



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODBUS/TCP SERVER

MODBUS/TCP SERVER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Oskar Kaufman

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2023

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Oskar Kaufman
ID: 230093

Ročník: 3
Akademický rok: 2022/23

NÁZEV TÉMATU:

Modbus/TCP server

POKYNY PRO VYPRACOVÁNÍ:

- 1) Provedte literární rešerši o komunikačních možnostech řídicího systému LOGIX.
- 2) Podle návrhu realizujte prototyp Modbus/TCP serveru a DPS.
- 3) Vytvořte koncepci potřebného firmware pro mikrokontroler a implementujte funkce Modbusu s diagnostikou.
- 4) Vytvořte jednoduchou testovací aplikaci s PLC řady LOGIX.
- 5) Ověřte své řešení.

DOPORUČENÁ LITERATURA:

Modbus Messaging Implementation Guide. Modbus Organization.
2006.

https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Radek Štohl, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Táto bakalárska práca sa zaoberá implementáciou priemyselného komunikačného protokolu Modbus TCP a komunikáciou s PLC rady Logix od Allen-Bradley. Cieľom práce bolo vytvoriť prototyp hardvéru, implementovať komunikačný protokol do mikrokontroléru a overiť funkčnosť samotného zariadenia a jeho komunikácie cez Modbus TCP.

Vytvorené zariadenie je vybavené digitálnymi vstupmi a výstupmi, ktoré je možné ovládať pomocou protokolu Modbus TCP. Overenie komunikácie sa zameriava na správne ovládanie týchto vstupov a výstupov prostredníctvom PLC rady Logix.

Kľúčové slova

Firmware, ESP32-POE, Modbus TCP, priemyselná komunikácia, I2C, digitálne vstupy a výstupy, PLC, Logix, ethernet

Abstract

This bachelor's thesis focuses on the implementation of the industrial communication protocol Modbus TCP and communication with Allen-Bradley's Logix PLC series. The aim of this work was to create a hardware prototype, implement the communication protocol into a microcontroller, and verify the functionality of the device itself and its communication through Modbus TCP.

The developed device is equipped with digital inputs and outputs that can be controlled using the Modbus TCP protocol. The verification of communication focuses on the proper control of these inputs and outputs through the Logix PLC series.

Keywords

Firmware, ESP32-POE, Modbus TCP, industrial communication, I2C, digital inputs and outputs, PLC, Logix, ethernet

Bibliografická citácia

KAUFMAN, Oskar. Modbus/TCP server [online]. Brno, 2023 [cit. 2023-05-19].
Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/151640>. Bakalářská práce.
Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
Ústav automatizace a měřicí techniky. Vedoucí práce Radek Štohl.

Prehlásenie autora o pôvodnosti diela

Meno a priezvisko: *Oskar Kaufman*

VUT ID študenta: *230093*

Typ práce: *Bakalárska práca*

Akademický rok: *2022/23*

Téma záverečnej práce: *Modbus/TCP server*

Prehlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúceho záverečnej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si naplno vedomí následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákonníka č. 40/2009 Sb.

V Brne dňa: 17. mája 2023

podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Radekovi Štohlovi Ph.D. za odbornú pomoc a rady pri spracovaní práce.

V Brne dňa: 17. mája 2023

podpis autora

Obsah

ZOZNAM OBRÁZKOV	9
ZOZNAM TABULIEK	10
ÚVOD.....	11
1. PRIEMYSELNÁ KOMUNIKÁCIA.....	12
1.1 NÁROKY NA KOMUNIKAČNÉ ZBERNICE	12
1.2 SÉRIOVÉ KOMUNIKAČNÉ PROTOKOLY	12
1.2.1 Modbus RTU.....	12
1.2.2 Zbernica CAN.....	12
1.2.3 Profibus.....	13
1.2.4 CC-Link.....	13
1.3 PRIEMYSELNÝ ETHERNET.....	13
1.3.1 Modbus TCP.....	13
1.3.2 Profinet.....	13
1.3.3 EtherCAT.....	14
1.3.4 EtherNet/IP.....	14
1.3.5 Powerlink.....	14
2. KOMUNIKAČNÉ MOŽNOSTI RIADIACEHO SYSTÉMU LOGIX.....	15
2.1 KOMUNIKAČNÉ PROTOKOLY RADY LOGIX	15
2.1.1 Protokol DF1.....	15
2.1.2 Protokol Ethernet/IP.....	16
2.1.3 Protokol Modbus.....	16
2.1.4 DeviceNet.....	16
2.1.5 Protokol ASCII.....	16
2.1.6 Protokol DH+.....	17
2.1.7 Protokol DH485.....	17
3. MODBUS	18
3.1 MODBUS RTU.....	18
3.2 MODBUS TCP	19
3.2.1 Klient/server model (master / slave).....	19
3.2.2 Dátový rámec protokolu Modbus TCP/IP.....	20
3.2.3 Popis hlavička MBAP.....	20
4. HARDVÉR ZARIADENIA	22
4.1 POPIS ZARIADENIA	22
4.1.1 Blokové schémy.....	22
4.2 ZÁKLADNÁ DOSKA	23
4.2.1 Výber procesoru.....	23
4.2.2 ESP32.....	24
4.2.3 OLIMEX ESP32-POE.....	24
4.2.4 Prepínanie adresy.....	25
4.2.5 Indikácia na doske plošného spoja.....	26
4.2.6 Zdrojová časť.....	26

4.2.7	<i>Prevodník napät'ovej úrovne</i>	28
4.2.8	<i>Procesorová časť</i>	29
4.2.9	<i>Návrh DPS</i>	30
4.2.10	<i>Popis dosky a svorkovnic – základná doska</i>	30
4.3	ROZŠIRUJÚCA DOSKA PRE DIGITÁLNE VSTUPY A VÝSTUPY	31
4.3.1	<i>Digitálne vstupy</i>	31
4.3.2	<i>Digitálne výstupy</i>	32
4.3.3	<i>Návrh DPS</i>	33
4.3.4	<i>Popis dosky a svorkovnic – rozširujúca doska</i>	34
5.	FIRMWARE	35
5.1	KOMUNIKÁCIA ZÁKLADNEJ DOSKY S ROZŠIRUJÚCIMI DOSKAMI	35
5.1.1	<i>Komunikačný protokol I2C</i>	35
5.1.2	<i>Implementácia I2C</i>	35
5.2	PRIPOJENIE DO LOKÁLNEJ SIETE	37
5.3	KOMUNIKÁCIA CEZ MODBUS TCP	38
5.3.1	<i>Implementované funkcie Modbusu</i>	38
5.3.2	<i>Nastavenie výstupov a vstupov</i>	38
6.	TEST KOMUNIKÁCIE	39
6.1	PROGRAM QMODMASTER	39
6.2	CODESYS A VIRTUÁLNE PLC	40
6.2.1	<i>Konfigurácia pre Modbus TCP server</i>	41
6.2.2	<i>Vizualizácia testovacieho programu</i>	42
6.3	TESTOVACIA APLIKÁCIA NA REÁLNO M PLC	42
6.3.1	<i>Nastavenie požiadavky na server</i>	44
6.3.2	<i>Diagnostika chýb pri komunikácii</i>	44
6.3.3	<i>Vizualizácia komunikácie</i>	45
7.	ZÁVER	46
	LITERATÚRA	47
	ZOZNAM SYMBOLOV A SKRATIEK	49
	ZOZNAM PRÍLOH	50

ZOZNAM OBRÁZKOV

Obrázok 1 Komunikácia v automatizovanom systéme pre rady Logix [Príloha A , str. 39].....	15
Obrázok 2 Allen Bradley ControlLogix 5580 komunikujúci cez Ethernet/IP [5].....	16
Obrázok 3 Komunikácia PLC Logix [6].....	17
Obrázok 4 Typy správ protokolu Modbus TCP/IP [9].....	19
Obrázok 5 Základný rámec protokolu Modbus TCP [9].....	20
Obrázok 6 Rámec s hlavičkou MBAP pre Modbus TCP [9]	20
Obrázok 7 Prototyp - Modbus TCP server [Príloha G].....	22
Obrázok 8 Bloková schéma zariadenia	22
Obrázok 9 Hlavné časti zariadenia	23
Obrázok 10 Procesor ESP32 s anténou WIFI [10].....	24
Obrázok 11 Vývojová doska OLIMEX EPS32-POE [11].....	24
Obrázok 12 ESP32-POE výstupné piny [12].....	25
Obrázok 13 Možnosť opravy prepínania adresy.....	26
Obrázok 14 Schéma zapojenia stabilizátora 5V [Príloha B].....	27
Obrázok 15 Schéma zapojenia stabilizátora 3,3V [Príloha B].....	28
Obrázok 16 Odstránenie prevodníka napät'ovej úrovne [Príloha B].....	28
Obrázok 17 Odstránenie prevodníka napät'ovej úrovne na DPS	29
Obrázok 18 Pripojenie vývojovej dosky ESP32-POE [Príloha B].....	29
Obrázok 19 Návrh DPS - procesorová časť [Príloha C]	30
Obrázok 20 Popis dosky a svorkovnic - základná doska	30
Obrázok 21 Pripojenie vstupov a výstupov do Arduina Nano [Príloha B].....	31
Obrázok 22 Vstupný obvod [Príloha B]	32
Obrázok 23 Výstupný obvod [Príloha B]	33
Obrázok 24 Návrh DPS - Rozširujúca doska [Príloha B].....	33
Obrázok 25 Popis dosky a svorkovnic - rozširujúca doska.....	34
Obrázok 26 Nastavenie pinov I2C pre ESP32-POE	35
Obrázok 27 Požiadavka na zápis v zbernici I2C.....	36
Obrázok 28 Požiadavka na čítanie v zbernici I2C	36
Obrázok 29 Inicializácia pinov pre LAN8710A	37
Obrázok 30 Schéma zapojenia ethernetového modulu LAN8710A [Príloha D].....	37
Obrázok 31 Príkaz pre spustenie Modbus TCP serveru.....	38
Obrázok 32 Výpis z programu Wireshark	39
Obrázok 33 Program QModMaster a výpis komunikácie.....	40
Obrázok 34 Nastavenie komunikácie s Modbus TCP serverom.....	41
Obrázok 35 Nastavenie požiadaviek na server	41
Obrázok 36 Priradenie premenných na adresy registrov	42
Obrázok 37 Vizualizácia testovacieho programu v Codesyse [Príloha F].....	42
Obrázok 39 Program pre Modbus TCP v periodickom tasku	43
Obrázok 38 Add-on inštrukcia pre klient.....	43
Obrázok 40 Nastavenie požiadaviek - Allen Bradley	44
Obrázok 41 Globálna diagnostika pre Modbus TCP klienta.....	44
Obrázok 42 Stav a parametre požiadaviek.....	45
Obrázok 43 Client status.....	45
Obrázok 44 Hodnoty prvkov v registroch.....	45

ZOZNAM TABULIEK

Tabuľka 1 Správa protokolu Modbus RTU [8].....	18
Tabuľka 2 Registre protokolu Modbus [8]	18
Tabuľka 3 Hlavička MBAP [9, str. 5].....	21
Tabuľka 4 JP1 - popis pinov	30
Tabuľka 5 JP1 a JP2 - popis pinov.....	34
Tabuľka 6 JP3 - popis pinov	34

ÚVOD

Táto bakalárska práca sa zaoberá priemyselným komunikačným protokolom Modbus TCP, s dôrazom na PLC rady Logix od Allen-Bradley. Cieľom je vytvoriť prototyp hardvéru, implementovať komunikačný protokol do zariadenia a overiť funkčnosť samotného zariadenia a jeho komunikácie cez Modbus TCP.

Zariadenie je vybavené digitálnymi vstupmi a výstupmi, ktoré je možné ovládať pomocou protokolu Modbus TCP. Overenie komunikácie sa zameriava na správne ovládanie týchto vstupov a výstupov prostredníctvom PLC rady Logix.

Práca poskytuje prínos v oblasti implementácie protokolu Modbus TCP na mikrokontroléroch, konkrétne na platforme ESP32-POE. Využitie existujúcej knižnice umožňuje efektívne a spoľahlivo realizovať komunikáciu cez Modbus TCP.

Pred samotným testovaním na reálnom PLC rady Logix od Allen-Bradley, bolo potrebné overiť správnosť implementácie pomocou programov, ktoré simulujú Modbus TCP klienta a virtuálne PLC. Táto simulácia poskytuje priblíženie sa k reálnemu prostrediu a umožňuje overiť funkčnosť a spoľahlivosť systému pred jeho testovaním na skutočnom hardvéri.

Okrem samotnej funkčnosti zariadenia, bola pozornosť venovaná aj diagnostike komunikácie. Tá bola testovaná na reálnom PLC rady Logix. Bola overená spoľahlivosť a efektívnosť komunikácie a vyhodnotené správne fungovanie diagnostických nástrojov.

1. PRIEMYSELNÁ KOMUNIKÁCIA

Komunikácia medzi rôznymi systémami a perifériami v priemysle je veľmi dôležitá, zvyšuje efektivitu a produktivitu vďaka overeným štandardom a šetrí množstvo potrebného hardware (vodičov), nakoľko sa informácie prenášajú pomocou konkrétnych protokolov v sérii. Komunikácia v priemysle musí byť bezpečná, spoľahlivá a hlavne odolná voči rušeniu z okolitého prostredia. [1]

1.1 Nároky na komunikačné zbernice

- Prenos údajov v reálnom čase,
- odolnosť voči rušivým vplyvom prostredia, kde sa komunikačná zbernica nachádza,
- odstránenie novej chyby, napr. *redundancia*,
- štandardizované komunikačné protokoly, ktoré fungujú pre všetkých výrobcov [2]

1.2 Sériové komunikačné protokoly

V sériovej komunikácii sa správy odosiľajú v podobe postupnosti bitov, kde každá správa môže obsahovať určený počet bitov pre dáta, riadiace bity a kontrolné súčty na zabezpečenie integrity prenesených dát. Správy môžu byť jednosmerné (odosielané z jedného zariadenia na druhé) alebo obojsmerné (s oboma zariadeniami, ktoré môžu posielat' a prijímat' správy). [3]

1.2.1 Modbus RTU

Open-source protokol, ktorý dokáže zabezpečiť komunikáciu z 247 zariadeniami, ktoré sú zapojené za sebou. Dokáže komunikovať rýchlosťou 115 kb/s a to po komunikačnej zbernici RS-232, RS-422 a RS-485. Tento protokol bude ešte podrobnejšie vysvetlený v ďalšej časti práce 1.2.1. [3]

1.2.2 Zbernica CAN

Spoľahlivý komunikačný protokol vyvinutý do automobilového priemyslu, ale našiel využitie aj iných oblastiach priemyslu. Kvôli jeho rýchlosti sa používa aj na komunikáciu medzi frekvenčnými meničmi a PLC pri polohovaní. Zbernica CAN môže využívať CANopen protokol a DeviceNet protokol, ktoré sú štandardizované na túto zbernicu. Rýchlosť komunikácie je až 1Mb/s. CANopen protokol dokáže obsluhovať 127 zariadení nakoľko DeviceNet je obmedzený na 64 zariadení. [3]

1.2.3 Profibus

Rýchly sériový protokol, ktorý zabezpečuje komunikáciu procesných a pomocných dát medzi zariadeniami do rýchlosti až 12 Mb/s. Umožňuje komunikáciu so 126 zariadeniami. Zvyčajne ho môžeme spoznať podľa fialového káblu a špeciálneho konektora. [3]

1.2.4 CC-Link

Protokol vyvinutý firmou Mitsubishi, rozšírený v Japonsku a po celej Ázii. Pracuje na zbernici RS-485 a komunikuje maximálne so 64 zariadeniami do rýchlosti 10 Mb/s. [3]

1.3 Priemyselný Ethernet

Komunikácia cez priemyselný Ethernet sa týka prenosu dát v priemyselných prostrediach pomocou ethernetových sietí. Ethernet je bežný štandard pre prenos dát v počítačových sieťach a priemyselný Ethernet je jeho aplikácia vo výrobnom a priemyselnom prostredí. [3]

1.3.1 Modbus TCP

Rozšírenie protokolu Modbus. Tento protokol umožňuje zasielanie správ cez TCP / IP na Ethernete. Hoci je jednoduchá implementácia na štandardnej sieti Ethernet, časová a deterministická komunikácia nie je zaručená, pri komunikácii môže vzniknúť časové oneskorenie. [3]

1.3.2 Profinet

Je široko používaný hlavnými výrobcami zariadení v priemyselnom Ethernete. Aby bolo možné zaručiť požadovaný výkon siete musí byť správne vybraná topológia systému. Topológia môže byť typu Branch alebo Star. Delí sa na tri triedy.

Trieda A používa vzdialené volanie procedúr TCP/IP a vytvára most medzi rozhraním Ethernet a *PROFIBUS*, kedy proxy zaistí prístup k sieti *PROFIBUS*. Používa sa väčšinou na nastavenie parametrov a cyklickú obsluhu vstupov/výstupov, kedy cyklus trvá približne 100 ms. *PROFINET* triedy A sa obvykle používa pre infraštruktúru a automatizáciu budov.

Trieda B má dobu cyklu približne 10 ms a je založený na softvéri, ktorý pracuje v reálnom čase. Trieda B je tiež označovaná ako *PROFINET Real-Time (PROFINET RT)*. Tento protokol je možné použiť v aplikáciách typu PLC. Obvykle sa používa v továrni a automatizácii procesov.

Trieda C vyžaduje špeciálny hardvér, aby bolo možné skrátiť dobu cyklu na menej ako 1 ms, čo mu umožní dostatočne prenášať výkon v reálnom čase pre operácie riadenia pohybu. [3]

1.3.3 EtherCAT

Umožňuje spracovanie paketov pre aplikácie v automatizácii v reálnom čase. Je optimalizovaný pre procesné dáta a používa štandardné rámce IEEE 802.3 Ethernetu. Umožňuje komunikáciu s 65 535 uzlami systému. [3]

1.3.4 EtherNet/IP

Protokol aplikačnej vrstvy, ktorý používa protokol *Common Industrial Protocol* (CIP) cez protokol TCP/UDP. CIP je možné používať vo viacerých fyzických médiách, ako je zbernica CAN DeviceNet či ControlNet alebo Ethernet (EtherNet/IP). EtherNet/IP môže mať neobmedzený počet uzlov systému, pretože používa štandardný ethernet a switche. Hoci je kompatibilný s radom štandardných internetových a ethernetových protokolov, má obmedzené schopnosti v reálnom čase. [3]

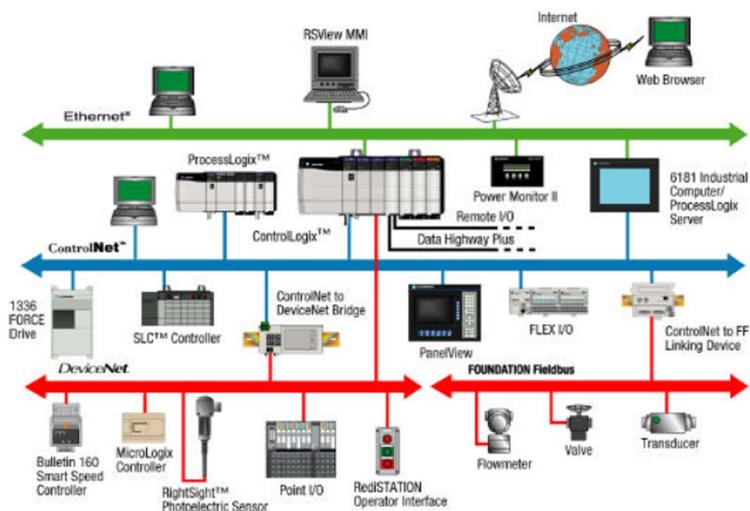
1.3.5 Powerlink

Umožňuje voľnú voľbu topológie siete, pretože beží nad IEEE 802.3. Tento protokol využíva špeciálnu techniku dotazovania, ktorá umožňuje výmenu dát v reálnom čase. Je vhodný pre rad automatizačných systémov od *PLC to PLC* komunikáciu na riadenie pohybu a ovládanie vstupov / výstupov. [3]

2. KOMUNIKAČNÉ MOŽNOSTI RIADIACEHO SYSTÉMU LOGIX

Programovateľné automaty rady Logix značky Allen Bradley, ktoré aktuálne patria firme Rockwell Automation majú rôzne možnosti komunikácie medzi ďalšími zariadeniami v automatizovanom systéme. PLC rady Logix dokážu komunikovať sériovými protokolmi, ale aj protokolmi založenými na ethernete.

Prepojenie s komunikačnými sieťami je modulárne, s výnimkou tzv. sériového portu RS-232 (protokol DF1) zabudovaného v prednej časti PLC Logix. Samostatné komunikačné moduly sú prepojené cez Ethernet, ControlNet, DeviceNet alebo DH+ s PLC. [4]



Obrázok 1 Komunikácia v automatizovanom systéme pre rady Logix [Príloha A , str. 39]

2.1 Komunikačné protokoly rady Logix

2.1.1 Protokol DF1

Ide o komunikačný protokol určený pre klasickú sériovú linku (RS232/RS422). Protokol umožňuje komunikáciu *Full-duplex* (typ spojenia point to point, po zbernici RS232) alebo *Half-duplex* (Master to Slave) s adresáciou PLC.

Tento protokol podporujú automaty, ktoré pracujú s dátovými oblasťami (DataFile), napríklad: SLC500 (obr. 3), MicroLogix, atď. [4]

2.1.2 Protokol Ethernet/IP

Je vybudovaný na architektúre NetLinx, ktorý je implementovaný aj v sieťach DeviceNet a ControlNet. Ethernet/IP je spravovaný nezávislou organizáciou ODVA, ktorá má na starosti aj DeviceNet.

Tento protokol podporujú automaty, ktoré pracujú s dátovými objektmi (na rozdiel od dátových oblastí pre protokol DF1).

Napríklad sa jedná o PLC: Micro820, ControlLogix (obrázok 2), CompactLogix, SoftLogix, DriveLogix, GuardLogix, atď. [4]



Obrázok 2 Allen Bradley ControlLogix 5580 komunikujúci cez Ethernet/IP [5]

2.1.3 Protokol Modbus

Modbus je štandardný, veľmi často používaný protokol pre Ethernet-klienta a pre sériovú linku (PLC automat môže pracovať ako Master, ale aj ako Slave).

Protokol Modbus RTU podporuje napríklad: Micro820, MicroLogix 1200/1100/1500.

Protokol Modbus TCP podporuje napríklad: Micro820, Micrologix1400. [4]

2.1.4 DeviceNet

Zabezpečuje komunikáciu PLC so vstupno-výstupnými modulmi, bezpečnostnými modulmi, snímačmi alebo akčnými členmi systému. Rýchlosť prenosu dát môže byť 125 kbit/s, 250 kbit/s alebo 500 kbit/s. Komunikácia funguje aj na vzdialenosti väčšie ako 100 metrov. [4]

2.1.5 Protokol ASCII

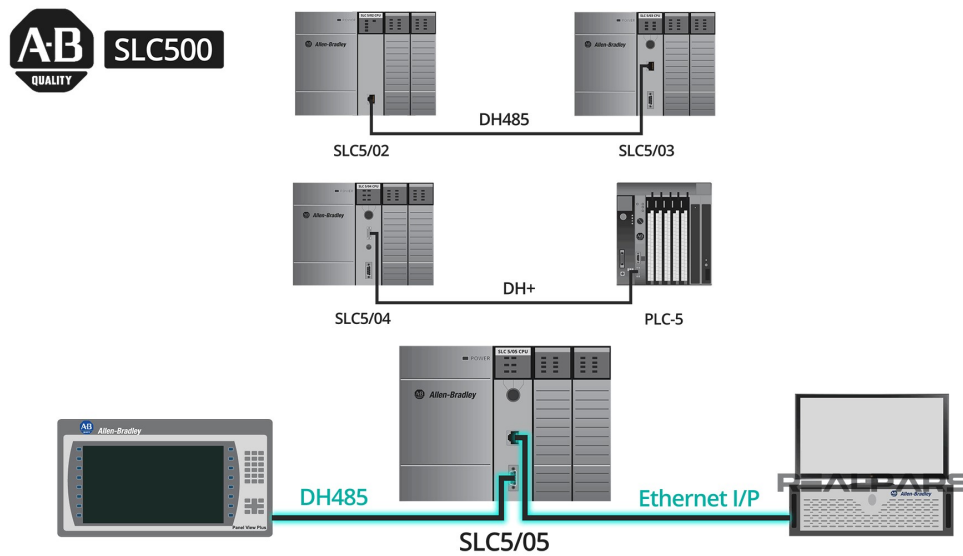
Ide o voľne konfigurovateľný znakový komunikačný protokol určený pre klasickú sériovú linku (RS232/RS422). Tento protokol podporujú niektoré PLC. Napríklad: Micro820, CompactLogix (na druhom porte sériovej linky), MicroLogix 1100/1200/1500, atď. [4]

2.1.6 Protokol DH+

Podporuje max 64 uzlov, 230 kB/s (nevie SLC 501 až 3). Je to interný protokol Allen-Bradley. [4]

2.1.7 Protokol DH485

Ide o komunikačný protokol určený pre sériovú linku. Podporuje max. 32 uzlov, komunikuje rýchlosťou 19 kB/s (multi-master = komunikácia každý s každým). Je to interný protokol Allen-Bradley. [4]



Obrázok 3 Komunikácia PLC Logix [6]

3. MODBUS

Komunikácia Modbus bude vysvetlená podrobnejšie nakoľko je základom tejto práce. Modbus je komunikačný protokol založený na architektúre master/slave alebo klient/server. Hlavným účelom protokolu je zaisťovať jednoduchú, spoľahlivú a rýchlu komunikáciu medzi automatizačnými zariadeniami a zariadeniami na procesnú inštrumentáciu.

Rozlišujeme dva rôzne typy tejto komunikácie Modbus RTU a Modbus TCP. Hoci bol pôvodne vyvinutý pre Schneider Electric Modicon PLC, mnohé značky vrátane Allen Bradley môžu komunikovať cez Modbus, čo z neho robí univerzálny a pomerne často používaný protokol.[7]

3.1 Modbus RTU

Na prenos dát pomocou Modbus RTU sa využívajú rozhrania RS-485, RS-422, RS-232. Správa protokolu Modbus RTU (Tabuľka1) sa skladá z adresy zariadenia SlaveID, kódu funkcie, špeciálnych dát v závislosti od kódu funkcie a kontrolného súčtu CRC. [8]

Tabuľka 1 Správa protokolu Modbus RTU [8]

SlaveID	Kód funkcie	Speciální data	CRC
---------	-------------	----------------	-----

Bez adresy SlaveID a kontrolného súčtu CRC, zostáva PDU (protocol data unit) alebo dátová jednotka protokolu. SlaveID je adresa zariadenia, ktorá môže mať hodnotu od 0 do 247, adresy od 248 do 255 sú vyhradené. Komunikácia funguje tak, že master pošle požiadavku na slave a slave odpovedá. Dáta v module sú uložené v 4 tabuľkách (registroch). Dve tabuľky sú iba na čítanie a dve umožňujú čítanie aj zápis. V každej tabuľke je umiestnených 9999 hodnôt.[8]

Tabuľka 2 Registre protokolu Modbus [8]

ČÍSLO REGISTRU	HEXADECIMÁLNÍ ADRESA REGISTRU	TYP	NÁZEV	TYP
1-9999	0000 až 270E	čtení-zápis	Smyčky diskretního výstupu	DO
10001-19999	0000 až 270E	čtení	Kontakty diskretního vstupu	DI
30001-39999	0000 až 270E	čtení	Registry analogového vstupu	AI
40001-49999	0000 až 270E	čtení-zápis	Holding registry analogového výstupu	AO

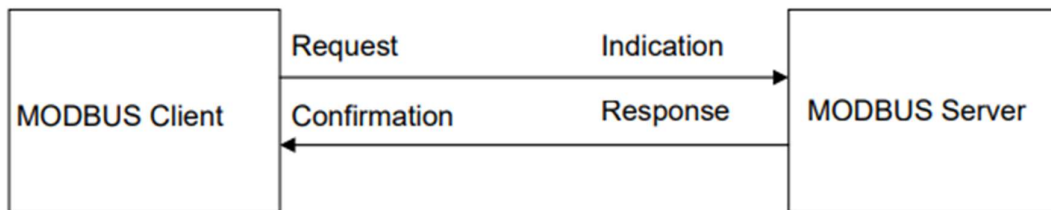
3.2 Modbus TCP

Tento typ komunikácie bude využitý aj v tejto práci na komunikáciu s PLC Logix. V sieti Ethernet je adresou zariadenia jeho IP adresa. Zariadenia sú obvykle umiestnené v rovnakej podsieti, kde sa adresy IP líšia podľa posledných čísič napr. 192.168.1.20 pri použití najbežnejšej masky podsiete 255.255.255.0. Rozhranie je sieť Ethernet, protokol pre prenos dát je TCP/IP. Používaný port TCP je: 502, vyhradený konkrétne pre Modbus TCP. [8]

3.2.1 Klient/server model (master / slave)

Modbus zabezpečuje komunikáciu medzi klientom a serverom pripojeným k sieti Ethernet TCP/IP. Tento model je založený na štyroch typoch správ:

- požiadavka,
- indikácia,
- odpoveď,
- potvrdenie.



Obrázok 4 Typy správ protokolu Modbus TCP/IP [9]

Požiadavka je správa s požiadavkou na server odoslaná klientom.

Indikácia je správa s požiadavkou prijatá na strane servera.

Odpoveď je správa odpovede odoslaná serverom.

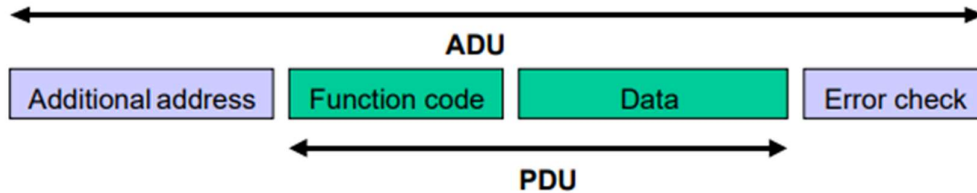
Potvrdenie je správa s odpoveďou prijatá na strane klienta.

Modbus (klient – server model) sa využíva na výmenu informácií v reálnom čase:

- medzi dvoma aplikačnými zariadeniami,
- medzi aplikačným zariadením a iným zariadením,
- medzi SCADA aplikáciami a zariadeniami,
- medzi PC a programom zariadenia poskytujúcim online služby. [9]

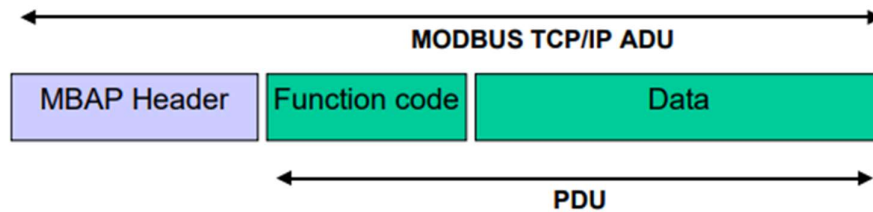
3.2.2 Dátový rámec protokolu Modbus TCP/IP

Existujú dve dátové jednotky. Jednoduchá dátová jednotka protokolu (PDU), ktorá je nezávislá od základných komunikačných vrstiev. Mapovanie protokolu MODBUS na špecifické zbernice alebo siete môže zaviesť niektoré ďalšie polia na aplikačnej dátovej jednotke (ADU). [9]



Obrázok 5 Základný rámec protokolu Modbus TCP [9]

Klient, ktorý iniciuje komunikáciu MODBUS, vytvorí aplikačnú dátovú jednotku (ADU). Funkčný kód indikuje serveru, aký druh akcie má vykonať. Na identifikáciu aplikačnej dátovej jednotky v sieti TCP/IP sa používa vyhradená hlavička nazývaná MBAP. [9]



Obrázok 6 Rámec s hlavičkou MBAP pre Modbus TCP [9]

3.2.3 Popis hlavička MBAP

Je dlhá 7 bajtov. Umožňuje prepojenie komunikácie pomocou routerov, bridgeov a brány v sieti TCP/IP. Všetky požiadavky a odpovede sú riešené tak, že príjemca môže overiť dokončenie správy.

Pre funkčné kódy, kde má PDU pevnú dĺžku, stačí samotný funkčný kód. Ak funkčný kód má premenlivé množstvo dát v žiadosti alebo odpovedi zahrňuje dátové pole počet bajtov.

V hlavičke MBAP sa prenášajú ďalšie informácie o dĺžke. Príjemca je schopný rozpoznať hranice správ aj keď bola správa rozdelená do viacerých paketov. Všetky MODBUS/TCP ADU sa odosielajú prostredníctvom TCP do registrovaného portu 502.

Hlavička MBAP obsahuje nasledujúce polia:

Tabuľka 3 Hlavička MBAP [8]

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

Transaction Identifier (identifikátor transakcie)

Využíva sa pre párovanie transakcie, server Modbus skopíruje identifikátor transakcie požiadavky.

Protocol Identifier (identifikátor protokolu)

Ak zariadenie komunikuje pomocou protokolu Modbus identifikátor protokolu je nastavený na '0'.

Length (dĺžka)

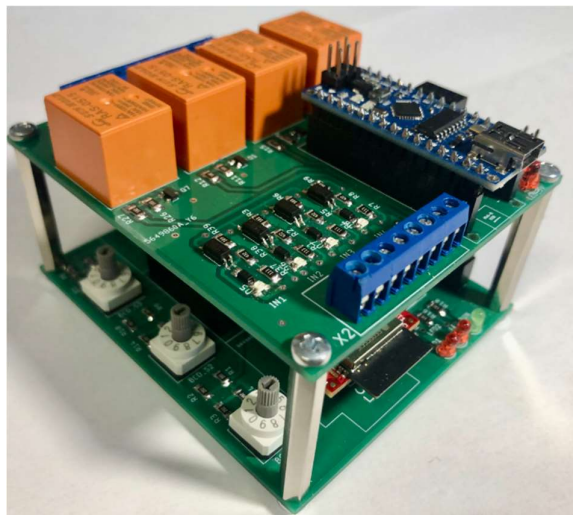
Pole dĺžky je počet bajtov nasledujúcich polí vrátane identifikátora jednotky a dátových polí.

Unit Identifier (identifikátor jednotky)

Toto pole sa používa na účely smerovania v rámci systému. Zvyčajne sa používa na komunikáciu so slave zariadením Modbus, cez bránu medzi sieťou Ethernet TCP/IP a sériovou linkou Modbus. Toto pole je nastavené klientom Modbus v žiadosti a musí byť vrátené s rovnakou hodnotou v odozve serverom.

4. HARDVÉR ZARIADENIA

Zariadenie je schopné prijímať požiadavky Modbus klienta, spracovať ich a následne na základe prijatej správy obsluhovať digitálne vstupy a výstupy. Je napájané štandardne z 24V zdroja a je možné meniť IP adresu zariadenia pri jeho spustení.

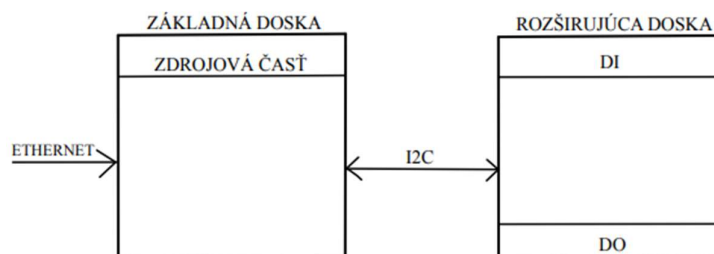


Obrázok 7 Prototyp - Modbus TCP server [Príloha G]

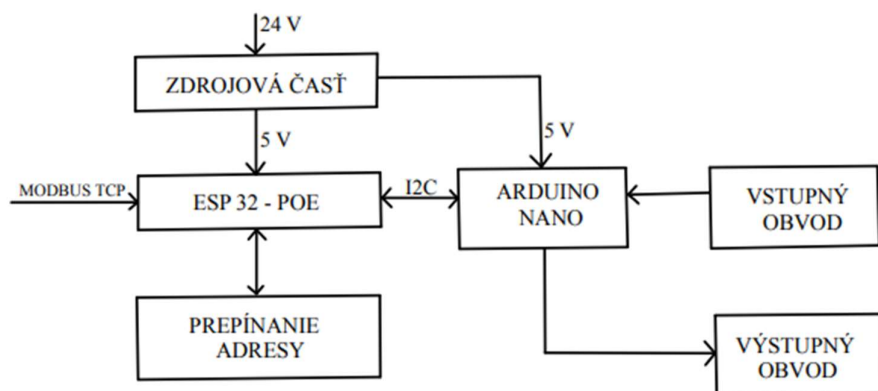
4.1 Popis zariadenia

Zariadenie sa skladá minimálne z dvoch dosiek plošných spojov, ktoré sa dajú rozšíriť o ďalšie rozširujúce dosky s rôznymi funkciami. Dosky sú uložené nad sebou a prepojené pomocou zbernice I2C. Spodná doska obsahuje hlavný procesor, ktorý prijíma požiadavky cez protokol Modbus TCP a následne komunikuje so serverom I2C (obrázok 8), prípadne viacerými servermi.

4.1.1 Blokové schémy



Obrázok 8 Bloková schéma zariadenia



Obrázok 9 Hlavné časti zariadenia

4.2 Základná doska

Jej úlohou je spracovávať požiadavky Modbus TCP klienta, napájať všetky rozširujúce dosky a prepínať adresu Modbus serveru. Kompletná schéma zapojenia sa nachádza v Príloha B .

4.2.1 Výber procesoru

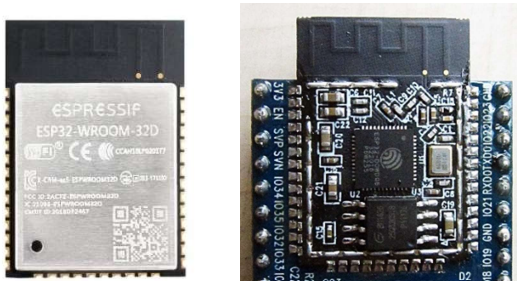
Dôležitou otázkou pri návrhu nového zariadenia je aký bude obsahovať procesor. Cieľom pri návrhu dosky pre Modbus TCP server je možnosť modularity, čiže užívateľ bude mať možnosť rozšírenia funkcií, napr.: zvýšenie počtu vstupov a výstupov, pripojenie externých zariadení na zbernicu I2C, atď.

Modbus TCP môže komunikovať pomocou ethernetu alebo WIFI. Komunikácia pomocou WIFI v priemyselnom systéme nie je úplne bezpečná, pretože na ňu môže pôsobiť rušenie z prostredia. Preferovaná bude teda komunikácia pomocou ethernetu. Prieskum trhu pre výber procesora bude zameraný na tie, ktoré podporujú aj WIFI, v prípade využitia zariadenia na mieste, kde bude možné a bezpečné pomocou WIFI komunikovať.

Po preskúmaní aktuálneho trhu dostupných procesorov bolo rozhodnuté, že najvhodnejším bude ESP32. Je v puzdre spolu s anténou na WIFI a bluetooth, čo je veľmi výhodné. Má dostatočný výpočtový výkon pre realizované zariadenie.

4.2.2 ESP32

ESP32 zahŕňa 32-bitový LX6 mikroprocesor, pracujúci od 160 až 240 MHz, pamäť RAM 320KiB a ROM 448 KiB. Má 34 programovateľných vstupno-výstupných portov, komunikačné rozhranie SPI, I2C, UART. Obsahuje vstavanú zbernicu pre ethernet RMI. [10]

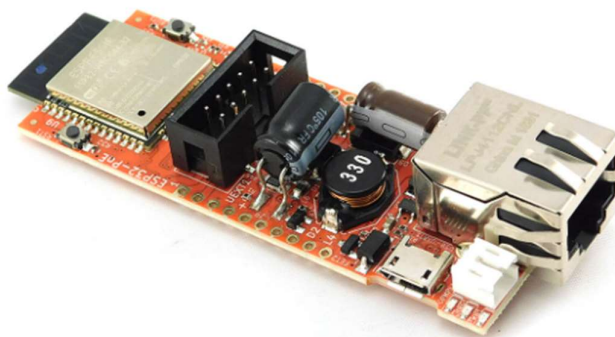


Obrázok 10 Procesor ESP32 s anténou WIFI [10]

Ďalšou otázkou je, či použiť samotný procesor v puzdre alebo vývojovú dosku, ktorá obsahuje už aj periférie, ktoré by sa museli v každom prípade umiestniť na dosku plošného spoja napr.: oscilátor, stabilizátor napätia, prevodník na programovanie, atď. V tejto práci bude určite výhodnejšie použiť vývojovú dosku nakoľko bude vytvárané zariadenie, ktorého funkčnosť nie je overená.

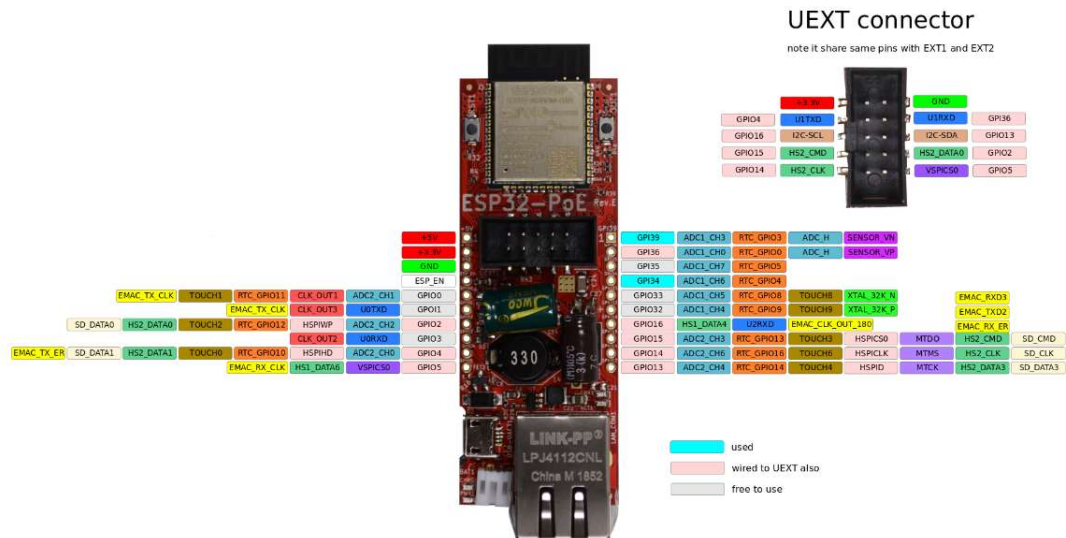
Prieskumom trhu zameraným na vývojové dosky ESP32 bolo zistené, že najideálnejšou možnosťou pre danú prácu je vývojová doska ESP32-POE od výrobcu OLIMEX, ktorá obsahuje ethernetový konektor.

4.2.3 OLIMEX ESP32-POE



Obrázok 11 Vývojová doska OLIMEX EPS32-POE [11]

ESP32-POE je vývojová doska podporujúca WIFI, bluetooth a ethernet. Má funkciu POE – power over ethernet, je možné napájať zariadenie pomocou ethernetu. Na napájanie zariadenia pomocou POE musí byť napätie v ethernetovom kábli aspoň 37V. Doska obsahuje vstupno-výstupné piny pre pripojenie rôznych snímačov a akčných členov.



Obrázok 12 ESP32-POE výstupné piny [12]

Programuje sa cez micro USB konektor, čo je výhodou, pretože nie je potreba externý programátor.

ESP32-POE nemá galvanicky oddelené ethernetové napájanie, pri programovaní dosky cez micro USB konektor, by mal byť ethernetový kábel odpojený, ak je doska napájaná pomocou POE. [13]

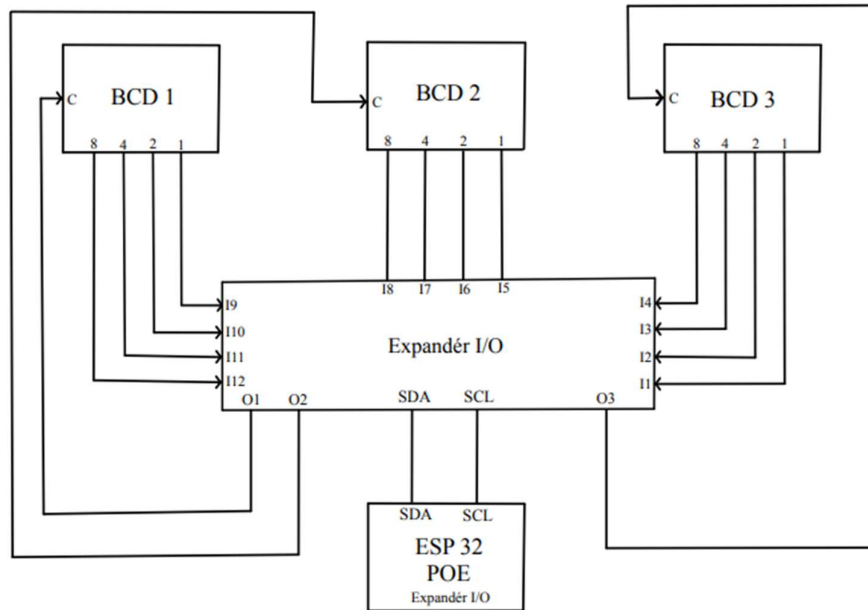
4.2.4 Prepínanie adresy

Požiadavkou na novú dosku pre komunikačné rozhranie bola možnosť prepínania posledných troch číslic IP adresy zariadenia. Riešenie spočíva v troch BCD prepínačoch, na ktorých užívateľ zadá presne adresu posledných troch číslic. Adresa sa na začiatku programu načíta a v priebehu komunikácie už nebude možné adresu meniť, len v prípade reštartu.

Pri testovaní vyčítania adresy pomocou ESP32-POE sme zistili nedostatok v návrhu a tak je možné meniť posledné tri číslice IP adresy len v rozsahu 0 – 9. Na nastavenie požadovanej adresy je potrebné nastaviť adresu len na jednom BCD prepínači, nezáleží, na ktorom prepínači je nastavená adresa, ale ostávajúce dva musia byť nastavené na 0.

Chyba návrhu spočíva v tom, že výstupy všetkých BCD prepínačov sú spojené a pri čítaní, prepnutie jedného prepínača ovplyvňuje výstup druhého. Nemôžeme použiť na vyčítanie metódu multiplexovania.

Opravou návrhu by bolo priviesť všetky výstupy z jednotlivých BCD prepínačov na vstupy ESP32-POE. Bolo by potrebné využiť 3 výstupy a 12 vstupov. Realizované zariadenie nemá dostatok voľných vstupov a tak by bolo potrebné pridať expandér pre vstupy a výstupy (obrázok 13).



Obrázok 13 Možnosť opravy prepínania adresy

4.2.5 Indikácia na doske plošného spoja

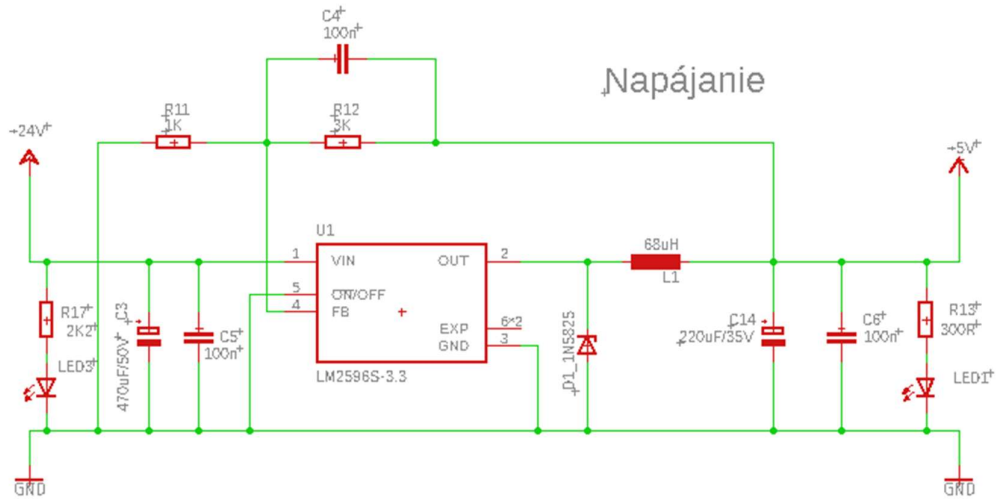
Po pripojení napájacieho napätia do zariadenia sa rozsvietia dve červené LED diódy a jedna zelená, ktoré signalizujú správnu funkciu zdrojovej časti a prítomnosti napájacieho napätia. Jedna LED dióda signalizuje správnu funkciu stabilizátora napätia z 24V na 5V, druhá stabilizátora z 5V na 3,3V a tretia je zelená, ktorá sa nachádza pri napájacích svorkách. Na základnej doske sa nachádza ešte jedna zelená LED dióda, ktorá signalizuje, že Modbus server beží a je pripojený v ethernetovej sieti (obrázok 20).

Rozširujúca doska obsahuje len indikáciu napájania pre 5V, ktorá je zabezpečená červenou LED diódou (obrázok 25).

4.2.6 Zdrojová časť

Napájacie napätie dosky je 24V, napájanie vývojovej dosky s ethernetom je 5V, ale vstupy a výstupy sú len na 3,3V, a taktiež by niektoré rozširovacie dosky mohli byť na 3,3V. Kvôli tomu je na doske stabilizovaný zdroj 5V a 3,3V.

Pre stabilizované napätie 5V použijeme stabilizátor LM2596S, ktorý potrebuje nastaviť správny pomer rezistorov pre 5V výstupné napätie. Tento stabilizátor je pomerne komplikovaný na vyhotovenie na doske plošného spoja, nakoľko potrebuje veľa komponentov pre správne fungovanie. Má však pomerne vysoký výstupný prúd až 3A.



Obrázok 14 Schéma zapojenia stabilizátora 5V [Príloha B]

Je to impulzný zdroj, ktorý využíva základnú vlastnosť cievky. Zmena prúdu cievkou nesmie byť skoková a preto pri rozopnutí sa obvod uzatvára cez diódu D1 ako je vidieť na obrázku 14 a následne sa napätie vyfiltruje na kondenzátore C14. Integrovaný obvod LM2596 je spínač, ktorý výstupné napätie spína na základe spätnej väzby. Spätá väzba je tvorená odporový deličom s rezistormi R11 a R12 ich pomerom nastavujeme výstupné napätie stabilizátora podľa vzorca 1.

Výpočet rezistorov pre stabilizované napätie 5V:

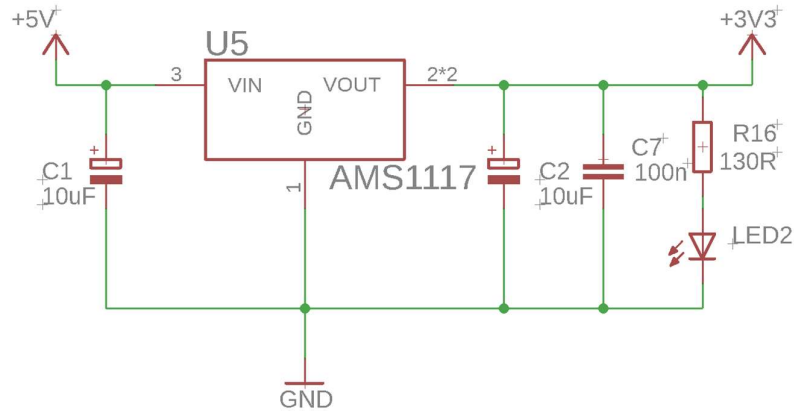
$$U_{OUT} = 1,23 \left(1 + \frac{R12}{R11} \right) \quad (1)$$

R11 bolo zvolené 1 kΩ podľa katalógu [Príloha B], výpočet R12 z rovnice 1:

$$R12 = \left(\frac{U_{OUT}}{1,23} - 1 \right) \cdot R11 = \left(\frac{5}{1,23} - 1 \right) \cdot R11 = 3065,04 \Omega$$

Zvolené bolo R12 = 3 kΩ.

Napätie 3,3V zabezpečuje lineárny stabilizátor AMS1117 z 5V.

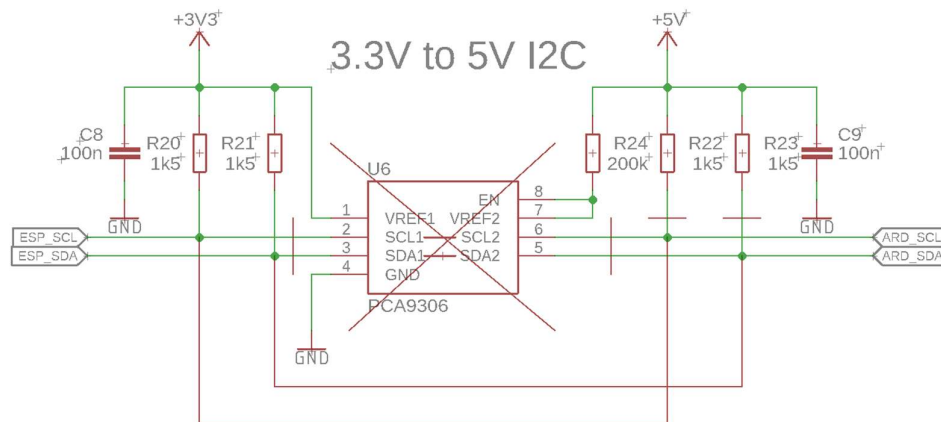


Obrázok 15 Schéma zapojenia stabilizátora 3,3V [Príloha B]

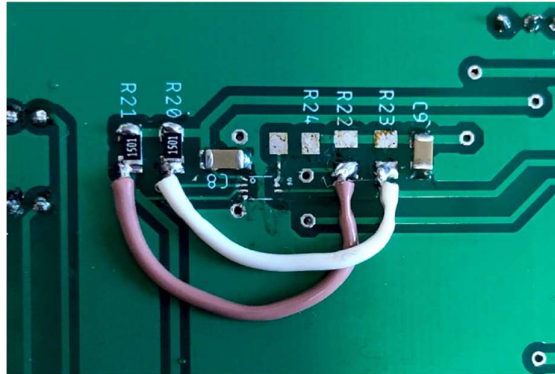
4.2.7 Prevodník napät'ovej úrovne

Rozširujúce periférie zariadenia budú komunikovať, s procesorom na základnej doske, pomocou zbernice I2C. Výstup procesoru, na základnej doske, je 3,3V a výstup na rozširujúcej doske je 5V. Je potrebné prevádzať napät'ové úrovne na jednotlivých doskách pri vzájomnej komunikácii. Bude na to využitý prevodník napät'ovej úrovne PCA9306.

Pri oživovaní výrobku boli s týmto prevodníkom problémy a komunikácia I2C nefungovala. Prevodník bol odstránený (obrázok 16) nakoľko bolo zistené, že I2C komunikácia funguje aj bez prevodníka napät'ovej úrovne. Komunikácia funguje, pretože aktívnou logickou úrovňou pri zbernici I2C je 0V. Neaktívnu úroveň sme ponechali na 3,3V, čo procesor na rozširujúcej doske zaznamenáva ako logické jednotku.



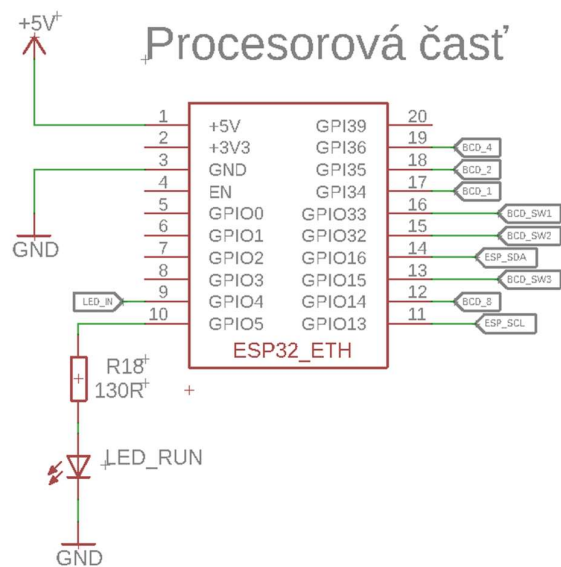
Obrázok 16 Odstránenie prevodníka napät'ovej úrovne [Príloha B]



Obrázok 17 Odstránenie prevodníka napäťovej úrovne na DPS

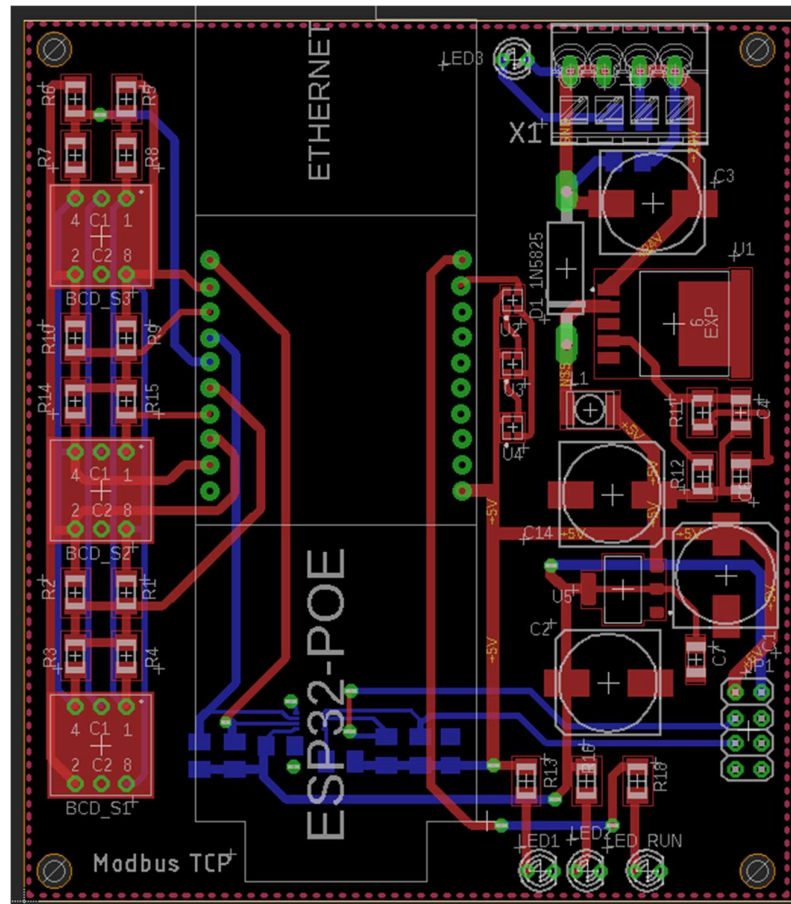
4.2.8 Procesorová časť

Piny GPIO0-GPIO3 sa nemôžu využívať, pretože sa využívajú na programovanie a debugovanie. Zelená LED dióda na GPIO5 signalizuje, že Modbus server beží a zariadenie je pripojené na ethernetovú sieť.



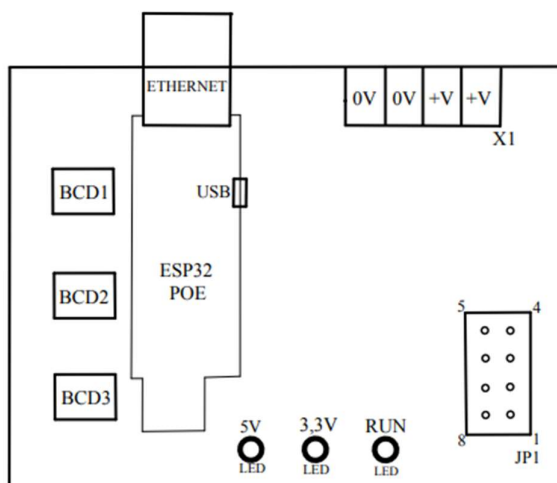
Obrázok 18 Pripojenie vývojovej dosky ESP32-POE [Príloha B]

4.2.9 Návrh DPS



Obrázok 19 Návrh DPS - procesorová časť [Príloha C]

4.2.10 Popis dosky a svorkovnic – základná doska



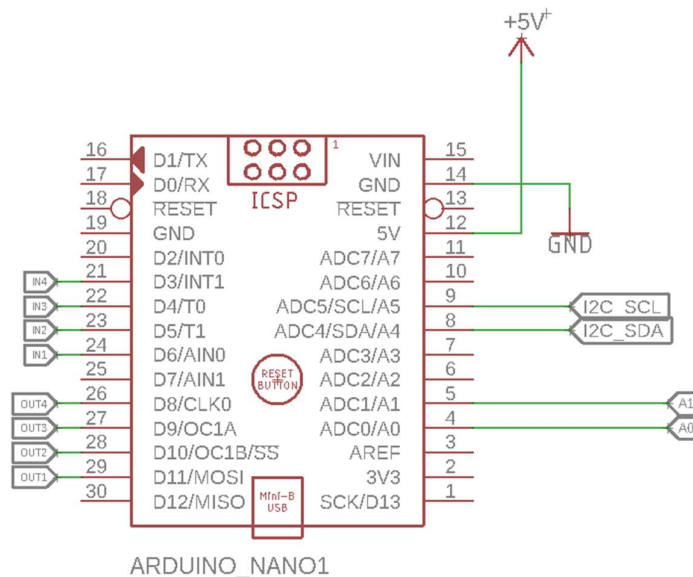
Tabuľka 4 JP1 - popis pinov

1	GND
2	NC
3	NC
4	3,3V
5	5V
6	I2C-SCL
7	I2C_SDA
8	GND

Obrázok 20 Popis dosky a svorkovnic - základná doska

4.3 Rozširujúca doska pre digitálne vstupy a výstupy

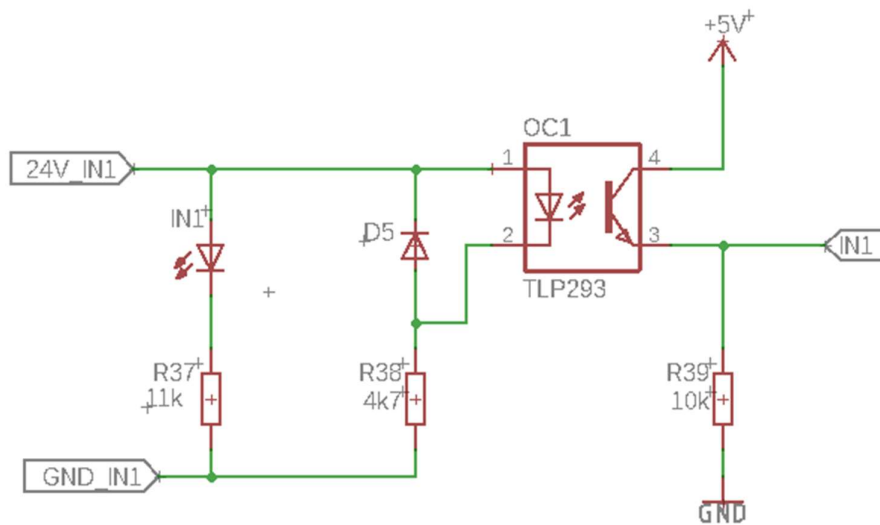
Doska je určená na čítanie digitálnych vstupov a spínanie digitálnych výstupov. Obsahuje obvod digitálnych vstupov a obvod digitálnych reléových výstupov. Obsluhu týchto obvodov zabezpečuje vývojová doska Arduino Nano s procesorom Atmega 328P, ktorá bola vybraná na základe malých rozmerov, pomerne veľkému počtu vstupno-výstupných pinov a možnosti programovania cez USB. Obsahuje taktiež 8 analógových pinov, ktoré možno využívať ako analógový vstup alebo výstup. Dva z nich A0 a A1 sú vyvedené na doske. Kompletná schéma sa nachádza v Príloha B .



Obrázok 21 Pripojenie vstupov a výstupov do Arduina Nano [Príloha B]

4.3.1 Digitálne vstupy

Požiadavkou pre digitálne vstupy je galvanické oddelenie procesoru od vstupu. V tomto prípade je oddelenie zabezpečené optočlenom TLP293. Na vstup procesoru nesmie byť pripojené napätie väčšie ako 5V. Do série so vstupom optočlenu je potrebné pripojiť vhodný rezistor aby znížil napätie na vstupnej dióde. Výstup optočlenu je pripojený priamo do digitálnych vstupov Arduina Nano.



Obrázok 22 Vstupný obvod [Príloha B]

Výpočet veľkosti rezistoru R38:

[Príloha B]

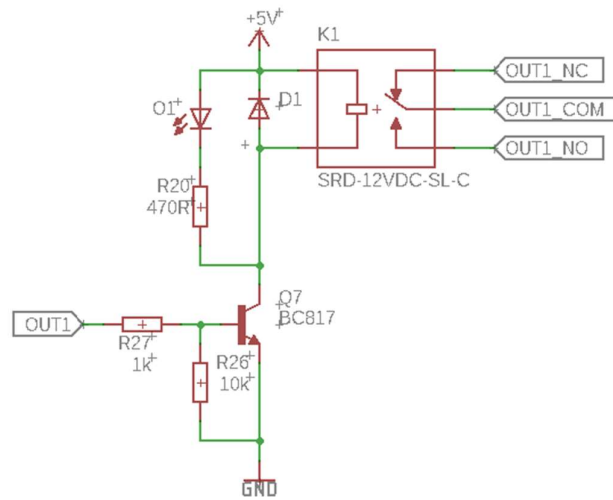
$I_{MAX} = 50 \text{ mA}$	maximálny prúd vstupom optočlenu
$I_D = 5 \text{ mA}$	požadovaný prúd vstupom optočlenu
$U_{IN} = 24 \text{ V}$	vstupné napätie
$U_D = 1,25$	napätie na dióde v priepustnom smere

$$R38 = \frac{U_{IN} - U_D}{I_D} = \frac{24 - 1,25}{5 \cdot 10^{-3}} = 4550 \Omega \quad (2)$$

Bol zvolený $R38 = 4,7 \text{ k}\Omega$.

4.3.2 Digitálne výstupy

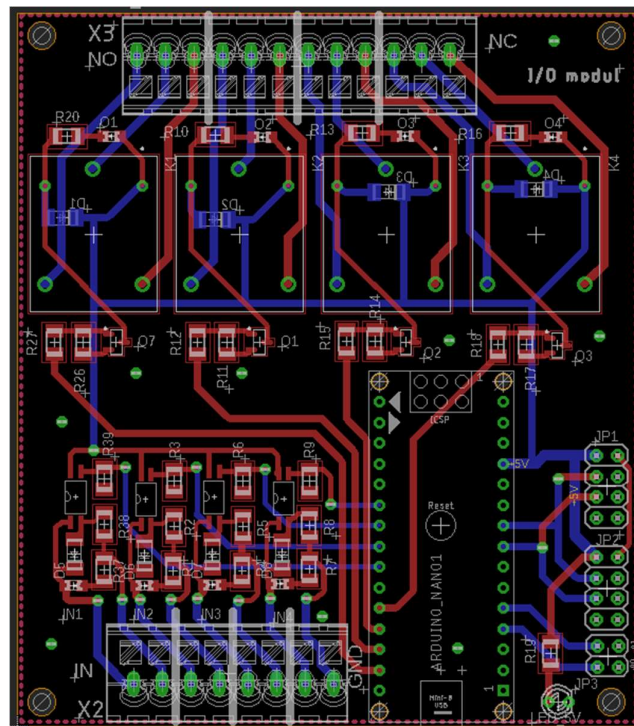
Podobne ako pri vstupnom obvode aj výstupný musí byť galvanicky oddelený od procesora. Výstup je oddelený pomocou relé, to znamená, že výstup nebude mať veľkú spínaciu frekvenciu, ale dokáže spínať veľký prúd. Na spínanie cievky relé je využívaný bipolárny NPN tranzistor BC817, cez ktorý môže tiecť prúd $I_{MAX} = 500 \text{ mA}$.



Obrázok 23 Výstupný obvod [Príloha B]

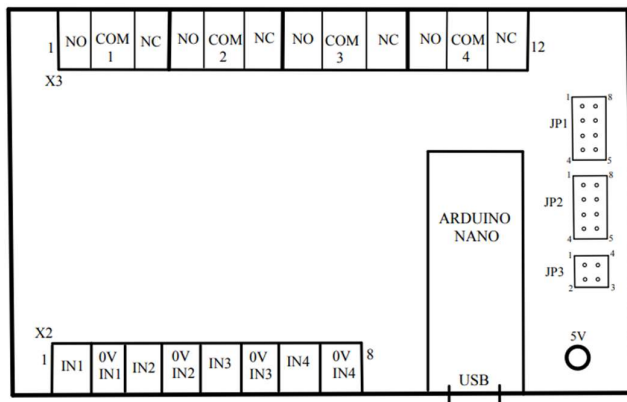
Tranzistor spína Arduino Nano digitálnym výstupom, ktorý je pripojený na delič napätia, ktorý jemne zníži napätie na báze tranzistora, pretože výstup Arduino má 5V a maximálne napätie na báze tranzistora je $U_{BE} = 5V$.

4.3.3 Návrh DPS



Obrázok 24 Návrh DPS - Rozširujúca doska [Príloha B]

4.3.4 Popis dosky a svorkovnic – rozširujúca doska



Obrázok 25 Popis dosky a svorkovnic - rozširujúca doska

Tabuľka 5 JP1 a JP2 - popis pinov

1	5V
2	I2C-SCL
3	I2C-SDA
4	GND
5	GND
6	NC
7	NC
8	3,3V

Tabuľka 6 JP3 - popis pinov

1	GND
2	GND
3	A1
4	A0

5. FIRMWARE

Na tvorbu softwaru bolo využité vývojové prostredie Visual Studio Code s rozšírením Platform IO pre programovanie vývojových dosiek ako je Arduino a ESP32. Použitý programovací jazyk je C++. Vývojové prostredie je prehľadné a veľmi uľahčuje programovanie dopĺňaním slov. Kompletný program s doxygen dokumentáciou pre ESP32-POE a Arduino Nano je v Príloha E .

5.1 Komunikácia základnej dosky s rozširujúcimi doskami

Pre komunikáciu medzi jednotlivými doskami je využitá komunikácia I2C. Na prenos dát využíva dva vodiče. Základná doska je master (klient), bez konkrétnej adresy, a rozširujúce dosky sú všetky slave (server) s konkrétnou adresou.

5.1.1 Komunikačný protokol I2C

Skratka "Inter-Integrated Circuit" je to sériový komunikačný protokol, ktorý umožňuje prenášať dáta medzi rôznymi zariadeniami. I2C protokol bol vyvinutý spoločnosťou Philips a bol pôvodne navrhnutý na komunikáciu medzi rôznymi časovačmi, obvody pre spracovanie signálov a inými zariadeniami na jednej doske plošných spojov.

I2C bol pôvodne navrhnutý pre komunikáciu 100 kb/s, v priebehu rokov však dosiahol rýchlosti až do 3,4 Mbit/s.

I2C protokol používa dva vodiče - SDA (Data) a SCL (Clock). SDA vodič slúži na prenos dát a SCL vodič slúži ako hodinový signál na synchronizáciu komunikácie medzi zariadeniami. Aktívna úroveň signálu je 0V. Znamená to, že nie je problém s komunikáciou medzi zariadeniami, ktoré nemajú zhodné napäťové úrovne. Je to presne prípad komunikácie medzi ESP32 a Arduino v tejto práci. Komunikácia prebieha počas tzv. "transakcií", ktoré sa skladajú z posielania adresy zariadenia a dát.

Každé zariadenie na zbernici má svoju unikátnu 7-bitovú adresu (alebo 10-bitovú adresu v rozšírenom móde), ktorá sa používa na identifikáciu zariadenia počas komunikácie. Z týchto 7 bitov sa ale jeden používa na označenie zápisu alebo čítania zariadenia, takže skutočný počet adries pre slave zariadenia je 2^6 , teda 128. [14]

5.1.2 Implementácia I2C

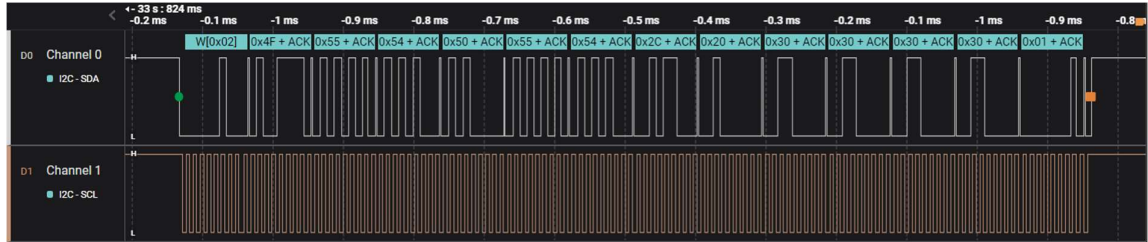
Pre ESP32 a Arduino je dostupná knižnica *Wire.h*, ktorá zabezpečuje komunikáciu cez I2C. Pri ESP32-POE je potrebné nastaviť piny SDA a SCL príkazom:

```
Wire.begin (13, 16); //spustenie i2c komunikacie pre piny SDA->13 a CLK->16
```

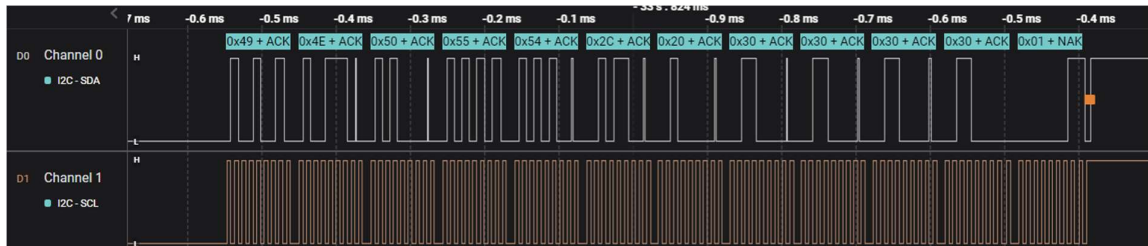
Obrázok 26 Nastavenie pinov I2C pre ESP32-POE

Základná doska nemá konkrétnu adresu, pretože je master a len posiela požiadavky na zápis alebo čítanie na zariadenie slave.

Rozširujúca doska má už konkrétnu adresu, v tomto prípade 0x02. V prípade, že prijme požiadavku od zariadenia master odpovie na ňu. Ukážka komunikácie medzi ESP32-POE a Arduino Nano je na obrázku 20, bola získaná pomocou logického analyzátora.



Obrázok 27 Požiadavka na zápis v zbernici I2C



Obrázok 28 Požiadavka na čítanie v zbernici I2C

Na obrázku 20 vidieť požiadavku na zápis (write), nasleduje adresa zariadenia 0x02 a následne správa: “OUTPUT, 0000”.

Na obrázku 21 vidieť požiadavku na čítanie (read), nasleduje adresa zariadenia 0x02 a následne správa: “INPUT, 0000”.

Správa pozostáva z dvoch častí. Prvá časť rozlišuje vstupy a výstupy, za ktorou nasleduje ich hodnota. Procesory spracujú správy a následne rozširujúca doska nastaví reléové výstupy podľa správy od zariadenia master. Súčasne prijatú správu spracuje master a nastaví modbusový register diskretných vstupov na požadované hodnoty.

5.2 Pripojenie do lokálnej siete

Aby bolo možné pripojiť ESP32-POE do lokálnej ethernetovej siete, je využitá knižnica *ETH.h* a *WiFi.h*. Pre inicializáciu pripojenia je potrebné nastaviť pevnú IP adresu, bránu a podsieť. Pevnú IP adresu musí mať aj Modbus klient. Knižnica *Wifi.h* sa využíva na kontrolu pripojenia na sieť.

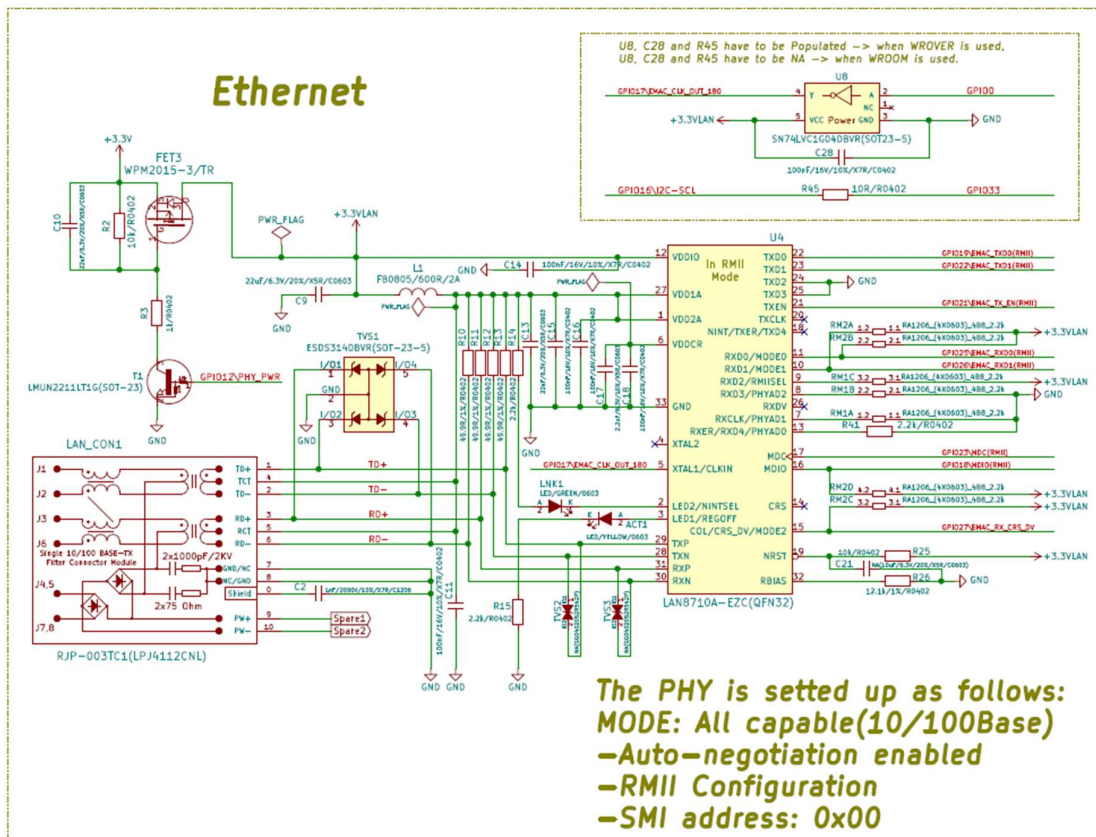
Dôležité je nastavenie pinov pre hodinové impulzy integrovaného obvodu LAN8710A, ktorý sa nachádza na vývojovej doske ESP32-POE a obsluhuje ethernetový port. Následne nastavenie pinu, ktorý spína napájanie k tomuto integrovanému obvodu.

Definícia pinov pre obvod LAN8710:

```
#define ETH_CLK_MODE ETH_CLOCK_GPIO17_OUT //pin CLK pre ethernetovy cip LAN8710A
#define ETH_PHY_POWER 12 //pin pre spinanie napajania pre ethernetovy cip LAN8710A
```

Obrázok 29 Inicializácia pinov pre LAN8710A

Pripojenie do siete overíme na konzole ESP32-POE pri pripojení do siete vypíše parametre pripojenia do siete alebo z príkazového riadku pomocou príkazu **ping**.



Obrázok 30 Schéma zapojenia ethernetového modulu LAN8710A [Príloha D]

5.3 Komunikácia cez Modbus TCP

Na implementáciu Modbusu TCP je viac knižníc, väčšina je určená pre Arduino a na platforme ESP32-POE sa nesprávajú vhodne. Bolo rozhodnuté použiť knižnicu *ModbusIP_ESP8266.h*, ktorá je určená pre procesory rodiny ESP.

Komunikácia vyžaduje stále pripojenie na ethernetovú sieť, nezáleží či prostredníctvom ethernetu alebo wifi.

Na začiatku je potreba inicializovať adresy a registre, ktoré sa budú využívať. Následne môžeme spustiť server príkazom, ktorý sa opakuje s periódou 1ms:

```
mb.task(); //beh server
```

Obrázok 31 Príkaz pre spustenie Modbus TCP serveru

Keď server beží, už je možné zapisovať do registrov a na adresy, ktoré sme inicializovali.

5.3.1 Implementované funkcie Modbusu

Použitá knižnica pre Modbus TCP server podporuje všetky funkcie Modbusu. V práci je využitý len register diskretných vstupov (discrete inputs) a register cievok (coils), čo obmedzuje využitie všetkých funkcií Modbusu. Pre vyhotovený prototyp je možné využiť štyri funkcie. Sú nimi:

- Read coils (funkčný kód 01),
- Read Discrete Inputs (funkčný kód 02),
- Write Single Coil (funkčný kód 05),
- Write Multiple Coils (funkčný kód 15).

5.3.2 Nastavenie výstupov a vstupov

Na obsluhu digitálnych vstupov a výstupov sú využité vytvorené funkcie **ReadInput()** a **SetRelay()**.

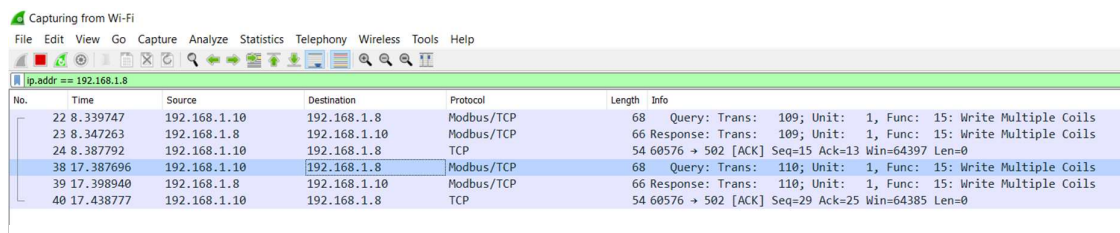
Pri zavolaní funkcie **ReadInput**, ktorá nemá vstupné parametre sa do globálnych premenných vstupov zapíšu aktuálne hodnoty vstupov. Následne sa prepíšu hodnoty v registri diskretných vstupov (discrete inputs) na hodnoty globálnych premenných. Perióda vzorkovania je 1 ms.

Funkcia **SetRelay**, sa volá taktiež s periódou 1 ms, vstupnými parametrami sú hodnoty registra cievok (coils) na požadovaných adresách, v tomto prípade adresy 1 – 4 a podľa nich sa nastaví reléové výstupy.

6. TEST KOMUNIKÁCIE

Pri tvorbe softwaru bolo potrebné priebežné testovanie, predovšetkým funkčnosti samotného protokolu Modbus TCP. Počas hľadania vhodnej knižnice bol využívaný testovací program QModMaster, ktorý simuluje požiadavky na server a dokáže vypísať požiadavky klienta a odpovede serveru. Monitorovanie komunikácie bolo ešte overené programom Wireshark (obrázok 32).

Keď server fungoval správne, bolo potrebné odskúšať pripojenie serveru k PLC. Na otestovanie pripojenia k PLC bolo využité virtuálne PLC v programe Codesys a následne vytvorený jednoduchý program, ktorý dokázal spínať jednotlivé relé a čítať digitálne vstupy. Hneď ako zariadenie dokázalo bezchybne komunikovať s virtuálnym PLC bolo možné pristúpiť k testovaniu na reálnom PLC od Allen-Bradley.



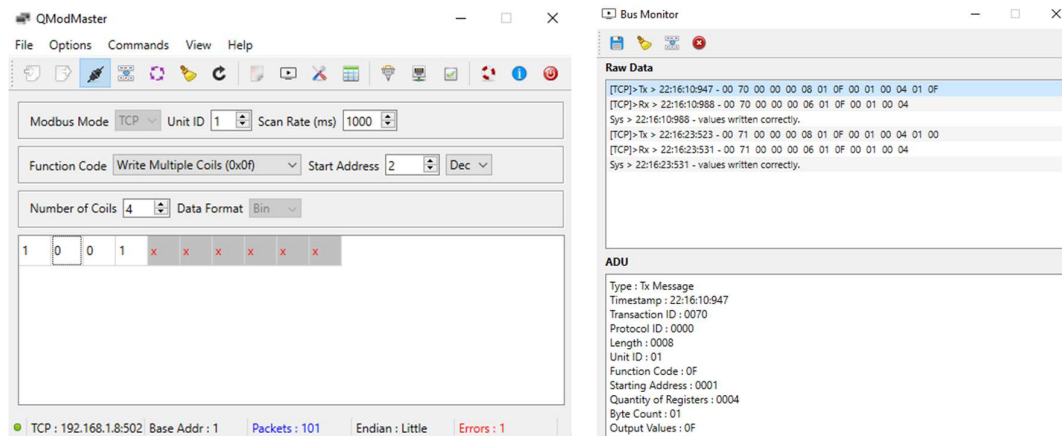
The screenshot shows the Wireshark interface with a packet capture filter set to 'ip.addr == 192.168.1.8'. The packet list pane displays several packets, with the following data extracted from the visible rows:

No.	Time	Source	Destination	Protocol	Length	Info
22	8.339747	192.168.1.10	192.168.1.8	Modbus/TCP	68	Query: Trans: 109; Unit: 1, Func: 15: Write Multiple Coils
23	8.347263	192.168.1.8	192.168.1.10	Modbus/TCP	66	Response: Trans: 109; Unit: 1, Func: 15: Write Multiple Coils
24	8.387792	192.168.1.10	192.168.1.8	TCP	54	60576 → 502 [ACK] Seq=15 Ack=13 Win=64397 Len=0
38	17.387696	192.168.1.10	192.168.1.8	Modbus/TCP	68	Query: Trans: 110; Unit: 1, Func: 15: Write Multiple Coils
39	17.398940	192.168.1.8	192.168.1.10	Modbus/TCP	66	Response: Trans: 110; Unit: 1, Func: 15: Write Multiple Coils
40	17.438777	192.168.1.10	192.168.1.8	TCP	54	60576 → 502 [ACK] Seq=29 Ack=25 Win=64385 Len=0

Obrázok 32 Výpis z programu Wireshark

6.1 Program QModMaster

Je to simulátor pre Modbus klienta, ktorý podporuje Modbus RTU a Modbus TCP. V prvom rade je potrebné nastaviť mód TCP a správnu IP adresu serveru. Následne je možné zadať kód funkcie, pre jej vykonanie, napr.: zápis do viac cievok (write multiple coils), čo má kód 0x0F. Ešte pred odoslaním požiadavky je potrebné nastaviť, od ktorej adresy sa začína prepisovať register a počet adries, ktoré chceme prepísať. Pozor v tomto programe je adresa o jeden väčšia ako reálne na serveri. Napríklad ak je potreba prepísať štyri cievky od adresy 1 na reálnom serveri, musím do programu QModMaster zadať: Start Address = 2 a Number of Coils = 4.



Obrázok 33 Program QModMaster a výpis komunikácie

6.2 Codesys a virtuálne PLC

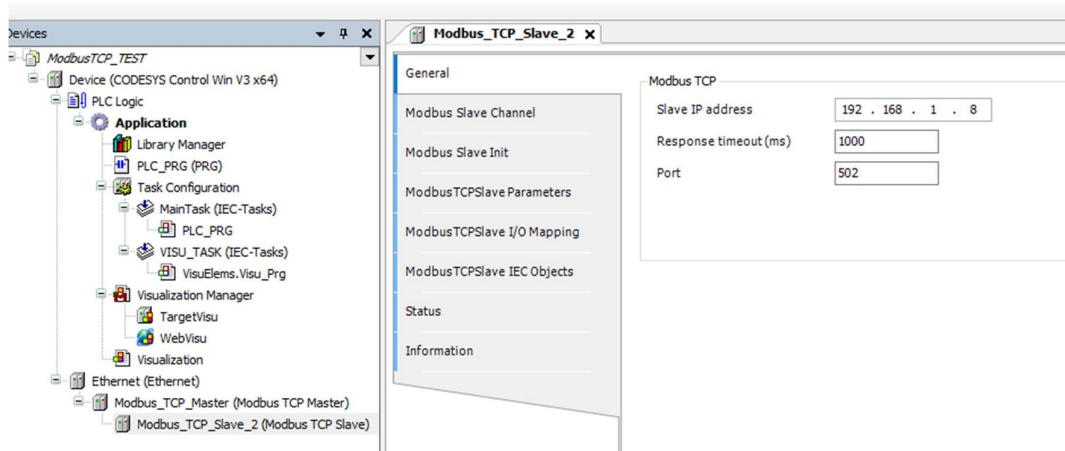
Codesys je softvérové prostredie pre vývoj a programovanie automatizačných systémov, ktoré sa používajú v priemyselnej automatizácii a v riadení procesov. Ide o integrované vývojové prostredie (IDE), ktoré umožňuje programátorom vytvárať a testovať aplikácie pre rôzne typy automatov, ako sú napríklad PLC (programovateľné logické automaty), CNC stroje, HMI (rozhranie medzi človekom a strojom) a ďalšie.

Je možné využívať virtuálne PLC. Simulátor PLC umožňuje simulovať a testovať PLC programy bez fyzického PLC, čo zjednodušuje a zrýchľuje vývoj aplikácií.

Simulátor PLC ponúka množstvo funkcií, ktoré sú typické pre reálne PLC, vrátane digitálnych a analógových vstupov a výstupov, komunikačných protokolov a mnoho ďalších. Okrem toho, simulátor PLC umožňuje programátorom testovať a ladiť aplikácie v simulovanom prostredí, čo prispieva k väčšej kvalite a spoľahlivosti výslednej aplikácie. [15]

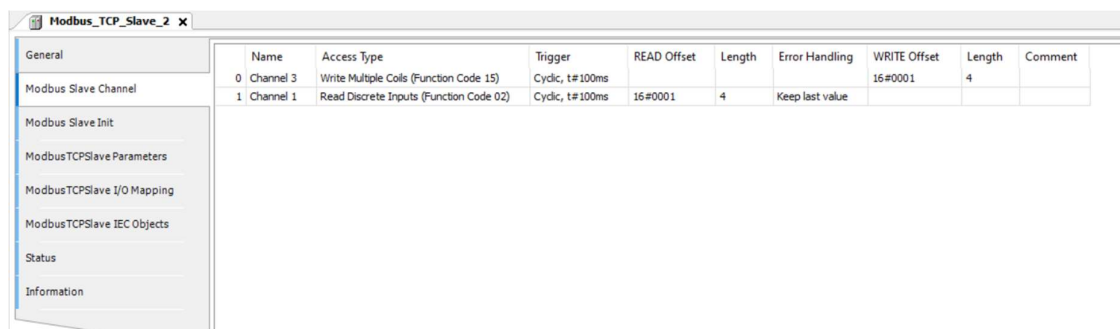
6.2.1 Konfigurácia pre Modbus TCP server

Pripojenie k Modbus serveru si vyžaduje správnu konfiguráciu programu, predovšetkým samotného projektu. Je potrebné nastaviť akým spôsobom bude virtuálne PLC komunikovať, v tomto prípade bude zvolená možnosť *Ethernet*. Pridať do projektu zariadenie *Ethernet* a následne je potrebné do ethernetu pridať Modbus TCP Master. Keď projekt obsahuje ethernet a Modbus TCP Master, tak do Modbus TCP Master je možné pridať požadovaný server teda Modbus TCP Slave, kde je potrebné nastaviť IP adresu a port.



Obrázok 34 Nastavenie komunikácie s Modbus TCP serverom

Po nastavení IP adresy a portu by malo byť možné sa pripojiť na server. Ďalším krokom je nastavenie požiadaviek na server. V tomto prípade je využité zapisovanie do viac cievok (Write Multiple Coils) s kódom 15 a čítanie štyroch diskretných vstupov (Read Discrete Inputs) s kódom 2.



Obrázok 35 Nastavenie požiadaviek na server

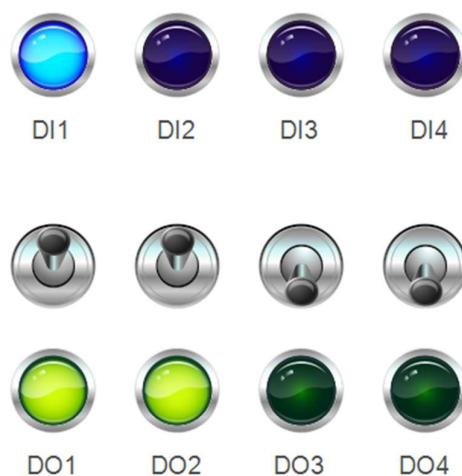
Aby bolo možné využiť požadované hodnoty z registrov Modbusu, je potrebné priradiť premenné na konkrétne adresy. Príklad priradenia premenných na adresy registrov je na obrázku 36.

Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 3	%QB0	ARRAY [0..0] OF BYTE		Write Multiple Coils
		Channel 3[0]	%QB0	BYTE		Write Multiple Coils
Application.PLC_PRG.RELAY_1	↔	Bit0	%QW0-0	BOOL		0x0001
Application.PLC_PRG.RELAY_2	↔	Bit1	%QW0-1	BOOL		0x0002
Application.PLC_PRG.RELAY_3	↔	Bit2	%QW0-2	BOOL		0x0003
Application.PLC_PRG.RELAY_4	↔	Bit3	%QW0-3	BOOL		0x0004
		Channel 1	%IB0	ARRAY [0..0] OF BYTE		Read Discrete Inputs
		Channel 1[0]	%IB0	BYTE		Read Discrete Inputs
Application.PLC_PRG.INPUT_1	↔	Bit0	%IX0-0	BOOL		0x0001
Application.PLC_PRG.INPUT_2	↔	Bit1	%IX0-1	BOOL		0x0002
Application.PLC_PRG.INPUT_3	↔	Bit2	%IX0-2	BOOL		0x0003
Application.PLC_PRG.INPUT_4	↔	Bit3	%IX0-3	BOOL		0x0004

Obrázok 36 Priradenie premenných na adresy registrov

6.2.2 Vizualizácia testovacieho programu

Pre jednoduchšie ovládanie a kontrolu funkcionality bola vytvorená jednoduchá vizualizácia, ktorá ukazuje stav vstupov a výstupov servera a umožňuje spínanie výstupov.



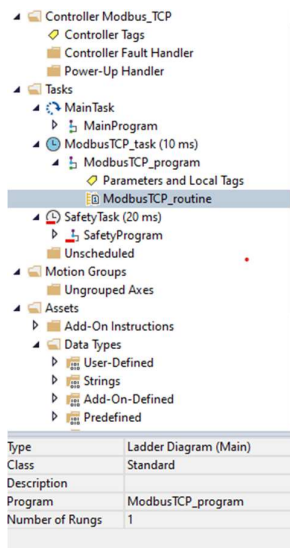
Obrázok 37 Vizualizácia testovacieho programu v Codesyse [Príloha F]

6.3 Testovacia aplikácia na reálnom PLC

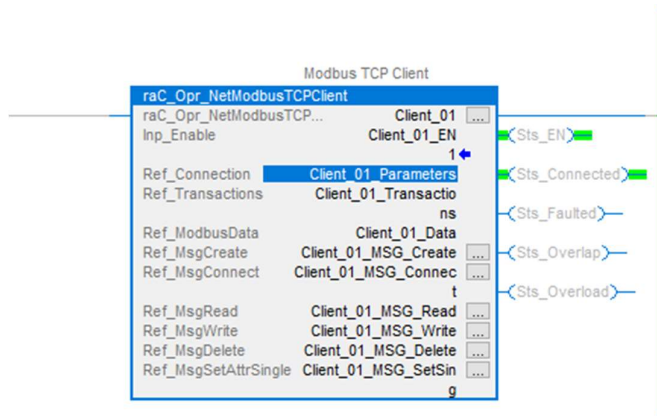
Testovacia aplikácia na reálnom PLC rady Logix od Allen-Bradley, má dve hlavné oblasti test samotnej komunikácie s vizualizáciou a diagnostika prípadných chýb pri komunikácii.

Allen-Bradley má pre komunikáciu pomocou Modbusu TCP vytvorenú Add-on inštrukciu, ktorá sa líši pre klienta a server. Add-on inštrukcia [Príloha F] zahŕňa podrobnú dokumentáciu na správne nastavenie a popis jednotlivých premenných v štruktúre.

Po nahraní Add-on inštrukcie do projektu môžeme pomocou štruktúr v globálnych premenných nastaviť IP adresu a požiadavky na server. Každá požiadavka na server obsahuje diagnostiku chýb. Add-on inštrukcia pre klienta a server musí byť v periodickom tasku ako je vidieť na obrázku 39.



Obrázok 38 Program pre Modbus TCP v periodickom tasku



Obrázok 39 Add-on inštrukcia pre klienta

6.3.1 Nastavenie požiadavky na server

Klient umožňuje vykonávať maximálne štyri požiadavky (transakcie) na server. Na obrázku 28 je vidieť nastavenie požiadaviek na zápis do štyroch cievok (Write Multiple Coils) od adresy 1 a štruktúru diagnostiky tejto požiadavky.

Name	Value	Force Mask	Style	Data Type	Class	Description
Client_01_Parameters		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[0]		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[0].BeginAddress		1	Decimal	DINT	Standard	Beginning address in...
Client_01_Transactions[0].Count		4	Decimal	INT	Standard	The number of items...
Client_01_Transactions[0].Diagnostic		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[0].Diagnostic.IntervalTimer		(-)	(-)	TIMER	Standard	IntervalTimer, used t...
Client_01_Transactions[0].Diagnostic.NoReplyCounter		(-)	(-)	COUNTER	Standard	No Reply Counter
Client_01_Transactions[0].Diagnostic.Overlap		0	Decimal	BOOL	Standard	Transaction is not co...
Client_01_Transactions[0].Diagnostic.Overlap		0	Decimal	BOOL	Standard	Transaction is not co...
Client_01_Transactions[0].Diagnostic.PendingTimer		(-)	(-)	TIMER	Standard	Timing pending requ...
Client_01_Transactions[0].Diagnostic.ReqBult		0	Decimal	BOOL	Standard	Transaction request s...
Client_01_Transactions[0].Diagnostic.Request	'5F850050050050850050F500501500504501501'	(-)	(-)	STRO500	Standard	Actual buffer used to...
Client_01_Transactions[0].Diagnostic.TransID		8443	Decimal	INT	Standard	Serial Number of the...
Client_01_Transactions[0].Diagnostic.TransLastError		0	Decimal	INT	Standard	Last Error on transac...
Client_01_Transactions[0].Enabled		1	Decimal	BOOL	Standard	Enables (1) or disabl...
Client_01_Transactions[0].LocalAddress		0	Decimal	INT	Standard	Beginning address in...
Client_01_Transactions[0].PollInterval		1000	Decimal	DINT	Standard	Time between trigge...
Client_01_Transactions[0].StationID		0	Decimal	SINT	Standard	Station ID (Unit Iden...
Client_01_Transactions[0].TransComplete		1	Decimal	BOOL	Standard	Transaction Complet...
Client_01_Transactions[0].TransStat		0	Decimal	INT	Standard	Status result of the tr...
Client_01_Transactions[0].TransType		15	Decimal	INT	Standard	Transaction code for...
Client_01_Transactions[1]		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[2]		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[3]		(-)	(-)	raC_UDT_ModbusCie...	Standard	
Client_01_Transactions[4]		(-)	(-)	raC_UDT_ModbusCie...	Standard	

Obrázok 40 Nastavenie požiadaviek - Allen Bradley

6.3.2 Diagnostika chýb pri komunikácii

Diagnostika je rozdelená na dve časti. Prvá časť kontroluje chyby pri jednotlivých požiadavkách (Obrázok 28) a druhá spojuje chyby jednotlivých požiadaviek a kontroluje pripojenie do siete (Obrázok 29). Premenná *LastError* v štruktúre *Client*, určuje aktuálny stav komunikácie a z tejto premennej je možné vytvoriť históriu chýb.

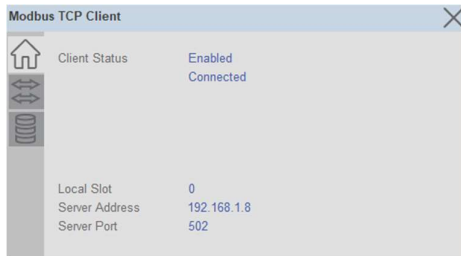
Overenie diagnostiky spočívalo v zadávaní chybných dát do požiadavky na sever, napríklad zlý kód funkcie alebo prístup k neplatnej adrese registra. Následne to bolo vyhodnotené ako chyba transakcie a v premennej *TransLastError* (obrázok 28) sa objavil kód pre konkrétnu chybu.

Name	Value	Force Mask	Style	Data Type	Class	Description
Client_01		(-)	(-)	raC_Oprt_NetModbus...	Standard	Modbus TCP Client
Client_01.EnableIn		1	Decimal	BOOL	Standard	Modbus TCP Client E...
Client_01.EnableOut		1	Decimal	BOOL	Standard	Modbus TCP Client E...
Client_01.Inp_Enable		1	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Sts_Connected		1	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Sts_EN		1	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Sts_Faulted		0	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Sts_LastError		0	Decimal	DINT	Standard	Modbus TCP Client T...
Client_01.Sts_Overlap		0	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Sts_Overload		0	Decimal	BOOL	Standard	Modbus TCP Client
Client_01.Params		(-)	(-)	raC_HWT_ModbusCie...	Standard	

Obrázok 41 Globálna diagnostika pre Modbus TCP klienta

6.3.3 Vizualizácia komunikácie

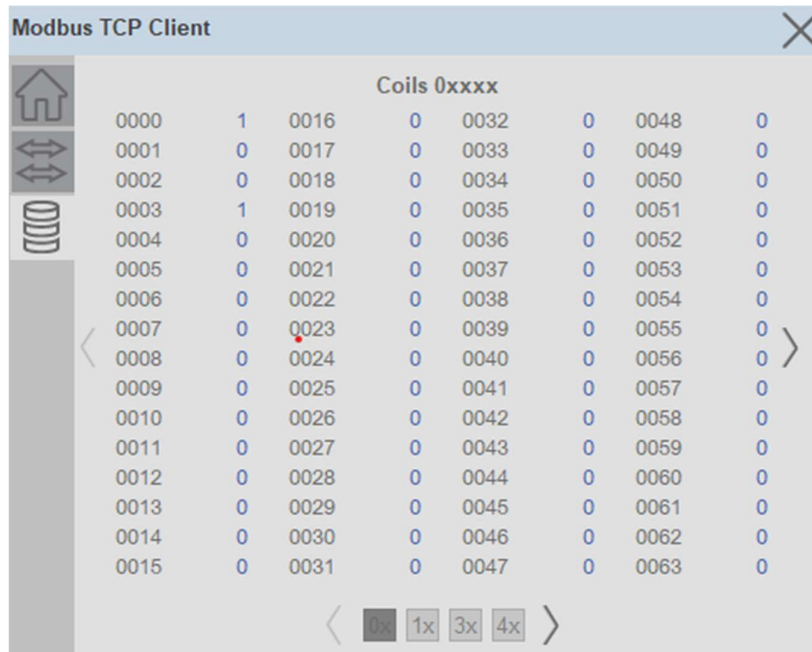
Obsahuje tri okná, prvé zobrazuje aktuálny stav klienta, jeho IP adresu a port (obrázok 43). Druhé zobrazuje jednotlivé požiadavky, ich aktuálny stav a parametre (obrázok 44). Posledné okno udáva hodnoty v jednotlivých registroch (obrázok 45). [Príloha F]



Obrázok 43 Client status



Obrázok 42 Stav a parametre požiadaviek



Obrázok 44 Hodnoty prvkov v registroch

7. ZÁVER

Bakalárska práca sa zaoberala implementáciou priemyselného komunikačného protokolu Modbus TCP na platformu ESP32-POE a jeho komunikáciu s PLC rady Logix od Allen-Bradley. Hlavné ciele boli vytvorenie prototypu hardvéru, implementácia komunikačného protokolu do zariadenia a overenie funkčnosti samotného zariadenia a jeho komunikácie cez Modbus TCP.

Výsledkom práce je úspešná implementácia protokolu Modbus TCP na platforme ESP32 POE. Využitie existujúcej knižnice umožnilo efektívne a spoľahlivo realizovať komunikáciu s PLC rady Logix. Pred samotným testovaním na reálnom hardvéri bola overená správnosť implementácie pomocou simulácie Modbus TCP klienta a následne pomocou virtuálneho PLC. Táto simulácia umožnila priblíženie k reálnemu PLC a zabezpečiť, že systém funguje správne a spoľahlivo.

Okrem samotnej funkčnosti zariadenia bola venovaná pozornosť aj diagnostike komunikácie. Na reálnom PLC rady Logix bola overená spoľahlivosť a efektívnosť komunikácie a vyhodnotil správne fungovanie diagnostických nástrojov.

Úspešne bolo ukázané, že je možné efektívne a spoľahlivo komunikovať s PLC rady Logix pomocou tohto protokolu. Implementácia umožňuje ovládanie digitálnych vstupov a výstupov zariadenia prostredníctvom Modbusu TCP, čo otvára možnosti pre rôzne priemyselné aplikácie.

V budúcnosti by bolo zaujímavé rozšíriť túto prácu o ďalšie funkcie a možnosti protokolu Modbus TCP, napríklad podporu analógových vstupov a výstupov. Taktiež by sa dalo zamerať na optimalizáciu komunikačného procesu, prepracovať prepínanie adresy a vytvoriť nové rozširujúce dosky s rôznymi funkciami.

LITERATÚRA

- [1] *Portál Siemens.com* [online]. 2023. Dostupné z URL: <https://new.siemens.com/sk/sk/produkty/priemyselna-automatizacia/priemyselna-komunikacia.html>
- [2] *Komunikácia v priemyselnej automatizácii* [online]. 2023. Dostupné z URL: https://elearning.mechatronika.cool/wp-content/uploads/2017/05/kapitola_1.pdf
- [3] *Ethernet vs. sériové protokoly v priemyslu* [online]. 4. 10. 2018 [cit. 2021-12-25]. Dostupné z URL: <https://automatizace.hw.cz/ethernet-vs-seriove-protokoly.html> MICHALEC, Libor. Praha.
- [4] *Komunikace s PLC automaty Allen Bradley od firmy Rockwell Automation* [online]. Dostupné z URL: <https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/PLC/ABradley.htm>, promotic.eu
- [5] *Obrázok ControlLogix 5580* [online]. [cit. 2023-05-17]. Dostupné z: <https://adjditec.com/en/productos/1756-l83e-allen-bradley-controllogix-logix-5580-processor-10mb-usb-ethernet>
- [6] *Obrázok Komunikácia Allen Bradley* [online]. [cit. 2023-05-17]. Dostupné z: <https://realpars.com/allen-bradley-plc/>
- [7] *Modbus is an open communications protocol that commonly used on serial networks, Modbus RTU, and Ethernet networks, Modbus TCP* [online]. [cit. 2023-05-17]. Dostupné z: <https://twcontrols.com/allen-bradley-modbus-communications>
- [8] *Protokol Modbus RTU v kostce s popisy a příklady* [online]. [cit. 2023-05-17]. Dostupné z: <https://ipc2u.cz/blogs/news/protokol-modbus-rtu-v-kostce-s-popisy-a-priklady>
- [9] *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b* [online]. [cit. 2023-05-17]. Dostupné z: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

- [10] *Obrázok ESP32 s anténou WIFI* [online]. [cit. 2023-05-17]. Dostupné z: <https://en.wikipedia.org/wiki/ESP32>
- [11] *Obrázok ESP32-POE* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware>
- [12] *ESP32-POE pinout* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/resources/ESP32-POE-GPIO.png>
- [13] *ESP32-POE* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware>
- [14] *Čo je protokol I2C?* [online]. [cit. 2023-05-17]. Dostupné z: <https://sk.go-travels.com/48271-overview-of-i2c-819001-1009150>
- [15] *WHY CODESYS?* [online]. [cit. 2023-05-17]. Dostupné z: <https://www.codesys.com/the-system/why-codesys.html>

ZOZNAM SYMBOLOV A SKRATIEK

Skratky:

FEKT	Fakulta elektrotechniky a komunikačných technológií
VUT	Vysoké učení technické v Brně
TCP	Transmission control protocol
IP	Internetový protokol
RT	Real time
PLC	Programmable Logic Controller
RTU	Telemetry unit
ID	Identifikátor
ADU	Aplikačná dátová jednotka
PDU	Protokolová dátová jednotka
MBAP	hlavička Modbus rámca
I2C	Inter-Integrated circuit
WIFI	Wireless fidelity
RAM	Random access memory
ROM	Read only memory
SPI	Special peripheral interface
UART	Universal asynchronous receiver-transmitter
POE	Power over ethernet
USB	Universal serial bus
BCD	Binary code decimal
LED	Light-emitting diode
NPN	usporiadanie polovodičov v tranzistore
kb/s	Kilobajt za sekundu
Mb/s	Megabajt za sekundu
KiB	Kibibajt
DPS	Doska plošných spojov

Symboly:

U	napätie	(V)
I	prúd	(A)
Ω	odpor	(Ω)

ZOZNAM PRÍLOH

PRÍLOHA A	DOKUMENT OD FIRMY ALLEN BRADLEY	51
PRÍLOHA B	SCHÉMA ZAPOJENIA PROTOTYPU	52
PRÍLOHA C	DPS PROTOTYPU	53
PRÍLOHA D	SCHÉMA ZAPOJENIA ESP32-POE.....	54
PRÍLOHA E	SOFTVÉR.....	55
PRÍLOHA F	TESTOVACIE PROGRAMY.....	56
PRÍLOHA G	PROTOTYP ZARIADENIA	57

Príloha A Dokument od firmy Allen Bradley

Manuál pre PLC rady ControlLogix, ako elektronická príloha na CD.

Príloha B Schéma zapojenia prototypu

Ako elektronická príloha, návrh v programe Eagle.

Príloha C DPS prototypu

Ako elektronická príloha, návrh v programe Eagle.

Príloha D Schéma zapojenia ESP32-POE

Ako elektronická príloha na CD.

Príloha E Softvér

Ako elektronická príloha na CD.

Príloha F Testovacie programy

Ako elektronická príloha na CD.

Príloha G Prototyp zariadenia

Fyzické zariadenia.