



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

BOOTLOADER PRO SCI-TRACE

BOOTLOADER FOR SCI-TRACE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Lukáš Novák

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Pavel Šteffan, Ph.D.

BRNO 2016



VYSOKÉ UČENÍ FAKULTA ELEKTROTECHNIKY
TECHNICKÉ A KOMUNIKAČNÍCH
V BRNĚ TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor Mikroelektronika
Ústav mikroelektroniky

Student: Bc. Lukáš Novák

ID: 134573

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Bootloader pro Sci-Trace

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte úpravu elektroniky pro systém Sci-Trace, která umožní sledování stavu teploty a dalších aktuálních informací na displeji. Zaměřte se na jednoduchost programování firmware do řídicího mikrokontroléru přes rozhraní USB a RS485, včetně obslužného programu pro operační systém Windows. Dále navrhněte možnost propojení řídicí elektroniky a propojovacích podružných modulů. Navrhněte nový způsob zapojení a umístění zdroje a základního modulu.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 8.2.2016

Termín odevzdání: 26.5.2016

Vedoucí práce: doc. Ing. Pavel Šteffan, Ph.D.

Konzultant diplomové práce:

doc. Ing. Lukáš Fojcik, Ph.D., *předseda oborové rady*

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení částí druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Brno

ABSTRAKT

V diplomové práci je podrobně rozebrána problematika Bootloaderu pro mikrokontroléry AVR a následně proveden návrh pro implementaci do zařízení Sci-Trace. Úpravy elektronické části se zabývají propojovací deskou, která slouží k propojení ostatních modulů a řídicí deskou, která obstarává komunikaci mezi počítačem a ostatními moduly. Ostatní moduly jsou například pro řízení optiky nebo řízení krokových motorů. Z hlediska uživatelského rozhraní je vyřešen výběr informačního displeje, na kterém se umožní zobrazování vnitřních teplot zařízení. Z mechanického hlediska je vyřešeno uložení elektroniky do rackového systému a s tím spojená zaměnitelnost modulů. Z programové stránky je vyvinut firmware pro řídicí jednotku i pro podružný modul. Pro obsluhu je vyvinut program pro nahrávání aplikačních programů do mikrokontrolérů.

KLÍČOVÁ SLOVA

Atmega644P, FT232R, Bootloader, AVR, Fuse, Rack, Schroff, Atmel, LIBS, Sci-Trace, MAX481.

ABSTRACT

In this thesis analyze in detail issue Bootloader for microcontrollers AVR and made proposal for implementation to the device Sci-Trace. Modification electronics part deal with patch board, where used connect between others module and control board. Control Board handle with computer and individually module. Example others module are control optics or control stepper motors. From user interface is resolve choice information display for show temperature inside the device. From mechanical part is resolve store electronic part to the rack section. From programming part is create firmware for control board and others module. For users operation is create program for operation system for upgrade application firmware to microcontroller.

KEYWORDS

Atmega644P, FT232R, Bootloader, AVR, Fuse, Rack, Schroff, Atmel, LIBS, Sci-Trace, MAX481.

NOVÁK, L. Bootloader pro Sci-Trace. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky, 2016. 59 s, 9 s příloh. Diplomová práce. Vedoucí práce doc. Ing. Pavel Šteffan, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „**Bootloader pro Sci-Trace**“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu diplomové práce doc. Ing. Pavlu Šteffanovi, Ph. D. za důležité, věcné a odborné rady, které pomohly k realizaci této práce. Dále bych rád poděkoval svým rodičům, kteří mi svou podporou umožnili studium na této škole. V neposlední řadě týmu z Laboratoře laserové spektroskopie na Vysokém učení technickém v Brně.

V Brně dne

.....

(podpis autora)

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072
Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

OBSAH

Obsah	vii
Seznam obrázků	ix
Seznam tabulek	xi
Úvod	1
1 Popis návrhu konstrukce	2
1.1 Popis mechanické konstrukce	2
1.2 Popis návrhu propojovací desky	4
2 Návrh schémat	5
2.1 Rozložení pinů v propojovacím konektoru	5
2.2 Napájecí napětí	7
2.3 Řídící jednotka s mikrokontrolérem.....	8
2.4 Komunikace s počítačem	9
2.5 Zobrazovací displej	12
2.5.1 Porovnání displejů	12
2.5.2 Rozmístění pinů na displeji.....	14
3 Výběr teplotního čidla	16
3.1 Parametry číslicového teploměru DS18B20	16
3.2 Popis číslicového teploměru.....	17
3.3 Zapojení číslicového teploměru přes externí napájení.....	17
4 Bootloader	18
4.1 Grafické znázornění zápisu do paměti	18
4.1.1 Rozdělení Flash paměti.....	19
4.1.2 Instrukce SPM.....	19
4.1.3 Smazání stránky	21
4.1.4 Načtení dat do Page Bufferu	21
4.1.5 Zápis do paměti.....	21
4.1.6 Příznak zaneprázdnění RWW sekce	21
5 Bootloader-Master	22

5.1	Komunikace s počítačem	23
5.2	Inicializační část	23
5.3	„Stavový automat“	24
5.4	Program HexLoaderMaster	31
5.4.1	Popis programu	31
5.4.2	Správný postup nahrávání firmwaru do řídicí desky	34
6	Bootloader-Slave	35
6.1	Inicializační část	36
6.2	Stavový automat	38
6.3	Program HexLoaderSlave	44
6.3.1	Popis programu	45
6.3.2	Správný postup nahrávání do externích modulů	48
7	Aplikační část řídicí jednotky	49
7.1	Výpočty pro děliče napětí	49
7.2	Výpočet převodu pro program	50
7.3	Inicializační část	50
7.4	Firmwarové úpravy aplikačního programu	52
8	Sestavené zařízení	53
9	Závěr	54
	Literatura	55
	Seznam zkratk a vysvětlivky	57
	Seznam příloh	59
A	Návrh zařízení	60

SEZNAM OBRÁZKŮ

Obr. 1: Mechanická montáž pro elektrotechnická zařízení o velikosti 3U.....	3
Obr. 2: Rozměry propojovací desky s uložením konektorů.	4
Obr. 3: Rozložení napájení v propojovacím konektoru.	6
Obr. 4: Rozložení synchronizační části v propojovacích konektorech.....	6
Obr. 5: Návrh napájecí části.....	7
Obr. 6: Rozložení pinů u mikrokontroléru ATmega644P.	8
Obr. 7: Vysílač MAX481 pro sběrnici RS485.....	9
Obr. 8: Obvod FT232R v pouzdrů SSOP.	10
Obr. 9: a) Aktivní displej s tekutými krystaly. b) Pasivní displej s aktivními krystaly. c) Alfnumerický displej. d) OLED displej.	13
Obr. 10: Zapojení pinů displeje na konektor v řídicí desce.	15
Obr. 11: Napájení číslicového teploměru přes externí zdroj.	17
Obr. 12: Vývojový diagram pro zápis dat do paměti.....	18
Obr. 13: Rozdělení FLASH paměti v Atmega mikrokontrolérech [10].	19
Obr. 14: Registr SPMCR.	20
Obr. 15: Inicializační část Bootloader-Master.	24
Obr. 16: Stavový automat první část.	25
Obr. 17: Stavový automat druhá část.....	26
Obr. 18: Vývojový diagram pro zápis do FLASH a EEPROM paměti.....	28
Obr. 19: Stavový automat pro čtení z FLASH paměti.....	29
Obr. 20: Stavový automat pro čtení z EEPROM paměti.	30
Obr. 21: Popis programu HexLoaderMaster	31
Obr. 22: List „MCU settings“.....	33
Obr. 23: List „HEX uploader“.....	34
Obr. 24: Inicializace programu Bootloader-Slave.	37
Obr. 25: Vývojový diagram pro příkazy 1, 11, 12, 13 a 14.....	38
Obr. 26: Vývojový diagram pro příkazy 15 až 19.	39
Obr. 27: Vývojový diagram pro příkazy 20 až 22.....	40
Obr. 28: Vývojový diagram pro zápis do FLASH paměti.....	41
Obr. 29: Vývojový diagram pro zápis do EEPROM paměti.	42
Obr. 30: Konec zápisu do paměti.	43

Obr. 31: Vývojový diagram pro příkazy 26 až 28.	44
Obr. 32: List „Module settings“.....	45
Obr. 33: List „Tribus uploader“.....	47
Obr. 34: Manuální zadávání příkazů.	48
Obr. 35: Inicializace aplikačního programu.	51
Obr. 36: Vývojový diagram s upraveným firmwarem.....	52
Obr. 37: Model celého zařízení.	53

SEZNAM TABULEK

Tab. 1: Přehled mechanických konstrukcí od firmy Schroff.	2
Tab. 2: Rozložení pinů u převodníku FT232R.....	10
Tab. 3: Tabulka rozložení pinů na displeji.....	14
Tab. 4: Porovnání teplotních čidel.	16
Tab. 5: Funkce jednotlivých pinů u teploměru Dallas.	17
Tab. 6: Popis registru SPMCR (SPMCSR).....	20
Tab. 7: Tabulka příjmu verifikace dat.....	27
Tab. 8: Tabulka výpisu paměti z jedné stránky.	28
Tab. 9: Výpis celé paměti.	29
Tab. 10: Datový paket protokolu Trinamic.....	35
Tab. 11: Datový paket příjmu pro komunikaci s Bootloader-Slave.	35
Tab. 12: Odchozí datový paket pro komunikaci s řídicí jednotkou.	36
Tab. 13: Tabulka výpočtu děličů pro AD převodník.	49
Tab. 14: Tabulka výpočtu převodu na skutečnou hodnotu.	50

ÚVOD

Zařízení Sci-Trace slouží pro prvkovou analýzu vzorků v Laboratoři laserové spektroskopie na Vysokém učení technickém v Brně. Měření je založené na metodě laserové spektrometrie buzeného plazmatu (LIBS). Zařízení se skládá z přístrojové skříně obsahující pulzní laser, spektrální analyzátor, řídicí elektroniku, vzduchotechniku a další části. Na vrchní části na optické desce je uložena interakční komora, ve které probíhá samotná analýza.

V interakční komoře je uložen vzorek a pomocí pulzního laseru je odpařena část vzorku do vzduchu, případně do vakua či přetlaku a pomocí spektrálního analyzátoru se zjišťuje složení vzorku.

Pro ovládání Sci-Trace zařízení je zásadní elektronika, která je rozdělená na základní propojovací desku, do které se vkládají moduly obsluhující jednotlivé části, jako je ovládání manipulátoru či ovládání osvětlení uvnitř interakční komory. Nedostatkem současného řešení je propojovací deska, u které není možné měnit moduly mezi jednotlivými sloty, bez složitého vnitřního zásahu. Toto řešení není příliš efektivní při údržbě a rozšiřování elektronické části. Jedním z cílů této práce je vyřešit výše zmíněný problém.

Dalším zásadním problémem je aktualizace softwaru u modulů. V současné době se provádí aktualizace, vyjmutím modulu a přenesením do servisních prostor. V servisních prostorech je přes programovací jednotku, například AVR Dragon, přehrán software modulu. Následně je modul navrácen do elektroniky. Tyto nutné zásahy do elektroniky zvenčí jsou problematické při sériové výrobě zařízení. Zařízení lze totiž prodat do celého světa a aktualizace firmwaru prodávaného přístroje je problematická. Aktualizací firmwaru je možné opravit bugy, případně změnit vnitřní logiku systému.

V současné době je znám systém zvaný Bootloader, který dokáže tento problém vyřešit a lze nahrávat nový firmware do jednotlivých modulů přes stávající rozhraní k elektronice. Pro zavedení systému Bootloader je však nutná úprava jak elektronické, tak firmwarové části.

1 POPIS NÁVRHU KONSTRUKCE

Před návrhem samotné elektronické části, je důležité vybrat potřebnou konstrukci, ve které bude samotná elektronika uložena. Pro jednoduchost výměny a standardizování je nutné zvážit nejvhodnější konstrukci do zařízení. Tuto vlastnost splňuje mechanická konstrukce tzv. rackový systém pro elektrotechnická zařízení dle normy [1].

Nezbytným krokem je proto nutné navrhnout propojovací desku uchycenou v mechanické konstrukci, která propojuje jednotlivé moduly a lze s nimi tedy komunikovat prostřednictvím řídicí desky, kterou je nutné navrhnout v dalším kroku. Propojovací deska musí být navržena s ohledem na záměnnost modulů mezi různými pozicemi. Tato vlastnost je velmi prospěšná pro přidávání nových typů modulů.

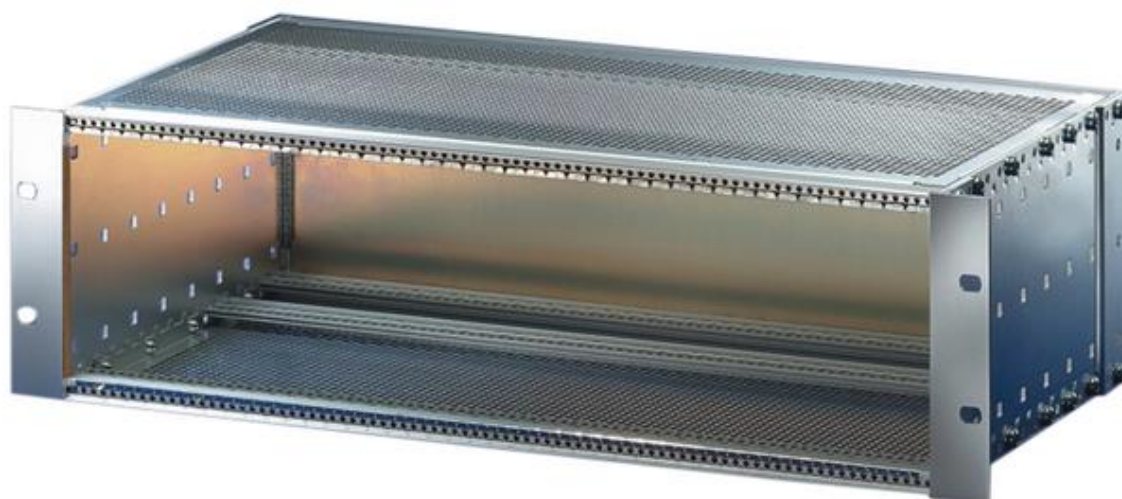
1.1 Popis mechanické konstrukce

Mechanický konstrukční systém vyrábí několik firem a lze je zaměňovat díky standardizaci. Z nepřeberného výběru firem, byla vybrána pouze jedna firma Schroff, která již řadu let vyrábí mechanické konstrukce pro elektrotechnická zařízení a lze jí považovat za ověřenou firmu s kvalitními výrobky a to lze říci i z osobních zkušeností. Jelikož firma Schroff vyrábí několik typů konstrukcí, je v tabulce **tab. 1** zobrazen a zpracován jen výběr několika mechanických konstrukcí.

Tab. 1: Přehled mechanických konstrukcí od firmy Schroff.

Výrobní označení	Počet karet [HP]	Výška[U]	Hloubka [mm]	Rozdělení hloubky [mm]
24563-171	84	3	175	175
24563-172	84	3	235	175/235
24563-173	84	3	295	175/235/295
24563-174	84	3	355	175/235/295/355
24563-461	84	6	175	175
24563-462	84	6	235	175/235
24563-463	84	6	295	175/235/295
24563-464	84	6	355	175/235/295/355

Značení výšky pro 19 palcové mechanické konstrukce od firmy Schroff je v rackových jednotkách pod písmenem U (Unit), kde jedna jednotka odpovídá velikosti 44,45 mm. Počet karet naopak u mechanických konstrukcí je značen jednotkou HP (Horizontal Pitch), která je definována normou pro eurokartu ČSN EN 60603-2 [2]. Jedna HP jednotka odpovídá velikosti 5,08 mm. Základní rozměry mechanické konstrukce pro elektrotechnická zařízení řady 19 palcových mechanických konstrukcí je možné nalézt v normě ČSN EN 60297-3-100 [1]. U rozdělení hloubky lze nalézt až čtyři rozměry, kde každý rozměr představuje příčku rozdělující mechanickou konstrukci, pro představu u výrobku 24563-172 (viz **obr. 1**) rozměr o velikosti 175 mm značí místo pro jednu eurokartu a druhý rozměr 235 mm značí poslední příčku, která odpovídá celkové hloubce mechanické konstrukce. Lze tedy jednoduchou matematikou dopočítat zbývající prostor o velikosti 60 mm, kde lze navrhnout řídicí jednotku, vložit napájecí zdroj, případně vyvést konektory z řídicí jednotky na zadní část mechanické konstrukce panelu.



Obr. 1: Mechanická montáž pro elektrotechnická zařízení o velikosti 3U.

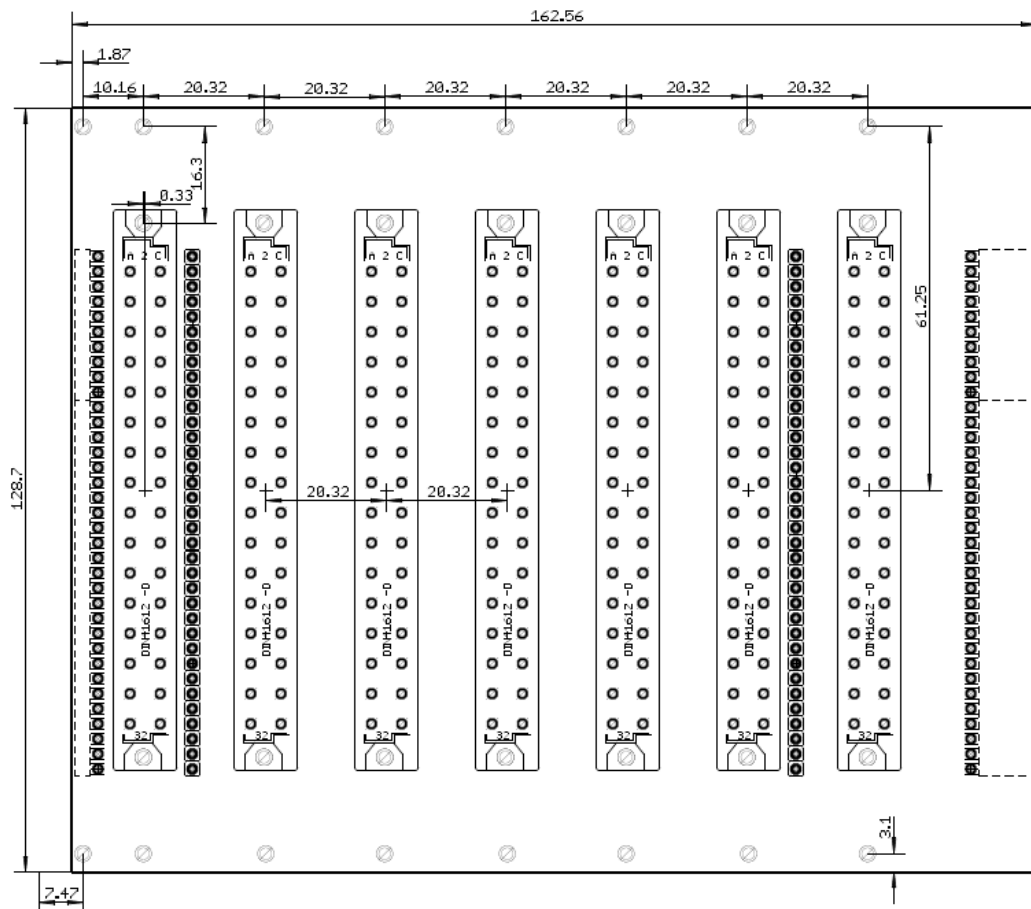
Výška mechanické konstrukce byla zvolena 3 U, neboť moduly samotné nejsou nijak složité a nepotřebují dvojnásobný počet místa. Pro eurokartu o výšce 3 U je definován pouze jeden konektor se standardním označením DIN 41612. Tento typ konektoru lze nalézt v nepřeberném množství variant, kde největší rozdíl tvoří počet řad a roztečí mezi jednotlivými piny na výšku. Typ o třech řadách disponuje 96 piny s roztečí 2,54 mm a typ o dvou řadách s roztečí 5,08 mm má 32 pinů. Pro velikost 6 U je počet pinů dvojnásobný a pro 9 U je počet dokonce trojnásobný. Avšak pro naše potřeby vysoce postačuje typ o dvou řadách s 32 piny pro napájecí napětí a signálové komunikační vodiče propojující moduly. Z tabulky byla nakonec vybrána mechanická konstrukce s výrobním označením 24563-174, která má dostatečný zbývající prostor pro uložení řídicí elektroniky, napájecího zdroje a volný prostor pro případné další komponenty.

1.2 Popis návrhu propojovací desky

Po návrhu mechanické konstrukce, je nutné navrhnout propojovací desku (viz **obr. 2**). Jako jeden z hlavních bodů je záměnnost karet v mechanické konstrukci. Díky nutnosti tohoto bodu odpadá možnost vyvést jednotlivé vstupně výstupní piny na zadní konektor eurokarty, což by ztratilo možnost této výměny. Propojovací deska proto bude sloužit pouze pro synchronizaci, komunikaci a napájení. Proto v další iteraci je nutné upravit moduly s vyvedenými vstupně výstupní piny na konektory umístěnými na předních panelech.

I přes tyto negativa, lze považovat vyvedení pinů na přední strany panelů za pozitivum, jelikož díky vybrané mechanické konstrukci lze přesunout do volného prostoru řídicí jednotku a napájecí zdroj. A to díky vybrané mechanické konstrukci s 180 mm místa mezi vnitřní příčkou a zadní částí. Řídicí jednotka a zdroj byly v předchozí verzi umístěny jako moduly, lze tak přesunutím do „skryté části“ ušetřit v mechanické konstrukci místo pro nové přídavné moduly.

Samotnou propojovací desku bylo nutné navrhnout dle normy [1] s libovolným počtem konektorů, kde byl nakonec zvolen minimální počet 7 modulů o velikosti 4 HP s další možností připojení stejné desky vedle sebe. Možnost rozšíření bylo navrženo se standartními Molex konektory a lze tak zaplnit celou mechanickou konstrukci bez nutnosti úpravy desky.



Obr. 2: Rozměry propojovací desky s uložením konektorů.

2 NÁVRH SCHÉMAT

Návrh celé elektronické části musel být rozdělen na dvě části, pro řídicí jednotku a pro propojovací desku, do které se připojují externí moduly pro ovládání zařízení Sci-Trace. Samotná řídicí jednotka musela být, pro přehlednost schémat, navržena do několika částí. Kompletní schémata lze nalézt v přílohách na konci dokumentu.

První část zahrnuje samotné rozložení pinů v konektoru, a propojení mezi propojovací deskou a řídicí deskou. Druhá část zahrnuje návrh napájecích napětí a rozvedení po desce. Třetí část zahrnovala návrh samotné řídicí jednotky s mikrokontrolérem a převodníkem UART na RS485. Čtvrtá část byla zaměřena na převodník z UART na USB s galvanickým oddělením. V této části se muselo zároveň počítat s implementací softwarové části Bootloader, která potřebuje pin při nahrávání nové verze firmware do mikrokontroléru připojený na reset naší řídicí jednotky. Poslední část zahrnovala návrh konektoru pro námi zvolený displej a konektor pro připojení teplotních čidel k mikrokontroléru přes 1-Wire sběrnici.

Propojovací deska oproti tomu nebyla z hlediska elektrického rozložení příliš složitá, jelikož obsahuje pouze konektory a filtrační kondenzátory. Nejsložitější návrh u této desky bylo dodržení všech rozmístění a uložení dle normy [2].

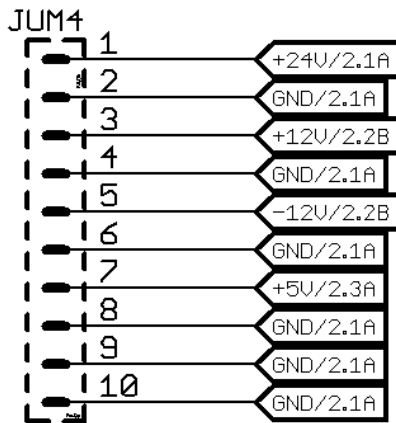
2.1 Rozložení pinů v propojovacím konektoru

Rozložení pinů pro propojovací konektor bylo rozděleno na napájecí část (viz **obr. 3**) a synchronizační část (viz **obr. 4**), kde synchronizační část byla rozdělena na dvě části. Toto rozdělení synchronizační části bylo zvoleno pouze z praktického hlediska, kde byly vybrány standardní velikosti konektorů bez nutnosti objednávky speciálních konektorů s definovaným počtem pinů.

U napájecí části byly zvoleny následující napětí:

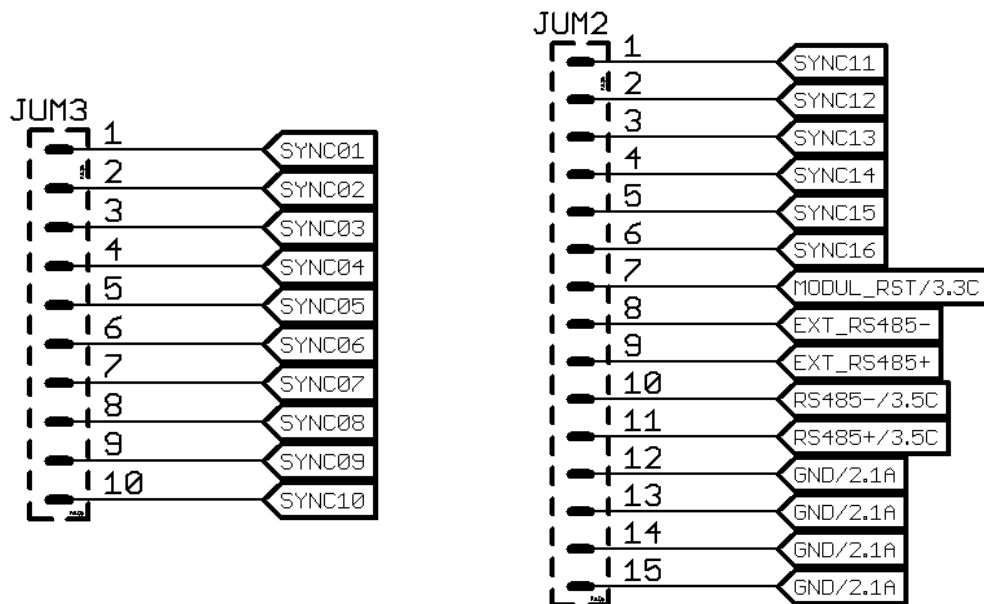
- Napájení +24 V
- Napájení +12 V
- Napájení -12 V
- Napájení +5 V

Napájení +24 V bylo zvoleno, kvůli krokovým motorům, které v zařízení Sci-Trace jsou ovládány přes tuto napěťovou logickou úroveň. Napájecí napětí +12 V a -12 V nebylo v současné době využito, avšak lze do budoucna možnost tyto napětí využít v nových modulech bez změny celé propojovací a řídicí desky. Napájecí +5 V bylo zvoleno pro napájení mikrokontrolérů jak v řídicí desce, tak pro napájení mikrokontrolérů v ostatních modulech.



Obr. 3: Rozložení napájení v propojovacím konektoru.

U synchronizační části byly signály zvoleny pro sběrnici RS485 komunikující mezi moduly, resetující pin pro využití softwarové části Bootloader a ostatních synchronizačních signálů.



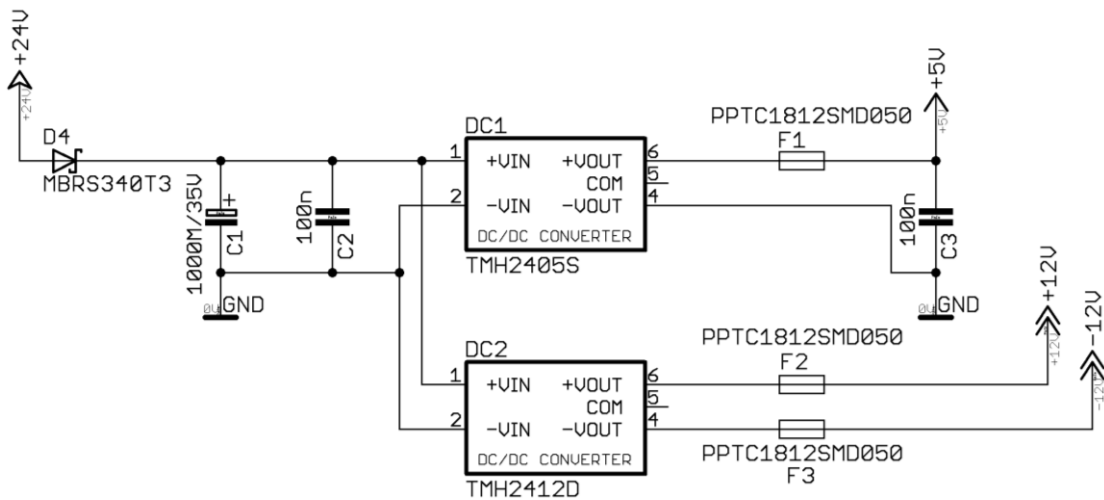
Obr. 4: Rozložení synchronizační části v propojovacích konektorech.

2.2 Napájecí napětí

Napájení celé sestavy bylo zvoleno ze spínaného zdroje SP-150-24, který ze síťového napětí vytváří +24 V, které je dále rozvedeno po desce pro využití napájení součástek, měničů či koncových zařízení.

Pro vytvoření menších napětí bylo potřeba vybrat usměrňovače, případně DC-DC měniče, které splňovaly nároky pro vysokou dobou pracovního režimu, vysokou účinností, dostatečným výkonem pro napájení zařízení a ochranou proti zkratu zařízení. Tyto vlastnosti spíše splňovaly DC-DC měniče, u kterých byla vybrána série TMH od firmy TRACO POWER. Tato série se vyznačuje s účinností 83 %, pracovní dobou 8 hodin, kde po tuto dobu je deklarován 100 % chod součástky s ochranou proti zkratu, která je ovšem limitována pouze na maximálně 1 sekundu. Bohužel 100 % ochrana proti zkratu se u jiných měničů nachází avšak ve vyšší cenové kategorii, proto dosavadní ochrana proti zkratu byla ještě doplněna o ochranu ve formě elektronické pojistky. Tato řada se zároveň vyznačuje i měniči se symetrickým napájením, kde byl vybrán model TMH2412D pro usměrnění ± 12 V a druhý bez symetrického napájení s označením TMH2405S pro napájení +5 V větve.

Elektronická pojistka se vyznačuje malým odporem, který se zvýší při dosažení dvojnásobného pracovního proudu. Po odstranění zkratu, tedy zmenšení protékajícího proudu se pojistka vrátí do normálního režimu s malým odporem. Nejdůležitější je však doba dosažení vyššího odporu pojistky, kde u vybraného typu dosahuje 0,15 sekund. Tato doba nám tak splňuje maximální limit ochrany proti zkratu u měniče a zkrat nám tak nezničí náš měnič. Výběr napájecího napětí ze schématu je zobrazeno na **obr. 5**.



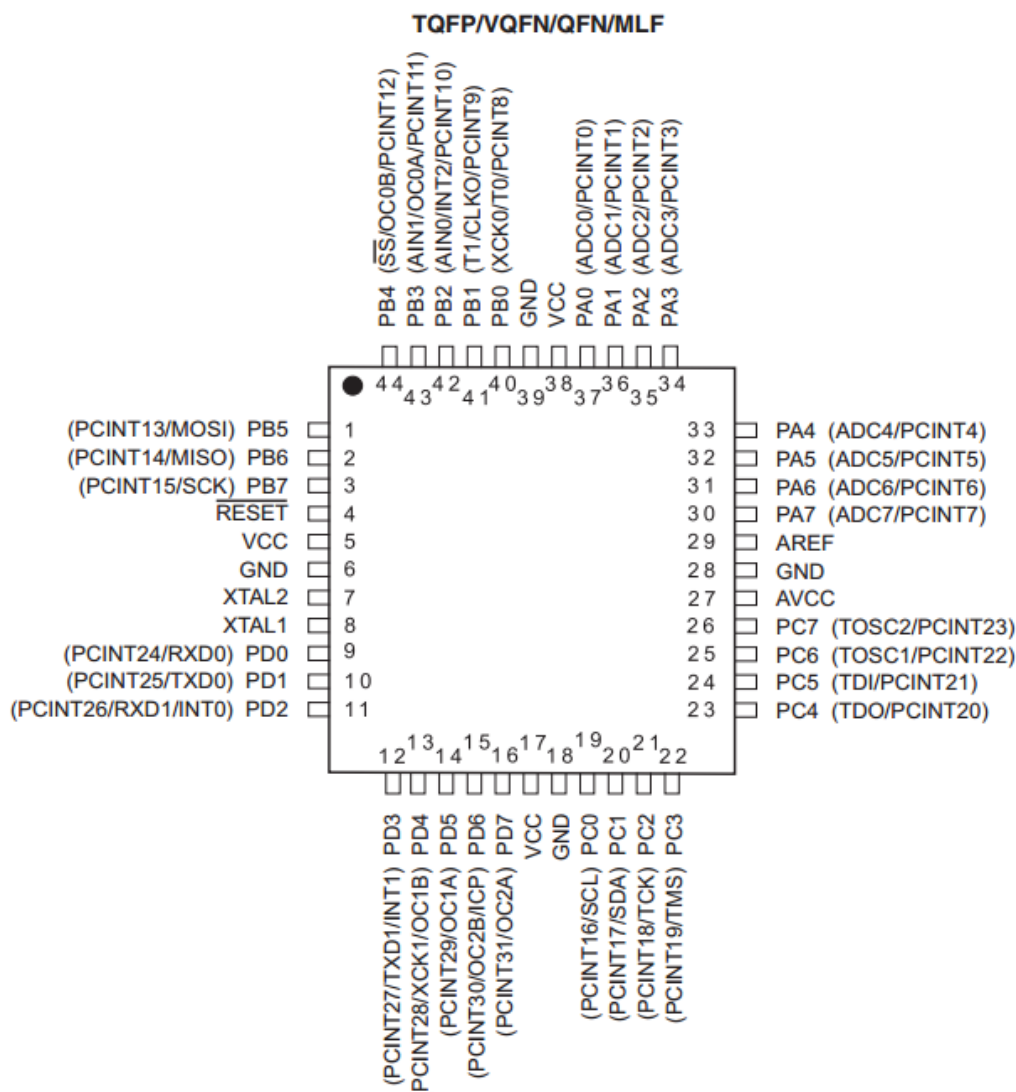
Obr. 5: Návrh napájecí části.

2.3 Řídicí jednotka s mikrokontrolérem

Jedna z nejdůležitějších částí v elektronice je samotná řídicí jednotka, u které je potřeba komunikovat jak s uživatelem, přes počítač s využitím USB portu, tak s jednotlivými moduly přes RS485 sběrnici. Z těchto důvodů musel být vybrán mikrokontrolér s dvěma hardwarovými UART sběrnici, dostatkem pinů pro ovládání displeje, připojení teplotních čidel a dostatečnou velikostí paměti pro zpracování dat.

Naším konkrétním specifikacím vyhovoval osmibitový mikrokontrolér od firmy Atmel [3] s označením ATmega644 (viz **obr. 6**), s následujícími parametry:

- Velikostí paměti 64kB,
- dvě UART sběrnice,
- maximální pracovní frekvencí 20 MHz
- maximálním počtem 32 využitelných vstupně výstupních pinů.

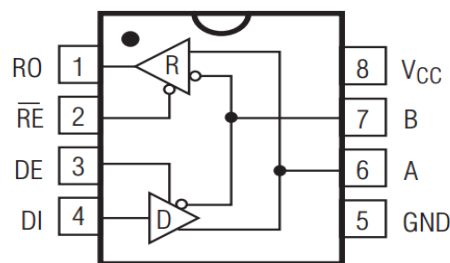


Obr. 6: Rozložení pinů u mikrokontroléru ATmega644P.

Mikrokontrolér se vyrábí ve velkém množství pouzder. Pro naše využití bylo v konečném důsledku zvoleno SMD provedení v pouzdru TQFP se 44 piny pro naši aplikaci. Tento mikrokontrolér je i pinově kompatibilní s ATmega1284P, který se vyznačuje dvojnásobnou velikostí paměti a lze ho případně využít při nedostatku paměti.

Zapojení mikrokontroléru bylo provedeno dle zapojení v katalogovém listu s 2,5 V referencí vytvořenou přes napěťový referenční čip TL431 v pouzdru SO08. Pro vývoj firmwaru zařízení byl přidán konektor JTAG (nestandardizovaný) bez nutnosti změny návrhu desky a konektor pro sériovou výrobu pro prvotní naprogramování konektor ISP (nestandardizovaný).

Pro komunikaci přes sběrnici RS485 byl vybrán vysílač od firmy Maxim Integrated s výrobním číslem MAX481 [4], vyobrazený na **obr. 7**, ke kterému dokážeme připojit až 32 zařízení na jednu sběrnici, s polo-duplexním režimem a maximální rychlostí přenosu až 2 Mbps. Tento čip byl vybrán v SMD provedení v pouzdru SO08.



Obr. 7: Vysílač MAX481 pro sběrnici RS485.

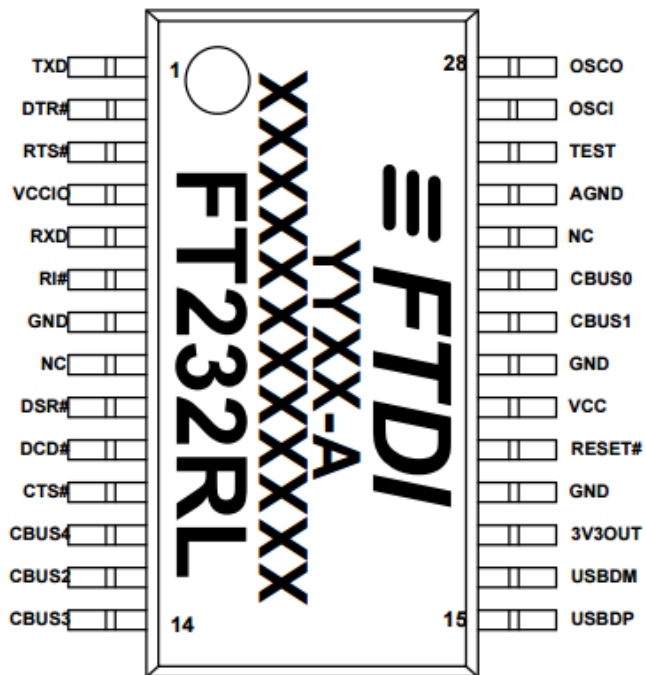
Piny RO a DI, slouží pro připojení k UART sběrnici mikrokontroléru. Dále piny \overline{RE} a DE slouží pro ovládání připojení k pinu mikrokontroléru pro externí přerušování obstarávající řízení po sběrnici. Piny A a B jsou propojovací sběrnice RS485 k dalšímu čipu a dalšímu čipu MAX481 uloženém například v externím modulu, který komunikuje s dalším mikrokontrolérem.

2.4 Komunikace s počítačem

Při návrhu složitější elektroniky řízené přes počítač se řídicí jednotka neobejde bez zvolení vhodného přenosu dat. Jeden z nejčastějších přenosů dat je prováděn přes USB rozhraní, které je dobře aplikovatelné do mikrokontrolérů přes převodník, případně v samotné implementaci USB rozhraní přes HID zařízení zprostředkující komunikaci s uživatelem. Druhá možnost je však náročnější na paměť a samotné naprogramování softwarové části, proto pro naše využití byla zvolena první možnost přes převodník UART na USB rozhraní.

Nejpoužívanějším zástupce převodníků na USB je obvod FT232R [5] vyráběný firmou Future Technology Devices International. Lze ho napájet jak 5 voltovou logikou, případně až nižším napětím +3,3 V. Samozřejmostí u tohoto obvodu je velká podpora operačních systémů a knihoven pro návrh softwaru ze strany počítače.

Pro velkou podporu knihoven pro vývoj softwaru ze strany počítače, byl tento obvod využit v návrhu. Obvod se zároveň vyrábí ve dvou pouzdech QFN a SSOP, kde pro miniaturizaci návrhu bylo zvoleno pouzdro SSOP, které lze vidět na **obr. 8**.



Obr. 8: Obvod FT232R v pouzdrú SSOP.

Pro přehlednost byly do tabulky zapsány popisy všech pinů.

Tab. 2: Rozložení pinů u převodníku FT232R.

Číslo pinu	Název	Typ	Popis
1	TXD	Výstup	Vysílání asynchronních dat po UART.
2	DTR#	Výstup	Výstupní signál UART rozhraní.
3	RTS#	Výstup	Výstupní signál UART rozhraní.
4	VCCIO	PWR	Napájení pro UART komunikaci a CBUS skupinu.
5	RXD	Vstup	Příjem asynchronních dat po UART.
6	RI#	Vstup	Vstupní signál UART rozhraní.
7	GND	PWR	Uzemnění zařízení
8	N.C.	NC	Nezapojený pin.
9	DSR#	Vstup	Vstupní signál UART rozhraní.
10	DCD#	Vstup	Vstupní signál UART rozhraní.
11	CTS#	Vstup	Vstupní signál UART rozhraní.

Číslo pinu	Název	Typ	Popis
12	CBUS4	I/O	Konfigurační pin, funkce pinu lze nakonfigurovat ve vnitřní EEPROM. Při uvedení zařízení do továrního nastavení funkce pinu odpovídá režimu SLEEP#. Podrobnější informace lze nalézt v tabulce CBUS konfiguraci v katalogovém listu.
13	CBUS2	I/O	Konfigurační pin, funkce pinu lze nakonfigurovat ve vnitřní EEPROM. Při uvedení zařízení do továrního nastavení funkce pinu odpovídá režimu TXDEN. Podrobnější informace lze nalézt v tabulce CBUS konfiguraci v katalogovém listu.
14	CBUS3	I/O	Konfigurační pin, funkce pinu lze nakonfigurovat ve vnitřní EEPROM. Při uvedení zařízení do továrního nastavení funkce pinu odpovídá režimu PWREN#. Podrobnější informace lze nalézt v tabulce CBUS konfiguraci v katalogovém listu.
15	USBDP	I/O	USB datový kladný pin.
16	USBDM	I/O	USB datový záporný pin.
17	3V3OUT	Výstup	Výstupní napětí +3,3 V pro LDO regulátor.
18	GND	PWR	Uzemnění zařízení.
19	RESET#	Vstup	Resetovací pin. Pro normální režim, lze nechat nezapojen, případně připojit přes Pull-Up na napájení.
20	VCC	PWR	Napájení zařízení. Napájení od +3,3 V o +5,25 V.
21	GND	PWR	Uzemnění zařízení.
22	CBUS1	I/O	Konfigurační pin, funkce pinu lze nakonfigurovat ve vnitřní EEPROM. Při uvedení zařízení do továrního nastavení funkce pinu odpovídá režimu RXLED#. Podrobnější informace lze nalézt v tabulce CBUS konfiguraci v katalogovém listu.
23	CBUS0	I/O	Konfigurační pin, funkce pinu lze nakonfigurovat ve vnitřní EEPROM. Při uvedení zařízení do továrního nastavení funkce pinu odpovídá režimu TXLED#. Podrobnější informace lze nalézt v tabulce CBUS konfiguraci v katalogovém listu.
24	N.C.	NC	Nezapojený pin.
25	AGND	PWR	Analogová zem pro uzemnění vnitřních hodin.
26	TEST	Vstup	Pro normální režim musí být uzemněn.
27	OSCI	Vstup	Může být ponechán nepřipojen pro normální režim.
28	OSCO	Výstup	Může být ponechán nepřipojen pro normální režim.

Zapojení pinů (viz **tab. 2**) samotného převodníku bylo provedeno dle katalogového listu. V prvním návrhu se počítalo s galvanickým oddělením komunikačních pinů RXD, TXD a DTR, které jsou připojeny k mikrokontroléru k pinům RX, TX, RESET a samotný FTDI čip by byl napájen z připojeného USB portu. Tento návrh však byl v pozdějším návrhu změněn, kvůli chybě, kde při vypnutí elektroniky se neodpojil USB v počítači. Software v počítači tak nepoznal, jestli je elektronika zapojena nebo ne. Na desce jsou dále připojeny informační LED diody pro indikaci příjmu a vysílání. Tyto diody slouží v testovací fázi pro kontrolu datového toku mezi řídicí deskou a počítačem, jinak neplní žádnou jinou funkci.

USB konektor pro připojení k převodníku byl pro miniaturizaci zvolen mini USB. Pro optickou kontrolu napájení, neboli připojení k počítači, byla přidána LED dioda mezi napěťový pin a zem USB portu.

2.5 Zobrazovací displej

Podmínkou pro zařízení byl zobrazovací displej, který by měl obstarávat informace o vnitřní teplotě a dalších informacích jako připojení k počítači, hodnota napětí na napájecích signálech, připojené řídicí jednotce k USB portu, či napětí signálových pinech RS485 sběrnice.

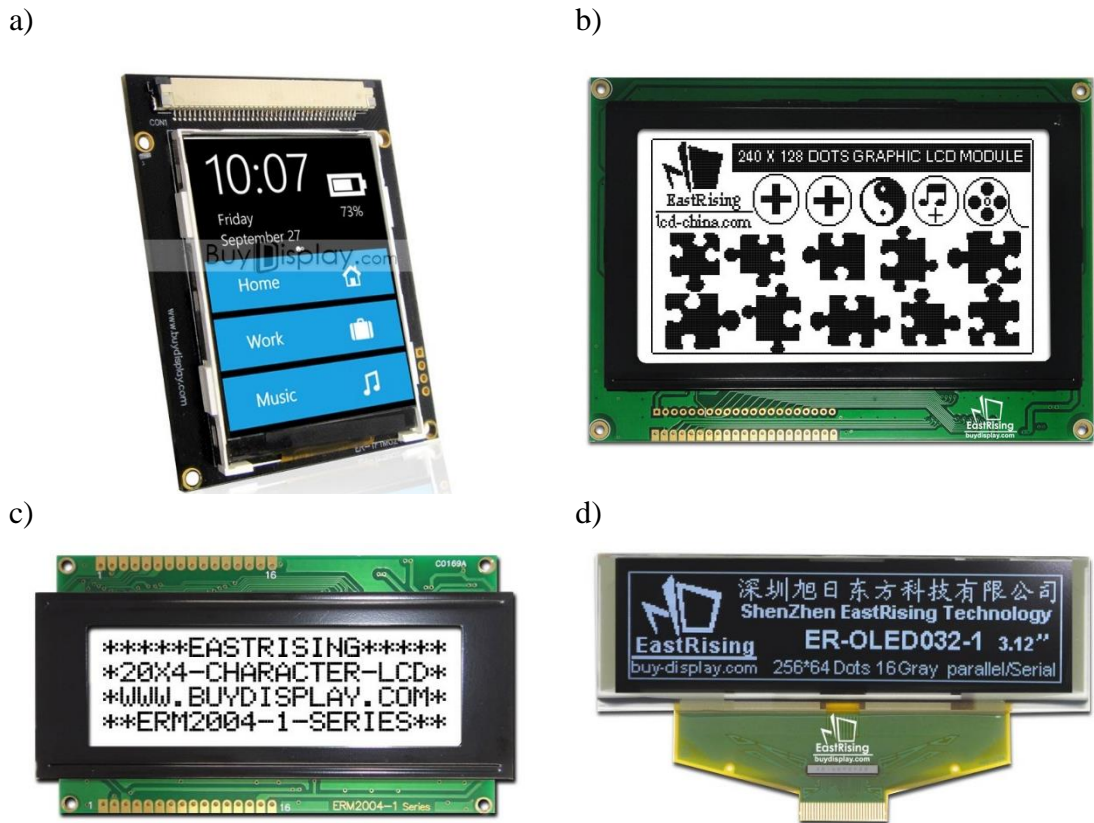
Displej by měl být umístěn v přední části mechanické konstrukce v panelu, proto by neměl mít příliš velké rozměry, avšak pro dobrou čitelnost ani příliš malé. Jelikož z konstrukčního hlediska byla vybrána mechanická konstrukce s výškou 3 U, musí mít displej maximální montážní rozměry na výšku 13,335 cm. Dalším parametrem pro výběr displeje byla dobrá čitelnost a malá náročnost paměti ze softwarového hlediska.

2.5.1 Porovnání displejů

Výběr displejů pro naše zařízení byl proveden u zahraničního prodejce na adrese [6], kvůli vysokým pořizovacím nákladům displejů od tuzemských dodavatelů. U zvoleného zahraničního prodejce lze vybrat několik druhů displejů, od aktivního displeje s tekutými krystaly, pasivního displeje s tekutými krystaly, alfanumerickým displejem nebo OLED displejem.

Z výše uvedených skupin displejů byly vybrány zástupci pro porovnání:

- Aktivní displej s tekutými krystaly – ER-TFTM024-3 (viz **obr. 9a**).
- Pasivní displej s tekutými krystaly – ERM240128FS-2 [7] (viz **obr. 9b**).
- Alfanumerický displej – ERM2004FS-1 (viz **obr. 9c**).
- OLED displej – ER-OLED032-1W (viz **obr. 9d**).



Obr. 9: a) Aktivní displej s tekutými krystaly. b) Pasivní displej s aktivními krystaly.
 c) Alfanaumerický displej. d) OLED displej.

Všechny displeje mají své výhody i nevýhody. Pro nás byly nejdůležitější parametry dobrá viditelnost a hlavně co nejmenší nároky na paměť mikrokontroléru. Z pohledu komunikace, se zdálo být nejlepší využít alfanumerický displej, který má velkou podporu knihoven a řadič je dosti využíván, ale nevýhodou je pouze čtyřřádkový displej, což je nevyhovující z uživatelského hlediska. Další možností je využití aktivního displeje s tekutými krystaly, jenže tento displej je barevný a je tedy náročnější pro ovládání z pohledu mikrokontroléru. OLED displej naproti tomu s výbornou spotřebou, ale opět s malými rozměry na výšku. Kvůli výše uvedeným důvodům byl vybrán pasivní displej s tekutými krystaly s bílým pozadím a černým textem pro dobrou čitelnost, i snadnějšímu ovládání displeje.

K ovládání pasivního displeje lze využít ovládání přes osmibitovou paralelní Intel sběrnici 8080. Zároveň výrobce poskytuje demonstrační kód s knihovnou pro tento displej s komunikací pro osmibitové mikrokontroléry od firmy Atmel, kterou lze využít v našem firmwaru. U displeje je 21 montážních děr s roztečí 2,54 mm pro montáž Molex konektoru.

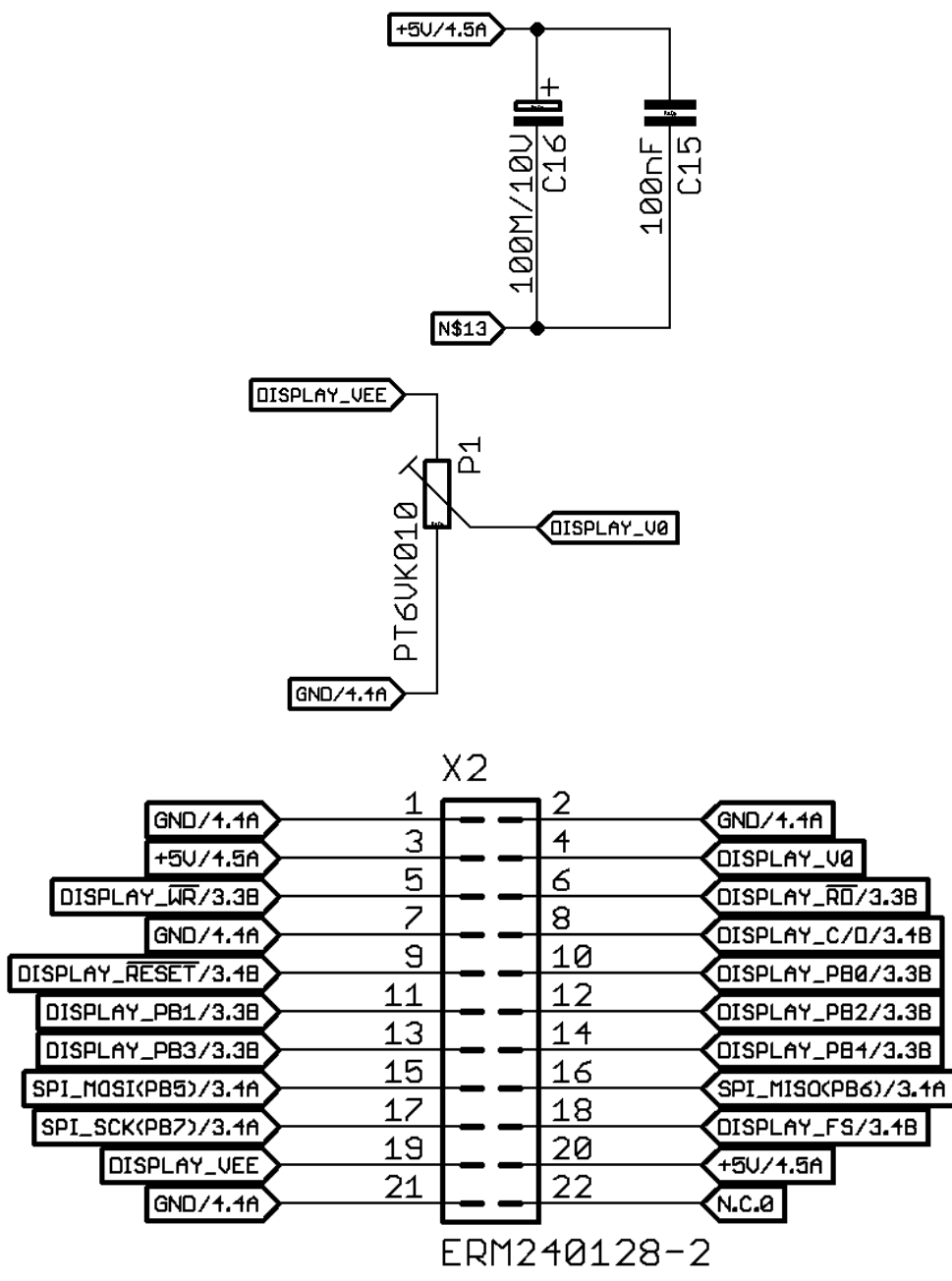
Pro návrh do řídicí desky se využilo konektoru s označením 87832-2220 s SMD montáží a s 22 piny (viz **obr. 10**). Protikus konektoru je konstrukčně řešený pro plochý kabel s roztečí 2 mm, což je rychlejší způsob výroby pro pozdější sériovou výrobu. Dále na desce byl přidán potenciometr pro regulaci jasu displeje. Tento potenciometr bude možné využít pouze při výrobě pro doladění jasu a je nepřístupný pro standardního uživatele.

2.5.2 Rozmístění pinů na displeji

Pro přehlednost jsou do **tab. 3** vypsány jednotlivé piny u displeje a následně popsány.

Tab. 3: Tabulka rozložení pinů na displeji.

Číslo pinu	Název pinu	Popis
1	FG	Uzemnění rámečku, připojení na VSS pin.
2	VSS	Zemnicí pin. Připojení na GND.
3	VDD	Napájecí pin.
4	V0	Vstupní pin pro změnu kontrastu displeje.
5	\overline{WR}	Zápis dat, nebo příkazu.
6	\overline{RD}	Čtení dat, nebo příkazu.
7	\overline{CE}	Povolovací signál.
8	\overline{CD}	Nastavení displeje.
9	\overline{RST}	Resetování displej, při stažení do logické nuly.
10	DB0	8 bitová datová komunikace
11	DB1	
12	DB2	
13	DB3	
14	DB4	
15	DB5	
16	DB6	
17	DB7	
18	FS1	Nastavení rozměrů fontu.
19	VEE	Posilovací pin pro kontrast
20	LED_A	Anoda – Podsvícení pozadí
21	LED_K	Katoda – Podsvícení pozadí



Obr. 10: Zapojení pinů displeje na konektor v řídicí desce.

3 VÝBĚR TEPLOTNÍHO ČIDLA

Pro zobrazování teplot na informačním displeji je potřeba vybrat čidla, která budou uložena uvnitř zařízení. Kvůli velkému trhu s teplotními čidly je níže uvedena tabulka **tab. 4** s aktuální dostupností na trhu. I přes široké spektrum čidel, byli celkově pro výběr zvoleny pouze čtyři. Od výrobce Dallas, Philips, Smartec a Tyco Electronics.

Tab. 4: Porovnání teplotních čidel.

Čidlo	Měřicí teplota	Rozlišení	Rozsah	Cena
DS18B20	-55 ... +85 °C	±0.5 °C	-20 ... +85 °C	62,- Kč *
LM75AD	-55 ... +125 °C	±2 °C	-25 ... +100 °C	16,- Kč *
SMT160	-45 ... +130 °C	±0.7 °C	-30 ... +100 °C	97,- Kč *
TSYS01	-40 ... +125 °C	±0.1 °C	-5 ... +50 °C	478,- Kč **

* Cena převzata z <http://gme.cz>

** Cena převzata z <http://cz.farnell.com>

V tabulce lze nalézt čidla, která mají různou přesnost měření, rozsah i cenu. Největší prodejní cena je za čidlo TSYS01 od firmy Tyco Electronics, které má výborné rozlišení i dostačující rozsah pro naše zařízení. Bohužel cena je dosti vysoká, a proto bylo čidlo vyřazeno jako první. Další čidlo, které bylo vyřazeno je LM75AD od firmy Philips i to přes velmi nízké náklady, je rozlišení dosti nízké a pro naše zařízení nedostačující. V posledním fázi výběru vyhrálo čidlo DS18B20 od firmy Dallas kvůli nižší ceně i většímu rozlišení, i přesto že čidlo SMT160 od firmy Smartec mělo dosti srovnatelné parametry, ale bylo o třetinu dražší.

3.1 Parametry číslicového teploměru DS18B20

Zde uvedené parametry jsou převzaty z katalogového listu od výrobce Dallas [8].

- Napájecí napětí od 3,0 V do 5,5 V.
- 1-Wire sběrnice.
- Pracovní maximální měřicí teplota od -55°C až do +125°C.
- Při rozlišení ±0.5°C pracovní měřicí teplota od -10°C do +85°C.
- Maximální doba komunikace při 12 bitovém rozlišení je 750 ms.

3.2 Popis číslicového teploměru

Tento číslicový teploměr od firmy Dallas lze použít pro sběr teploty při čtyřech různých teplotních rozlišeních. Toto rozlišení se nastavuje v 9., 10., 11. nebo 12. uživatelsky konfigurovatelném bitu. Primárně je teploměr přednastaven na 12 bitů.

Číslicového teploměru DS18B20 má 3 piny, VDD, GND a DQ. Funkce jednotlivých pinů jsou níže uvedené **tab. 5**.

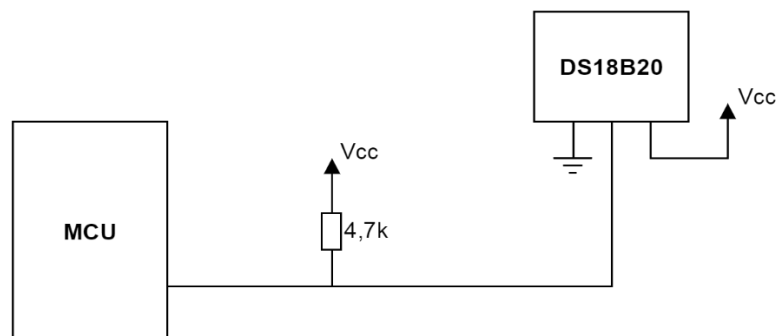
Tab. 5: Funkce jednotlivých pinů u teploměru Dallas.

Pin	Funkce pinu
V _{DD}	Tento pin slouží jako napájecí vývod, na který je možno připojit napětí v rozmezí 3 V – 5,5 V. Lze napájet dvěma způsoby, buď přes port u mikrokontroléru za parazitního napětí, nebo pomocí externího napájení.
GND	Společný zemnicí potenciál.
DQ	Vstupně-Výstupní datový vývod, sloužící ke komunikaci s mikrokontrolérem.

Každý teploměr má svoje vlastní 64 bitové sériové číslo uložené ve vnitřní paměti ROM, pomocí kterého lze na 1-Wire sběrnici rozlišovat až $(2^{64} - 1)$ připojených zařízení.

3.3 Zapojení číslicového teploměru přes externí napájení

Zapojení číslicového teploměru (viz **obr. 11**) k mikrokontroléru a připojení napětí lze realizovat dvěma různými způsoby. Každý z těchto způsobů má své výhody i nevýhody. Jako nejlépe vyhovující zapojení se zdá být, zapojení s externím napájením teplotního čidla. Datový vodič je připojen na pin u mikrokontroléru. Teplotní čidlo je napájeno přes napětí +5 V. Zároveň je připojen Pull-Up rezistor o hodnotě 4,7 k Ω mezi napájecí napětí a datový vodič, jak uvádí výrobce ve svém katalogovém listu. Nevýhoda tohoto zapojení je, že se musí k teploměru přivádět tři vodiče místo dvou, avšak pro nás tato nevýhoda není zdrcující a lze jí akceptovat.



Obr. 11: Napájení číslicového teploměru přes externí zdroj.

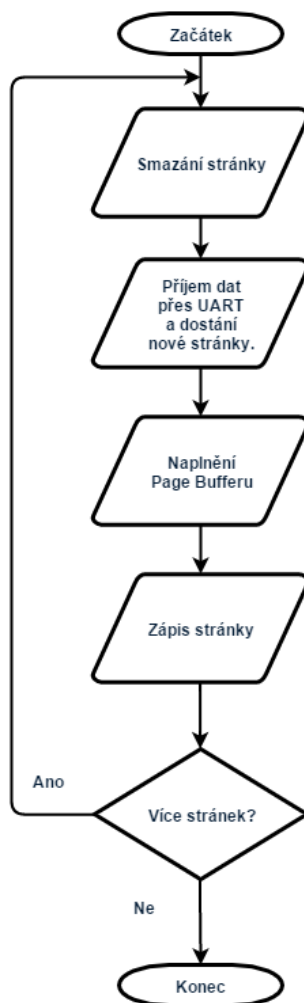
4 BOOTLOADER

V poslední době systém Bootloader vešel ve známost nejen z pohledu zkušených programátorů a vývojářů mikrokontrolérů, ale i pro užší část veřejnosti. Za tuto skutečnost může open-source Arduino založené na mikrokontroléru Atmega od výše zmíněné firmy Atmel, u kterého není potřeba žádného externího programátoru a lze zařízení naprogramovat s minimem součástek přes USB rozhraní.

Samotný systém Bootloader, též zvaný zavaděč, je část kódu v paměti mikrokontroléru. Kód nemusí být nikterak složitý a může obsahovat pouze jednoduchou změnu logických úrovní na pinech až po složitější kód obsahující přeprogramování sebe sama, což je i účel samotného systému. Popis pro implementaci tohoto systému je uveden v katalogovém listu s označením AVR109 [9].

4.1 Grafické znázornění zápisu do paměti

Pro lepší přehlednost je přidán vývojový diagram na obr. 12 pro zápis stránek při příjmu dat po UART rozhraní.

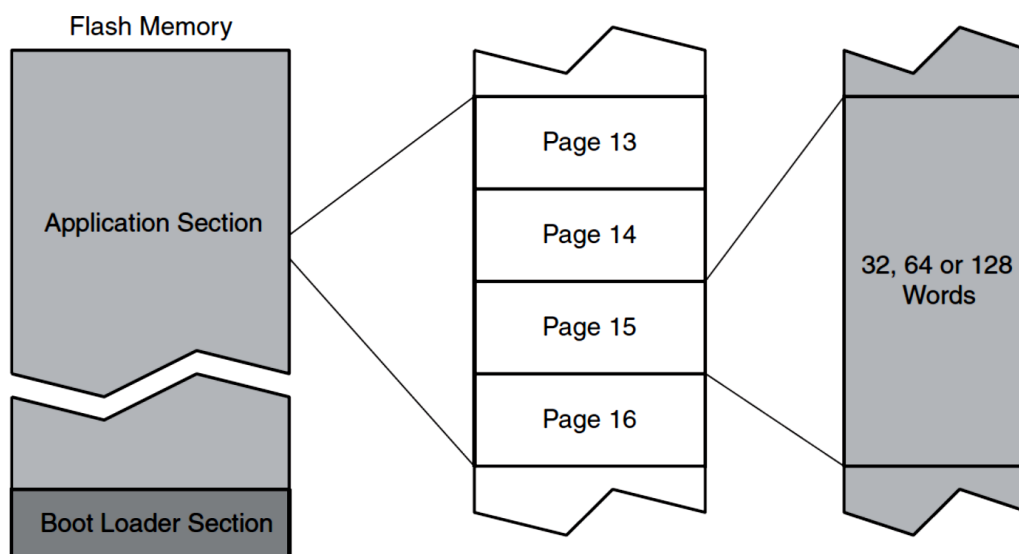


Obr. 12: Vývojový diagram pro zápis dat do paměti.

V následujících bodech jsou rozepsány jednotlivé části systému Bootloader a popis vnitřní paměti mikrokontroléru.

4.1.1 Rozdělení FLASH paměti

Pro vývoj Bootloader systému, je potřeba znát rozdělení FLASH paměti (viz **obr. 13**) u mikrokontroléru Atmega, která zahrnuje aplikační část a Bootloader část, kde aplikační část zahrnuje kód pro aplikace a Bootloader část pro přeprogramování sebe sama. FLASH paměť je rozdělena do tzv. stránek po 32, 64, 128, případně 256 slovech. Jak aplikační část, tak Bootloader část je rozdělena do těchto stránek, kde pro představu zařízení s velikostí 8kB FLASH paměti a stránkováním o velikosti 32 slov (64 bajtů) počet stránek odpovídá počtu 128 v paměti.

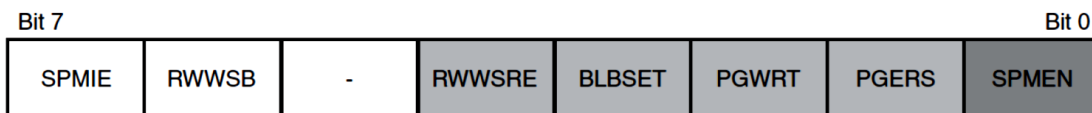


Obr. 13: Rozdělení FLASH paměti v Atmega mikrokontrolérech [10].

Velikost Bootloader části lze nastavit pomocí pojistných bitů (Fuses) s označením **BOOTSZ_x**. Bohužel velikost této části není libovolná, ale rozdělena na 4 předdefinované velikosti. Pojistkové bity mohou být změněny pouze přes sériové, případně paralelní naprogramování. Dalším pojistkovým bitem je bit **BOOTRST**, kterým se nastavuje samotné startování paměti od adresy 0, nebo od adresy začínající Bootloader částí, po resetování celého zařízení.

4.1.2 Instrukce SPM

Tato instrukce dovoluje smazat a uložit data na libovolnou stránku ve FLASH paměti, používá k tomu registr **SPMCR** případně **SPMCSR**. Rozdělení registru je zobrazeno na **obr. 14** a popsán v **tab. 6**. Dále používá registr **Z**, registr **R0** a **R1** a speciálně vyhrazenou paměť zvanou „Page Buffer“.



Obr. 14: Registr SPMCR.

Page Buffer je speciálně vyhrazená paměť o velikosti jedné stránky mikrokontroléru, do které se za běžných okolností nedá zapsat nebo z ní číst. Toho lze pouze dosáhnout přes instrukci SPM po jednotlivých slovech.

Tab. 6: Popis registru SPMCR (SPMCSR).

Bit	Název	Popis
0	SPMEN	Bit pro provolení instrukce SPM
1	PGERS	Bit pro smazání stránky.
2	PGWRT	Data z Page Bufferu jsou zapsána do paměti.
3	BLBSET	Nastavení zamykacích bitů pro Bootloader sekci.
4	RWWSRE	Indikace dostupnosti čtení RWW sekce.
5	SIFRD	(Pouze u SPMCSR registru) Signalizace čtení řádku.
6	RWWSB	Indikace dostupnosti RWW sekce.
7	SPMIE	Povolení přerušit SPM instrukce.

Při přehrávání softwarové části v řídicí, či modulové elektronice, je zapotřebí znát postup přehrávání pro každou stránku. Uvedené kroky je potřeba dodržet:

- smazání dané stránky,
- načtení nových dat (část kódu) do „Page bufferu“,
- nahrání „Page bufferu“ do paměti.

4.1.3 Smazání stránky

Celá FLASH paměť se mění stránka po stránce, avšak před zápisem nových dat do jednotlivých stránek. Za použití registru Z se vybere stránka, která se má smazat. Registr Z je pak dále rozdělen na Bajt na stránce pro smazání a na spodní bity vybraného Bajtu stránky, které jsou ignorovány. Například při zařazení se stránkou po 32 slovech (64 Bajtů), spodních 6 bitů v registru Z je ignorováno.

Pro smazání stránky, je potřeba nastavit PGERS a SP MEN bity v registru SPMCR a vykonat instrukci SPM. Všechny stránky musí být před samotným zápisem smazány.

4.1.4 Načtení dat do Page Bufferu

K zápisu nových dat do stránky, musíme nejprve naplnit registr R0 (Spodní Bajt) a R1 (Horní Bajt), ze kterého se poté zapíšou jako slovo do Page Bufferu. Zároveň v registru Z musí být správný ukazatel na danou adresu slova. Pro zapsání dat z registru R0 a R1 do Page Bufferu musíme nastavit bit SP MEN z registru SPMCR do logické jedničky a vykonáme instrukci SPM po dobu 4 cyklů.

4.1.5 Zápis do paměti

Poté co jsou v Page Bufferu načteny nová data, musí být zapsána do FLASH paměti. Toho docílíme nastavením Z registru stejným postupem jako u smazání dat a následným nastavením bitů PGWRT a SP MEN v registru SPMCR do logické jedničky a vykonáme instrukci SPM po dobu 4 cyklů.

4.1.6 Příznak zaneprázdnění RWW sekce

Při operaci smazání či zápisu dat do RWW sekce se hardwarově nastaví příznak RWWSB v registru SPMCR, který indikuje nepřístupnost dané sekce. Tento příznak by se měl vymazat softwarově po dokončení instrukce SPM. To je nastaveno bity RWWSRE a SP MEN do logické jedničky v registru SPMCR, následným vykonáním instrukce SPM po dobu 4 cyklů. Při uzamčení RWW sekce, při jakýmkoliv pokusu o čtení dat z libovolné adresy, se nám vrátí hodnota 0xFFFF.

5 BOOTLOADER-MASTER

Pro vývoj firmwaru Bootloaderu bylo zapotřebí nastudovat katalogový list pro použitý Mikrokontrolér Atmega644P a funkce v základních knihovnách z vývojového prostředí Atmel studio. Knihovna pro vývoj Bootloaderu byla využita s označením boot.h [11], která obsahuje makra naplnění paměti *Page Buffer* i samotnému zápisu stránky do FLASH paměti v aplikační části. Díky těmto knihovnám není nutné vyvíjet vlastní funkce v assembler kódu.

I přes pomocnou knihovnu bylo zapotřebí vytvořit makra pro usnadnění práce a případné využití firmwaru, pouze z drobnými úpravami, pro elektroniku s jiným typem mikrokontroléru, kde se všechna nastavení provedou pouze v hlavičce souboru, bez zásahu do samotného kódu s označením Bootloader-Master.

Základní makra pro definování jsou uvedeny níže:

- **BOOTLOADER**
 - Jedná se o definování velikosti Bootloader části, v datovém typu Word, která se nastavuje přes pojistkové bity při sériovém programování mikrokontroléru.
 - Dle současných zvolených pojistných bitů a pro tento konkrétní Mikrokontrolér je hodnota 4096.
- **FLASH**
 - Makro definující celkovou velikost FLASH paměti v bajtech jako číslo.
 - Pro tento konkrétní typ je hodnota 65536.
- **BOOT_WORDS**
 - Makro vypočítané pro velikost stránky v datovém typu Word.
 - Pro tento konkrétní typ je hodnota 128.
- **BOOT_BYTES**
 - Pomocné makro pro pozdější výpočty o velikosti stránky dle typu mikrokontroléru.
 - Pro tento konkrétní typ je hodnota 256.
- **PAGE_SIZE**
 - Počet stránek v mikrokontroléru vypočítané z velikosti FLASH paměti a velikosti použité Bootloader části
 - Pro tento konkrétní typ je hodnota 256.
- **BOOT_SIZE**
 - Výpočet velikosti, kolik zabírá stránek Bootloader část z FLASH paměti.
 - Pro tento konkrétní typ je hodnota 32.
- **APP_SIZE**
 - Počet stránek zabírající Aplikační části ve FLASH paměti.
 - Pro tento konkrétní typ je hodnota 224.

Výše popsaná makra jsou pouze základní výpočty potřebné pro další výpočty hraničních adres aplikační a bootloader části. Při změně mikrokontroléru je nutné změnit pouze velikost makra BOOTLOADER a FLASH, na kterých jsou závislé ostatní níže uvedené i neuvedené makra v hlavičce souboru.

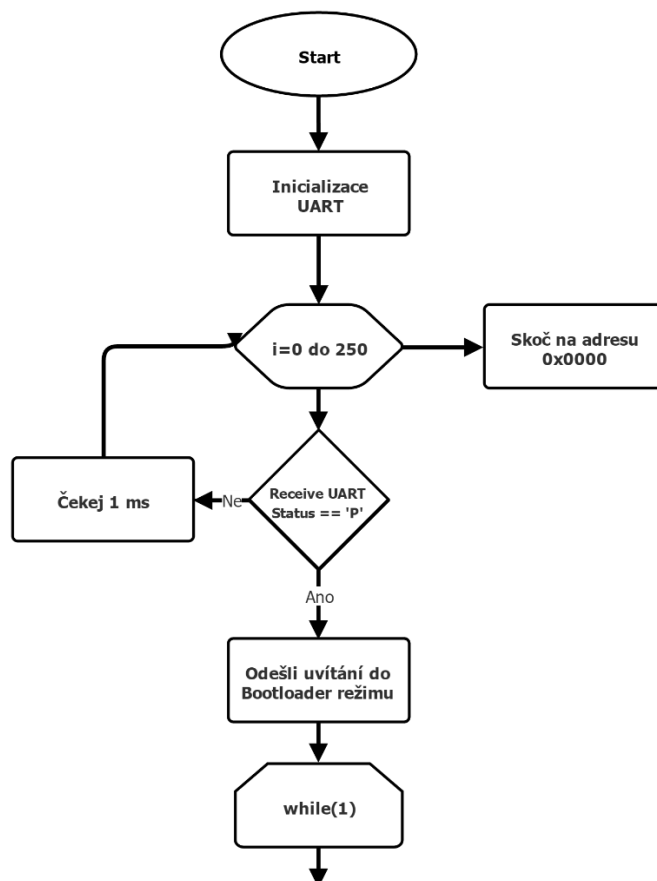
5.1 Komunikace s počítačem

Se zvolenou komunikací s počítačem přes UART rozhraní bylo nutné definovat velikost datového přenosu 1 bajtu, kde byl zvolen přenos s 8 datovými bity, 1 stop bitem s asynchronním přenosem a bez parity. Samotná rychlost komunikace byla zvolena 115200 Bd a díky externímu přesnému krystalu o velikosti 14,7456 MHz lze při takto vyšších rychlostech zajistit nulovou chybovost přenosu. Při vývoji firmwaru s využitím UART rozhraní bylo potřeba navrhnout samotnou logiku Bootloaderu, která je popsána v následujícím textu. Program pro Bootloader-Master je psán pro Mikrokontrolér Atmega644P, kde všechny hodnoty uvedené níže jsou patrné pro tento konkrétní typ. Pro jiné typy konektorů se můžou tyto adresy, či čísla lišit.

Po externím restartování mikrokontroléru, a správném nastavení pojistných bitů, startuje mikrokontrolér od adresy 0x7000, kde je implementován kód Bootloaderu, ve kterém při prvních 250 ms kontroluje příjem datové linky. Pokud v tento definovaný čas po restartu uživatel nepošle příkaz pro zahájení Bootloader režimu, automaticky skočí na adresu 0x0000, kde je, i nemusí být, aplikační část a tudíž samotný firmware aplikace.

5.2 Inicializační část

Pro rozsáhlost vývojového diagramu u programu Bootloader-Master je nutné rozdělit na dvě části, inicializační část a samotný stavový automat. V inicializační části, která je zobrazena do vývojového diagramu na **obr. 15**, se zakáže přerušování, které se nevyužívá, a nastaví se rychlost UART rozhraní. Jako další krok se čeká po dobu 250 ms na příjem znaku ‚P‘, kterým se zapne Bootloader režim. Pokud po uplynutí této doby přijde jiný nebo žádný znak, program skočí do aplikační části.



Obr. 15: Inicializační část Bootloader-Master.

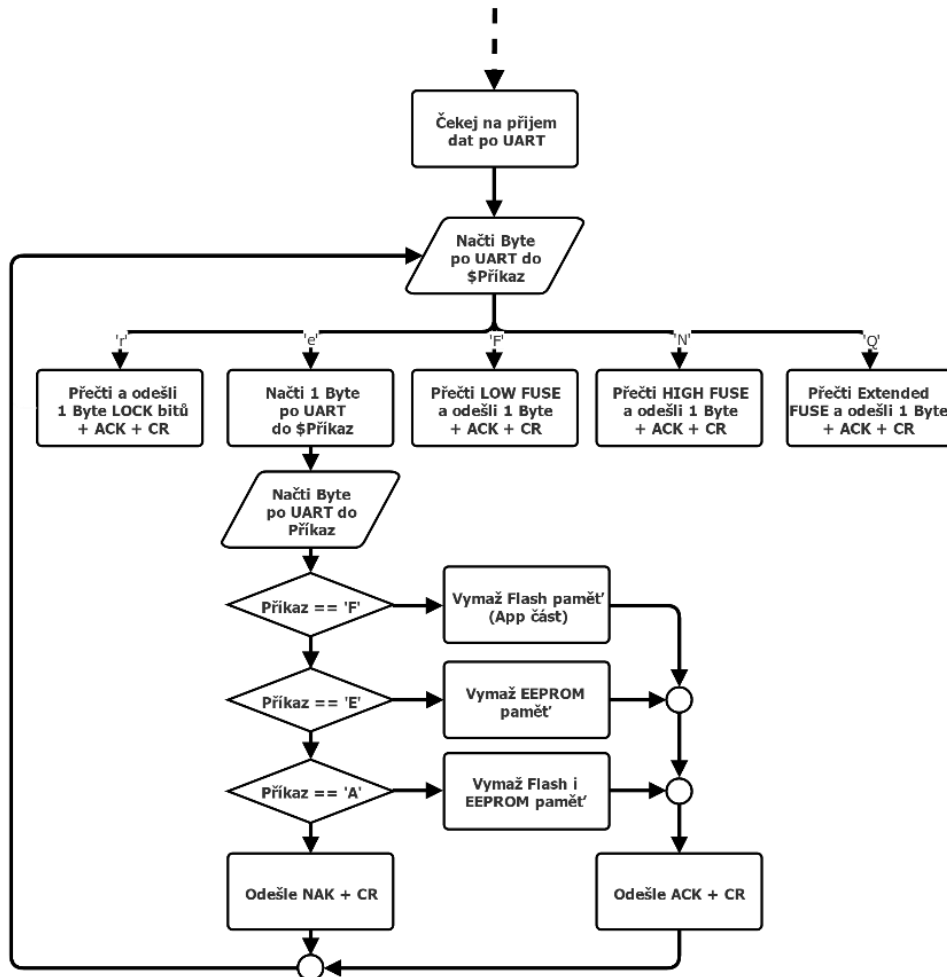
5.3 Stavový automat

Pokud uživatel aktivuje Bootloader režim, program dále čeká na příjem dat o velikosti 1 bajtu po UART rozhraní a následně analyzuje stavovým automatem a vykoná příslušný příkaz. Stavový automat pro Bootloader-Master je rozdělen do několika samostatných obrázků, kvůli přehlednosti.

Seznam příkazů z první části vývojového diagramu vyobrazených na obr. 16:

- ‚r‘ – Přečtení LOCK bitů
 - Příkaz odpoví 1 bajt dat + ACK + CR
- ‚F‘ – Přečtení LOW FUSE bitů
 - Příkaz odpoví 1 bajt dat + ACK + CR
- ‚N‘ – Přečtení HIGH FUSE bitů
 - Příkaz odpoví 1 bajt dat + ACK + CR
- ‚Q‘ – Přečtení EXTENDED FUSE bitů
 - Příkaz odpoví 1 bajt dat + ACK + CR
- ‚e‘ – Vymazání paměti
 - Po příjmu příkaz čeká na 1 bajt pro rozhodovací úroveň, která z paměti má být vymazána.
 - ‚F‘ – Příkaz vymaže FLASH paměť (Aplikační část).

- ‚E‘ – Příkaz vymaže EEPROM paměť.
 - Kvůli malému přepisu paměti je zavedena kontrola dat na konkrétní adrese. Lze tedy zajistit vymazání pouze dat, která jsou jiná než 0xFF (Defaultní hodnota).
- ‚A‘ – Příkaz pro zkombinování obou předchozích příkazů.

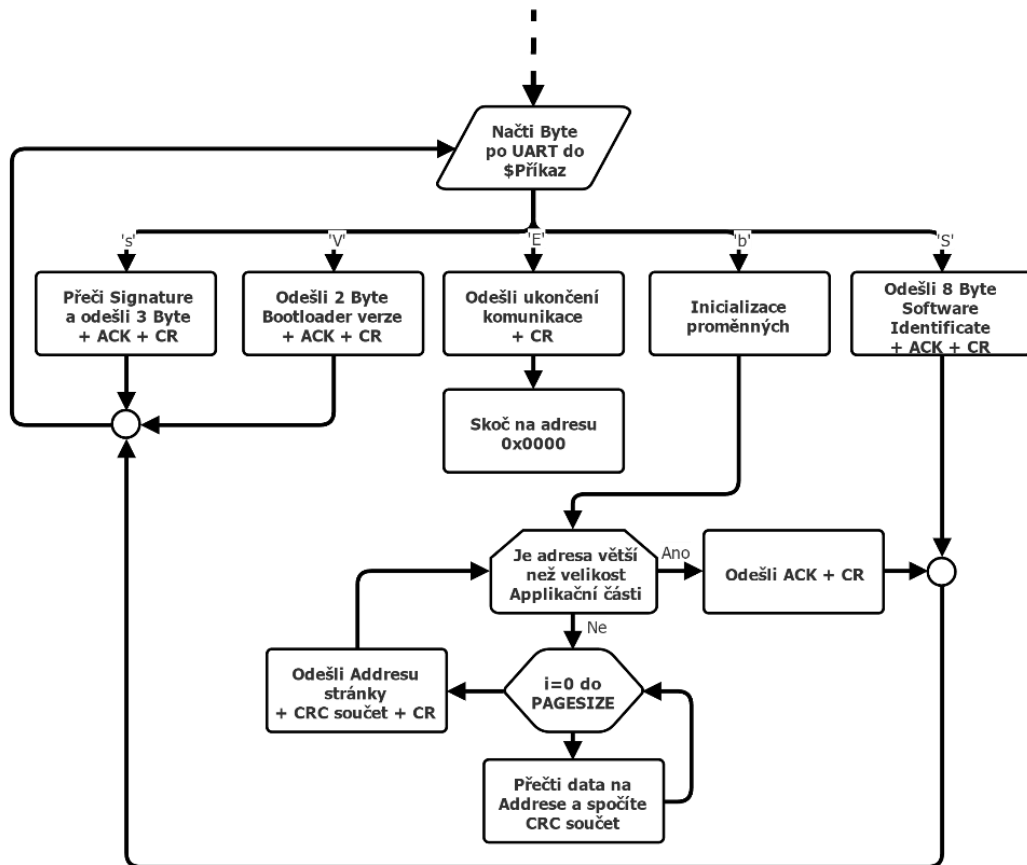


Obr. 16: Stavový automat první část.

Seznam příkazů z druhé části vývojového diagramu vyobrazených na obr. 17

- ‚s‘ – Přečtení Signature Byte
 - Příkaz vrátí podpis mikrokontroléru o velikosti 3 bajtů + ACK + CR
- ‚V‘ – Přečtení verze Bootloaderu
 - Příkaz vrátí aktuální verzi Bootloaderu o velikosti 2 bajtů + ACK + CR
- ‚E‘ – Ukončení režimu Bootloaderu
 - Příkaz vrátí CR bajt a skočí na adresu 0x0000
- ‚S‘ – Přečtení Softwarového identifikátoru
 - Příkaz vrátí aktuální verzi firmwaru o velikosti 8 bajtů + ACK + CR
- ‚b‘ – Verifikace paměti FLASH
 - Příkaz pro kontrolu správného nahrání firmwaru do FLASH paměti

- Nejprve se provede inicializace proměnných Adresa, Remainder (Výpočet CRC součtu). Samotná funkce obsahuje while cyklus ve kterém se kontroluje aktuální adresa, jestli není větší jak poslední adresa v aplikační části. Po splnění této podmínky nastává for cyklus čtení dat na adrese a kontrolní CRC součet o velikosti stránky page bufferu. V odpovědi pro přehlednost je uživateli nejprve poslána adresa aktuální součtu stránky, následně odeslán její CRC součet a CR bajt. V dalším kroku se zvětší adresa o velikost stránky, vynuluje se kontrolní součet a opět se kontroluje přetečení adresy přes aplikační část. Po výpočtu CRC součtu poslední stránky v aplikační paměti se odešle odpověď dle výše uvedeného popisu a nastává opuštění while cyklu a odeslání ukončení verifikace paměti ve tvaru ACK + CR bajt.
- Příklad odpovědi od verifikace paměti 0x0000 až po adresu 0xD000 (Poslední adresa stránky v aplikační paměti) a CRC součtem na každé stránce stejné 0xAA je uveden v následující **Chyba! Nenalezen zdroj odkazů.**

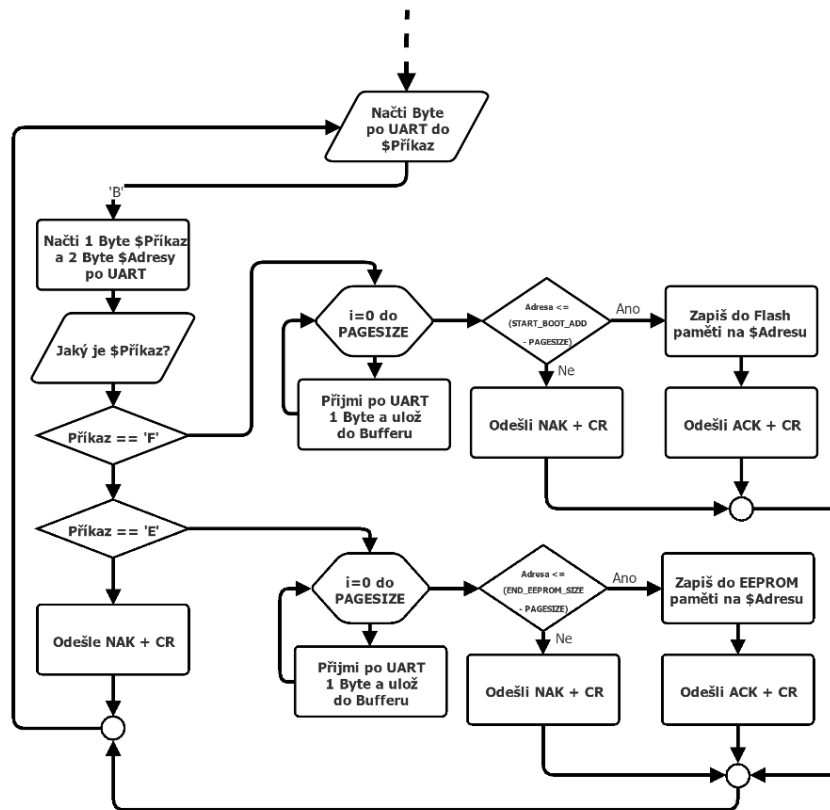


Obr. 17: Stavový automat druhá část.

Tab. 7: Tabulka příjmu verifikace dat.

Adresa 0x0000	Adresa 0x0100	...	Adresa 0xCF00	Adresa 0xD000	Konec
00 00 AA 0D	01 00 AA 0D	...	CF 00 AA 0D	D0 00 AA 0D	06 0D

- ‚B‘ – Zápis do paměti
 - Po příjmu příkazu program čeká na 1 bajt rozhodovací úrovně, do které paměti má být zapisováno a 2 bajty adresy stránky, do které se aktuálně má zapisovat. Vývojový diagram je zobrazen na **obr. 18**.
- ‚B‘ + ‚F‘ – Příkaz pro zápis do FLASH paměti
 - Po výběru příkazu pro zápis do FLASH paměti, program čeká na příchozí data, které se postupně ukládají do interního bufferu o velikosti zapisované stránky. Kontrola adresy po příjmu dat zajišťuje zápis pouze do aplikační části. Při správné adrese se začnou plnit data do Page Bufferu mikrokontroléru z interního bufferu přes for cyklus a makro `boot_page_fill()`. V tomto kroku jsou data nahrána v Page Bufferu a přes makro `boot_page_write()` s parametrem zápisu do adresy se přesunou data na konkrétní adresu stránky. Posledním příkazem `Boot_spm_busy_wait()`, zajistíme čekání na dokončení přesunu dat.
 - Po kompletním zápisu dat se odešle odpověď ve tvaru ACK + CR bajt.
 - V případě špatné adresy se odešle odpověď ve tvaru NAK + CR bajt.
- ‚B‘ + ‚E‘ – Příkaz pro zápis do EEPROM paměti
 - Po výběru příkazu pro zápis do EEPROM paměti, program čeká na příchozí data, které se postupně ukládají do interního bufferu o velikosti zapisované stránky. For cyklus zajistí zápis dat z interního bufferu na konkrétní adresu. Díky makru `eeprom_update_byte()` se přepíší pouze jiná data, takto lze zajistit vyšší životnost paměti.
 - Po kompletním zápisu dat se odešle odpověď ve tvaru ACK + CR bajt.
 - V případě špatné adresy se odešle odpověď ve tvaru NAK + CR bajt.



Obr. 18: Vývojový diagram pro zápis do FLASH a EEPROM paměti.

- ,g' – Čtení z paměti
 - Po příjmu příkazu program čeká na 1 bajt rozhodovací úrovně, ze které paměti mají být data čtena, 2 bajty adresy stránky a 1 bajt, který určuje, kdy má být přenos ukončen (proměnná End).
- ,g' + ,F' – Příkaz pro čtení z FLASH paměti
 - Uživatel u čtení z FLASH paměti si může vybrat ze dvou stavů čtení. U prvního stavu lze vypsát data pouze z jedné konkrétní stránky adresy a u druhého lze vypsát data od konkrétní adresy stránky až po konec samotné aplikační části. Vývojový diagram je zobrazen na **obr. 19**.
 - Odpověď je zformulována do následujícího tvaru: Odeslání konkrétní adresy stránky + X bajtů dat o velikosti stránky + CR. Pokud je vybrán první stav, je čtení ukončeno a odeslán ACK + CR bajt. Při výběru druhého stavu se adresa posune o velikost stránky a čtení, s následným odesláním uživateli, se opakuje do konce aplikační části, kde je čtení ukončeno a odeslán ACK + CR bajt.
 - Příklad odpovědi pro první stav od adresy 0x0000, kde přenos končí po jedné stránce je uveden v **tab. 8**.

Tab. 8: Tabulka výpisu paměti z jedné stránky.

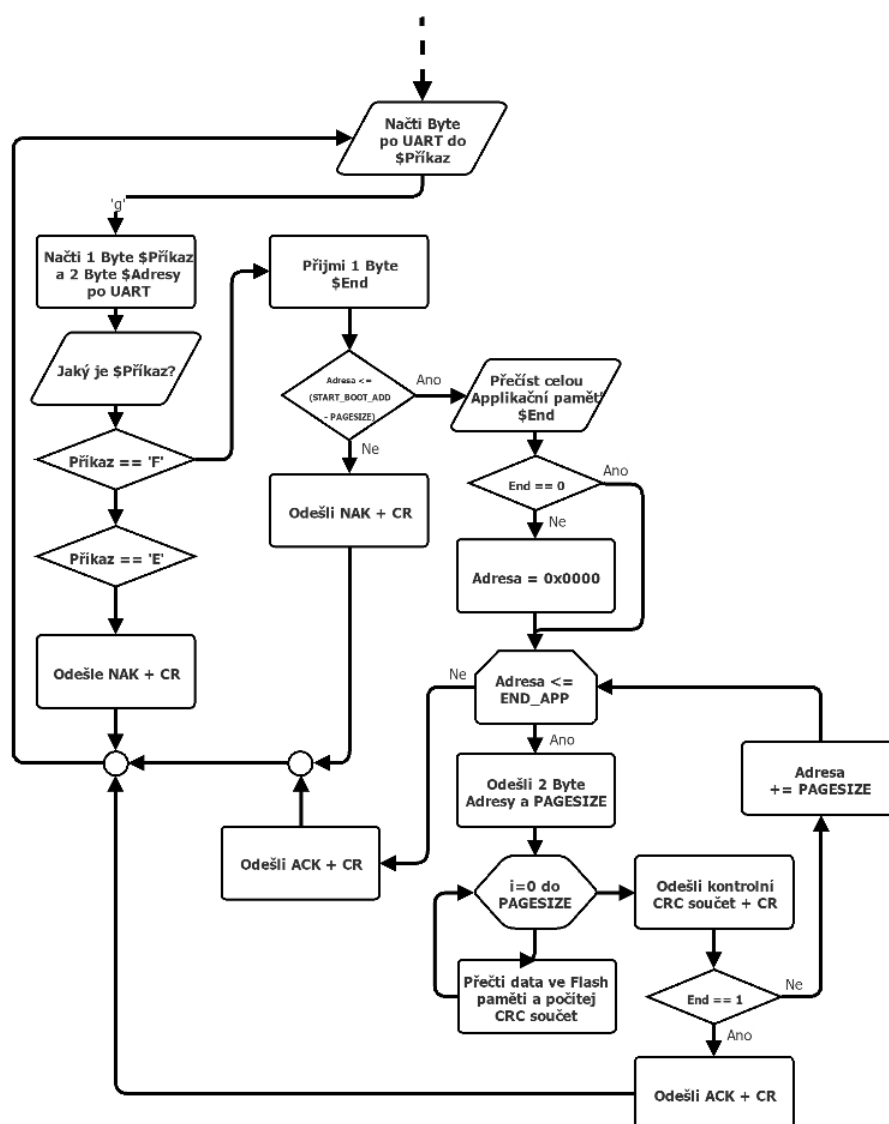
Adresa	Data	CR	ACK	CR
00 00	256 bajtů	0D	06	0D

- A příklad odpovědi uvedený v **tab. 9** pro druhý stav od adresy 0x000 kde přenos končí až koncem aplikační části (Poslední adresa stránky v aplikační paměti je 0xD000).

Tab. 9: Výpis celé paměti.

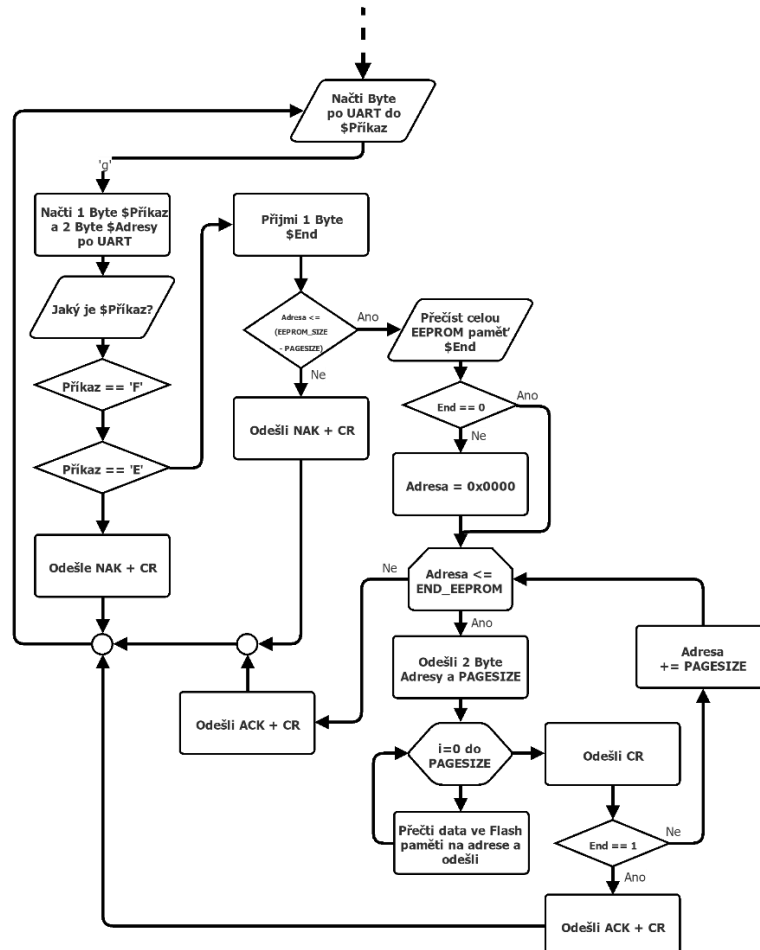
První stránka	Druhá stránka	Předposlední stránka	Poslední stránka	Konec
00 00 (256 bajtů) 0D	01 00 (256 bajtů) 0D	CF 00 (256 bajtů) 0D	D0 00 (256 bajtů) 0D	06 0D

- V případě špatné adresy se odešle odpověď, pouze ve tvaru NAK + CR bajt.



Obr. 19: Stavový automat pro čtení z FLASH paměti.

- ,g‘ + ,E‘ – Příkaz pro čtení z EEPROM paměti
 - Uživatel u čtení z EEPROM paměti si může vybrat ze dvou stavů čtení. U prvního stavu lze vypsát data pouze z jedné konkrétní stránky adresy a u druhého lze vypsát data od konkrétní adresy stránky až po konec samotné aplikační části. Vývojový diagram je zobrazen na **obr. 20**.
 - Odpověď je zformulována do následujícího tvaru: Odeslání konkrétní adresy stránky + X bajtů dat o velikosti stránky + CR. Pokud je vybrán první stav, je čtení ukončeno a odeslán ACK + CR bajt. Při výběru druhého stavu se adresa posune o velikost stránky a čtení, s následným odesláním uživateli, se opakuje do konce aplikační části, kde je čtení ukončeno a odeslán ACK + CR bajt.
 - Příklad odpovědi pro první stav od adresy 0x0000, kde přenos končí po jedné stránce. Přenos je stejný u FLASH paměti dle **tab. 8**.
 - A příklad odpovědi pro druhý stav od adresy 0x000 kde přenos končí až koncem aplikační části (Poslední adresa stránky EEPROM paměti je 0x0700). Přenos je stejný u FLASH paměti dle **tab. 9**.
 - V případě špatné adresy se odešle odpověď ve tvaru NAK + CR bajt.



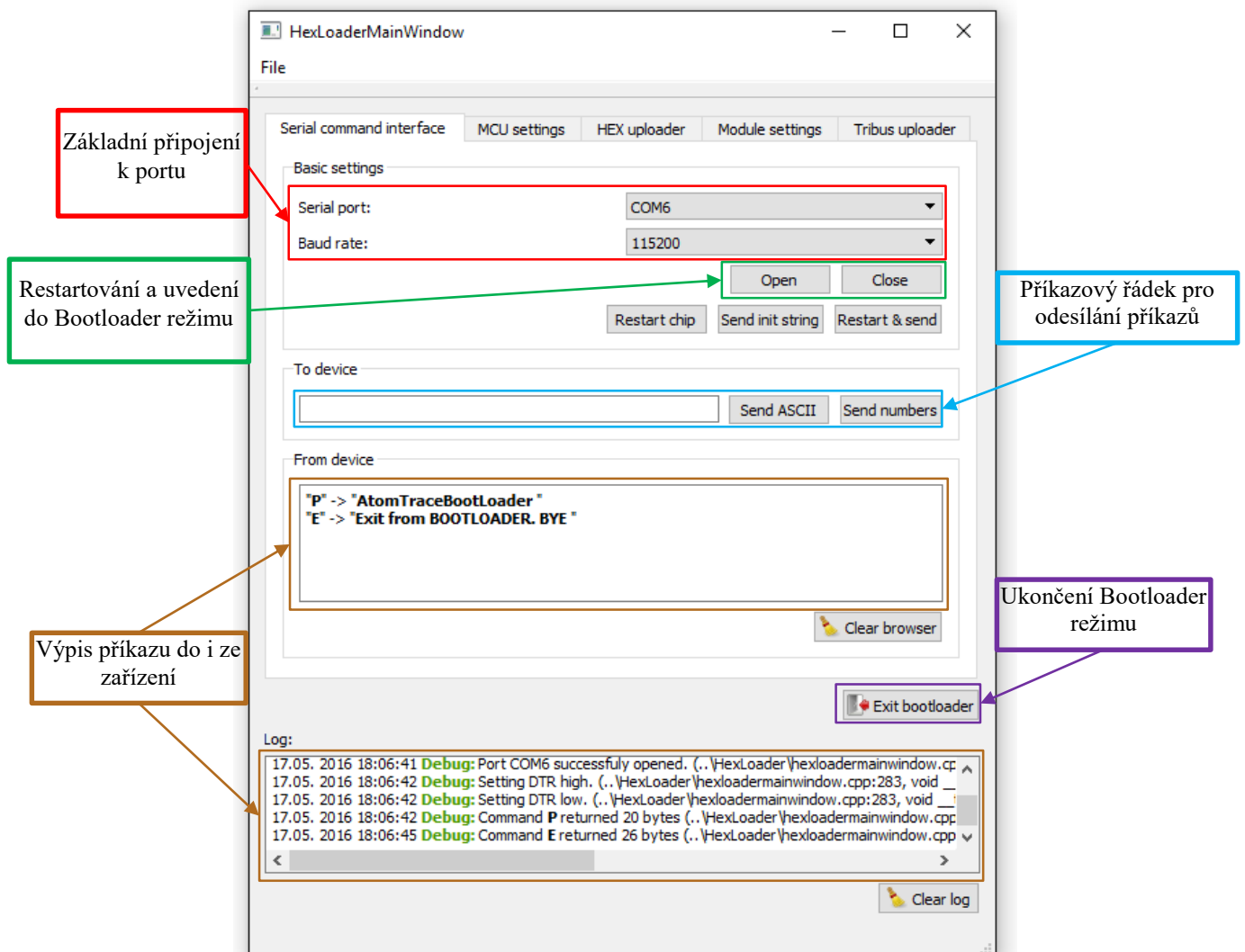
Obr. 20: Stavový automat pro čtení z EEPROM paměti.

5.4 Program HexLoaderMaster

Firmware Bootloader-Master napsaný pro Mikrokontrolér je sám o sobě bez uživatelského programu pro nahrání aplikačního firmwaru nepoužitelný. I když lze využít nějakého volně dostupného terminálu pro komunikaci (např. RealTerm), je přívětivější mít vytvořen vlastní program pro nahrávání. Pro tuto funkci bylo nutné vytvořit program HexLoaderMaster, který postup nahrávání zjednodušuje.

5.4.1 Popis programu

Program je rozdělen na 5 samostatných listů Serial command interface (viz **obr. 21**), MCU settings (viz **obr. 22**), Hex uploader (viz **obr. 23**) a dvou listů, které jsou součástí nahrávání aplikačního firmwaru do externích modulů. Program pro nahrávání aplikačního firmwaru do řídicí jednotky je nutné zapnout až po připojení řídicí desky do USB portu počítače. Při jiném postupu nelze zaručit, že program nedokáže nalézt příslušný port.



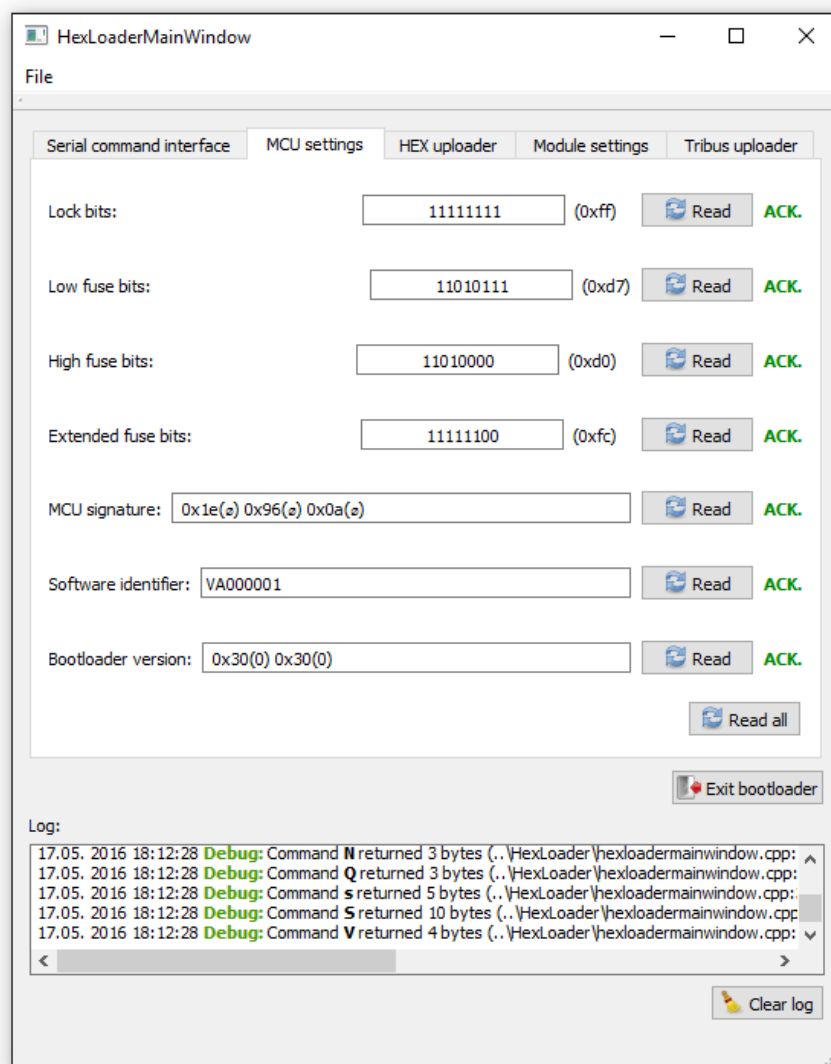
Obr. 21: Popis programu HexLoaderMaster

List serial command interface slouží k otevření portu pro řídicí modul a jeho přepnutí do Bootloader režimu. Dále obsahuje uživatelské ruční zadávání příkazů do zařízení a výpis odpovědí ze zařízení. V posledním případě ještě obsahuje tlačítko pro opuštění Bootloader režimu a kompletní výpis odesílaný do i ze zařízení.

V první řadě je potřeba vybrat sériový port, přenosovou rychlost a otevřít port. Poté zmáčknout tlačítko „*Restart & Send*“, které kombinuje resetování mikrokontroléru přes datový pin DTR u převodníku FT232RL a následnému odeslání příkazu ‚P‘ pro zaktivování Bootloader režimu, pokud se zaktivuje Bootloader režim, ve výpisu se nám zobrazí následující výpis „P“ -> „AtomTraceBootloader“. Pro testování lze využít příkazového řádku, pro odesílání jednotlivých příkazů.

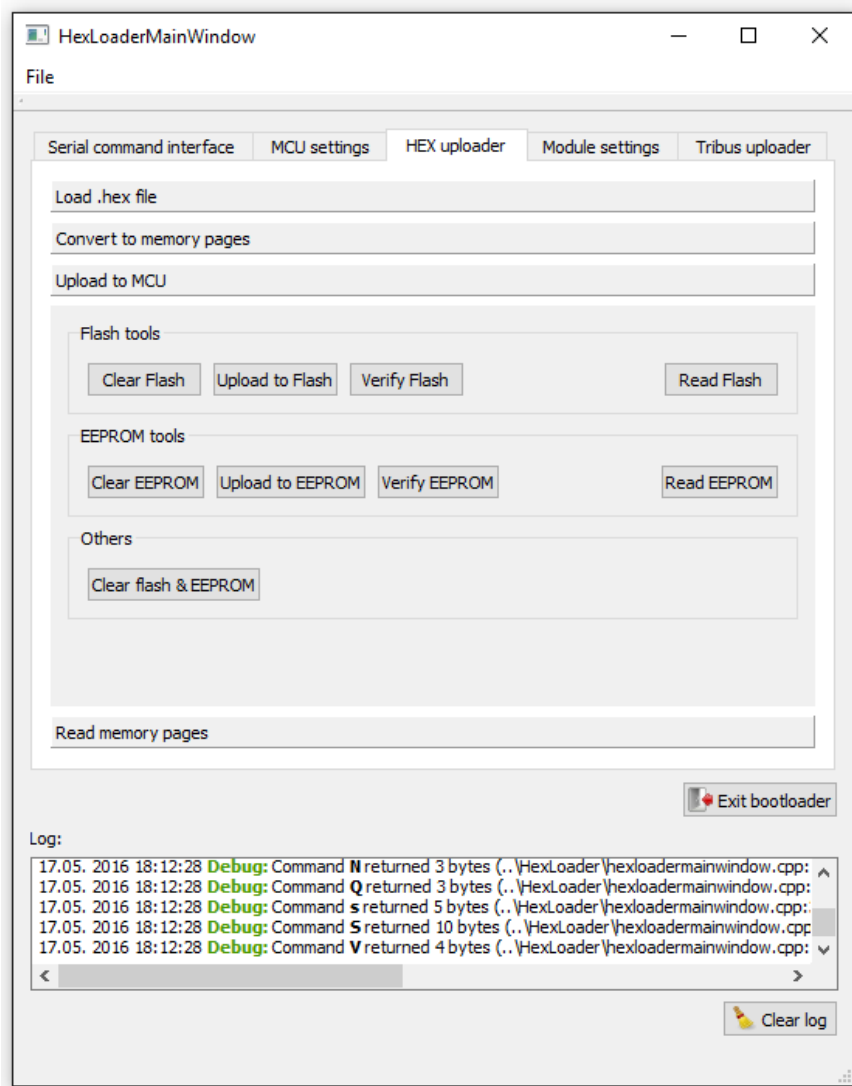
Pokud chce uživatel odejít z Bootloader režimu, stačí zmáčknout tlačítko „*Exit bootloader*“ a ve výpisu se nám zobrazí hláška „E“ -> „Exit from BOOTLOADER. BYE“.

List MCU settings slouží pro výpis informací o samotném mikrokontroléru a o verzi firmwaru. Výpis lze provést u konkrétních příkazů tlačítkem „*Read*“ a nebo přes tlačítko „*Read all*“ přečíst všechny informace.



Obr. 22: List „MCU settings“.

List Hex uploader slouží pro nahrávání a čtení FLASH a EEPROM paměti. Při otevření listu si lze povšimnout tlačítek „Clear Flash“ pro vymazání aplikační paměti, „Clear Eeprom“ pro vymazání EEPROM či jejich kombinaci. Pro nahrání aplikačního firmwaru je nutné nahrát do programu příslušný soubor s koncovkou hex, založený na standardu Intel HEX. Nahrání se provede přes „File->Open HEX“. Stejný postup je pro nahrání dat EEPROM paměti. Kontrolu dat lze provést rozkliknutím „Load .hex file“ a zkontrolování převedení na náš datový přenos o velikosti 1 stránky kliknutím na „Convert to memory pages“. Dalším krokem pro nahrání do paměti je zmáčknout tlačítko „Upload to Flash“ nebo „Upload to EEPROM“ dle nahrávané paměti. Po nahrání všech dat dojde na pozadí programu k resetu Mikrokontrolér a uvedení do Bootloader režimu. Tento úkon se provádí kvůli verifikaci a čtení paměti, neboť i když data jsou nahrána na svém místě v paměti tak i přes toto Mikrokontrolér posílá stará data.



Obr. 23: List „HEX uploader“.

5.4.2 Správný postup nahrávání firmwaru do řídicí desky

Jelikož výše uvedený program je vytvořen pro testování, byly některé funkce potřebné pouze při testování. Avšak ve finální verzi již tyto funkce nejsou potřebné pro nahrávání programu do řídicí jednotky.

Před spuštěním programu je nutné zapojit řídicí desku přes USB kabel k počítači a zapnutí elektroniky do polohy „Zapnuto“. Po zapnutí se rozsvítí displej, bez jakýchkoliv informací (Pokud není v paměti aplikační firmware). V dalším kroku lze spustit program HexLoaderMaster a otevřít příslušný port k řídicí elektronice. Stlačením tlačítka „Restart & Send“ uvést řídicí jednotku do Bootloader režimu a zkontrolovat příkazem MCU signature podpis mikrokontroléru. Smazáním FLASH paměti se zajistí neovlivnění nové verze starými daty v paměti. Nahráním firmwaru a stlačením tlačítka „Upload Flash“ se nahrají data do aplikační paměti a tlačítkem „Verify Flash“ se zkontroluje správnost nahraných dat v paměti. V posledním kroku se zmáčkne tlačítko „Exit Bootloader“ pro skočení na adresu 0x0000 do aplikační paměti.

6 BOOTLOADER-SLAVE

U firmwaru pro nahrávání do externích modulů byl návrh složitější z důvodů nutnosti nahrávání aplikačního programu přes RS485 s protokolem Trinamic, který je již implementován v aplikačním firmwaru a přidávání další sběrnice mezi jednotlivé moduly by bylo zbytečné. Tento program není implementován pouze pro jeden konkrétní modul, ale pro více modulů, kde adresa modulu je uložena v EEPROM paměti. Takto modul má vždy konkrétní adresu, kterou lze i samozřejmě softwarově změnit. Protokol Trinamic je složen z 9 bajtového paketu (viz **tab. 10**), kde standardní příkaz obsahuje adresu, příkaz, typ, motor, 4 bajty dat a CRC součet.

Tab. 10: Datový paket protokolu Trinamic.

Adresa modulu	Číslo příkazu	Číslo typu	Číslo motoru	4 bajty dat				Kontrolní součet
Adresa	Příkaz	Typ	Motor	Data3	Data2	Data1	Data0	CRC

Jelikož všechny moduly připojené ke sběrnici RS485 dostávají všechny data, musí mít každý modul svoji specifickou adresu, kterou se určí, jestli data jsou určena pro konkrétní modul, např. ovládání osvětlení, nebo pro jiný modul. Pro příklad máme na sběrnici připojený řídicí modul, který má adresu 0 a je připojen k počítači. Dále je na sběrnici připojen modul pro ovládání osvětlení s adresou 1 a tlakový systém s adresou 2. Řídicí modul přešlává všechny pakety, na sběrnici RS485, kromě paketů s jeho adresou 0. Pokud přijde z počítače paket s adresou 1 řídicí modul přešle celý paket na sběrnici. Tento paket dostane jak modul ovládání osvětlení, tak i tlakový systém. Ovládání osvětlení jelikož má adresu 1 zjistí, že paket je pro ně a zpracuje zprávu, pro tlakový systém, který má adresu 2 a tento paket zahodí.

Upravený paket **tab. 11**, pro nahrávání aplikačního firmwaru, musí tedy obsahovat adresu modulu, příkaz, kterým se vybere, co se má provést dle stavového automatu, adresu v paměti, která slouží pro čtení a zápis do FLASH nebo EEPROM paměti, 4 bajty dat a kontrolní součet.

Tab. 11: Datový paket příjmu pro komunikaci s Bootloader-Slave.

Adresa modulu	Číslo příkazu	Adresa v paměti		4 bajty dat				Kontrolní součet
Adresa	Příkaz	Add1	Add0	Data3	Data2	Data1	Data0	CRC

Odpověď ve formě paketu je sestavena taktéž dle protokolu Trinamic, kde se odesílá odpověď adrese, která adresa odpovídá, status, příkaz, 4 bajty dat a kontrolní součet. Adresa odpovědi je nastavená dle protokolu na hodnotu 2, adresa modulu je dána hodnotou uloženou v EEPROM paměti modulu, status příkaz je chybová nebo pozitivní odpověď na vykonaný příkaz, na který odpovídá a 4 bajty dat.

Pro naše účely odpovědi na daný příkaz potřebujeme odpovídat se změněnými hodnotami u status příkazu (dále uvedeným pouze jako Status) a 4 bajty dat (dále uvedené pouze jako jedna hodnota Value), ostatní data jsou dána konkrétním modulem. Celý paket pro sestavení je znázorněn v **tab. 12**.

Tab. 12: Odchozí datový paket pro komunikaci s řídicí jednotkou.

Odpověď Adrese	Adresa modulu	Status příkazu	Příkaz	4 bajty dat				Kontrolní součet
Adresa	Adresa	Status	Příkaz	Data3	Data2	Data1	Data0	CRC

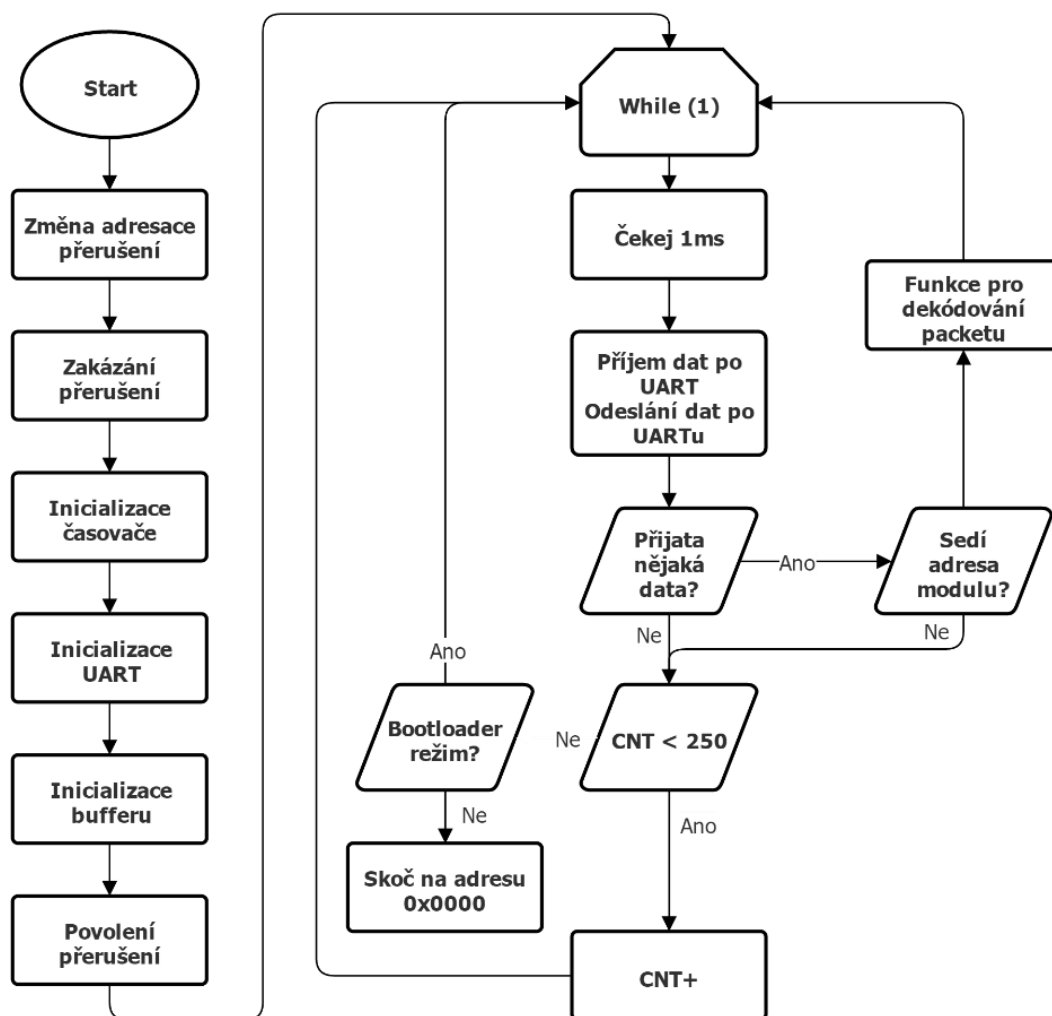
6.1 Inicializační část

Program Bootloader-Slave je podobně složen jako program pro řídicí modul. Rozložen je na inicializační část, která se spustí po hardwarovém restartování mikrokontroléru a na protokol Trinamic. Na začátku inicializace, kvůli využití přerušení pro komunikaci na sběrnici RS485, je nutné změnit uložení adres přerušení, dle katalogového listu. Pokud bychom nezměnili nastavení registrů BOOTRST a IVSEL, program by nám nepracoval správně a mohl by dokonce zamrznout. Registrem BOOTRST se nastaví adresa po resetu zařízení. Registrem IVSEL nastavíme adresu vektoru přerušení. Pro správný chod přerušení se dle katalogového listu doporučuje nastavit BOOTRST a IVSEL do logické jedničky.

Kvůli nastavení UART rozhraní a časovače se zakáže přerušení, které se povolí až po inicializacích. Inicializace časovače se nastaví na 1 ms přes režim Fast PWM, který následně slouží jako prodleva mezi pakety. Nastavení UART rozhraní se provede v režimu RS485 poloduplexní sběrnice s rychlostí uloženou v EEPROM paměti, případně s výchozím nastavením 115200 Bd, pokud se program spustil poprvé. Inicializace bufferu je vložena z důvodu úpravy nahrávání do paměti, kde se kvůli malému datovému paketu po RS485 odešlou pouze 4 bajty dat pro jednu konkrétní adresu v paměti. Celý buffer se vyplní hodnotou 0xFF a tak buffer přepisujeme pouze daty. Pokud tedy chceme do paměti nahrát poslední 1 bajt dat do nové stránky, tak nemusíme následně z počítače posílat pouze data s hodnotou 0xFF vyplnit buffer a tak zbytečně zahltit linku. V posledním kroku inicializace se provede povolení přerušení.

Po inicializaci následuje nekonečný cyklus while, ve kterém voláme funkci pro čekání programu mezi pakety 1 ms, kontrola příjmu a odeslání paketu po RS485, dekodování paketu a zpožděné vypnutí Bootloader režimu.

Pokud program zjistí příjem dat po datové lince, naplní buffer paketu a zkontroluje adresu modulu. Jestliže adresa modulu paketu se shoduje s adresou uloženou v EEPROM paměti mikrokontroléru, program dekóduje zprávu a pokračuje čítačem příznaku, jinak paket zahodí. Pokud uživatel nepošle příkaz pro zapnutí Bootloader režimu, příznak zajistí automatický skok v paměti na adresu 0x0000 do aplikační paměti po přibližné době 250 ms. Zjednodušený vývojový diagram je znázorněn v níže uvedené obrázku **obr. 24**.



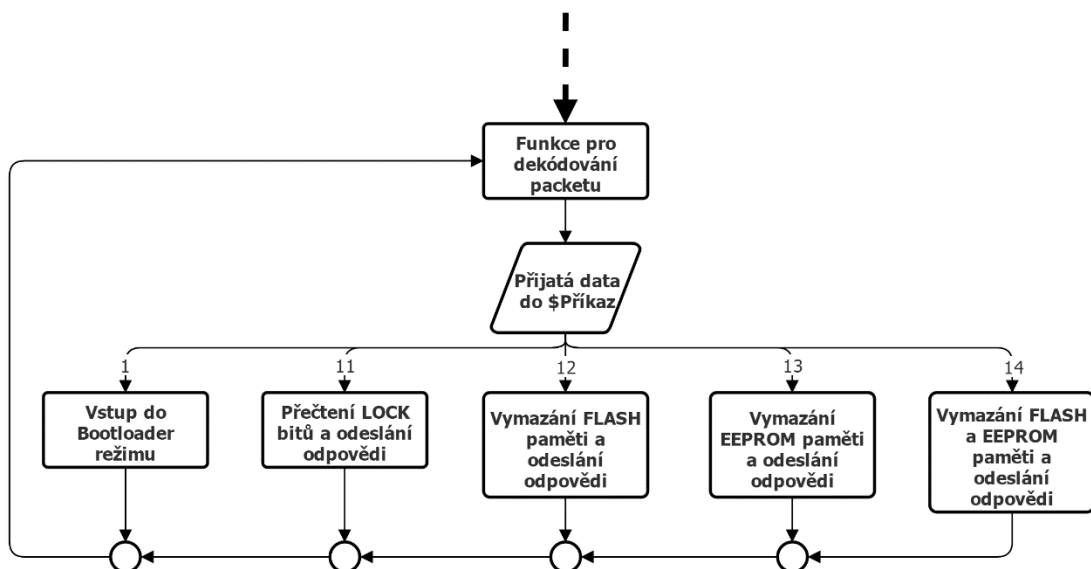
Obr. 24: Inicializace programu Bootloader-Slave.

6.2 Stavový automat

Ve funkci kontroly příjmu dat programem kontrolujeme příjem, adresu a plníme buffer paketu. Datový paket se dekoduje v oddělené funkci, kde se zjišťuje, jaký příkaz byl přijat s následným vykonáním dané funkce příkazu. U každého příkazu v níže uvedeném seznamu jsou hodnoty Status a Value, které určují odpověď na daný příkaz.

Seznam příkazů stavového automatu:

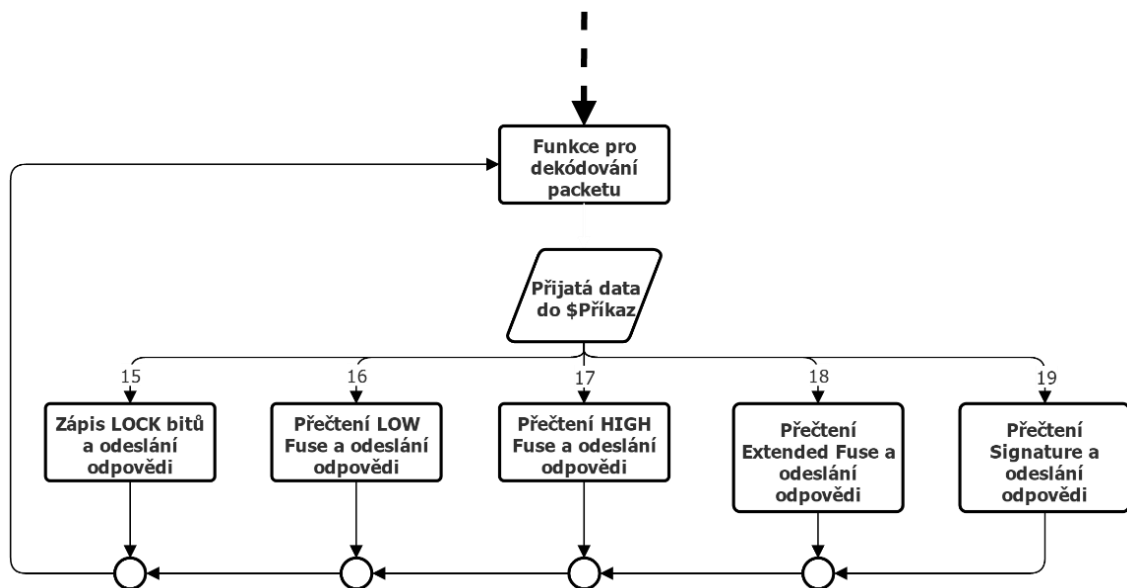
- ,1‘ – Aktivace Bootloader režimu, u kterého se anuluje příznak vyskočení do aplikační paměti.
 - Status – 100, Value – 1.
- ,11‘ – Přečtení LOCK bitů a odeslání odpovědi v následujícím tvaru.
 - Status – 100, Value – Hodnota LOCK bitů.
- ,12‘ – Vymazání FLASH paměti a odeslání odpovědi
 - Příkaz vymaže celou aplikační paměť.
 - Status – 100, Value – 12.
- ,13‘ – Vymazání EEPROM paměti a odeslání odpovědi
 - Kvůli malému přepisu paměti je zavedena kontrola dat na konkrétní adrese. Lze tedy zajistit vymazání pouze dat, která jsou jiná než 0xFF (Defaultní hodnota).
 - Status – 100, Value – 13.
- ,14‘ – Vymazání FLASH a EEPROM paměti.
 - Kombinace příkazu 12 a 13.
 - Status – 100, Value – 14.



Obr. 25: Vývojový diagram pro příkazy 1, 11, 12, 13 a 14.

- ,15‘ – Zápis LOCK bitů a odeslání odpovědi
 - Příkaz pro uzamčení mikrokontroléru při velkovýrobě.
 - Status – 100, Value – 15.

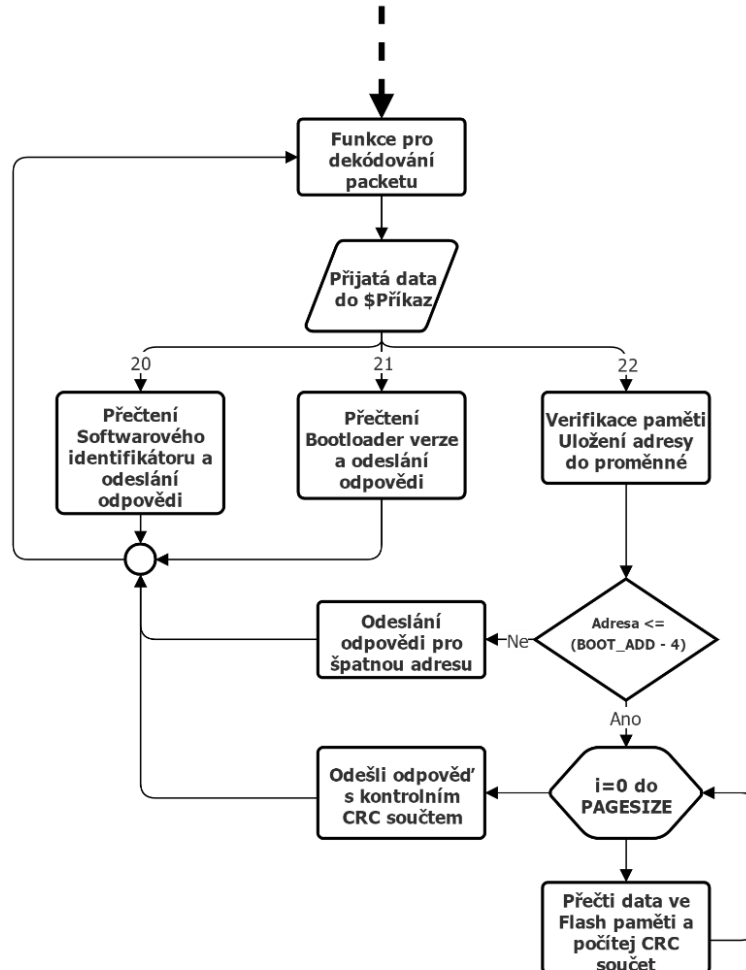
- ,16‘ – Přečtení LOW fuse bitů a odeslání odpovědi
 - Přečtení spodních pojistek mikrokontroléru.
 - Status – 100, Value – Hodnota LOW fuse bitů
- ,17‘ – Přečtení HIGH fuse bits a odeslání odpovědi
 - Přečtení horních pojistek mikrokontroléru.
 - Status – 100, Value – Hodnota HIGH fuse bitů
- ,18‘ – Přečtení EXTENDED fuse bits a odeslání odpovědi
 - Přečtení rozšířených pojistek mikrokontroléru.
 - Status – 100, Value – Hodnota EXTENDED fuse bitů
 - V případě, že Mikrokontrolér neobsahuje rozšířené pojistky, Status hodnota je 102.
- ,19‘ – Přečtení Signature Byte
 - Přečtení podpisu mikrokontroléru (typ).
 - Status – 100, Value – Hodnota podpisu mikrokontroléru



Obr. 26: Vývojový diagram pro příkazy 15 až 19.

- ,20‘ – Přečtení softwarového identifikátoru a odeslání odpovědi.
 - Status - 100, Value – Hodnota softwarového identifikátoru.
- ,21‘ – Přečtení Bootloader verze v mikrokontroléru
 - Status – 100, Value - Hodnota Bootloader verze.
- ,22‘ – Verifikace FLASH paměti.
 - U příkaz pro verifikaci se kontroluje aktuální adresa, jestli je menší než konec aplikační části snižený o velikost stránky.
 - Při nesplnění této podmínky se uživateli vrací odpověď: Status – 102, Value – Přijátá adresa paměti.

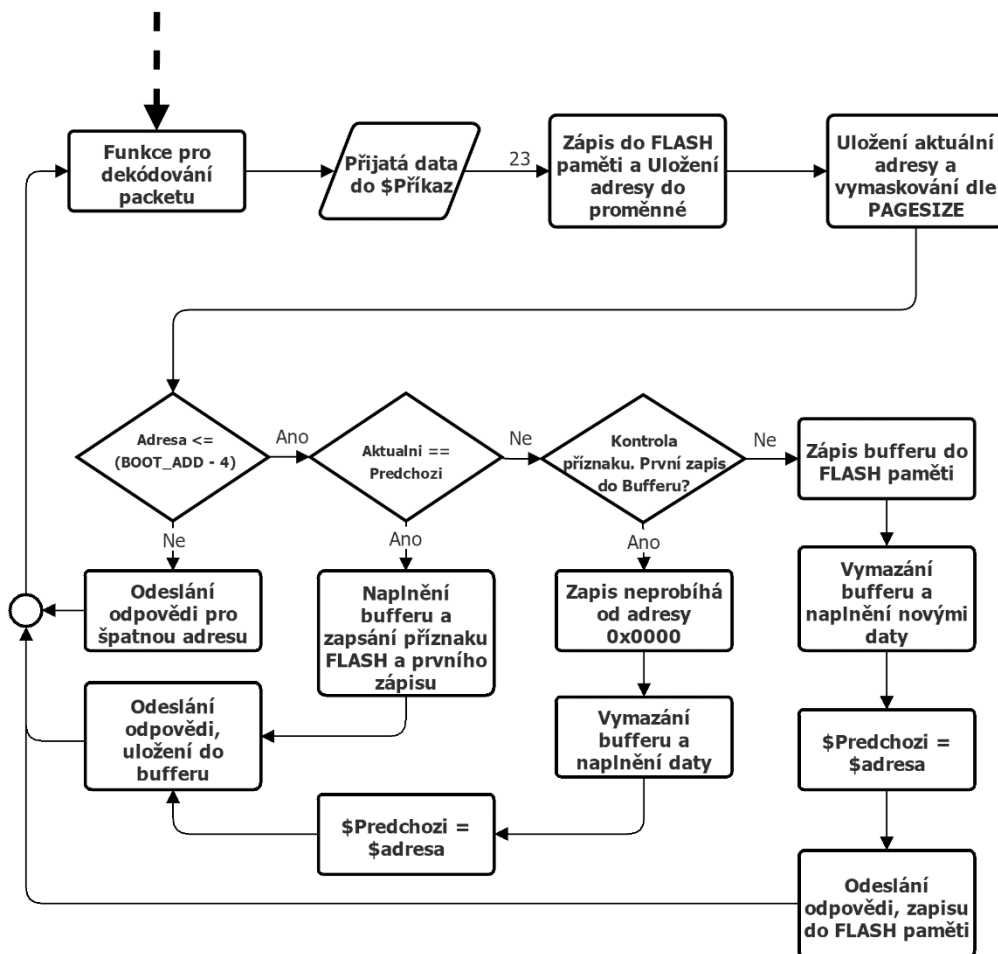
- Pokud adresa splňuje podmínku, funkce vypočítá kontrolní CRC součet z celé stránky paměti.
- Příkaz vrací odpověď s parametry Status – 100, Value je hodnota kontrolního CRC součtu pro celou stránku adresy.



Obr. 27: Vývojový diagram pro příkazy 20 až 22.

- ,23‘ – Příkaz pro zápis do FLASH paměti vyobrazen na **obr. 28**.
 - Zápis do FLASH paměti se musí provádět přes interní buffer o velikosti stránky, do kterého se zapisují přijatá data.
 - Nejprve se uloží aktuální adresa, která se vymaskuje dle velikosti stránky. Podmínka zápisu zajistí ochranu proti zápisu mimo aplikační paměť s chybnou odpovědí ve tvaru Status – 102, Value jsou přijatá adresa paměti.
 - V dalším kroku se kontroluje aktuální adresa, jestli je stejná jako předchozí adresa. Inicializace předchozí adresy je nastavena na 0x0000. Interní buffer se naplní, zapíší se příznaky zápisu do FLASH paměti, a odpoví se s parametry Status – 100, Value – 23.

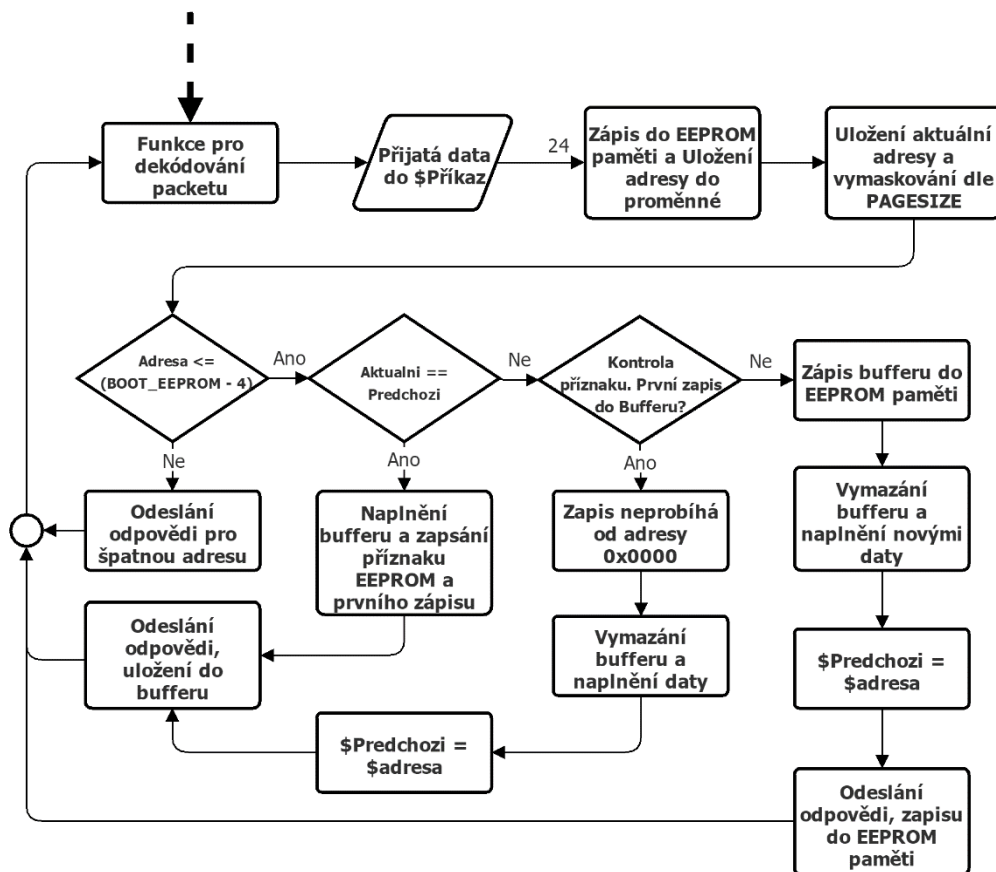
- Pokud se aktuální adresa neshoduje s předchozí, funkce zkontroluje příznak předchozích zápisů do FLASH. V případě zápisu se zavolá funkce pro zápis do page bufferu a následnému zápisu do samotné FLASH paměti. Interní buffer se naplní defaultními daty a naplní se aktuálními přijatými daty. V posledním kroku se předchozí adresa přepíše aktuální adresou a odešle se odpověď s parametry Status - 100 a Value – Hodnota zapisované adresy stránky.
- V případě prvního zápisu na jinou adresu než 0x0000 se inicializuje interní buffer a naplní se přijatými daty. V posledním kroku se předchozí adresa přepíše aktuální adresou a odešle se odpověď s parametry Status – 100 a Value - 23.



Obr. 28: Vývojový diagram pro zápis do FLASH paměti.

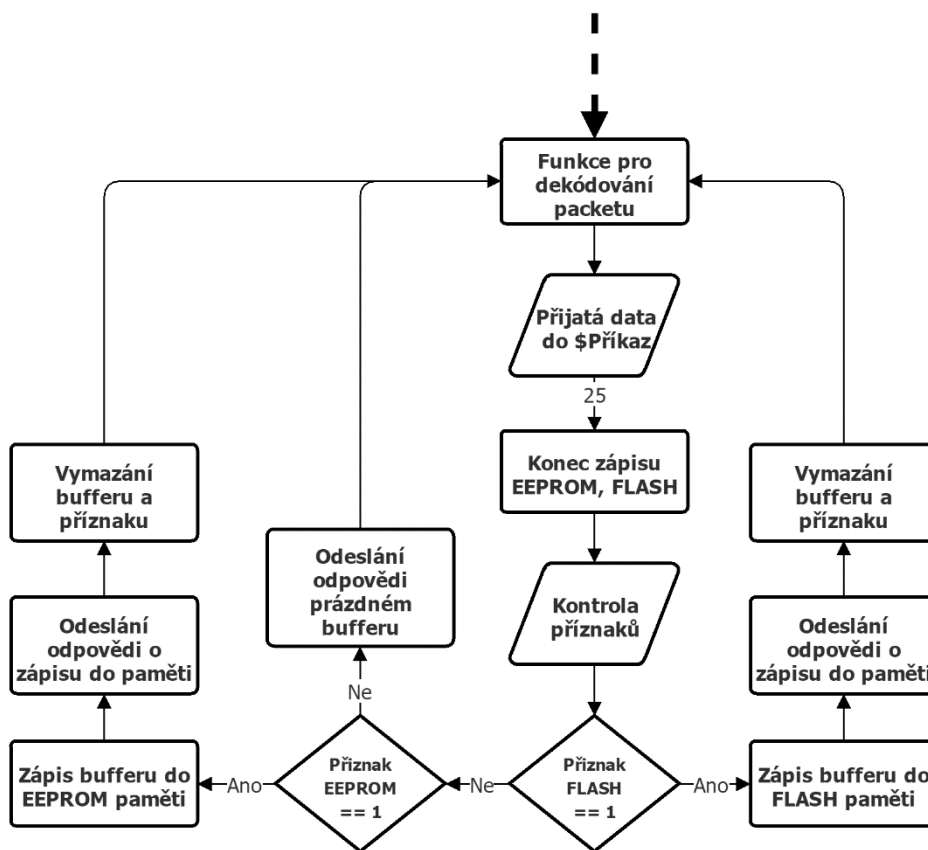
- ,24‘ – Příkaz pro zápis do EEPROM paměti vyobrazen na **obr. 29**.
 - Zápis do EEPROM paměti se musí provádět přes interní buffer o velikosti stránky, do kterého se zapisují přijatá data.
 - Nejprve se uloží aktuální adresa, která se vymaskuje dle velikosti stránky. Podmínka zápisu zajistí ochranu proti zápisu mimo velikost EEPROM paměti s chybnou odpovědí ve tvaru Status – 102, Value - Přijátá adresa paměti.

- V dalším kroku se kontroluje aktuální adresa, jestli je stejná jako předchozí adresa. Inicializace předchozí adresy je nastavena na 0x0000. Interní buffer se naplní, zapíšu se příznaky zápisu do EEPROM paměti, a odpoví se s parametry Status – 100, Value - 24.
- Pokud se aktuální adresa neshoduje s předchozí, funkce zkontroluje příznak předchozích zápisů do EEPROM. V případě zápisu se zavolá funkce pro zápis EEPROM paměti. Interní buffer se inicializuje defaultními hodnotami a naplní se aktuálně přijatými daty. V posledním kroku se předchozí adresa přepíše aktuální adresou a odešle se odpověď s parametry Status – 100 a Value je hodnota zapisované adresy stránky.
- V případě prvního zápisu na jinou adresu než 0x0000 se inicializuje interní buffer a naplní se přijatými daty. V posledním kroku se předchozí adresa přepíše aktuální adresou a odešle se odpověď s parametry: Status – 100 a Value - 24.



Obr. 29: Vývojový diagram pro zápis do EEPROM paměti.

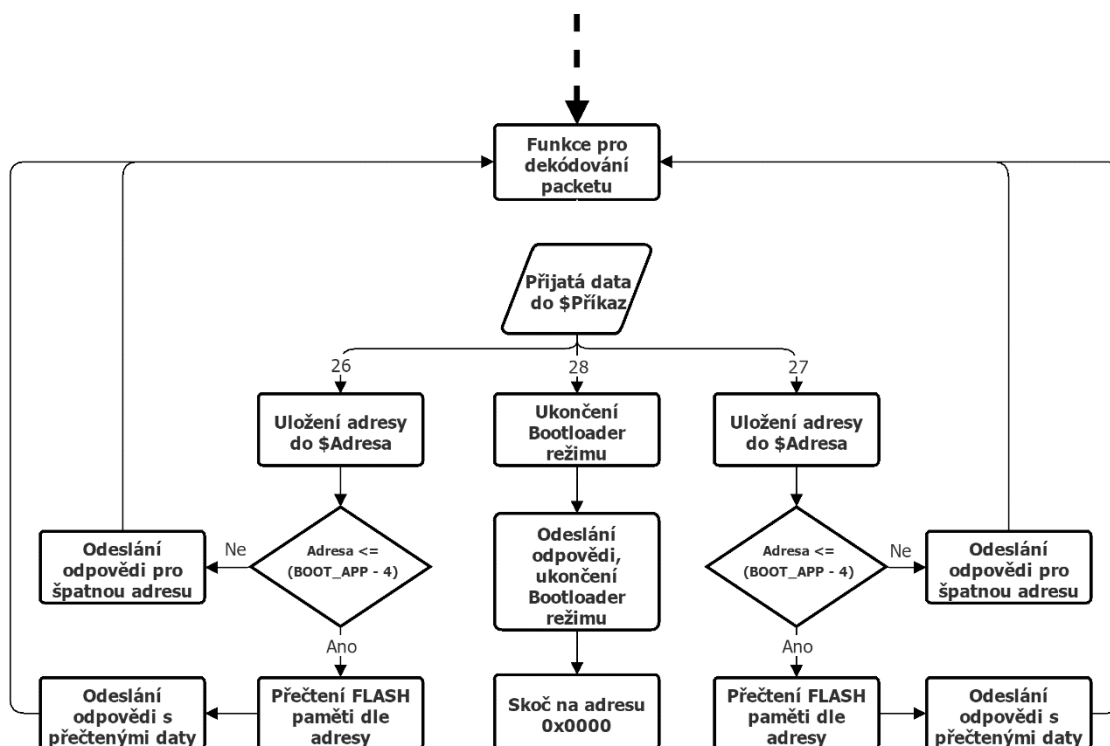
- ,25‘ – Konec zápisu do paměti vyobrazen na **obr. 30**.
 - Interní buffer má velikost stránky, což u některých mikrokontrolérů odpovídá velikosti 128, nebo 256 bajtů. Pokud zapisujeme do takto velkého bufferu poslední data do paměti a zapisujeme například pouze 1 bajt dat, nelze v příkazech 23, nebo 24 určit jestli jsou tyto data poslední nebo ne. Další možnost by byla v úpravě paketu s vyhrazením 1 bajtem na rozhodovací logiku ukončení zápisu. Ubráním jednoho bajtu z dat nám zvýší časovou náročnost pro zápis do paměti. U mikrokontroléru Atmega32 může být aplikační část velká až 28 kB. Ze zřejmých důvodů byl proto přidán příkaz pro ukončení zápisu a vynulování interního bufferu.
 - Příkaz nám dle globálního příznaku zjistí, do které paměti má zapsat, jestli do FLASH, nebo do EEPROM paměti s odpovědí Status – 102, Value – Pro FLASH paměti 23, pro EEPROM paměť 24.
 - V případě prázdného bufferu funkce odpoví s parametry Status - 102, Value – 0.



Obr. 30: Konec zápisu do paměti.

- ,26‘ – Přečtení FLASH paměti
 - Funkce zkontroluje čtení v aplikační části a odpoví s parametry Status – 100, Value – 4 bajty dat od přijaté adresy.
 - V případě špatné adresy funkce odpoví s parametry Status – 102, Value – Čtená adresa.

- ,27‘ – Přečtení EEPROM paměti
 - Funkce zkontroluje adresu, jestli uživatel nechce číst mimo EEPROM paměť a odpoví s parametry Status – 100, Value – 4 bajty dat od přijaté adresy.
 - V případě špatné adresy funkce odpoví s parametry Status – 102, Value – Čtená adresa.
- ,28‘ – Ukončení Bootloader režimu
 - Příkaz odešle uživateli splnění úkolu ve formě odpovědi s parametry Status – 100, Value – 28 a skočí na adresu 0x0000 v aplikační části paměti FLASH.



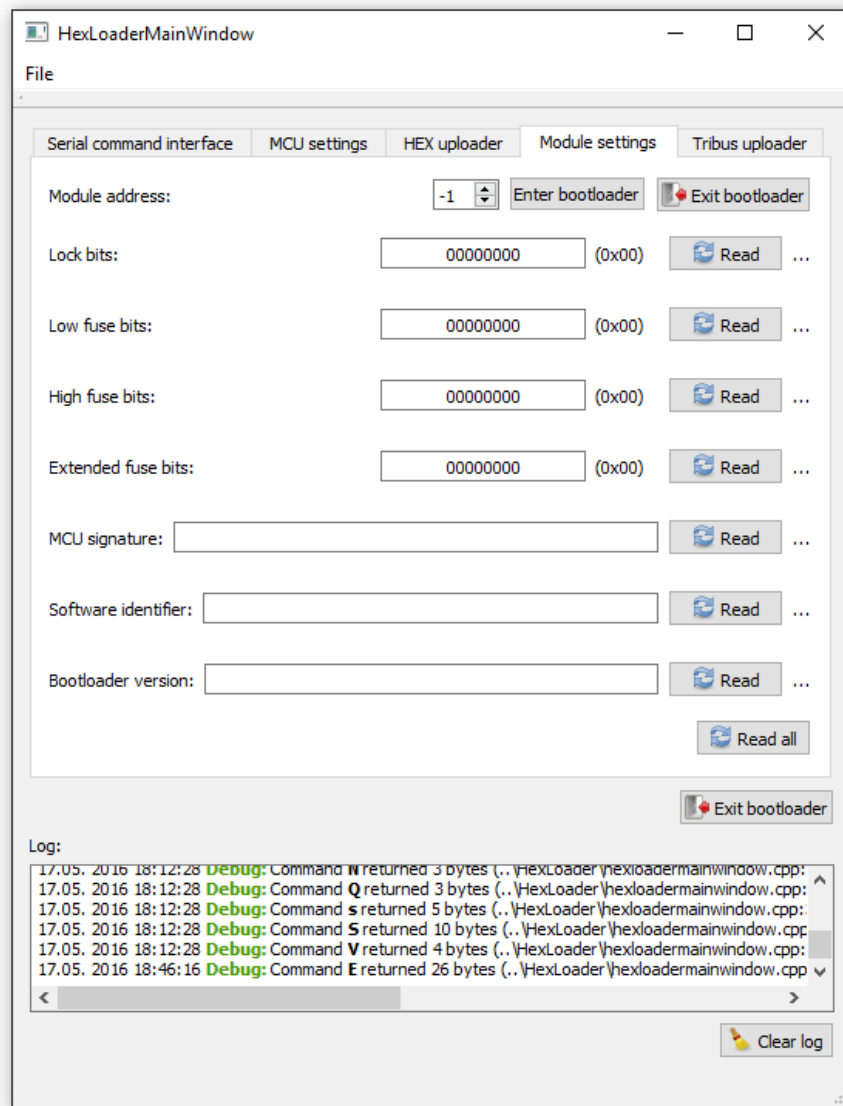
Obr. 31: Vývojový diagram pro příkazy 26 až 28.

6.3 Program HexLoaderSlave

Stejně jako firmware Bootloader-Master, tak i firmware Bootloader-Slave má stejný problém. Problém spočívá ve složitém nahrávání aplikačního firmwaru do mikrokontroléru bez uživatelského programu. Pro nahrávání lze využít již zmíněný volně dostupný terminál, avšak tento postup je dosti zdlouhavý a může nastat chyba v datech. Proto byl vytvořen program HexLoaderSlave, který zjednodušuje nahrávání aplikačního firmwaru. Program je součástí programu HexLoaderMaster.

6.3.1 Popis programu

Již v předchozí kapitole 5.4.1, je popsán program pro nahrávání aplikačního firmwaru do řídicí jednotky, ale bez popisu 2 dalších listů viditelných na obrázku, které patří do této kapitoly. List „*Module settings*“ (viz **obr. 32**) slouží k restartování externích modulů a zapnutí Bootloader režimu pro modul na konkrétní adrese, případnému opuštění režimu. V poslední řadě lze vyčíst na listu ostatní příkazy, jako je podpis mikrokontroléru, verze Bootloaderu nahraná v Bootloader části, případně verze nahraná v aplikační části. Všechny příkazy lze vypsat jednotlivě stlačením tlačítka „*Read*“, případně vyčíst všechny přes tlačítka „*Read all*“.

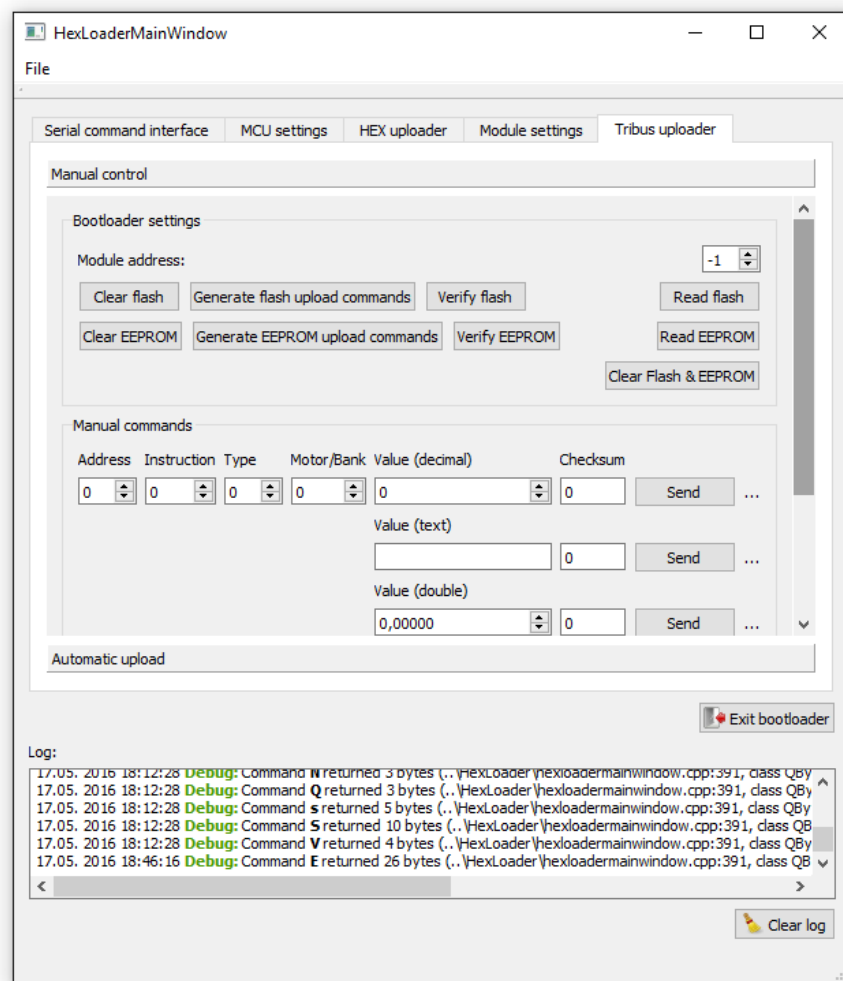


Obr. 32: List „*Module settings*“.

Další list „*Tribus uploader*“ je rozdělen na manuální nahrání a výpis příkazu po vygenerování pro nahrání. Manuální nahrání je rozděleno na manipulaci s FLASH a EEPROM paměti a na část pro vlastní poskládání a posílání Trinamic příkazů. Před samotnou manipulací s paměťmi je nutné v pravém horním rohu vyplnit adresu, se kterým modulem chceme komunikovat.

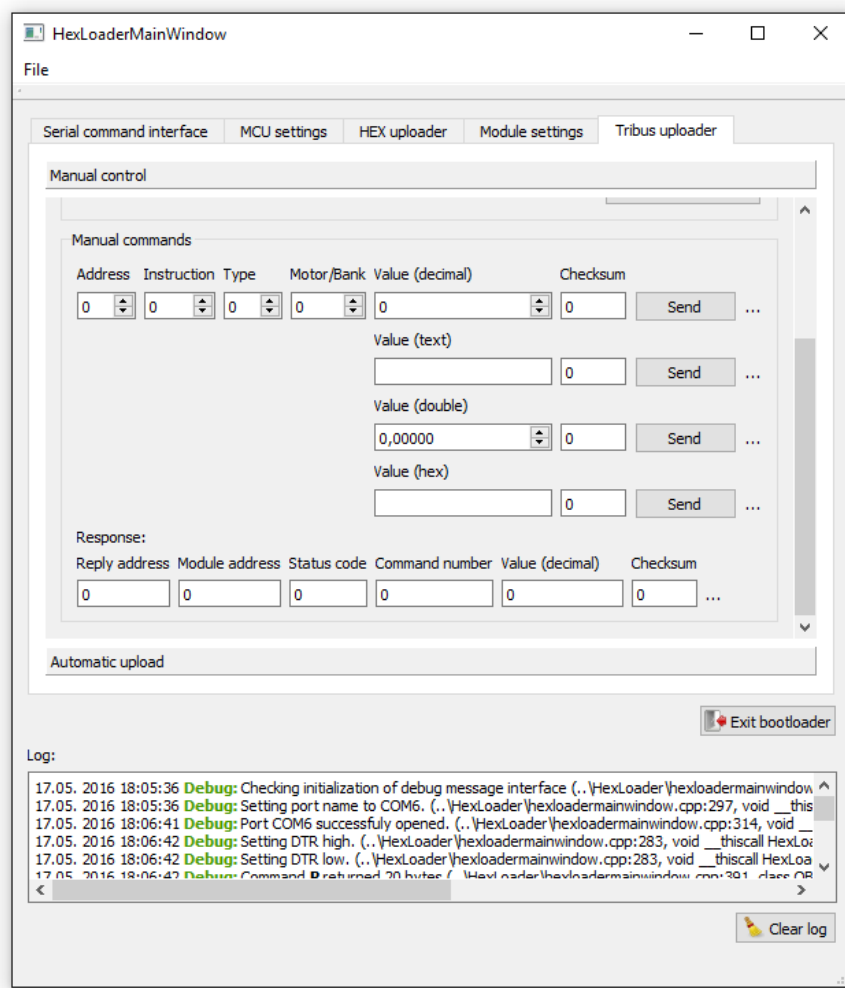
Pro přehlednost jsou tlačítka vypsaná do následujících odrážek:

- „*Clear Flash*“
 - Tlačítko pro vymazání aplikační části FLASH paměti do defaultní hodnoty 0xFF.
- „*Generate Flash upload commands*“
 - Generování Trinamic příkazů z Hex souboru s ukončovacím příkazem na konci, který určuje, že byli odeslána všechna data do Bootloader-Slave. Koncový příkaz musí být odeslán vždy po nahrání firmware do FLASH paměti. V případě neodeslání příkazu se poslední data uložená v Bufferu nezapišou do paměti.
- „*Verify Flash*“
 - Verifikace paměti, pro kontrolu dat ve FLASH paměti po nahrávání.
- „*Read Flash*“
 - Příkaz pro přečtení celé FLASH paměti.
- „*Clear Eeprom*“
 - Příkaz, který nastaví v EEPROM paměti data do defaultní hodnoty 0xFF.
- „*Generate Eeprom upload commands*“
 - Generování Trinamic příkazů z Hex souboru s ukončovacím příkazem na konci, který určuje, že byli odeslána všechna data do Bootloader-Slave. Koncový příkaz musí být odeslán vždy po nahrání firmware do EEPROM paměti. V případě neodeslání příkazu se poslední data uložená v Bufferu nezapišou do paměti.
- „*Verify Eeprom*“
 - Verifikace paměti, pro kontrolu dat v EEPROM paměti po nahrávání.
- „*Read Eeprom*“
 - Příkaz, který přečte celou EEPROM paměť.
- „*Clear Flash & Eeprom*“
 - Kombinace příkazů pro smazání FLASH i EEPROM paměti do jednoho.



Obr. 33: List „Tribus uploader“.

Jelikož samotné nahrávání je realizováno přes protokol Trinamic, který je implementován i v aplikačním firmwaru, je do programu implementováno i uživatelské sestavení vlastního příkazu pro konkrétní adresu modulu. Lze tedy program využít i pro komunikaci a ovládání jednotlivých modulů na úrovni aplikačního firmwaru. Ve vrchní části lze vidět nastavení adresy modulu, příkazu, typ, motor a 4 bajty dat. Hodnotu 4 bajtů dat lze zadat v několika formátech. Každý formát má své specifické tlačítko „Send“, po kterém se odešle celý příkaz s převedením do příslušného formátu po sběrnici. Ve spodní části je do kolonek rozdělena odpověď po sběrnici.



Obr. 34: Manuální zadávání příkazů.

6.3.2 Správný postup nahrávání do externích modulů

V případě nahrávání aplikačního firmwaru do mikrokontroléru stačí mít v listu „*Module settings*“ správně vyplněnou adresu modulu a ve vrchní liště zvolit „*File->Open Hex*“ pro nahrání aplikačního firmwaru do bufferu programu. V dalším kroku stlačením „*Clear Flash*“ preventivně vymazat FLASH paměť, aby v aplikační paměti nebyla něchtěná data. Pokud je vymazaná paměť a nahrán do programu správný aplikační firmware, stačí stlačit tlačítko „*Generate Flas upload commands*“, který nám vygeneruje příkazy pro nahrání do paměti. Kompletní přehled je zobrazen v podlistu *Automatic upload*, ve kterém lze zkontrolovat každý příkaz. Po stlačení tlačítka „*Send commands*“ se program postará o nahrávání firmwaru do aplikační paměti, případně dat do EEPROM paměti.

V posledním kroku musí být stlačeno tlačítko „*Exit Bootloader*“ v listu „*Module settings*“, čímž se opustím Bootloader režim a firmware Bootloader-Slave uvnitř mikrokontroléru skočí na adresu 0x0000 do aplikační paměti.

7 APLIKAČNÍ ČÁST ŘÍDÍCÍ JEDNOTKY

Aplikační část pro řídicí jednotku slouží ke zprostředkování komunikace mezi jednotlivými moduly a počítačem neboli uživatelem. K tomuto prostředku slouží Mikrokontrolér Atmega644P se dvěma UART rozhraními s datovým tokem přes RS485 a USB. Komunikace probíhá na protokolu od firmy Trinamic. Detailní popis protokolu je popsán v katalogovém listu firmware [12].

K řídicí jednotce při úpravě hardwarovém návrhu, byl přidán grafický displej zobrazující aktuální napětí na vnitřní +24 V, vnějších +24 V, +12 V, -12 V, +5 V, USB napětí, Bus-A a Bus-B na RS485 sběrnici. V neposlední řadě bylo přidáno teplotní čidlo od firmy Dallas, které měří teplotu uvnitř racku a zobrazuje na displeji. Tyto změny musí být zahrnuty i úpravě aplikačního firmwaru.

7.1 Výpočty pro děliče napětí

Pro měření napětí byl využit vnitřní 10 bitový AD převodník mikrokontroléru, který ale dokáže zvládnout pouze menší napětí než napájecí, v našem případě kvůli napěťové analogové referenci pouze menší než +2,5 V, ale za to přesnější.

Jako příklad výpočtu děliče se vychází z následujícího vzorce (7.1), kde jsme znali zkoumané napětí, neboli napájecí a maximální napětí možné na děliči. Pro prvotní výpočty se využilo napětí děliče +1,5 V. Jelikož katalogový list u AD pinů uvádí vnitřní odpor pinu na 10MΩ, lze tento vnitřní odpor zanedbat.

$$U_{Delic} = U_{Nap} * \frac{R_2}{R_2 + R_1} \quad (7.1)$$

Pomocí několika iterací se vždy zvolil jeden z rezistorů a dopočítal se druhý rezistor. Poté se tento rezistor nahradil nějakým z dostupných rezistorů na trhu a vypočítalo se napětí na děliči, se kterým se dále mohlo počítat při sestavování vyčítání a převádění na napětí (viz **tab. 13**). Jediný výpočet, u kterého byla změna, je záporné napětí, které jsme ještě museli odečíst od +5 V z důsledku nenulového potenciálu.

Tab. 13: Tabulka výpočtu děličů pro AD převodník.

	U_{Nap}	U_{Delic}	R_2	R_1
Vnitřní +24 V	+24 V	+1,819 V	100k	8,2k
Vnější +24 V	+24 V	+1,819 V	100k	8,2k
Vnitřní +12 V	+12 V	+2,112 V	220k	8,2k
Vnitřní +5 V	+5 V	+1,56 V	22k	10k
Napětí USB	+5 V	+1,56 V	22k	10k
Bus-A	+2,5 V	+0,781 V	22k	10k
Bus-B	+2,5 V	+0,781 V	22k	10k
Vnitřní -12 V	-12 V	+2,007 V	220k	47k

7.2 Výpočet převodu pro program

Z odporového děliče známe napětí odpovídající skutečnému napětí. AD převodník nám zjistí hodnotu mezi 0 a 1023 na děliči, které je dále nutné přepočítat na skutečné napájecí napětí. Dle vzorce pro výpočet skutečné hodnoty napětí na jednotlivých větvích (7.2). Konstanta pro výpočet je 10 bitová hodnota 1024, dále hodnota referenčního napětí na pinu AVCC, kde pro naše zapojení s čipem TL431, je hodnota +2,495 V uvedená v katalogovém listu.

$$Float = \frac{U_{Nap}}{\left(\frac{1024}{\left(\frac{2,495}{U_{Delic}}\right)}\right)} \quad (7.2)$$

Ze vzorce nám vyšly hodnoty v datovém typu Float, které potom byly ve firmwaru vynásobeny aktuální 10 bitovou hodnotou z převodníku. Z kompletního přehledu těchto hodnot byla sestavena **tab. 14**.

Tab. 14: Tabulka výpočtu převodu na skutečnou hodnotu.

	U_{Nap}	U_{Delic}	Float
Vnitřní +24 V	+24 V	+1,819 V	0,03152
Vnější +24 V	+24 V	+1,819 V	0,03152
Vnitřní +12 V	+12 V	+2,112 V	0,01080
Vnitřní +5 V	+5 V	+1,56 V	0,00765
Napětí USB	+5 V	+1,56 V	0,00765
Bus-A	+2,5 V	+0,781 V	0,00765
Bus-B	+2,5 V	+0,781 V	0,00765
Vnitřní -12 V	-12 V	+2,007 V	0,01293

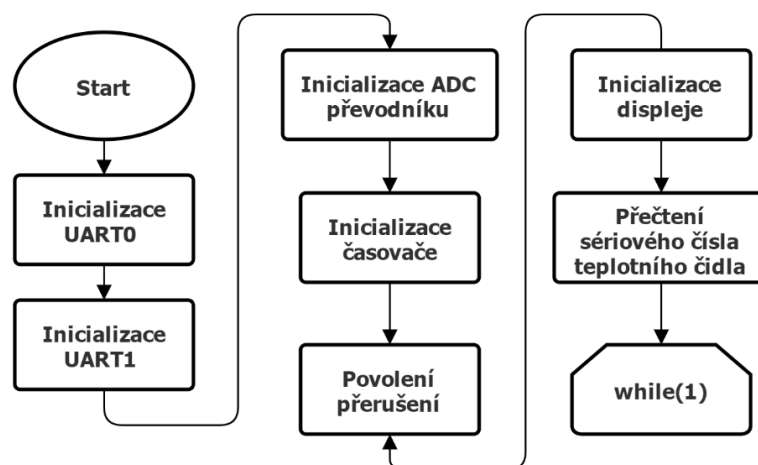
7.3 Inicializační část

Aplikační program je složen z části obstarávající komunikaci mezi uživatelem a moduly v rackovém systému a přidanou část pro informační displej. Celá část komunikace nebyla nijak upravována a byla ponechána dle původního protokolu od firmy Trinamic. Je zde uveden pouze vývojový diagram s inicializacemi sběrnic (viz **obr. 35**), časovači a samotným nekonečným cyklem, ve kterém se přijímají a přeposílají data, případně je mikrokontrolér sám dekóduje, pokud tyto data jsou určena jemu.

Ve vývojovém diagramu lze vidět přidané inicializace pro AD převodník, displej a teplotní čidla. U AD převodníku se nastaví v registru ADCSRA bit ADEN, kterým se aktivuje převodník a přes poslední 3 bity se nastaví výběr převodní dělička, kde pro program byl zvolen převod 128. Tento údaj udává při krystalu 14,7456 MHz konverzní převod 115,2 kHz. Menší konverzní převod by znamenal menší přesnost měření. Nastavením registru ADCSRB lze zvolit zdroj vnějšího spuštění převodu a registrem DIDR0 se nastaví aktivní piny pro převodník. Jelikož v našem zařízení není použit žádný externí způsob vyvolání převodu, je registr ADCSRB nastaven na hodnotu 0 a registr DIDR0, kvůli využití všech pinů pro převod, je nastaven na hodnotu 255. V poslední řadě je nutné nastavit registr ADMUX, ve kterém se nastaví výběr referenčního napětí, v našem případě nastavíme přivedené přesné napětí na pin AREF s vypnutým interním referenčním napětím. Ostatní nastavení registru ADMUX nevyužijeme, proto jej nastavíme do nuly.

Pro inicializaci displeje využijeme knihovnu GLCD, kde v hlavičce souboru nastavíme jednotlivé piny, dle hardwarového nastavení. Poté jenom zavoláme funkci pro vyčištění textu, generátoru pole a grafickou část.

V poslední fázi inicializace provedeme přečtení sériového čísla teplotního čidla. Tento krok bychom mohli vynechat, pokud bychom do paměti zadali pevné sériové číslo teploměru. Jelikož se ale počítá se sériovou výrobou, je tento krok nutný, jinak by musel být pro každý nový vyrobený kus upraven firmware, což je neekonomické. Po inicializaci je v programu vytvořena nekonečná smyčka, kde je příjem a odeslání samotný dat.



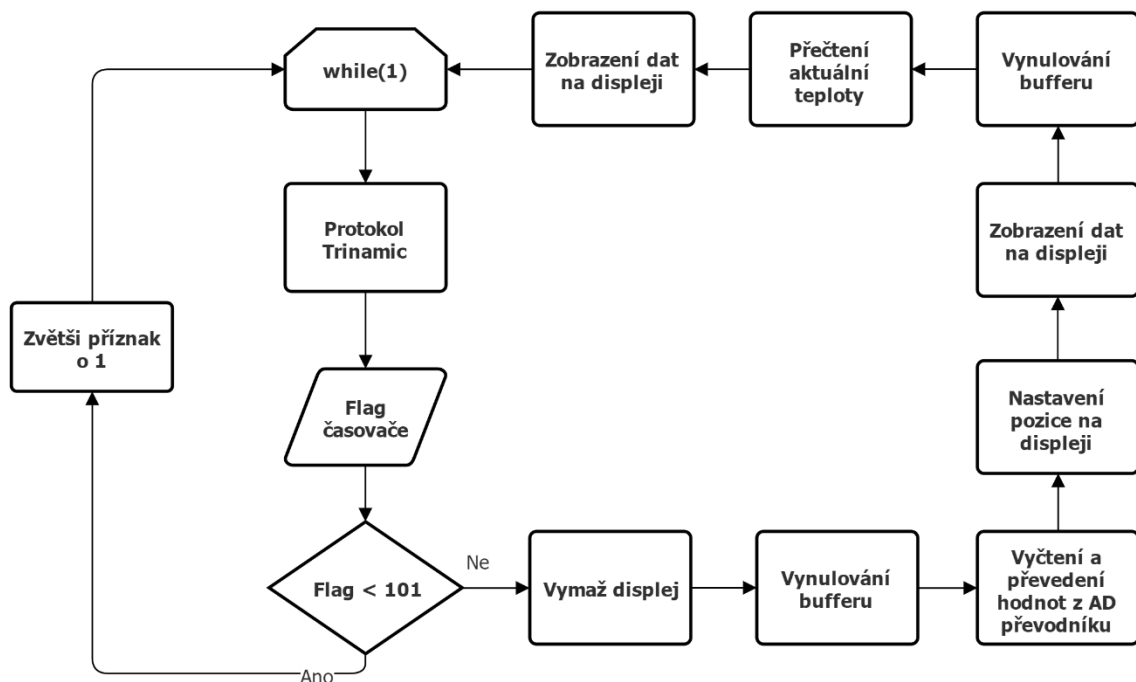
Obr. 35: Inicializace aplikačního programu.

7.4 Firmwarové úpravy aplikačního programu

Nekonečná smyčka v aplikačním programu se skládá z funkce, která obstarává spojnici mezi uživatelem a jednotlivými moduly přes protokol Trinamic. Tato funkce je složena z naplnění bufferu příjmu z počítače od uživatele, kde je následně tento příkaz zpracován a přeposlán na RS485 sběrnici, případně se data dekódují, pokud jsou určena pro řídicí jednotku. Dále je naplněn buffer pro odeslání odpovědi z řídicí jednotky, pokud je tento krok vyžadován, a buffer pro odeslání odpovědi ze sběrnice. Připraví se data z bufferů a odešlou se jak uživateli, tak modulům. V posledním kroku je zaveden 10 ms časovač, který nechává tuto zvolenou prodlevu mezi zprávami.

Po samotném protokolu Trinamic přichází kontrola příznaku časovače a čítač, který je nastaven na velikost menší než 101, což odpovídá 1 sekundě, ve které se vždy aktualizují data na displeji. V případě popisu programu je zde slovně popsán program ve větším rozsahu než je vidět na vývojovém diagramu. Ve smyčce v čítači se opakují jednotlivé řádky, proto je vývojový diagram (viz **obr. 36**) zjednodušen pouze pro jeden řádek výpisu napájecího napětí a jeden řádek teploty.

V případě smyčky aktualizace dat na displeji se nejprve vymaže celý displej, který se maže z důvodu prevence před chybnými zobrazenými daty tzv. „Artefakty“. V dalším kroku je vynulován výstupní buffer, do kterého jsou ukládána výstupní data do datové proměnné string přes funkci *sprintf*. Po vynulování bufferu se přejde k prvnímu kroku, kde se nastaví X a Y pozice na displeji a vypíše se samotný text nadpisu s označením celého zařízení Sci-Trace. Pro přehlednost mezi nadpisem a výpisem dat je vynechán jeden řádek. Ve výpisu dat jsou po levé straně vypsány měřené napětí a na pravé straně aktuální změřené napětí převodníkem. Jediná výjimka u připojení USB, kde místo napájecího napětí je zobrazen text „Connect“, pokud je zařízení připojeno k počítači a v případě odpojení od počítače zobrazí „N.C“.

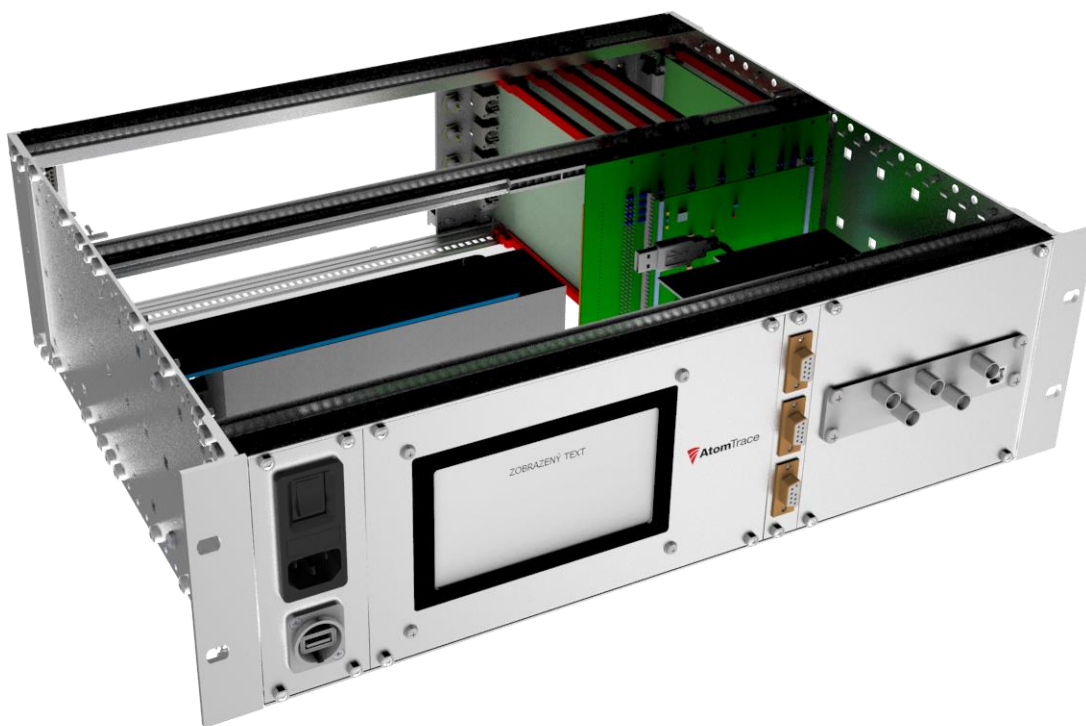


Obr. 36: Vývojový diagram s upraveným firmwarem.

8 SESTAVENÉ ZAŘÍZENÍ

Kompletní zařízení je sestaveno do rackového systému od firmy Schroff. V přední části je připevněn do panelu displej ze vstupním konektorem pro silnoproudou část elektroniky, vypínačem s pojistkou a USB konektor, připojený v racku do řídicí jednotky. Vedle displeje jsou vyvedeny konektory D-sub pro externí vyvedení RS485 sběrnice. Poslední přední část zabírá delay generátor od firmy Quantum Composer Sapphire 9200, který je pouze uložen do racku bez napojení na elektroniku.

Na modelu (viz **obr. 37**) lze vidět na levé části uvnitř racku připevněný spínaný zdroj SP-150-24, který napájí celou elektroniku přes napájecí napětí +24 V. Na střední příčce je dále připevněna propojovací karta, do které je ze strany displeje vložena řídicí jednotka. Ze zadní části racku jsou vloženy externí výměnné moduly připojené do propojovací karty. Sestavené zařízení připraveno pro montáž do zařízení Sci-Trace bez dalšího vnitřního zásahu.



Obr. 37: Model celého zařízení.

9 ZÁVĚR

Cílem této práce bylo navrhnout a realizovat elektroniku pro zařízení Sci-Trace, s požadavkem zobrazení informačního stavu elektroniky na displeji. Kvůli možnosti prodeje zařízení do celého světa bylo dále potřeba vyřešit jednoduché nahrávání firmwaru do řídicí elektroniky, i do podružných modulů, bez hardwarové záměny.

Prvním konceptem návrhu modulárního způsobu výměny podružných modulů bez vnitřního zásahu do elektroniky v rackovém systému, bylo přesunutí konektorů ze zadní části rackového dílu na přední stranu samotných modulů. Přesunutím konektorů na podružné moduly, se tak celá sestava mohla obrátit. Vkládání modulů z přední strany tak bylo přesunuto do zadní strany rackového systému. Po uvolnění místa se na přední stranu racku mohl z koncepčního hlediska umístit informační displej s vypínačem a přívodním kabelem. Též byl do přední vnitřní části racku umístěn spínaný zdroj. Z koncepčního hlediska byla na displej vybrána data o stavu napájecích větví, připojené elektroniky k počítači a stavu teploty uvnitř racku.

Z konceptu byl nejprve vybrán informační displej s připevněním do racku, s odpovídajícími rozměry a parametry, z několika různých typů. Po výběru displeje bylo dále vybráno pouze jedno teplotní čidlo pro měření teploty uvnitř racku, z několika různých čidel od různých výrobců. Při návrhu řídicí elektroniky bylo nutné brát ohled na vybraný typ Mikrokontroléru, u kterého byly nároky na rozdělenou paměť na dvě části s povoleným zápisem z jedné části do druhé pro samo přeprogramování. S připojením datového pinu převodníku USB do resetu Mikrokontroléru bylo zajištěno externí restartování řídicí jednotky a možnost aktivace konceptu Bootloader. Kde malá část kódu, dokáže přeprogramovat zbytek paměti.

U návrhu rozměrů propojovací desky bylo nutné dodržet všech standardů dle norem ČSN pro rackový systém. Z důvodu ušetření místa pro podružné moduly, byla řídicí deska přesunuta na zadní stranu propojovací desky. Samotné propojení zajistilo rozvedení napájecích napětí, datových signálů i resetování Mikrokontrolérů na podružných deskách a tím nahrání nového firmwaru.

Po navržení hardwarové stránky řídicí jednotky i propojovací desky, byl vytvořen koncept ve formě vývojového diagramu firmwaru Bootloader pro řídicí jednotku, nazvaného též jako Bootloader-Master, který obsahoval zápis a čtení dat z a do paměti mikrokontroléru. Kromě manipulaci s pamětí bylo v konceptu zahrnuto i čtení informací o verzích firmwarů v mikrokontroléru, i konkrétní typ mikrokontroléru.

Na základně navrženého konceptu vývojového diagramu Bootloader-Master, byl v programu Atmel Studio vyvinut firmware a odzkoušen na samotném zařízení přes volně dostupný terminál. Pro zefektivnění a zjednodušení nahrávání aplikačního firmwaru přes Bootloader-Master, byl vytvořen program pro prostředí operačního systému Windows.

V poslední fázi diplomové práce, byla celá elektronika připevněna do vybraného rackového systému osazená řídicí jednotkou s bootloader a aplikačním firmwarem. Pro demonstraci byl dále do zařízení přidán externím modul pro ovládání tlakového systému s bootloader firmwarem.

LITERATURA

- [1] ČSN, Mechanické konstrukce pro elektronická zařízení - Rozměry mechanických konstrukcí řady 482,6 mm (19 palců) - Část 3-100: Základní rozměry čelních panelů, skříní, stojanů, zásuvných jednotek a koster, 1.9.2009.
- [2] ČSN, Konektory pro frekvence do 3 MHz pro desky s plošnými spoji - Část 2: Předmětová specifikace pro nepřímé konektory s hodnocenou jakostí pro desky s plošnými spoji se základní sítí 2,54 mm (0,1 palce) se společnými montážními vlastnostmi, 1.5.1991.
- [3] A. Comporation, „Datasheet Atmega644P,“ 2015. [Online]. Available: http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf.
- [4] I. Maxim. [Online]. Available: <https://www.maximintegrated.com/en/products/interface/transceivers/MAX481.html>. [Přístup získán 7 12 2015].
- [5] FTDI, „Future Technology Device International,“ [Online]. Available: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf. [Přístup získán 7 12 2015].
- [6] BuyDisplay. [Online]. Available: www.BuyDisplay.com. [Přístup získán 7 12 2015].
- [7] BuyDisplay, „ERM240128-2,“ [Online]. Available: http://www.buydisplay.com/download/manual/ERM240128-2_Series_Datasheet.pdf. [Přístup získán 7 12 2015].
- [8] Maxim Integrated, „DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet,“ 2008. [Online]. Available: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. [Přístup získán 29 květen 2014].
- [9] A. Comporation, "AVR109: Self Programming," Atmel, 2004. [Online]. Available: <http://www.atmel.com/images/doc1644.pdf>.
- [10] O. Karas, „AVR,“ [Online]. Available: <http://programujte.com/clanky/92-avr/>. [Přístup získán 29 květen 2014].
- [11] Avr-libc, „Avr-libc - Boot.h,“ [Online]. Available: http://www.nongnu.org/avr-libc/user-manual/group__avr__boot.html. [Přístup získán 5 5 2016].
- [12] „TMCM - 6110,“ Trinamic Motion control, [Online]. Available: <http://www.trinamic.com/products/modules/controller-driver-stepper/tmcm-6110>. [Přístup získán 2 5 2016].
- [13] „PROGRAMUJEME AVR V JAZYKU C - 2. ČÁSTĚ,“ 2012. [Online]. Available: <http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-2-cast-446.html>. [Přístup získán 29 květen 2014].

- Elektronické konstrukce, „Knihovny pro AVR,“ [Online]. Available:
[14] <http://mujweb.cz/elecon/cz/knihovnyavr/knihovnyavr.html>. [Přístup získán 29 květen 2014].
- Atmel Corporation, „AVR Libc Reference Manual,“ [Online]. Available:
[15] http://www.atmel.no/webdoc/AVRLibcReferenceManual/group__avr__eeprom.html. [Přístup získán 29 květen 2014].
- Schroff, „Subracks and 19" chassis,“ [Online]. Available:
[16] <http://new.schroff.de/catalogue/catalogue.do;jsessionid=527ECE678E3B2076DB7B67C44BAABA5A?act=showBookmark&favOid=0000000500023c0300010023&lang=en&catId=DE>.

SEZNAM ZKRATEK A VYSVĚTLIVKY

RACK	Mechanická konstrukce pro elektrotechnická zařízení.
TQFP	Thin Quad Flat Pack, druh pouzdra
MLF	Micro Lead Frame, druh pouzdra
RACK	Mechanická konstrukce pro elektrotechnická zařízení.
USART	Universal Synchronous / Asynchronous Receiver and Transmitter, synchronní / asynchronní komunikace přes sériovou linku
PDIP	Plastic Dual Inline Package, druh pouzdra
U	Unit, jednotka pro rackový systém
HP	Horizontal Pitch, jednotka pro eurokarty
IIC	Inter-Integrated Circuit, typ sběrnice
Bootloader-Slave	Program v paměti mikrokontroléru externího modulu
Bootloader-Master	Program v paměti mikrokontroléru řídicí jednotky
LIBS	Laser-Induced Breakdown Spectroscopy, metoda k detekci chemických prvků
N.C.	Not connected, nepřipojeno
Fuses	Pojistné bity pro hardwarové nastavení mikrokontroléru Atmega
Page Buffer	Speciálně vyhrazená paměť o velikosti 1 stránky
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná paměť
FLASH	Programová paměť, pro uložení aplikačního firmwaru
SSOP	Shrink Small Outline Package, druh pouzdra
QFN	Quad Flat No-leads package, druh pouzdra
HID	Human Interface Device, standard zprostředkující komunikaci s uživatelem
FTDI	Future Technology Devices International, společnost vyrábějící zařízení komunikující na USB protokolu
bps	bit per sec, jednotka rychlosti komunikace
Mbps	Megabit per sec, jednotka rychlosti komunikace
SO08	Small Outline Package, druh pouzdra s 8 vývody
RS485	Standard sériové komunikace
USB	Universal Serial Bus, sériová sběrnice.

JTAG	Joint Test Action Group, Konektor pro testování plošných spojů a programování paměti.
SMD	Surface Mount Device, součástka pro povrchovou montáž
OLED	Organic Light-Emitting Diode, typ displej
RWW	Read-While-Write, druh paměti přístupná při čtení
NRWW	No Read-While-Write, druh paměti nepřístupná při čtení
GND	Ground, uzemnění
Buffer	Vyrovňovací paměť, pro uchování dat
ACK	Acknowledge byte, bajt potvrzující správnost dat
NAK	Not Acknowledge byte, bajt potvrzující chybnost dat
CR	Carriage Return, bajt ukončující přenos
PWM	Pulse Width Modulation, pulzní šířková modulace
1-Wire	Sběrnice navržená firmou Dallas, s nízkou datovou rychlostí.
ROM	Read-Only Memory, elektronická paměť, bez možnosti zápisu.
Bd	Baud, jednotka modulační rychlosti, značí počet přenesených bitů za sekundu
Firmware	Zdrojový kód pro mikrokontrolér
Software	Zdrojový kód pro počítač

SEZNAM PŘÍLOH

A	Návrh zařízení	60
Příloha 1:	Standardizované konektory pro připojení externích modulů.....	60
Příloha 2:	Filtrační kondenzátory a rezistory pro RS485.....	61
Příloha 3:	Propojovací konektory pro další propojovací desky a pro řídicí jednotku..	62
Příloha 4:	Propojovací konektory na řídicí desce.....	63
Příloha 5:	DC-DC měniče.....	64
Příloha 6:	Mikrokontrolér a komunikace po RS485.....	65
Příloha 7:	Komunikace k počítači po FTDI převodníku.....	66
Příloha 8:	Zapojení displeje.....	67
Příloha 9:	Zapojení AD převodníku pro jednotlivé větve.....	68

A NÁVRH ZAŘÍZENÍ

Tato část podléhá utajení

Příloha 1: Standardizované konektory pro připojení externích modulů.

Tato část podléhá utajení

Příloha 2: Filtrační kondenzátory a rezistory pro RS485.

Tato část podléhá utajení

Příloha 3: Propojovací konektory pro další propojovací desky a pro řídicí jednotku.

Tato část podléhá utajení

Příloha 4: Propojovací konektory na řídicí desce.

Tato část podléhá utajení

Příloha 5: DC-DC měniče.

Tato část podléhá utajení

Příloha 6: Mikrokontrolér a komunikace po RS485.

Tato část podléhá utajení

Příloha 7: Komunikace k počítači po FTDI převodníku.

Tato část podléhá utajení

Příloha 8: Zapojení displeje.

Tato část podléhá utajení

Příloha 9: Zapojení AD převodníku pro jednotlivé větve.