



Diplomová práce

Integrace a řízení UUGreenPower DC zdroje pomocí PLC Siemens s využitím CAN komunikace

Studijní program:

N2612 – Elektrotechnika a informatika

Studijní obor:

3906T001 – Mechatronika

Autor práce:

Bc. Martin Lupoměský

Vedoucí práce:

Ing. Lukáš Hubka, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



Master Thesis

Integration and Control of UUGreenPower DC Power Supply using Siemens PLC with CAN Communication

Study programme:

N2612 – Electrical Engineering and Informatics

Study branch:

3906T001 – Mechatronics

Author:

Bc. Martin Lupoměský

Supervisor:

Ing. Lukáš Hubka, Ph.D.

Institute of Mechatronics and Computer Engineering

Liberec 2023



Zadání diplomové práce

Integrace a řízení UUGreenPower DC zdroje pomocí PLC Siemens s využitím CAN komunikace

<i>Jméno a příjmení:</i>	Bc. Martin Lupoměský
<i>Osobní číslo:</i>	M21000184
<i>Studijní program:</i>	N2612 – Elektrotechnika a informatika
<i>Studijní obor (specializace):</i>	3906T001 – Mechatronika
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2023/2024

Zásady pro vypracování:

1. Seznamte se s DC zdrojem UUGreenPower a jeho komunikací přes CAN sběrnici.
2. Vyberte a charakterizujte rozšiřující moduly pro PLC Siemens S7-1200/1500 či ET200SP, které umožní CAN komunikaci se zdrojem. Prozkoumejte standardní možnosti komunikace těchto modulů s CAN zařízením.
3. Vytvořte vlastní knihovní funkce pro řízení zdroje a/i skupiny zdrojů. Implementujte nejen ovládání napětí a proudu, ale i čtení aktuálních parametrů.
4. Navrhněte způsob auto konfigurace pro více zdrojů na sběrnici.
5. Vytvořte obrazovky pro HMI umožňující nastavení parametrů, dohled, diagnostiku.



Rozsah grafických prací: Dle potřeby dokumentace
Rozsah pracovní zprávy: 40–50 stran
Forma zpracování práce: Tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

- [1] Rectifier module and monitoring unit CAN communication reference guide. UUGreenPower Electrical Co., Ltd., version 06, 2018.
- [2] Configuration of CM CAN for CAN transparent operating mode. Siemens, 2020.
- [3] Simatic. ET 200SP Communication Module CAN. Siemens, 2021.

Vedoucí práce: Ing. Lukáš Hubka, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2023
Předpokládaný termín odevzdání: 14. května 2024

Z.S.

prof. Ing. Zdeněk Plíva, Ph.D.

děkan

doc. Ing. Josef Černohorský, Ph.D.

vedoucí ústavu

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci 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.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

14. května 2024

Bc. Martin Lupoměský

Integrace a řízení UUGreenPower DC zdroje pomocí PLC Siemens s využitím CAN komunikace

Abstrakt

Diplomová práce se zabývá problematikou komunikace mezi PLC a napájecími zdroji v dobíjecích stanicích pro elektromobily s využitím specifického komunikačního modulu CM CAN od společnosti Siemens, práce analyzuje a navrhuje efektivní způsoby komunikace, které by zlepšily spolehlivost a efektivitu systému. Úvodní část práce se zaměřuje na problematiku komunikace pomocí sběrnice CAN a představuje alternativní řešení pro efektivní propojení mezi PLC a napájecími zdroji. Součástí je také detailní popis komunikačního modulu CM CAN od Siemensu a jeho konfigurační možnosti. Další část práce se zabývá softwarovou stránkou problému, kde je detailně popsán každý z funkčních bloků a funkcí navrženého programu. Je zde představena a navržena komplexní knihovna funkcí a funkčních bloků pro PLC, umožňující efektivní komunikaci s napájecími zdroji přes komunikační modul CM CAN. Následující část práce zahrnuje návrh autokonfigurace při využití více zdrojů, která usnadňuje proces nasazení a údržby energetických zdrojů. Poslední část práce se věnuje vizualizaci prostřednictvím HMI obrazovek, které slouží k diagnostice a ovládání napájecích zdrojů. Jsou zde popsány jednotlivé obrazovky, propojení proměnných s programem a postupy pro ovládání funkcí na obrazovkách. Cílem této práce je zjednodušit proces ovládání a monitorování napájecích zdrojů.

Klíčová slova

CAN, CANopen, transparentní, modul, CAN zpráva, proxy, funkční blok, komunikace

Integration and Control of UUGreenPower DC Power Supply using Siemens PLC with CAN

Abstract

This thesis deals with the issue of communication between PLC and power supplies in charging stations for electric vehicles using a specific communication module CM CAN from Siemens, the thesis analyzes and proposes effective ways of communication that would improve the reliability and efficiency of the system. The introductory part of the thesis focuses on the problematics of communication using the CAN bus and presents alternative solutions for effective interconnection between PLC and power supplies. It also includes a detailed description of the Siemens CM CAN communication module and its configuration options. The next part of the paper deals with the software side of the problem, where each of the function blocks and functions of the proposed program is described in detail. A comprehensive library of functions and function blocks for PLCs is presented and designed to enable efficient communication with power supplies via the CM CAN communication module. The following part of the paper covers the autoconfiguration when using multiple power supplies, which facilitates the process of power supply deployment and maintenance. The last part of the thesis is devoted to visualization through HMI screens, which are used to diagnose and control the on-power supplies. The different screens, the interfacing of variables with the program and the procedures for controlling the functions on the screens are described. The aim of this work is to simplify the process of controlling and monitoring power supplies.

Keywords

CAN, CANopen, transparent, module, CAN message, proxy, function block, communication

Poděkování

Tímto bych chtěl poděkovat především mému vedoucímu práce Ing. Lukáši Hubkovi, Ph.D. za trpělivost a užitečné poznatky k řešení diplomové práce a společnosti Siemens za poskytnutí zadání diplomové práce.

Obsah

Úvod.....	14
1 Rešerše.....	16
1.1 Moduly umožňující komunikaci CAN	16
1.1.1 Komunikační modul CM CANopen.....	16
1.1.2 Modul ET 200S 1SI CANopen	17
1.1.3 Komunikační modul PN/CAN Link.....	17
1.1.4 Komunikační modul ET 200SP CM CAN	17
1.2 Napájecí modul UR50060-E.....	18
1.3 Sběrnice CAN.....	20
1.4 CAN komunikace se zdroji.....	21
1.4.1 ID pole	21
1.4.2 Datové pole	22
1.5 Komunikační modul CAN.....	23
1.5.1 CANopen Manager.....	24
1.5.2 CANopen Slave.....	24
1.5.3 CAN Transparent.....	25
1.5.4 Porovnání.....	25
1.6 CAN Transparent.....	25
1.6.1 Konfigurovatelné zprávy CAN.....	26
1.6.2 Programovatelné zprávy CAN (proxy).....	27
1.6.3 Porovnání.....	27
1.6.4 Filtry pro příjem programovatelných CAN zpráv	28
1.6.5 Datový blok a proměnné modulu CM CAN	29
2 Softwarové řešení zařízení.....	30
2.1 Hardwarová konfigurace v TIA Portal.....	30
2.2 Řešení logické struktury programu a tagů	31

2.3	Status CM CAN.....	33
2.4	Konfigurace CM CAN	34
2.5	Odeslání transparentních zpráv	36
2.5.1	FB pro zpracování odesílaných zpráv.....	36
2.5.2	FB pro odeslání transparentní CAN zprávy	38
2.6	Nastavení filtrů pro příjem transparentních CAN zpráv	41
2.7	Přijetí transparentních CAN zpráv	43
2.7.1	FB pro přijetí transparentních CAN zpráv	44
2.7.2	FB na zpracování přijatých CAN zpráv	46
2.8	TIA knihovna pro řízení a monitorování napájecích zdrojů	48
3	Návrh autokonfigurace s využitím více zdrojů.....	49
3.1	Konfigurace komunikačního modulu	49
3.2	Logika programu	49
3.3	HMI a chybová hlášení.....	50
3.4	Výhody autokonfigurace	50
4	Vizualizace.....	51
4.1	Hlavní obrazovka.....	51
4.2	Obrazovka pro nastavení parametrů.....	52
4.3	Obrazovky diagnostiky.....	53
	Závěr.....	55
	Použitá literatura.....	57
	Přílohy	59

Seznam obrázků

Obrázek 1 - Nabíjecí modul UR50060-E [5].....	18
Obrázek 2 - Princip funkce modulu UR50060-E [5].....	19
Obrázek 3 - Jednotlivé piny konektoru [5].....	19
Obrázek 4 – Adresovací deska napájecích modulů [5].....	20
Obrázek 5 – Vzor CAN zprávy pro nastavení referenčního napětí	23
Obrázek 6 – Komunikační modul ET 200SP CM CAN [4].....	24
Obrázek 7 – Nastavení filtru přijatých zpráv uvnitř HSP	29
Obrázek 8 – Hardwarová konfigurace v TIA portálu.....	31
Obrázek 9 – Diagram toku dat od rozběhnutí PLC	32
Obrázek 10 – Funkce pro zjištění statusu CM CAN v transparentním režimu.....	33
Obrázek 11 – Přehled nakonfigurovaných proxy a jejich příslušné proměnné	34
Obrázek 12 – FB pro konfiguraci modulu ET 200SP CM CAN.....	35
Obrázek 13 – Funkční blok pro zpracování odesílaných práv	37
Obrázek 14 – FB pro odesílání transparentní datové zprávy	39
Obrázek 15 – Zapsaná data filtrů v DB „Transparent_messages“	42
Obrázek 16 – Funkční blok pro nastavení filtrů přijímaných CAN zpráv	43
Obrázek 17 – Funkční blok pro příjem transparentních CAN zpráv	46
Obrázek 18 – FB pro zpracování přijatých transparentních CAN zpráv.....	47
Obrázek 19 – Uživatelská knihovna v prostředí TIA Portal	48
Obrázek 20 – Hlavní obrazovka operačního panelu.....	51
Obrázek 21 – Obrazovka pro nastavení parametrů	52
Obrázek 22 – Obrazovka pro diagnostiku funkčních bloků.....	53
Obrázek 23 – Doplnující obrazovka pro diagnostiku funkčních bloků.....	54

Seznam tabulek

Tabulka 1 – Význam jednotlivých bitů v ID poli.....	21
Tabulka 2 – Význam jednotlivých bajtů v datovém poli.....	22
Tabulka 3 – Adresace jednotlivých modulů	28

Seznam použitých zkratek

PLC	Programmable logic controller
PN	Profinet
CPU	Central processing unit
ST	Standart
HMI	Human-machine interface
IM	Interface modul
BA	Bus adapter
CM	Communication modul
CAN	Controller area network
DQ	Digital output
DC	Direct current
DI	Digital input
ECC	Error correction code
PL	Programming logic
HSP	Hardware support packages
RTR	Remote transmit frame
SDO	Servis data objects
PDO	Process data objects
NMT	Network management
CiA	CAN in Automation
PFC	Power factor correction
EMI	Electromagnetic interference
DSP	Digital signal processor

Úvod

Diplomová práce vznikla ve spolupráci se společností Siemens, která patří mezi největší firmy v oblasti průmyslové automatizace. V dnešním technologickém prostředí, kde automatizované systémy a průmyslové provozy nabývají na stále větším významu, hraje klíčovou roli spolehlivá a efektivní komunikace mezi jednotlivými prvky. Jedním z klíčových prvků zmiňované komunikace je sběrnice CAN, která se především využívá v automobilovém průmyslu. Práce se zaměřuje na komunikaci mezi PLC a napájecími zdroji, které jsou součástí dobíjecí stanice pro elektromobily. K tomuto účelu se používá specifický komunikační modul od firmy Siemens, který je připojen k distribuované periferní jednotce.

Jedním z cílů práce je čtenáře seznámit s problematikou CAN komunikace a určit možné alternativy pro řešení této problematiky. Dále pak vytvořit komplexní knihovnu funkcí a funkčních bloků, která umožní efektivní komunikaci mezi PLC a napájecími zdroji pomocí komunikačního modulu CM CAN. Součástí práce je také návrh a implementace HMI obrazovek pro diagnostiku a ovládání těchto zdrojů, které mají za úkol zjednodušit proces ovládání a monitorování zdrojů, což povede k efektivnějšímu využití a správě energetických prostředků, jako je zvyšování energetické účinnosti.

Úvodní částí práce je představení různých možností komunikace pomocí modulů CAN, které společnost Siemens nabízí pro své PLC řady. Jsou zde představeny moduly z různých sérií PLC, které podporují komunikaci po CAN sběrnici, a je rozebráno, jak se hodí pro dané úkoly. Dále se popisují napájecí zdroje včetně principu funkce, zapojení a adresování modulů, spolu se stručným popisem sběrnice CAN. Součástí je i vysvětlení CAN komunikace mezi zdroji a PLC, zahrnující jednotlivé části CAN zprávy jako datové a identifikační pole, doplněné ukázkou takové zprávy. Hlavním zaměřením kapitoly je detailní popis komunikačního modulu CM CAN, včetně jeho konfiguračních možností, provozních režimů a porovnání metod pro správnou funkci dle specifikací.

Navazující část práce se zaměřuje na softwarovou stránku úlohy. Zde jsou popsány jednotlivé funkce, funkční bloky a hlavní program, včetně jejich struktury a organizace. Rozbor jednotlivých částí programů funkčních bloků a funkcí poskytne čtenáři srozumitelný přehled. Dále je podrobně vysvětlena tvorba knihovny pro komunikaci PLC se zdroji přes komunikační modul CM CAN. Tato knihovna je strukturována podle příjmu a odesílání CAN zpráv, filtraci zpráv podle typu a zpracování různých datových typů, jako jsou

napětí, proud nebo stav zdroje. Součástí práce je také zhotovení návrhu autokonfigurace při použití více zdrojů.

Závěrečná část práce se zabývá vizualizací, která je zhotovena za pomoci HMI obrazovek. Je zde uveden popis jednotlivých obrazovek, propojení proměnných s programem a postupů pro ovládání jednotlivých funkcí na obrazovkách.

1 Rešerše

Pro zajištění efektivní komunikace mezi PLC a zdroji bude modul CM CAN nakonfigurován tak, aby umožňoval přenos dat v reálném čase. Díky tomu bude PLC moci včas reagovat na změny stavu zdrojů a regulovat jejich chod. K monitorování a ovládání zdrojů se bude používat operační panel HMI, který umožní uživateli přehledně sledovat stav zdrojů a zadávat požadované parametry.

1.1 Moduly umožňující komunikaci CAN

Pro komunikaci na zařízení je využíván komunikační modul ET200SP CM CAN spolu s PLC řady SIMATIC S7-1511, ale pro PLC řady SIMATIC S7-1200 a 1500 jsou k dispozici různé moduly určené pro komunikaci po sběrnici CAN. Moduly jsou navrženy pro integrování do distribuovaného I/O systému ET 200S nebo ET 200SP, stejně jako pro připojení k rozhraní komunikačního modulu. Každý z těchto modulů podporuje aplikace CAN sběrnice. Moduly se odlišují v jejich schopnosti využívat standard CANOpen, podporovat různé protokoly CAN, jako je CAN 2.0A a CAN 2.0B nebo se liší v nabízených provozních režimech. Pro úspěšné provedení zadané úlohy je nezbytné správně vybrat modul, který bude splňovat potřebné úkony pro řešení práce.

1.1.1 Komunikační modul CM CANOpen

CM CANOpen umožňuje propojení libovolného zařízení CANOpen nebo CAN 2.0A s PLC SIMATIC S7-1200. Jedná se o modul typu CANOpen Manager/Slave, který se připojuje k rozhraní komunikačního modulu S7-1200 a slouží jako komunikační most mezi zařízeními CANOpen/CAN a PLC. Modul umožňuje PLC S7-1200 fungovat jako master v síti CANOpen a komunikovat až se šestnácti zařízeními CANOpen slave. K jednomu řídicímu systému S7-1200 lze připojit až tři moduly CM CANOpen. Mezi výhody patří podpora transparentního režimu CAN 2.0A pro implementaci vlastních protokolů CAN, spolu s optimalizovaným a intuitivním nástrojem pro konfiguraci CANOpen [1]. Tento modul není pro danou úlohu vhodný, protože podporovaný provozní režim Transparent umožňuje pouze práci s CAN zprávami o 11 bitech v identifikačním poli, což není dostačující pro požadavky úlohy.

1.1.2 Modul ET 200S 1SI CANopen

Modul poskytuje integrovat zařízení s protokolem CANopen do distribuovaného I/O systému ET200S, umožňuje tak bezproblémovou komunikaci mezi PLC S7-1200 nebo S7-1500 a zařízeními CANopen v síti. Tento modul podporuje protokol CANopen verze 2.1.1 s jedním kanálem CAN, maximální rychlostí přenosu dat až 1 Mbit/s a zajišťuje diagnostiku CANopen. Snadno se integruje do TIA Portalu. Jeho výhody zahrnují rozšíření možností komunikace PLC S7-1200 a S7-1500 o zařízení CANopen, bezproblémovou integraci do stávajících systémů ET200S, snadné použití, konfiguraci a vysokou spolehlivost a robustnost [2]. Nevýhodou modulu je, že nepodporuje provozní režim Transparent a pracuje pouze se standardním protokolem CANopen, který neumožňuje odesílat a přijímat volně konfigurovatelné zprávy CAN. Na základě tohoto není vhodný pro danou úlohu.

1.1.3 Komunikační modul PN/CAN Link

Komunikační modul propojuje řídicí systémy S7 s CAN sítěmi, umožňuje komunikaci se zařízeními v síti CAN a podporuje protokoly CAN 2.0A/B. Disponuje 1 kanálem CAN s rychlostí přenosu dat až 1 Mbit/s a poskytuje diagnostiku sítě. Kompatibilní je s PLC řady S7-1200 i S7-1500. Podporuje provozní režimy CANopen manager, CANopen slave a CAN Transparent pro flexibilní nastavení pomocí konfigurovatelných CAN zpráv nebo programovatelných CAN zpráv. Modul se také snadno integruje do TIA Portalu a jeho konfigurace pomocí přiřazeného HSP je snadná [3]. Tento modul by mohl být další variantou pro splnění požadavků dané úlohy. Konfigurace modulu je téměř obdobná používanému modulu CM CAN a vlastnosti jednotlivých provozních režimů jsou identické.

1.1.4 Komunikační modul ET 200SP CM CAN

Jak už bylo řečeno, tak tento modul se využívá na zařízení. Jedná se o komunikační modul CAN pro periferní stanici ET 200SP. Umožňuje implementaci protokolu CAN a CANopen v jednom modulu. Nabízí tři provozní režimy, a to CANopen Manager, CANopen Slave a CAN transparent. Podporuje standardní i rozšířený formát CAN a přenosové rychlosti 10 až 1000 kbit/s. Komunikační modul je kompatibilní s PLC řady S7-1200

i S7-1500. Komunikace s modulem probíhá prostřednictvím bajtových dat, která umožňují čtení a zápis dat do registrů modulu pro konfiguraci a nastavení parametrů sítě CAN. [4]

1.2 Napájecí modul UR50060-E

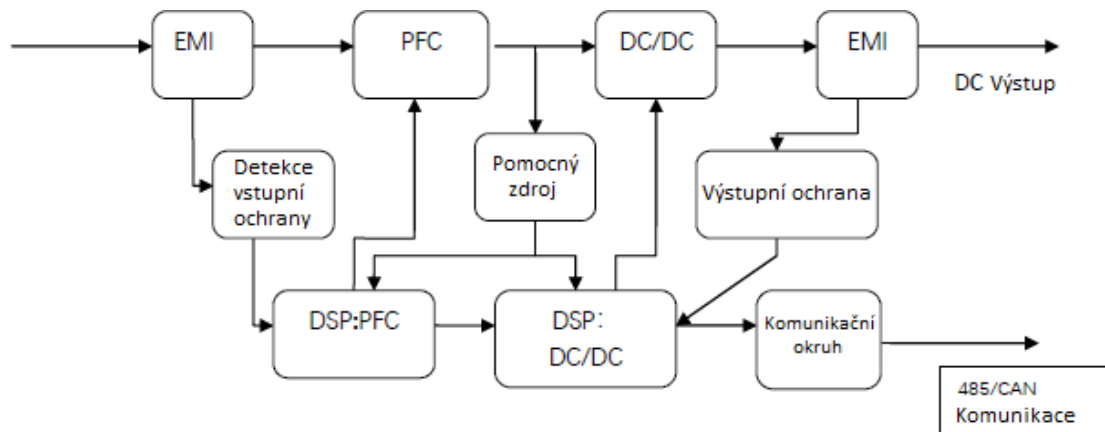
V práci jsou používány tři nabíjecí moduly od společnosti UUGreenPower. Jsou to zdroje UR50060-E, které mají výkon 30 kW, pracují s napětím v rozmezí 150–500 V a proudem v rozmezí 0–75 A. Komunikace s moduly je pomocí sběrnice CAN, ale lze je také propojit sériovou komunikací RS-485. Tyto zdroje mají třífázové napájení, účinnost přesahuje 95 % i při maximálním zatížení a nabízejí vysokou výkonovou hustotu.



Obrázek 1 - Nabíjecí modul UR50060-E [5]

Napájecí modul se skládá z aktivního usměrňovače, třífázového PFC, měniče DC/DC, pomocného zdroje a komplexních mechanismů detekce a ochrany vstupních/výstupních obvodů (EMI). Aktivní obvod PFC řízený DSP dosahuje usměrnění, filtrace a korekce účinníku, čímž splňuje normy EMC [17] s minimálním harmonickým zkreslením. Následný DC/DC měnič reguluje výstupní napětí a zajišťuje stabilní stejnosměrný výstup potřebný pro nabíjení. Pomocný napájecí zdroj generuje potřebné napájení řídicích obvodů z aktivního výstupu PFC. Detekční a ochranné obvody monitorují výstupní napětí, proud a teplotu, což umožňuje přesné řízení a zajišťuje bezpečnost.

Komunikace mezi moduly je umožněna prostřednictvím sběrnice CAN/485, což umožňuje efektivní koordinaci a komunikaci mezi více jednotkami.



Obrázek 2 - Princip funkce modulu UR50060-E [5]

Pro správné zapojení modulu je klíčové dodržet správnou indexaci pinů. CAN komunikace se připojuje na piny 1 a 2, kde port 1 zastává funkci CAN data high (CANH) a pin 2 je určen pro CAN data low (CANL). Piny 3 (485 data high) a 4 (485 data low) jsou rezervovány pro sériovou komunikaci RS-485. Piny 5, 7 a 8 slouží k připojení adresové desky (viz Obrázek 4), zatímco napájení je zajištěno přes piny 10, 11 (VOUT-) a 13, 14 (VOUT+). Zbývající piny, tj. 6 a 9, neplní žádnou specifickou funkci.



Obrázek 3 - Jednotlivé piny konektoru [5]

Adresovací deska slouží k individuálnímu určení adres jednotlivých modulů. Každý modul je vybaven možností nastavení své skupiny a jedinečné adresy. Pro nastavení skupiny slouží tlačítka a menu umístěné přímo na modulu. Navigace v menu probíhá pomocí šipek nahoru a dolů. Šipkami se uživatel dostane na výběr skupin, přičemž dlouhým podržením jakékoli šipky uživatel získá možnost volby skupiny. V této práci jsou všechny moduly zařazeny do skupiny číslo 1. Na každém modulu se nachází vlastní deska s DIP přepínači, zobrazenými v červené (J1) a modré (J2) barvě s čísly 1 až 4 [5]. DIP přepínače mají aktivní a neaktivní polohu, kdy aktivní poloha je definována jako přepnutí nahoru. Pro přidělení adresy modulu se používá následující rovnice:

$$\text{Adresa modulu} = 5 * (\text{J2 v aktivní poloze}) + \text{J1 v aktivní poloze}$$

Maximální hodnota adresy se rovná 24. Veškeré kombinace jsou uvedeny v příloze (viz B). V práci jsou použity adresy 1 pro modul 1, adresa 2 pro modul 2 a adresa 6 pro modul 3.



Obrázek 4 – Adresovací deska napájecích modulů [5]

1.3 Sběrnice CAN

CAN sběrnice je robustní sériová sběrnice, která se používá k propojení různých zařízení v automobilovém průmyslu, průmyslové automatizaci a dalších oblastech. Jedná se o dvou vodičový systém, který umožňuje přenos dat a řídicích signálů mezi různými uzly v systému. CAN sběrnice je známá svou spolehlivostí, odolností proti rušení a schopností pracovat v extrémních podmínkách. Každé zařízení na sběrnici má svou vlastní

adresu a může posílat nebo přijímat data bez zásahu centrálního řízení. To umožňuje efektivní a spolehlivou komunikaci mezi různými částmi systému.

CAN sběrnice funguje na principu uzlů a zpráv. Zařízení připojená ke sběrnici CAN se nazývají uzly. Každý uzel může vysílat a přijímat zprávy obsahující specifické údaje, jako například hodnoty napětí, proudů nebo informace o modulu. Pokud chtějí dvě zařízení odeslat zprávu současně, proběhne mezi nimi krátký arbitrážní proces. Zpráva s vyšší prioritou získá přístup k síti jako první. Zpráva s větší prioritou se odešle všem ostatním uzlům na sběrnici. Každý uzel si zprávu přečte a rozhodne, zda jsou pro něj její informace relevantní. Více informací ke sběrnici CAN (viz [13], [14], [15], [16]).

1.4 CAN komunikace se zdroji

V práci je důležité objasnit strukturu CAN zprávy, která se skládá z identifikačního pole, jež existuje ve dvou formátech. První formát, nazvaný standardní (CAN 2.0 A), obsahuje 11 bitů, zatímco druhý formát, označovaný jako rozšířený (CAN 2.0 B), má 29 bitů. Rozšířený formát CAN zprávy se využívá při řešení této úlohy. Dále zpráva obsahuje datové pole, jehož rozsah je od 0 do 8 bajtů.

1.4.1 ID pole

Jak bylo uvedeno výše, v práci se uplatňuje rozšířený formát CAN zprávy, nazývaný CAN 2.0 B, který disponuje 29 bity v identifikačním poli. Jednotlivé bity hrají v sestavení kompletního identifikačního pole zprávy důležitou roli.

Tabulka 1 – Význam jednotlivých bitů v ID poli

Č. bitů	28:25	24:21	20:14	13:9	8:0
Název	Protokol	Adresa monitoru	Adresa modulu	Datum výroby	Výrobní číslo
Hodnoty	0x01	Monitor - (0x01 - 0x0F) Všechny - 0x00	Modul - (0x01 - 0x7F) Všechny - 0x00	1-31 (volitelné)	0 (volitelné)

První 4 bity ID pole jsou rezervovány pro typ protokolu, kterým se informace šíří po sběrnici CAN z napájecího modulu k monitorovacímu zařízení. Hodnota 0x01 v hexadecimálním formátu představuje typ protokolu, po kterém napájecí moduly UUGreenPower komunikují s HMI obrazovkou. Následující 4 bity nesou informaci o adrese monitorovací jednotky na sběrnici. Hodnota může být v rozmezí 0x01 – 0x0F.

Pokud je zadaná hodnota 0x00, tak se informace přenáší na všechny připojené monitorovací jednotky na sběrnici. V této úloze se využije hodnota 0x01, která představuje základní nastavení. Dalších 7 bitů se využívá pro zadání adresy napájecích modulů. Hodnota může být v rozmezí 0x01 – 0x7F. Při zadané hodnotě 0x00 se informace přenášejí na všechny připojené napájecí moduly na sběrnici. Hodnoty adres napájecích modulů jsou uvedeny výše práci (viz kapitola 1.2). Bity 13 až 9 reprezentují datum výroby napájecího modulu. Je to volitelná informace a v této úloze je těchto 5 bitů vyplněno hodnotou 0x00. Posledních 9 bitů identifikují výrobní číslo napájecího modulu. Stejně jako u bitů pro datum výroby, je tato informace volitelná a je zde zapsaná hodnota 0x00. Je důležité upozornit, že u bitů, které jsou vyhrazeny pro datum a výrobní číslo modulu, není nutné vyplnit hodnoty pro odeslání CAN zprávy. Avšak pro příjem CAN zprávy od modulu jsou do těchto bitů zapisovány reálné hodnoty modulu [7]. Tento fakt je důležitý pro správné nastavení filtru pro příjem zpráv (viz kapitola 1.6.4).

1.4.2 Datové pole

Pro daný rozšířený formát CAN zprávy 2.0 B se v této práci využívá 8 bajtů v datovém poli. Každá hodnota v jednotlivých bajtech datového pole zprávy představuje odlišný požadovaný úkon, který se zasílá na napájecí modul.

Tabulka 2 – Význam jednotlivých bajtů v datovém poli

Č. Bajtu	0		1	2	3	4	5	6	7
Č. bitů	7:4	3:0	7:0	7:0	7:0	7:0	7:0	7:0	7:0
Název	Adresa skupiny	Typ zprávy	Typ příkazu	Rezerva		Data příkazu			
Funkce a hodnoty	0x01 - 0x09 všechny - 0x00	Nastavit data - 0x00 Odpověď modulu na příkaz nastavit data - 0x01 Vyčist data - 0x02 Odpověď modulu na příkaz vyčist data - 0x03	Vout - 0x00 Iout_pomalý - 0x01 Vout_referenční - 0x02 Iout_Limit - 0x03 Vypnutí_DCDC - 0x04 Status_Modulu - 0x08 Vab - 0x20 Vbc - 0x21 Vca - 0x22 Adresa_skupiny - 0x23	x	x	Hodnoty napětí a proudů (mA, mV) v hexadecimálním formátu.			

Byte číslo 0 v datovém poli CAN zprávy se skládá ze 4 vyšších bitů (7-4) pro adresaci skupiny a 4 nižších bitů (3-0) pro definování typu zprávy. To umožňuje flexibilní komunikaci mezi PLC a napájecími moduly v síti CAN. Byte číslo 1 v datovém poli CAN zprávy slouží k určení typu příkazu zprávy. Patří sem nastavení výstupních napětí, referenčních hodnot, výstupního proudu, proudových limitů, zapnutí/vypnutí modulu a dotazy na stav modulu. V kombinaci s bytem 0 určuje, zda se bude číst nebo zapisovat zadaný parametr. To umožňuje precizní kontrolu nad moduly v síti CAN.

Následující 2 bajty v datovém poli CAN zprávy jsou rezervní. Nemají žádnou specifickou funkci a lze je vyplnit pouze hodnotami 0x00. Byte číslo 4 až 7 v datovém poli CAN zprávy slouží k zadání hodnot napětí a proudů v mili jednotkách (mV a mA) v hexadecimálním formátu. Například pro nastavení výstupního referenčního napětí na 320 V na modulu číslo 2 je nutné nejprve převést hodnotu na 320000 mV a poté ji konvertovat do hexadecimálního formátu. Hodnota 320000 mV v hexadecimálním formátu odpovídá hodnotě 4E200. Tato hodnota se pak zapíše do bajtů 4 až 7 v pořadí 0x00 (Byte 4), 0x04 (Byte 5), 0xE2 (Byte 6) a 0x00 (Byte 7) [7].

ID pole									
Č. bitů	28:25	24:21	20:14	13:9	8:0				
Název	Protokol	Adresa monitoru	Adresa modulu	Datum výroby	Výrobní číslo				
Hodnoty	0x01	0x00	0x02	0x00	0x00				
DATA pole									
Č. Bajtu	0		1	2	3	4	5	6	7
Č. bitů	7:4	3:0	7:0	7:0	7:0	7:0	7:0	7:0	7:0
Název	Adresa skupiny	Typ zprávy	Typ příkazu	Rezerva		Data příkazu			
Funkce a hodnoty	0x01	0x00	0x02	0x00	0x00	0x00	0x04	0xE2	0x00

Obrázek 5 – Vzor CAN zprávy pro nastavení referenčního napětí

1.5 Komunikační modul CAN

Na základě řešení modulů pro komunikaci po sběrnici CAN od společnosti Siemens byl vybrán komunikační modul CM CAN pro distribuovanou periferní stanici ET 200SP. Modul umožňuje komunikaci bez protokolu CANOpen a odesílání samostatných datových zpráv po dvojvodičové CAN sběrnici. Pro danou problematiku je metoda použití datových zpráv plně dostačující a jejich následná konfigurace v příloženém HSP je snadná a efektivní. Není nutné používat nadřazené a podřízené prvky komunikace CAN, jelikož napájecí moduly komunikují přímo s komunikačním modulem CM CAN. Modul umožňuje komunikaci s využitím protokolu 2.0B, který disponuje 29bitovým identifikačním polem a 8bajtovým datovým polem. Právě tyto parametry jsou nezbytné pro komunikaci mezi napájecími zdroji a komunikačním modulem CM CAN [4].

Modul je také kompatibilní s PLC řady SIMATIC S7-1200 a 1500. Vzhledem k tomu, že na reálném zařízení je použito PLC řady S7-1500 s CPU 1511-1 PN a distribuovanou periferní stanicí ET 200SP, představuje výběr komunikačního modulu nejvhodnější řešení.



Obrázek 6 – Komunikační modul ET 200SP CM CAN [4]

1.5.1 CANopen Manager

CANopen manager je provozní režim sítě CANopen, který řídí a konfiguruje připojená zařízení CANopen slave. Plně podporuje standardní komunikační protokoly CANopen a umožňuje konfiguraci pomocí SDO a výměnu dat v reálném čase s PDO. Manager řídí správu sítě, včetně stavu NMT zařízení a jejich spouštění dle norem CiA a umožňuje spravovat až 60 slave zařízení. Je ideální pro centralizované řízení a konfiguraci rozsáhlých CANopen sítí, například v průmyslové automatizaci, budovách, medicíně a dopravě. Oba režimy, manager i slave, vyžadují dodržování specifických protokolů CANopen definovaných asociací CiA [4] [8] [11].

1.5.2 CANopen Slave

CANopen slave zařízení, řízené CANopen managerem, plní v síti roli dodavatele dat a funkcí. Dodržuje komunikační protokol CANopen slave dle normy CiA 302 Part 2 a reaguje na zprávy SDO a PDO iniciované managerem. Rovněž reaguje na zprávy NMT odeslané managerem. Zařízení je omezeno na komunikaci s jedním CANopen managerem a je ideální pro aplikace, kde zařízení nabízí specifické funkce v síti CANopen

(například senzory, akční členy). Oba režimy, CANOpen manager i slave, vyžadují dodržování specifických komunikačních protokolů CANOpen definovaných asociací CiA [4] [8] [11].

1.5.3 CAN Transparent

Provozní režim CAN Transparent slouží pro přímou výměnu surového datového rámce CAN mezi modulem a řídicí jednotkou, čímž obchází protokol CANOpen. Nepodporuje funkce CANOpen a uživatelský program řídí přenos a příjem zpráv na základě předkonfigurovaných ID a délek dat, případně dynamického nastavení ID a délek. Nezapojuje se do správy sítě CANOpen a neovlivňuje tak škálovatelnost sítě. Jeho hlavní funkcí je zprostředkování výměny dat na nízké úrovni mezi zařízeními, která nepotřebují funkce CANOpen, ale chtějí si vyměňovat data přes CAN sběrnici [4].

1.5.4 Porovnání

Vzhledem k problematice této práce byl vybrán provozní režim CAN Transparent. I když se na první pohled zdá, že CANOpen Manager je vhodnější volbou, v tomto specifickém případě s energetickými zdroji UUGreenPower a PLC má provozní režim CAN Transparent své opodstatnění. Hlavním argumentem pro CAN Transparent je jednoduchá a přímá komunikace, která dává plnou kontrolu nad komunikací. V tomto případě není potřeba složité funkce CANOpen, jako je centralizované řízení a konfigurace, správa sítě ani škálovatelnost pro velké množství zařízení. Místo toho se práce zaměřuje na přímou výměnu dat mezi PLC a energetickými zdroji. Díky CAN Transparent může být definován formát a obsah datových telegramů, čímž se dosáhne maximální flexibility a optimalizace pro specifické potřeby, což může být užitečné, pokud jsou specifické požadavky na datovou výměnu, které standardní protokoly CANOpen nepodporují. Navíc provozní režim CAN Transparent snižuje složitost implementace a údržby, což ušetří čas a zdroje, zejména v případě menšího systému.

1.6 CAN Transparent

V provozním režimu CAN Transparent jsou všechny funkce CANOpen deaktivovány a modul komunikuje přes sběrnici pouze s protokolem CAN 2.0A(B) podle normy ISO 11898-2. [12] Nastavení režimu modulu se provádí pomocí přiřazeného HSP v TIA Portal. Uvedený provozní režim má pouze 2 stavy. Stav "OFF" a "ON". Ve stavu "OFF"

se nepřenášejí ani nepřijímají žádné CAN zprávy nebo chybové rámce, modul se neúčastní komunikace na sběrnici CAN. Tento stav modul použije, když není připojen k řídicí jednotce SIMATIC S7 nebo pokud má datová vazba ze SIMATIC S7 stav "špatný poskytovatel I/O". V opačném stavu, "ON", se modul účastní komunikace na sběrnici CAN, odesílá a přijímá CAN zprávy, s výjimkou případů, kdy je řídicí jednotka CAN v stavu "Bus-Off". Pro přenos dat po sběrnici CAN se dají využít 2 možnosti. První z možností je pomocí konfigurovatelných CAN zpráv a druhá možnost je za využití programovatelných CAN zpráv neboli proxy [6].

1.6.1 Konfigurovatelné zprávy CAN

Konfigurovatelné CAN zprávy v TIA portálu mají pevné ID a délku a lze je nastavit v HSP. ID zpráv podporuje standardní 11bitový i rozšířený 29bitový formát CAN 2.0A(B). Řízení toku není k dispozici, takže rychlý příjem dat z CAN sběrnice může přepsat mezilehlé hodnoty. Změny na sběrnici CAN nemusí být viditelné, protože ztráta mezilehlých hodnot je přípustná. Pevné zprávy se přenášejí buď po změně dat, nebo cyklicky. V režimu cyklického přenosu se uživatelská data CAN zpráv přenášejí mezi PLC a modulem v cyklickém obrazu vstupů a výstupů. Funkce "Transmit Messages" umožňuje modulu výstupního procesního obrazu odesílat zprávy CAN při změně dat, po přijetí rámce CAN RTR s odpovídajícím ID zprávy, při přechodu do stavu "ON" a cyklicky podle nastavení. Spouštěče odesílání zpráv jsou nezávislé, takže i když je zpráva odeslána na základě změny dat nebo RTR, cyklický vysílací cyklus probíhá stále. Funkce "Receive Messages" umožňuje modulu přijímat zprávy CAN, které se cyklicky přenášejí do PLC. Přijaté hodnoty dat jsou pomocí PLC přiřazeny podle konfigurace modulu. Před odesláním dat do PLC modul upraví pořadí bajtů, aby se sjednotila "endianita". Modul také sleduje příjem zpráv CAN pro všechna nastavená ID a informace o nich předává PLC. Pokud nebyla přijata žádná zpráva CAN, do PLC je odeslána hodnota nula. Při odesílání modul upraví pořadí bajtů u pevných zpráv CAN, aby odpovídalo "Big Endian" formátu sítě PROFINET. Po úspěšném nakonfigurování a následném zkompileování hardwarové konfigurace v programovacím prostředí TIA Portal se automaticky vytvoří proměnné zadaných zpráv v defaultní tabulce tagů [6] [9].

1.6.2 Programovatelné zprávy CAN (proxy)

V režimu "CAN transparent" lze používat programovatelné zprávy CAN, jejichž ID a délka jsou definovány uživatelským programem za běhu. Kromě vlastních dat se přenášejí i informace o protokolu mezi PLC a modulem v cyklickém obrazu IO. Formát ID zprávy určuje bit číslo 29 v ID zprávy. Nastavení bitu znamená použití rozšířeného formátu ID zprávy s 29 bity, jinak se použije standardní formát. Uživatelský program může odesílat a přijímat zprávy CAN pomocí vysílacích a přijímacích proxy. V režimu "CAN transparent" lze konfigurovat "Transmit Proxies" a "Receive Proxies" v TIA Portalu. Tyto proxy slouží jako "komunikační kanály", které umožňují uživatelskému programu odesílat a přijímat libovolné zprávy CAN, které nebyly předem nakonfigurovány. Použití více vysílacích/přijímacích proxy serverů usnadňuje programování, protože jsou nezávislé na komunikačních kanálech, což umožňuje flexibilnější a efektivnější programování. Vysílací proxy server umožňuje různým částem kódu odesílat zprávy CAN bez nutnosti synchronizace. Uživatelský program může odesílat zprávy CAN pomocí vysílacího proxy, který obsahuje cyklická vstupní a výstupní data PROFINET. Pořadí bajtů v odesílaných zprávách CAN se nemění v modulu. Přijímací proxy server umožňuje filtraci zpráv CAN podle ID a tím usnadňuje zpracování v uživatelském programu. Uživatelský program může přijímat zprávy CAN pomocí proxy pro příjem. Modul spravuje seznam ID zpráv pro příjem, který lze předkonfigurovat nebo za běhu upravit. Přijaté zprávy jsou uloženy do vyrovnávací paměti a přeneseny do řídicí jednotky S7 v cyklických vstupních datech. Maximální počet vysílacích a přijímaných proxy se odvozuje z kapacity I/O daného komunikačního modulu CM CAN. Tento modul je limitován na 255 volných I/O dat, což znamená, že je schopen odeslat maximálně 19 vysílacích proxy a 8 přijímacích proxy [6].

1.6.3 Porovnání

Konfigurovatelné zprávy CAN nabízejí pevné ID a délku, což omezuje možnost upravit je za běhu, existuje riziko ztráty dat v důsledku rychlého příjmu, který může přepsat mezilehlé hodnoty, což může způsobit nekonzistenci dat. Spouštění zpráv je omezeno na změnu dat nebo cyklický přenos, bez možnosti detailního nastavení spouštěčů, a struktura dat je předdefinovaná, což omezuje kontrolu uživatele nad formátem a strukturou dat. Na druhou stranu, programovatelné zprávy CAN (proxy) umožňují dynamické nastavení ID a délky zpráv za běhu, což poskytuje maximální flexibilitu, zajišťují ochranu

dat tím, že modul sleduje příjem zpráv a v případě jejich absence odesílá nulu, čímž se minimalizuje riziko ztráty dat. Uživatelský program má možnost přesně definovat podmínky pro spouštění zpráv a plnou kontrolu nad formátem a strukturou dat dle specifických potřeb. Z těchto důvodů a také kvůli snadší manipulaci v programovacím prostředí TIA portal byla pro práci vybrána možnost s použitím programovatelných CAN zpráv.

1.6.4 Filtry pro příjem programovatelných CAN zpráv

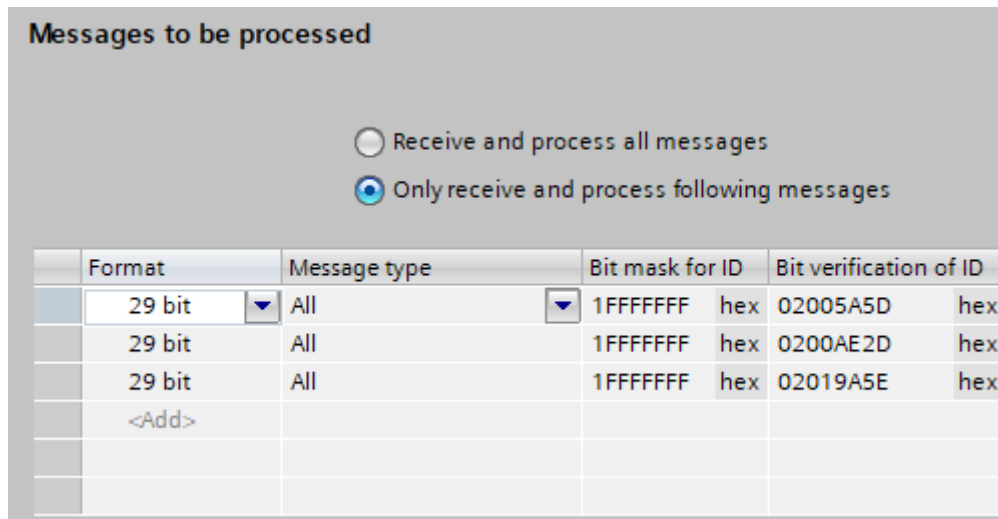
Další klíčovou součástí správné konfigurace je nastavení filtru pro příjem CAN zpráv. V přiřazeném HSP v programovacím prostředí TIA Portal se nachází část nazvaná "Receive proxy definition", kde lze pro jednotlivé proxy nastavit, zda se mají přijímat všechny zprávy nebo pouze zprávy s určitým ID. Pokud je zaškrtnuta možnost přijímání veškerých zpráv, na výstupech funkčního bloku pro příjem transparentních zpráv se objevují jak požadované zprávy, tak i zprávy, které si například posílají napájecí moduly mezi sebou. Tyto dodatečné zprávy jsou nežádoucí a nepodstatné pro danou úlohu a zabírají místo v zásobníku přijatých zpráv. Maximální počet zpráv v tomto zásobníku lze rovněž nastavit v uvedené části HSP, kde maximální hodnota je 100 zpráv. Pro zajištění správné funkce úlohy a uvolnění kapacity v zásobníku přijatých zpráv je nutné vybrat pouze zprávy s předem nastaveným ID. V tomto případě se jedná o zprávy s ID jednotlivých modulů, které obsahují i datum výroby s výrobním číslem. Jak již bylo zmíněno, pro odesílání zpráv nejsou bity, které reprezentují tyto parametry nutné, avšak pro přijaté zprávy od modulu obsahující reálné hodnoty těchto parametrů, je nezbytné do nastavení filtru uvést ID zprávy, které tyto bity parametrů obsahují. Adresace a ID těchto zpráv jsou uvedeny v tabulce (viz Tabulka 3).

Tabulka 3 – Adresace jednotlivých modulů

Modul	Protokol	Monitor	Modul	Datum	Vyrobní číslo	ID (hexa)	ID pro filtr (hexa)
	29:25	24:21	20:14	13:9	8:0		
Modul 1 (adresa = 1)	0001	0000	0000001	00000	000000000	02004000	02005A5D
Modul 2 (adresa = 2)	0001	0000	0000010	00000	000000000	02008000	0200AE2D
Modul 3 (adresa = 6)	0001	0000	0000110	00000	000000000	02018000	02019A5E
Všechny moduly	0001	0000	0000000	00000	000000000	02000000	x

Zásobník přijatých zpráv je nastaven maximálně na pět přijatých zpráv. Důležitými parametry pro nastavení filtru přijatých zpráv, které se ukládají do zásobníku přijatých zpráv, je maska a kritérium bitů. Pole kritéria bitů přesně definuje ID zprávy v hexadecimálním formátu, které má být propuštěno do zásobníku přijatých zpráv. Maska se nastaví tak, aby

všechny bity měly hodnotu 1, což zajistí, že všechny bity ID uvedené v kritériu bitů budou brány v úvahu. Například pro filtrování zpráv pouze z modulu 1 do zásobníku přijatých zpráv se do pole kritéria bitů zapíše hodnota 02005A5D a do pole masky se zapíše hodnota 1FFFFFFF. Tím se zajistí, že se do zásobníku přijatých zpráv přijímají pouze zprávy s identickým ID [6].



Obrázek 7 – Nastavení filtru přijatých zpráv uvnitř HSP

1.6.5 Datový blok a proměnné modulu CM CAN

Po úspěšné konfiguraci a následné kompilaci modulu pomocí přiděleného HSP v programovacím prostředí TIA Portal se automaticky vygeneruje datový blok, ve kterém je uloženo poslední nastavení nahrané konfigurace. Dále jsou automaticky vytvořeny dvě proměnné pro monitorování stavu modulu a pro zapnutí a vypnutí modulu, aby se modul mohl zapojit do komunikace na sběrnici CAN. Proměnná pro monitorování stavu modulu je nazvána "Transparent_status" a proměnná pro zapnutí a vypnutí modulu je nazvána "Transparent_control". Tyto proměnné jsou uloženy v defaultní tabulce tagů a jsou typu byte [6].

2 Softwarové řešení zařízení

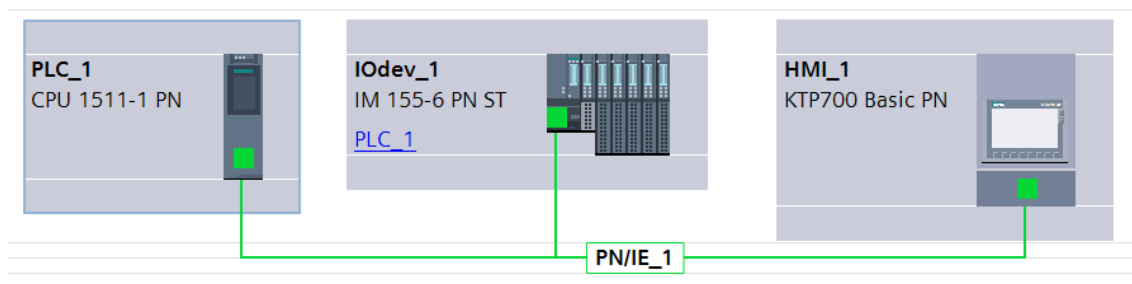
Další kapitola se zabývá softwarovým řešením zařízení. Kompletní program včetně vizualizace je vytvořen pomocí programu Siemens TIA Portal V18. Program je logicky členěn do sekcí a vytvořen především v jazyce FBD pro lepší a jednodušší čitelnost. Je zde také použit jazyk SCL, který je vhodnější pro složitější matematické výpočty nebo stavové automaty. Nejdůležitější částí kódu je komunikace s napájecími moduly, která probíhá skrze komunikační modul ET 200SP CM CAN spolu s aktuální blokovou knihovnou CANopen TransparentProxy V17.

Hlavní program (Main) je volán cyklicky a obsahuje funkce a funkční bloky jako funkci pro zobrazení statusu komunikačního modulu ET 200SP CM CAN, pomocnou funkci datového přetypování pro HMI vizualizaci nebo funkční bloky pro konfiguraci modulu, odeslání a přijímání transparentní zprávy skrze modul a nastavení filtru zpráv. K jednotlivým funkčním blokům jsou přiřazeny datové bloky, které obsahují vstupy, výstupy nebo statické proměnné. K určitým proměnným jsou také přiděleny vytvořené datové typy z výše zmíněné knihovny.

2.1 Hardwarová konfigurace v TIA Portal

Hardwarová konfigurace vytvořená v programovacím prostředí TIA Portal zahrnuje PLC řady SIMATIC S7-1500, konkrétně CPU 1511-1 PN s verzí firmwaru 02.09.04, který je propojen s distribuovanou periferní stanicí ET 200SP, IM 155-6 PN ST, přes profinet kabel. V této konfiguraci jsou integrovány moduly digitálních vstupů, digitálních výstupů a komunikační modul CM CAN s verzí firmwaru 01.01.00. Dále zahrnuje HMI obrazovku KTP700 Basic verze 15.01.0.0 pro vizualizaci a dva technologické moduly TM ECC PL ST pro řízení a monitorování nabíjení elektrických vozidel. Tyto dva moduly jsou vzhledem k zadání práce irelevantní, protože není nutné připojit skutečné elektrické vozidlo. Konfigurace se zaměřuje pouze na propojení PLC, HMI a napájecích modulů, které jsou spojeny pomocí dvou vodičů do komunikačního modulu CM CAN a tím pádem nejsou zahrnuty v hardware konfiguraci TIA portálu. Je klíčové, aby hardwarová konfigurace vytvořená v programovacím prostředí TIA Portal odpovídala reálným modulům, aby výměna dat mezi jednotlivými prvky probíhala v pořádku.

Důležité je také správně nastavit IP adresy pro komunikaci mezi zařízeními a zabránit překrývání adres na síti. Jedinečné IP adresy byly přiřazeny PLC, HMI obrazovce a distribuované periferní jednotce ET 200SP.



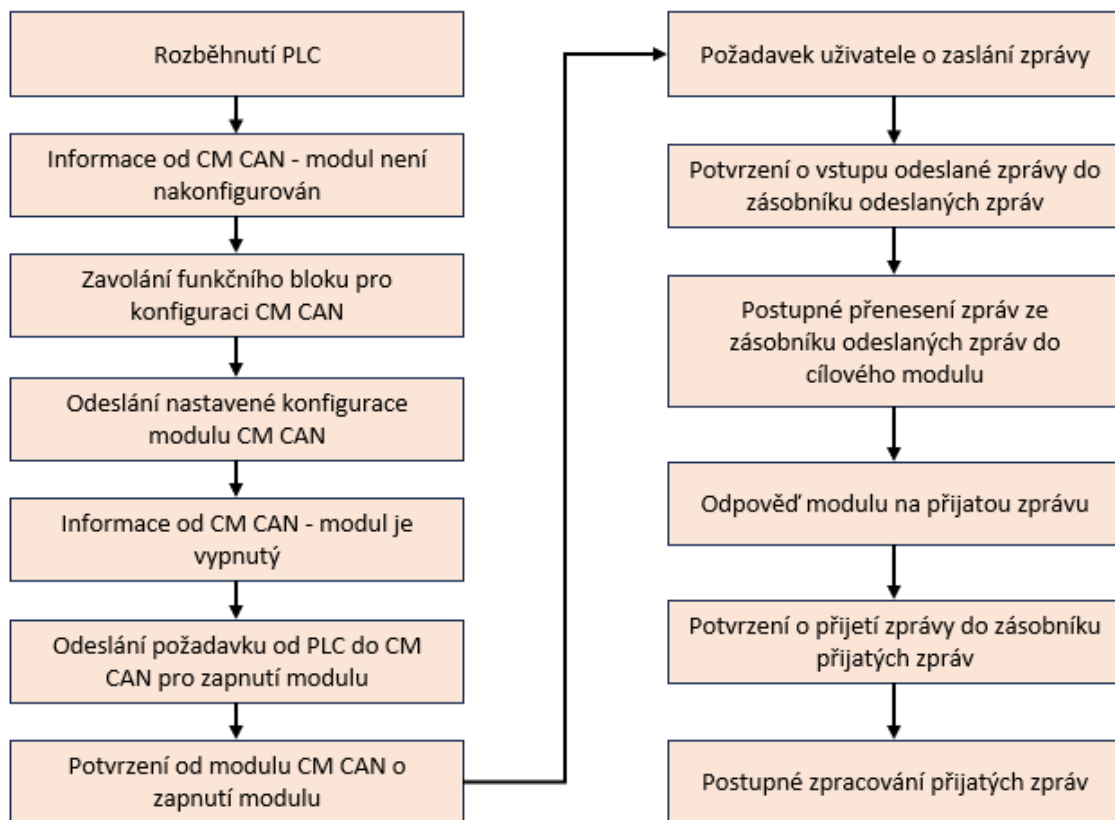
Obrázek 8 – Hardwarová konfigurace v TIA portálu

2.2 Řešení logické struktury programu a tagů

Aby byl kód snadno čitelný, je důležité dobře strukturovat program a vhodně označovat jednotlivé proměnné. Proto jsou použity jednoznačné názvy proměnných, což zvyšuje srozumitelnost a usnadňuje interpretaci kódu. Struktura tagů odesílaných zpráv je systematicky vytvořena takto <Modul>_<Příkaz>_<Typ>, což umožňuje snadnou identifikaci modulu a významu proměnné. Například proměnná M1_Real_Uout označuje skutečnou hodnotu výstupního napětí na modulu M1. Byl vytvořen globální datový blok „Transsparent_messages“, který obsahuje pole struktur pro nakonfigurované „Transmit proxies“, „Receive proxies“ a filtr „Receive proxies“. Pole jsou navržena na maximální počet přijatých a odeslaných proxy. Datový blok je vytvořen za účelem univerzálnosti řešení a také ke snadnější úpravě programu dle budoucích potřeb uživatele. Samotný program je organizován do několika složek. Hlavní program se nachází v první organizační složce a je rozdělen do podsítí, které odpovídají jednotlivým požadavkům programu. Následující organizační složky se věnují stavu komunikačního modulu CM CAN, jeho konfiguraci a operacím, jako je odesílání, přijímání CAN zpráv a nastavování filtru přijatých zpráv. V těchto složkách se převážně nacházejí funkce, funkční bloky a datové bloky.

Následující diagram toku dat (viz Obrázek 9) znázorňuje, jak se program chová od rozběhnutí PLC. Jako první přijde informace od komunikačního modulu CM CAN do PLC, že komunikační modul není nakonfigurován. Tuto zprávu pod sebou nese proměnná "Transparent_Status" s hodnotou bajtu 0010 1000. Dále se zavolá funkční blok "ET200SPCM_CANConfig", který následně odešle komunikačnímu modulu poslední

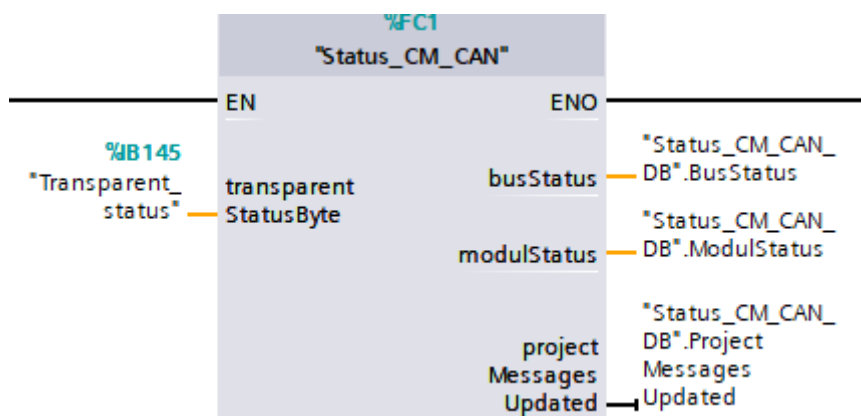
nahranou a zkompilovanou konfigurací komunikačního modulu. Po úspěšném odeslání konfigurace komunikačnímu modulu, CM CAN zašle PLC informaci o vypnutém modulu. Tato informace je viditelná v proměnné "Transparent_Control", která nese hodnotu 16x00. Proto, aby se komunikační modul mohl účastnit jakékoliv komunikace na sběrnici CAN, je nutné, aby tento byte byl nastaven na hodnotu 16x01. Po zapnutí komunikačního modulu CM CAN je uživatel schopen odeslat žádanou zprávu prostřednictvím proxy zprávy. Pomocí funkce na zpracování odeslaných zpráv se zadaný úkon přenesse do funkčního bloku pro odesílání zpráv. Následuje potvrzení o vstupu odeslané zprávy do zásobníku odeslaných zpráv a následné přenesení informace do cílového modulu, který si uživatel zažádal. Jakmile napájecí modul obdrží tuto zprávu, odešle zprávu s potvrzením o přijetí odeslané zprávy do zásobníku přijatých zpráv. Po přijetí zprávy se odešle potvrzení o validním přijetí zprávy pomocí funkčního bloku pro přijetí zpráv a následně se tato datová zpráva zpracuje pomocí funkce pro zpracování přijatých zpráv.



Obrázek 9 – Diagram toku dat od rozběhnutí PLC

2.3 Status CM CAN

Funkce je napsána v jazyce SCL a slouží k zobrazení statusu komunikačního modulu CM CAN v transparentním režimu. Funkce je převzata z knihovny komunikačního modulu CM CAN. V kódu jsou definovány speciální proměnné pro zpracování informací, které jsou přijaty z CAN sběrnice. Tyto informace jsou uloženy v proměnné nazvané „Transparent_Status“, která obsahuje různé bity reprezentující stav sběrnice, aktualizace zpráv projektu a stav modulu. Proměnná „Transparent_Status“ se využívá jako permanentní vstup do funkčních bloků pro odesílání, přijímání a nastavení filtru zpráv. Dle statusu komunikačního modulu CM CAN se ovlivňují procesy dat ve funkčních blocích. Pro získání konkrétních informací z této proměnné jsou použity různé bitové operace, jako je například operace AND, která izoluje určité bity podle definovaných maskovacích hodnot. Tyto extrahované informace jsou poté uloženy do dočasných proměnných, jako je „tempBusStatus“, „tempprojectMessagesUpdated“ a „tempModulStatus“, které slouží k uchování těchto specifických stavových informací (viz příloha C). Proto, aby se modul mohl účastnit komunikace na sběrnici CAN je nutné, aby status měl hodnotu 3, která říká, že je vše v pořádku. Je také nutnost pro tento stav, aby byl modul CM CAN zapnutý neboli hodnota proměnné „Transparent_Control“ byla 16x01. V opačném případě se na statusu objeví hodnota 0, která říká, že modul je vypnutý. Dále jsou v kódu dočasné proměnné převedeny na trvalejší formát pomocí konverzních funkcí, aby byly snadno použitelné v rámci PLC programu.



Obrázek 10 – Funkce pro zjištění statusu CM CAN v transparentním režimu

2.4 Konfigurace CM CAN

Jedná se o speciální funkční blok z knihovny komunikačního modulu CM CAN, který přenáší přiřazené parametry do komunikačního modulu ET 200SP CM CAN. Nejdříve je podstatná samotná konfigurace modulu v hardwarové konfiguraci. Po přidání komunikačního modulu CM CAN do periferní stanice ET 200SP v hardwarové konfiguraci TIA Portal se zobrazí HSP pro konfiguraci přidaného modulu. Prvním krokem správného nakonfigurování modulu bylo zvolení transparentního provozního režimu v sekci „Module parameters“. Přenosová rychlost komunikačního modulu je nastavena na 125 kbit/s. Sekce „Transmit message“ a „Receive message“ je prázdná, protože v práci se využívá programovatelných CAN zpráv proxy a nikoli konfigurovatelné zprávy CAN. Dalším krokem bylo vytvoření příslušných proxy zpráv pro odesílání a přijímání. Komunikační modul komunikuje se třemi napájecími zdroji, a proto byly vytvořeny tři „Transmit proxies“ pro odesílání zpráv jednotlivým napájecím modulům a jedna „Receive proxies“ pro příjem zpráv ze všech modulů. Každá Proxy má své vstupní a výstupní adresy (viz Obrázek 11). Pro jednotlivé adresy bylo nutné nadefinovat příslušné proměnné v tabulce tagů „CAN_transparent“. Definované proměnné mají datový typ převzatý z knihovny komunikačního modulu a slouží jako vstupní a výstupní oblast příslušných funkčních bloků pro odesílání a přijímání transparentních CAN zpráv. Předdefinované datové typy jsou k nalezení ve složce „PLC data types“. Posledním krokem konfigurace bylo nastavení filtrů přijímací proxy a nadefinování maximálního počtu přijatých zpráv. Nastavení těchto parametrů je vysvětleno v kapitole pro nastavení filtrů přijatých CAN zpráv (viz kapitola 1.6.4).

Index	Name	I address	Q address
1	hex M1_Send_Proxy	146	338 .. 350
2	hex M2_Send_proxy	147	351 .. 363
3	hex M3_Send_Proxy	148	364 .. 376

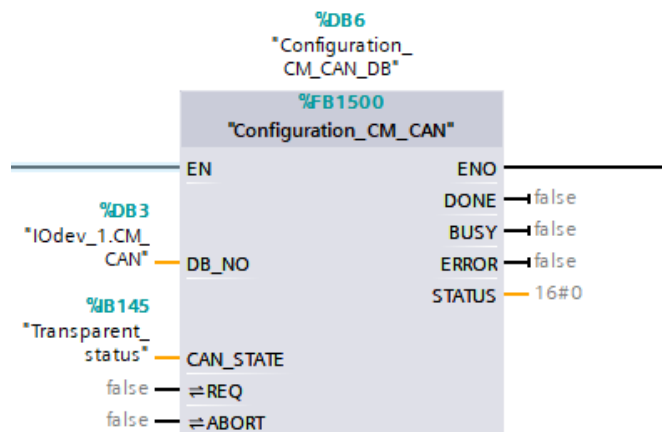
Index	Name	Max. numb...	Receive and..	Subnumber	I address	Q address
1	hex Receive_proxy	64	No	3	149 .. 161	377

CAN_transparent			
	Name	Data type	Address
1	◀▶ M1_IN_Area_Send_Proxy	"LCan_typeSendInTransparentMessage"	%I146.0
2	◀▶ IN_Area_Receive_Proxy	"LCan_typeRcvInTransparentMessage"	%I149.0
3	◀▶ OUT_Area_Receive_Proxy	"LCan_typeRcvOutTransparentMessage"	%Q377.0
4	◀▶ M1_OUT_Area_Send_Proxy	"LCan_typeSendOutTransparentMessage"	%Q338.0
5	◀▶ M2_IN_Area_Send_Proxy	"LCan_typeSendInTransparentMessage"	%I147.0
6	◀▶ M2_OUT_Area_Send_Proxy	"LCan_typeSendOutTransparentMessage"	%Q351.0
7	◀▶ M3_IN_Area_Send_Proxy	"LCan_typeSendInTransparentMessage"	%I148.0
8	◀▶ M3_OUT_Area_Send_Proxy	"LCan_typeSendOutTransparentMessage"	%Q364.0

Obrázek 11 – Přehled nakonfigurovaných proxy a jejich příslušné proměnné

Po kompilaci hardwarové konfigurace se automaticky vygeneruje datový blok „<HW název I/O zařízení>.<HW název komunikačního modulu>_<DB>“ s nastavenou konfigurací modulu. Po vyžádání konfigurace od komunikačního modulu CM CAN se funkční blok zavolá v procesu runtime a přenesení poslední nahranou konfiguraci do komunikačního modulu.

Vstupy funkčního bloku zahrnují „CAN_STATE“, který poskytuje informace o stavu modulu, „DB_NO“, kam se ukládají data určená k přenosu do modulu. Na vstup „DB_NO“ se přiřadí kompletní datový blok, který vznikne po úspěšné kompilaci konfigurace komunikačního modulu. Na vstup „CAN_STATE“ se přiřadí proměnná „Transparent_Status“ typu byte. Dále funkční blok obsahuje vstupy/výstupy „REQ“, který aktivuje přenos dat po náběžné hraně signálu a „ABORT“, který po aktivování náběžné hrany přerušuje probíhající přenos dat. Tyto I/O jsou určeny k manuálnímu ovládání komunikace po sběrnici CAN. Komunikační modul CM CAN lze zapnout pomocí proměnné „Transparent_Control“ nebo náběžnou hranou vstupu „REQ“. Na výstupu funkčního bloku jsou čtyři proměnné. „BUSY“ označuje probíhající přenos dat, „DONE“ indikuje úspěšné dokončení přenosu a „ERROR“ signalizuje chybu během přenosu. Signál „STATUS“ informuje o stavu funkčního bloku (viz příloha D). Funkční blok nejprve provádí kontrolu stavu modulu a datového bloku. Pokud jsou v pořádku, spustí se přenos dat po přijetí signálu „REQ“. Během přenosu je aktivní signál „BUSY“. Po dokončení se aktivuje signál „DONE“. Pokud dojde k chybě, aktivuje se signál „ERROR“. Stav funkčního bloku je k dispozici na výstupu „STATUS“, který je přenesen s podrobnějším popisem na diagnostickou obrazovku HMI vizualizace (viz Obrázek 21).



Obrázek 12 – FB pro konfiguraci modulu ET 200SP CM CAN

2.5 Odeslání transparentních zpráv

Pro zajištění správného odeslání transparentních CAN zpráv přes komunikační modul CM CAN byl vytvořen nadřazený funkční blok pro zpracování odesílaných zpráv. Tento blok spolupracuje s funkčním blokem z knihovny komunikačního modulu CM CAN. Pro každý modul existuje samostatný funkční blok propojený s nadřazeným blokem pro odesílání transparentních CAN zpráv. Zprávy se do modulů odesílají cyklicky s nastaveným intervalem, čímž se zabrání přetížení CAN sběrnice. Po spuštění modulu se na vybraný modul v rámci jednoho cyklu nejprve odešle zpráva pro jeho zapnutí. Následuje 10 zpráv sloužících k nastavení parametrů nebo jejich čtení a po požadavku na vypnutí modulu se odešle jedenáctá zpráva pro vypnutí modulu. Pro zapnutí a vypnutí všech modulů najednou slouží samostatný funkční blok, který odesílá pouze zprávy pro zapnutí a vypnutí modulů.

2.5.1 FB pro zpracování odesílaných zpráv

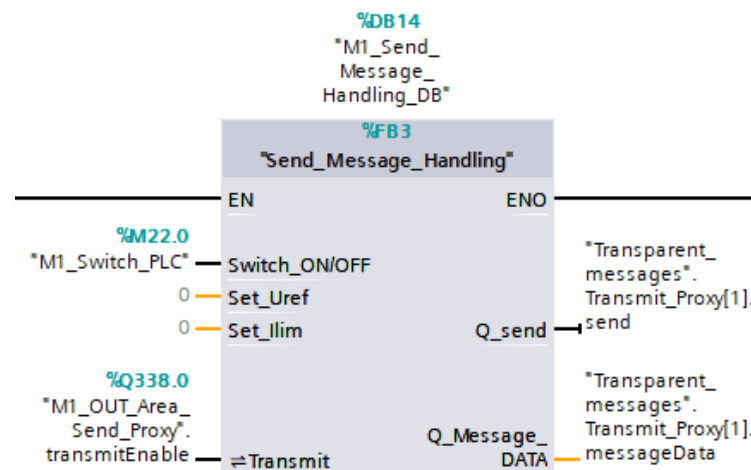
Vnitřní SCL kód funkčního bloku "Send_Message_Handling" obsahuje stavový automat, který na základě vstupních dat odesílá funkčnímu bloku "Send_Transparent_Message" potřebné informace pro odeslání transparentní CAN zprávy do určeného modulu. Jedná se o výstupy "Q_Send", které aktivují vstup "execute" na bloku pro odesílání transparentních CAN zpráv, a výstup "Q_Message_DATA", sloužící jako bajtové datové pole pro vstup "messageData" bloku "Send_Transparent_Message". V tomto poli bajtů je zakódována příslušná zpráva. Vstupy funkčního bloku "Send_Message_Handling" jsou proměnné představující reálné hodnoty napětí a proudů v jednotkách voltů a ampér. Tyto proměnné se nazývají "Set_Uref" a "Set_Ilim". Zprávy pro nastavení těchto parametrů zahrnují převod jednotek na milivoly a miliampéry, jelikož se hodnota zapisuje do posledních čtyř bajtů datového pole. Do těchto bajtů se hodnoty musí konvertovat do hexadecimálního formátu. Z tohoto důvodu je nutné zadané hodnoty nejprve vynásobit tisícem a poté je převést z datového typu "Dint" na datový typ "DWord". Dalším I/O funkčního bloku „Send_Message_Handling“ je „transmit“, kterému je přiřazen bit „transmitEnable“ z proměnné pro výstupní část vysílané proxy. Každému modulu musí být přiřazena odpovídající výstupní část vysílací proxy. Pokud je bit nastaven, jsou odpovídající přijatá data odesílané zprávy zpracována a odeslána do zásobníku odesílaných zpráv. Po nastavení bitu na hodnotu „false“ je možné přijmout další odesílanou zprávu ke zpracování. Sekvence se spustí po požadavku na zapnutí určeného modulu.

Se spuštěním jakéhokoli modulu se nastaví hodnota proměnné „Transparent_control“ na 16x01, tato hodnota zaručuje, že se komunikační modul CMCAN účastní komunikace na sběrnici CAN.

Následně se odesílají zprávy v tomto pořadí:

- 1.) Zpráva pro zapnutí modulu.
- 2.) Zpráva pro nastavení výstupního referenčního napětí.
- 3.) Zpráva pro nastavení proudového omezení.
- 4.) Zpráva pro čtení výstupního napětí.
- 5.) Zpráva pro čtení výstupního proudu.
- 6.) Zpráva pro čtení výstupního referenčního napětí.
- 7.) Zpráva pro čtení proudového omezení.
- 8.) Zpráva pro čtení statusu modulu.

Po dokončení cyklu se sekvence opakuje, dokud nepřijde požadavek na vypnutí modulu. V ten moment se sekvenční automat automaticky nastaví na krok 9, který slouží k odeslání CAN zprávy pro vypnutí daného modulu. Při požadavku na vypnutí se modul ihned vypne bez ohledu na to, v jakém kroku se stavový automat aktuálně nachází. Zprávy se odesílají cyklicky, jelikož zpráva pro zapnutí modulu musí být opakovaně odesílána, jinak dojde k vypnutí modulu přibližně po jedenácti vteřinách. Pro zajištění spolehlivého chodu CAN sběrnice a správného odesílání transparentních CAN zpráv do modulů je každý krok stavového automatu zpožděn o 400 milisekund. Toto zpoždění zabraňuje překrytí odesílaných zpráv v zásobníku odesílaných zpráv.

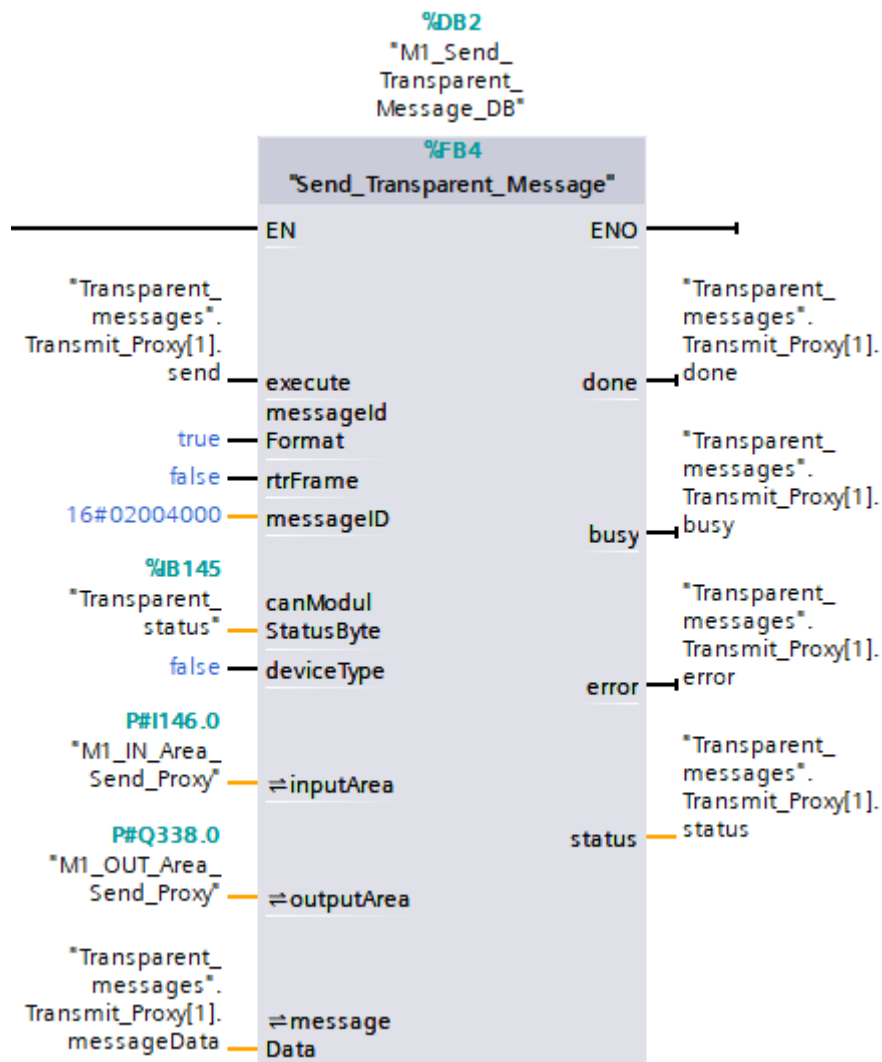


Obrázek 13 – Funkční blok pro zpracování odesílaných zpráv

2.5.2 FB pro odeslání transparentní CAN zprávy

Pomocí funkčního bloku "Send_Transparent_Message" je možné odeslat volně konfigurovatelnou zprávu CAN s kódem až 8 bajtů prostřednictvím komunikačního modulu CM CAN. Kód ve funkčním bloku je napsán v jazyce SCL. Tento funkční blok je součástí knihovny přiřazené komunikačnímu modulu CM CAN a jeho vstupy jsou propojeny s výstupy výše zmíněného funkčního bloku pro zpracování odesílaných zpráv. Pro správné odeslání transparentní CAN zprávy skrze komunikační modul CM CAN je nutné pro každý modul propojit tyto dva funkční bloky. Na základě výstupů z funkčního bloku pro zpracování odesílaných zpráv se vždy vytvoří unikátní transparentní CAN zpráva, která je připravena k odeslání pomocí funkčního bloku "Send_Transparent_Message". Jedná se především o vstupy, které charakterizují danou zprávu. Vstupy reprezentující typ zařízení, RTR frame, stav komunikačního modulu CM CAN, formát identifikačního pole zprávy CAN, vstupní a výstupní oblast proxy a ID zprávy jsou nastaveny permanentně, jelikož se jedná o shodné nastavení pro všechny odesílané zprávy. Formát identifikačního pole zprávy je nastaven na 29bitový rozšířený formát zprávy CAN, tedy na hodnotu "true". Typ zařízení je nastaven na hodnotu "false", čímž je indikováno, že se jedná o komunikační modul CM CAN pro distribuovanou periferní stanici ET 200SP a nikoliv o modul PN/CAN Link. Vzhledem k tomu, že se odesílají pouze datové zprávy, je vstup "rtrFrame" nastaven na hodnotu "false". Ke vstupu "messageID" je přiřazena odpovídající hodnota příslušného modulu (viz Tabulka 3). Na vstupy „inputArea“ a „outputArea“ jsou přiřazeny proměnné z tabulky tagů „CAN_Transparent“, které odpovídají adresám vstupů a výstupů nakonfigurovaných proxy pro odesílání. Poslední konstantní proměnnou je proměnná "Transparent_Status", která indikuje aktuální stav komunikačního modulu CM CAN. Proměnné "messageData" a "execute" představují variabilní vstupy funkčního bloku. Výstup "Q_Message_DATA" z funkčního bloku "Send_Message_Handling" charakterizuje datové pole transparentní CAN zprávy, které je rozděleno do pole 8 bajtů a reprezentuje požadovaný úkon zprávy pro zvolený modul. Vstup "execute" odešle příslušnou transparentní CAN zprávu s aktivní hranou do zásobníku odesílaných zpráv, který je uložen v interní paměti komunikačního modulu CM CAN a následně do vybraného napájecího modulu. Výstupy funkčního bloku "Send_Transparent_Message" indikují aktuální stav přenášených dat. Pokud jsou vstupní data funkčního bloku správná a sběrnice CAN není v přetížení, aktivuje se výstup "done", čímž se signalizuje úspěšné odeslání transparentní CAN zprávy do zásobníku odesílaných

zpráv. V případě chybně zadaných vstupních dat se aktivuje výstup "error" a v případě přetížení sběrnice CAN se aktivuje výstup "busy". Výstup "status" slouží k podrobnějšímu zobrazení aktuálního stavu funkčního bloku (viz příloha E). Datový typ výstupu "status" je "DWord" a jeho popis se zobrazuje na diagnostické obrazovce HMI (viz Obrázek 23).



Obrázek 14 – FB pro odesílání transparentní datové zprávy

Sepsaný kód uvnitř funkčního bloku „Send_Transparent_Message“ využívá stavový automat k řízení procesu přenosu zpráv CAN. Provádí validaci vstupů, tvorbu zprávy, povolení přenosu a základní detekci chyb. Pro lepší čitelnost a orientaci v programu je kód rozdělen do několika sekcí (regionů).

První z regionů stavového automatu je region TRIGGERING, který kontroluje potřebu spuštění funkčního bloku na základě několika podmínek. To zahrnuje aktivaci vstupu „execute“, který se stává pravdivým při vzestupné hraně signálu, a kontrolu předchozího volání, které musí být ukončeno (podmínka „statExecuteOld“ je nepravdivá) a zároveň nesmí probíhat žádné další volání (podmínka „statStatus“ = „STATUS_NO_CALL“). Pokud je funkční blok spuštěn poprvé, inicializují se interní proměnné a stavový automat je nastaven na „FB_STATE_VALIDATE_INPUTS“. Taktéž dochází k inicializaci struktury výstupní zprávy („statOutCanTransMessage“) a vymazání datových bytů. Nakonec je pokus o serializaci uživatelských dat zprávy („messageData“) do pole bytů. V případě chyby serializace nebo překročení délky dat je nastaven chybný stav. V programu je použita podmínka, která kontroluje, zda se nekoná žádné volání nebo dokončené provedení. Pokud je podmínka splněna, tak program předčasně ukončí činnost, aby se zabránilo zbytečnému zpracování.

Dalším regionem je stavový automat, který zajišťuje správný průběh procesu přenosu zprávy CAN a detekci případných chyb během tohoto procesu. Tato část kódu řídí průběh přenosu zprávy pomocí různých stavů, které jsou určeny hodnotou „statFBState“. Výchozím stavem je „FB_STATE_NO_PROCESSING“, který je aktivní, když neprobíhá žádné zpracování dat. Následuje stav „FB_STATE_VALIDATE_INPUTS“, který ověřuje uživatelské vstupy, jako je ID zprávy, délka dat a stav modulu CAN. Provádí se kontrola platnosti formátu ID zprávy a samotného ID, přičemž se také zjišťuje stav sběrnice CAN a modulu. V případě selhání jakékoli validace je nastaven chybový stav. Pokud všechny proběhlé kontroly proběhnou úspěšně, tak se následně přechází do nadcházejícího stavu „FB_STATE_COPY_DATA_AND_SEND“, který připravuje strukturu zprávy CAN („statOutCanTransMessage“) pro odeslání. Tento stav nastavuje formát ID zprávy, příznak RTR a počet datových bajtů k odeslání. Poté jsou kopírovány hodnoty ID zprávy a uživatelsky poskytnuté datové bajty do struktury zprávy a je povolen přenos. Následuje přechod do stavu „FB_STATE_CHECK_RESULT“, který provádí kontrolu chyb během přenosu a zjišťuje specifické chybové příznaky ze vstupní oblasti. Poslední stav je stav „FB_STATE_WAIT_FOR_ACK“. Tento stav čeká na potvrzení („acknowledgement“) od uzlu, který zprávu přijal. Je nutné vždy zprávu po přijetí a následném zpracování potvrdit, aby funkční blok mohl zpracovat a následně odeslat další zprávu.

Poslední region se jmenuje „OUTPUTS“, který aktualizuje výstupy funkčního bloku na základě aktuálního stavu a průběhu provedení. Nastavuje proměnné „done“, „busy“,

„error“ a „status“ podle hodnot vnitřních proměnných. Tyto výstupy jsou resetovány, pokud se vstup „execute“ stane nepravdivým a funkční blok je dokončen nebo se vyskytne chyba. Při resetování ostatních výstupů je také resetován výstup specifický pro aplikaci („statOutCanTransMessage“). Nakonec jsou přiřazeny konečné hodnoty výstupům funkčního bloku.

2.6 Nastavení filtrů pro příjem transparentních CAN zpráv

Pro nastavení filtrů přijímací proxy je použit funkční blok „Set_Transparent_Filtr“ z knihovny komunikačního modulu CM CAN. Nastavení filtrů hraje v této úloze velkou roli, protože napájecí moduly mezi sebou neustále komunikují pomocí datových zpráv s určitými protokoly, které se liší od protokolu používaného v této úloze pro odesílání a přijímání zpráv přes komunikační modul CM CAN. Zprávy, které si navzájem napájecí moduly zasílají, jsou pro danou úlohu nežádoucí. Jak bylo zmíněno, komunikační modul pracuje se zprávami o 29 bitech v identifikačním poli a o 8 bajtech v datovém poli. Pro eliminaci nežádoucích přijatých zpráv od napájecích modulů je tedy nutné správně nastavit parametry funkčního bloku. Vstup „proxyIndex“ odkazuje na index používané přijímací proxy. V tomto případě se jedná o hodnotu 1, jelikož je použita pouze jedna přijímací proxy s indexem 1. Do parametru „hwloOfTransparentModul“ se přiřadí hardwarový port, tedy komunikační modul CM CAN. Vstup „cntOfFilter“ indikuje počet použitých filtrů pro danou přijímací proxy. Jelikož jsou použity tři napájecí moduly a každý z modulů má unikátní adresu, výrobní číslo a datum výroby, hodnota tohoto parametru je nastavena na 3. Jednotlivé ID odchozích zpráv z napájecích modulů musejí být zapsány do proměnné „filter_data“. Tato proměnná obsahuje pole šestnácti datových typů „LCan_typeTransparentFiltr“ z knihovny komunikačního modulu CM CAN. Pole je šestnáctimístné z důvodu toho, že pro jednotlivou proxy lze nastavit až šestnáct různých filtrů ID zpráv. V souladu se zadáním práce byly využity pouze 3 pole, do kterých byly zapsány požadované ID a formát zpráv (viz Obrázek 15). Pokud se vstup „execute“ nastaví na hodnotu „true“, zvolené nastavení filtrů bude aktivní a komunikační modul CM CAN bude následně přijímat pouze přednastavené ID zpráv.

Po úspěšném zadání parametrů a aktivování vstupu „execute“ se aktivuje výstup „done“ na hodnotu „true“, čímž se potvrdí aktivace filtrů. V případě špatně zadaných vstupních parametrů se aktivuje výstup „error“. To může být způsobeno například hodnotou „proxyIndex“ nula, protože přijímací proxy jsou indexovány až od čísla 1. Výstup „busy“ je aktivní poté co na funkčním bloku probíhá zpracování určitých dat. Ostatní výstupy slouží pro diagnostiku funkčního bloku, která je přenesena do diagnostické HMI obrazovky. Jedná se o výstupy „status“ a „diagnostics“. Jednotlivé stavy těchto výstupů jsou popsány v přílohách (viz příloha G). Kód funkčního bloku „Set_Transparent_Filtr“ opět využívá stavového automatu, validaci vstupních parametrů a je rozdělen do regionů.

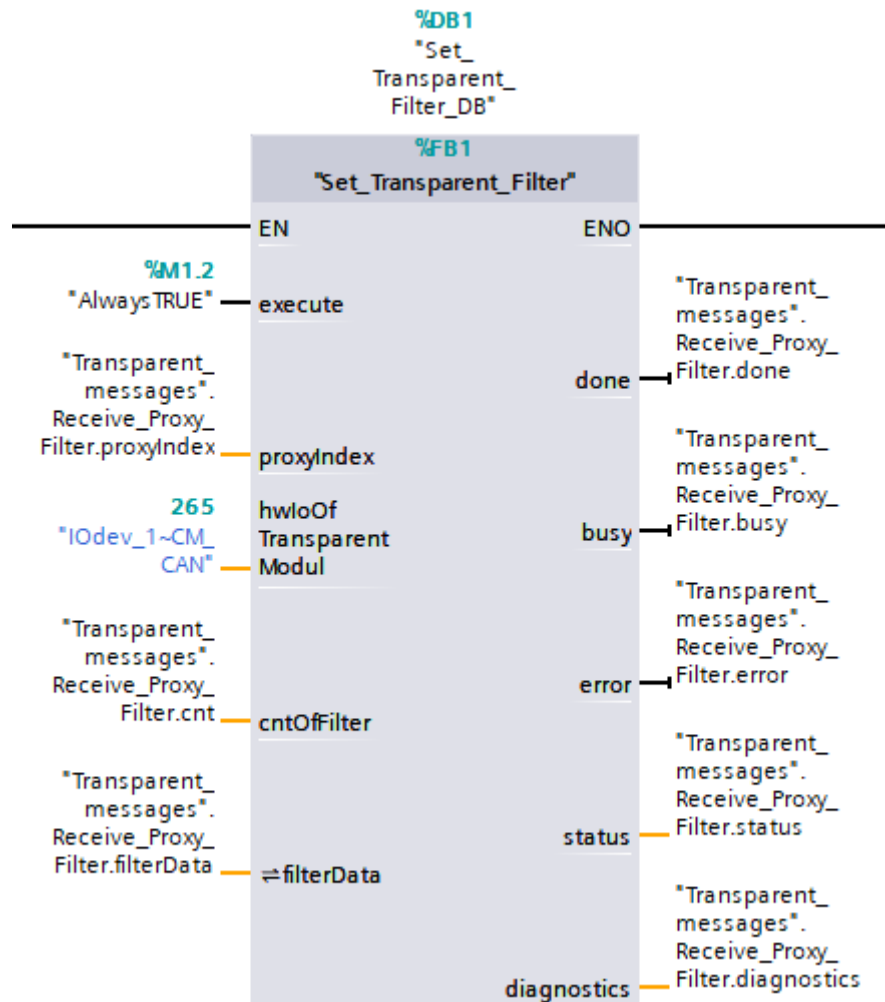
4	◀	▣	Receive_Proxy_Filter	Struct	
5	◀	▣	execute	Bool	false
6	◀	▣	cnt	USInt	3
7	◀	▣	proxyIndex	Byte	16#1
8	◀	▣	filterData	Array[1..16] of *LCa...	
9	◀	▣	filterData[1]	*LCan_typeTranspar...	
10	◀	▣	ignoreFilter	Bool	false
11	◀	▣	messageFilter	USInt	0
12	◀	▣	messageIdFormat	USInt	2
13	◀	▣	messageId	DWord	16#02005A5D
14	◀	▣	filterData[2]	*LCan_typeTranspar...	
15	◀	▣	filterData[3]	*LCan_typeTranspar...	

Obrázek 15 – Zapsaná data filtrů v DB „Transparent_messages“

První z regionů „TRIGGERING“ se zabývá inicializací, správou stavů a logikou zpracování bloku. Při inicializaci nastavuje počáteční hodnoty a stavy. Správa stavů řídí přechody mezi stavy na základě podmínek jako je vstup #execute a stav předchozího stavu. Region „STATE_MACHINE“ definuje různé stavy, přičemž každý z nich reprezentuje jeden krok zpracování. První stav #FB_STATE_NO_PROCESSING označuje nečinný stav, kdy žádné zpracování není aktivní, a toto pravidlo musí být vždy přítomno a zůstává prázdné. Stav #FB_STATE_VALIDATE_INPUT indikuje proces ověřování vstupů, jako je „indexProxy“ a „cntOfFilter“, aby se zajistilo, že jsou v platných rozsazích. Pokud ano, stavový automat přechází na kopírování dat, jinak nastavuje chybové stavy. Stav #FB_STATE_COPY_DATA označuje fázi, kdy se data filtru kopírují do vyrovnávací paměti pro další zpracování. Úspěšné kopírování přechází na spuštění operace zápisu, jinak nastavuje chybové stavy. Stav #FB_STATE_START_WRREC představuje začátek

operace zápisu, která spouští zápis filtrovaných dat, monitoruje dokončení této operace a přechází podle toho.

Poslední región „OUTPUTS“ zahrnuje správu stavu a chyb tím, že nastavuje statické výstupy. Tyto výstupy poskytují diagnostické informace v případě chyb, resetují výstupy podle potřeby a zapisují statické hodnoty.



Obrázek 16 – Funkční blok pro nastavení filtrů přijímaných CAN zpráv

2.7 Přijetí transparentních CAN zpráv

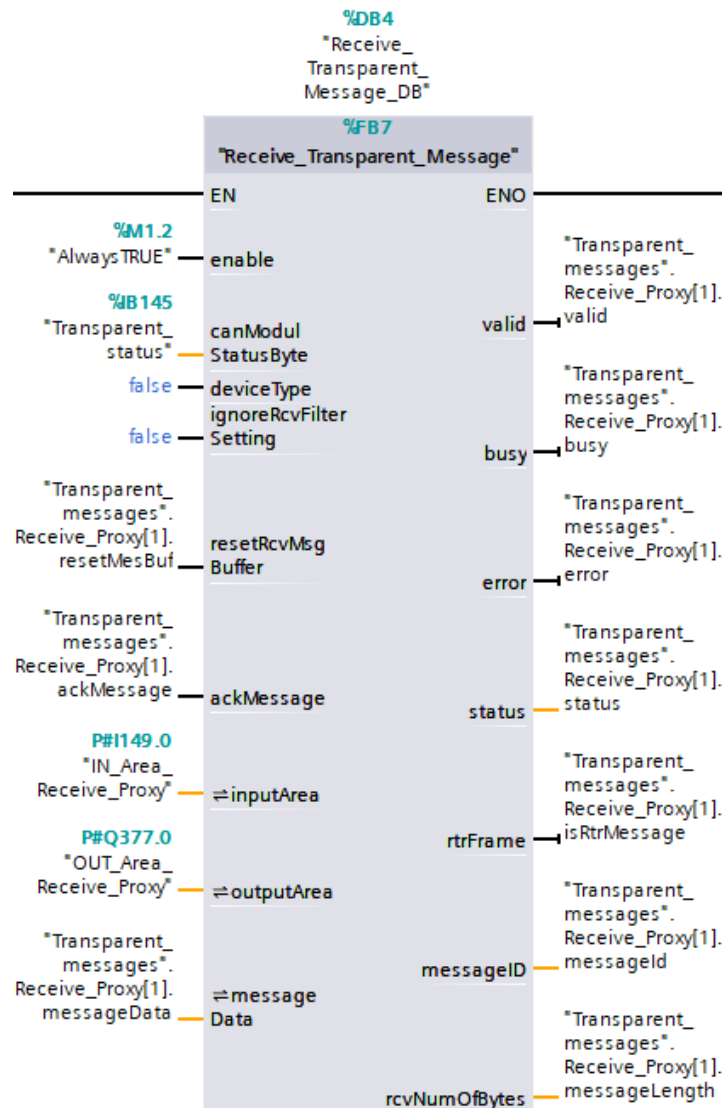
Obdobně jako s odesíláním transparentních CAN zpráv bylo nezbytné vytvořit funkční blok spolupracující s funkčním blokem pro příjem transparentních CAN zpráv. Na rozdíl od odesílání transparentních CAN zpráv není v tomto případě nutné pro každý modul vytvářet dvojici funkčních bloků. Po odeslání transparentní CAN zprávy do cílového modulu tento modul zašle potvrzení o přijetí zprávy do funkčního bloku pro příjem zpráv.

Toto potvrzení představuje odpověď modulu na požadavek odeslaný prostřednictvím funkčního bloku pro odesílání transparentních CAN zpráv. Z funkčního bloku pro příjem zpráv jsou poté pomocí funkčního bloku pro zpracování přijatých zpráv extrahována relevantní data, která jsou následně přenesena na HMI obrazovku. Selektace přijatých zpráv od modulů probíhá na základě nastaveného filtru pro ID zpráv a dále dle typu zprávy.

2.7.1 FB pro přijetí transparentních CAN zpráv

Jedná se o funkční blok "Receive_Transparent_Message", který je součástí knihovny komunikačního modulu CM CAN. Funkce programu zahrnuje několik klíčových úkolů. Za prvé, program umožňuje příjem transparentních zpráv CAN. Tyto zprávy obsahují datové pole a ID zprávy, které identifikuje odesílající a přijímající uzel. Po přijetí zprávy od napájecích modulů jsou zprávy filtrovány podle nastavení filtrů pro příjem transparentních CAN zpráv. Program z přijatých CAN zpráv extrahuje a zpracovává data zprávy a ukládá je do definované oblasti paměti, tzv. zásobníku pro přijaté zprávy. Z této interní paměti si funkční blok "Message_Receive_Handling" přebírá potřebná data pro finální zpracování transparentních CAN zpráv. Pro příjem zpráv je nutné nejprve aktivovat vstup "enable", který umožňuje funkčnímu bloku "Receive_Transparent_Message" přijímat transparentní zprávy CAN. Permanentně nastaveným vstupem funkčního bloku je vstup "canModulStatusByte", který slouží k předání informace o stavu komunikačního modulu CM CAN do funkčního bloku pro příjem transparentních CAN zpráv. Dále je nutné nastavit vstup "deviceType" na hodnotu "false", která indikuje komunikační modul CM CAN pro distribuovanou periferní stanici ET 200SP. Vstup "ignorRcvFilterSetting" se také nastaví na hodnotu "false", jelikož nastavený filtr pro příjem zpráv eliminuje nežádoucí zprávy od modulů, jako například komunikaci mezi nimi. Posledními permanentními vstupy jsou "inputArea" a "outputArea", kterým se přiřadí adresy vstupů a výstupů přijaté proxy z tabulky tagů "CAN_Transparent". Resetování interní paměti funkčního bloku, tedy zásobníku přijatých zpráv, se provede aktivací náběžné hrany na vstupu "resetRcvMsg". Po úspěšném přijetí transparentní CAN zprávy funkčním blokem "Receive_Transparent_Message" se aktivuje výstup "valid" a s ním se na výstupech a vstupech funkčního bloku zobrazí aktuální data přijaté zprávy. Jedná se o I/O "messageID", které reprezentuje ID přijaté zprávy, "rcvNumOfBytes", jež znázorňuje počet přijatých bajtů datového pole, a "rtrFrame", který indikuje, zda se jedná o datovou

nebo RTR zprávu. Při použití nastaveného filtru pro příjem zpráv bude počet bajtů datového pole zprávy vždy osm a RTR bude vždy hodnota "false". Na vstup "messageData" se nahraje bajtový rámec přijaté zprávy. Jak již bylo zmíněno výše, na základě ID zprávy a datového rámce zprávy se transparentní CAN zpráva dále zpracovává pro další využití. Po dokončení zpracování zprávy pomocí funkčního bloku "Message_Receive_Handling" je nutné potvrdit její přijetí. Potvrzení se provede aktivací náběžné hrany na vstupu "ackMessage" funkčního bloku "Receive_Transparent_Message". Proces příjmu zprávy se monitoruje na základě výstupů funkčního bloku. Jedná se o výstupy "valid", "busy", "error" a "status". Výstup "error" se aktivuje v případě nesprávných přijatých dat, "busy" v případě přetížení sběrnice CAN. Podrobný popis stavu funkčního bloku pro příjem transparentních CAN zpráv (viz příloha F) je k dispozici na diagnostické HMI obrazovce (viz Obrázek 22).



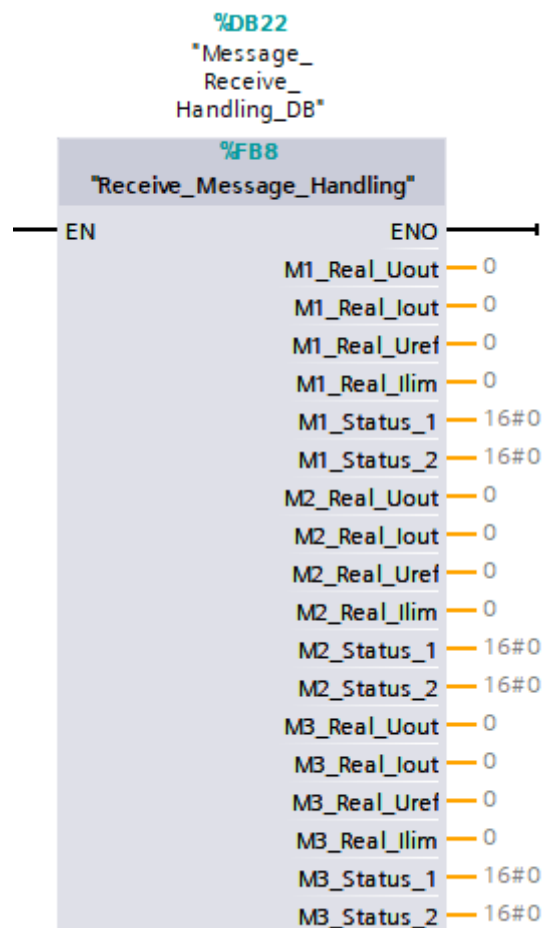
Obrázek 17 – Funkční blok pro příjem transparentních CAN zpráv

Kód funkčního bloku je napsán v programovacím jazyce SCL. Klíčovou částí programu je stavový automat, který se skládá z pěti navazujících stavů. V prvním stavu, nazvaném "RCV_STATE_NO_PROCESSING", program čeká na aktivaci vstupního signálu "enable" a neprovádí žádné zpracování. Ve stavu "RCV_STATE_ENABLING" program provede úvodní kontroly před zahájením příjmu dat na základě stavu komunikačního modulu CM CAN a nastaveného filtru pro příjem transparentních CAN zpráv. Ve stavu "RCV_STATE_BUFFER_RESET" program resetuje buffer přijatých zpráv, aby do něj byly ukládány pouze nejnovější zprávy. Ve stavu "RCV_STATE_MESSAGE_PROCESS" program zpracuje přijatou zprávu CAN, extrahuje ID, datový rámec zprávy a nastaví výstupní signály. Ve stavu "RCV_STATE_ACK_MESSAGE" program čeká na potvrzení přijaté transparentní CAN zprávy a po jeho obdržení se vrátí do stavu "RCV_STATE_NO_PROCESSING".

2.7.2 FB na zpracování přijatých CAN zpráv

Funkční blok "Message_Receive_Handling" slouží k závěrečnému zpracování přijatých transparentních CAN zpráv z napájecích modulů. Vnitřní program je napsán v jazyce SCL a jedná se o stavový automat. Po zavolání funkčního bloku se aktivuje vstup "enable" u funkčního bloku "Receive_Transparent_Message", čímž se zajistí, že funkční blok je schopen přijímat zprávy z interní paměti, tzv. zásobníku přijatých zpráv. Dále se aktivuje vstup "execute" funkčního bloku "Set_Transparent_Filtr" pro filtrování předem nastavených ID přijatých zpráv. Následuje stavový automat, který třídí přijaté zprávy nejprve podle ID přijaté transparentní CAN zprávy. Na základě tabulky adresace modulů (viz Tabulka 3) stavový automat rozpozná, od kterého modulu přijatá zpráva pochází. Následné třídění zpráv se provádí na základě bajtového rámce transparentní CAN zprávy. Podle hodnoty nultého bajtu stavový automat určí, zda se jedná o odpověď modulu na odeslanou zprávu pro čtení nebo nastavení. Pokud se hodnota nultého bajtu datového rámce rovná 16x11, jedná se o odpověď modulu na odeslanou zprávu pro nastavení parametru. Hodnota 16x13 reprezentuje odpověď modulu na požadavek čtení parametrů. Následující bajt datového rámce určuje, o jaký parametr se jedná. Může se jednat o zápis nebo čtení proudového omezení nebo referenčního napětí, aktuálního výstupního napětí, proudu nebo čtení aktuálního stavu napájecího modulu. Posledním krokem stavového automatu je rozlišení, zda se jedná o odpověď na zapnutí nebo vypnutí daného modulu.

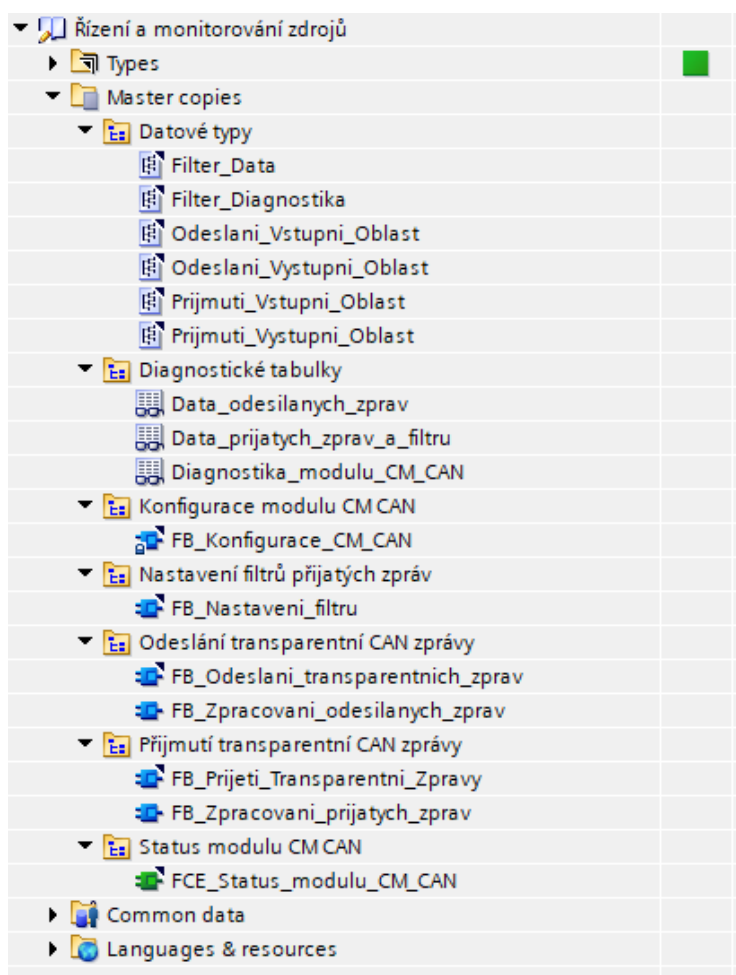
Shodně s funkčním blokem "Message_Send_Handling" se hodnoty parametrů ukládají v datovém typu "DWord". Ten je následně nutné převést na datový typ "DInt" a vydělit tisícem, jelikož hodnoty zapisované do posledních čtyř bajtů datového rámce jsou uváděny v mili jednotkách voltů a ampér. Po rozpoznání typu transparentní CAN zprávy stavový automat odešle náběžnou hranu na vstup "ackMessage" funkčního bloku „Receive_Transparent_Message“, čímž potvrdí přijetí a zpracování zprávy. Po potvrzení je možné přijmout a zpracovat další zprávu. Výstupy funkčního bloku "Message_Receive_Handling" představují aktuální hodnoty jednotlivých parametrů v jednotkách voltů a ampér a aktuální stav jednotlivých napájecích modulů. Stav je rozdělen do dvou datových typů "Word", jelikož každý bit z obou bajtů reprezentuje specifickou informaci o modulu. Tyto jednotlivé bity se poté transformují do chybových hlášení na HMI obrazovkách, kde se zobrazuje odpovídající popis stavu. Podrobnosti ohledně všech chybových hlášení je sepsán v přílohách (viz příloha H).



Obrázek 18 – FB pro zpracování přijatých transparentních CAN zpráv

2.8 TIA knihovna pro řízení a monitorování napájecích zdrojů

V souladu se zadáním práce byla vytvořena univerzální knihovna „Řízení a monitorování zdrojů“, která obsahuje komplexní sadu funkčních bloků, funkcí, datových bloků, datových typů a diagnostických tabulek pro usnadnění budoucího řízení a monitorování napájecích modulů UR50060-E od společnosti UUGreenPower. Jedná se jak o převzaté prvky z knihovny komunikačního modulu CM CAN, které jsou upraveny pro zadanou úlohu, tak i nově přidané unikátní prvky, které byly vytvořeny pro tuto práci. Knihovna poskytuje rychlý přístup k jednotlivým prvkům programu potřebných pro správné řízení a diagnostiku napájecích zdrojů. Díky vytvořené knihovně se zjednodušuje práce s těmito prvky a minimalizuje se riziko chyb při jejich implementaci do programu. Knihovna obsahuje také diagnostické tabulky, které umožňují rychlou identifikaci a řešení problémů s napájecími zdroji. To výrazně snižuje dobu potřebnou k odhalení a opravě poruch v systému. Knihovna byla vložena jako elektronická příloha do portálu STAG spolu s dalšími přílohami a diplomovou prací.



Obrázek 19 – Uživatelská knihovna v prostředí TIA Portal

3 Návrh autokonfigurace s využitím více zdrojů

Řešení diplomové práce je navrženo pro tři napájecí moduly UUGreenPower, přičemž každý modul je ovládán individuálně. Hlavní nevýhodou stávajícího řešení je duplikace dvojic funkčních bloků pro odesílání transparentních CAN zpráv a rozdělení funkčních bloků pro zpracování odeslaných a přijatých transparentních CAN zpráv. Cílem návrhu autokonfigurace je sloučit tyto dva funkční bloky do jednoho, čímž se zpracování transparentních CAN zpráv bude odehrávat v rámci jediného bloku. Výsledkem bude snížení zatížení sběrnice CAN, jelikož odeslané zprávy počkají na doručení odpovědi a teprve poté bude odeslána další zpráva. Celý proces bude monitorován pomocí HMI a zobrazování chybových hlášení.

3.1 Konfigurace komunikačního modulu

Jelikož přidání „Transmit proxies“ a „Receive proxies“ se musí zadat manuálně v příslušném HSP, je nutné, aby uživatel nakonfiguroval jednu „Transmit“ a jednu „Receive“ proxy. Následně musí uživatel znát ID přijímaných zpráv od modulů pro následnou konfiguraci filtrů „Receive proxies“. Po správném nakonfigurování a zkompilování je nutné vytvořit proměnné pro vstupní a výstupní oblasti. Dále je nutné vytvořit proměnné v globálním datovém bloku pro odesílání, přijímání a nastavení filtrů transparentních CAN zpráv. S touto konfigurací je poté možné pokračovat v tvorbě funkčního bloku a logiky programu.

3.2 Logika programu

Funkční blok by zahrnoval část pro zpracování odeslaných a přijatých transparentních CAN zpráv. Na základě požadavku uživatele zadaného na HMI obrazovkách by funkční blok identifikoval, o jaký modul nebo skupinu modulů se jedná, a zahájil by komunikaci. V závislosti na výběru modulu na HMI by se spustil příslušný stavový automat s odpovídajícím ID zprávy, který by odesílal předem nakonfigurované zprávy. Úvodní zpráva by byla odeslána do funkčního bloku pro odesílání transparentních CAN zpráv z knihovny komunikačního modulu CM CAN. Následně by byla z modulu pomocí funkčního bloku pro příjem transparentních zpráv z knihovny komunikačního modulu CM CAN přijata odpověď na odeslanou zprávu a následně by byla zpráva zpracována stejným funkčním blokem jako pro zpracování odeslaných zpráv. Po ověření, že na přijímací

funkční blok dorazila odpovídající odpověď, by byla odeslána další zpráva. Výsledkem by byla odlehčená sběrnice CAN bez rizika překrytí dat.

3.3 HMI a chybová hlášení

Je potřeba aby program neustále monitoroval stav procesu přenášení dat a vizualizoval patřičné chybové hlášení na obrazovky HMI. V programu by se kontrolovali nejen výstupní statusy funkčních bloků pro příjem, odeslání a nastavení filtrů transparentních CAN zpráv, ale také interní komunikace funkčního bloku pro zpracování zpráv. Pro odeslání a následný příjem transparentní CAN zprávy by byl vytvořen časovač s určitým časovým zpožděním pro zpracování odeslané a přijaté zprávy. Pokud by zpráva byla do časového intervalu přijata, tak by se na HMI obrazovce signalizovalo úspěšné zpracování zprávy a potvrzení správného přiřazení ID zprávy. V opačném případě by se zobrazilo chybové hlášení o nedokončeném zpracování zprávy nebo špatném přiřazení ID zprávy a stavový automat by na základě této informace odeslal požadavek na znovu odeslání patřičné zprávy.

3.4 Výhody autokonfigurace

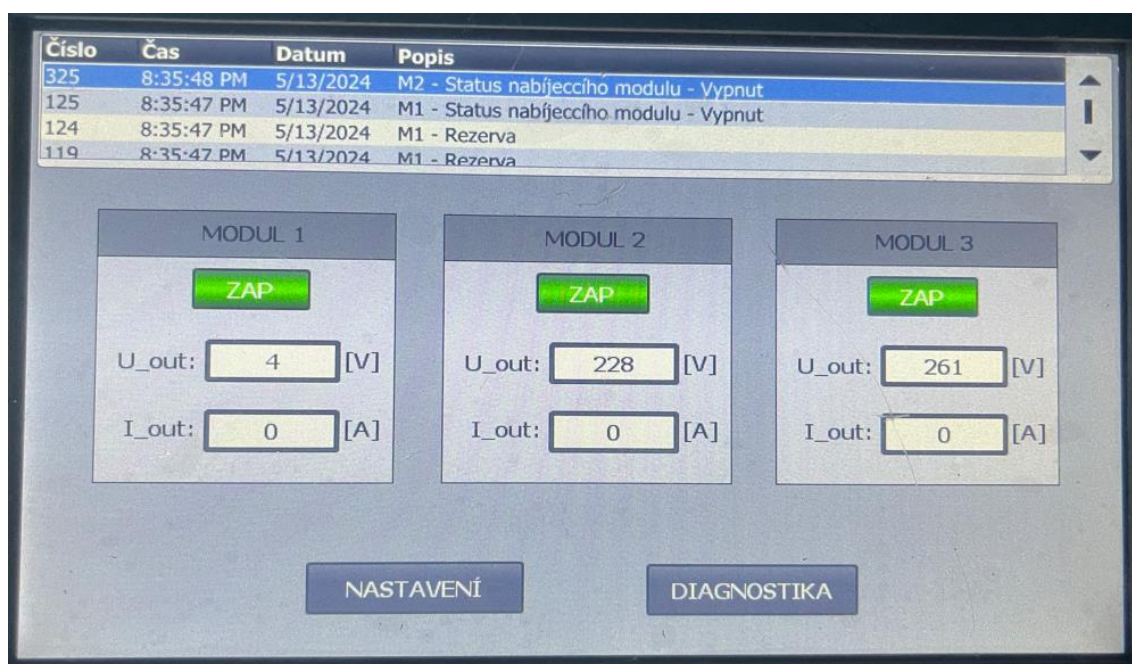
Navržený systém autokonfigurace nabízí několik výhod. První z nich je snížení opakování kódu díky eliminaci duplicitních funkčních bloků pro odesílání a příjem transparentních CAN zpráv, což zjednodušuje údržbu a snižuje objem kódu. Efektivita sběrnice CAN je zlepšena tím, že se zprávy odesílají pouze po přijetí a zpracování předchozí odpovědi, což snižuje zahlcení sběrnice a optimalizuje využití síťových zdrojů. Implementovaný mechanismus znovu odeslání a detekce chyb zajišťuje spolehlivou komunikaci i v případě dočasných problémů s připojením nebo chyb modulu. Integrace s HMI umožňuje uživatelům snadno konfigurovat parametry, sledovat stav komunikace a diagnostikovat případné problémy. Nakonec, modulární design a centralizovaná logika zjednodušují testování a validaci celého systému, což snižuje riziko chyb a zvyšuje celkovou spolehlivost.

4 Vizualizace

Důležitou součástí monitorování a ovládání napájecích modulů je možnost identifikace a zobrazení informací o modulech skrze rozhraní HMI. Pro usnadnění ovládání a monitorování napájecích zdrojů byly vytvořeny obrazovky HMI na operačním panelu KTP700 Basic PN ve verzi 15.01.0.0. Jelikož operační panely Basic nepodporují funkci „faceplate“ nebylo možné vytvořit šablony jednotlivých bloků pro zobrazení a nastavení hodnot parametrů. Jednotlivé části bloků byly sdruženy do skupiny bloků a textů, které je snadné následně kopírovat. Tyto obrazovky HMI nabízí funkci sledování aktuálních hodnot parametrů jednotlivých modulů a jejich nastavení, dále pak detailní diagnostiku napájecích modulů pro komplexní přehled o zařízení i procesech jednotlivých funkčních bloků a stavu procesu přenášených dat. Obrazovky obsahují také aktuální zobrazení chybových hlášení, které se vždy nachází v horní části každé obrazovky HMI. Výsledkem je uživatelsky přívětivé prostředí pro snadnou a efektivní manipulaci s napájecími moduly

4.1 Hlavní obrazovka

Tato obrazovka slouží pro komplexní přehled aktuálních parametrů jednotlivých napájecích modulů. Uživatel může pomocí tlačítek pro vypnutí a zapnutí modulu ovládat daný napájecí modul. Pokud je modul zapnutý, tlačítko se podbarví zelenou barvou, v opačném případě červenou barvou.

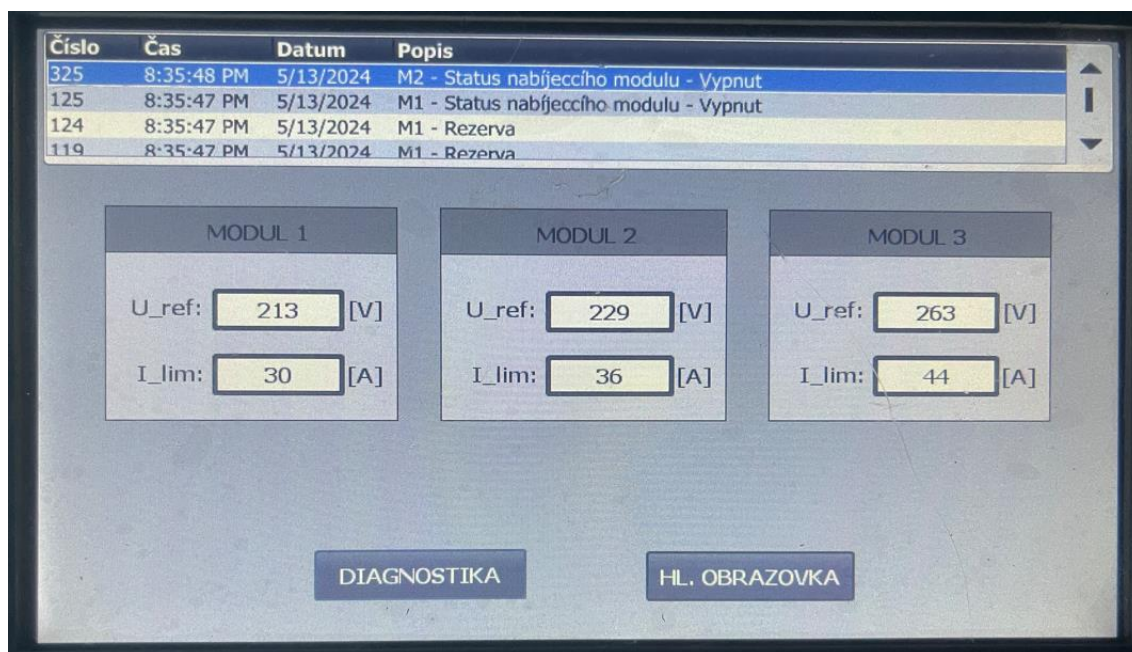


Obrázek 20 – Hlavní obrazovka operačního panelu

Jednotlivé parametry slouží pouze pro zobrazení aktuálních hodnot napětí a proudu a není možné do nich zapisovat. Obrazovka dále obsahuje tlačítka pro navigaci na další obrazovky, které slouží pro diagnostiku napájecích modulů nebo nastavení jednotlivých parametrů. V horní části informační obrazovky se nachází pole aktuálních chybových hlášení, ve kterém je možné pomocí šipek procházet nahoru a dolů.

4.2 Obrazovka pro nastavení parametrů

Po stisknutí tlačítka "NASTAVENÍ" na hlavní obrazovce se uživatel přesměruje na obrazovku pro nastavení jednotlivých parametrů napájecích modulů. Po kliknutí do pole pro zadání požadované hodnoty parametru se zobrazí virtuální klávesnice, do které uživatel zadá požadovanou hodnotu v jednotkách ampér nebo voltů. Z důvodu bezpečnosti jsou k jednotlivým polím přiřazeny limity zadávaných hodnot. Tyto limity jsou zobrazeny na virtuální klávesnici u pole pro zadání požadované hodnoty. Pokud by uživatel zadal hodnotu mimo stanovený limit, po stisknutí klávesy "Enter" se v horní části obrazovky vyhrazené pro aktuální chybová hlášení zobrazí chybová hláška a hodnota se nepropíše do PLC. V případě, že autor PLC kódu nastaví například limit referenčního výstupního napětí na 20–150 V a po spuštění modulu je na modulu přednastaveno 200 V, aktuální hodnota příslušného parametru se zbarví oranžovou barvou, která signalizuje překročení přednastavených limitů.

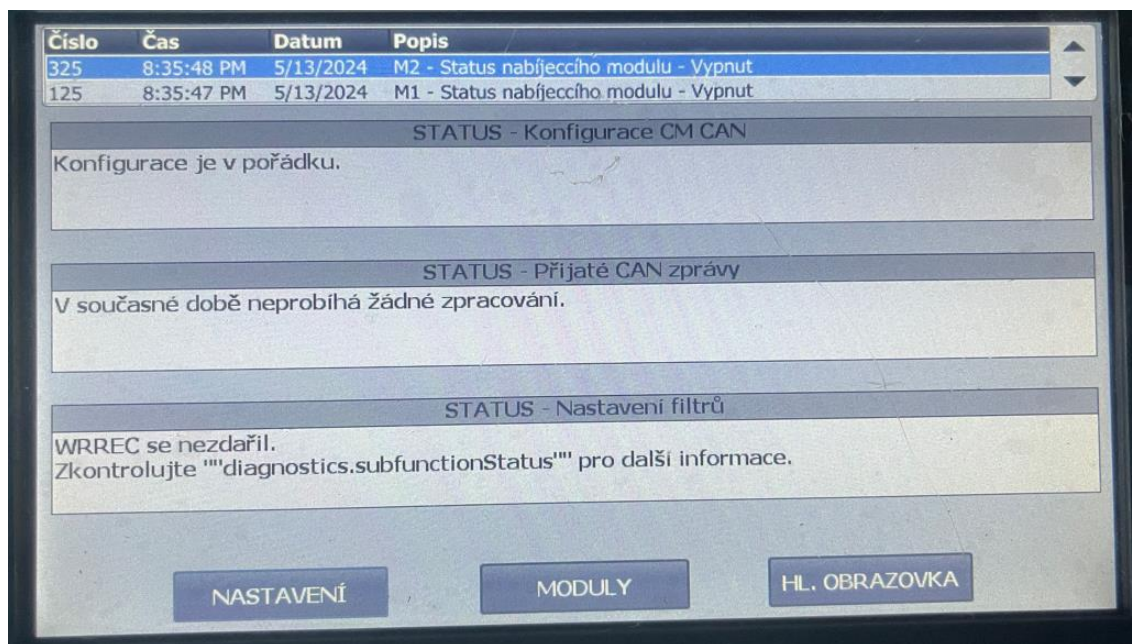


Obrázek 21 – Obrazovka pro nastavení parametrů

V případě zadání správné hodnoty a následného stisknutí klávesy "Enter" se požadovaná hodnota přenesla do PLC programu. Na obrazovce se nacházejí i další tlačítka, která uživatele přeměrují na odpovídající HMI obrazovky.

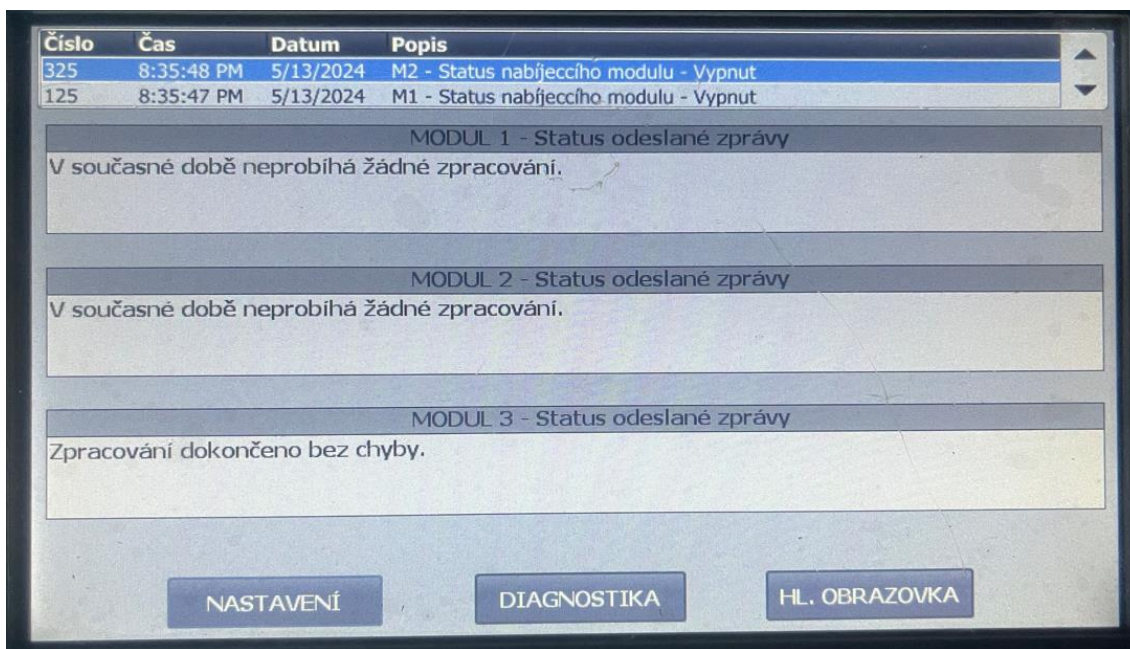
4.3 Obrazovky diagnostiky

Po stisknutí tlačítka "DIAGNOSTIKA" ve spodní části HMI obrazovky se uživatel přenesl na obrazovku pro zobrazení stavů jednotlivých funkčních bloků. Každý funkční blok disponuje výstupem pro zobrazení stavu procesu dat. Tento výstup má datový typ "DWord" a jeho hodnota je uložena v hexadecimálním formátu. Pro snazší pochopení stavu funkčních bloků byla tato hodnota převedena do decimálního formátu. Pomocí textových listů jsou k této hodnotě přiřazeny odpovídající popisy stavů jednotlivých funkčních bloků. Text uvnitř polí je dynamický a při každé změně výstupního stavu se v poli zobrazí aktuální popis. Z důvodu omezeného prostoru bylo nutné vytvořit navazující obrazovku "MODULY" pro zobrazení zbylých funkčních bloků. Po stisknutí tlačítka "DIAGNOSTIKA" se zobrazí pole pro výstupní stavy funkčních bloků pro konfiguraci komunikačního modulu CM CAN, nastavení filtrů přijímaných zpráv a výstupní stav funkčního bloku pro příjem transparentních CAN zpráv.



Obrázek 22 – Obrazovka pro diagnostiku funkčních bloků

Tlačítko "MODULY" uživatele přesměruje na zbývající výstupní stavy funkčních bloků. Jedná se o stavy funkčních bloků pro odesílání transparentních CAN zpráv. Toto tlačítko je viditelné pouze na diagnostických obrazovkách operačního panelu a není možné se na tuto obrazovku dostat z informativní hlavní HMI obrazovky ani z obrazovky pro nastavení parametrů napájecích modulů.



Obrázek 23 – Doplnující obrazovka pro diagnostiku funkčních bloků

V horní části obrazovek se opět zobrazují aktuální chybová hlášení a ve spodní části řada tlačítek pro přepínání mezi jednotlivými obrazovkami.

Závěr

Diplomová práce se zaměřila na sestavení funkčního zařízení pro ovládání a monitorování napájecích modulů UUGreenPower, které komunikují po sběrnici CAN s PLC pomocí komunikačního modulu od společnosti Siemens. Zařízení je navrženo pro ovládání a monitorování až tří napájecích modulů s využitím komunikačního modulu CM CAN pro distribuovanou periferní stanici ET 200SP. Práce se zabývá detailním popisem kroků konfigurace komunikačního modulu a práce s programovatelnými datovými zprávami, které slouží ke komunikaci mezi napájecími moduly a PLC. Výsledkem je funkční program umožňující efektivní a rychlou komunikaci s napájecími moduly a zajišťující uživatelsky přívětivé prostředí pro jejich ovládání a monitorování pomocí operačního panelu KTP700 Basic.

Čtenář byl seznámen s problematikou komunikace napájecích modulů po sběrnici CAN a byl obeznámen s tvorbou datových zpráv. Součástí práce bylo sestavení detailního návodu pro správnou konfiguraci komunikačního modulu CM CAN v provozním režimu Transparent. Práce se zaměřila na možnost s využitím přijímacích a odesílaných proxy transparentních zpráv, které disponují jednoduchostí ze strany komunikace s napájecími zdroji. Za použití programovatelných datových zpráv bylo dosaženo požadovaných výsledků dle zadaných bodů zadání.

V programovací části se podařilo vytvořit a implementovat kód s využitím funkcí a funkčních bloků. Některé části kódu byly navrženy s ohledem na budoucí rozšíření o další napájecí moduly, avšak hlavní logika programu je určena pouze pro komunikaci se třemi napájecími zdroji. Příkladem je funkční blok pro zpracování transparentních CAN zpráv, který je přidělen každému modulu zvlášť. Hlavní nevýhodou tohoto přístupu je vyšší nárok na paměť PLC a zatížení sběrnice CAN. To vede k opožděnému odesílání a čtení transparentních CAN zpráv, což se projevuje v časovém zpoždění při zápisu a čtení parametrů na HMI obrazovkách. Pro dosažení lepších výsledků a rychlejší komunikace po sběrnici CAN mezi PLC a napájecími zdroji byl navržen kompatibilnější systém autokonfigurace s využitím více zdrojů. Myšlenkou návrhu je sjednocení zpracování odesílaných a přijímaných transparentních CAN zpráv do jednoho uceleného funkčního bloku. Výsledkem by byla plynulejší komunikace po méně zatížené sběrnici CAN. I když návrh nebyl v rámci diplomové práce implementován, byly stanoveny kroky pro usnad-

nění rozšíření zařízení o další napájecí moduly. Jedním z nich je vytvoření knihovny komplexních funkčních bloků a funkcí, které uživateli usnadní přidávání opakovaně používaných funkčních bloků. Dále byla struktura proměnných navržena tak, aby se minimalizovala nutnost uživatelských zásahů do programovací části. Ve vizualizaci byly vytvořeny skupiny zapisovacích polí, bloků a textů, které umožňují snadné zkopírování v případě přidání dalších napájecích modulů.

Diplomová práce přispěla k hlubšímu pochopení komunikace po sběrnici CAN a programování PLC od společnosti Siemens. Byly vytvořeny návody a příklady kódu, které mohou být pro další vývojáře užitečné. Celkově lze konstatovat, že diplomová práce představuje cenný přínos v oblasti ovládání a monitorování napájecích modulů UUGreenPower. Implementované řešení je funkční, flexibilní a snadno rozšiřitelné, splňující požadavky moderních průmyslových řídicích systémů. Pro budoucí práci se doporučuje implementovat kompatibilnější systém autokonfigurace s využitím více zdrojů pro optimalizaci komunikace po sběrnici CAN, rozšíření systému o další napájecí moduly a testování jeho funkčnosti a stability.

Použitá literatura

- [1] SIEMENS. CM CANOpen User Manual. Siemens, 2023. Dostupné z: <http://www.er-soft.com/descargas/documentacion/14845735537646.pdf>. [cit. 2024-02-14].
- [2] SIEMENS. SIMATIC ET 200S - Modul 1SI CANopen - Uživatelská příručka. Siemens, 2023. Dostupné z: <https://support.industry.siemens.com/cs/document/49717118/canopen-module-from-hms-available-for-et-200s?dti=0&lc=en-WW>. [cit. 2024-02-14].
- [3] SIEMENS. SIMATIC PN/CAN Link - Uživatelská příručka. Siemens, 2023. Dostupné z: https://support.industry.siemens.com/cs/attachments/109751077/109751077_network_transitions_pncan_link_manager_slave_en.pdf. [cit. 2024-02-14].
- [4] SIEMENS. SIMATIC ET 200SP - Modul CM CAN - Uživatelská příručka. Siemens, 2023. Dostupné z: https://support.industry.siemens.com/cs/attachments/109778981/et200sp_cm_can_manual-en_US_en-US.pdf. [cit. 2024-02-14].
- [5] Rectifier Module User Manual: UR50060. PDF. UUGreenPower, 2018. [cit. 2024-01-06].
- [6] SIEMENS. Configuring the SIMATIC ET 200SP CM CAN communication module for CAN transparent mode. Siemens, 2023. Dostupné z: https://support.industry.siemens.com/cs/attachments/109779316/109779316_Config_CM_CAN_transparent_DOC_V2_en.pdf. [cit. 2024-02-18].
- [7] Rectifier module and monitoring unit CAN communication reference guide. PDF. Document version: 06. Shenzhen UUGreenPower Electrical Co., 2018. [cit. 2024-01-19].
- [8] CAN IN AUTOMATION. Online. CAN in Automation (CiA). 2024. Dostupné z: <https://www.can-cia.org/can-knowledge/canopen-fd-the-art-of-embedded-networking>. [cit. 2024-05-14].
- [9] SIEMENS AG. Siemens SiePortal. Online. Dostupné z: <https://sieportal.siemens.com/cs-cz/home>. [cit. 2024-01-29].
- [10] CAN IN AUTOMATION. Online. CAN in Automation (CiA). 2024. Dostupné z: <https://www.can-cia.org/can-knowledge>. [cit. 2024-03-14].

- [11] SIEMENS. Configuration of the SIMATIC ET 200SP CM CAN communication module for CANopen Manager and CANopen Slave. Siemens, 2023. Dostupné z: https://support.industry.siemens.com/cs/attachments/109779316/109779316_Config_CM_CANopen_Manager_Slave_DOC_V2_en.pdf. [cit. 2024-03-12]
- [12] ISO. ISO 11898-2:2024, Road vehicles — Controller area network (CAN). Edition 3. 2024. [cit. 2024-04-15].
- [13] CANLAB S.R.O. CAN - Controller Area Network. Online. CAN - Controller Area Network. 2014. Dostupné z: https://www.canlab.cz/cs/can_bus. [cit. 2024-04-02].
- [14] BOSCH. CAN Protocols. Online. CAN Protocols. 2024. Dostupné z: <https://www.bosch-semiconductors.com/ip-modules/can-protocols/>. [cit. 2024-04-05].
- [15] WIKIPEDIA. CAN bus. Online. CAN bus. 2005. Dostupné z: https://cs.wikipedia.org/wiki/CAN_bus. [cit. 2024-04-07].
- [16] VÝVOJ.HW.CZ. Aplikování sběrnice CAN. Online. Aplikování sběrnice CAN. 2004. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/rozhrani/aplikovani-sber-nice-can.html>. [cit. 2024-04-09].
- [17] ČSN. EN 61000-1-2, Elektromagnetická kompatibilita (EMC). Ed. 2. 2007. [cit. 2024-05-02].

Přílohy

A Seznam HW komponentů

Počet [ks]	HW	Katalogové číslo	Verze FW	Popis
1	SIMATIC S7-1500, CPU 1511-1 PN	6ES7511-1AK02-0AB0	02.09.04	PLC
1	SIMATIC ET 200SP, IM 155-6 PN ST	6ES7155-6AU01-0BN0	04.02.00	Kompaktní periferní stanice
1	SIMATIC ET 200SP, BA 2XRJ45	6ES7193-6AR00-0AA0	x	Bus adaptér
1	SIMATIC ET 200SP, CM CAN	6ES7137-6EA00-0BA0	01.01.00	Komunikační modul CAN
1	SIMATIC ET 200SP, DQ 16X 24V DC/0.5A ST	6ES7132-6BH01-0BA0	00.00.00	Digitální výstupy
1	SIMATIC ET 200SP, DI 16X 24V DC ST	6ES7131-6BH01-0BA0	00.00.00	Digitální vstupy
2	SIMATIC ET 200SP TM ECC PL ST	6FE1242-6TM20-0BB1	01.00.05	Technologický modul pro řízení a monitorování nabíjení elektrických vozidel
1	KTP700 Basic PN	6AV2123-2GB03-0AX0	15.01.0.0	Operační panel

B Kombinace adres modulů

Adresa modulu	Aktivní poloha
1	J1 [1] + J2 [0]
2	J1 [2] + J2 [0]
3	J1 [3] + J2 [0]
4	J1 [4] + J2 [0]
5	J1 [0] + J2 [1]
6	J1 [1] + J2 [1]
7	J1 [2] + J2 [1]
8	J1 [3] + J2 [1]
9	J1 [4] + J2 [1]
10	J1 [0] + J2 [2]
11	J1 [1] + J2 [2]
12	J1 [2] + J2 [2]
13	J1 [3] + J2 [2]
14	J1 [4] + J2 [2]
15	J1 [3] + J2 [3]
16	J1 [1] + J2 [3]
17	J1 [2] + J2 [3]
18	J1 [3] + J2 [3]
19	J1 [4] + J2 [3]
20	J1 [4] + J2 [4]
21	J1 [1] + J2 [4]
22	J1 [2] + J2 [4]
23	J1 [3] + J2 [4]
24	J1 [4] + J2 [4]

C Význam statusu CM CAN

Proměnná	Hodnota	Popis
"CanStatus".m1BusStatus	0	Modul je vypnutý
	1	Sběrnice vypnuta nebo není synchronizovaná
	2	Chyba pasivní
	3	Vše v pořádku
"CanStatus".m1ModulStatus	0-4	Rezerva
	5	Modul není nakonfigurován
	6	Špatná parametrizace
	7	Rezerva
"CanStatus".m1projectMessagesUpdated	True	Aktualizováno od posledního obnovení
	False	Zastaralé

D Význam statusu FB konfigurace CM CAN

Kód statusu hexa	Kód statusu deka	Popis
0000 8001	32769	Instrukce již běží.
0000 8002	32770	Blok v parametru DB_NO není globální datový blok. Zadejte globální datový blok se symbolickým názvem nebo "%DBxy".
0000 8003	32771	Nastavte číslo datového bloku v parametru "DB_NO".
0000 8004	32772	Datový blok je příliš krátký.
0000 8005	32773	V datovém bloku je příliš málo datových záznamů. Musí být přítomen alespoň jeden datový záznam.
0000 8006	32774	Datový záznam je příliš velký. Maximální velikost datového záznamu je 4096 bajtů.
DF80 B500	3749754112	Rozšířená konfigurace byla ignorována, protože modul nebyl ve stavu "není nakonfigurován".
DF80 B600	3749754368	Datová věta nebyla přijata.
DF80 B800	3749754880	Byla přijata nesprávná data pro přiřazení parametrů

E Význam statusu FB pro odesílání CAN zpráv

Kód statusu hexa	Kód statusu deka	Popis
0000	0	Zpracování dokončeno bez chyby.
7000	28672	V současné době neprobíhá žádné zpracování.
7001	28673	První volání po příchozím příkazu (vzestupná hrana na "execute").
7002	28674	Následné volání v průběhu zpracování bez dalších podrobností.
8000	32768	Zpráva nebylo možné přenést, protože je neplatný datový typ. - byl použit ve variantě InOut "messageData", nebo - bylo použito více než 8 bajtů.
8001	32769	Zpráva nemohla být přenesena, protože ID zprávy použité na vstupu "messageId" je příliš dlouhé. Maximální hodnota je 16#7FF pro 11bitovou zprávu a 16#1FFF_FFFF pro 29bitovou zprávu.
8002	32770	Zpráva nemohla být odeslána, protože - na vstupu "canModulStatusByte" nebyl připojen žádný stavový bajt modulu, nebo - modul je v chybovém stavu.
8100	33024	Zprávu nebylo možné odeslat, protože vysílací vyrovnávací paměť modulu je plná. Snižte frekvenci přenosu zprávy
8101	33025	Poslední zpráva nemohla být přenesena, protože data přenášená blokem byla neplatná.
8600	34304	Chyba v důsledku nedefinovaného stavu ve stavovém stroji.

F Význam statusu FB pro příjem CAN zpráv

Kód statusu hexa	Kód statusu deka	Popis
7000	28672	V současné době neprobíhá žádné zpracování.
7001	28673	Probíhá zpracování
7002	28674	Čeká se na zpětnou vazbu o obnovení vyrovnávací paměti
8000	32768	Data ve variantě InOut "messageData" jsou neplatná. - Ve variantě InOut "messageData" je použit neplatný datový typ, nebo - přijatá zpráva obsahuje větší užitečné zatížení, než se vejde do InOut "messageData". Potvrzeno vzestupnou hranou na vstupu "ackMessage".
8001	32769	Na modulu není nastaven žádný platný filtr. - Nastavte na modulu platný filtr nebo - Nastavte vstupní bit "ignoreRcvFilterSetting" a restartujte blok s vzestupnou hranou na vstupu "enable".
8002	32770	V současné době nelze přijmout žádnou zprávu, protože - na vstupu "canModulStatusByte" nebyl propojen žádný stavový bajt modulu, nebo - modul je v chybovém stavu. Automaticky se resetuje, jakmile modul již nesignalizuje chybový stav.
8600	34304	Chyba v důsledku nedefinovaného stavu ve stavovém stroji. Restartujte blok s vzestupnou hranou na vstupu "enable".

G Význam statusu FB pro nastavení filtrů

Kód statusu hexa	Kód statusu deka	Popis
0000	0	Zpracování dokončeno bez chyby.
7000	28672	V současné době neprobíhá žádné zpracování.
7001	28673	První volání po příchozím příkazu (vzestupná hrana pro "execute").
7002	28674	Následné volání v průběhu zpracování bez dalších podrobností.
8000	32768	Index proxy je roven 0 nebo větší než 64 (v šestnáctkové soustavě).
8001	32769	Chyba, protože "cntOfFilter" = 0 nebo > 16 (desetinné číslo).
8002	32770	WRREC se nezdařil. Zkontrolujte "diagnostics.subfunctionStatus" pro další informace.
8100	33024	"MessageFilter" je na indexu pole "filterData", který je mimo platné hodnoty. Zkontrolujte "diagnostics.subfunctionStatus" pro informace o pozici v poli "filterData".
8101	33025	ID zprávy je na indexu pole "filterData", který je mimo platné hodnoty. Zkontrolujte "diagnostics.subfunctionStatus" pro informaci o pozici v poli "filterData".
8600	34304	Chyba v důsledku nedefinovaného stavu ve stavovém stroji.

H Seznam chybových hlášení

ID	Name	Alarm text
100	M1_bit_00	M1 - Selhání modulu AC vstupu - Přepětí
101	M1_bit_01	M1 - Selhání modulu AC vstupu - Podpětí
102	M1_bit_02	M1 - Ochrana modulu - vypnutí v důsledku přepětí AC
103	M1_bit_03	M1 - Rezerva
104	M1_bit_04	M1 - Rezerva
105	M1_bit_05	M1 - Rezerva
106	M1_bit_06	M1 - Přepětí na výstupu - DC přepětí
107	M1_bit_07	M1 - Ochrana modulu - vypnutí v důsledku přepětí DC
108	M1_bit_08	M1 - Podpětí na výstupu - DC podpětí
109	M1_bit_09	M1 - Porucha větráku
110	M1_bit_10	M1 - Rezerva
111	M1_bit_11	M1 - Rezerva
112	M1_bit_12	M1 - Ochrana proti přehřátí - Nadměrná teplota prostředí
113	M1_bit_13	M1 - Rezerva
114	M1_bit_14	M1 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 1
115	M1_bit_15	M1 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 2
116	M1_bit_16	M1 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 1
117	M1_bit_17	M1 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 2
118	M1_bit_18	M1 - Porucha modulu - PFC a DC komunikace
119	M1_bit_19	M1 - Rezerva
120	M1_bit_20	M1 - Porucha modulu - Porucha PFC
121	M1_bit_21	M1 - Porucha modulu - Porucha DCDC
122	M1_bit_22	M1 - Rezerva
123	M1_bit_23	M1 - Rezerva
124	M1_bit_24	M1 - Rezerva
125	M1_bit_25	M1 - Status nabíjecího modulu - Vypnut

126	M1_bit_26	M1 - Rezerva
127	M1_bit_27	M1 - Rezerva
128	M1_bit_28	M1 - Rezerva
129	M1_bit_29	M1 - Rezerva
130	M1_bit_30	M1 - Rezerva
131	M1_bit_31	M1 - Rezerva
200	M2_bit_00	M2 - Selhání modulu AC vstupu - Přepětí
201	M2_bit_01	M2 - Selhání modulu AC vstupu - Podpětí
202	M2_bit_02	M2 - Ochrana modulu - vypnutí v důsledku přepětí AC
203	M2_bit_03	M2 - Rezerva
204	M2_bit_04	M2 - Rezerva
205	M2_bit_05	M2 - Rezerva
206	M2_bit_06	M2 - Přepětí na výstupu - DC přepětí
207	M2_bit_07	M2 - Ochrana modulu - vypnutí v důsledku přepětí DC
208	M2_bit_08	M2 - Podpětí na výstupu - DC podpětí
209	M2_bit_09	M2 - Porucha větráku
210	M2_bit_10	M2 - Rezerva
211	M2_bit_11	M2 - Rezerva
212	M2_bit_12	M2 - Ochrana proti přehřátí - Nadměrná teplota prostředí
213	M2_bit_13	M2 - Rezerva
214	M2_bit_14	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 1
215	M2_bit_15	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 2
216	M2_bit_16	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 1
217	M2_bit_17	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 2
218	M2_bit_18	M2 - Porucha modulu - PFC a DC komunikace
219	M2_bit_19	M2 - Rezerva
220	M2_bit_20	M2 - Porucha modulu - Porucha PFC
221	M2_bit_21	M2 - Porucha modulu - Porucha DCDC
222	M2_bit_22	M2 - Rezerva
223	M2_bit_23	M2 - Rezerva
224	M2_bit_24	M2 - Rezerva
225	M2_bit_25	M2 - Status nabíjecího modulu - Vypnut
226	M2_bit_26	M2 - Rezerva
227	M2_bit_27	M2 - Rezerva
228	M2_bit_28	M2 - Rezerva
229	M2_bit_29	M2 - Rezerva
230	M2_bit_30	M2 - Rezerva
231	M2_bit_31	M2 - Rezerva
300	M3_bit_00	M2 - Selhání modulu AC vstupu - Přepětí
301	M3_bit_01	M2 - Selhání modulu AC vstupu - Podpětí
302	M3_bit_02	M2 - Ochrana modulu - vypnutí v důsledku přepětí AC
303	M3_bit_03	M2 - Rezerva
304	M3_bit_04	M2 - Rezerva
305	M3_bit_05	M2 - Rezerva
306	M3_bit_06	M2 - Přepětí na výstupu - DC přepětí

307	M3_bit_07	M2 - Ochrana modulu - vypnutí v důsledku přepětí DC
308	M3_bit_08	M2 - Podpětí na výstupu - DC podpětí
309	M3_bit_09	M2 - Porucha větráku
310	M3_bit_10	M2 - Rezerva
311	M3_bit_11	M2 - Rezerva
312	M3_bit_12	M2 - Ochrana proti přehřátí - Nadměrná teplota prostředí
313	M3_bit_13	M2 - Rezerva
314	M3_bit_14	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 1
315	M3_bit_15	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí PFC 2
316	M3_bit_16	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 1
317	M3_bit_17	M2 - Ochrana proti přehřátí - Ochrana proti přehřátí DC 2
318	M3_bit_18	M2 - Porucha modulu - PFC a DC komunikace
319	M3_bit_19	M2 - Rezerva
320	M3_bit_20	M2 - Porucha modulu - Porucha PFC
321	M3_bit_21	M2 - Porucha modulu - Porucha DCDC
322	M3_bit_22	M2 - Rezerva
323	M3_bit_23	M2 - Rezerva
324	M3_bit_24	M2 - Rezerva
325	M3_bit_25	M2 - Status nabíjecího modulu - Vypnut
326	M3_bit_26	M2 - Rezerva
327	M3_bit_27	M2 - Rezerva
328	M3_bit_28	M2 - Rezerva
329	M3_bit_29	M2 - Rezerva
330	M3_bit_30	M2 - Rezerva
331	M3_bit_31	M2 - Rezerva

I Nahrané přílohy

Na portál STAG byl nahrán soubor typu .rar, který obsahoval následující přílohy:

- Diplomova_prace_Lupomesky.pdf
- CAN_Comunication.rar – Kompletní složka s programem a knihovnou