umožnění prezenčního studia vysoké školy při zaměstnání.

A konečně děkuji svému kolegovi ze zaměstnání, Luďovi Suchánkovi, za cenné připomínky ohledně programování mikropočítače ESP32.

V Pardubicích dne 21. 8. 2023

Bc. Michal Florián

ANOTACE

Diplomová práce pojednává o návrhu akviziční a kontrolní jednotky, která obousměrně komunikuje s programovatelným automatem a rovněž obousměrně komunikuje se serverem pomocí MQTT zpráv. Za pomoci programovatelného automatu, HMI panelu a klientského zařízení pro vzdálené ovládání zmíněného programovatelného automatu je vytvořen SCADA systém, což je akronym pro vzdálené řízení a sběr dat.

KLÍČOVÁ SLOVA

PLC, mikropočítač ESP32, HMI panel, Node-RED, MQTT, Modbus-RTU, RS-485

TITLE

Acquisition and control unit of a programmable logic controller (SCADA)

ANNOTATION

The thesis deals with the design of an acquisition and control unit that communicates in both directions with a programmable logic controller and communicates in both directions with a server using MQTT messages. With the help of the programmable logic controller, HMI panel, Node-RED server and client device for remote control of the mentioned programmable logic controller is created a SCADA system, which is an acronym for supervisory control and data acquisition.

KEYWORDS

PLC, microcontroller ESP32, HMI panel, Node-RED, MQTT, Modbus-RTU, RS-485

OBSAH

SEZNAM ZKRATEK	9
SEZNAM ILUSTRACÍ	11
SEZNAM TABULEK	14
ÚVOD	15
1 TEORETICKÁ ČÁST	16
1.1 Elektromagnetické relé	16
1.1.1 Historie elektromagnetického relé	17
1.1.2 Popis funkce a konstrukce relé	18
1.1.3 Polovodičové relé	19
1.1.4 Reléové systémy	21
1.2 Programovatelné logické automaty	23
1.2.1 Definice programovatelného logického automatu	23
1.2.2 Historie PLC	23
1.2.3 Dělení PLC	25
1.2.4 Srovnání konkrétních typů PLC	31
1.2.5 Programování PLC dle normy IEC 61131-3	33
1.2.6 Komunikační protokoly	38
1.3 Mikropočítač ESP32-WROOVER-IE	55
1.4 Komerčně vyráběné akviziční a kontrolní jednotky	61
1.4.1 Siemens Simatic WinCC	63
1.4.2 Beckhoff TwinCAT 3	63
2 PRAKTICKÁ ČÁST	65
2.1 Návrh a realizace akviziční a kontrolní jednotky	65
2.1.1 Popis funkce realizovaného SCADA systému	65
2.2 Konfigurace PLC Beckhoff CX8190	66
2.2.1 Softwarová konfigurace PLC Beckhoff CX8190	66
2.2.2 Program PLC Beckhoff CX8190	75
2.3 HMI panel Weintek	77
2.4 Akviziční a kontrolní jednotka	80
2.4.1 Napájecí obvody	81
2.4.2 Obvod pro komunikační rozhraní RS-485	82

2.4.3	Externí moduly	82
2.4.4	Celkové schéma jednotky	84
2.4.5	Programování akviziční a kontrolní jednotky	85
2.5	Vzdálený Server diskstation	85
2.5.1	Node-RED	86
2.5.2	Mosquitto Broker	87
2.6	Klientské zařízení SCADA systému	88
3	TESTOVÁNÍ AKVIZIČNÍ A KONTROLNÍ JEDNOTKY PROGRAMOVATELNÉHO AUTOMATU (SCADA)	90
3.1	Deska plošného spoje akviziční a kontrolní jednotky	90
3.2	Lokální ovládaní z HMI panelu	91
3.3	Vzdálené ovládaní z klientské aplikace webového prohlížeče	91
3.4	Závěr z testování SCADA systému	92
	ZÁVĚR	94
	POUŽITÁ LITERATURA	95
	PŘÍLOHY	99

SEZNAM ZKRATEK

ADS	Automation Device Specification
ADU	Application Data Unit
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
CSS	Cascading Style Sheets
DC	Direct Current
DIN	Deutsches Institut für Normung
DIP	Dual In-line Package
DUT	Data Unit Type
EEPROM	Electrically Erasable Programmable Read-Only Memory
FBD	Function Block Diagram
FMMU	Fieldbus Memory Management Unit
GPIO	General-Purpose Input/Output
GPRS	General Packet Radio Service
SSH	Secure Shell
GUI	Graphical User Interface
GVL	Global Variable List
HMI	Human Machine Interface
HTML	HyperText Markup Language
I/O	Input/Output
IL	Instruction List
IP	Internet Protocol
ISO	International Organization for Standardization
LD	Ladder Diagram
LED	Light Emitting Diode
LLC	Logical Link Control
MAC	Media Access Control
MODICON	Modular Digital Controlle
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MQTT	Message Queuing Telemetry Transport

NIC	Network Interface Card
OSI	Open Systems Interconnection
OTP	One-Time Password
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
POU	Program Organization Unit
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read-Only Memory
RSA	Rivest–Shamir–Adleman
RTC	Real-Time Clock
RTU	Remote Terminal Unit
RX	Receive
SCADA	Supervisory Control and Data Acquisition
SFC	Sequential Function Chart
SHA-2	Secure Hash Algorithm 2
SQL	Structured Query Language
SRAM	Static Random Access Memory
SSR	Solid State Relay
ST	Structured Text
STP	Shielded Twisted Pair
TC	TwinCAT
TX	Transmit
ULP	Ultra Low Power
UNIX	Operační systém
URL	Uniform Resource Locator
USART	Universal Synchronous / Asynchronous Receiver and Transmitter
USB	Universal serial bus
UTP	Unshielded Twisted Pair
WD	Watchdog
WYSIWYG	What You See Is What You Get

SEZNAM ILUSTRACÍ

Obrázek 1.1 – Elektromagnetické relé v patici na DIN lištu (Papouch, 2023)	16
Obrázek 1.2 – Polovodičové relé (SSR) (CNCSshop, 2010)	16
Obrázek 1.3 – Morseův telegraf z roku 1837 (Electrical telegraph, 2023)	17
Obrázek 1.4 – Elektromagnetické relé v klidovém stavu (Elektromagnetické relé, 2023)	18
Obrázek 1.5 – Schématická značka relé (Elektromagnetické relé, 2023)	19
Obrázek 1.6 – Elektromagnetické relé sepnutém, resp. rozepnutém stavu (Elektromagnetické relé, 2023)	19
Obrázek 1.7 – Schématický řez MOSFET tranzistorem (MOSFET, 2023)	20
Obrázek 1.8 – Polovodičové relé s MOSFET tranzistorem (Elliot, 2013)	20
Obrázek 1.9 – Logická funkce NOR realizována pomocí relé (Hades, nedatováno)	21
Obrázek 1.10 – Logická funkce NAND realizována pomocí relé (Hades, nedatováno)	21
Obrázek 1.11 – Reléový systém (Relay, 2023)	22
Obrázek 1.12 – MODICON 084 (Schneider Electric, 2019)	24
Obrázek 1.13 – Konfigurace MODICONu 084	25
Obrázek 1.14 – Obecné blokové schéma programovatelného automatu	25
Obrázek 1.15 – Blokové schéma prvního cyklu programu PLC	26
Obrázek 1.16 – Kompaktní PLC Eaton EC4P (Eaton, nedatováno)	27
Obrázek 1.17 – Kompaktní i modulární PLC Simatic řady S7-1200 (RS-ONLINE, nedatováno)	28
Obrázek 1.18 – Mikro PLC Allen-Bradley Micro810 (Rockwell Automation, nedatováno) ..	29
Obrázek 1.19 – Modulární PLC Beckhoff CX8190 (Beckhoff, 2023)	30
Obrázek 1.20 – Rozvaděč vozu na štěrku MFS40	31
Obrázek 1.21 – Zásobníkový vůz na štěrku MFS 40 (Zutwargowiec, nedatováno)	31
Obrázek 1.22 – Realizace pomocí jazyka LD	33
Obrázek 1.23 – Realizace pomocí jazyka FBD	34
Obrázek 1.24 – Realizace pomocí jazyka ST	35
Obrázek 1.25 – Realizace pomocí jazyka SFC	35
Obrázek 1.26 – Realizace pomocí jazyka IL	36
Obrázek 1.27 – Programovací prostředí CODESYS	37
Obrázek 1.28 – Programovací prostředí TwinCAT 3	37
Obrázek 1.29 – Model ISO/OSI (Počítačové sítě, 2010)	39
Obrázek 1.30 – Příklady implementace (Ronešová, 2005)	44
Obrázek 1.31 – Základní tvar Modbus telegramu (Ronešová, 2005)	44
Obrázek 1.32 – Korektní transakce požadavku a odpovědi (Ronešová, 2005)	45
Obrázek 1.33 – Transakce požadavku a odpovědi s chybou (Ronešová, 2005)	45
Obrázek 1.34 – Datový model s oddělenými bloky (Ronešová, 2005)	46
Obrázek 1.35 – Datový model s jediným blokem (Ronešová, 2005)	47
Obrázek 1.36 – Příklad adresování dle pravidel Modbus (Ronešová, 2005)	47
Obrázek 1.37 – Vývojový diagram zpracování Modbus požadavku na straně serveru (Ronešová, 2005)	48
Obrázek 1.38 – Tvar zprávy Modbus RTU a Modbus ASCII (Ronešová, 2005)	51
Obrázek 1.39 – Struktura rámce Modbus RTU (Ronešová, 2005)	51
Obrázek 1.40 – Dynamické zpracování telegramu (EtherCAT, nedatováno)	53

Obrázek 1.41 – Decentralizace pomocí EtherCAT sběrnice (EtherCAT, nedatováno)	54
Obrázek 1.42 – Funkční blokové schéma mikropočítače ESP32 (Espressif Systems, 2023)	55
Obrázek 1.43 – ESP32-WROOVER-IE	59
Obrázek 1.44 – Vývody ESP32-WROOVER-IE (Espressif Systems, 2023)	59
Obrázek 1.45 – Architektura SCADA systému (Ecava Intergraxor, 2020)	62
Obrázek 1.46 – WinCC SCADA systém firmy Siemens (Mesidas, nedatováno)	63
Obrázek 1.47 – Nástroj Visualisation v prostředí TwinCAT 3	64
Obrázek 2.1 – SCADA systém	65
Obrázek 2.2 – HW konfigurace včetně signalizace konfiguračního režimu	67
Obrázek 2.3 – Založení nového projektu s názvem DP_MODBUS_RTU.....	67
Obrázek 2.4 – Přidání instance projektu PLC pomocí pravého tlačítka myši	68
Obrázek 2.5 – Vytvoření standartního PLC projektu	68
Obrázek 2.6 – Strom projektu PLC	69
Obrázek 2.7 – Před rozkliknutím menu volby PLC	69
Obrázek 2.8 – Po rozkliknutím menu volby PLC	69
Obrázek 2.9 – Prohledání dostupných PLC na síti Ethernet	70
Obrázek 2.10 – Potvrzení přidání cesty k PLC	70
Obrázek 2.11 – Znázornění tlačítka <i>Scan</i> , pro prohledání připojených terminálů	71
Obrázek 2.12 – Nalezené terminály připojen k PLC	71
Obrázek 2.13 – První krok aktivace nastavení PLC	72
Obrázek 2.14 – Nastavení statické IP adresy PLC	72
Obrázek 2.15 – Aktivace nastavení pro vzdálené připojení programem CERHOST.EXE	73
Obrázek 2.16 – Přihlášení do nastavení PLC přes aplikaci CERHOST.EXE	73
Obrázek 2.17 – Nastavení síťového switchu PLC	74
Obrázek 2.18 – Licence projektu v programu TwinCAT 3	74
Obrázek 2.19 – Strom projektu s vytvořenými složkami a položkami	75
Obrázek 2.20 – Program obsluhující Modbus RTU a RS-485	76
Obrázek 2.21 – Spojení proměnné programu a fyzického vstupu PLC	77
Obrázek 2.22 – Výběr správného nastavení pro komunikaci s Beckhoff PLC	78
Obrázek 2.23 – Nastavení IP a NetID v HMI panelu	78
Obrázek 2.24 – NetID PLC Beckhoff CX8190	79
Obrázek 2.25 – Hotové GUI HMI panelu	79
Obrázek 2.26 – Blokové schéma akvizitní a kontrolní jednotky	80
Obrázek 2.27 – Schéma obvodu mikropočítače ESP32-WROOVER-IE	80
Obrázek 2.28 – Schéma obvodu převodníku napětí 24 V DC na 3,3 V DC	81
Obrázek 2.29 – Schéma obvodu převodníku napětí 3,3 V DC na 4 V DC	81
Obrázek 2.30 – Schéma zapojení obsluhující komunikaci RS-485	82
Obrázek 2.31 – Externí modul SIM800L se zapojenou anténou	83
Obrázek 2.32 – Externí programovací modul USB-UART	83
Obrázek 2.33 – Celkové schéma akvizitní a kontrolní jednotky	84
Obrázek 2.34 – Návrh desky plošného spoje	84
Obrázek 2.35 – Webové zobrazení pro interakci se serverem	85
Obrázek 2.36 – Ilustrační obrázek Synology DiskStation 218+	86
Obrázek 2.37 – <i>Flow</i> programu na Node-RED	86
Obrázek 2.38 – Povolení služby SSH	88
Obrázek 2.39 – Klientské zobrazení v prohlížeči stolního počítače	88
Obrázek 2.40 – Klientské zobrazení v prohlížeči mobilního telefonu	89

Obrázek 3.1 – Prototyp zapojené jednotky do SCADA systému	90
Obrázek 3.2 – HMI panel Weintek	91
Obrázek 3.3 – Klientské zařízení ve formě mobilního telefonu pro vzdálené ovládání.....	92
Obrázek 3.4 – Vytvořený SCADA systém	93

SEZNAM TABULEK

Tabulka 1.1 – Srovnání konkrétních typů PLC (Beckhoff, 2023; Rockwell Automation, nedatováno; Eaton, nedatováno)	32
Tabulka 1.2 – Přehled vybraných komunikačních protokolů	38
Tabulka 1.3 – Stručný přehled funkcí jednotlivých vrstev (IJS, nedatováno)	43
Tabulka 1.4 – Datový model Modbus (Ronešová, 2005)	46
Tabulka 1.5 – Chybové kódy Modbus (Ronešová, 2005)	49
Tabulka 1.6 – Adresy a jejich cíle v síti Modbus (Ronešová, 2005)	50
Tabulka 1.7 – Popis napájecích režimů ESP32-WROOVER-IE	58
Tabulka 1.8 – Výpis vývodů ESP32-WROOVER-IE (Espressif Systems, 2023)	60
Tabulka 1.9 – Označení podtypů vývodů ESP32-WROOVER-IE (Espressif Systems, 2023)	61
Tabulka 2.1 – Nalezené rozšiřující terminály	72

ÚVOD

Programovatelné logické automaty (PLC) mají v průmyslu klíčovou roli již více než půl století. V průběhu více než půl století se u PLC vyvinuly mnohé aspekty. Největší zlepšení, u PLC za posledních 50 let a více spatřujeme především v oblasti rychlosti zpracování dat, možnostech komunikace s dalšími systémy, robustnosti a v jejich programovatelnosti a flexibilitě. Díky těmto inovacím mohou PLC lépe monitorovat a analyzovat provozní data, což zvyšuje efektivitu a spolehlivost průmyslových systémů. Nyní můžeme tvrdit, že PLC ovládají téměř všechny průmyslové procesy po celém světě. Troufám si říct, že pokud vzdáleně dokážeme ovládat PLC, znamená to, že vzdáleně ovládáme technologické procesy na celém světě.

Hlavní motivací pro návrh a realizaci akviziční a kontrolní jednotky, tak aby zapadala do vytvořeného SCADA systému, bylo mé zaměstnání ve společnosti HROCHOSTROJ a.s., kde se dnes a denně setkávám s PLC. Žádné z těchto PLC, zatím, nedisponuje možností sběru dat ze železničních pracovních strojů, které jsou zmíněnými PLC ovládány. V rámci provozuschopnosti těchto strojů se může zdát výhodné sledovat jejich parametry, které mohou přispět k přesnější identifikaci závad anebo k prediktivní údržbě, která bude hrát hlavní roli ve snížení nákladů na provoz těchto železničních pracovních strojů v budoucnosti.

Akviziční a kontrolní jednotky programovatelných automatů, které se dají připojit k PLC, nejsou na trhu vůbec k dispozici. Pokud chceme využít SCADA systém, často musíme využít již hotový, robustní a drahý systém, což může být výhoda, pokud navrhujeme nový řídicí systém, ale v případě, že chceme zlepšit stávající systém, není takové zlepšení možné. Akviziční a kontrolní jednotka, kterou se zabývá tato práce, pomáhá zmíněný problém vyřešit, neboť jednotka dovede komunikovat s jakýmkoli PLC, které podporuje komunikační protokol Modbus-RTU, což v dnešní době dovede drtivá většina PLC.

1 TEORETICKÁ ČÁST

1.1 ELEKTROMAGNETICKÉ RELÉ

Relé je elektricky ovládaný spínač, který se skládá z kontaktů pro jeden nebo více řízených signálů a z ovládacích kontaktů. Tento spínač může mít libovolný počet kontaktů pro řízené signály ve více formách, jako jsou rozpínací a spínací kontakty nebo jejich kombinace. Ovládacími kontakty se rozumí kontakty cívky. Tradiční elektromagnetická forma relé využívá elektromagnet k sepnutí nebo rozepnutí kontaktů. Běžně používané relé i s patičí na DIN lištu je vyobrazeno na obrázku 1.1. (Černý, 1968; Relay, 2023).



Obrázek 1.1 – Elektromagnetické relé v patičí na DIN lištu (Papouch, 2023)

Později byla vynalezena i relé využívající jiné principy činnosti, například polovodičová relé (SSR), která k ovládní využívají vlastnosti polovodičů, aniž by se spoléhala na pohyblivé části, polovodičové relé je znázorněné na obrázku 1.2. (Omron, nedatováno).



Obrázek 1.2 – Polovodičové relé (SSR) (CNCShop, 2010)

Relé s kalibrovanými provozními charakteristikami a někdy i s více provozními cívkami se používají k ochraně elektrických obvodů před přetížením nebo poruchami. V moderních

elektroenergetických systémech tyto funkce plní digitální přístroje, které se stále nazývají ochranná relé. Blokovací relé (impulzní relé nebo také pomocné relé s přídrží) vyžadují k trvalému ovládní spínače pouze jeden impuls ovládacího proudu. Další impuls přivedený na druhou sadu ovládacích svorek nebo impuls s opačnou polaritou spínač resetuje, zatímco opakované impulsy stejného druhu nemají žádné účinky. Magnetická blokovací relé jsou užitečná v aplikacích, kdy by přerušené napájení nemělo mít vliv na obvody, které relé ovládá. Relé se používají tam, kde je třeba ovládat obvod nezávislým signálem s malým výkonem, nebo tam, kde je třeba ovládat několik obvodů jedním signálem. (Omron, nedatováno).

1.1.1 Historie elektromagnetického relé

V roce 1809 navrhl Samuel Thomas von Sömmerring elektrolytické relé jako součást svého elektrochemického telegrafu. Výhradně elektrická relé začala vznikat jako další vylepšení telegrafů, a to díky americkému vědci Josephu Henrymu, který je často uváděn jako vynálezce relé v roce 1835, aby vylepšil svou verzi elektrického telegrafu vyvinutou dříve v roce 1831. Oficiální patent však byl vydán až v roce 1840 Samuelu Morsemu na jeho telegraf. Morseův telegraf z roku 1837 je znázorněn na obrázku 1.3. Popsaný reléový mechanismus fungoval jako digitální zesilovač, který opakoval telegrafní signál, a umožňoval tak šíření signálů na libovolnou vzdálenost. Slovo relé se v souvislosti s elektromagnetickými operacemi objevuje od roku 1860. Byla to vůbec první forma automatizace, za použití relé, která nahradila operátorky v telefonních ústřednách, které ručně přepojovaly telegrafy, aby se dostaly na určené místo. (Černý, 1968; Relay, 2023; Littman, nedatováno).

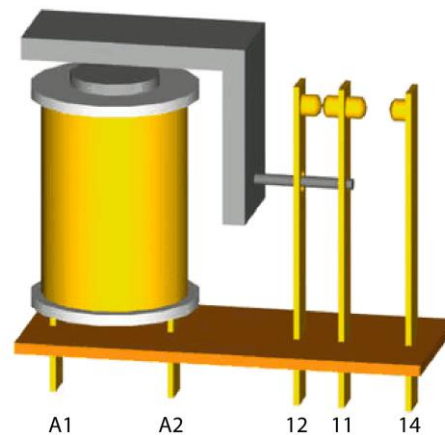


Obrázek 1.3 – Morseův telegraf z roku 1837 (Electrical telegraph, 2023)

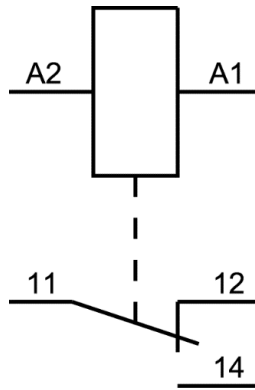
1.1.2 Popis funkce a konstrukce relé

Jednoduché elektromagnetické relé se skládá z cívky vodiče ovinuté kolem jádra z magneticky měkkého materiálu (solenoid), pevného kontaktu z magneticky měkkého materiálu, který zajišťuje nízkou reluktanci magnetického toku, pohyblivé železné kotvy a jedné nebo více sad kontaktů. Kotva je držena na místě pružinou, přičemž když je relé bez napětí, vzniká v elektrickém obvodu vzduchová mezera a relé není aktivní. (Černý, 1968)

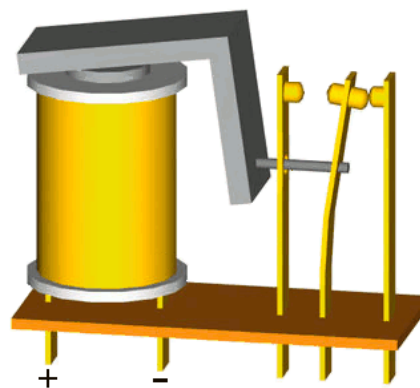
Proud procházející cívkou generuje magnetické pole, které přitáhne kotvu, a následný pohyb kontaktu kotvy buď vytvoří, nebo přeruší (v závislosti na konstrukci) spojení s pevným kontaktem. Pokud byly kontakty sepnuté, když bylo relé bez napětí, jedná se o rozpínací kontakty, pokud tomu bylo naopak, jedná se o spínací kontakty relé. Po vypnutí proudu do cívky se kotva vrátí silou, která je přibližně o polovinu menší než magnetická síla, do uvolněné polohy. Obvykle tuto sílu zajišťuje pružina nebo napružené kontakty samotné kotvy. Na obrázku 1.4 je znázorněná vnitřní konstrukce relé včetně popsaných kontaktů a tato ilustrace koresponduje se schématickou značkou, která je ilustrována na obrázku 1.5. Kontakty s označení A1 a A2 jsou kontakty cívky, kontakt 11 je ovládán pohyblivou kotvou. Po přivedení signálu na kontakty cívky se kontakt 11 rozpíná s kontaktem 12 a zároveň spíná s kontaktem 14. Na obrázku 1.6 je potom znázorněn stav kontaktů po přivedení signálu z původního obrázku 1.4. (Černý, 1968; Elektromagnetické relé, 2023; Relay, 2023).



Obrázek 1.4 – Elektromagnetické relé v klidovém stavu (Elektromagnetické relé, 2023)



Obrázek 1.5 – Schématická značka relé (Elektromagnetické relé, 2023)



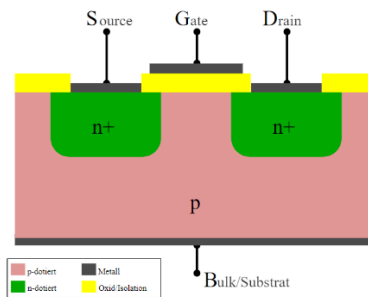
Obrázek 1.6 – Elektromagnetické relé sepnutém, resp. rozepnutém stavu (Elektromagnetické relé, 2023)

1.1.3 Polovodičové relé

Vzhledem k masivnímu přecházení na polovodičové prvky a součástky, což je zejména díky tomu, že nemají pohyblivé části jako například elektromagnetické relé a tím pádem nedochází k mechanickému opotřebení, je potřeba zmínit i polovodičové relé. Tato polovodičová relé principiálně fungují stejně jako elektromagnetická relé, tedy přivedením nízko výkonného ovládacího signálu se sepnou kontakty řízeného, zpravidla silového signálu. (Krejčířík, 2006; Crydom, 2009; Elliot, 2013;).

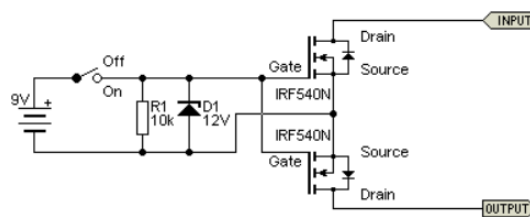
Polovodičové relé (SSR) s výkonovým polovodičovým prvkem např. MOSFETem je řízeno na základě jeho elektrických vlastností. MOSFET (Metal Oxide Semiconductor Field Effect Transistor) je typ tranzistoru, který otevírá nebo zavírá průchod proudu mezi emitorem (source) a kolektorem (drain) na základě změny napětí na bráně (gate) tohoto tranzistoru.

Schématický řez MOSFETem je znázorněn na obrázku 1.7. (Krejčířík, 2006; Crydom, 2009; Elliot, 2013;).



Obrázek 1.7 – Schématický řez MOSFET tranzistorem (MOSFET, 2023)

V konstrukci polovodičového relé jsou implementovány dva hlavní prvky – optočlen a samotný výkonový tranzistor. Využití optočlenu je dáno potřebou pro galvanické oddělení vstupní a výstupní části relé a pro přenos řídicího signálu. Přepínání výstupního obvodu je zajišťováno výkonovým tranzistorem. Když je signál přiveden na vstup optočlenu, LED v optočlenu je rozsvícena a ovlivňuje fototranzistor, který je aktivován. Tímto procesem je měněno napětí na bráně výkonového tranzistoru. Pokud dojde ke zvýšení napětí na bráně výkonového tranzistoru, je umožněn průchod proudu mezi kolektorem a emitorem. Pokud je naopak napětí na bráně sníženo, proud neprochází a výkonový tranzistor nevede. (Krejčířík, 2006; Crydom, 2009; Elliot, 2013;).



Obrázek 1.8 – Polovodičové relé s MOSFET tranzistorem (Elliot, 2013)

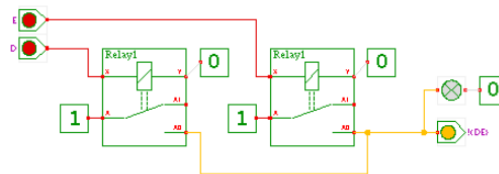
Polovodičová relé nemají žádné mechanické části, které by se mohly opotřebovat, což přispívá k jejich delší životnosti a spolehlivosti. Další výhodou polovodičového relé je rychlá reakční doba výkonových tranzistorů, a proto je možné toto relé zapnout a vypnout velmi rychle. Nicméně při vysokých provozních proudech je generováno teplo, které je třeba odvádět chladičem. Polovodičových relé je mnoho typů. Dělí se například použitím polovodičového spínacího prvku. Dále lze polovodičové relé rozdělit dle ovládaného proudu, zdali má tento

proud charakter stejnosměrný nebo střídavý. Na obrázku 1.8 je znázorněné schéma vnitřního zapojení polovodičového relé za použití dvou MOSFET tranzistorů. (Krejčířík, 2006; Crydom, 2009; Elliot, 2013;).

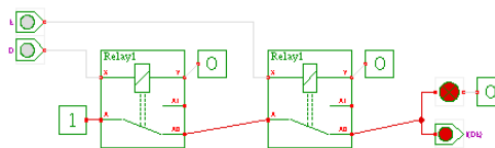
V dnešních PLC se u silových digitálních výstupních signálech využívají elektromagnetická relé i polovodičová relé, a to přímo v konstrukci kompaktního PLC nebo jako rozšiřující modulu v případě modulárního PLC systému. O typech PLC se bude dále zabývat kapitola 1.2.3.

1.1.4 Reléové systémy

V předcházejících odstavcích bylo popsáno elektromagnetické relé a polovodičové relé, které jsou základním kamenem v řízení akčních členů a tím pádem mají své místo v automatizaci průmyslových procesů. Reléové systémy, tedy mnoho relé zapojených do obvodu, který plní účel a má svoji vnitřní logiku, byla první forma automatizace za použití elektronických zařízení. Takový reléový systém je ilustrován na obrázku 1.11. Aby se dal účinně řídit proces, je potřeba mít možnost tyto relé zapojovat do logických funkcí. Reléová logika je tedy založena na použití elektromagnetických relé k vytvoření a řízení logických funkcí. Reléová logika využívá vlastností relé, které umožňují přepínání elektrických obvodů pomocí nízko výkonných signálů. Reléová logika se skládá z několika relé, která jsou propojena tak, aby tvořila hradla základních logických funkcí jako jsou AND, OR, NOT nebo hradel univerzálních logických funkcí NAND, NOR. Na obrázku 1.9 je znázorněné jakým způsobem se pomocí relé dá realizovat logická funkce NOR. Na obrázku 1.10 je ilustrované zapojení pomocí logické funkce NAND. (Černý, 1968; Relay, 2023).

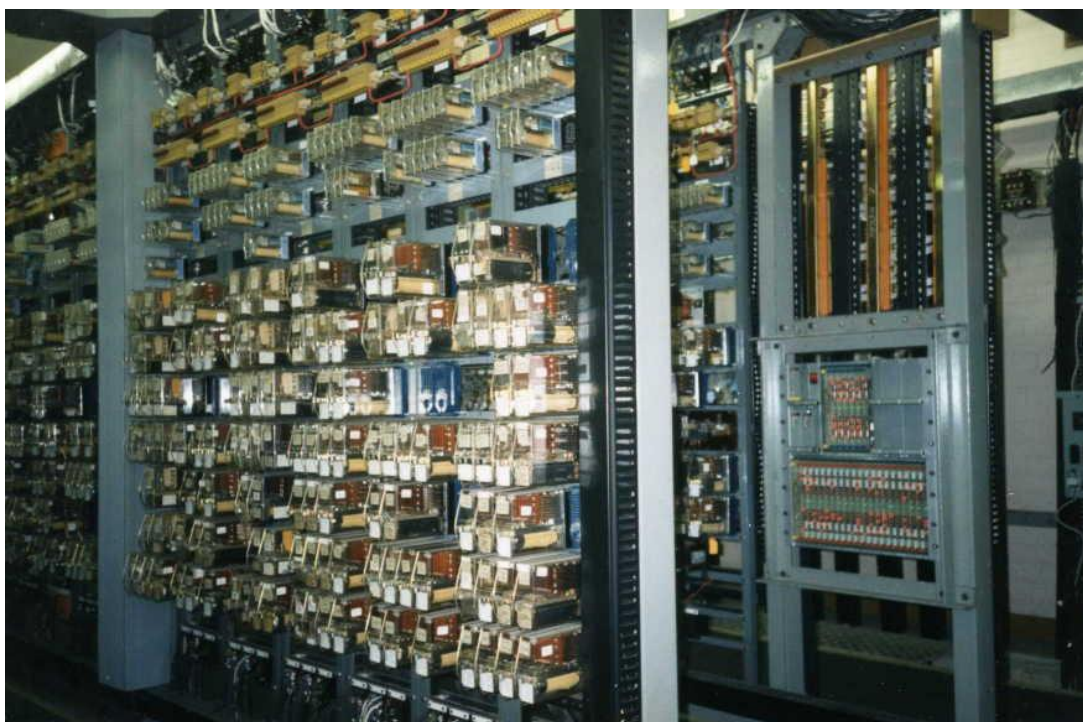


Obrázek 1.9 – Logická funkce NOR realizována pomocí relé (Hades, nedatováno)



Obrázek 1.10 – Logická funkce NAND realizována pomocí relé (Hades, nedatováno)

V automatizaci procesů byla reléová logika využívána zejména pro sekvenční řízení, kdy bylo možné nastavit aktivaci jednoho relé na základě stavu jiného, což umožňovalo provádění kroků v určeném pořadí. Procesy a úkoly mohly být vybírány a upřednostňovány reléovou logikou, kde relé s různými prioritami mohly řídit, který proces měl být v případě konfliktu nebo přetížení proveden. Monitorování a ochrana technologických procesů a strojů byly také možné díky reléové logice, kde byla přidána bezpečnostní relé do obvodu pro sledování stavu stroje nebo procesu a pro automatické vypnutí nebo blokování při výskytu problému nebo nebezpečí. Reléová logika, i když byla dříve široce používána pro řízení průmyslových procesů, byla postupně nahrazena modernějšími technologiemi jako jsou programovatelné logické automaty kvůli jejich větší flexibilitě, snadnější údržbě a lepší integraci s ostatními systémy. Nevýhody reléové logiky zahrnují složitost a velikost obvodů, omezenou flexibilitu, nižší spolehlivost kvůli omezené životnosti elektromagnetických relé a potřebu pravidelné údržby. S příchodem programovatelných logických automatů a dalších digitálních řídicích systémů se reléová logika stala méně běžnou v průmyslových aplikacích. (Černý, 1968; Relay, 2023).



Obrázek 1.11 – Reléový systém (Relay, 2023)

Programovatelné logické automaty nabízí větší flexibilitu, snadnější údržbu, vyšší spolehlivost a lepší integraci s ostatními systémy, což umožňuje efektivnější řízení

technologických procesů a strojů. I přes vše výše uvedené se elektromagnetické relé stále hojně v automatizaci využívá, nicméně dnes zejména pro řízení vysokovýkonných signálů. V další části teoretické části bude podrobně popsán programovatelný logický automat.

1.2 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

1.2.1 Definice programovatelného logického automatu

Programovatelný logický automat, z anglického originálu Programmable Logic Controller (dále jen PLC), je průmyslový počítač, který se používá k řízení a automatizaci zejména technologických průmyslových procesů, strojů a systémů. PLC umožňují uživatelům vytvářet programy pro řízení a monitorování různých procesů, čímž zlepšují efektivitu, spolehlivost a bezpečnost výroby. PLC je tedy průmyslový počítač s upraveným operačním systémem i specifickým hardwarem přizpůsobený pro běh v reálném čase a s co nejkratší dobou odezvy. Tento průmyslový počítač je navržen tak, aby byl odolný vůči extrémním podmínkám, jako jsou teploty, vlhkost, prach a vibrace, které jsou běžné v průmyslovém prostředí. (Šmejkal, Martinásková, 1999; Šmejkal, 2005).

1.2.2 Historie PLC

První PLC

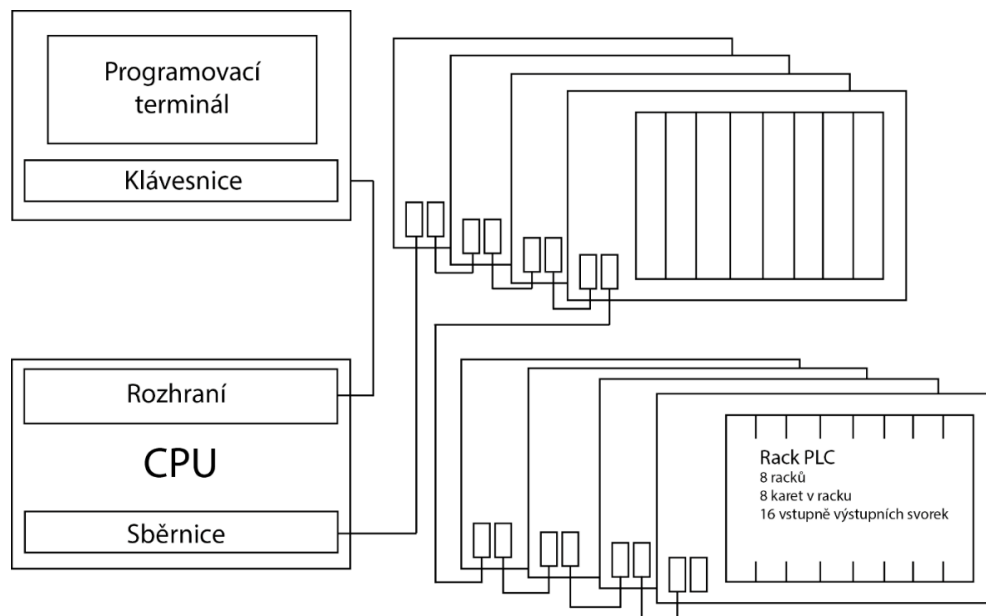
Po roce 1960 se začala americká společnost vyrábějící automobily, General Motors, zabývat myšlenkou náhrady reléových řídicích systémů s pevnou logikou počítačovými systémy schopnými pružněji reagovat na potřebné změny v konfiguraci reléového řídicího systému, což do teď znamenalo kompletní přepojení někdy i mnoha stovek kontaktů relé. Tato nejen časová náročnost vedla k vzniku konceptu pružného automatizačního systému. V roce 1968 byla vyhlášena soutěž na dodávku počítačového řízení pro výrobní závody právě společnosti General Motors. Základem pro vyhlášení soutěže byl referát, který byl přednesen Billem Stonem na konferenci General Motors Corporation ve Westinghouse, kde byly řešeny výrobní záležitosti a byl vytvořen návrh na standardizaci strojového řízení. Následující čtyři společnosti se přihlásily do soutěže: společnost Information Instruments (kterou později zakoupila společnost Allen-Bradley, nyní Rockwell Automation), Digital Equipment Corp. (DEC), Century Detroit a Bedford Associates (později Modicon). Tyto společnosti navrhly (pravděpodobně podobné) řídicí systémy částečně podobné minipočítačům té doby. Bedford

Associates v soutěži zvítězila a v roce 1969 byl vyroben první PLC. První programovatelný logický automat je znázorněn na obrázku 1.14. (Schneider Electric, 2019).



Obrázek 1.12 – MODICON 084 (Schneider Electric, 2019)

Mikroprocesor byl přizpůsoben pro provoz v průmyslovém prostředí. Tvořil jádro systému a byl instalován v racku, do kterých byly také instalovány vstupně / výstupní moduly každý až o 16 binárních signálech. Celková konfigurace umožňovala obsluhu až 256 signálů. Schématický náčrt konfigurace takového PLC je znázorněn na obrázku 1.15. Významné už tehdy bylo, že modul vstupů nebo výstupů mohl být instalován do libovolné pozice a libovolného racku podle požadavků uživatele. První PLC dostalo označení 084 jako 84. projekt společnosti Bedford Associates, která se poté věnovala vývoji, výrobě a servisu tohoto produktu. Produkt byl uveden na trh pod názvem MODICON (zkratka ze slov MO-dular DI-gital CON-troller), jako modulární systém pro řízení procesů. Číslice 84 na konci označení byly nadále používány jako synonymum pro PLC MODICON až do označení 984. Jeden z prvních vyrobených kusů PLC Modicon 084 byl v provozu 20 let a je vystaven v sídle Modiconu, North Andover, Massachusetts. Richard „Dick“ E. Morley byl jedním z prvních z vývojářů PLC a je považován za „otce“ PLC. (Schneider Electric, 2019).



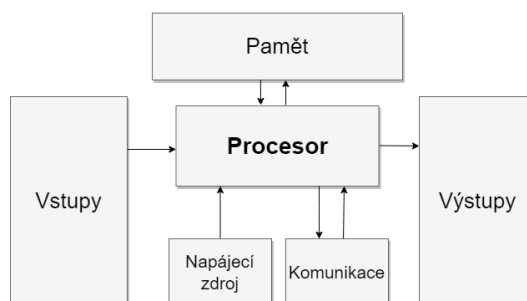
Obrázek 1.13 – Konfigurace MODICONu 084

1.2.3 Dělení PLC

Programovatelné logické automaty jsou velice dobře škálovatelné a vždy záleží na konkrétní aplikaci řízení, abychom určili, jak velké PLC potřebujeme. PLC se proto v základě dělí na kompaktní, modulární a v poslední době se objevuje i termín Mikro PLC, což je malé kompaktní PLC. Ještě než tyto druhy PLC podrobně popíšeme je třeba si také podrobně popsat PLC obecně. (Šmejkal, Martinásková, 1999; Šmejkal, 2005).

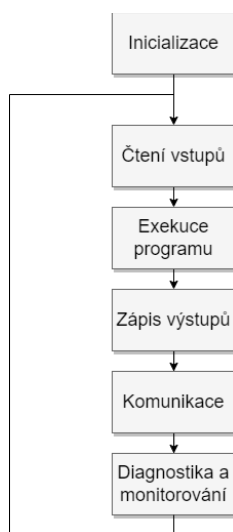
Obecná specifikace PLC

Všechny druhy PLC se dají zjednodušit obecným blokovým schématem, které je znázorněné na obrázku 1.14.



Obrázek 1.14 – Obecné blokové schéma programovatelného automatu

Výše vyobrazené blokové schéma se skládá z 6 bloků. Hlavní blok s názvem Procesor je výpočetní jednotka (CPU) a jedná se o mikroprocesor, který zpracovává a řídí všechny funkce systému. Mikroprocesor interpretuje a provádí programy, které jsou nahrány do paměti zařízení programátorem. Dalším blokem je Paměť, která uchovává nejen programy, které byly nahrány do PLC, ale také data ze vstupních a výstupních bloků. Paměť bývá typu RAM i typu ROM (EEPROM nebo FLASH). Do vstupních a výstupních bloků se připojují vnější zařízení jako jsou senzory, spínače a relé. Vstupní blok přijímá signály ze sensorů a převádí je na digitální hodnoty, které může procesor zpracovat. Výstupní blok pak převádí digitální signály z procesoru na analogové nebo digitální hodnoty, které mohou být použity k řízení vnějších zařízení, jako jsou motory, ventily, relé a další akční členy. Napájecí zdroj zajišťuje napájení všech komponent PLC včetně procesoru, paměti, vstupních a výstupních bloků. Napájecí zdroj může být interní nebo externí a musí být schopen poskytnout dostatečný proud pro všechny komponenty systému. Komunikační rozhraní umožňuje propojení PLC s jinými zařízeními a systémy jako jsou HMI panely, počítače nebo jiné automaty. Komunikační rozhraní může zahrnovat sériovou komunikaci nebo Ethernet porty a podporovat různé průmyslové komunikační protokoly jako je Modbus nebo novější EtherCAT. Přes komunikační rozhraní zpravidla probíhá i samotné programování PLC. Na PLC je spuštěný program vytvořený vývojářem, který vykonává požadovanou funkci. Tento program se stále opakuje ve smyčce a opakovaně plní předdefinované úkony, dokud není např. pozastaven operátorem. Na obrázku č. 1.13 je znázorněné blokové schéma prvního cyklu programu PLC po spuštění. (Šmejkal, Martinásková, 1999; Šmejkal, 2005).



Obrázek 1.15 – Blokové schéma prvního cyklu programu PLC

První cyklus programu začíná inicializací. PLC se spustí a inicializuje své komponenty jako jsou procesor, paměť, vstupní bloky, výstupní bloky a komunikační rozhraní. Během inicializace se také načte program z paměti do procesoru. V dalším kroku probíhá čtení vstupů, kdy PLC pravidelně čte data ze vstupních bloků a ukládá je do paměti. Po přečtení vstupů proběhne samotná exekuce programu. Procesor provádí program uložený v paměti, který zahrnuje logické a aritmetické operace, časování, počítání a další funkce. Následuje zápis výstupů na základě výsledků zpracování programu procesor aktualizuje data ve výstupních blocích, které ovládají akční členy. Dále PLC komunikuje s dalšími systémy a zařízeními prostřednictvím svého komunikačního rozhraní, aby předávalo informace nebo přijímalo pokyny. Posledním krokem je diagnostika a monitorování. PLC sleduje svůj vlastní stav a provoz, aby detekovalo potenciální problémy a udržovalo optimální výkon. Tato smyčka se opakuje po celou dobu běhu PLC a krok inicializace se vždy provádí jen jednou po startu PLC. (Šmejkal, Martinásková, 1999; Šmejkal, 2005).

Kompaktní PLC

Kompaktní PLC, jak z názvu vypovídá, je malé, efektivní a z pravidla se jedná o PLC s pevným počtem vstupů, výstupů nebo s pevně danými možnostmi komunikace s dalšími zařízeními. Výhodou kompaktních PLC jsou zejména jejich malé rozměry. Navzdory menším rozměrům si často zachovávají robustní sadu funkcí a lze je použít v různých aplikacích průmyslové automatizace. V aplikacích například pro domácí automatizaci, kde předem známe počet vstupů a výstupů by bylo vhodné použít právě kompaktní PLC. Kompaktní PLC od firmy Eaton s označením EC4P je znázorněné na obrázku 1.16. (Šmejkal, Martinásková, 1999; Šmejkal, 2005).



Obrázek 1.16 – Kompaktní PLC Eaton EC4P (Eaton, nedatováno)

Nicméně i přes to, že kompaktní PLC se často používá na předem stanovené počty vstupů a výstupů, některá kompaktní PLC jsou rozšiřitelná o moduly, takže některé kompaktní PLC jsou i modulární. Příkladem takového PLC je Simatic S7-1200 od firmy Siemens, které je zároveň kompaktní i modulární a je znázorněné na obrázku 1.17. (Šmejkal, Martinásková, 1999; Šmejkal, 2005, Siemens, 2023).



Obrázek 1.17 – Kompaktní i modulární PLC Simatic řady S7-1200 (RS-ONLINE, nedatováno)

I přes to, že Simatic S7-1200 vypadá na první pohled za malé a kompaktní, toto PLC umožňuje rozšíření pomocí až o 8 modulů, a proto se o něm dá uvažovat i jako o modulárním PLC. Tato možnost rozšíření je však také omezena na připojení pouze na jedné DIN liště, jejíž délku určuje velikost samotného rozvaděče. (Siemens, 2023).

Nevýhodou kompaktních PLC je, že v takovéto konfiguraci toto řízení neumožňuje decentralizaci. To znamená, že řídicí proces je ovládán z jednoho místa, tedy z jednoho rozvaděče.

Mikro PLC

Mikro PLC jsou, jak již bylo řečeno, malé kompaktní PLC, které se vyznačují nižší cenou. Byly navrženy pro jednoduché aplikace s malým počtem vstupů a výstupů. Svoje místo mají v automatizaci, resp. v ovládání menšího stroje nebo procesu s menší náročností na vstupy

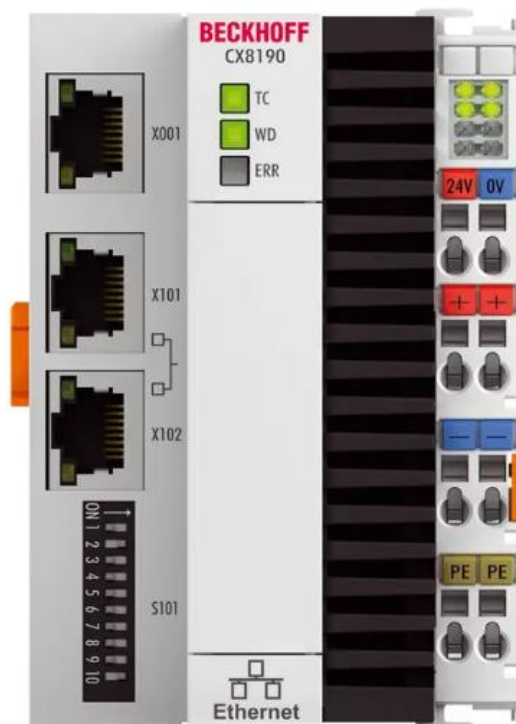
a výstupy, ale také v domácí automatizaci. Příkladem Mikro PLC je Allen-Bradley typu Micro810, které je znázorněné na obrázku 1.18.



Obrázek 1.18 – Mikro PLC Allen-Bradley Micro810 (Rockwell Automation, nedatováno)

Modulární PLC

Modulární PLC jsou definovány jako rozšiřitelné a flexibilnější PLC, které jsou sestaveny z různých modulů. Základem je vždy procesorový modul (CPU). Dále jsou k dispozici paměťové moduly pro případ, že je pro danou aplikaci potřeba uchování velkého množství dat. Komunikační moduly slouží zejména při potřebě komunikovat se zařízením třetích stran jako jsou HMI panely nebo speciální sensorové jednotky. Vstupní a výstupní moduly slouží pro interakci s okolním prostředím, kde se v případě vstupů jedná o signály z tlačítek nebo ze sensorů a výstupní moduly pro ovládání akčních členů či LED signalizace. Tyto moduly jsou spojeny zpravidla interní komunikační sběrnici vlastní pro daného výrobce a s procesorovou jednotkou, což umožňuje konfiguraci PLC dle potřeb konkrétní aplikace. Příkladem takové interní sběrnice, je EtherCAT (Ethernet for Control Automation Technology), což je sběrnice vyvinutá firmou Beckhoff Automation. Komunikační sběrnice EtherCAT bude podrobněji popsána v kapitole 1.2.6. Příkladem modulárního PLC je například PLC CX8190 rovněž od firmy Beckhoff Automation, který je znázorněný na obrázku 1.19. (Šmejkal, Martinásková, 1999; Šmejkal, 2005; Beckhoff, 2023).



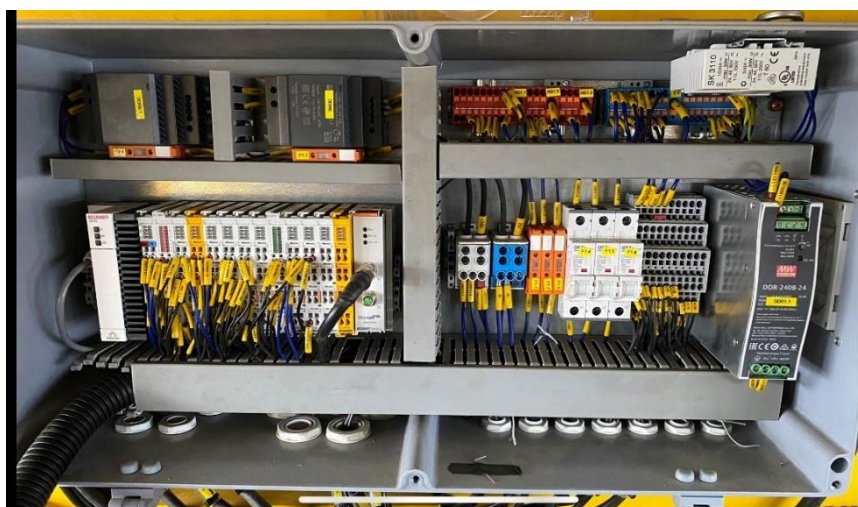
Obrázek 1.19 – Modulární PLC Beckhoff CX8190 (Beckhoff, 2023)

PLC CX8190 je typickým příkladem modulárního PLC, protože se jedná pouze o procesorovou jednotku, která se sice dá naprogramovat, ale bez dalších modulů, resp. terminálů vykonávaný program nemá možnost jakékoli interakce s okolím, snad jen kromě možnosti naprogramovat LED signalizační prvek s názvem WD, který je v procesorové jednotce integrován. Pro splnění funkce PLC, což je řízení technologických procesů, je nutné přidat k procesorové jednotce další terminály. Těchto terminálů může být na komunikační sběrnici EtherCAT až 65 635. Modulární PLC jsou pro svoji konfiguraci zpravidla složitější na nastavení, nicméně možnosti škálovat složitý řídicí proces nám poskytují právě modulární PLC. V této diplomové práci se dále zaměříme na modulární PLC firmy Beckhoff Automation nesoucí označení CX8190. (Beckhoff, 2023).

Příklad použití PLC Beckhoff CX8190

Na obrázku 1.20 je ilustrován rozvaděč, jehož jádro tvoří právě PLC CX8190, které má za úkol zajistit provoz zásobníkového vozu MFS40 na šterk z kolejového lože. Z obrázku 1.20 je patrné, že modulární PLC CX8190 je osazeno o další terminály, mezi které patří digitální vstupní terminály, digitální výstupní terminály, analogové vstupní terminály, dále žluté

bezpečnostní terminály a také komunikační terminál pro sběrnici EtherCAT, která je pak zasazená zcela vpravo všech modulů PLC. Tento vůz na štěrk je ilustrován na obrázku 1.21.



Obrázek 1.20 – Rozvaděč vozu na štěrk MFS40

Stejně jako všechny PLC se i modulární PLC programují pomocí specifického softwaru dodávaného výrobcem. Tento software umožňuje uživatelům vytvářet, testovat, ladit a spouštět programy, které řídí provoz stroje nebo procesu. Programování a programovacímu prostředí bude věnována kapitola 1.2.5.



Obrázek 1.21 – Zásobníkový vůz na štěrk MFS 40 (Zutwargowiec, nedatováno)

1.2.4 Srovnání konkrétních typů PLC

Srovnání vybraných PLC na základě parametrů pro návrh řídicího systému jsou přehledně uvedeny v tabulce 1.1. Od každého typu PLC bude uveden jeden představitel. Za

kompaktní PLC bude uvedené PLC Eaton EC4P, za mikro PLC bude do srovnání přidáno PLC Allen-Bradley Micro810 a konečně za modulární PLC, Beckhoff CX8190.

Tabulka 1.1 – Srovnání konkrétních typů PLC (Beckhoff, 2023; Rockwell Automation, nedatováno; Eaton, nedatováno)

Typ PLC	Eaton EC4P	Allen-Bradley Micro810	Beckhoff CX8190
	Kompaktní	Mikro	Modulární
Rozměry (šířka, výška, hloubka)	107,5 mm x 90 mm x 72 mm	75 mm x 91 mm x 59 mm	71 mm x 100 mm x 73 mm
Počet vstupů	12x DI (z toho 4x AI)	8x DI (z toho 4 AI)	Rozšiřitelné
Počet výstupů	8x DO, 1x AO	4x DO	Rozšiřitelné
Komunikační rozhraní	Ethernet, CANopen, RS232	USB 2.0	Ethernet, rozšiřitelné
Podporované programovací jazyky	LD, FBD, ST	LD, FBD, ST	LD, FBD, ST, IL, SFC, C++, Matlab/Simulink
Velikost operační paměti	256 kB	Výrobce neuvádí	1 MB
Typ a rychlost procesoru	Infineon XC161 40 MHz	Výrobce neuvádí	ARM Cortex™-A9 800 MHz
Odolnost vůči prostředí	-20 až 55 °C	-20 až 55 °C	-25 až 60 °C
Stupěň krytí	IP20	Výrobce neuvádí	IP20
Bezpečnostní funkce	Výrobce neuvádí	Výrobce neuvádí	Rozšiřitelné pomocí bezpečnostního modulu
Požadavky na napájení	20,4 až 28,8 VDC	11,4 až 26,4 VDC 85 až 263 VAC	20,4 až 28,8 VDC

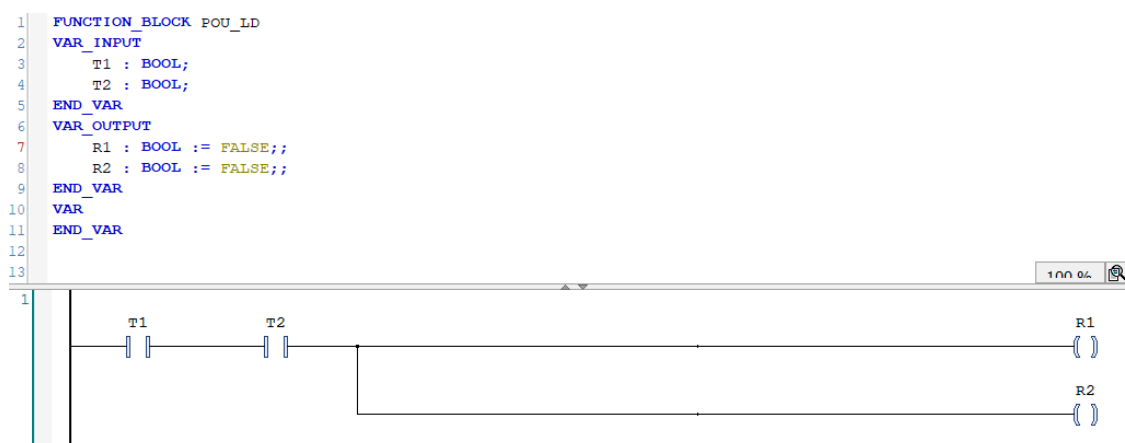
Následující kapitola bude pojednávat o mezinárodní normě IEC 61131-3, která určuje standard pro programování PLC. Norma se primárně zaměřuje na programovací jazyk Ladder Diagram, Structured Text, Function Block Diagram, Sequential Function Chart a Instruction List. (Teco a.s., 2007).

1.2.5 Programování PLC dle normy IEC 61131-3

Norma IEC 61131-3 je mezinárodní standard, který se zabývá průmyslovými řídicími systémy, konkrétně pak programovacími jazyky pro PLC. Jak již bylo řečeno, tato norma definuje pět programovacích jazyků, které budou v následujících podkapitolách popsány a uvedeny příklady podoby programu. Programovací jazyky mohou být textové, ale i grafické. Všechny jazyky, které norma popisuje mají svoje využití na různé aplikace, dle složitosti a podstaty úlohy, kterou má program na PLC vykonávat. Avšak jednoduché logické úlohy jsou realizovatelné pomocí všech pěti programovacích jazyků. Pro představu bude v následujících podkapitolách uvedené řešení jednoduché aplikace. Aplikace má za úkol při aktivaci signálů T1 a zároveň T2, kdy T1 a T2 reprezentují tlačítka, aktivovat výstup signálu R1 a R2, což reprezentuje například relé R1 a relé R2. Jako programovací prostředí je použit software TwinCAT, kde se v horní části vždy u každého z jazyků deklarují proměnné a v dolní části je realizace syntaxe programu dle použitého programovacího jazyka. (Teco a.s., 2007).

Ladder Diagram

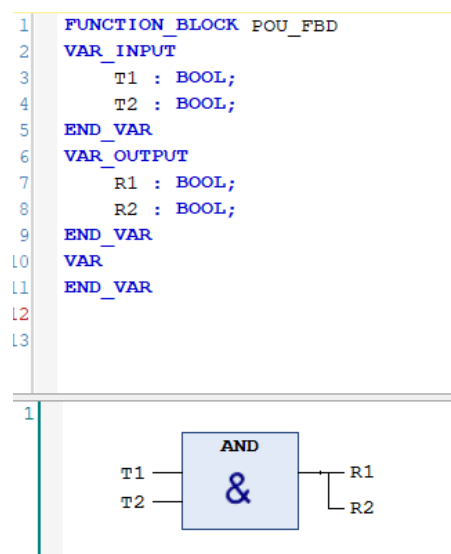
Ladder Diagram (LD) je grafický jazyk, který je speciálně navržen pro vizualizaci a programování logických obvodů a reléových systémů. Logické operace jsou v LD prezentovány jako schéma zapojení relé, což umožňuje jednoduché porozumění a ladění. LD se ukázal jako efektivní nástroj pro simulaci a testování reléových logických obvodů, čímž je zjednodušena diagnostika a odstraňování chyb v systémech. Na obrázku 1.22 je znázorněné řešení úlohy dvou tlačítek a dvou relé. (Teco a.s., 2007).



Obrázek 1.22 – Realizace pomocí jazyka LD

Function Block Diagram

Function Block Diagram (FBD) je dalším grafickým jazykem, který je ideální pro modelování systémů založených na funkcích a interakcích mezi nimi. V tomto jazyce jsou procesy modelovány pomocí vizuálních funkčních bloků a spojů mezi nimi, které reprezentují signály mezi bloky. Díky své vizuální povaze umožňuje FBD uživatelům snadno pochopit strukturu a logiku programu, což usnadňuje proces ladění, případné údržby nebo změny. Na obrázku 1.23 je znázorněné řešení úlohy dvou tlačítek a dvou relé za pomoci jazyka FBD. (Teco a.s., 2007).



Obrázek 1.23 – Realizace pomocí jazyka FBD

Structured Text

Structured Text (ST) neboli strukturovaný text je jedním z textových jazyků, který je vhodný pro programování složitých algoritmů a matematických operací. Tento jazyk vychází z tradičního textového programování a umožňuje vytváření komplexních a modulárních programů použitím běžných programovacích konstrukcí, jako jsou cykly, podmíněné větve a funkce. Logika programů je vyjádřena v textové formě a pro hlubší pochopení a dokumentaci kódu je zde možnost přidávání komentářů. Syntaxe a struktura tohoto jazyka jsou podobné tradičním vysokoúrovňovým jazykům jako je Pascal nebo C. Na obrázku 1.24 je znázorněné řešení úlohy dvou tlačítek a dvou relé za pomoci jazyka ST. (Teco a.s., 2007).

```

1 FUNCTION_BLOCK POU_ST
2 VAR_INPUT
3     T1 : BOOL;
4     T2 : BOOL;
5 END_VAR
6 VAR_OUTPUT
7     R1 : BOOL := FALSE;
8     R2 : BOOL := FALSE;
9 END_VAR

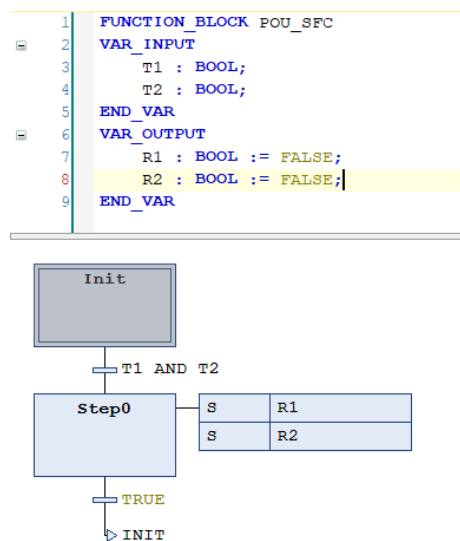
1 IF T1 AND T2 THEN
2     R1 := TRUE;
3     R2 := TRUE;
4 ELSE
5     R1 := FALSE;
6     R2 := FALSE;
7 END_IF

```

Obrázek 1.24 – Realizace pomocí jazyka ST

Sequential Function Chart

SFC je ideální pro vizualizaci, návrh a implementaci komplexních kontrolních procesů. Je založen na principu řízení sekvencí a uživatelům je umožněno vytvářet stavové diagramy, které reprezentují různé stavy a přechody mezi stavy řídicího procesu. Pomocí SFC je proces modelován jako série kroků a přechodů. K akcím mohou být přidruženy kroky, zatímco přechody reprezentují podmínky mezi kroky, které musí být splněny, než může k přechodu na další krok dojít. Na obrázku 1.25 je znázorněné řešení stejného problému, tedy dvou tlačítek, která aktivují dvě relé. (Teco a.s., 2007).



Obrázek 1.25 – Realizace pomocí jazyka SFC

Instruction List

Instruction List je textový jazyk používaný pro programování na nízké úrovni, kde je potřeba detailní kontroly nad hardwarem. Syntaxe je podobná jazyku Assembler pro mikropočítače. Při vývoji v IL je prioritou účinnost a minimalizace paměťové náročnosti. Každý řádek kódu v IL reprezentuje jednu operaci, resp. instrukci, což umožňuje uživatelům provést detailní analýzu a kontrolu chování systému. Software TwinCAT 3, což je aktuální používaná verze, tento jazyk již nepodporuje, proto jsou řádky označené červenou vlnovkou, ale i přes to je na obrázku 1.28 znázorněné řešení tlačítek a relé, které by bylo akceptované softwarem TwinCAT 2. (Mitsubishi, 2006)

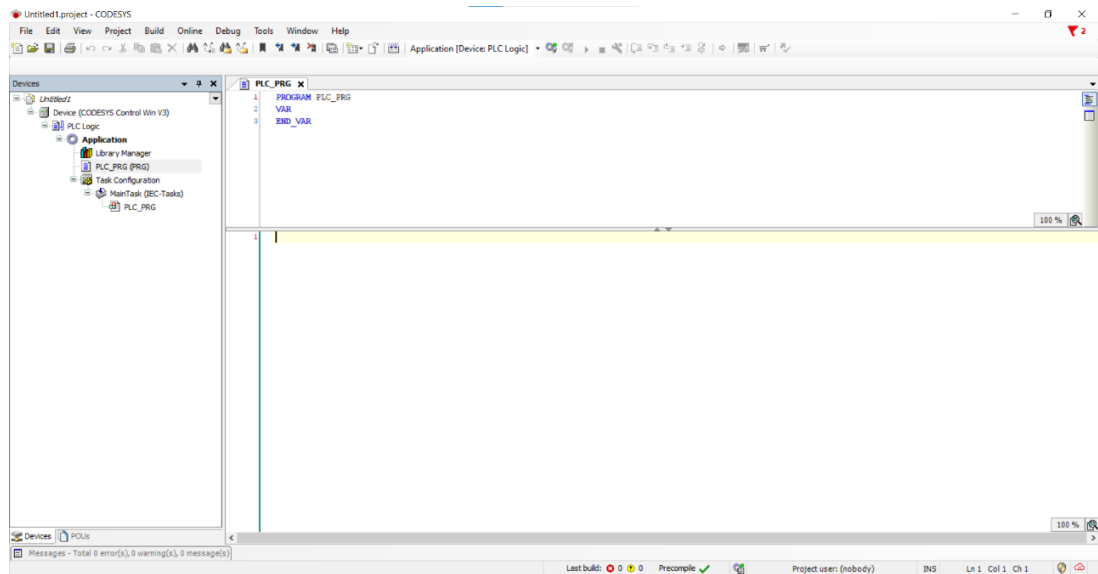
```
1  FUNCTION_BLOCK POU_IL
2  VAR_INPUT
3      T1 : BOOL;
4      T2 : BOOL;
5  END_VAR
6  VAR_OUTPUT
7      R1 : BOOL := FALSE;
8      R2 : BOOL := FALSE;
9  END_VAR
10 VAR
11
12 LD  T1
13 AND T2
14 ST  R1
15 ST  R2
```

Obrázek 1.26 – Realizace pomocí jazyka IL

Programovací prostředí PLC

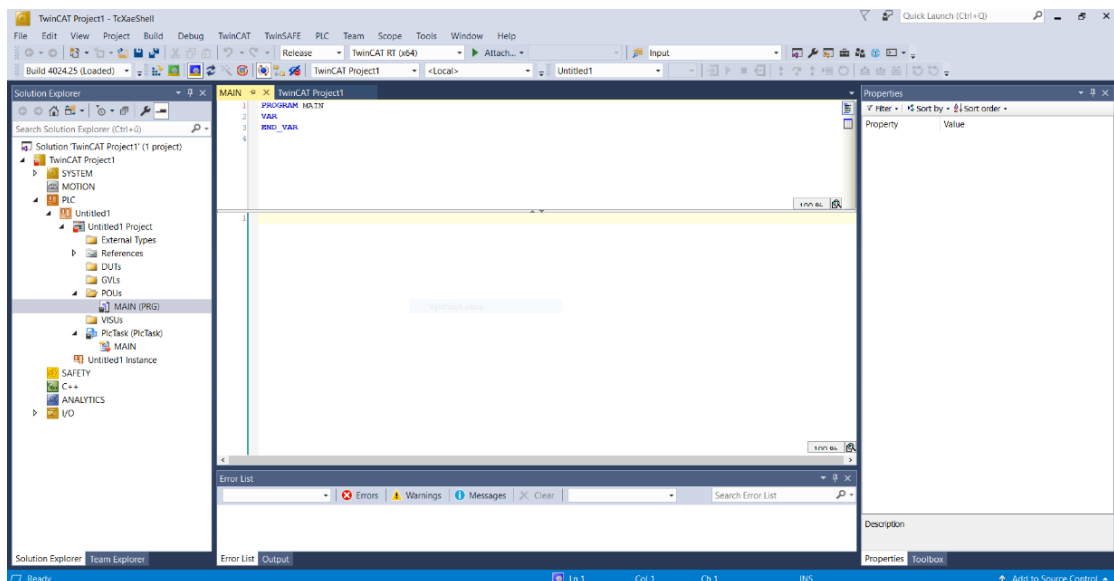
V oblasti průmyslové automatizace je využití programovacích prostředí pro PLC zásadní. CODESYS, univerzální programovací prostředí, je oblíbený pro svou schopnost vytvářet, testovat a ladit programy pro řídicí systémy od různých výrobců. Toto prostředí je charakterizováno podporou všech pěti jazyků definovaných v normě IEC 61131-3, včetně ST, IL, LD, FBD a SFC. Tato podpora zajišťuje jeho flexibilitu a kompatibilitu s mnoha typy PLC. Na obrázku 1.27 je znázorněné prostředí CODESYS. Beckhoff, renomovaný výrobce PLC, vytvořil vlastní programovací prostředí s názvem TwinCAT. Tento software se vyznačuje integrací PLC s operačním systémem Windows Embedded Compact 7 a s možností programování v jazycích normy IEC 61131-3. TwinCAT zároveň podporuje jazyk C++ a má podporu aplikace MATLAB/Simulink. Díky těmto vlastnostem je TwinCAT schopen poskytnout vysokou úroveň integrace a komunikace se systémy Beckhoff. Další výhodou prostředí TwinCAT je pak možnost programování bezpečnostních aplikací anebo návrhem

vizualizace pro HMI panely. Programovací prostředí TwinCAT je znázorněné na obrázku 1.28. (Beckhoff, 2023).



Obrázek 1.27 – Programovací prostředí CODESYS

Každé z těchto prostředí, ať už je to CODESYS nebo TwinCAT 3, poskytuje uživatelům silné nástroje pro vývoj, ladění a implementaci PLC programů, přičemž každé má své unikátní vlastnosti a výhody. Průmyslová automatizace je díky těmto programovacím prostředím značně usnadněna.



Obrázek 1.28 – Programovací prostředí TwinCAT 3

1.2.6 Komunikační protokoly

Tato podkapitola pojednává o vybraných komunikačních protokolech programovatelných logických automatů. Jedná se zejména o protokoly, které se v průmyslu používají nejčastěji. Vybrané komunikační protokoly jsou Modbus a EtherCAT. Komunikační protokoly hrají důležitou roli při výměně informací mezi různými zařízeními a systémy v průmyslovém prostředí. V první řadě však bude krátce představen termíny Fieldbus a Průmyslový Ethernet (Industrial Ethernet), což jsou rodiny průmyslových komunikačních protokolů. Následně dojde k podrobnějšímu představení modelu ISO/OSI, který je klíčový k porozumění komunikačních protokolů a práci s těmito protokoly. Dále budou popsány vybrané komunikační protokoly. (Michalec, 2020).

Fieldbus a Průmyslový Ethernet

Pojmy Fieldbus a Průmyslový Ethernet (Ethernet/IP) se označují dvě rodiny komunikačních protokolů, které se v průmyslu nejvíce vyskytují. Hlavním rozdílem mezi nimi je ten, že Fieldbus je založený primárně na sériových komunikačních sběrnicích, jako je např. RS-485, kdežto Průmyslový Ethernet, jak už z názvu vypovídá, je založený na ethernetu. V kontextu referenčního modelu ISO/OSI se potom jedná o rozdíly, kdy Fieldbus zpravidla nevyužívá všechny jeho vrstvy, kdežto Průmyslový Ethernet využívá všechny vrstvy modelu ISO/OSI. V tabulce 1.2 je uvedený stručný přehled průmyslových komunikačních protokolů, zdali jsou členem Fieldbus rodiny komunikačních protokolů anebo členy rodiny Průmyslového Ethernetu, přičemž EtherCAT je jako jediný kombinací obou rodin komunikačních protokolů. Dalo bys se říct, že spojuje to nejlepší z obou světů. EtherCAT bude podrobně popsán dále v této kapitole. (Michalec, 2020; EtherCAT, nedatováno, Peterka, 2021).

Tabulka 1.2 – Přehled vybraných komunikačních protokolů

Protokol	Rodina	Maximální počet jednotek	Maximální délka vodiče
FOUNDATION Fieldbus H1	Fieldbus	32 na segment (až 240 s opakovači)	Až 1900 m při rychlosti 31,25 kbit/s
Profibus DP	Fieldbus	127	Až 1200 m
Profibus PA	Fieldbus	32 na segment	Až 1900 m

Tabulka 1.2 – Přehled vybraných komunikačních protokolů, pokračování

Modbus RTU nebo ASCII (RS-485)	Fieldbus	32 na segment (až 247 s opakovači)	Až 1200 m
DeviceNet	Fieldbus	64 (až 247 s opakovači)	Až 500 m při rychlosti 125 kbit/s
CAN (nízká rychlost)	Fieldbus	110	Až 1000 m
CAN (vysoká rychlost)	Fieldbus	110	Až 40 m při rychlosti 1000 Mbit/s
EtherCAT	Průmyslového Ethernetu s prvky Fieldbus	65535	100 m (pro Fast Ethernet), až 2 km pomocí optického kabelu
Profinet	Průmyslový Ethernet	Záleží na implementaci, typicky až tisíce	100 m (Fast Ethernet)
Ethernet/IP	Průmyslový Ethernet	Bez limitu, záleží na typu sítě	100 m (Fast Ethernet)
Modbus TCP	Průmyslový Ethernet	Bez limitu, záleží na typu sítě	100 m (Fast Ethernet)
CC-Link IE	Průmyslový Ethernet	120	Až 550 m mezi stanicemi
Powerlink	Průmyslový Ethernet	Bez limitu, záleží na typu sítě	100 m (Fast Ethernet)

Model ISO/OSI

Model ISO/OSI (International Standards Organization/Open Systems Interconnection) je referenční model, který popisuje, jak se přesouvají informace mezi počítači a je základem pro pochopení komunikace mezi PLC a mezi dalšími prvky, které se v průmyslové automatizaci používají jako jsou senzory, akční členy a regulátory. Model ISO/OSI má 7 vrstev, kdy každá z těchto vrstev má specifickou úlohu v procesu komunikace. Není nutné, aby každý průmyslový komunikační protokol využíval všechny vrstvy modelu, ale každý komunikační protokol musí využívat alespoň dvě vrstvy. Na obrázku 1.29 je graficky znázorněný model ISO/OSI. (Peterka, 1992; IJS, nedatováno).



Obrázek 1.29 – Model ISO/OSI (Počítačové sítě, 2010)

V modelu ISO/OSI je od každé vrstvy vyžadováno, aby poskytovala služby následující vyšší vrstvě, přičemž detaily realizace služby jsou pro vyšší vrstvu skryty. Před přesunem dat mezi vrstvami jsou data rozdělena do paketů. K těmto paketům jsou poté v každé vrstvě přidávány další doplňkové informace, jako jsou formátování a adresa, které jsou nezbytné pro úspěšný přenos dat v síti. Tímto způsobem je zajištěna modularita a interoperabilita vrstev, které umožňují vylepšení nebo nahrazení jednotlivých vrstev bez nutnosti změny celého systému, a také efektivnější řešení problémů v síti prostřednictvím zaměření na konkrétní vrstvu. V následující podkapitole bude proveden popis jednotlivých vrstev. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Vrstvy ISO/OSI modelu

Fyzická vrstva (Physical Layer)

Fyzická vrstva je první vrstvou modelu ISO/OSI a tato vrstva zajišťuje přenos jednotlivých bitů mezi příjemcem a odesílatelem dat prostřednictvím přenosové cesty, což mohou být metalické vodiče, optická vlákna anebo radiové vlny. V případě metalických vodičů se tato vrstva zabývá např. napěťovými úrovněmi logické „0“ a logické „1“, jaký časový rámec je určen právě pro jeden bit, kolik kontaktů mají metalické vodiče mít, jaký typ konektoru je použit atp. Příkladem pro fyzickou vrstvu je komunikační sériový standard RS-485, který na všechny výše uvedené otázky odpovídá.

Jak již bylo řečeno, každá vrstva komunikuje pouze se „sousední“ vrstvou a v tomto případě fyzická vrstva předkládá informace vrstvě linkové ve formě přijímaných bitů, popř. odeslaných bitů. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Linková vrstva (Data Link Layer)

Vrstva označovaná jako spojová, nebo také vrstva datového spoje, se stará o seskupení přenesených bitů do tzv. rámců o velikosti několik stovek bajtů. Tato vrstva má za úkol, aby byl rozpoznán začátek a konec rámce, a jejich správnost je kontrolována pomocí cyklického redundantního kontrolního součtu (CRC). Pokud je požadováno zajištění spolehlivého přenosu na této vrstvě, musí být odesílatel informován příjemcem, pokud je rámec přijat poškozený a je nutné jej odeslat znovu. Na linkové vrstvě je také prováděno řízení toku, což je řízení rychlosti přenosu tak, aby příjemce byl schopen zpracovávat rámce. Pokud by přenos byl příliš rychlý a příjemci by již nestačila vyrovnávací paměť k ukládání rámců, bylo by nutné další rámce zahodit. Rychlost přenosu je tedy zpravidla řízena příjemcem pomocí určitého mechanismu.

Tato vrstva se dále dělí na dvě podvrstvy a to LLC (Logical Link Control), která se zabývá již popsanými úkoly, a MAC (Media Access Control), která hlídá kolizní stavy při přístupu více uzlů na společné medium v lokálních LAN sítích.

Tato vrstva zajišťuje přenos pouze u přímého spojení, tím pádem i adresy na úrovni linkové vrstvy jsou mají jeden rozměr. Bezprostředně vyšší síťové vrstvě jsou poskytovány služby typu odešli rámeček z daného uzlu, resp. přijmi rámeček z daného uzlu. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Síťová vrstva (Network Layer)

Síťová vrstva se stará o směrování dat dále než k sousedním uzlům, což je důležité v situacích, kdy spojení mezi příjemcem a odesílatelem není přímé a vede přes jeden či více mezilehlých uzlů. Data jsou v této vrstvě členěna do tzv. paketů, které jsou bloky dat s hlavičkou na úrovni síťové a vyšší vrstvy. Jsou zde používány IP adresy obou koncových účastníků a také informace o potvrzování nebo o řízení toku. Směrování neboli rozhodování o směru, jakým budou data putovat, je prováděno na základě znalosti topologie sítě. U spojovaných přenosů je nejdříve určena cesta mezi odesílatelem a příjemcem, zatímco u nespojovaných přenosů je směr přenosu určován pro každý blok znovu, někdy dokonce znovu i v každém přestupním uzlu. Síťová vrstva také zajišťuje volbu vhodné trasy přes mezilehlé uzly a postupné předávání jednotlivých paketů po této trase od původního odesílatele až ke konečnému příjemci. Aby toto mohla zvládnout, musí si být vědoma konkrétní topologie sítě, což znamená způsob vzájemného přímého propojení jednotlivých uzlů. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Transportní vrstva (Transport Layer)

Transportní vrstva je zodpovědná za manipulaci s balíky dat. Při odesílání dat je rozdělení datového balíku na packety prováděno touto vrstvou, která následně umožňuje, aby tyto packety byly posílány směrem k příjemci. Na druhé straně je na této vrstvě sestavení přijatých paketů zpět do původního balíku dat. Dále je transportní vrstvou vyrovnáván rozdíl mezi třemi síťově orientovanými spodními vrstvami a třemi aplikačně orientovanými vyššími vrstvami. Chyby mohou být transportní vrstvou detekovány, někdy dokonce i opraveny. Tato vrstva je schopná rozpoznat nesprávné pořadí doručených paketů díky jejich číselnému označení a uspořádat je správně. Je také schopná rozpoznat, který program je příjemcem dat, a na základě toho je správně doručit.

Je důležité poznamenat, že implementace transportní vrstvy obvykle nenastává v meziuzlech, jako jsou směrovače nebo mosty, ale spíše v koncových zařízeních. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Relační vrstva (Session Layer)

Relační vrstva se zabývá řízením komunikace mezi dvěma komunikujícími stranami. Je na ní založeno navazování, udržování a ukončení spojení mezi koncovými uzly. Při navazování spojení je rozhodováno, zda bude komunikace poloduplexní, kde komunikace probíhá v jednom směru najednou, nebo plně duplexní, kde obě strany komunikace mohou vysílat a přijímat současně. Také je zde určováno, zda bude spojení šifrováno. Během udržování spojení je relační vrstva zodpovědná za řízení toku dat a zajišťuje, že data jsou správně přijata. Na konci komunikační relace je relační vrstva také odpovědná za ukončení spojení a uvolnění zdrojů.

Tato vrstva pracuje v těsné spolupráci s transportní vrstvou, která zajišťuje bezchybné doručení datových packetů, a s aplikační vrstvou, která poskytuje uživatelské rozhraní pro komunikaci. I přes to, že bylo výše uvedeno, že každá z vrstev spolupracuje pouze se sousední vrstvou, aplikační vrstva má přístup i do nižších vrstev, včetně relační vrstvy, referenčního modelu, a proto relační vrstva spolupracuje i s vrstvou aplikační. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Prezentační vrstva (Presentation Layer)

Prezentační vrstva modelu ISO/OSI je zodpovědná za překládání a formátování dat tak, aby byla srozumitelná pro všechny nižší vrstvy na odesílací straně a pro všechny vyšší vrstvy na přijímací straně. Tato vrstva tedy umožňuje konverzi mezi různými formáty dat, což je klíčové pro zajištění správné komunikace mezi různými systémy a aplikacemi, které mohou používat různé kódovací schéma. Mimo to se prezentační vrstva může také zabývat kompresí dat pro efektivnější přenos a jejich šifrováním pro zajištění bezpečnosti přenosu. Je důležité si uvědomit, že se prezentační vrstva stará o strukturu a formát dat, nikoliv o jejich význam. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Aplikační vrstva (Application Layer)

Aplikační vrstva je na nejvyšší úrovni a zajišťuje interakci mezi koncovými uživateli a síťovými aplikacemi. Tato vrstva zaměřena na poskytování a správu služeb pro aplikace, a je to jediná vrstva, do které má uživatel přímý přístup. Součástí aplikační vrstvy jsou pouze standardizované prvky aplikací, jako jsou mechanismy pro přenos elektronické pošty. Avšak

konkrétní uživatelská rozhraní nejsou součástí aplikační vrstvy a mohou se lišit dle preferencí konkrétního výrobce. Nicméně aplikační vrstva se rovněž skládá z obecně použitelných mechanismů a zahrnuje pouze ty části aplikací, které jsou společné pro všechny uzly v síti. Například v případě elektronické pošty zajišťuje aplikační vrstva předávání zpráv v síti, zatímco uživatelské rozhraní, které se může lišit v každém uzlu, již není považováno za součást aplikační vrstvy.

Závěrem, aplikační vrstva poskytuje uživatelům přístup k síťovým službám a spravuje komunikaci mezi různými aplikacemi. Je to vrstva, kde se nachází programy pro řízení databází, souborové a tiskové servery, a příkazy operačního systému. Jejím úkolem je usnadnit komunikaci a umožnit interoperabilitu mezi různými síťovými aplikacemi.

Výše byly podrobně popsány jednotlivé vrstvy referenčního modelu ISO/OSI. V tabulce 1.3 jsou potom stručně uvedeny funkce jednotlivých vrstev. (Peterka, 1992; IJS, nedatováno; Počítačové sítě, 2010).

Tabulka 1.3 – Stručný přehled funkcí jednotlivých vrstev (IJS, nedatováno)

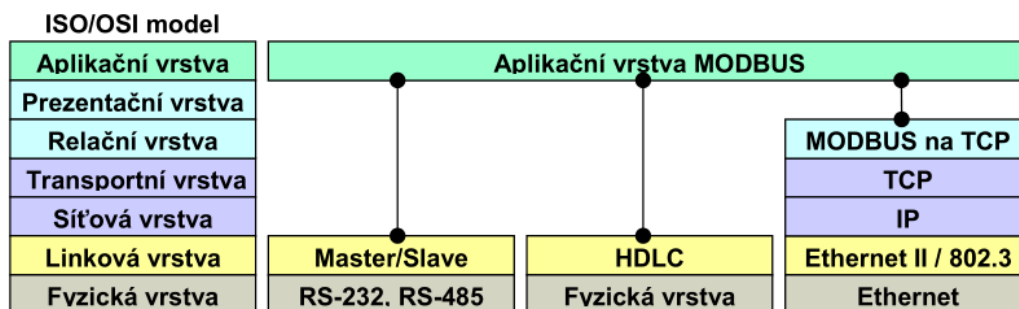
Název vrstvy	Funkce vrstvy
Aplikační	Vytvoření zprávy v aplikaci
Prezentační	Převedení zprávy do srozumitelného formátu pro příjemce
Relační	Vytvoření spojení s příjemcem a jeho údržba
Transportní	Dohled nad spolehlivým přenosem zpráv, opravy chyb, vytvoření packetů
Síťová	Vytvoření trasy a opatření packetů adresami a dalšími náležitostmi
Linková	Vytvoření rámců a jejich vysílání.
Fyzická	Přenos bitů elektrickými, optickými nebo rádiovými signály.

V následující podkapitolách bude podrobně popsán komunikační protokol Modbus obecně, poté bude popsána implementace Modbus RTU a následně bude popsán komunikační protokol EtherCAT, který je založený na Ethernetu.

Komunikační protokol Modbus

Modbus je jeden z nejstarších a nejrozšířenějších průmyslových komunikačních protokolů. Byl vyvinut firmou Modicon v roce 1979 a stal se standardem pro komunikaci průmyslových elektronických zařízení. Modbus funguje na aplikační vrstvě ISO/OSI modelu a umožňuje interakci typu klient – server mezi různými zařízeními napříč mnoha typy sítí a sběrnic. Podporuje různá komunikační média jako jsou sériové linky RS-232, RS-422 a RS-485, na kterých se používá ve variantách Modbus RTU nebo Modbus ASCII. Sériové linky v případě Modbus RTU a Modbus ASCII reprezentují fyzickou vrstvu. V případě Modbus TCP

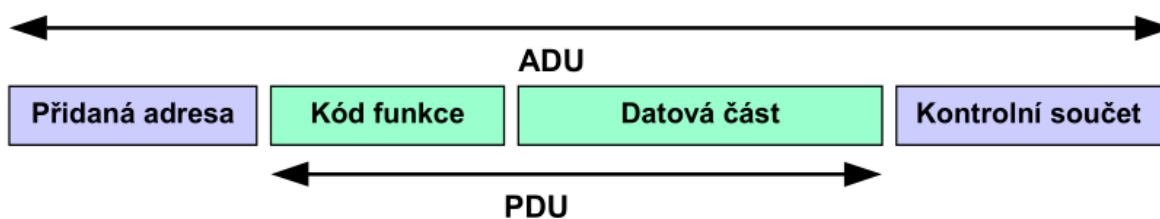
je možné jako komunikační médium použít, což v modelu ISO/OSI reprezentuje fyzickou vrstvu a Ethernet s protokolem TCP/IP. Proces komunikace je prováděn na základě požadavku a odpovědi, kde požadovaná funkce je určena funkcionalitou, která je integrovaná do požadavku. Na obrázku 1.30 jsou graficky znázorněny implementace, v kontextu ISO/OSI modelu, protokolu Modbus. (Ronešová, 2005).



Obrázek 1.30 – Příklady implementace (Ronešová, 2005)

Popis protokolu

Protokol Modbus definuje strukturu zprávy (telegramu) jako jednoduchou protokolovou datovou jednotku (PDU – Protocol Data Unit), která je nezávislá na základních komunikačních vrstvách, které definuje model ISO/OSI. Mapování protokolu Modbus na konkrétní sběrnici nebo síť může zavést některá další pole, což je pak aplikační datová jednotka (ADU – Application Data Unit). Grafické znázornění zprávy protokolu Modbus je ilustrováno na obrázku 1.31. (Ronešová, 2005).

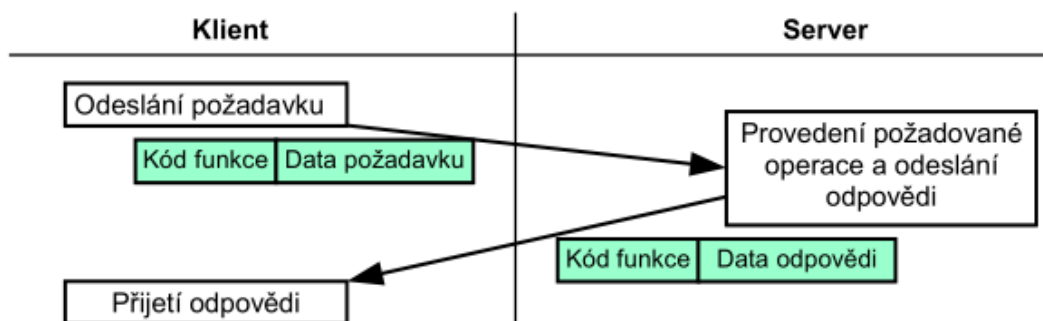


Obrázek 1.31 – Základní tvar Modbus telegramu (Ronešová, 2005)

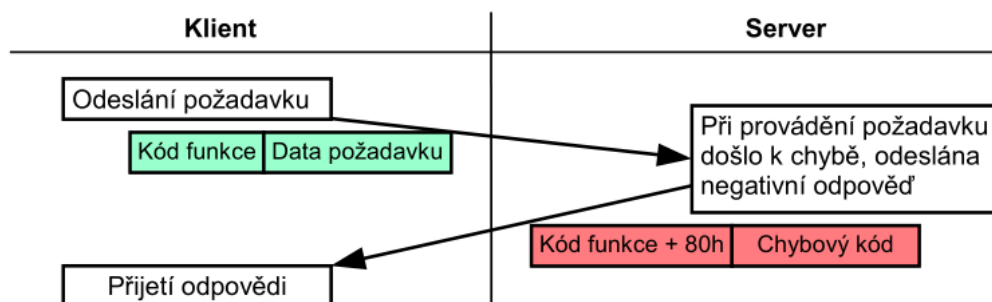
Aplikační datová jednotka je vytvořena klientem, který komunikaci iniciuje. Kód funkce indikuje serveru, jaký druh akce má provést. Kód funkce datové jednotky má velikost jednoho bajtu. Platné kódy jsou tedy v rozsahu 1 až 255. Rozsah 128 až 255 je vyhrazen pro odpovědi na výjimky, resp. chyby. Když je zpráva odeslána z klienta na server, kódu funkce říká serveru, jaký druh akce má provést. Kód funkce "0" není platný. K některým kódům funkcí se přidávají dílčí kódy funkcí, které definují více akcí. Datové pole zpráv odesílaných z klienta na server obsahují další informace, které server použije k provedení akce definované kódem

funkce. To může zahrnovat položky jako jsou diskrétní adresy a adresy registrů, množství položek, které mají být zpracovány, a počet skutečných bajtů v datové části. Datová část může být u některých druhů požadavků neexistující (nulové délky). V takovém případě server žádné další informace nevyžaduje. (Ronešová, 2005).

Pokud se ve správně přijatém telegramu nevyskytne žádná chyba související s požadovanou funkcí, datová část odpovědi ze serveru ke klientovi indikuje korektně přijatá data. Pokud dojde k chybě související s požadovanou funkcí, pole obsahuje kód výjimky, který může serverová aplikace použít k určení další akce, která má být provedena nebo tento kód výjimky může sloužit i při ladění komunikace. Na obrázku 1.32 je graficky znázorněna transakce s bezchybným provedením požadavku a odpovědi. Na obrázku 1.33 je pak ilustrována transakce požadavku s chybovým kódem. (Ronešová, 2005).



Obrázek 1.32 – Korektní transakce požadavku a odpovědi (Ronešová, 2005)



Obrázek 1.33 – Transakce požadavku a odpovědi s chybou (Ronešová, 2005)

Kódování dat

Protokol Modbus při odesílání dat delších, než je jeden bajt používá tzv. „big-endian“, což znamená, že je jako první odeslán nejvyšší bajt a nejnižší bajt je odeslán jako poslední. Například tedy 16bitové číslo 4F52h se pomocí protokolu Modbus odešle nejprve bajt 4Fh a poté 52h. (Ronešová, 2005).

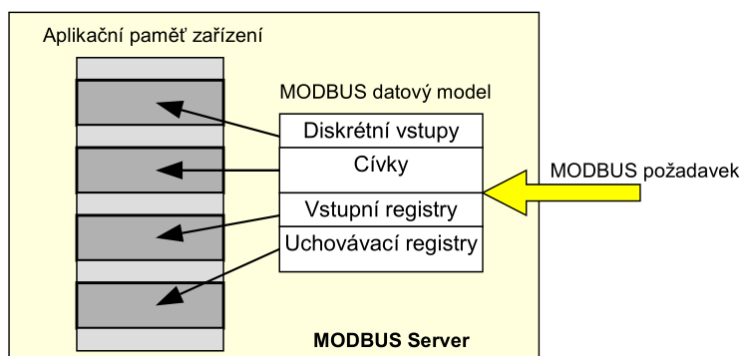
Datový model

Protokol Modbus je založen na 4 tabulkách, kdy každá z nich má svůj charakteristický význam. V tabulce 1.4 jsou tyto typy tabulek přehledně popsány. (Ronešová, 2005).

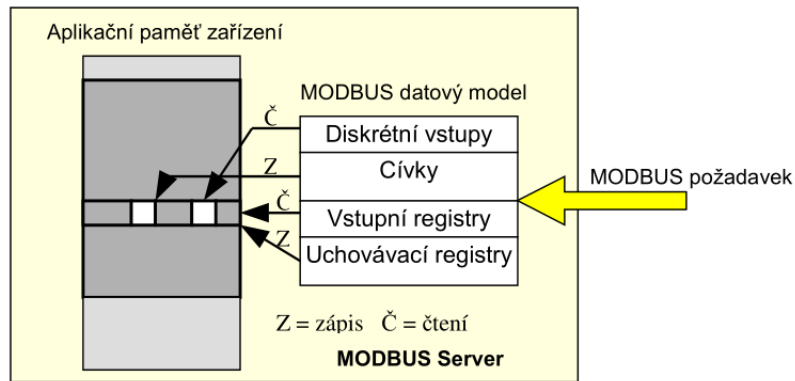
Tabulka 1.4 – Datový model Modbus (Ronešová, 2005)

Typ tabulky	Datový typ položky	Přístup	Popis	Adresa
Diskrétní vstupy (Discrete inputs)	1 bit	Čtení	Data poskytovaná I/O systémem	10000 až 19999
Cívky (Coils)	1 bit	Čtení / zápis	Data modifikovatelná aplikačním programem	0 až 9999
Vstupní registry (Input registers)	16bitové slovo (word)	Čtení	Data poskytovaná I/O systémem	30000 až 39999
Uchovávací registry (Holding registers)	16bitové slovo (word)	Čtení / zápis	Data modifikovatelná aplikačním programem	40000 až 49999

Adresní prostor je mapován do tabulek, což je vždy závislé na konkrétním zařízení, kde každá z tabulek může mít vlastní adresní prostor. Teoreticky může mít každá z tabulek až 2^{16} (65536) položek. Rozdělení na bloky o 10000, jak je patrné ze sloupce Adresa v tabulce 1.3, bývá kvůli zpětné kompatibilitě. Existují 2 možné způsoby, jak organizovat data v zařízení. Na obrázku 1.34 je první způsob, kde datový model má 4 oddělené bloky a na následujícím obrázku 1.35 je pak způsob organizace dat s jediným blokem. (Ronešová, 2005).



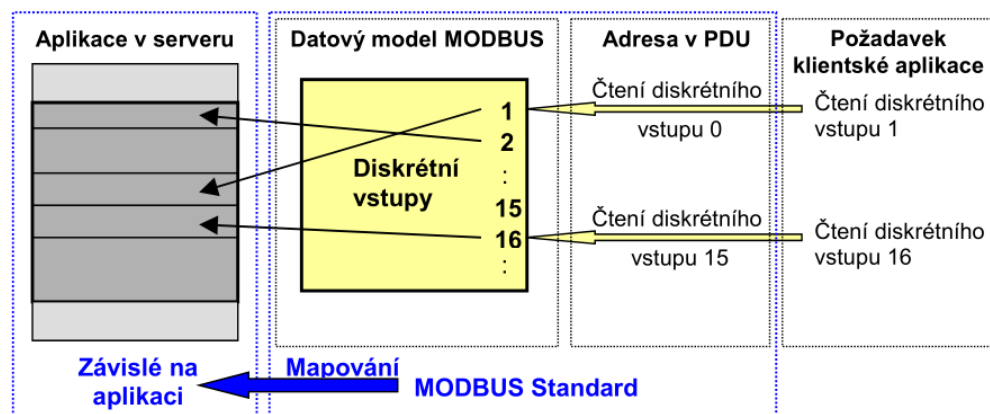
Obrázek 1.34 – Datový model s oddělenými bloky (Ronešová, 2005)



Obrázek 1.35 – Datový model s jediným blokem (Ronešová, 2005)

Adresovací model

Adresovací pravidla ve zprávách (PDU) jsou v protokolu Modbus přesně definována. Datové položky jsou v Modbus zprávách (PDU) adresovány od 0 do 65535. Adresování v rámci datového modelu složeného ze 4 datových bloků (tabulek) je dále definováno. Položky v datových blocích jsou v Modbus datovém modelu číslovány od 1 do n. Mapování položek Modbus datového modelu do aplikace v serveru je v režii výrobce plně ponecháno. Příklad adresování od požadavku klienta až po aplikaci v serveru je znázorněn na obrázku 1.36. Z obrázku je poznat, že datová položka X v datovém modelu je v PDU adresována jako položka X-1. (Ronešová, 2005).

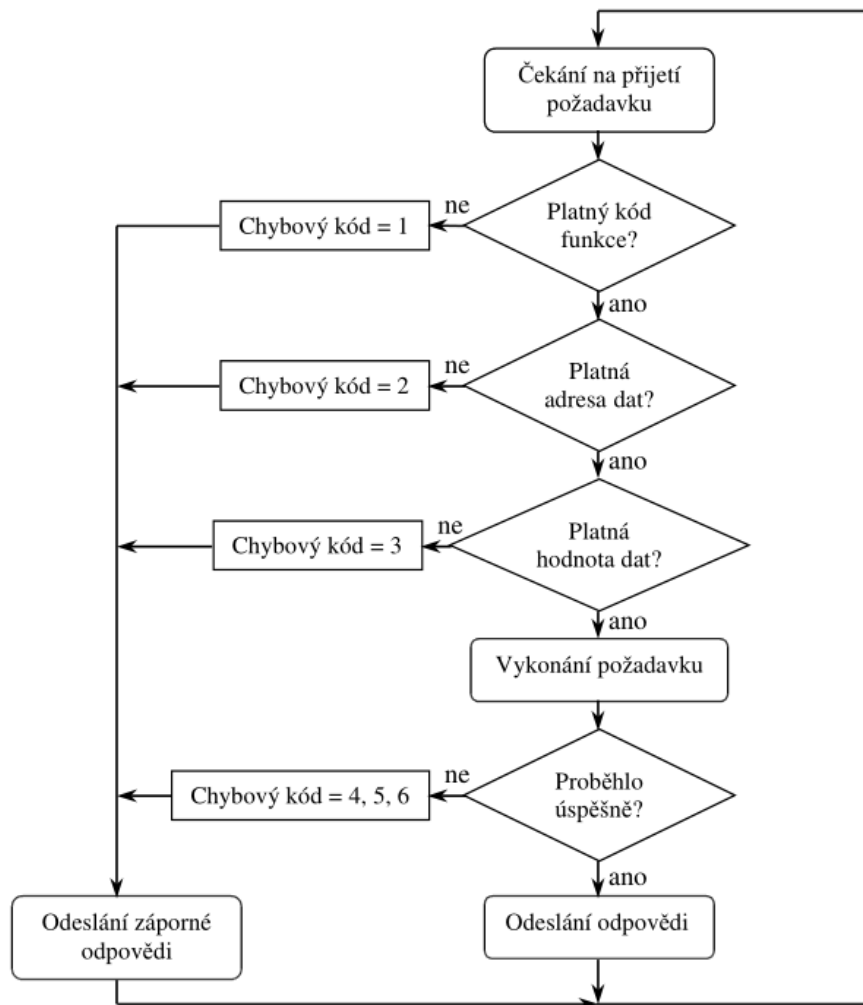


Obrázek 1.36 – Příklad adresování dle pravidel Modbus (Ronešová, 2005)

Definice Modbus transakce

Po zpracování požadavku serverem (ať již úspěšném či neúspěšném) je sestavena odpověď a klientovi odeslána zpět. V závislosti na výsledku zpracování požadavku je vytvořena

jedna ze dvou možných odpovědí. Když je odpověď pozitivní (Response), tak je kód funkce v odpovědi roven kódu funkce v požadavku. Pokud je odpověď negativní (Exception Response), je kód funkce v odpovědi roven kódu funkce v požadavku s přidaným 80h a je vrácen kód chyby, který udává důvod neúspěchu. Vývojový diagram, který graficky znázorňuje obecný postup zpracování požadavku na straně serveru je na obrázku 1.37. (Ronešová, 2005).



Obrázek 1.37 – Vývojový diagram zpracování Modbus požadavku na straně serveru (Ronešová, 2005)

Záporné (negativní) odpovědi (Exception response)

Když je klientem zaslán požadavek, očekává se odpověď serveru na tento požadavek. Můžeme identifikovat čtyři možné scénáře.

1. Server bezchybně přijme požadavek a může jej zpracovat a klientovi je vrácena normální odpověď,

2. server z důvodu komunikační chyby požadavek nepřijme, odpověď není vrácena. Na straně klienta poté dochází k vypršení časového limitu pro příjem odpovědi,
3. server požadavek přijme, ale rozpozná komunikační chybu, žádná odpověď není vrácena. To opět vede k vypršení časového limitu pro příjem odpovědi na straně klienta,
4. server požadavek přijme v pořádku, ale z nějakého důvodu jej nemůže zpracovat, klient dostane zápornou odpověď s vysvětlením důvodu neúspěchu.

Normální odpověď a záporná odpověď se od sebe liší podle nejvyššího bitu kódu funkce. Pokud je tento bit nulový, jde o normální odpověď. Pokud je tento bit nastaven, jedná se o zápornou odpověď. Datová část záporné odpovědi obsahuje kód chyby. Seznam možných chybových kódů je uveden v tabulce 1.5. (Ronešová, 2005).

Tabulka 1.5 – Chybové kódy Modbus (Ronešová, 2005)

Kód	Název	Význam
01	Ilegální funkce	Požadovaná funkce není serverem podporována
02	Ilegální adresa dat	Zadaná adresa je mimo serverem podporovaný rozsah
03	Ilegální hodnota dat	Předávaná data jsou neplatná
04	Selhání zařízení	Při provádění požadavku došlo k neodstranitelné chybě
05	Potvrzení	Kód určený k použití při programování. Server hlásí přijetí platného požadavku, ale jeho vykonání bude trvat delší dobu
06	Zařízení je zaneprázdněné	Kód určený k použití při programování. Server je zaneprázdněn vykonáváním dlouho trvajícího příkazu
08	Chyba parity paměti	Kód určený k použití při práci se soubory. Server při pokusu přečíst soubor zjistil chybu parity
0A	Brána – přenosová cesta nedostupná	Kód určený k práci s bránou (gateway). Brána není schopná vyhradit interní přenosovou cestu od vstupního portu k výstupnímu. Pravděpodobně je přetížená nebo nesprávně nastavená.
0B	Brána – cílové zařízení neodpovídá	Kód určený k práci s bránou (gateway). Cílové zařízení neodpovídá, pravděpodobně není přítomno

Implementace komunikačního protokolu Modbus

Kromě aplikační vrstvy ISO/OSI modelu jsou v Modbus standardu definovány i některé implementace protokolu pro specifický typ sítě či sběrnice. Jedná se o Modbus na TCP/IP a Modbus na sériové lince, přičemž se Modbus na sériové lince (Modbus Serial Line) dělí na

Modbus ASCII a Modbus RTU. Toto dělení bere v potaz pouze formát, ve kterém jsou data odesílána a dekodována. Režim Modbus ASCII není u zařízení podporující Modbus povinný. Tato diplomová práce se bude dále věnovat pouze sériovému Modbus v režimu RTU. (Ronešová, 2005).

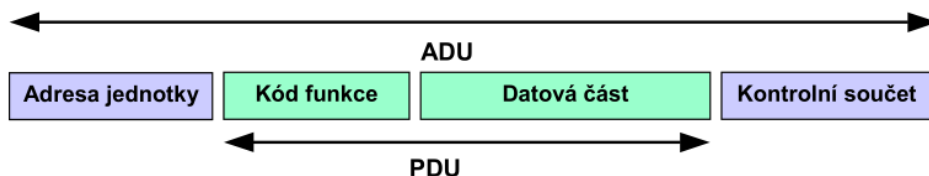
Modbus Serial Line

Protokol Modbus Serial Line je definován na 2. (linkové) úrovni ISO/OSI modelu a jedná se o protokol typu master – slave. Na 1. (fyzické) úrovni ISO/OSI modelu mohou být implementována různá sériová rozhraní jakými jsou například RS-232 nebo RS-485. Díky skutečnosti, že se jedná o master – slave protokol, může být na sběrnici pouze 1 master a 1 až 247 slave jednotek. Zahájení komunikace je vždy v kompetenci jednotky master, zatímco slave jednotka nesmí vysílat data bez předchozího pověření od mastera. Master může odesílat požadavky slave jednotkám ve dvou různých režimech. V režimu unicast je požadavek mastera adresován konkrétní slave jednotce, která následně posílá odpověď. V režimu broadcast master rozesílá požadavek na adresu 0, což odpovídá všem jednotkám, avšak žádná z jednotek na tento požadavek neodpovídá, protože tento požadavek je vždy příkaz k zápisu. Adresní prostor je určen 256 adresami. Vyhrazené adresy a jejich cíle definuje tabulka 1.6. (Ronešová, 2005).

Tabulka 1.6 – Adresy a jejich cíle v síti Modbus (Ronešová, 2005)

0	1 až 247	248 až 255
Broadcast adresa	Individuální adresa slave jednotky	Rezervováno

V síti Modbus nemá master specifikovanou adresu. Pouze slave jednotky jsou vybaveny adresou, která musí být v rámci celé Modbus sítě jedinečná. Základní formát Modbus aplikační zprávy na sériové lince je znázorněn na obrázku 1.38. Tato zpráva kromě standardní Modbus PDU obsahuje také pole s názvem "Adresa jednotky", jež ukazuje adresu konkrétní slave jednotky. Dále je zde pole "Kontrolní součet", které má za úkol detekovat chyby. Toto pole může obsahovat buď CRC nebo LRC kód v závislosti na zvoleném vysílacím režimu. Jak již bylo uvedeno, existují dva sériové vysílací režimy, a to Modbus RTU a Modbus ASCII. Je důležité poznamenat, že všechny jednotky na téže sběrnici musí komunikovat ve stejném vysílacím režimu. (Ronešová, 2005).

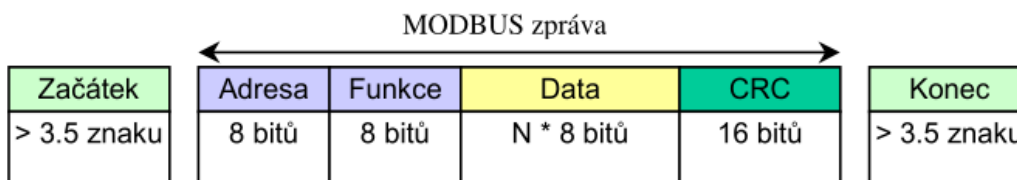


Obrázek 1.38 – Tvar zprávy Modbus RTU a Modbus ASCII (Ronešová, 2005)

Modbus RTU

V RTU režimu se každý byte zprávy skládá z osmi bitů, které reprezentují dva 4bitové hexadecimální znaky. Vysílání takové zprávy musí probíhat nepřerušovaně a mezerou mezi jednotlivými znaky nesmí být delší než doba trvání 1,5 znaku. Začátek a konec každé zprávy je rozpoznán díky přerušení na sběrnici, které trvá déle než 3,5 znaku. Struktura RTU rámce je zobrazena na obrázku 1.39. Pole CRC v Modbus zprávě je polynom, jehož rovnice odpovídá vzorci 1.1. Pole CRC v Modbus zprávě je 16bitový polynom, jehož rovnice odpovídá vzorci 1.1. (Ronešová, 2005).

$$\text{Výpočet CRC} = x^{16} + x^{15} + x^2 + 1. \quad (1.1)$$



Obrázek 1.39 – Struktura rámce Modbus RTU (Ronešová, 2005)

V této kapitole byl obecně popsán komunikační protokol Modbus a jeho podtyp Modbus Seriál Line, resp. Modbus na sériové lince. V následující kapitole bude popsán komunikační protokol EtherCAT, který je založený na Ethernetu, ale využívá prvky vlastní rodině komunikačních protokolů z rodiny Fieldbus. (Ronešová, 2005)

Komunikační protokol EtherCAT

EtherCAT, vyvinutý společností Beckhoff Automation v dubnu roku 2003, je vysokovýkonný Průmyslový Ethernetový protokol určený pro potřeby rychlé komunikace s nízkou latencí, který byl vyvinut pro potřeby průmyslové automatizace. Díky tomu, že obsahuje prvky z rodiny komunikačních protokolů Fieldbus a jedná se o kombinaci těchto dvou

rodin, je tak často nazýván jako „Ethernet Fieldbus“. EtherCAT plní mezinárodní elektronický standard (IEC 61158), který se vyznačuje nejen otevřeností, ale také stabilitou. Do dnešního dne nebyly specifikace nikdy změněny, pouze byla rozšířena kompatibilita. To znamená, že současná zařízení lze bez problémů používat ve stávajících systémech a není nutné uvažovat o různých verzích. Rozšíření zahrnuje Safety over EtherCAT pro bezpečnost strojů a personálu ve stejné síti a EtherCAT P pro komunikaci a napájecí napětí (2 x 24 VDC) na stejném čtyřvodičovém kabelu. Potřeba pro vývoj EtherCATu spočívala v limitech sítí Ethernetu, tak Fieldbusu. U Ethernetu zejména proto, že se nehodí pro řízení v reálném čase, jinými slovy není deterministický. U Fieldbusu z důvodu jeho limitů v počtu zařízení v síti i kvůli přenosovým rychlostem. (EtherCAT, nedatováno).

Omezení Průmyslového Ethernetu

Největší omezení Průmyslového Ethernetu, resp. Ethernet/IP spočívá v tom, že se nehodí pro řízení v reálném čase, jinými slovy není deterministický a má vyšší odezvy, než jsou požadovány v řídicích aplikacích. Toto omezení však platí pouze na Průmyslový Ethernet v jeho základní formě. (EtherCAT, nedatováno).

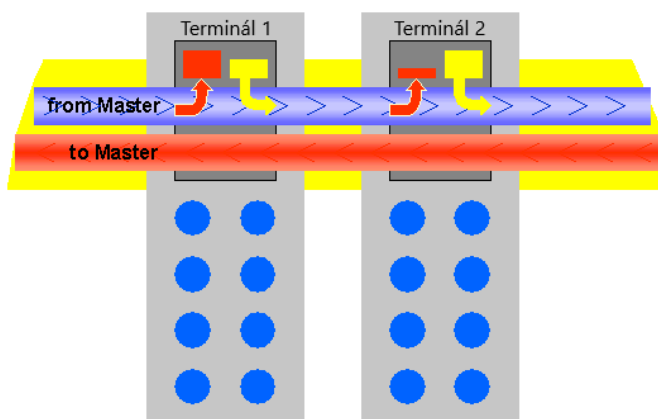
Existuje mnoho různých přístupů, které zajišťují pro Průmyslový Ethernet schopnost pracovat v reálném čase. Příkladem je přístupová procedura CSMA/CD, což je vlastně deaktivování vyšších vrstev protokolu a nahrazení procedurou časových úseků nebo dotazováním. Další možností je použití speciálních přepínačů, které distribuují pakety Průmyslového Ethernetu přesně řízeným časováním. Zatímco tato řešení mohou být schopna přenášet datové pakety více či méně rychle a přesně do připojených uzlů Průmyslového Ethernetu, časy potřebné pro přesměrování na výstupy nebo řídicí jednotky pohonů a pro čtení vstupních dat silně závisejí na implementaci. Zejména u modulárních systémů je obvykle nutná také dílčí sběrnice, která je sice synchronizovaná, nicméně vždy přidává do komunikace drobná zpoždění, kterým se nelze vyhnout. (EtherCAT, nedatováno).

Pokud se pro každé zařízení používají jednotlivé Ethernetové rámce, je využitelná rychlost přenosu dat v zásadě velmi nízká. Nejkratší Ethernetový rámec je dlouhý 84 bajtů včetně mezipaketové mezery (inter-packet gap). Pokud například akční člen cyklicky odesílá 4 bajty informací o aktuální hodnotě stavu a přijímá 4 bajty informací o nastavené hodnotě, je při 100% zatížení sběrnice, tj. při nekonečně krátké odezvě akčního členu, dosaženo využitelné rychlosti přenosu dat pouze $4/84 = 4,7\%$. Při průměrné době odezvy $10\ \mu\text{s}$ klesne rychlost na $1,9\%$. Tato omezení platí pro všechny přístupy k Ethernetu v reálném čase, které posílají rámec

Ethernetu každému zařízení (nebo očekávají rámec od každého zařízení) bez ohledu na protokoly použité v rámci Ethernetu. (EtherCAT, nedatováno).

Řešení protokolu EtherCAT

Výše zmíněné omezení, která jsou vlastní jiným řešením Ethernetu, jsou překonávána právě technologií EtherCAT. Paket již není přijímán, následně interpretován a kopírován jako procesní data při každém připojení. Nově vyvinutá jednotka, resp. mikročip EtherCATu, nazvaná FMMU se nachází v každém terminálu (rozšiřující modul) a čte data, která jsou mu adresována, zatímco telegram pokračuje přes zařízení. Podobně se pak vkládají vstupní data, zatímco telegram prochází. Telegramy se zpožďují pouze o několik nanosekund. Na obrázku 1.40 je graficky znázorněné, jak je telegram dynamicky zpracováván průchodem přes terminály na sběrnici EtherCAT. (EtherCAT, nedatováno).



Obrázek 1.40 – Dynamické zpracování telegramu (EtherCAT, nedatováno)

Vzhledem k tomu, že Ethernetový rámec přistupuje k datům mnoha zařízení, jak při odesílání, tak při příjmu, dosahuje efektivní využití přenosové rychlosti přibližně 80 %.

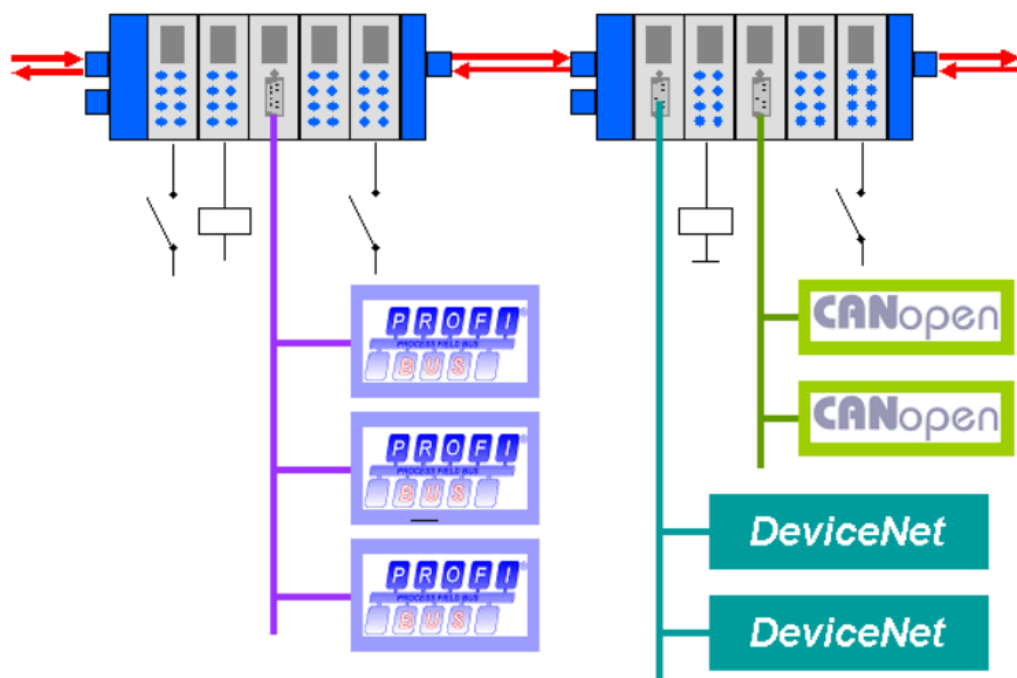
Využívají se plně vlastnosti 100BaseTx s plným duplexem, což umožňuje dosažení efektivní přenosové rychlosti větší než $100 \text{ Mbit} \cdot \text{s}^{-1}$ (až 80 % z celkových $2 \times 100 \text{ Mbit} \cdot \text{s}^{-1}$).

(EtherCAT, nedatováno).

Dle IEEE 802.3 zůstává protokol Ethernet zachován až ke koncovým terminálům a eliminuje se tím potřeba jakýchkoli mezičlánkových sběrnic. V EtherCATovém spoji (coupler) se fyzický mění pouze přenos z krouceného páru nebo optického vlákna na sběrnici EtherCAT (tzv. E-bus), aby bylo vyhověno požadavkům elektronické svorkovnice. Na straně zařízení rozšiřuje FMMU technologii ovladačem TwinCAT Y pro Ethernet. Tento ovladač se

transparentně integruje do systému, který působí jako síťový ovladač kompatibilní s operačním systémem, a také jako sběrnice karta TwinCAT. Na straně vysílání existuje interní systém prioritizace a vyrovnávací paměť, který vždy vyhledá volný přenosový kanál pro Ethernetové rámce z reálného časového systému, které mohou čekat ve frontě. Ethernetové rámce z operačního systému jsou pak přeneseny v mezerách následně, když je k dispozici čas navíc.

Díky tomu, že se stále jedná o sběrnici Ethernetu s prvky Fieldbusu, je možné integrovat pomocí sběrnice EtherCAT další zařízení z jiných Fieldbusových sběrnic jako jsou Profibus, CANopen, DeviceNet a další. Na obrázku 1.41 je graficky znázorněná možná topologie decentralizace řízení na sběrnici EtherCAT s implementací dalších komunikačních sběrnic. (EtherCAT, nedatováno).



Obrázek 1.41 – Decentralizace pomocí EtherCAT sběrnice (EtherCAT, nedatováno)

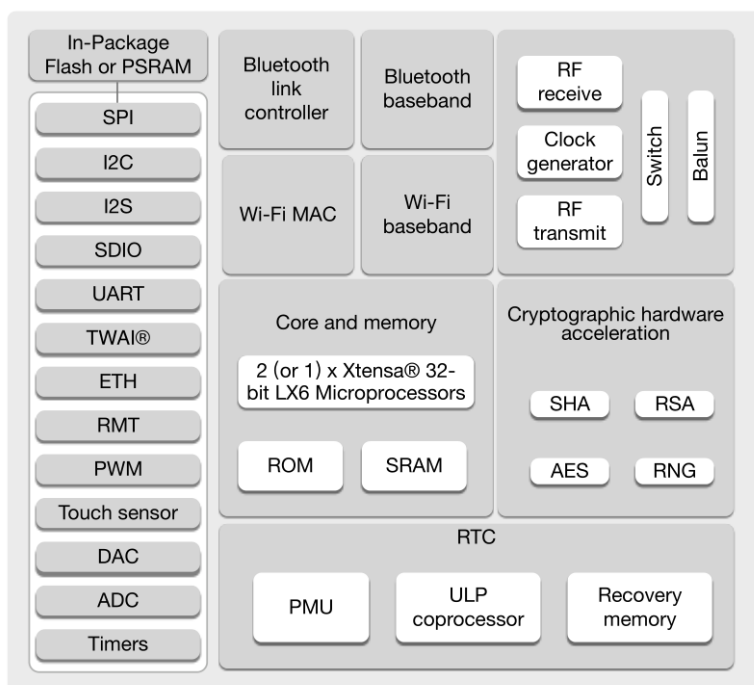
Shrnutí protokolu EtherCAT

EtherCAT přináší vynikající výkon, snadnou instalaci a kompatibilitu s jinými protokoly. Tento protokol definuje nové standardy v oblasti komunikace, kde tradiční sběrnice Fieldbus systémy dosahují svých omezení. Umožňuje přenos přes 1 000 vstupně-výstupních signálů v časovém rozmezí 30 μ s, nabízí volbu mezi krouceným párem nebo optickým vláknem a vzhledem ke své integraci s technologiemi Ethernet dosahuje přenosových rychlostí až 100 $\text{Mbit} \cdot \text{s}^{-1}$. Na rozdíl od jiných Ethernetových řešení, EtherCAT nepotřebuje

speciální hardware v řídicím systému, ale může pracovat s cenově přijatelnými standardními ethernetovými kartami. EtherCAT transformuje koncept Ethernetu až na úroveň koncového zařízení a činí jej technicky realizovatelným a ekonomicky efektivním. EtherCAT je považován za ideální sběrnici pro průmyslová PC nebo jako doplněk k menším řídicím systémům a očekává se, že najde široké uplatnění v různých aplikacích. (EtherCAT, nedatováno).

1.3 MIKROPOČÍTAČ ESP32-WROOVER-IE

Mikropočítače z rodiny ESP32 jsou 32bitové procesory typu SoC, které se vyznačují nízkou spotřebou, integrovaným Wi-Fi a Bluetooth rozhraním, anténními přepínači, mnoha podporovanými komunikačními sběrnici, několika typy pamětí, kryptografickými funkcemi, podporou dotykových vývodů a v neposlední řadě také ultra nízko výkonovým koprocesorem. Tyto mikropočítače využívají procesory Tensilica Xtensa LX6 ve dvoujádrové i jednojádrové variantě. ESP32 vytvořila a vyvinula v roce 2016 čínská společnost Espressif Systems, která má sídlo v Šanghaji a vyrábí jej společnost TSMC na svém 40 nm procesu. Mikropočítače ESP32 jsou nástupci mikropočítače ESP8266. Na obrázku 1.42 je znázorněné funkční blokové schéma. (Espressif Systems, 2023; ESPBoards, 2023).



Obrázek 1.42 – Funkční blokové schéma mikropočítače ESP32 (Espressif Systems, 2023)

Mikropočítačů z rodiny ESP32 je mnoho typů, v této diplomové práci se budeme zabývat zejména typem ESP32-WROOVER-IE.

Vlastnosti mikropočítače ESP32-WROOVER-IE

Wi-Fi

V mikropočítači ESP8266 také byla integrovaná podpora Wi-Fi rozhraní, ale v jeho nástupci tedy v mikropočítači ESP32 byla ještě vylepšena podporou standardů 802.11b/g/n a na standardu 802.11n na 2.4 GHz při přenosové rychlosti až $150 \text{ Mbit} \cdot \text{s}^{-1}$. Implementací WMM (Wi-Fi Multimedia) došlo k možnosti prioritizaci síťového provozu. Agregace rámce TX/RX A-MPDU a RX A-MSDU byla zařazena zvýšená efektivita přenosu. Wi-Fi na ESP32 dovede také pracovat v režimu AP s možností až čtyř virtuálních Wi-Fi rozhraní. (Espressif Systems, 2023; ESPBoards, 2023).

Bluetooth

Mikropočítač ESP32 byl rovněž vybaven rozšířenými schopnostmi Bluetooth, které jsou kompatibilní s normou Bluetooth v4.2 BR/EDR a specifikacemi Bluetooth LE. Byly také začleněny funkce pro synchronní spojení, audio kodeky a podpora pro vytváření malých lokálních sítí Bluetooth. ESP32 byl navržen tak, aby umožnil více spojení v klasickém Bluetooth i v Bluetooth LE a nabídl schopnost současně vysílat a skenovat Bluetooth signály. (Espressif Systems, 2023; ESPBoards, 2023).

Procesor a paměti

V mikropočítači ESP32 je integrován Xtensa jednojádrový nebo dvoujádrový 32bitový mikroprocesor LX6 s jádrem nebo jádry pracující na frekvenci 240 MHz. Co se týče paměti, bylo integrováno 448 KB ROM, 520 KB SRAM a dalších 16 KB SRAM v RTC. QSPI bylo navrženo tak, aby podporovalo více FLASH a SRAM čipů. (Espressif Systems, 2023; ESPBoards, 2023).

Hodiny reálného času a časovače

V mikropočítači ESP32 je integrován interní oscilátor s frekvencí 8 MHz a také interní RC oscilátor, oba s možností kalibrace. Externí krystalový oscilátor, který podporuje frekvenční rozsah od 2 MHz do 60 MHz, je také součástí tohoto mikropočítače, přičemž je třeba

poznámenat, že pro funkce Wi-Fi a Bluetooth je k dispozici pouze 40 MHz. Pro hodiny reálného času (RTC) byl v ESP32 osazen ještě další externí krystalový oscilátor s frekvencí 32 kHz. RTC krystalový oscilátor má také možnost kalibrace. Pokud jde o časovače, ESP32 obsahuje dvě skupiny časovačů. Každá z těchto skupin zahrnuje 2 časovače o 64 bitech a jeden hlavní watchdog časovač. Kromě toho je součástí ESP32 také časování pro reálný čas s watchdog časovačem, který monitoruje a obnovuje systém v případě, že dojde k chybám nebo nefunkčním stavům. (Espressif Systems, 2023; ESPBoards, 2023).

Pokročilé periferní rozhraní

V mikropočítači ESP32 jsou integrovány různé periferní rozhraní. Celkem 34 programovatelných GPIO vývodů, z nichž 5 je označeno jako tzv. strapping, což znamená, že při startu systému ESP32 musí být jejich stavy jasně definovány. Tato definice však platí pouze v času restartu ESP32, resp. ihned po přivedení napájení na napájecí vývody. Dále je integrován 12bitový SAR analogově-digitální převodník s 18 kanály a dva 8bitové digitálně-analogové převodníky. Pro dotykové funkce bylo začleněno 10 dotykových vývodů. Součástí ESP32 jsou čtyři SPI rozhraní, dvě I2S, dvě I2C a tři UART rozhraní. Hostitelské rozhraní s podporou SD, eMMC a SDIO bylo začleněno společně s jedním slave rozhraním podporujícím SDIO a SPI. Ethernetové MAC rozhraní s podporou IEEE 1588 bylo přidáno pro pokročilou síťovou komunikaci. TWAI®, kompatibilní s ISO 11898-1, byl integrován společně s RMT pro vysílání a příjem. Dále ESP32 disponuje podporou pulzní šířkové modulace pro krokové motory a pulzní šířkovou modulaci pro LED až s 16 kanály. (Espressif Systems, 2023; ESPBoards, 2023).

Správa napájení

V oblasti správy napájení bylo u ESP32 implementováno přesné řízení spotřeby energie prostřednictvím volitelné frekvence procesoru, střídání režimů provozu Wi-Fi a individuálního řízení vnitřních komponentů. Bylo navrženo pět napájecích režimů, které jsou přehledně uvedeny v tabulce 1.7. Pro ještě účinnější správu energie byly integrovány ultra nízko příkonný (ULP) koprocessor. Napájecí napětí mikropočítače ESP32 je 3,3 V DC. (Espressif Systems, 2023; ESPBoards, 2023).

Tabulka 1.7 – Popis napájecích režimů ESP32-WROOVER-IE

Režim	Průměrný proudový odběr	Stručný popis režimu
Aktivní	95 až 240 mA	Vše aktivní
Modem sleep	2 až 50 mA	Všechny modemy neaktivní (Wi-Fi, Bluetooth, Rádio)
Light sleep	0,8 mA	Aktivní pouze ULP koprocessor, periferie a RTC
Deep sleep	10 až 15 μ A	Aktivní pouze ULP koprocessor a RTC
Hibernace	5 μ A	Vše neaktivní, pouze RTC aktivní

Bezpečnost

V oblasti bezpečnosti byly u mikropočítače ESP32 implementovány pokročilé funkce a nástroje pro ochranu dat a zabezpečení aplikací. Byl zaveden bezpečný zaváděcí proces, který zajišťuje, že systém nahraje pouze ověřený a důvěryhodný kód. Kromě toho bylo přidáno šifrování FLASH paměti, což chrání uložená data před neoprávněným přístupem. Pro trvalé uložení důvěrných dat byl začleněn 1024bitový OTP (One-Time Programmable) čip, z něhož až 768 bitů je k dispozici pro potřeby zákazníků. Dále byla začleněna hardwarová akcelerace pro kryptografické operace, což zvyšuje rychlost a zabezpečení při provádění těchto funkcí. Mezi podporované kryptografické algoritmy patří AES pro symetrické šifrování, hashovací funkce SHA-2 pro generování otisků prstů, RSA pro asymetrické šifrování a digitální podpisy, ECC pro kryptografii na eliptických křivkách, a nakonec byl integrován generátor náhodných čísel (RNG) pro generování kryptograficky silných náhodných čísel pro bezpečnostní aplikace. (Espressif Systems, 2023; ESPBoards, 2023).

Pouzdro

Mikropočítač ESP32 je zapouzdřený v plechovém šasi, přičemž varianta ESP32-WROOVER-IE ještě navíc disponuje externí MHF I konektorem, které vyrábí firma I-PEX. Tento konektor slouží pro rozhraní Wi-Fi. Na obrázku 1.43 je znázorněné ESP32-WROOVER-IE. (Espressif Systems, 2023; ESPBoards, 2023).

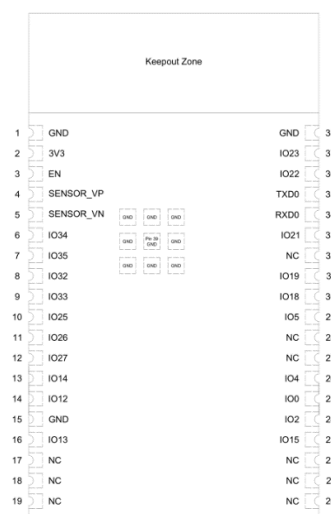


Obrázek 1.43 – ESP32-WROOVER-IE

Pod pouzdrém se nacházejí další podpůrné obvody, resp. napájecí obvod a paměťové moduly, přičemž samotný mikročip se nazývá ESP32-D0WD-V3 a platí pro něj vše výše uvedené. Toto plechové zapouzdrění je zde pouze kvůli lepší mechanická ochraně a kvůli interní anténě. (Espressif Systems, 2023; ESPBoards, 2023).

Popis vývodů

ESP32 má celkem 38 vývodů, kde většina z nich implementuje více funkcí. Vývody mikropočítače ESP32 se dají rozdělit na vývody pro všeobecné účely (GPIO), analogové vývody, napájecí vývody a komunikační vývody. Na obrázku 1.44 je graficky znázorněné pouzdro ESP32-WROOVER-IE s označením vývodů. (Espressif Systems, 2023; ESPBoards, 2023).



Obrázek 1.44 – Vývody ESP32-WROOVER-IE (Espressif Systems, 2023)

V tabulce 1.5 jsou přehledně tyto vývody vypsány a v tabulce 1.6 je uvedena popis jednotlivých typů vývodů.

Tabulka 1.8 – Výpis vývodů ESP32-WROOVER-IE (Espressif Systems, 2023)

Název vývodu	Číslo vývodu	Typ vývodu	Funkce vývodu
GND	1	N	Nulový potenciál (zem)
3V3	2	N	Napájecí napětí 3,3 V DC
EN	3	V	ENABLE
SENSOR_VP	4	V	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	V	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	V	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	V	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	V/V	GPIO32, XTAL_32K_P (32.768 kHz vstup krystalového oscilátoru), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	V/V	GPIO33, XTAL_32K_N (32.768 kHz výstup krystalového oscilátoru), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	V/V	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	V/V	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	V/V	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	V/V	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	V/V	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	N	Nulový potenciál (zem)
IO13	16	V/V	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
Nepřipojen	17–22	-	-
IO15	23	V/V	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	V/V	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	V/V	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	V/V	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
Nepřipojen	27-28	-	-
IO5	29	V/V	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	V/V	GPIO18, VSPICLK, HS1_DATA7
IO19	31	V/V	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
Nepřipojen	32	-	-
IO21	33	V/V	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	V/V	GPIO3, U0RXD, CLK_OUT2

Tabulka 1.8 - Výpis vývodů ESP32-WROOVER-IE, pokračování

TXD0	35	V/V	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	V/V	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	V/V	GPIO23, VSPID, HS1_STROBE
GND	38	N	Nulový potenciál (zem)

Typy vývodů, které mají označení N, jsou napájecí vývody, dále V a V/V, kde V označuje pouze vstupní vývod a V/V označuje vstupně/výstupní vývod, tedy univerzální vývod.

Tabulka 1.9 – Označení podtypů vývodů ESP32-WROOVER-IE (Espressif Systems, 2023)

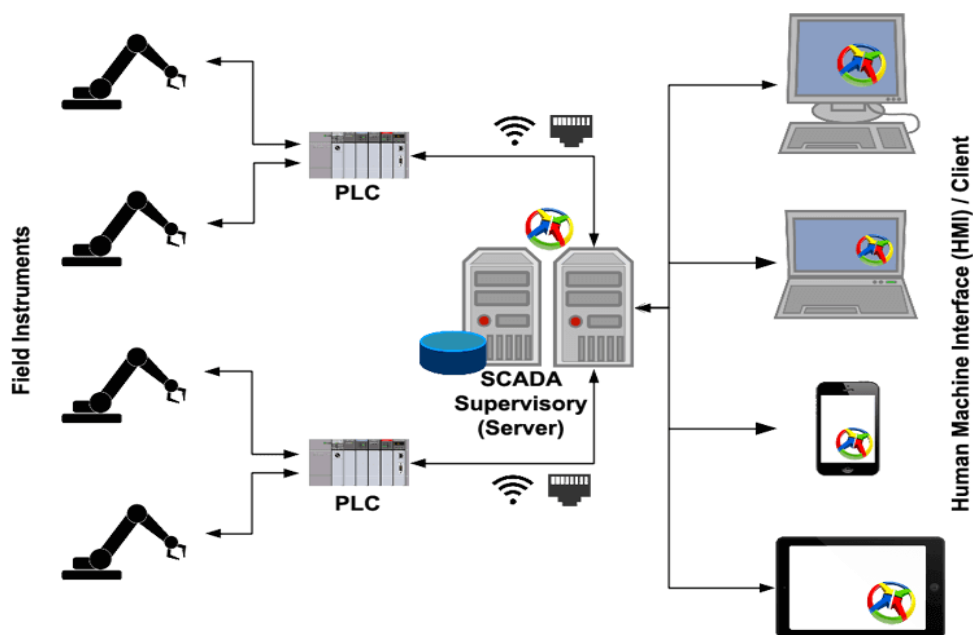
Označení podtypu vývodu	Popis vývodu
GPIO	Univerzální vstupně/výstupní vývod
ADC	Vývod analogově/digitálního převodníku
XTAL	Vývod na krystal
DAC	Vývod digitálně/analogového převodníku
RTC	Vývod pro hodiny reálného času
EMAC	Vývod pro rozhraní Ethernet
MTMS	Strapping vývod
MTDI	Strapping vývod
HSPI	Vysokorychlostní SPI
TOUCH	Vývod dotykového senzoru

Mikropočítač ESP32-WROOVER-IE je v dnešní době velice dostupný a za poměrně nízkou cenu 72 Kč za jeden čip. Tento mikropočítač se jeví jako ideální volba pro tvorbu elektronických aplikací, jak je třeba akviziční jednotka pro programovatelný automat.

1.4 KOMERČNĚ VYRÁBĚNÉ AKVIZIČNÍ A KONTROLNÍ JEDNOTKY

Akviziční a kontrolní jednotky jako takové nejsou na trhu dostupné. Vždy se totiž jedná o akviziční a kontrolní systém, což se v průmyslu objevuje jako tzv. SCADA systém, kdy akronym SCADA znamená *Supervisory control and data acquisition*. Tento SCADA systém je v dnešní době software, který se napojí na síť programovatelného logického automatu a tento systém dovede kontrolovat, ukládat a měnit parametry sledovaného řídicího systému. V podstatě se tedy jedná o PLC s komunikací s řídicím počítačem a grafickým znázorněním procesu, přičemž vlastností SCADA systému je možnost dálkového ovládní daného procesu z více míst. SCADA systém často zahrnuje také grafické na dotykových displejích, které se

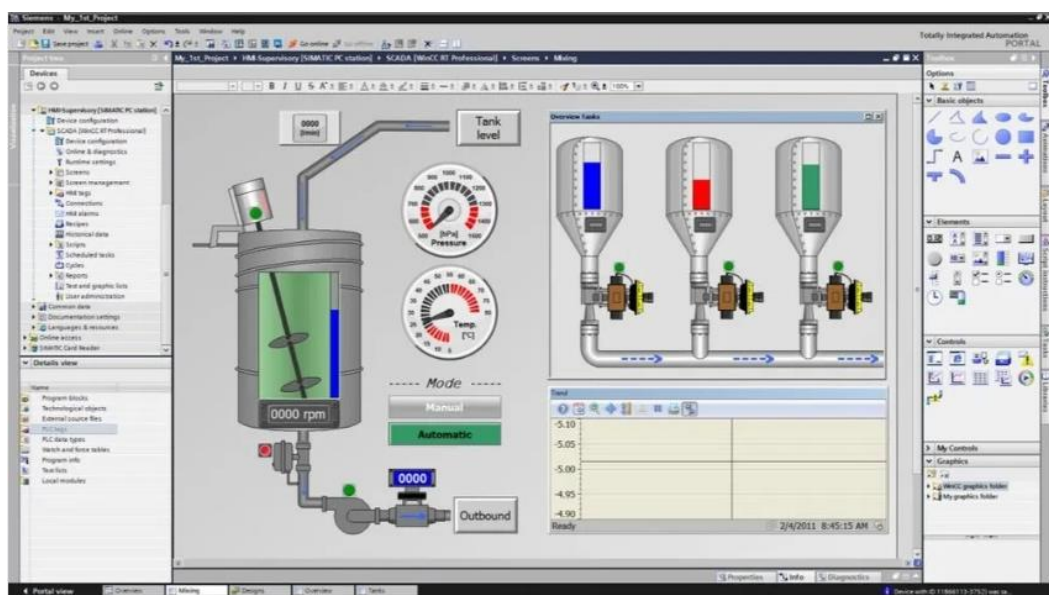
nazývají HMI panely. SCADA systémy se používají zejména u dálkových ovládní všech druhů distribučních sítí, jako je monitorování a ovládní různých technologických procesů v odlehlých oblastech jako jsou vodovodních zdroje, elektrárny a plynovody. Blokové schéma typického SCADA systému je znázorněné na obrázku 1.45. Z obrázku je patrné, že řídicí proces, v tomto případě akční člen manipulátoru, je řízen pomocí PLC a díky SCADA softwaru je zpracován a dále zobrazen nebo i ovládn přes různá klientská zařízení, jako jsou počítače, mobilní telefony, kterým se v tomto případě říká operátorské HMI panely. SCADA systém se tedy dá zjednodušit do řízení členů (akční členy, senzory, tlačítka nebo signalizace), které jsou připojeny k PLC, který tyto členy ovládá. PLC je následně připojeno do sítě a v síti je spuštěna služba, nejčastěji programem, která komunikuje s okolním světem. Na stejné síti jsou poté připojeny klientská zařízení, kterými se řízené členy vzdáleně ovládají, resp. se dají rovněž jejich stavy nebo hodnoty ukládat a tím dochází k akvizici (sběru) těchto hodnot nebo stavů. V následujících kapitolách bude proveden popis dvou vybraných SCADA řešení. Prvním z nich bude řešení firmy Siemens, která vydala svoji první verzi SCADA systému v roce 1996. Tento SCADA systém se nazývá Simatic WinCC a jeho aktuální verze je 7.5. Dalším vybraným SCADA systémem je již zmíněný software TwinCAT od německé firmy Beckhoff Automation, který podporuje všechny náležitosti, které SCADA systém má splňovat. (Kolbaba, 2004; Beckhoff, nedatováno)



Obrázek 1.45 – Architektura SCADA systému (Ecava Intergraxor, 2020)

1.4.1 Siemens Simatic WinCC

Softwarová aplikace na Windows s názvem Simatic WinCC je SCADA systém, na který lze napojit PLC od firmy Siemens z řady S7 i řady S5. Tento software umožňuje vizualizaci, ovládání, a telemetrii technologických procesů. Grafické rozhraní aplikace Simatic WinCC je znázorněné na obrázku 1.46. Toto řešení bylo navrženo tak, aby bylo flexibilní pro různé aplikace, ať už malé nebo rozsáhlé distribuované systémy, čímž je zajištěna vyšší efektivita a bezpečnost v průmyslových operacích. Bohužel je systém Simatic WinCC placený, a proto nebylo možné tento SCADA systém lépe popsat. (Kolbaba, 2004)



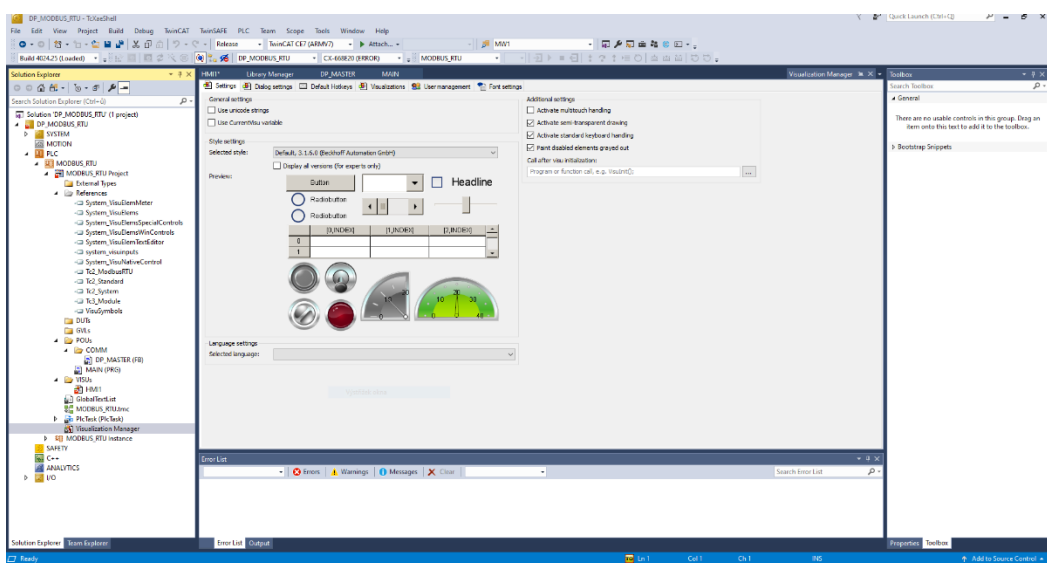
Obrázek 1.46 – WinCC SCADA systém firmy Siemens (Mesidas, nedatováno)

1.4.2 Beckhoff TwinCAT 3

TwinCAT 3 je integrované automatizačním řešením, které kombinuje PLC, NC a CNC funkce v jediném software. Běžící na Windows operačním systému s vlastnostmi v reálném čase, TwinCAT 3 umožňuje vývojářům programování PLC v souladu s normou IEC 61131-3. Nabízí podporu pro různé programovací jazyky a je navržen tak, aby spolupracoval s průmyslovými počítači Beckhoff v rámci jejich konceptu "PC Control". Díky podpoře mnoha průmyslových komunikačních protokolů, včetně vlastního a již v této práci zmíněného EtherCAT, je TwinCAT ideálním řešením pro různé průmyslové aplikace. Když je kombinován s TwinCAT HMI, nabízí kompletní SCADA funkcionalitu, umožňující vytváření pokročilých vizualizací a uživatelských rozhraní pro monitorování a ovládání průmyslových procesů. (Beckhoff, nedatováno).

TwinCAT HMI

TwinCAT HMI je nástroj určený k vytváření HMI zobrazení. Pro konfiguraci se využívá grafický WYSIWYG editor, což znamená, že není vyžadováno programování. Ovládací prvky z nástrojové palety mohou být na uživatelském rozhraní organizovány a spojeny s proměnnými v reálném čase, např. z PLC nebo z modulu C++. Uživatelské ovládací prvky lze využít ke grafickému vytváření a konfiguraci specifických ovládacích prvků. Vývojáři JavaScriptu mohou vytvářet vlastní ovládací prvky. Během vývoje lze funkcionalitu testovat s živými daty z reálného časového systému. Modifikace jsou možné i v tomto živém režimu. Logika HMI může být implementována buď na straně klienta v JavaScriptu nebo jako rozšíření serveru. Rozšíření serveru umožňuje vytvářet rozšíření v C#. Protože je HMI založeno na webu a je responzivní, založené na HTML5 a JavaScriptu, tím pádem se rozlišení a orientace automaticky přizpůsobí. To umožňuje použít stejnou stránku pro různé velikosti displejů, poměry stran a orientace. TwinCAT HMI se nastavuje pomocí nástroje Visualisation přímo v prostředí TwinCAT 3. Na obrázku 1.47 je ilustrovaná základní obrazovka nástroje Visualisation Manager, který slouží k přednastavení jako jsou možnosti použitých písem, správa uživatelů, klávesové zkratky a další. Další funkcionalitou je vytvoření tzv. cíle vizualizace, kdy TwinCAT HMI podporuje nejen dotykové obrazovky, ale také možnosti zadávání přes virtuální klávesnici. Co se týče samotných SCADA funkcionalit, cíl vizualizace může být i webový a to znamená, že při správně nasměrované síti, toto zobrazení můžeme sledovat klidně z druhého konce světa, za předpokladu, že je možnost připojení k internetu. (Beckhoff, nedatováno).

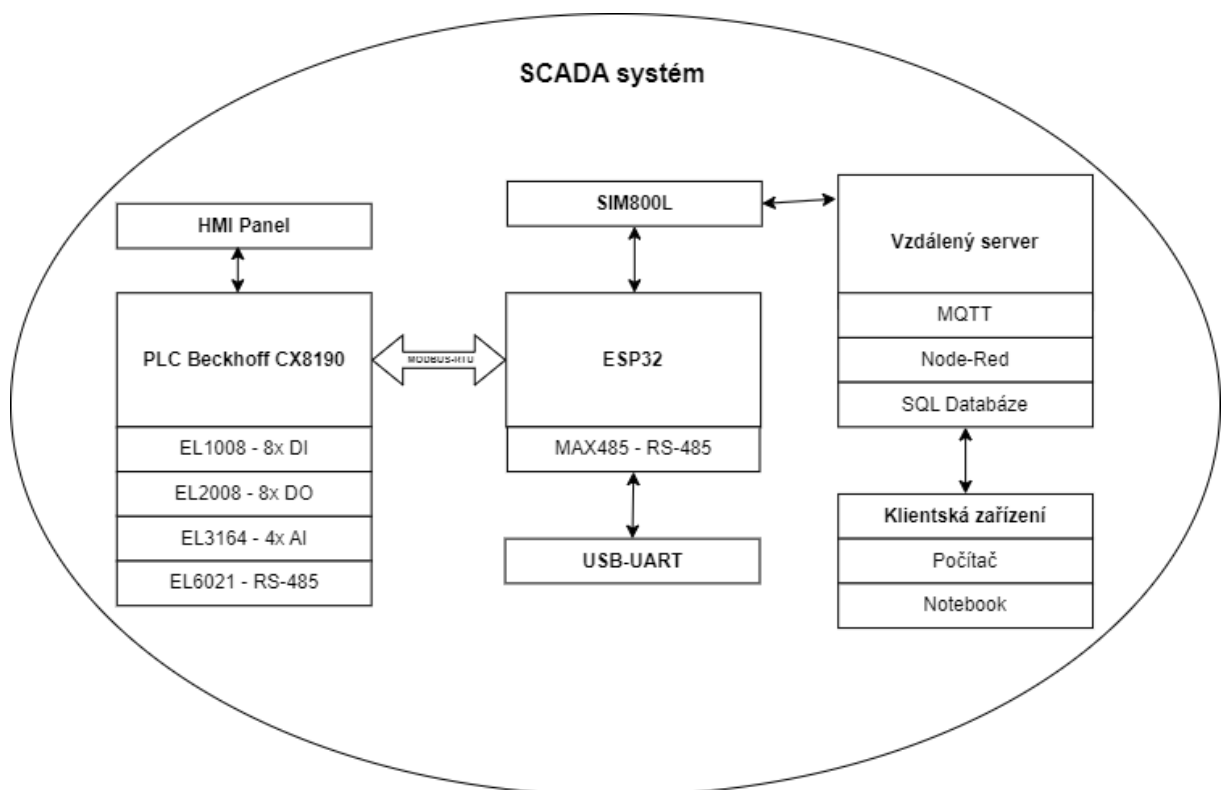


Obrázek 1.47 – Nástroj Visualisation v prostředí TwinCAT 3

2 PRAKTICKÁ ČÁST

2.1 NÁVRH A REALIZACE AKVIZIČNÍ A KONTROLNÍ JEDNOTKY

Cílem praktické části je návrh a realizace akviziční a kontrolní jednotky, která je realizována jako deska plošného spoje s napájecími obvody, komunikačních rozhraním RS-485 a připojením na internet skrze externí modul SIM800L, který se připojuje pomocí protokolu MQTT na vzdálený server. Tato jednotka je však pouze jedna součást celku, který má tvořit SCADA systém. Tedy dohledový a ovládací systém pro programovatelný logický automat. Blokové schéma celého SCADA systému je znázorněné na obrázku 2.1.



Obrázek 2.1 – SCADA systém

2.1.1 Popis funkce realizovaného SCADA systému

Akviziční a kontrolní jednotka samotná je založená na mikro počítači ESP32-WROOVER-IE, který je pro tyto účely plně dostačující. Jednotka je propojena komunikací přes sběrnici RS-485 s programovatelným automatem Beckhoff CX8190, komunikační protokol mezi jednotkou a PLC je Modbus-RTU. K jednotce je dále připojen komunikační modul SIM800L, na který se připojuje anténa do konektoru typu MHF I. Anténa slouží pro příjem

GPRS signálu, díky kterému se připojuje na vzdálený MQTT server. Komunikační modul se připojuje pomocí 5 vývodového konektoru typu DUPONT a není součástí zmíněné desky plošného spoje. Programovací konektor má rovněž 5 vývodů, do kterého se připojuje USB-UART převodník, který také není součástí desky plošného spoje jednotky, a to z důvodu, že se nepředpokládá, že by se jednotka programovala opětovně. Vzdálený server aktuálně běží na Synology DiskStation 218+ ve verzi 6, která má možnost instalaci uživatelských balíčků, jako je právě MQTT server Mosquitto. Do DiskStation je možné rovněž nahrát software Docker, který je známý jako virtualizační nástroj a je možné v něm spouštět aplikace nebo podpůrné servery jako je Node-Red, který slouží jako aplikační rozhraní pro komunikaci MQTT a také pro akvizici dat na SQL server. SQL server typu Maria-db je spuštěn také na Synology DiskStation. Propojením všech těchto součástí do sebe je možné hovořit o systému, který plní definici SCADA systému. V dalších částech praktické části budou podrobně popsány kroky, které vedly k uvedení toho SCADA systému do provozu. Kroky se dají rozdělit do konfigurace PLC a vstupně výstupního panelu Weintek, návrh a realizace jednotky, instalace a nastavení serverů pro MQTT a pro Node-Red.

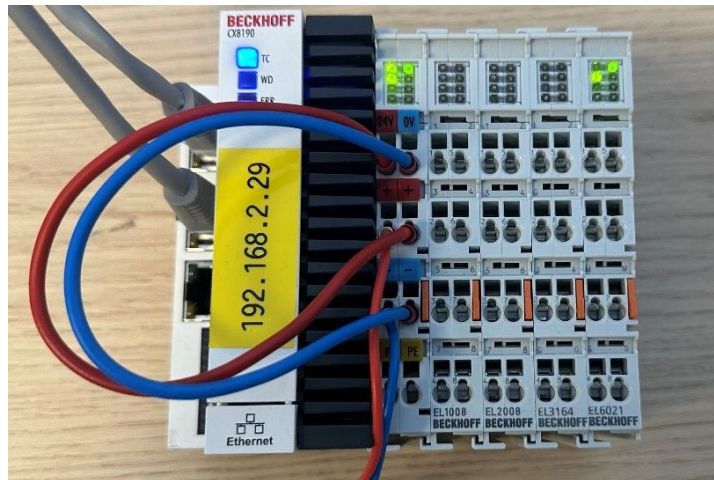
2.2 KONFIGURACE PLC BECKHOFF CX8190

Konfiguraci samotného PLC je možné rozdělit do hardwarové a softwarové konfigurace. Hardwarová konfigurace zahrnuje připojení korektního napájení pro PLC, zapojení rozšiřujících modulu v daném pořadí, které zároveň musí korespondovat se softwarovou konfigurací v programovacích prostředí. Dále je do hardwarové konfigurace zahrnuta korektní zapojení kabeláže pro další ovládané periferie jako jsou akční členy nebo senzory. PLC Beckhoff CX8190 ke správné funkci vyžaduje připojení 24 VDC pro systémové napětí a také pro silové napětí, které je rozvedeno pomocí kontaktů zahrnutých v mechanické konstrukci každého rozšiřujícího modulu. Softwarová konfigurace je poněkud složitější záležitost, a proto tomuto tématu bude věnovaná samostatná podkapitola. Správně zapojená hardwarová konfigurace je znázorněná na obrázku 2.2.

2.2.1 Softwarová konfigurace PLC Beckhoff CX8190

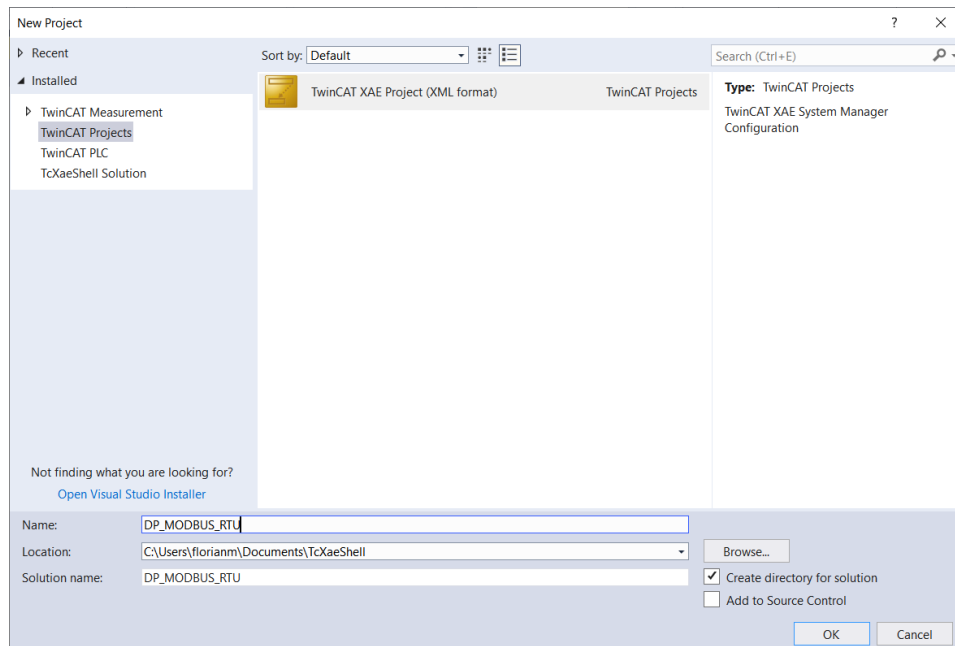
Před softwarovou konfigurací je nutné nejdříve provést hardwarovou konfiguraci, aby PLC bylo v provozuschopném stavu, což mj. zajišťuje správné napájecí napětí. Po přivedení napájení se PLC Beckhoff CX8190 spustí tzv. konfiguračním módu (*Config mode*), který je signalizován modrou barvu LED signalizace na předním panelu samotného PLC s označením

TC. Dále je veškerá konfigurace prováděná v programovacím prostředí TwinCAT 3. Tato signalizace je také znázorněná na obrázku 2.2.



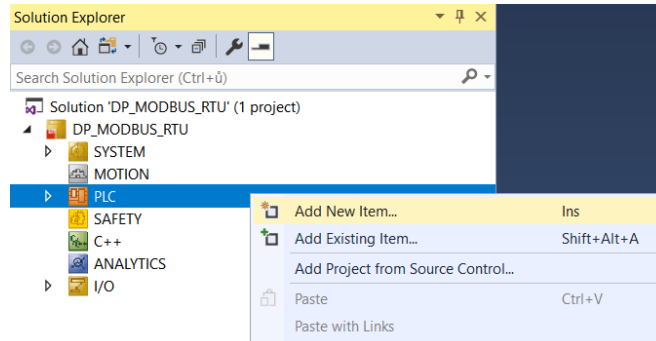
Obrázek 2.2 – HW konfigurace včetně signalizace konfiguračního režimu

Dále je nutné se připojit pomocí programovacího prostředí TwinCAT 3. V programovacím prostředí TwinCAT je nutné založit projekt. Projekt se zakládá na úvodní obrazovce pomocí *File – Add – New Project*, v tomto případě se jedná o zatím prázdný projekt s názvem *DP_MODBUS_RTU*. Ilustrace vytváření prázdného projektu je znázorněn na obrázku 2.3.



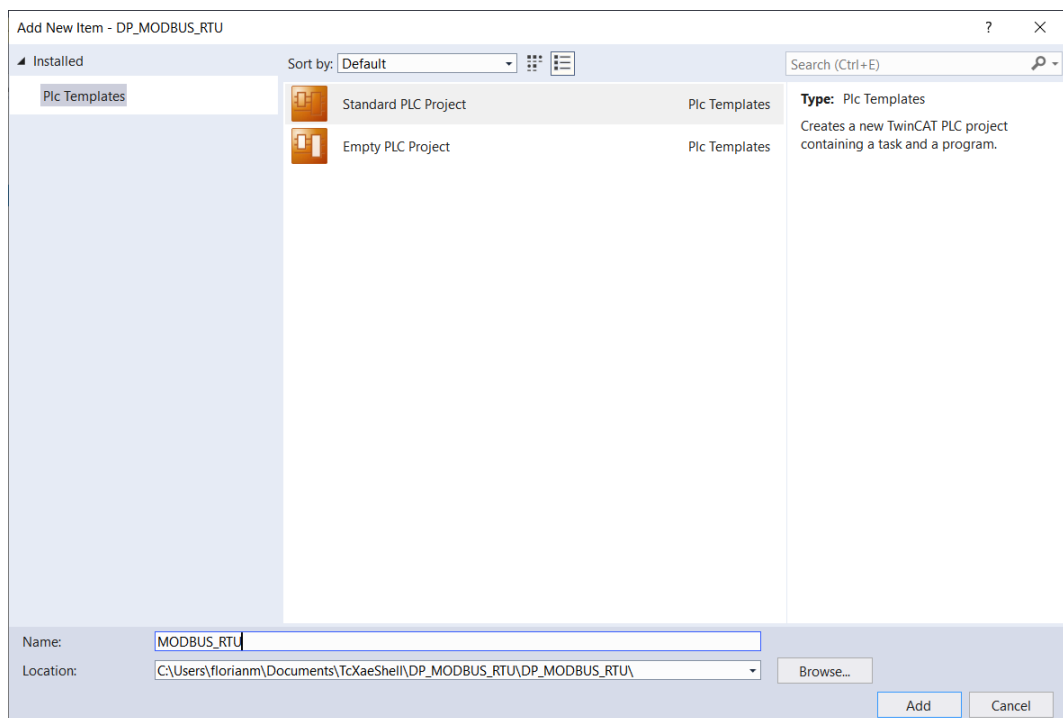
Obrázek 2.3 – Založení nového projektu s názvem DP_MODBUS_RTU

Dalším krokem je vytvoření instance projektu, zde byl zvolen standartní PLC projekt s názvem *MODBUS_RTU*. Vytváření projektu probíhá pomocí kliknutí pravého tlačítka myši na položku „PLC“ ve stromu projektu a je znázorněné na obrázku 2.4.



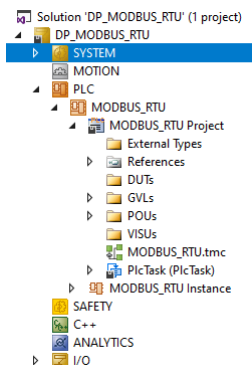
Obrázek 2.4 – Přidání instance projektu PLC pomocí pravého tlačítka myši

Dále je na obrázku 2.5 ilustrované jaký typ projektu bude vybrán pro potřeby naší implementace. Standartní PLC byl zvolen proto, že již obsahuje všechny užitečné součásti instance projektu, jako je seznam globálních proměnných (*GVLs*), organizační jednotky programu (*POUs*), volitelné typy datových jednotek (*DUTs*) a konečně, součást potřebnou pro vizualizaci (*VISUs*).



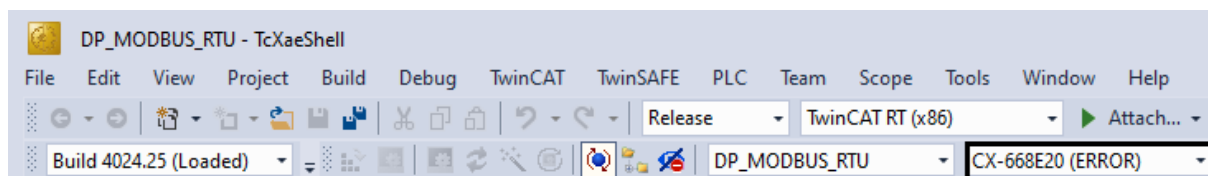
Obrázek 2.5 – Vytvoření standartního PLC projektu

V této diplomové práci budeme využívat pouze seznam globálních proměnných a organizační jednotky programu. Celý strom projektu, včetně přidáných složek a položek do *GVLs* a *POUs*, se kterými pracuje samotný program PLC, je znázorněný na obrázku 2.6.



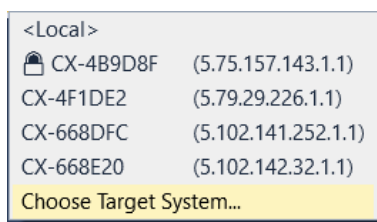
Obrázek 2.6 – Strom projektu PLC

Nyní máme založený projekt a dalším krokem je připojení k samotnému PLC. První krok připojení probíhá pomocí přidání tzv. „cesty“ (route) k PLC. Nabídka pro přidání cesty se aktivuje pomocí šipky směrem dolů u výrobního označení PLC v tomto případě CX-668E20. Těchto 6 znaků jsou zároveň poslední 3 dvojčíslí z fyzické adresy (MAC adresa) PLC, která má v tomto případě tvar „...68-8E-20“. Na obrázku 2.7 je znázorněný výřez ovládacího panelu programovacího prostředí TwinCAT 3, pomocí které se k přidání cesty dostaneme.



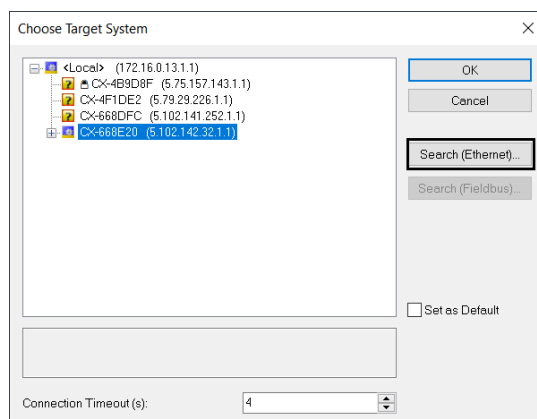
Obrázek 2.7 – Před rozkliknutím menu volby PLC

Nyní se rozbalí menu, ve kterém je potřeba kliknout na *Choose Target System*, tak jak to je ilustrované na obrázku 2.8.



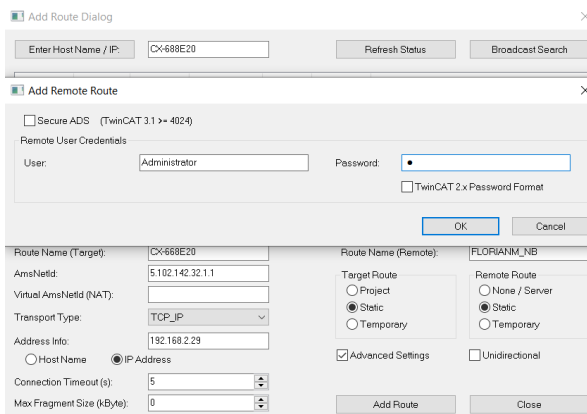
Obrázek 2.8 – Po rozkliknutí menu volby PLC

V následujícím okně vybereme volbu *Search (Ethernet)*. Tato volba je ilustrovaná na obrázku 2.9.



Obrázek 2.9 – Prohledání dostupných PLC na síti Ethernet

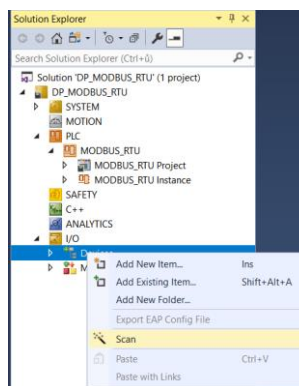
Nyní se již zobrazí dialog pro přidání cesty, kam vložíme výrobní označení PLC, tedy *CX-668E20* a potvrdíme pomocí *Enter Host Name / IP*: Aby tato funkce byla dostupná a skutečně se nám podařilo PLC objevit pomocí zadání výrobního označení, počítač, na kterém běží TwinCAT 3 a PLC, musí být na stejné podsíti. Pokud je PLC a počítač na stejné podsíti, v tabulce se objeví naše PLC. Kliknutím na *Add Route* a zadáním hesla „1“ k uživatelskému jménu „Administrator“, které je předvyplněné, se přidá cesta k PLC a tím jsme k PLC připojeni. Tento dialog pro zadání uživatele a hesla je znázorněný na obrázku 2.10.



Obrázek 2.10 – Potvrzení přidání cesty k PLC

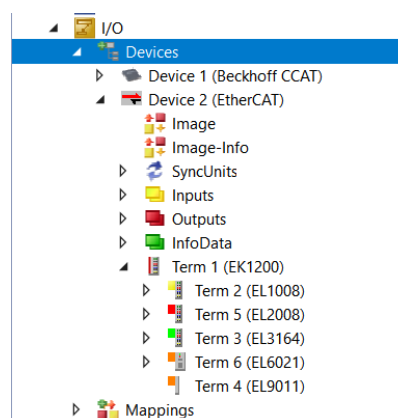
Dále je potřeba přidat terminály, které jsem k PLC připojili v hardwarové konfiguraci, aby se s nimi dalo pracovat v prostředí TwinCAT 3. Přidání terminálu lze provést ručně nebo pomocí automatického průvodce (*Scan*) pro objevování připojených terminálů k samotnému PLC. Vzhledem k tomu, že při návrhu systému můžeme mít do PLC připojených až stovky rozšiřujících terminálů, jeví se tato volba jako lepší řešení. Ruční přidávání terminálů se vyplatí

spíše při přidávání jednotlivých rozšiřujících modulů, v již nakonfigurovaném systému. I přes to, že v našem případě máme pouze 4 rozšiřující terminály, použijeme průvodce pro automatické objevování rozšiřujících terminálů. Průvodce se vyvolá pomocí kliknutí pravého tlačítka myši na položku *Device* v části *I/O* stromu projektu a po kliknutí na tlačítko *Scan*, dojde k prohledání připojených terminálů k PLC. Tato volba je znázorněná na obrázku 2.11.



Obrázek 2.11 – Znáznornění tlačítka *Scan*, pro prohledání připojených terminálů

Před začátkem hledání je ještě nutné zvolit na jakých zařízení má hledání probíhat, přičemž *Device 1* je tzv. Beckhoff CCAT, což je adaptér pro různá rozhraní (např. Ethernet) a *Device 2* je komunikační sběrnice pro EtherCAT, přes kterou rozšiřující terminály komunikují s výpočetní jednotkou PLC. Po dokončení hledání se ve stromu projektu PLC zobrazí tyto 2 podsložky. V podsložce *Device 1* se nově zobrazí *Term 1 (EK1200)*, což reprezentuje samotnou výpočetní jednotku PLC CX8190 a další připojené terminály, které jsou přehledně uvedené v tabulce 2.2, včetně stručného popisu. Grafická ilustrace, v prostředí TwinCAT 3, přidávaných terminálů je znázorněná na obrázku 2.12.

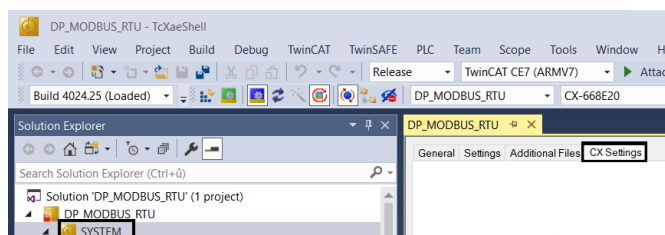


Obrázek 2.12 – Nalezené terminály připojeny k PLC

Tabulka 2.1 – Nalezené rozšiřující terminály

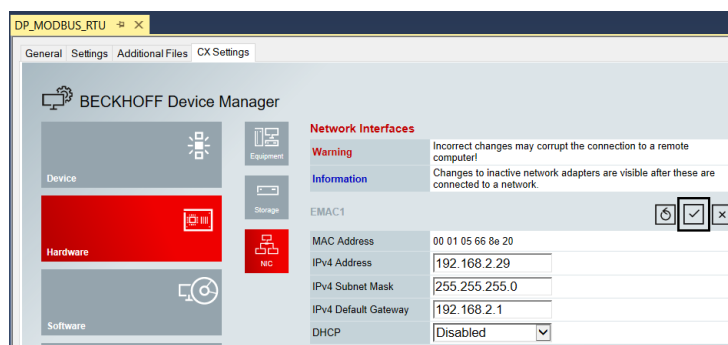
Název	Označení	Popis terminálu
Term 1	EK1200	Výpočetní jednotka PLC CX8190
Term 2	EL1008	8 vstupový digitální terminál (24 V DC)
Term 3	EL2008	8 výstupový digitální terminál (24 V DC; 0,5 A na jeden výstup)
Term 4	EL3164	4 vstupový analogový terminál 0–10 V DC (rozlišení 16 bitů)
Term 5	EL6021	Komunikační terminál pro rozhraní RS485/RS422
Term 6	EL9011	Zakončovací terminál. Slouží ke krytí kontaktů sběrnice EtherCAT

Po takto přidaných terminálech je ještě nutné nastavit pro PLC možnost připojení k dotykovému displeji (HMI), přes který budeme ovládat samotné PLC na eventuálním pracovním stanovišti. Nastavení může probíhat v prostředí TwinCAT 3 anebo i ve webovém prohlížeči na adrese <https://IP-adresa-PLC/config>. V této práci proběhne ukázka přímo v prostředí TwinCAT 3, kliknutím na *SYSTEM* a následně na *CX Settings*, dle ilustrace na obrázku 2.13.



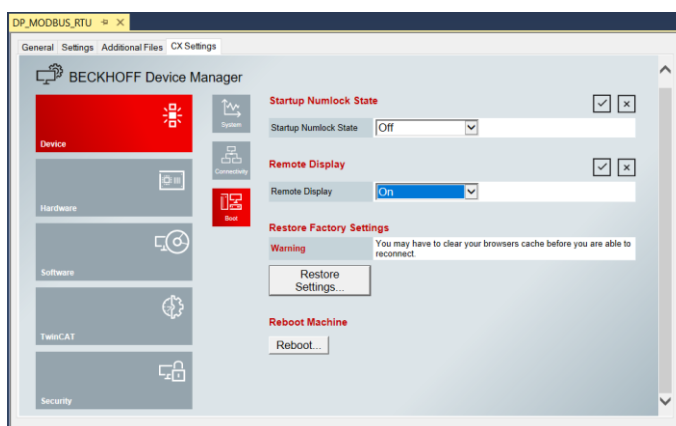
Obrázek 2.13 – První krok aktivace nastavení PLC

Dále je nutné potvrdit bezpečnostní volbu certifikátu, aby se mohlo pokračovat v nastavení. V nastavení PLC je nejprve vhodné nastavit statickou IP adresu PLC, masku podsítě a výchozí bránu. K tomu nastavení se dostaneme pomocí zvolení volby *Hardware* a poté volby *NIC*. Toto nastavení se ukládá pomocí fajfky je znázorněné na obrázku 2.14.



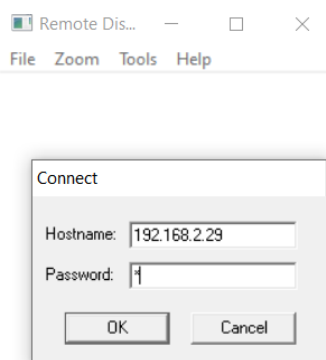
Obrázek 2.14 – Nastavení statické IP adresy PLC

Dále je nutné povolit vzdálené připojení k PLC pomocí programu CERHOST.EXE, kterým lze přidat a nastavovat síťová rozhraní. Toto nastavení se zobrazí po kliknutí na volbu *Device* a následně na *Boot*. Přepnutím volby *Remote Display* na volbu *ON* je vzdálené připojení aktivováno. Grafická ilustrace nastavení pro vzdálené připojení přes program CERHOST.EXE je znázorněná na obrázku 2.15.



Obrázek 2.15 – Aktivace nastavení pro vzdálené připojení programem CERHOST.EXE

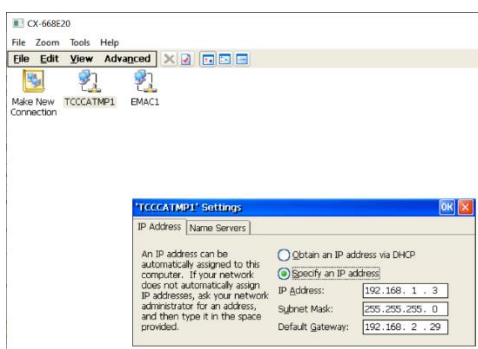
PLC CX8190 má v sobě celkem 3 Ethernet porty, kde jeden je pro programování a další 2 jsou možné nastavit jako síťový switch. Brána tohoto síťového switche se nejprve musí nastavit pomocí DIP switchů, aby byla adresa síťového switche uchovaná i v případě výpadku napájení a tento síťový switch se poté ještě musí nakonfigurovat pomocí aplikace CERHOST.EXE. Po spuštění aplikace CERHOST.EXE, klikneme na *File*, poté na *Connect*, zadáme námi zvolenou IP adresu s heslem „1“. Toto zobrazení je znázorněno na obrázku 2.15.



Obrázek 2.16 – Přihlášení do nastavení PLC přes aplikaci CERHOST.EXE

Po připojení do PLC pomocí programu CERHOST.EXE je nutné nastavit síťový switch (TCCCATMP1) dle obrázku 2.17. Tím pádem známe IP adresu switche, kterou budeme potřebovat pro nastavení HMI panelu, který bude komunikovat po ethernetu a zapojený bude

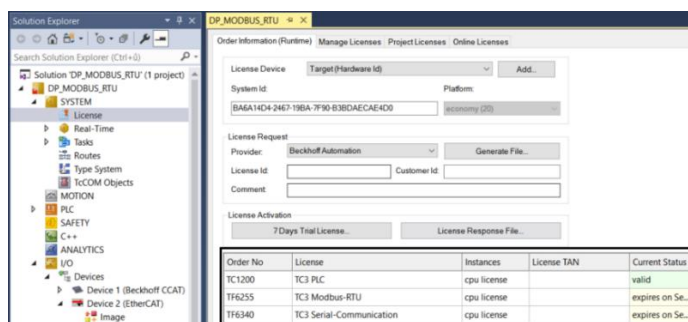
právě do tohoto síťového switche. V poslední řadě je nutné zjistit tzv. NetID našeho PLC, protože tato informace bude nutná zadat při konfiguraci programovacího prostředí EasyBuilder Pro, ve kterém dochází k návrhu grafického rozhraní HMI panelu. Tímto je softwarová konfigurace hotová a je možné přejít k programování samotného PLC, aby vykonával svoji činnost, která zapadá do funkce našeho SCADA systému.



Obrázek 2.17 – Nastavení síťového switche PLC

Licencování funkcionalit v programu TwinCAT 3

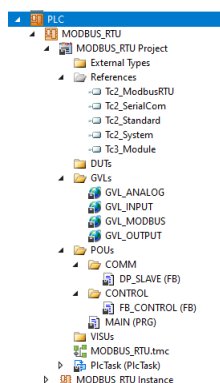
Společnost Beckhoff Automation poskytuje programovací prostředí zdarma, ale na druhou stranu, pokud chceme programovat jejich PLC, musíme si koupit licenci na „TC3 PLC“. Máme-li v úmyslu používat i komunikační protokol Modbus RTU na sériovém rozhraní, program TwinCAT 3 bude vyžadovat licenci na *TC3 Modbus-RTU* a licenci na *TC3 Serial-Communication*. Programovací prostředí TwinCAT 3 však umožňuje použití sedmi denních licencí zdarma, které se generují pomocí zadání vytvořeného kódu. Po 7 dnech se PLC Beckhoff přepne do konfiguračního režimu a program PLC přestane probíhat, dokud se opět nevygenerují dočasné licence, popř. nedojde-li k zakoupení trvalé licence přímo u zastoupení firmy Beckhoff. Na obrázku 2.18 jsou znázorněné aktivované licence pro výše uvedené funkcionality.



Obrázek 2.18 – Licence projektu v programu TwinCAT 3

2.2.2 Program PLC Beckhoff CX8190

Hlavní činností PLC v takovém SCADA systému je zejména obsluha a ovládání periférií, tedy senzorů a akčních členů. V této diplomové práci budeme obsluhovat (číst) 8 digitální vstupů, ovládat (zapisovat) 8 digitálních výstupů a číst 4 analogové vstupy. Dalším důležitým, ale velice důležitým úkolem, je pak komunikace po zvoleném komunikačním rozhraní, v tomto případě sériové linka RS-485 a zároveň obsluha tohoto rozhraní pomocí komunikačního protokolu se samotnou akviziční jednotkou. V druhé řadě pak PLC komunikuje s HMI panelem od firmy Weintek. HMI panelu bude věnována samostatná podkapitola. Na obrázku 2.19 je znázorněný strom programu PLC.



Obrázek 2.19 – Strom projektu s vytvořenými složkami a položkami

Jako v každém jiném programu, nejprve načítáme knihovny, poté deklarujeme proměnné, a nakonec se zabýváme vlastní logikou, kterou má program vykonávat. Stejně je tomu tak i v programování pomocí aplikace TwinCAT. Nejprve přidáme použité knihovny v části *References*. Dále v části *GVLs* vytváříme seznamy globálních proměnných pro analogové vstupní hodnoty, pro digitální vstupy i výstupy a také pro globální proměnné komunikačního protokolu Modbus-RTU. Nakonec v části *POUs* máme 2 složky, *COMM* (akronym pro Communications, resp. Komunikace) a *CONTROL*, který znamená v překladu mj. ovládání. Ve složce *COMM* se nachází program s názvem *DP_SLAVE* pro obsluhu sériové sběrnice RS485 a pro obsluhu komunikačního protokolu Modbus-RTU v režimu slave. Programy *DP_SLAVE* i *FB_CONTROL* jsou napsány v jazyce Structured Text. Tyto programy se musí inicializovat z hlavního programu (*MAIN*), který tyto uživatelské programy vykonává cyklicky. Ukázka programu *DP_SLAVE* je znázorněná na obrázku 2.20. V horní části dochází k deklaraci lokálních proměnných a ve spodní části jsou pak zavolány objekty instancí, které byly deklarovány výše. Instance objektu v tomto případě slouží k nastavení komunikačního

protokolu Modbus RTU a k nastavení komunikačního rozhraní RS-485. Kompletní zdrojové kódy, mj. PLC programu, budou přílohou této diplomové práce.

```
1 FUNCTION_BLOCK DP_SLAVE
2 VAR_INPUT
3 END_VAR
4 VAR_OUTPUT
5 END_VAR
6 VAR
7 MBSlave: ModbusRtuSlave_KL6x22B;
8 SlaveCom: KL6configuration;
9 MSErrorId: MODBUS_ERRORS;
10 ExecuteConfig: BOOL;
11 ConfigDone: BOOL;
12 ConfigBusy: BOOL;
13 ConfigError: BOOL;
14 ConfigErrorID: ComError_t;
15 END_VAR
16
17 MBSlave(
18   UnitID:=1 ,
19   AdrInputs:= ADR(GVL_MODBUS.Inputs),
20   SizeInputBytes:= SIZEOF(GVL_MODBUS.Inputs),
21   AdrOutputs:=ADR(GVL_MODBUS.Outputs) ,
22   SizeOutputBytes:=SIZEOF(GVL_MODBUS.Outputs) ,
23   AdrMemory:=ADR(GVL_MODBUS.Memory) ,
24   SizeMemoryBytes:=SIZEOF(GVL_MODBUS.Memory) ,
25   ErrorId=>MSErrorId );
26
27 SlaveCom(
28   Execute:=ExecuteConfig ,
29   Mode:=SERIALLINEMODE_KL6_22B_STANDARD ,
30   Baudrate:=38400 ,
31   NoDatabits:=8 ,
32   Parity:=PARITY_NONE ,
33   Stopbits:=1 ,
34   Handshake:=HANDSHAKE_NONE ,
35   ContinousMode:=TRUE ,
36   pComIn:=ADR(MBSlave.InData) ,
37   pComOut:=ADR(MBSlave.OutData) ,
38   SizeComIn:=SIZEOF(MBSlave.InData) ,
39   Done=>ConfigDone ,
40   Busy=>ConfigBusy ,
41   Error=>ConfigError ,
42   ErrorId=>ConfigErrorID );
```

Obrázek 2.20 – Program obsluhující Modbus RTU a RS-485

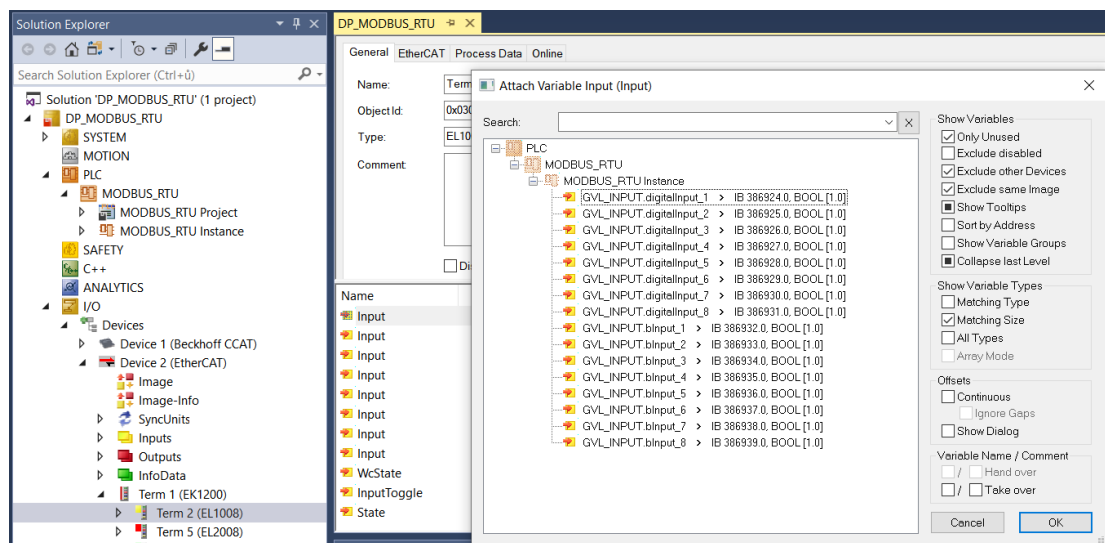
Logika stavby programu

Jak již bylo řečeno mj. v teoretické části této diplomové práce, program PLC pracuje ve stále se opakujícím cyklu. Program čte hodnoty vstupů, což jsou v našem případě tlačítka na panelu HMI a tlačítka na webovém zobrazení, resp. ovládání našeho SCADA systému. Dále tyto údaje zpracovává a následně zapisuje výstupy. Výstupy v našem případě reprezentují stavy LED signalizace na HMI panelu a také na webovém zobrazení, resp. ovládání našeho SCADA systému. Komunikace přes protokol Modbus-RTU probíhá paralelně s výše uvedeným. Modbus-RTU z pohledu PLC zapisuje hodnoty z předem nadefinovaných globálních proměnných do výstupů a čte hodnoty ze vstupů. Tato data jsou odesílána, resp. přijímána přes sběrnici RS485 do akviziční jednotky, která je dále zpracovává.

Spojení proměnných programu s fyzickým vývodem PLC

Spojení proměnných z programu PLC a fyzických vývodů vstupně výstupních rozšiřujících modulů probíhá v programovacím prostředí TwinCAT 3. Každý rozšiřující modul, v sekci I/O má ke každému terminálu možnost zvolit, k jaké proměnné je svázán. Při deklaraci

proměnných je nutné v programu uvést, zdali je proměnná vstupní nebo výstupní a jakého je datového typu. Rozšiřujících digitální terminály, ať už vstupní nebo výstupní, pracují vždy s datovým typem *boolean*, jinak by ke svázání proměnné nemohlo dojít. Analogové, vstupní či výstupní, rozšiřující terminály zase pracují s datovým typem *integer* a tím pádem je nutné tyto proměnné deklarovat jako *integer*. Svázání proměnné typu *integer* s analogovým vstupním rozšiřujícím terminálem je znázorněné na obrázku 2.21. Globální proměnná *GVL_ANALOG.analogValue_1* je tedy datového typu *integer*.

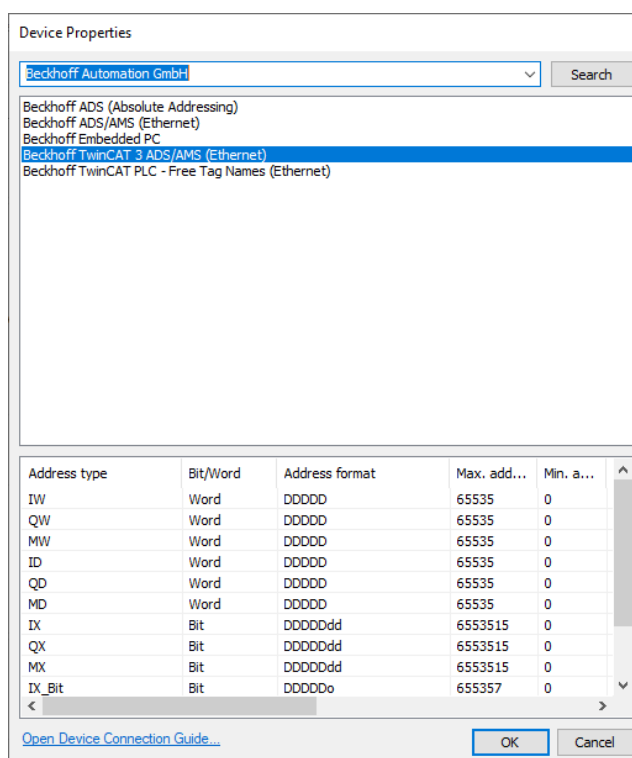


Obrázek 2.21 – Spojení proměnné programu a fyzického vstupu PLC

2.3 HMI PANEL WEINTEK

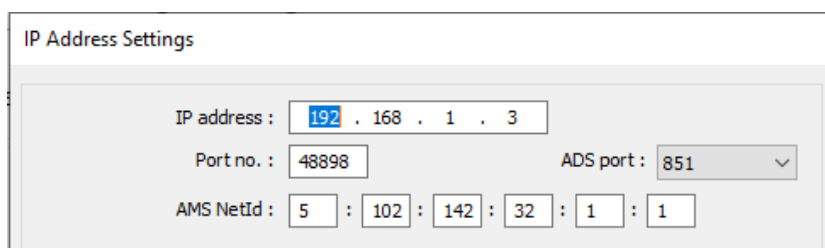
Pro interakci s okolím v bezprostřední blízkosti PLC byl zvolen HMI panel od firmy Weintek s produktovým označením MT8071iP. V následujících odstavcích bude proveden popis nastavení tohoto panelu, aby se jím daly vyčítat data z PLC a zároveň, aby se do PLC daly hodnoty (stavy) zapisovat. HMI panel vyžaduje napájení 24 V DC a připojení pomocí UTP nebo STP kabelu z PLC. Vzhledem k tomu, že nastavení PLC pomocí programu CERHOST.EXE bylo dokončeno, zbývá nastavit program EasyBuilder Pro, ve kterém se mj. navrhuje grafické rozhraní (GUI) HMI panelu. Při vytváření projektu se musí zadat správný typ HMI panelu, aby program správně zkompiloval navržené grafické rozhraní ve smyslu rozlišení displeje, výpočetní výkon atp. V druhém kroku, po zvolení správného typu HMI, je nutné nastavit komunikaci mezi PLC a HMI. V předchozích kapitolách bylo nastaveno PLC a nyní toto nastavení, zejména IP adresy, musí respektovat i HMI panel přes aplikaci EasyBuilder Pro.

Na obrázku 2.22 je znázorněné nastavení, které zajišťuje výběr správného komunikačního protokolu pro komunikaci s Beckhoff PLC.



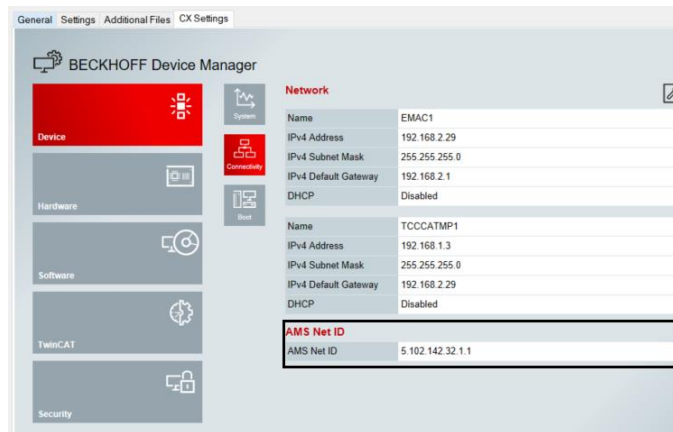
Obrázek 2.22 – Výběr správného nastavení pro komunikaci s Beckhoff PLC

V našem případě se jedná o Beckhoff ADS protokol. Dále jsou na obrázku 2.23 znázorněné informace, které musí programátor zadat, aby displej správně komunikoval s PLC.



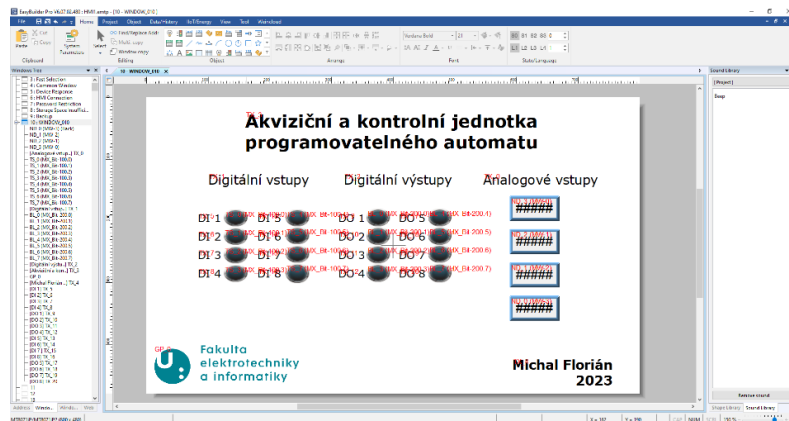
Obrázek 2.23 – Nastavení IP a NetID v HMI panelu

Jedná se o IP adresu síťového switchu v PLC a o NetID samotného PLC. NetID si PLC generuje samo a dá se zjistit přes nastavení PLC, které bylo popisováno v kapitole 2.2.1. Ukázka konkrétního NetID použitého při nastavení HMI panelu je znázorněná na obrázku 2.24.



Obrázek 2.24 – NetID PLC Beckhoff CX8190

Kdyby toto nastavení nebylo vyplněné správně, HMI panel by s PLC nekomunikoval. V dalším kroku probíhá samotný návrh grafického rozhraní. Již hotové rozhraní je znázorněné na obrázku 2.25.



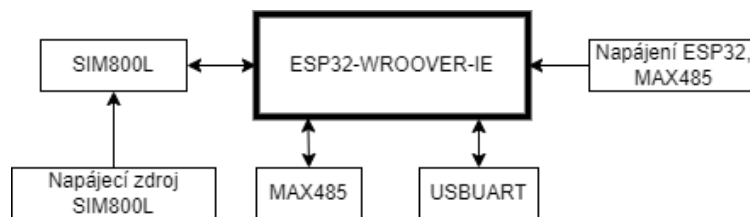
Obrázek 2.25 – Hotové GUI HMI panelu

Po dokončení návrhu je nutné tento návrh zkompileovat pomocí tlačítka *Compile* na kartě *Project*. Po kompilaci je možné nahrát vytvořené grafické rozhraní do samotného HMI panelu. Nahrávání probíhá pomocí USB klíčenky, která však musí být naformátovaná do souborového systému FAT32, což se spouští tlačítkem *Build Download Files*, výběrem správné jednotky, kterou počítač přidělil USB klíčence a konečně stisknutím tlačítka *Build*. Soubory projektu se tímto nahrají na USB klíčenku. Vložením USB klíčenky do HMI panelu dojde k zobrazení dialogového okna, kde zvolíme *Download*. Po stisknutí tlačítka *Download* je nutné zadat heslo do HMI panelu, které je v základním nastavení ve tvaru „111111“. Po zadání hesla je nutné zvolit v souborovém manažeru správnou USB klíčenku a poté již HMI panel zpracovává soubory z USB klíčenky a ukládá do své paměti projekt. Po dokončení nahrávání se HMI panel

restartuje a po restartu je na displeji námi vytvořený projekt. Tímto je nastavení HMI panelu dokončeno.

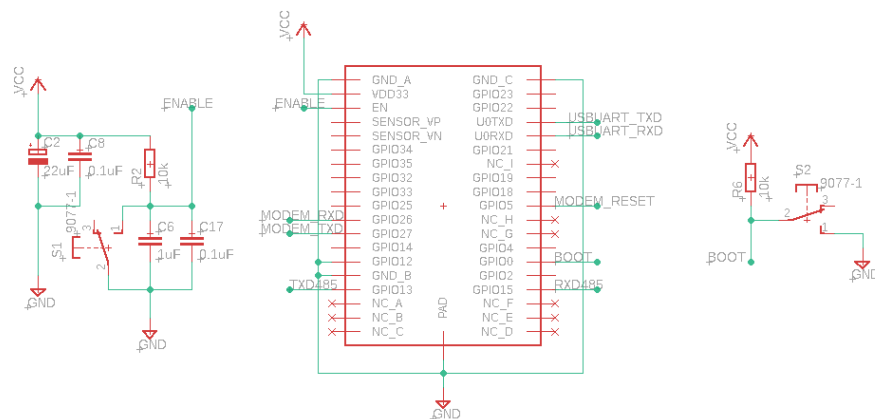
2.4 AKVIZIČNÍ A KONTROLNÍ JEDNOTKA

Jednotka samotná je realizovaná jako deska plošného spoje, která do SCADA systému zapadá jako komunikační rozhraní mezi PLC a webovým serverem. Blokové schéma jednotky je znázorněné na obrázku 2.26.



Obrázek 2.26 – Blokové schéma akviziční a kontrolní jednotky

Jednotka s PLC komunikuje pomocí komunikačního protokolu Modbus-RTU, který je realizován pomocí komunikačního rozhraní RS-485. Komunikace mezi jednotkou a webovým serverem probíhá na základě připojení na internet skrze modul SIM800L a samotná komunikace je realizována pomocí protokolu MQTT. Jednotka je založená na mikro počítači ESP32-WROOVER-IE, který byl zvolen na základě jeho dostupnosti a ceně. V následujících podkapitolách dojde k popisu jednotlivých komponent, které ze zmíněného mikro počítače tvoří akviziční a kontrolní jednotku. Schéma obvodu mikro počítače ESP32-WROOVER-IE je znázorněné na obrázku 2.26.



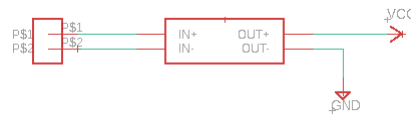
Obrázek 2.27 – Schéma obvodu mikro počítače ESP32-WROOVER-IE

2.4.1 Napájecí obvody

Vzhledem k tomu, že celý SCADA systém je realizován s průmyslovými komponenty, které jsou zpravidla napájeny 24 V DC, je v jednotce nutné zajistit převod této napěťové úrovně na úroveň, se kterou jednotlivé komponenty pracují spolehlivě. V jednotce jsou použity 2 typy převodníků napětí.

Převodník napětí ze 24 V DC na 3,3 V DC

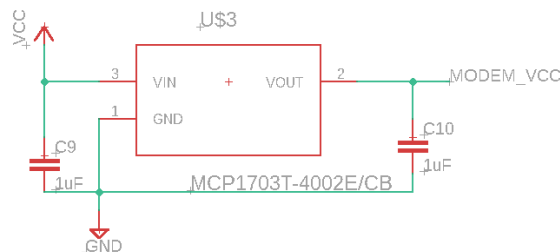
Jako hlavní zdroj napětí a proudu je převodník, *AIMTEC AM6CW-2403S-NZ*, se vstupním rozsahem 9 ÷ 36 V DC na 3,3 V DC. Tento převodník dodává až 1,5 A na napěťové úrovni 3,3 V DC, což plně dostačuje. Na napěťové úrovni 3,3 V DC pracuje samotný mikropočítač ESP32-WROOVER-IE a integrovaný obvod MAX485. Jeho zapojení je jednoduché, má 2 vstupní vývody a 2 výstupní vývody. Schéma zapojení je znázorněné na obrázku 2.27.



Obrázek 2.28 – Schéma obvodu převodníku napětí 24 V DC na 3,3 V DC

Převodník napětí ze 3,3 V DC na 4 V DC

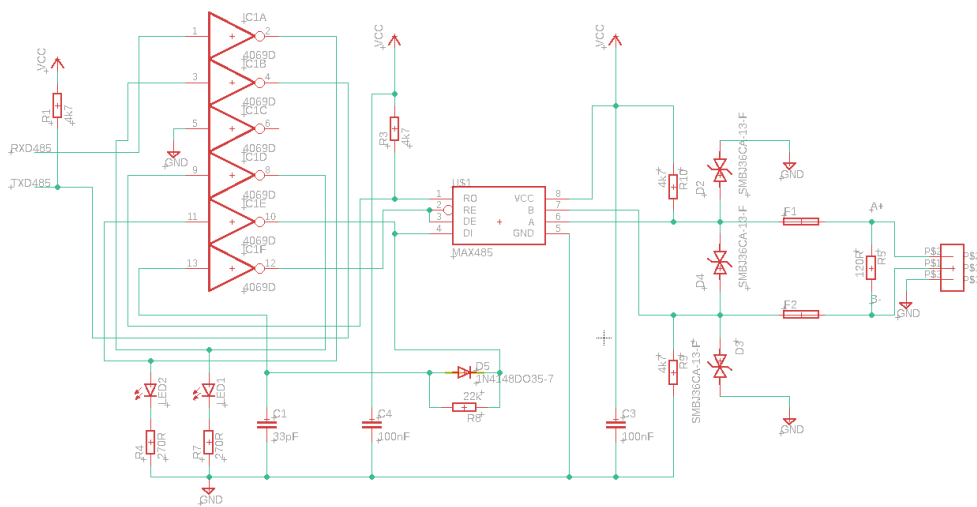
Dále je nutné zajistit napájení 4 V DC se špičkovým odběrem proudu až 1 A pro externí modul *SIM800L*. Pro tento účel byl použitý převodník ze 3,3 V DC na 4 V DC, *MCP1703T-4002E/CB*. Tento převodník napětí dodává pouze 250 mA, z toho důvodu jsou použity 4 tyto převodníky v paralelním zapojení. Schéma zapojení je znázorněné na obrázku 2.28.



Obrázek 2.29 – Schéma obvodu převodníku napětí 3,3 V DC na 4 V DC

2.4.2 Obvod pro komunikační rozhraní RS-485

Jedním z hlavních úkolů akviziční jednotky je schopnost komunikace pomocí rozhraní RS-485. Pro tento účel jednotka disponuje integrovaným obvodem *MAX485*. K tomu integrovanému obvodu je implementován další integrovaný obvod, s názvem *CD4069UBE*, který obsahuje 6 invertorů a jeho zapojení garantuje nekolizní stav mezi příjmem a odesíláním dat po rozhraní RS-485 přímo ze stavu vývodu pro vysílání a pro příjem. Další výhodou je, že se tímto ušetří vývod mikropočítače, který řídí směr komunikace. Na obrázku 2.29 je schéma zapojení této části akviziční jednotky. (Monahan, 2023).



Obrázek 2.30 – Schéma zapojení obsluhující komunikaci RS-485

2.4.3 Externí moduly

Jak již bylo zmíněno výše, akviziční jednotka používá 2 externí moduly. Modulem *SIM800L* se navazuje komunikace přes datovou síť GPRS. Druhým externím modulem je programovací modul, resp. převodník z USB na UART založený na integrovaném obvodu *FTDI232*.

Komunikační *SIM800L*

Externí modul *SIM800L* lze zakoupit i jako samostatný integrovaný obvod, ale při pokusech s jeho připájením mikro páječkou vždy došlo k poškození samotného integrovaného obvodu a z toho důvodu byl zvolen externí modul, který se připojuje 5 vývody do headeru na akviziční jednotce. Tento externí modul je k jednotce připevněn kromě samotného zapojení

5 vývody do headeru pomocí tavné pistole. Ilustrační obrázek tohoto modulu se zapojenou anténou je znázorněn na obrázku 2.30.



Obrázek 2.31 – Externí modul SIM800L se zapojenou anténou

Programovací převodník USB-UART FTDI232

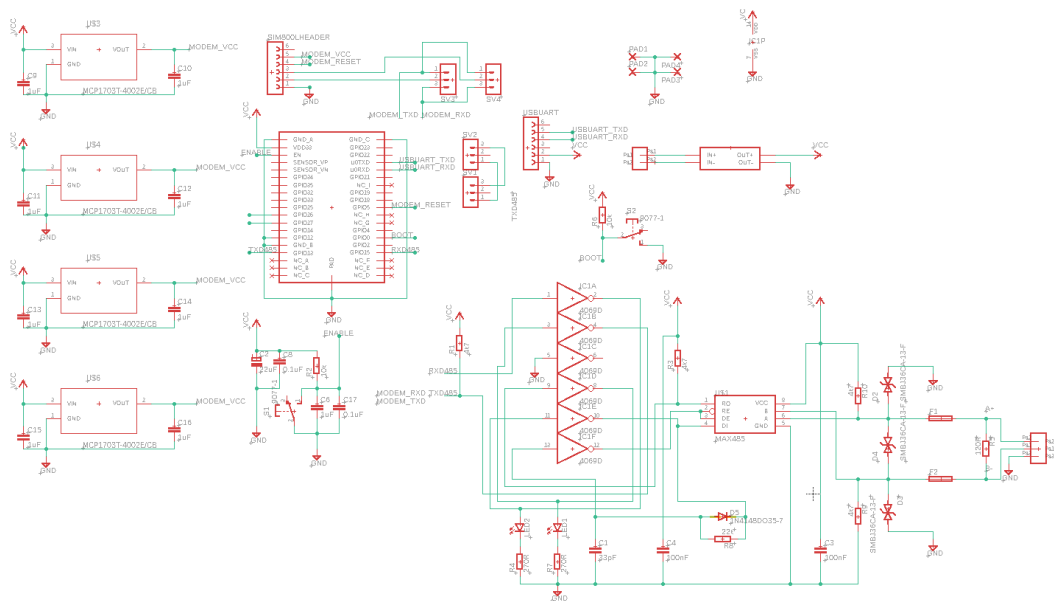
Tento převodník disponuje konektorem MiniUSB, kterým se připojuje k počítači a poté 6 vývodovým headrem, který se připojuje do akviziční jednotky. Header je spojený s hardwarovou sériovou linkou samotného mikropočítače a dochází tím k jeho naprogramování. Po naprogramování se externí programovací modul odpojí a dále nezůstává součástí jednotky. Ilustrační obrázek tohoto programovacího modulu je znázorněný na obrázku 2.31.



Obrázek 2.32 – Externí programovací modul USB-UART

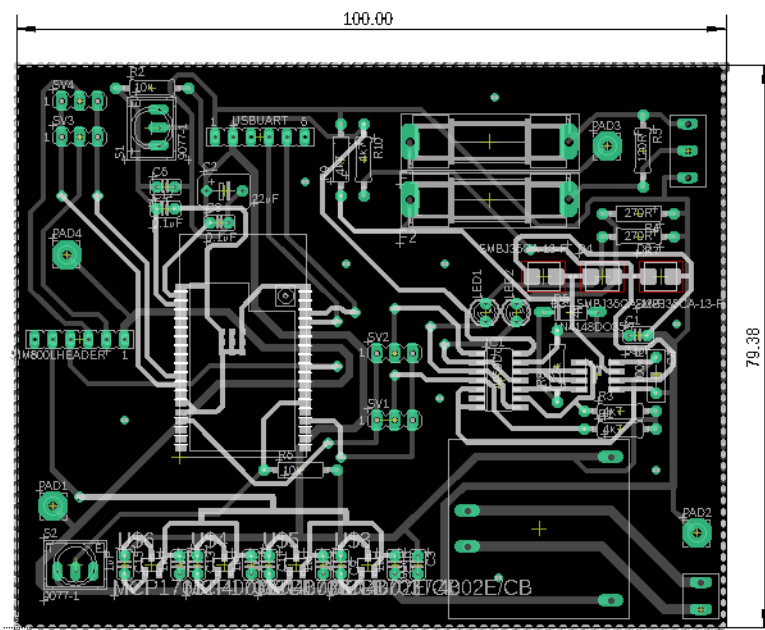
2.4.4 Celkové schéma jednotky

Na obrázku 2.33 je znázorněné kompletní schéma akviziční a kontrolní jednotky.



Obrázek 2.33 – Celkové schéma akviziční a kontrolní jednotky

Na obrázku 2.34 je znázorněný návrh desky plošného spoje akviziční a kontrolní jednotky. Z obrázku je patrné, že se jedná o dvouvrstvý plošný spoj.



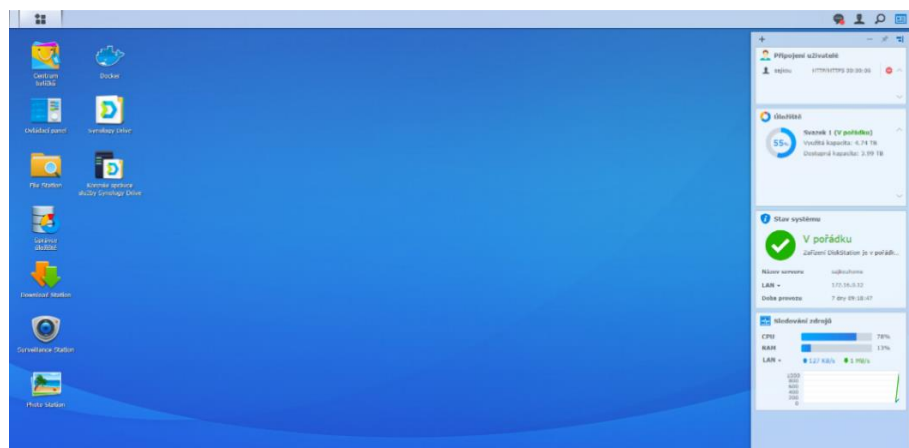
Obrázek 2.34 – Návrh desky plošného spoje

2.4.5 Programování akviziční a kontrolní jednotky

Tento mikropočítač umožňuje programování pomocí frameworku Arduino v programovacím prostředí PlatformIO, které je nadstavbou programovacího prostředí Visual Studio Code. Programovacím jazykem je jazyk C++ a tím pádem je možné využít jakoukoli knihovnu, která je dostupná v jazyce C++. Je však zásadní použít knihovnu, která je zároveň vytvořená pro práci s mikropočítači z rodiny ESP32. Například byla použita knihovna TinyGSM+, která obsluhuje navazování komunikaci s GPRS sítí pomocí externího modulu SIM800L. Dále jsou použity ModbusMaster pro navázání komunikace přes protokol Modbus a knihovna PubSubClient, která zajišťuje komunikace přes protokol MQTT. Na obrázku 2.35 je znázorněný vývojový diagram programu akviziční a kontrolní jednotky. Kompletní vývojový diagram programu akviziční a kontrolní jednotky je součástí přílohy A této diplomové práce.

2.5 VZDÁLENÝ SERVER DISKSTATION

Server DiskStation je spuštěn na zařízení Synology DS218+. Server DiskStation je založený na UNIXovém operačním systému a jedná se primárně server pro síťové úložiště s možností instalací dalších serverových aplikací. Rozhraní serveru je ovládané z webového prohlížeče. Nicméně nastavení některých služeb probíhá přes konzoli typu SecureShell (SSH). Uživatelské grafické rozhraní je znázorněné na obrázku 2.35.



Obrázek 2.35 – Webové zobrazení pro interakci se serverem

Důležitým uživatelským balíčkem, který je možné nainstalovat je Docker, ve kterém je spuštěna serverová aplikace Node-RED. O aplikaci Node-RED pojednává následující podkapitola. Dalším balíčkem, který lze nainstalovat je serverová aplikace Moquitto, která slouží jako MQTT server. Serverová aplikace Mosquitto nemá uživatelské grafické rozhraní

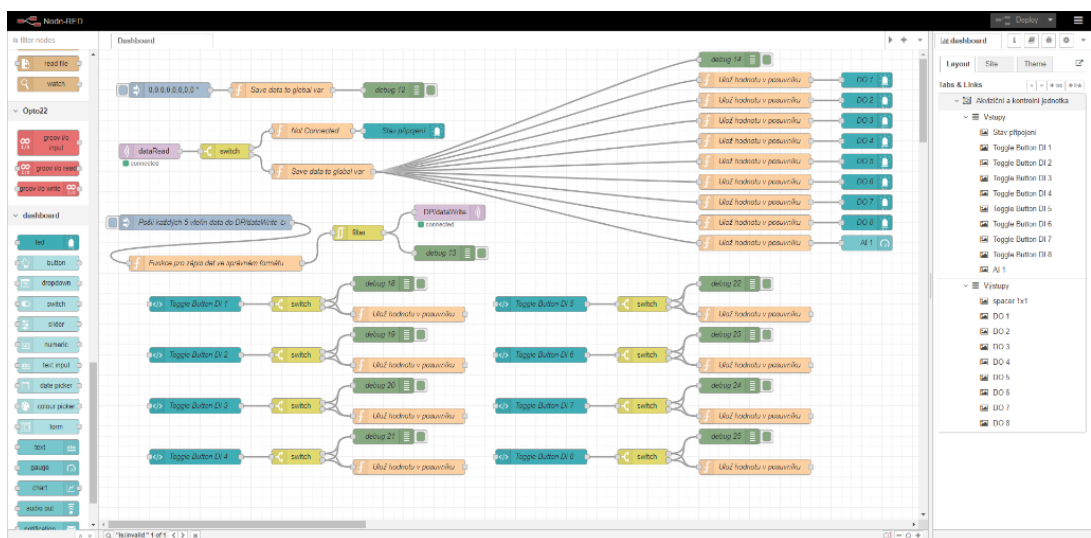
a jediná možnost, jak ji nastavit, je použití konzole. O tomto nastavení bude rovněž pojednávat následující podkapitola. Server Synology DS218+ je znázorněný na obrázku 2.36.



Obrázek 2.36 – Ilustrační obrázek Synology DiskStation 218+

2.5.1 Node-RED

Node-RED serverová aplikace se ovládá přes uživatelské grafické rozhraní, které je ve formě webové stránky. Programování v aplikaci Node-RED probíhá pomocí grafických bloků, které jsou k sobě pospojovány. Každý blok má své předem definované nastavení, které lze v případě, že je nutné doprogramovat funkci, využít bloku funkce. Do bloku funkce lze vepsat kód v jazyce JavaScript a tím pádem lze dosáhnout téměř čehokoli. V této diplomové práci bylo nutné pracovat na Node-REDu s globálními proměnnými a tímto se dalo dosáhnout pouze za pomoci bloku funkce a kódu v jazyce JavaScript. Node-RED pracuje s tzv. *flow*, což se volně překládá jako tok a je to reprezentací toku dat mezi jednotlivými bloky. Již hotové *flow*, které je pak součástí SCADA systému, je znázorněné na obrázku 2.37.



Obrázek 2.37 – *Flow* programu na Node-RED

2.5.2 Mosquitto Broker

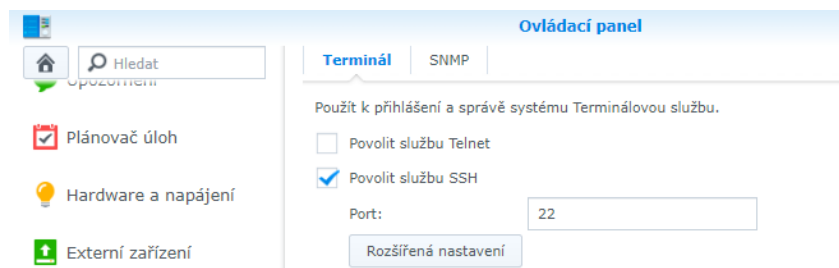
Serverová aplikace Mosquitto se instaluje z centra balíčků přímo v Synology DiskStation pomocí webového uživatelského rozhraní a jedná se o serverovou aplikaci pro MQTT. Než bude popsáno nastavení aplikace Mosquitto, bude v následující podkapitole stručně popsán komunikační protokol MQTT.

Komunikační protokol MQTT

Akronym MQTT znamená Message Queuing Telemetry Transport. Jedná se o komunikační protokol, jímž je umožněna komunikace prostřednictvím TCP/IP, kdy mezi subjekty jsou odesílány krátké zprávy. Efektivitu a nenáročnost této komunikace zajišťuje skutečnost, že se zprávy skládají z malých datových paketů. Zprávami mohou například být posílány stavy vstupů či výstupů, která jsou odesílána do jednoho centrálního bodu nazývaného MQTT Broker, resp. MQTT server. Každá zpráva označená konkrétním názvem tématu (Topic), je brokerem distribuována všem klientům, kteří si toto téma nastavili pro příjem. Mezi klienty mohou patřit webové služby, programy nebo jiná zařízení, která dále zpracovávají přijaté zprávy. Tím pádem jsou v pravidelných časových intervalech, broker a jeho klienti, informováni prostřednictvím daného tématu. Zjednodušeně řečeno, zařízení odesílající zprávu posílá brokeru zprávu označenou určitým tématem a broker následně tuto zprávu rozdistribuuje všem klientům, kteří mají nastaven odběr tohoto konkrétního tématu. Toto zařízení (server) je zodpovědné za přijímání publikovaných zpráv a jejich následné preposílání odběratelům. Díky standardizaci MQTT protokolu existují různé implementace MQTT brokeru, což je v našem případě Mosquitto. (Martin, nedatováno).

Nastavení Mosquitto Brokera

Mosquitto se nastavuje pomocí konzole SSH serveru Synology DiskStation. Pro navázání spojení přes konzoli SSH je však nutné povolit službu SSH v nastavení Synology. Toto nastavení je dostupné v *Ovládacím panelu* v sekci *Terminál a SNMP*. Dále je nutné aktivovat volbu *Povolit službu SSH*. Toto nastavení dále umožňuje nastavit libovolný síťový port, na kterém se služba SSH spustí. Výchozí číslo portu je 22. Grafické znázornění tohoto nastavení je ilustrované na obrázku 2.38.

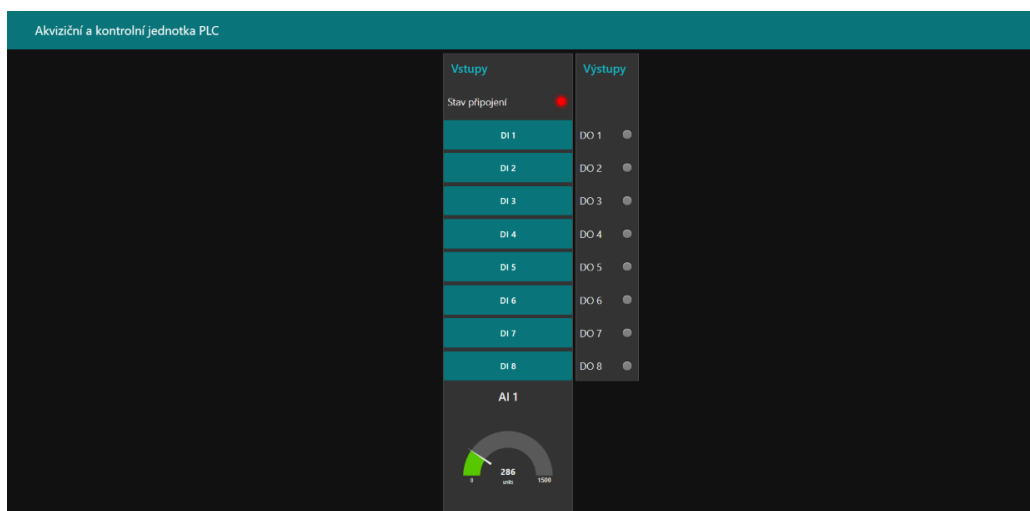


Obrázek 2.38 – Povolení služby SSH

Přes libovolnou konzolovou aplikaci se pomocí SSH připojíme k serveru. Nalezneme v souborovém systému konfigurační soubor *mosquitto.conf*, který v libovolném textovém editoru upravíme. Úprava spočívá pouze v přidání řádku *allow_anonymous true*. Toto nastavení umožňuje připojit se k MQTT brokerovi jako anonymní uživatel. Soubor uložíme a ukončíme editor. V našem případě se soubor *mosquitto.conf* nachází ve složce */volume1/@appstore/mosquitto/var*.

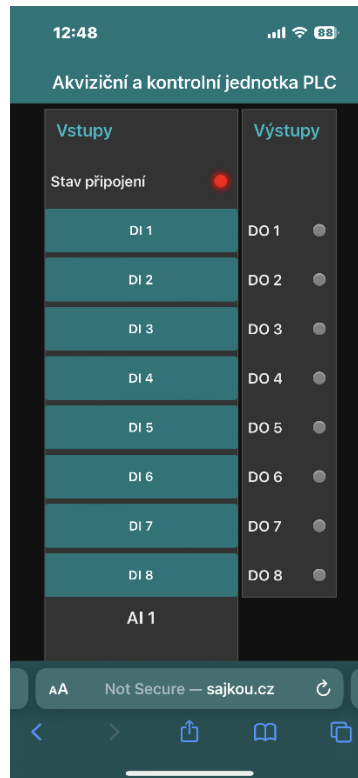
2.6 KLIENSKÉ ŽARÍZENÍ SCADA SYSTÉMU

Uživatelské grafické našeho SCADA systému je webová stránka, kterou generuje samotná serverová programovací aplikace Node-RED. Dashboard lze poměrně snadno nastavit pomocí grafického nástroje přímo v aplikaci Node-RED anebo za použití kaskádových stylů (CSS). Zjednodušeně lze říct, že klientem, což je v tomto případě webová stránka spuštěna na libovolném mobilním telefonu, který je připojený k internetové síti, lze vzdáleně ovládat programovatelný logický automat. Grafické znázornění klientské aplikace našeho SCADA systému je ilustrováno na obrázku 2.39.



Obrázek 2.39 – Klientské zobrazení v prohlížeči stolního počítače

Toto rozhraní je dostupné na veřejné adrese a pomocí libovolného zařízení, které disponuje připojením k internetu, je možné se k němu připojit. Pro přístup z mobilního zařízení, je ukázka tohoto rozhraní, znázorněná na obrázku 2.40.



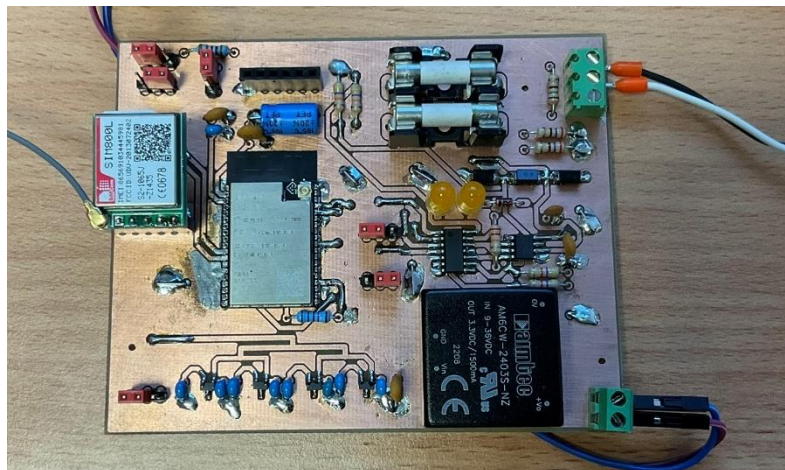
Obrázek 2.40 – Klientské zobrazení v prohlížeči mobilního telefonu

3 TESTOVÁNÍ AKVIZIČNÍ A KONTROLNÍ JEDNOTKY PROGRAMOVATELNÉHO AUTOMATU (SCADA)

Testování jednotky probíhalo na již známém PLC Beckhoff CX8190, na kterém se ovládalo 8 digitálních vstupních signálů, které byly svázány s 8 digitálními výstupy, které by mohly ovládat LED signalizaci, protože jejich maximální proudový odběr dosahuje maximálně 0,5 A na výstup anebo například vhodné relé, které pak může ovládat jakýkoli akční člen od elektromotoru po hydraulický přímočarý motor. Dalším testovaným signálem, které PLC četlo, byl analogový signál ($0 \div 10 \text{ V}$) z čidla odměřování vzdálenosti (*TE SP3-25*), což je lankový snímač do vzdálenosti 635 mm.

3.1 DESKA PLOŠNÉHO SPOJE AKVIZIČNÍ A KONTROLNÍ JEDNOTKY

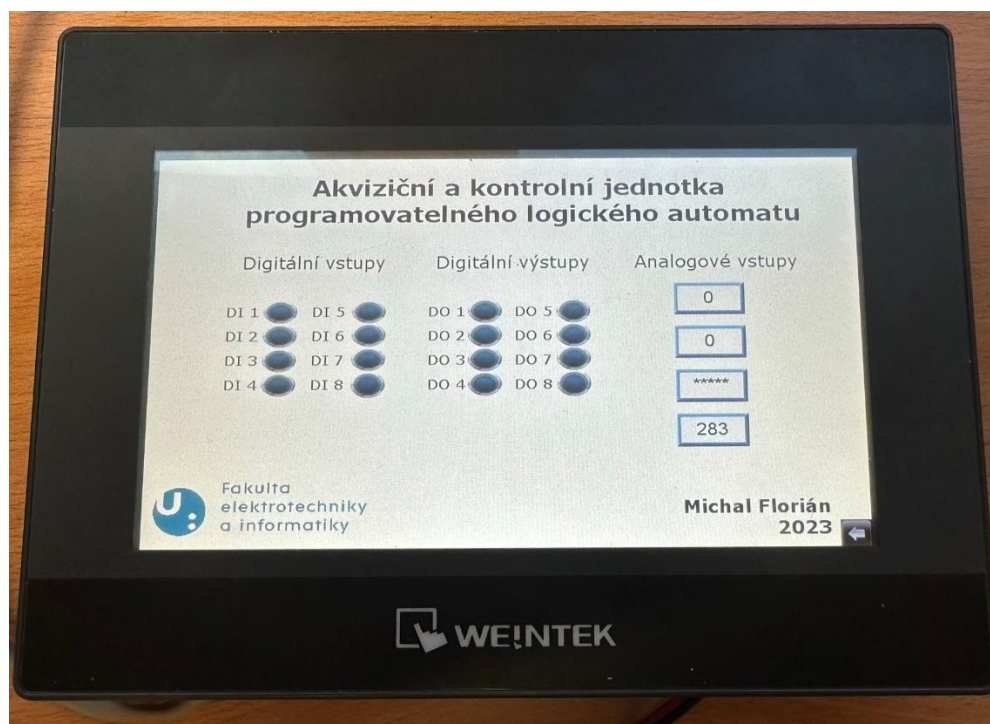
Hlavní komponentou SCADA systému je právě akviziční a kontrolní jednotka, na které testování probíhalo. Tato jednotka slouží jako most mezi vzdáleným serverem a PLC. Hotový prototyp jednotky je znázorněn na obrázku 3.1. K jednotce se připojuje napájecí napětí 24 V DC, resp. $9 \div 36 \text{ V DC}$, anténa do externího SIM800L modulu a konečně komunikační rozhraní RS-485, přes které komunikuje protokol Modbus-RTU. Při testování jednotka vykazovala bezchybné chování.



Obrázek 3.1 – Prototyp zapojené jednotky do SCADA systému

3.2 LOKÁLNÍ OVLÁDÁNÍ Z HMI PANELU

Lokální testování probíhalo s PLC Beckhoff CX8190 připojeným k HMI panelu Weintek MT8071iP. Odhad odezvy vstupního signálu při ovládání z HMI panelu dosahoval v řádu jednotek milisekund, protože na takto připojeném displeji přes rozhraní Ethernet je možné mluvit o ovládání v reálném čase. Je žádoucí, aby PLC pracovalo s odezvami takto nízkými, protože čas, za který má být vstupní signál zpracován, z titulu řízení technologických procesů, má být co nejkratší. Na obrázku 3.2 je znázorněn HMI panel, přes který lze ovládat PLC.



Obrázek 3.2 – HMI panel Weintek

3.3 VZDÁLENÉ OVLÁDÁNÍ Z KLIENSKÉ APLIKACE WEBOVÉHO PROHLÍŽEČE

Z testování vyšlo, že se PLC podařilo vzdáleně ovládat pomocí klientského zařízení, jak ve webovém prohlížeči stolního počítače, tak ve webovém prohlížeči mobilního zařízení. Odhadovaná odezva vstupního signálu při vzdáleném ovládání z klientské aplikace dosahovala času v řádech nižších jednotek sekund. Tato mnohonásobná odezva, oproti lokálnímu ovládání z HMI panelu, je způsobena tím, že pro připojení na internet využívá akviziční a kontrolní jednotka datovou síť GPRS, která je známa odezvou až kolem jedné sekundy. Na obrázku 3.3

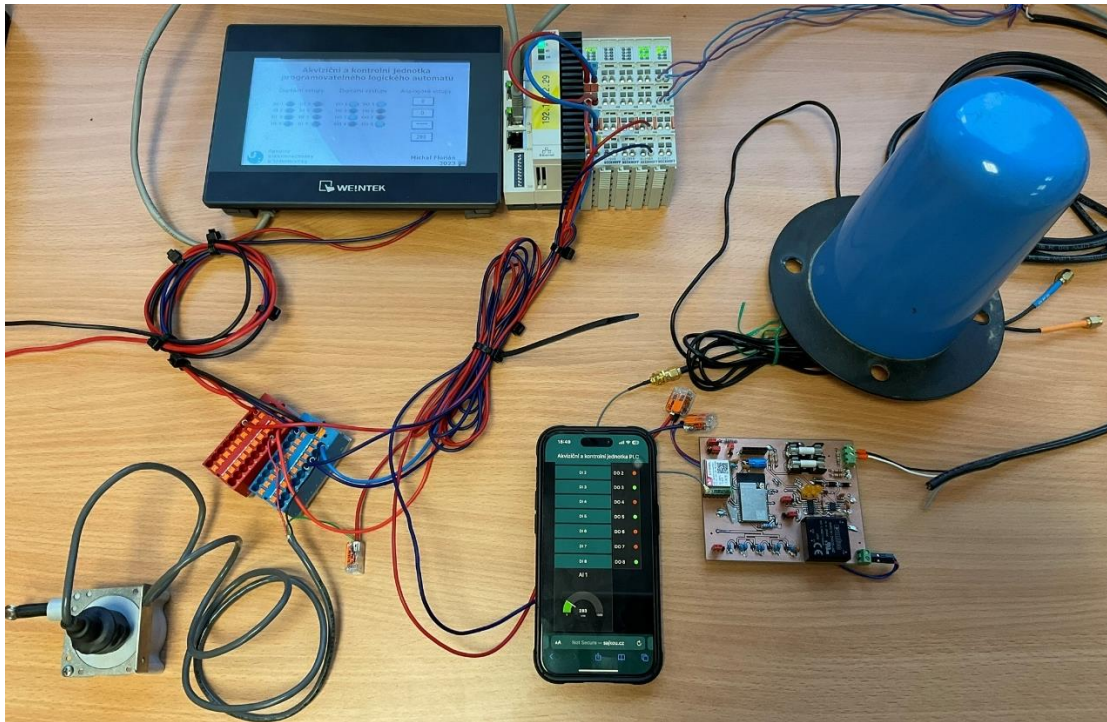
je znázorněný mobilní telefon se spuštěným webovým prohlížečem na příslušné adrese serveru, přes který lze ovládat PLC. Pro funkční kontrolu, zdali jsou zobrazovaná stejná data, je pro porovnání přiložen i HMI panel.



Obrázek 3.3 – Klientké zařízení ve formě mobilního telefonu pro vzdálené ovládání

3.4 ZÁVĚR Z TESTOVÁNÍ SCADA SYSTÉMU

Testování akviziční a kontrolní jednotky programovatelného automatu jako součást vytvořeného SCADA systému proběhlo bez větších problémů. Pro zjednodušení lze říct, že cílem diplomové práce, ale i závěrečného testování, bylo vzdáleně ovládat programovatelný logický automat a sbírat z něj data, která se ukládají do MySQL databáze na server. Z výše uvedeného vyplývá, že se podařilo vytvořit SCADA systém, tedy vzdáleně ovládané PLC s možností sběru dat o zpracovaných signálech. SCADA systém se skládá z PLC, HMI panelu, akviziční a kontrolní jednotky, antény pro GPRS signál a konečně z klientského zařízení v podobě mobilního telefonu se spuštěným internetovým prohlížečem. Tento systém je jako celek znázorněný na obrázku 3.4.



Obrázek 3.4 – Vytvořený SCADA systém

ZÁVĚR

Programovatelné logické automaty jsou dnes nejpoužívanější zařízení pro řízení technologických procesů, jejich význam, je nejen v průmyslu a průmyslové automatizaci, enormní.

Výstupem práce je akviziční a kontrolní jednotka programovatelného logického automatu, což je realizované jako deska plošného spoje, která je spojena s internetem a pomocí protokolu MQTT odesílá a přijímá zprávy ze serveru. Jednotka je dále připojená přes komunikační rozhraní RS-485 a komunikuje pomocí protokolu Modbus-RTU s PLC, ke kterému je ještě připojen operátorský HMI panel. Pomocí HMI panelu probíhá lokální ovládání PLC a pomocí klientské aplikace pak probíhá vzdálené ovládání. Sběr dat obstarává serverová aplikace Node-RED, která má přímo v sobě vestavěnou funkcionalitu k připojení se na MySQL databázi, kde se do definovaných tabulek ukládají data.

Testování vzdáleného ovládání probíhalo pomocí stolního počítače a mobilního telefonu, kdy se v obou případech využíval internetový prohlížeč jako uživatelské rozhraní pro ovládání. Testováním byla potvrzena funkce v plném rozsahu, tedy vzdálené ovládání PLC, což znamená, že s touto jednotkou, připojenou do PLC, které řídí libovolný technologický proces, přidáváme možnost vzdáleně tento proces ovládat.

Návrh i testování akviziční a kontrolní jednotky proběhlo bez větších problémů. Byly potvrzeny všechny zadané funkce.

POUŽITÁ LITERATURA

- BECKHOFF, 2023. *CX8190 | Embedded PC with different Ethernet protocols* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.beckhoff.com/en-en/products/ipc/embedded-pcs/cx8100-arm-cortex-a9/cx8190.html>
- BECKHOFF, 2023. *EK1521 | Iport EtherCAT fiber-optic junction* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.beckhoff.com/en-en/products/i-o/ethercat-terminals/ek1xxx-bk1xx0-ethercat-coupler/ek1521.html>
- BECKHOFF. *Scada Solutions* [online]. [cit. 2023-08-27]. Dostupné z: https://infosys.beckhoff.com/english.php?content=../content/1033/sw_os/2018929035.html&id=
- CNCSHOP, 2010. *Polovodičové jednofázové SSR relé 660VAC* [online]. [cit. 2023-08-27]. Dostupné z: <http://www.cncshop.cz/polovodicove-jednofazove-ssr-rele-660vac>
- CRYDOM INC., 2009. *Solid State Relays – The Inside Story* [online]. [cit. 2023-08-27]. Dostupné z: <https://erg.abdn.ac.uk/users/gorry/eg3576/resources/Crydon-SSR%20the%20inside%20story.pdf>
- ČERNÝ, J., 1968. *Elektromechanická relé*. Státní nakladatelství technické literatury. ISBN 04-532-68.
- EATON. *EC4P compact programmable logic controller* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.eaton.com/ie/en-gb/catalog/industrial-control--drives--automation---sensors/ec4p-compact-plc.html>
- ECAVA INTEGRAXOR, 2020. *SCADA system architecture*. [online] Dostupné z: <https://www.integraxor.com/blog/wpcontent/uploads/2016/05/scada-system-architecture.png>. [cit. 2023-08-27].
- Electrical telegraph, 2023. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2023 [cit. 2023-08-27]. Dostupné z: https://en.wikipedia.org/wiki/Electrical_telegraph#/media/File:Morse_Telegraph_1837.jpg
- Elektromagnetické relé, 2022. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2022 [cit. 2023-08-27]. Dostupné z: https://cs.wikipedia.org/wiki/Elektromagnetick%C3%A9_rel%C3%A9
- ELLIOTT, R., 2013. *MOSFET Solid State Relays*. [online]. Dostupné z: <https://sound-au.com/articles/mosfet-relay.htm>. [cit. 2023-08-27].
- ESPBOARDS, 2023. *The Powerful ESP32 Processor for IoT - Is it ARM-based?* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.espboards.dev/blog/is-esp32-arm-based/>
- ESPRESSIF SYSTEMS, 2023. *ESP32-WROVER-E & ESP32-WROVER-IE Datasheet* [online]. [cit. 2023-08-27]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf

- ETHERCAT. *EtherCAT – the Ethernet Fieldbus* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.ethercat.org/en/technology.html#:~:text=EtherCAT%20was%20introduced%20in%20April,Industrial%20Ethernet%20and%20fieldbus%20organization>
- HADES. *Relay-based AND and NAND gates* [online]. [cit. 2023-08-27]. Dostupné z: <https://tams.informatik.uni-hamburg.de/applets/hades/webdemos/05-switched/20-relays/and-nand.html>
- HADES. *Relay-based OR and NOR gates* [online]. [cit. 2023-08-27]. Dostupné z: <https://tams.informatik.uni-hamburg.de/applets/hades/webdemos/05-switched/20-relays/or-nor.html>
- IJS. *Referenční model ISO/OSI* [online]. [cit. 2023-08-27]. Dostupné z: http://ijs2.8u.cz/index.php?option=com_content&view=article&id=13&Itemid=119
- KOLBABA, V., 2004. *Automa – časopis pro automatizační techniku* [online]. Ústí nad Labem [cit. 2023-08-27]. Dostupné z: https://automa.cz/cz/casopis-clanky/simatic-wincc-flexible-vizualizace-od-mikropanelu-az-po-pc-2004_06_32374_1914/ ISSN 1210-9592.
- KREJČÍŘÍK, A., 2006. *Solid State relé*. Praha: BEN – technická literatura. ISBN 80-730-0081-4.
- LITTMAN, M., *The Morse system* [online]. [cit. 2023-08-27]. Dostupné z: <https://commons.princeton.edu/josephhenry/relay/>
- MARTIN, M. *MQTT na zařízeních Weintek* [online]. TECON spol. s r. o [cit. 2023-08-27]. Dostupné z: https://www.tecon.cz/pdf/Prirucka_MQTT.pdf
- MESIDAS. *SIMATIC WinCC Runtime Professional (SCADA)* [online]. [cit. 2023-08-27]. Dostupné z: <https://mesidas.com/wincc-runtime-professional/>
- MICHALEC, L., 2020. *EtherNet / IP versus PROFINET* [online]. [cit. 2023-08-27]. Dostupné z: <https://automatizace.hw.cz/ethernet-ip-versus-profinet.html>
- Mitsubishi, 2006. *Wayback machine* [online]. [cit. 2023-08-27]. Dostupné z: <https://web.archive.org/web/20170829002700/http://www.plcsystems.ru/catalog/Mitsubishi/doc/FX1S%20FX1N%20FX2N%28C%29%20FX3U%20Beginner's%20Manual.pdf>
- MOHANAN, V., 2023. *CIRCUITSTATE. MAX485-CD4069 RS-485 Module with Auto Data Direction Control – Pinout Diagram & Reference* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.circuitstate.com/pinouts/max485-cd4069-rs-485-module-with-auto-data-direction-control-pinout-diagram-and-pin-reference/>
- MOSFET, 2023. *In: Wikipedia: the free encyclopedia*. [online]. San Francisco (CA): Wikimedia Foundation, 2001-2023. Dostupné z: https://cs.wikipedia.org/wiki/MOSFET#/media/Soubor:Scheme_of_metal_oxide_semiconductor_field-effect_transistor.svg. [cit. 2023-08-27].
- OLEJÁR, M., 2015. *Programovanie PLC*. Slovenská poľnohospodárska univerzita. ISBN 978-80-552-1873-1.

- OMRON, *Technical Explanation for Solid-state Relays* [online]. [cit. 2023-08-27]. Dostupné z: https://www.ia.omron.com/data_pdf/guide/18/ssr_tg_e_9_2.pdf
- PAPOUCH, 2023. *Relé na lištu DIN 35 mm* [online]. [cit. 2023-08-27]. Dostupné z: <https://papouch.com/rele-na-listu-din-35-mm-p4670/>
- PETERKA, J., 1992. *Referenční model ISO/OSI – sedm vrstev*. Computerworld [online]. (13/92) [cit. 2023-08-27]. Dostupné z: <https://www.earchiv.cz/a92/a213c110.php3>
- PETERKA, J., 2021. *Lekce 4: Ethernet II* [online]. [cit. 2023-08-27]. Dostupné z: https://www.earchiv.cz/l226/gifs1/NSWI021v4_4.pdf
- POČÍTAČOVÉ SÍTĚ, 2010. *ISO/OSI*. [online] Dostupné z: <https://site-pc.webnode.cz/model-iso-osi/>. [cit. 2023-08-27].
- QUO POČÍTAČE. *Synology DS218+ DiskStation* [online]. [cit. 2023-08-27]. Dostupné z: <https://shop.quo-pocitace.cz/detail/Synology-DS218-DiskStation/479966?zoneId=>
- RADVANSKY, R.; BRODSKY, J. 2013. *Handbook of SCADA/Control Systems Security*. CRC Press. ISBN 978-1466502260.
- Relay, 2023. In: *Wikipedia: the free encyclopedia*. [online]. San Francisco (CA): Wikimedia Foundation, 2001-2023. Dostupné z: <https://en.wikipedia.org/wiki/Relay>. [cit. 2023-08-27].
- Rockwell Automation. *Micro810 Programmable Logic Controller Systems* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.rockwellautomation.com/en-us/products/hardware/allen-bradley/programmable-controllers/micro-controllers/micro800-family/micro810-controllers.html>
- RONĚŠOVÁ, A., 2005. *Řehled protokolu MODBUS* [online]. [cit. 2023-08-27]. Dostupné z: <http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>
- RS-ONLINE. *PLC procesor řada SIMATIC S7-1200, výstup: Digitální, relé Ethernet 14 vstupů/výstupů 1 MB 8 6 Lišta DIN, Nástěnná* [online]. [cit. 2023-08-27]. Dostupné z: <https://cz.rs-online.com/web/p/plc-programovatelne-logicke-kontrolery/8624461>
- Schneider Electric, 2019. *Modicon má šťavnatý životopis – vynález odstartoval dějiny automatizace* [online]. [cit. 2023-08-27]. Dostupné z: <https://elektro.tzb-info.cz/19075-modicon-ma-stavnaty-zivotopis-vynalez-odstartoval-dejiny-automatizace>
- SIEMENS, 2023. *SIMATIC S7-1200* [online]. [cit. 2023-08-27]. Dostupné z: <https://www.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html#:~:text=The%20SIMATIC%20S7%2D1200%20controller,a%20whole%20Orange%20of%20applications>
- ŠMEJKAL, L., MARTINÁSKOVÁ, M., 1999. *PLC a automatizace 1*. Praha: BEN – technická literatura. ISBN 80-860-5658-9.

- ŠMEJKAL, L., 2005. *PLC a automatizace 2*. Praha: BEN – technická literatura. ISBN 80-730-0087-3.
- ŠVABÍK, J., 2017. Typy komunikací: spolehlivá vs. nespolehlivá, spojená vs. nespojená, virtuální sítě, deterministický vs. nedeterministický přístup k sítím, vícenásobné (MA) přístupy k přenosu, příklady, Ethernet, xDSL [online]. [cit. 2023-08-27]. Dostupné z: <https://jansvabik.cz/matur/postvy/11.pdf>
- TECO a.s., 2007. *Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic*. (10). Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz
- ZUTWARGOWIEC. *Transporter materiałów sypkich MFS-40 przed naprawą*. [online] Dostupné z: <http://zutwagrowiec.com.pl/oferta/naprawa-i-modernizacja-maszyn-torowych/transporter-materialow-sypkich-mfs-40-2/>. [cit. 2023-08-27].

PŘÍLOHY

A – CD

Příloha k diplomové práci

Akviziční a kontrolní jednotka programovatelného automatu (SCADA)

Bc. Michal Florián

CD

OBSAH

1. Text diplomové práce ve formátu PDF
2. Schémata zapojení včetně návrhu desky plošných spojů
3. Zdrojový kód programu PLC i jednotky
4. Technický náčrt krytu jednotky
5. Soubory 3D modelu krytu jednotky ve formátu STL
6. Kompletní vývojový diagram programu akviziční a kontrolní jednotky