



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií



# Návrh měřicí desky se sběrnicí MODBUS pro PV elektrárnu

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 1802T007 – Informační technologie

*Autor práce:* **Bc. Pavel Dvorský**

*Vedoucí práce:* Ing. Miroslav Holada, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Design of the measuring board using MODBUS for the PV power plant

## Diploma thesis

*Study programme:* N2612 – Electrical Engineering and Informatics

*Study branch:* 1802T007 – Information Technology

*Author:* **Bc. Pavel Dvorský**

*Supervisor:* Ing. Miroslav Holada, Ph.D.



## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Pavel Dvorský**  
Osobní číslo: **M14000154**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Informační technologie**  
Název tématu: **Návrh měřicí desky se sběrníci MODBUS pro PV elektrárnu**  
Zadávací katedra: **Ústav informačních technologií a elektroniky**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Seznamte se s problematikou PV elektráren, zaměřte se na monitoring jednotlivých řetězců solárních panelů.
2. Seznamte se se stávající měřicí deskou pro monitorinng proudů ze stringů PV. Zhodnoťte její výhody a nevýhody.
3. Navrhněte alternativní měřicí desku s využitím moderního mikrokontroléru a součástek. V softwarové výbavě pro komunikaci přes MODBUS realizujte pouze základní vybrané funkce.
4. Navrženou desku realizujte a prototyp prakticky otestujte.
5. Proveďte zhodnocení dosaženého výsledku včetně srovnání se stávající deskou.

Rozsah grafických prací: **Dle potřeby dokumentace**

Rozsah pracovní zprávy: **cca 40-50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **BARRY, Richard. Using the FreeRTOS Real Time Kernel. 1. vyd. London: Real Time Engineers Ltd., 2010. ISBN 978-1-4461-7030-4**

Vedoucí diplomové práce: **Ing. Miroslav Holada, Ph.D.**

Ústav informačních technologií a elektroniky

Datum zadání diplomové práce: **12. září 2016**

Termín odevzdání diplomové práce: **15. května 2017**

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

prof. Ing. Ondřej Novák, CSc.  
vedoucí ústavu

V Liberci dne 12. září 2016

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

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

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

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 20.9.2017

Podpis:

## Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Miroslavu Holadovi, Ph.D. za poskytnutí materiálů potřebných pro vypracování této práce, jeho rad ohledně směřování projektu a spolupráce.

## Abstrakt

Zadání této práce není vhodné pro obor IT, a tak došlo po domluvě s vedoucím práce k drobným změnám. Místo vytváření zcela nové desky se práce zaměří pouze na malé moduly, které mohou nahradit stávající zapojení na deskách, které jsou k dispozici na pracovišti vedoucího.

Tato práce zjednodušeně popisuje funkci solárních panelů, jejich využití, výhody a nevýhody oproti ostatním zdrojům energie, uvádí podrobněji do problematiky sběrnice MODBUS a sériové komunikaci, a popisuje základní vlastnosti paměti Flash a SRAM.

Dále popisuje a porovnává vlastnosti a provedení měřících desek na pracovišti vedoucího práce. Na žádost vedoucího nejsou v práci uvedeny názvy desek a jejich výrobců, ani jejich materiálů, jako jsou schéma zapojení a podobně. V rámci dohody s výrobcí zůstávají tyto informace utajené.

V práci jsou navrženy 2 desky, které by mohly nahradit některé moduly u stávajících desek. Deska pro nahrazení napájecí části nebyla nakonec realizovaná, neboť její řešení, cena a výkon by nepředčily stávající řešení. Jediná výhoda by byla v použití dostupnějších součástek. Druhá deska byla realizována a nahrazuje stávající řešení měřící jednotky na jedné z desek. Součástí práce je i popis naprogramování komunikace přes sběrnici MODBUS v režimu ASCII a také popis rozdělení programu do jednotlivých modulů na základě jejich funkcionality.

### **Klíčová slova**

MODBUS, FLASH, solární panel, PV, STM32F411RE

## Abstract

The assignment of this thesis is not suitable for the IT field, therefore an agreement was made with the supervisor to make a few small changes. Instead of creating a brand new board, the thesis will focus only on small modules that can replace already existing board modules available at the supervisor's workplace.

This paper briefly describes the function of solar panels, their utilization, advantages and disadvantages compared to other energy sources, presents in more detail the problems of MODBUS protocol and serial communication, and describes the basic features of Flash and SRAM.

It also describes and compares the properties and performance of measuring boards at the workplace of the supervisor. At the request of the supervisor, the names of the boards and their manufacturers, or manuals such as the wiring diagram etc., are not listed. Under the agreement with manufacturers, this information remains confidential.

There are 2 boards designed to replace some of the modules on existing boards. The power supply replacement panel was not finally implemented because its solution, price and performance would not surpass existing solution. The only advantage would be use of more accessible components. The second board was implemented and replaces the existing solution of the measuring unit on one of the boards. Part of the thesis also describes the programming of communication using MODBUS in ASCII mode as well as a description of program distribution into individual modules based on their functionality.

### **Key words**

MODBUS, FLASH, solar panel, PV, STM32F411RE



# Obsah

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>1. ÚVOD .....</b>                                             | <b>12</b> |
| <b>2. PROBLEMATIKA PV ELEKTRÁREN .....</b>                       | <b>13</b> |
| 2.1. SOUČÁSTI SOLÁRNÍ ELEKTRÁRNY.....                            | 13        |
| 2.2. PRINCIP SOLÁRNÍCH PANELŮ .....                              | 13        |
| 2.3. VYUŽITÍ SOLÁRNÍCH PANELŮ.....                               | 14        |
| 2.4. VÝHODY A NEVÝHODY .....                                     | 15        |
| <b>3. SBĚRNICE MODBUS.....</b>                                   | <b>17</b> |
| 3.1. APLIKAČNÍ PROTOKOL .....                                    | 17        |
| 3.2. KOMUNIKACE.....                                             | 18        |
| 3.3. REŽIM PŘENOSU KOMUNIKACE PO SÉRIOVÉ LINCE .....             | 19        |
| 3.3.1. <i>Binární režim</i> .....                                | 19        |
| 3.3.2. <i>ASCII režim</i> .....                                  | 20        |
| 3.4. SEZNAM ZÁKLADNÍCH A CHYBOVÝCH FUNKCÍ PROTOKOLU MODBUS ..... | 22        |
| <b>4. PAMĚTI FLASH A SRAM .....</b>                              | <b>23</b> |
| 4.1. PAMĚŤ FLASH .....                                           | 23        |
| 4.2. PAMĚŤ SRAM .....                                            | 23        |
| <b>5. POUŽÍVANÉ DESKY NA PRACOVÍŠTI VEDOUČÍHO .....</b>          | <b>25</b> |
| 5.1. DESKA 1KVSCB .....                                          | 25        |
| 5.1.1. <i>Zdrojová část</i> .....                                | 26        |
| 5.1.2. <i>Měřicí senzory</i> .....                               | 26        |
| 5.1.3. <i>Zpracování dat a komunikace</i> .....                  | 26        |
| 5.2. DESKA PECONT .....                                          | 27        |
| 5.2.1. <i>Zdrojová část</i> .....                                | 27        |
| 5.2.2. <i>Měřicí senzory</i> .....                               | 27        |
| 5.2.3. <i>Zpracování dat a komunikace</i> .....                  | 27        |
| 5.3. DESKA SMU .....                                             | 28        |
| 5.3.1. <i>Zdrojová část</i> .....                                | 28        |
| 5.3.2. <i>Měřicí senzory</i> .....                               | 28        |
| 5.3.3. <i>Zpracování dat a komunikace</i> .....                  | 29        |
| 5.4. POROVNÁNÍ DESEK, ROZDÍLY .....                              | 29        |
| 5.4.1. <i>Volba napájení desky</i> .....                         | 29        |
| 5.4.2. <i>Senzory, zapojení, výhody a nevýhody</i> .....         | 30        |
| <b>6. NÁVRHY DESEK .....</b>                                     | <b>31</b> |
| 6.1. NÁVRH NAPÁJENÍ ZE SOLÁRNÍCH PANELŮ .....                    | 31        |
| 6.1.1. <i>Měnič napětí</i> .....                                 | 31        |
| 6.1.2. <i>Dělič napětí</i> .....                                 | 31        |
| 6.2. ŘÍDÍCÍ JEDNOTKA PRO DESKU PECONT .....                      | 34        |
| 6.2.1. <i>Mikrokontrolér</i> .....                               | 34        |
| 6.2.2. <i>Realizace desky</i> .....                              | 35        |
| <b>7. PROGRAM ŘÍDÍCÍ JEDNOTKY .....</b>                          | <b>37</b> |
| 7.1. STM32CUBEMX .....                                           | 37        |
| 7.2. KOMUNIKACE PO SBĚRNICI MODBUS.....                          | 38        |
| 7.2.1. <i>Funkce rozdělení bytu na 2 ASCII znaky</i> .....       | 39        |
| 7.2.2. <i>Funkce výpočtu LRC</i> .....                           | 39        |
| 7.2.3. <i>Sestavení rámce a odeslání odpovědi</i> .....          | 40        |
| 7.2.4. <i>Funkce pro převod dvou ASCII znaků na byte</i> .....   | 40        |
| 7.2.5. <i>Funkce pro příjem požadavku</i> .....                  | 40        |

|           |                                            |           |
|-----------|--------------------------------------------|-----------|
| 7.3.      | MODULARIZACE .....                         | 41        |
| 7.4.      | SHRNUTÍ CELÉHO PROGRAMU.....               | 41        |
| <b>8.</b> | <b>ZÁVĚR .....</b>                         | <b>43</b> |
| <b>9.</b> | <b>REFERENCE A ODBORNÁ LITERATURA.....</b> | <b>44</b> |

## Seznam obrázků

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| OBRÁZEK 1: FOTOGRAFIE SOLÁRNÍ FARMY [3] .....                                           | 13 |
| OBRÁZEK 2: P-N PŘECHOD SOLÁRNÍHO ČLÁNKU [1] .....                                       | 14 |
| OBRÁZEK 3: VZHLED MEZINÁRODNÍ VESMÍRNÉ STANICE PO JEJÍM DOKONČENÍ V ROCE 2011 [9] ..... | 15 |
| OBRÁZEK 4: MODBUS APLIKAČNÍ VRSTVA [4] .....                                            | 17 |
| OBRÁZEK 5: DATOVÝ RÁMEC MODBUS [4] .....                                                | 17 |
| OBRÁZEK 6: ADRESACE RÁMCE ADU [4] .....                                                 | 18 |
| OBRÁZEK 7: PŘÍKLAD ÚSPĚŠNÉ TRANSAKCE [10] .....                                         | 19 |
| OBRÁZEK 8: NASTAVENÍ PARITNÍHO BITU ZPRÁVY RTU REŽIMU [4] .....                         | 19 |
| OBRÁZEK 9: RÁMEC BINÁRNÍHO REŽIMU S PŘÍKLADEM PŘENOSU [4] .....                         | 20 |
| OBRÁZEK 10: NASTAVENÍ PARITNÍHO BITU ZPRÁVY ASCII REŽIMU [4] .....                      | 20 |
| OBRÁZEK 11: RÁMEC ASCII REŽIMU [4] .....                                                | 21 |
| OBRÁZEK 12: DESKA 1KVSCB .....                                                          | 25 |
| OBRÁZEK 13: DESKA PECONT S NAVRŽENOU ŘÍDÍCÍ JEDNOTKOU .....                             | 27 |
| OBRÁZEK 14: DESKA SMU .....                                                             | 28 |
| OBRÁZEK 15: TRANZISTOROVÝ DĚLIČ S POUŽITÍM 1 TRANZISTORU (ODHAD NAPĚTÍ V OBVODU) .....  | 32 |
| OBRÁZEK 16: TRANZISTOROVÝ DĚLIČ S POUŽITÍM 2 TRANZISTORŮ (ODHAD NAPĚTÍ V OBVODU) .....  | 32 |
| OBRÁZEK 17: TRANZISTOROVÝ DĚLIČ S POUŽITÍM 3 TRANZISTORŮ (ODHAD NAPĚTÍ V OBVODU) .....  | 33 |
| OBRÁZEK 18: BLOKOVÉ SCHÉMA ZAPOJENÍ PRO ŘÍDÍCÍ JEDNOTKU DESKY PECONT .....              | 34 |
| OBRÁZEK 19: HARVARDSKÁ ARCHITEKTURA [11] .....                                          | 35 |
| OBRÁZEK 20: REALIZOVANÁ ŘÍDÍCÍ JEDNOTKA PRO DESKU PECONT .....                          | 36 |
| OBRÁZEK 21: PŘÍKLAD NASTAVENÍ PINŮ PRO UART V PROGRAMU STM32CUBEMX .....                | 37 |
| OBRÁZEK 22: UKÁZKA DEKLARACE PROMĚNNÝCH - SYSTÉMOVÁ A UŽIVATELSKÁ ČÁST .....            | 38 |
| OBRÁZEK 23: VÝVOJOVÝ DIAGRAM PROGRAMU .....                                             | 42 |

## Seznam tabulek

|                                                         |    |
|---------------------------------------------------------|----|
| TABULKA 1: SEZNAM KÓDŮ ZÁKLADNÍCH FUNKCÍ .....          | 22 |
| TABULKA 2: SEZNAM KÓDŮ CHYBOVÝCH FUNKCÍ A VÝJIMEK ..... | 22 |

## Seznam zkratek a pojmů

|           |                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------|
| ADU       | Application Data Unit, aplikační datová jednotka                                                        |
| ASCII     | American Standard Code for Information Interchange, standardizovaná tabulka znaků                       |
| Cache     | vyrovnávací paměť mezi různě rychlými moduly                                                            |
| CRC       | Cyclic Redundancy Check, kontrolní součet v MODBUS komunikace                                           |
| DC        | Direct Current, stejnosměrný proud                                                                      |
| EEPROM    | Electrically Erasable Programmable Read-Only Memory, elektronicky mazatelná programovatelná paměť       |
| FLASH     | elektronicky mazatelná programovatelná paměť                                                            |
| FVE       | fotovoltaická elektrárna                                                                                |
| $I_p$     | maximální proudový rozsah                                                                               |
| $I_{PN}$  | jmenovitý proud                                                                                         |
| LED       | Light-Emitting Diode, svítivá dioda                                                                     |
| LRC       | Longitudinal Redundancy Check, kontrolní součet v MODBUS komunikaci                                     |
| Master    | prvek řídící jednosměrnou komunikaci v modelu master/slave                                              |
| MCU       | mikrokontrolér                                                                                          |
| MODBUS    | otevřený protokol pro komunikaci po sériové lince                                                       |
| MPPT      | Maximum Power Point Tracking, schopnost měniče sledovat bod maximálního výkonu solárního panelu         |
| PDU       | Protocol Data Unit, protokolová datová jednotka                                                         |
| PLC       | Programmable Logic Controller, programovatelný logický automat                                          |
| PV        | Photo Voltaic, fotovoltaický                                                                            |
| RAM       | Random Access Memory, operační paměť počítače                                                           |
| RTU       | Remote Terminal Unit, binární režim přenosu dat v MODBUS komunikaci                                     |
| Slave     | prvek v modelu master/slave, jenž neřídí jednosměrnou komunikaci                                        |
| String    | dva a více solárních panelů v sériovém zapojení                                                         |
| TCP/IP    | protokol pro komunikaci v počítačové síti                                                               |
| $U_{OUT}$ | výstupní napětí                                                                                         |
| $U_{REF}$ | referenční napětí bez zátěže                                                                            |
| USART     | Universal Synchronous / Asynchronous Receiver and Transmitter, zařízení pro komunikaci po sériové lince |

# 1. Úvod

Zájem o solární energii v minulých letech razantně stoupl a to nejen u velkých společností, zabývajících se výrobou elektřiny, jako například ČEZ, ale i v domácnostech a velkoobchodech. Využití solárního panelu vyžaduje pouze vstupní investici a dále už pak produkuje elektřinu bez potřeby jakékoli obsluhy vyjma občasných kontrol nebo servisních zásahů v případě poruchy. Solární energie je ale závislá na okolním prostředí a tím pádem není zcela stabilní. Proto se využívají různé měřicí desky, které s tímto proměnným generovaným napětím dále pracují.

Spousta společností i domácností v dnešní době používají a prodávají desky staršího data. Některé prošly poslední revizí před zhruba 10 lety. Z pohledu technologického pokroku se tak stávají zastaralými. I když jsou jistá provedení stále kvalitní, tedy návrh řešení a volba součástek při například využití napětí ze solárních panelů pro napájení samotné měřicí desky, samotná funkčnost již nemusí splňovat nové požadavky.

Cílem práce je navrhnout a realizovat desku, která nabídne náhradu či vylepšení stávajících měřících desek. A to ať už se jedná o funkčnost samotného měření, jednoduchost výměny modulů v případě poškození (například přepětím ze vstupu vlivem úderu blesku), anebo přidání nových funkcí.

Jelikož návrhy desek plošných spojů a jejich realizace nespádá pod specializaci oboru IT, vznikla tato práce v rámci kolaborace s vedoucím práce, který zajistil technické části týkající se elektroniky.

## 2. Problematika PV elektráren

Fotovoltaická elektrárna, známá také jako sluneční elektrárna či pod zkratkou PV elektrárna z anglického photovoltaic, je elektrárna, která k výrobě elektřiny využívá sluneční záření. Skládá se z mnoha stringů solárních panelů. Pod pojmem string si představme šňůru několika solárních panelů v řadě. Stringy se využívají z toho důvodu, protože samostatný solární panel nemá dostatečný výkon. Pokud ale připojíme několik solárních panelů do řady (tedy stringu), jejich společný výkon razantně stoupne.

Jako příklad velké solární elektrárny můžeme použít skupinu solárních elektráren pod označením FVE Ralsko. Tyto elektrárny jsou od sebe vzdáleny pouze jednotky kilometrů a nachází se v lokalitách Ralsko a Mimoň. Jejich celkový instalovaný výkon se udává 55,763 MW [3].



**Obrázek 1: Fotografie solární farmy [3]**

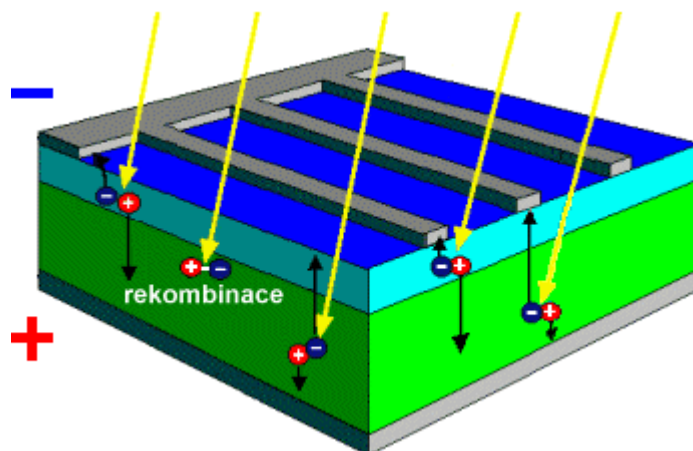
### 2.1. Součásti solární elektrárny

Solární elektrárna se skládá z několika částí. Tu nejznámější a rozlohou největší část tvoří solární panely. Za solární panely je připojen MPPT měnič pro optimální využití energie z panelů (kvůli kolísavému výstupu z panelů). Následuje střídač pro převod napětí ze stejnosměrného na střídavé a ochranné prvky (např. proti přepětí či zkratu). Poslední prvek v soustavě tvoří transformátor, který zajišťuje připojení elektrárny do sítě. So sestavy je možné připojit i nějaký druh baterie.

### 2.2. Princip solárních panelů

Solární panel se skládá z několika PV článků. Základem každého článku je polovodičová dioda obsahující 2 vrstvy polovodičů – typu P (anoda) a N (katoda). Jelikož se dioda skládá pouze ze dvou vrstev, nazývá se diodou typu P-N. Vrstva P obsahuje nedostatek

elektronů a vrstva N naopak přebytek. Jinými slovy se dá říct, že vrstva P obsahuje přebytek kladně nabitých „děr“.



Obrázek 2: P-N přechod solárního článku [1]

Dopadem slunečního záření vzniká k tzv. fotoelektrickému jevu, kdy dochází k uvolňování elektronů z obou vrstev vlivem elektromagnetického záření, které je součástí viditelného světla. Mezi vrstvami tak vznikne elektrické napětí o hodnotě přibližně 0,5 V. Pro navýšení tohoto napětí se lze připojit další články do sériového zapojení. Připojením článků do paralelního zapojení se naopak zvýší výstupní proud. V praxi se solární panel skládá ze sérioparalelního zapojení článků.

Aby došlo k uvolnění elektronů, musí mít dopadající foton ze slunečního záření dostatečnou energii, aby prošel zakázaným pásmem. U křemíku je to 1,12 eV [elektronvolt], což odpovídá 1105 nm [1]. Záření o delší vlnové délce projde skrz a je ignorováno, kdežto záření o kratší vlnové délce vyvolá fotoelektrický jev, kdy vznikne elektron a „díra“. Zbytek energie se převede na teplo, čímž vznikají ztráty a dochází tak k poklesu účinnosti přeměny solární energie na elektrickou. Teoreticky lze přeměnit na elektřinu až 50 % dopadajícího záření, v praxi se ale hodnoty pohybují do zhruba 16 %. Je možné dosáhnout i vyšších hodnot (24 % pro křemík), ale cena takových článků je příliš vysoká pro běžné použití.

### 2.3. Využití solárních panelů

V době solárního boomu došlo k rozmachu solárních panelů. Spousta domácností ale i obchodních domů a fabrik si tak na svém pozemku postavila nějaké množství solárních panelů, které jim zajišťují neustálou výrobu energie pro vlastní potřeby. Nejčastěji to je výroba energie například pro ohřev vody či osvětlení pozemku. Pro ohřev vody je ale lepší využít termických panelů, které převádí sluneční záření přímo na teplo.

Další využití mají v solárních elektrárnách, které nabízejí množství výhod i nevýhod oproti ostatním, například uhelným nebo jaderným elektrárnám.

Typicky se ale solární panely používají u vesmírných družic a stanic. V současnosti není k dispozici jednoduchý a levný způsob, jak dopravovat elektřinu do vesmíru, a tak mají stanice nainstalovány solární kolektory pro generování vlastní elektřiny, kde zároveň nedochází ke vzniku nežádoucího odpadu. Ve vesmíru navíc dosahují solární panely mnohem vyšší účinnosti [2], než na povrchu Země, neboť nedochází k ovlivnění okolním prostředím, jako například vysoká oblačnost.



**Obrázek 3: Vzhled Mezinárodní vesmírné stanice po jejím dokončení v roce 2011 [9]**

## 2.4. Výhody a nevýhody

Výhodou solárních panelů je zdroj jejich pro výrobu energie. Slunce se dnes (a pokud se lidstvu nepovede jeho životnost razantně zkrátit, tak dalších 4 až 5 miliard let bude) považuje za nevyčerpatelný zdroj, a tak solární energie spadá do kategorie obnovitelných zdrojů. Dále při převodu energie neprodukuje žádné nežádoucí odpadní látky (jako například zplodiny po spálení fosilních paliv či radioaktivní odpad z jaderných elektráren), a nedochází tak k znečištění životního prostředí. Dále jde o bezhlučný provoz a není potřeba obsluhy během provozu (pouze v případě poruchy či kontroly). Při překročení jejich životnosti je navíc možné panely i nadále využívat, rozdíl bude ale znát v poklesu jejich výkonu.

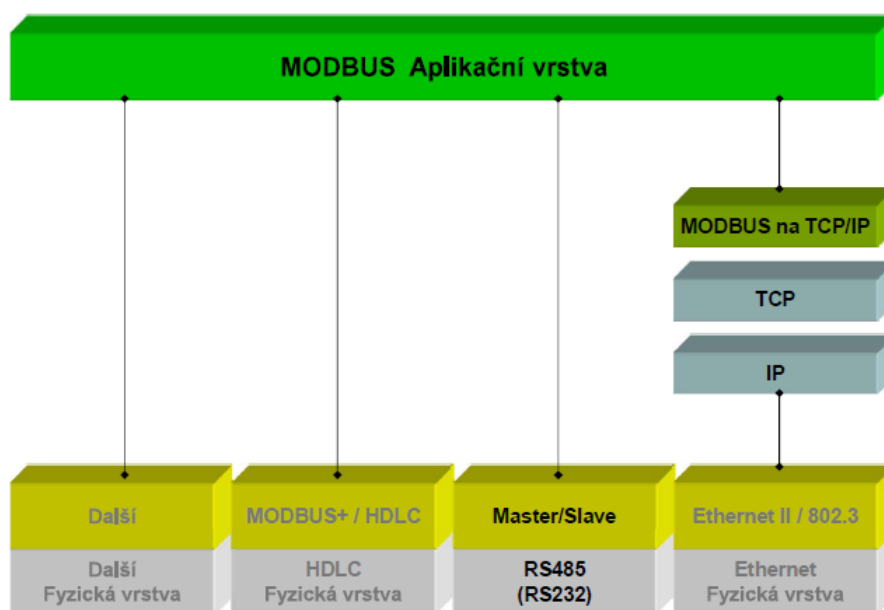
Každá výhoda ale přináší i nevýhodu. Výkon solárních panelů je nízký oproti jiným dostupným zdrojům energie a navíc také kolísavý. Aby byl výkon dostatečný, musí zaujímat velké prostranství, které často bývalo úrodnou půdou, tudíž zemědělci přicházejí o pole, což může dále ovlivnit ekonomiku státu. Pořizovací cena je stále poměrně vysoká a navíc se po dosloužení svých let musí recyklovat, nestačí je pouze odložit na skládku a nechat ladem.



Životnost je ve většině případech zaručena na pouhých 25 let, kdy jejich účinnost neklesne pod 80 %. Okolní prostředí (např. teplota, oblačnost atd.) a také úhel, pod kterým dopadá sluneční záření na panel, razantně ovlivňují jeho účinnost. V neposlední řadě je zde také jejich malá robustnost, kdy porucha jednoho článku může ochromit celý string panelů, případně prasklina na skle sníží účinnost celého panelu. Další ne zcela podložený problém je náročnost jejich výroby, kdy se údajně využije mnohem více energie (často z fosilních paliv, které spálením škodí životnímu prostředí) na jejich výrobu, než je výsledného užitku. Tento problém by se dle mého názoru dal vyřešit volbou jiného zdroje energie pro výrobu, ale dnešní svět se často ubírá cestou nejmenšího odporu.

### 3. Sběrnice MODBUS

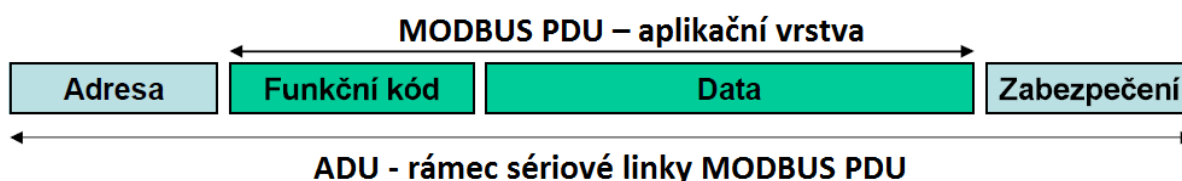
MODBUS je otevřený protokol zajišťující komunikaci mezi zařízeními po sběrnících a sítích jako například Ethernet TCP/IP či sériové linky RS-232, RS-485 a další. Nejčastěji se používá u dotykový displejů, PLC zařízení apod. Komunikace je postavena na předávání informací mezi dvěma stanicemi v zapojení master a slave. Master funguje jako klient, který řídí veškerou komunikaci mezi zařízeními. Slave je jedno z N zařízení (v zapojení jako server), se kterým master v danou chvíli komunikuje. Protokol MODBUS pracuje na aplikační vrstvě OSI modelu. Tato práce se zaměří na sériovou asynchronní komunikaci.



Obrázek 4: MODBUS aplikační vrstva [4]

#### 3.1. Aplikační protokol

Protokol MODBUS je definován PDU (Protocol Data Unit). Jde o samostatný rámec protokolu, který je neměnný a je definován na aplikační vrstvě OSI modelu. To jej odděluje od komunikace na spodních vrstvách OSI modelu. V praktickém zapojení se pak k rámci přidávají další informace, které tvoří ADU (Application Data Unit). ADU se tedy dle použití sítě může měnit.



Obrázek 5: Datový rámec MODBUS [4]

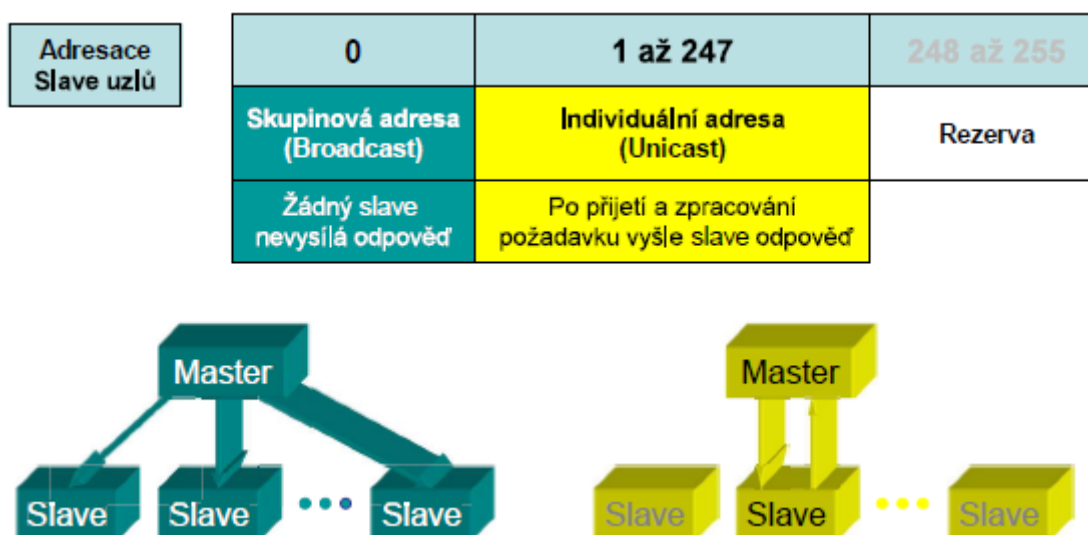
Funkční kód zde obsahuje informaci o akci, kterou má slave vykonat (např. číst/psát data). Pole Data obsahuje samostatná data, se kterými se v komunikaci pracuje. Například pokud si má slave zapsat určitou hodnotu, bude zde tato hodnota uvedena. V některých případech ale nemusí komunikace obsahovat žádná dodatečná data, což může být akce reset do továrního nastavení. Samostatná délka celého ADU je limitována fyzickou sítí. Pro komunikaci přes RS-485 jde o délku 256 bytů.

### 3.2. Komunikace

Komunikaci zahajuje vždy master a je založena na bázi žádostí a odpovědí (v případě komunikace s konkrétním slavem). Při vytváření ADU se zvolí adresa slavu, pro který je daná zpráva určena. Samotný master adresu nemá. Poté se za zprávu přidá PDU s funkcí a případnými daty, kterou má slave vykonat. Nakonec se vytvoří kontrolní součet, který slouží k ošetření správnosti zprávy.

Adresa má velikost 1 byte a obsahuje hodnoty od 0 do 247. Zbýlých 8 hodnot jsou rezervní. Adresa s hodnotou 0 je takzvaný Broadcast. Zpráva s touto adresou je určena pro všechna zařízení (slavy) a slouží pouze pro poslech. Žádné ze zařízení na tuto zprávu neodpovídá – jedná se o informativní rozhlas.

Pro samotná zařízení tedy zbývají adresy 1 až 247. Jelikož každé zařízení v jednom zapojení musí mít unikátní adresu, lze k jednomu masteru připojit maximálně 247 slávů. A aby nedocházelo ke kolizím, může v jednom zapojení být maximálně 1 master.

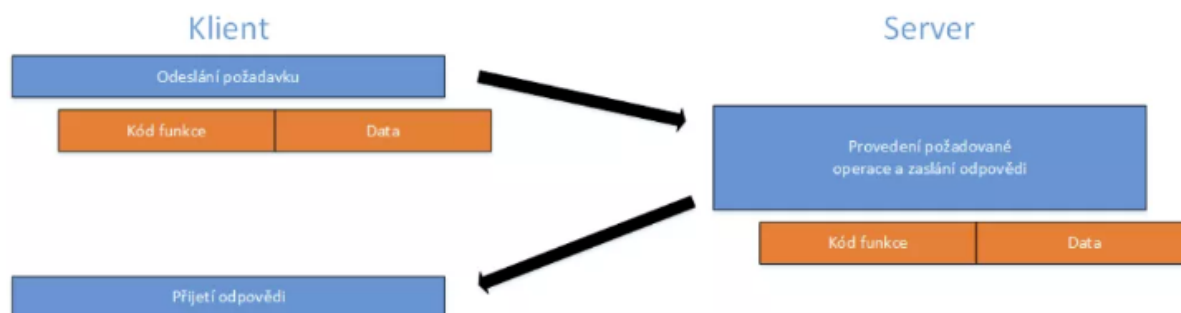


Obrázek 6: Adresace rámce ADU [4]

Jak již bylo zmíněno dříve, samotná komunikace probíhá na bázi žádostí a odpovědí. Master pošle požadavek na konkrétní slave, který vykoná požadovanou akci. Poté slave

vytvoří ADU a odešle masteru zpět odpověď. Jde tedy o dvoucestnou komunikaci. Vytvořená ADU se skládá z adresy slavu, kódu prováděné funkce, dat a zabezpečení. Zabezpečení se pro zjednodušení dá nazvat kontrolním součtem, byť to není zcela pravda. Jedná se o dokonalejší systém.

Pokud nastane chyba, ať už v přenosu dat nebo při vykonávání požadované operace, vrátí slave stejný kód funkce, který přijal v požadavku, a nejvyšší bit nastaví na hodnotu 1. Nastavení chyby se tedy provede přičtením hexadecimální hodnoty 0x80 ke kódu funkce.



Obrázek 7: Příklad úspěšné transakce [10]

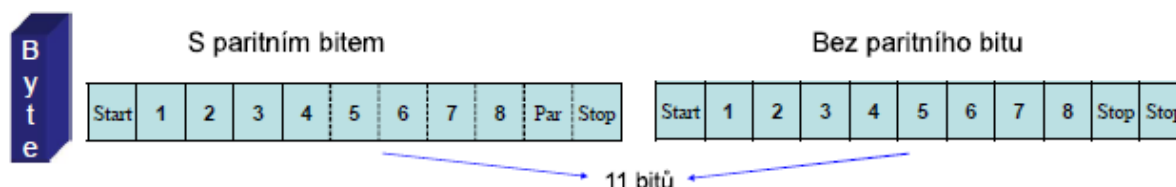
### 3.3. Režim přenosu komunikace po sériové lince

Režim přenosu určuje, v jakém digitálním formátu se data posílají. Přenos po sériové lince se dělí na 2 režimy:

- Binární (RTU – Remote Terminal Unit)
- Znakový (ASCII)

#### 3.3.1. Binární režim

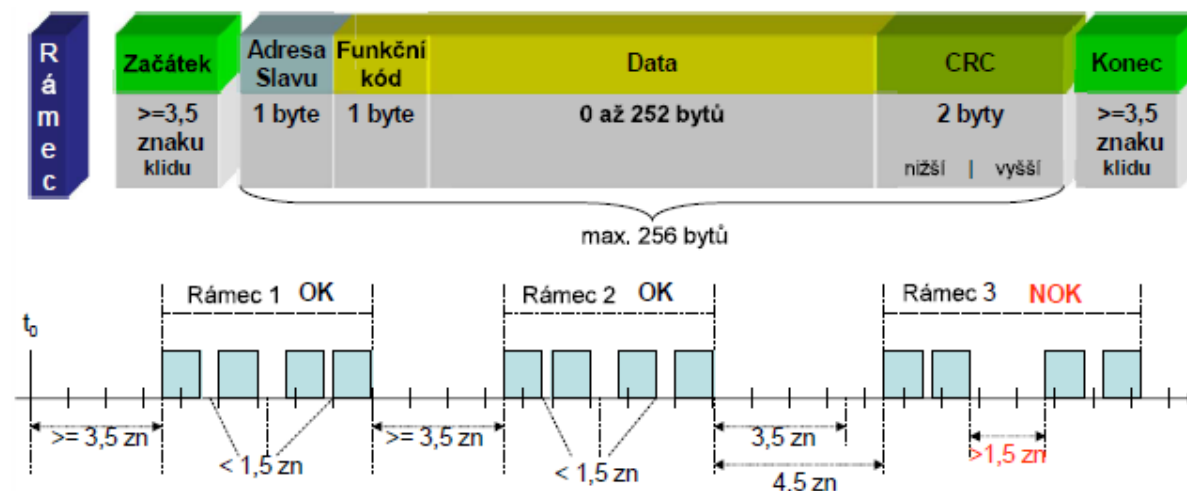
U binárního režimu obsahuje každá zpráva 11 znaků. Konkrétně jde o znaky začátku (start bit), 8 datových znaků, parity a konce (stop bit). Dle volby může probíhat komunikace s paritou nebo bez parity. V případě přenosu bez parity se paritní bit nastaví jako koncový.



Obrázek 8: Nastavení paritního bitu zprávy RTU režimu [4]

Rámec, jak již bylo zmíněno dříve, se skládá až ze 4 datových bloků. Konkrétně z adresy (1 byte), funkčního kódu (1 byte), dat (0 až 252 bytů) a zabezpečení (CRC, 2 byty). Mezi každým tímto datovým blokem musí být mezera nejdéle 1,5 znaku, jinak dojde

k chybnému přenosu. Mezi každým rámcem pak musí být mezera nejméně 3,5 znaků. Dále také platí podmínka, že přenos zprávy musí být kontinuální,

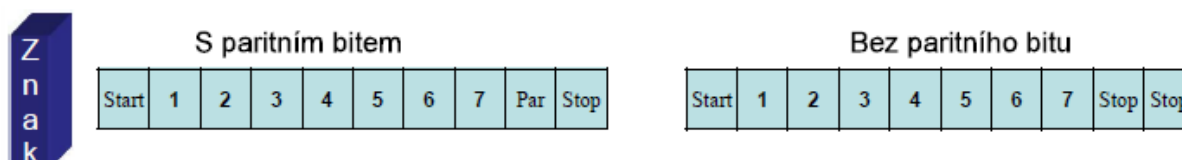


Obrázek 9: Rámec binárního režimu s příkladem přenosu [4]

Pro zabezpečení binárního režimu se používá CRC (Cyclic Redundancy Check) a to tak, že odesílatel zprávy provede výpočet nad odesílanými daty a přidá ho jako zabezpečení do rámce. Příjemce zprávy pak provede nad obdrženými daty kontrolní součet, a pokud sedí se zaslanou hodnotou, jsou data s největší pravděpodobností v pořádku. Samotný výpočet CRC ve zjednodušeném podání probíhá tak, že se vezme bitová posloupnost vstupních binárních dat, která je postupně (cyklicky) dělena předdefinovanou (generující) posloupností. Obě hodnoty lze zapsat polynomiálně, a tak se při výpočtu využívá matematických vlastností polynomů.

### 3.3.2. ASCII režim

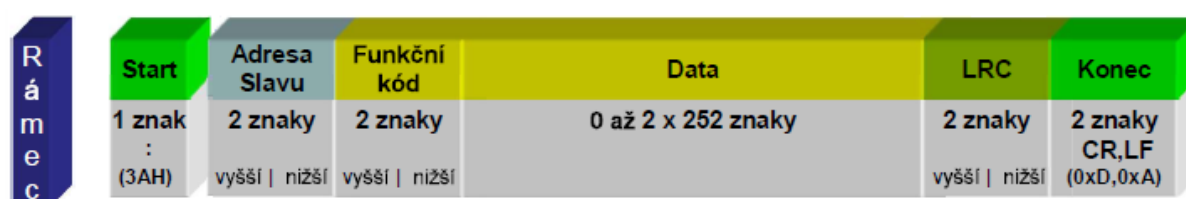
Každá zpráva ASCII režimu se skládá z 10 znaků. Obdobně jako v případě binárního režimu se skládá ze znaků začátku (start bit), 7 datových znaků, parity a konce (stop bit). Dle volby může probíhat komunikace s paritou nebo bez parity. V případě přenosu bez parity se paritní bit nastaví jako koncový.



Obrázek 10: Nastavení paritního bitu zprávy ASCII režimu [4]

Stejně jako v případě RTU režimu se rámec skládá ze 4 datových bloků, které se ale liší ve velikosti. Adresa i funkční kód jsou reprezentovány 16bitovou hodnotou (2 byty). Data

mohou být o velikosti 0 a  $2 \times 252$  znaků. Zabezpečení má stejnou velikost s tím rozdílem, že je zde použito kódování LRC (Longitudinal Redundancy Check). V komunikaci je každý byte reprezentován dvěma znaky. To znamená, že znak „K“, reprezentován hexadecimální hodnotou 0x4B ( $01001011_2$ ) bude uložen jako znak „4“ (0x34,  $00110100_2$ ) a znak „B“ (0x42,  $01000010_2$ ), tedy 00110100 01000010. Komunikace je tedy pomalejší, ale umožňuje mezeru až 1 vteřinu mezi zprávami, než dojde k chybnému přenosu. Reprezentace 16bitových hodnot je ve formátu Big-endian, tedy jednotlivé byty jsou v pořadí vyšší byte, nižší byte. Oproti RTU režimu není začátek a konec rámce omezen časovou prodlevou, ale konkrétními znaky. V případě začátku je to znak dvojtečky (0x3A) a v případě konce jde o znaky CR a LF (0xD, 0xA) ve stejném pořadí.



Obrázek 11: Rámec ASCII režimu [4]

Výpočet LRC je podstatně jednodušší, než výpočet CRC. V prvním kroku se sečtou byty adresy, funkčního kódu a dat ještě před převedením do ASCII. Adresa apod. se tedy skládají pouze z 8bitové hodnoty (nejsou ještě rozděleny do 2 znaků). Veškeré hodnoty, které se vlivem sčítání přenesou nad pozici 8. bitu, se zahodí. Takto získanou 8bitovou hodnotu znegujeme pomocí dvojkového doplňku (znegujeme hodnoty a poté přičteme jedničku) a dostaneme LRC byte, který převodem do ASCII rozdělíme na 2 byty. Součet celé zprávy včetně LRC tedy vyjde nula.

### 3.4. Seznam základních a chybových funkcí protokolu MODBUS

**Tabulka 1: Seznam kódů základních funkcí**

| <b>Kód</b> | <b>Název funkce</b>      | <b>Popis</b>                |
|------------|--------------------------|-----------------------------|
| <b>1</b>   | Read Coils               | Čtení stavu cívky           |
| <b>2</b>   | Read Discrete Inputs     | Čtení stavu vstupů          |
| <b>3</b>   | Read Holding Register    | Čtení holding registrů      |
| <b>4</b>   | Read Input Registers     | Čtení vstupních registrů    |
| <b>5</b>   | Write Single Coil        | Nastavení stavu jedné cívky |
| <b>6</b>   | Write Single Register    | Nastavení jednoho registru  |
| <b>15</b>  | Write Multiple Coils     | Nastavení více cívek        |
| <b>16</b>  | Write Multiple Registers | Nastavení více registrů     |

**Tabulka 2: Seznam kódů chybových funkcí a výjimek**

| <b>Kód</b> | <b>Název funkce</b>                     | <b>Popis</b>                                                                                                      |
|------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>1</b>   | Illegal Function                        | Neznámá funkce                                                                                                    |
| <b>2</b>   | Illegal Data Address                    | Vybraná adresa je neplatná nebo žádný ze slavů tuto adresu nemá                                                   |
| <b>3</b>   | Illegal Data Value                      | Slave danou hodnotu nepřijme                                                                                      |
| <b>4</b>   | Slave Device Failure                    | Chyba slavu při provádění operace, slave je zaseklý                                                               |
| <b>5</b>   | Acknowledge                             | Slave přijal požadavek, ale na vykonání potřebuje více času – ochrana před chybou vlivem vypršení časového limitu |
| <b>6</b>   | Slave Device Busy                       | Slave je zaneprázdněn operací, master by se měl ozvat později                                                     |
| <b>7</b>   | Negative Acknowledge                    | Slave nemůže provést operaci                                                                                      |
| <b>8</b>   | Memory Parity Error                     | Slave detekoval chybu paritního bitu                                                                              |
| <b>10</b>  | Gateway Path Unavailable                | Pro MODBUS brány – chybně nakonfigurovaná brána                                                                   |
| <b>11</b>  | Gateway Target Device Failed to Respond | Pro MODBUS brány – posláno když dotazovaný slave neodpověděl                                                      |

## 4. Paměti Flash a SRAM

V dnešní době existuje mnoho druhů pamětí, které se dělí mnoha způsoby. Jedno z dělení je na volatilní a nevolatilní. Volatilní paměť je taková paměť, která v případě ztráty napájení přijde o všechna data v rámci maximálně desítek milisekund. Příkladem je operační paměť RAM. Nevolatilní paměti, jak už název napovídá, si data uchovávají i po odpojení napájení. Typickým příkladem je paměť ROM. Další dělení jsou dle možnosti čtení a zápisu či typu přístupu.

V této kapitole jsou popsány paměti Flash a SRAM, neboť jsou nejčastěji využívány moderními mikrokontroléry.

### 4.1. Paměť Flash

Princip paměti Flash je postaven na bázi paměti EEPROM, jedná se tedy o nevolatilní elektricky mazatelnou a programovatelnou paměť. Paměťové buňky jsou strukturovány do bloků, se kterými lze samostatně pracovat. Na rozdíl od paměti EEPROM to přináší výhodu při mazání, neboť se smažou pouze konkrétní bloky a nikoli celá paměť.

Paměťové buňky této paměti se skládají z unipolárního tranzistoru se dvěma hradly, řídicím a plovoucím. Hlavní význam plovoucího hradla je otevření unipolárního tranzistoru. Pokud je buňka nenaprogramovaná, je hradlo bez náboje a neumožňuje otevření tranzistoru. Na buňce je tak logická hodnota 1. Přivedeme-li náboj na plovoucí hradlo, dojde k otevření tranzistoru (naprogramování) a buňka se nastaví na logickou hodnotu 0 [7].

Výhodami Flash paměti je její nízká výrobní cena, odolnost vůči poškození a možnost libovolně zapisovat a přepisovat. Vnitřní rozdělení do bloků také zajišťuje, že při mazání zůstanou ostatní data neporušená. Z toho důvodu je tento typ paměti hojně využit jako vnitřní úložiště v telefonech, přenosné Flash disky či jako součást SSD disků.

Nevýhodou této paměti je omezený počet zápisů na jednotlivé buňky a její životnost. Konkrétní hodnota počtu zápisů není známá, ale běžně se uvádí kolem 10 000. Životnost se udává kolem 10 let, kdy vlivem fyzikálních zákonů buňky ztratí svůj náboj.

### 4.2. Paměť SRAM

Zkratka SRAM znamená Static Random Access Memory. Princip funkčnosti je postaven na bistabilním klopném obvodu, kdy se paměťová buňka může nacházet právě v jednom ze dvou stavů [8]. Tím buňka určuje, zdali je v ní uložena binární hodnota 1 nebo 0.



Pojem statická tedy znamená, že paměť udržuje svou hodnotu po celou dobu, co je připojena ke zdroji napájení. Jedná se tedy o volatilní typ paměti.

Dynamické paměti fungují na principu elektrického náboje na kondenzátoru, který se postupně vybíjí, i když je paměť připojena ke zdroji napájení. Z toho důvodu se musí hodnota na buňce pravidelně obnovovat. Tento problém u statických pamětí nenastává.

Hlavní výhodou paměti SRAM je její přístupová rychlost, a tak se používá především u pamětí cache (vyrovnávací paměť mezi dvěma různě rychlými zařízeními), které mají obvykle násobně nižší velikost, než ostatní paměti. V dnešních standardech se paměť cache procesorů pohybuje řádově ve stovkách kilobytů až jednotkách megabytů, zatímco operační paměť RAM dosahuje velikosti jednotek až desítek gigabytů. Další výhodou, jak je zmíněno v předchozím odstavci, je absence potřeby obnovy dat.

Nevýhodou je složitost samotného zapojení a výroby, což zvyšuje cenu, výrobní náročnost a samotnou fyzickou velikost paměti.

## 5. Používané desky na pracovišti vedoucího

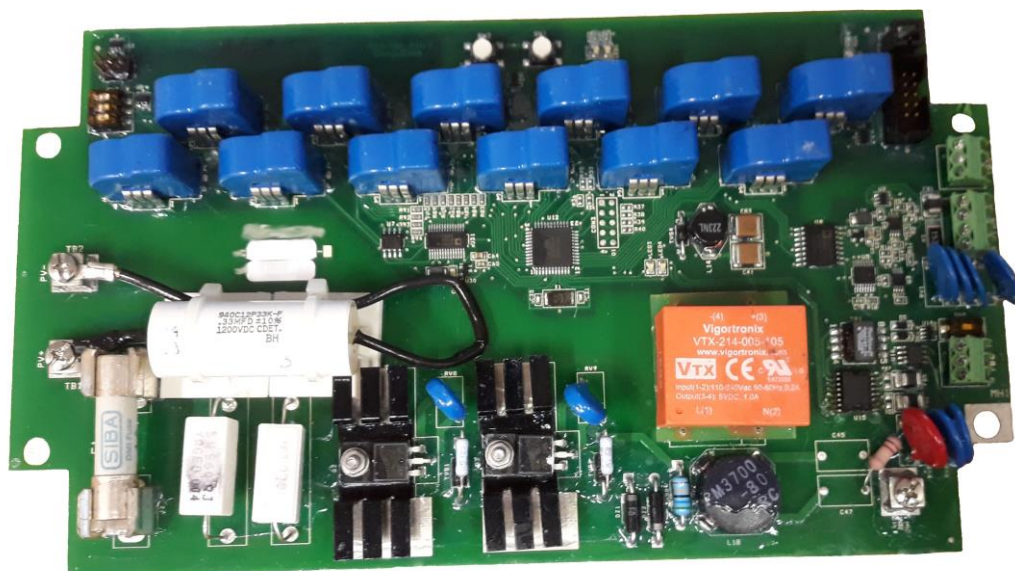
Tato kapitola se zaměřuje na desky aktuálně využívané pro měření solárních panelů. Práce vychází z těchto desek a snaží se navrhnout nové provedení či nahrazení stávajících částí desky za účelem vylepšení funkcí či nahrazení součástek dostupnější variantou pro případné budoucí opravy, dojde-li k poškození desky.

Na žádost vedoucího práce nejsou názvy desek ani názvy výrobců těchto desek uvedeny v této práci. Použitá označení desek jsou čistě interní.

### 5.1. Deska 1KVSCB

První použitá deska dostala označení 1KVSCB. Poslední revize této desky proběhla v první polovině roku 2009, což ji řadí v porovnání s dnešním rozvojem technologií mezi starší elektroniku.

Deska je navržena kvalitně, ale pro dnešní potřeby používá již zastaralé metody. Použité součástky je v dnešní době těžké sehnat, jsou příliš drahé nebo se musí kupovat v nepříjemném množství (řádově tisíce, zatímco požadované množství se pohybuje v řádu desítek). I když je pro desku navrhnuté zapojení paměti EEPROM, ve finální fázi není na desce k dispozici. Komunikace probíhá po sériové lince přes port RS-485.



Obrázek 12: Deska 1KVSCB

### 5.1.1. Zdrojová část

Deska je rozvrhnutá do 3 částí. První z nich tvoří napájecí část. Deska je napájena přímo ze solárních panelů, jejich výkon zároveň i měří. Jak již název napovídá, deska je koncipována pro vstupní napětí až 1 kV. Vstupní napětí je ošetřeno pojistkou proti přepětí a následně je pomocí dvou výkonových tranzistorů sníženo na přibližně 230 V. Následuje filtrace a pomocí měniče je napětí sníženo na 5 V, které deska využívá pro své napájení.

### 5.1.2. Měřicí senzory

Druhou část desky tvoří měřicí senzory. Skládají se z dvanácti proudových senzorů LTS 6-NP od firmy LEM a jednoho teplotního senzoru. LEM senzory měří proud v rozmezí 6 A s maximálním rozsahem až  $\pm 19,2$  A. Tato hodnota pak určí výstupní napětí ( $U_{OUT}$ ), které se posléze převede na digitální hodnotu pomocí 12bitového A/D převodníku. Ten dále posílá data do mikroprocesoru. Vzorec 1 vypočítává skutečné výstupní napětí ( $U_{OUT}$ ) v závislosti na maximálním proudovém rozsahu ( $I_P = \pm 19,2$  A) a jmenovitým proudem ( $I_{PN} = 6$  A). Referenční napětí  $U_{REF} = 2,5$  V bez zátěže.

$$U_{OUT} = U_{REF} \pm (0,625 \times I_P \div I_{PN}) \quad (1)$$

Teplotní senzor v tomto případě nemá žádnou zásadní funkci. Měří pouze teplotu desky pro kontrolu proti přehřívání.

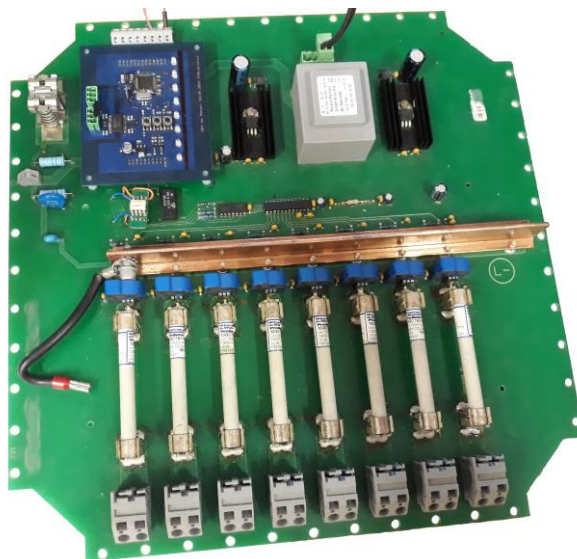
### 5.1.3. Zpracování dat a komunikace

Poslední část tvoří obvody zpracovávající naměřená data a zajišťující komunikaci. Ty se skládají z mikrokontroléru, dle návrhu paměti EEPROM, jenž na této verzi desky není přítomna (je nahrazena vnitřní pamětí EEPROM v mikrokontroléru), a portu RS-485. Paměť by měla sloužit k uložení naměřených dat ze senzorů, které se po připojení nahrají do masteru (například PC).

Mikrokontrolér je zde použit ATmega644. Jeho charakteristickými rysy jsou 4 KB paměti typu RAM, 3 časovače, 2 KB vnitřní EEPROM paměti a možnost komunikace po sériové lince a sběrnicích SPI a I<sup>2</sup>C. Mikrokontrolér je nastaven pro komunikaci jako slave, získává data z A/D převodníku pro každý string panelů zvlášť, na základě získané hodnoty vypočítá skutečnou naměřenou hodnotu a poté ji pošle po sériové lince RS-485 masteru.

## 5.2. Deska PECONT

Druhou deskou na pracovišti vedoucího je deska s označením PECONT. Její rozměry jsou podstatně větší než ostatní měřicí desky. Nabízí menší počet měřících senzorů s větším rozprostřením po desce, což pomáhá odstranit nežádoucí magnetické rušení při měření.



Obrázek 13: Deska PECONT s navrženou řídicí jednotkou

### 5.2.1. Zdrojová část

Napájení této desky je řešeno pomocí DC měniče pro napětí 42 V. Je tedy potřeba zajistit externí zdroj napětí nebo měnič, který poskytuje tento výstup. Ochrana proti přepětí například pomocí pojistky se na desce nevyskytuje.

### 5.2.2. Měřící senzory

Na této desce je použito celkem 8 senzorů LTS 6-NP, stejný typ jako používá deska 1KVSCB. Kabele ze solárních panelů jsou připojeny konektory přímo k desce, kde jsou pojistkami odděleny od samotných senzorů z důvodu ochrany senzorů. Zatím co u desky 1KVSCB se měří proudové pole přímo na kabelech, zde je to řešeno mechanicky na desce. Zatímco kladná polarita z panelů je přivedena na konektory k jednotlivým senzorům, záporná polarita je přivedena na kovový plát, který je společný pro všechny senzory.

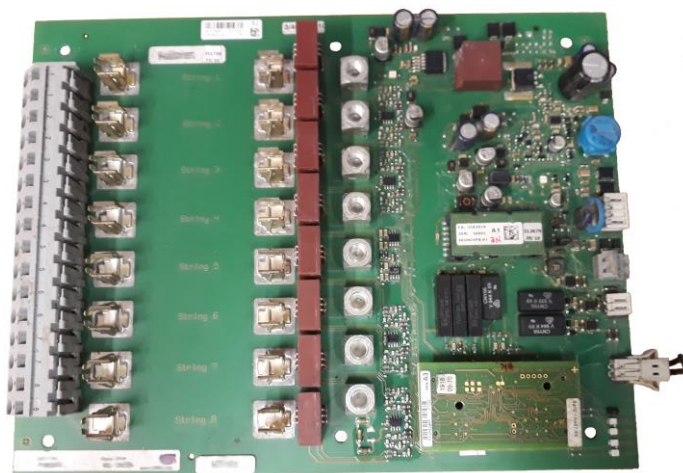
### 5.2.3. Zpracování dat a komunikace

Výstup měření vstupuje do A/D převodníku, který je součástí desky. Ten převede naměřenou analogovou hodnotu na digitální a posílá ji dále do řídicí jednotky. K řídicí

jednotce není známa žádná dokumentace, pouze že komunikace mezi jednotkou a PC probíhá po sériové lince. Z toho důvodu bylo rozhodnuto tuto jednotku nahradit vlastní.

### 5.3. Deska SMU

Třetí deska na pracovišti vedoucího projektu obdržela název SMU. Tato deska je pouze jednou částí ze setu monitorovacího zařízení, a tak je práce s ní poněkud složitější. K této desce se, stejně jako v případě desky PECONT, připojuje kladný výstup ze solárních panelů a pro připojení záporného pólu slouží druhá deska. Zem obou desek pro senzory je propojena pomocí přepět'ové ochrany (třetí jednotka setu).



Obrázek 14: Deska SMU

#### 5.3.1. Zdrojová část

Deska je navržena pro vstupní napětí 55 V stejnosměrného napětí. To je pak sníženo pomocí DC měniče na 5 V pro zbytek desky. Ošetření proti přepětí není řešeno na desce, ale pomocí jističe mimo desku (další jednotka setu).

#### 5.3.2. Měřicí senzory

Na desce jsou použité senzory VAC 4646–x661, vyráběny firmou Vacuumschmelze. Jmenovitý proud těchto senzorů ( $I_P$ ) je 25 A s výstupním referenčním napětím ( $U_{REF}$ ) 2,5 V bez zátěže. Maximální proudový rozsah je v rozmezí  $\pm 85$  A. Výpočet skutečného výstupního napětí ( $U_{OUT}$ ) je stejný jako u senzorů LTS 6-NP podle vzorce 1.

### 5.3.3. Zpracování dat a komunikace

Výstupní napětí ze senzorů vstupuje do vyměnitelného modulu, který je určen pro zpracování dat. Zde se nachází A/D převodník pro převod dat do digitální podoby a také mikrokontrolér, který data zpracovává a následně posílá dále.

Na desce se nachází další vyměnitelný modul, jenž slouží k zajištění komunikace mezi deskou a PC (či jiným zařízením pro sběr dat). Komunikace probíhá po sériové lince RS-485.

## 5.4. Porovnání desek, rozdíly

### 5.4.1. Volba napájení desky

Zásadní rozdíly desek jsou ve využití jednotlivých součástek, oddělení částí desky, ošetření vstupů a výstupů a také logika zapojení.

Hned první rozdíl je vidět u volby napájení desky. Zatímco 1KVSCB využívá napájení přímo ze solárních panelů (až 1 kV), desky PECONT a SMU potřebují externí napájecí zdroj. Volba napájení přímo ze solárních panelů je výhodou do velkých solárních elektráren. Odpadá zde potřeba rozvádět další napájecí zdroj po celém poli a tedy i finanční náklady spojené s tímto řešením. Nevýhodou je velký úbytek napětí při převodu na pracovní napětí desky 5 V, který se přeměňuje na teplo a pokud je deska uzavřena v malém prostoru, může a s největší pravděpodobností i bude docházet k přehřívání a tedy i k vysokým ztrátám. Samotný převod napětí je blíže popsán v kapitole návrhu desky.

Druhý způsob napájení je tedy využití externího zdroje. Mohou se využít například baterie, které poskytnou potřebné napájení, a tedy dojde k minimálním ztrátám, ale jejich doba životnosti je příliš krátká a pokud by byla deska uzavřena v krabici, jejich výměna by mohla být až zbytečně náročná. Navíc by toto řešení vyžadovalo implementaci kontroly jejich stavu, aby došlo k včasné výměně vybitých baterií. Každá výměna by navíc znamenala dočasný výpadek měření, což může být nežádoucí. Pro domácí kutily nebo dočasné měření (například výkon solárního autíčka pro závody) by ale bylo takového řešení vhodné.

Lepším řešením je využít rovnou rozšířeného a dostupného síťového napájení, tedy 230 V, které nalezneme v každé domácnosti. U tohoto zapojení odpadá oproti prvnímu řešení potřeba snižovat napětí pomocí děliče a tak lze rovnou využít měniče napětí pro snížení napětí na 5 V. Oproti bateriím tak odpadá problém s výměnou a kontrolou stavu baterií

a zároveň měření výkonu panelů probíhá bez výpadků (dokud nedojde k výpadku elektrické sítě). Toto řešení je nejvýhodnější při použití měření v domácnostech.

#### 5.4.2. Senzory, zapojení, výhody a nevýhody

Výstup solárních panelů prochází LEM senzory, které měří magnetický tok tvořený proudem pomocí Hallova článku. Měření je tak galvanicky oddělené. Každá deska má různý počet LEMů s různým rozestavěním (odstupem od sebe), což přináší jistá úskalí.

Výhoda 12 LEMů u desky 1KVSCB je zřejmá na první pohled, tedy dokáže najednou měřit data z více stringů. Kvůli dimenzím desky ale musí být senzory blíže u sebe a tak dochází k ovlivnění měření ostatními LEMy. Tento problém je vyřešen přímo na desce odstíněním rušivých elementů. Ovlivnění a nepřesnost měření ale nemají na svědomí pouze ostatní LEMy, ale samotný přívod měřených kabelů. Často se přivádějí najednou jako jeden tlustý kabel a pokud bude zapojení desky konstrukčně řešeno takovým způsobem, že tento kabel povede poblíž některého z LEMů, hardwarové ošetření k odstranění rušení již stačit nebude. Dojde tedy k nepřesnému měření, které je potřeba řešit softwarově.

Desky PECONT a SMU mají menší počet LEMů a tedy využívají většího rozestupu mezi nimi. Tím není zcela odstraněn vliv okolních LEMů při měření, ale je podstatně utlumen na téměř neznatelnou hodnotu. Stejně jako v předchozím případě může dojít k nepřesnostem měření vlivem přivedených měřených kabelů okolo LEMů, ale o tento problém se již stará mikroprocesor, který tyto odchylky řeší softwarově.

Zajímavé řešení zapojení senzorů má deska 1KVSCB. Zatímco ostatní desky zajišťují bezpečnost senzorů pomocí pojistek, tato deska řeší ochranu využitím tranzistoru před senzorem. Senzor tak měří vždy pouze v případě, kdy je měřicí deskou vyzván.

## 6. Návrhy desek

Jednotlivé návrhy desek se zaměřují na nahrazení stávajících částí desek, ať už z důvodu použití novějších technologií, vylepšení stávajícího řešení anebo použití levnějších či dostupnějších součástek.

### 6.1. Návrh napájení ze solárních panelů

Návrh první desky byl zaměřen na nahrazení systému napájení u desky 1KVSCB, tedy využití napájení ze solárních senzorů pro desku, za pomoci dostupnějších součástek a robustnějších součástek, aby se co nejvíce zvýšila životnost desky.

Aktuální řešení tvoří tranzistorový dělič složený ze tří Darlingtonových tranzistorů STP03D200. Hlavním rysem tohoto tranzistoru je napětí kolektor emitor 1,2 kV a vysoký proudový zisk 200 hFE. Tento dělič snižuje vstupní napětí na přibližně 230 V, které je následně pomocí měniče napětí sníženo na 5V logiku.

Abychom mohli s návrhem začít, musíme znát základní údaje pro desku. Běžné stringy na solárních farmách se skládají z 20 solárních panelů. Každý panel vygeneruje napětí o hodnotě až 40 V. Jednoduchým vzorcem (2) tedy vypočítáme vstupní napětí 800 V, ale v rámci ochrany před přepětím bylo po konzultaci s vedoucím projektu rozhodnuto vycházet z desky 1KVSCB a navrhnout tak desku pro vstupní napětí 1 kV.

$$20 \times 40 = 800 \quad (2)$$

#### 6.1.1. Měnič napětí

Nejjednodušší řešení je využít DC/DC měnič, který sníží napětí na hodnotu přibližně 230 V nebo v ideálním případě rovnou na 5 V, čímž by odpadla potřeba druhého měniče. Jedná se sice o jednoduché řešení, ale v praxi zcela nepoužitelné. Na stránkách obchodů jako tme.eu, gme.cz, farnell.com a mouser.com nebylo možné najít požadovaný měnič. Tím odpadá požadavek dostupných součástek. Měniče pro takto vysoké napětí se neprodávají.

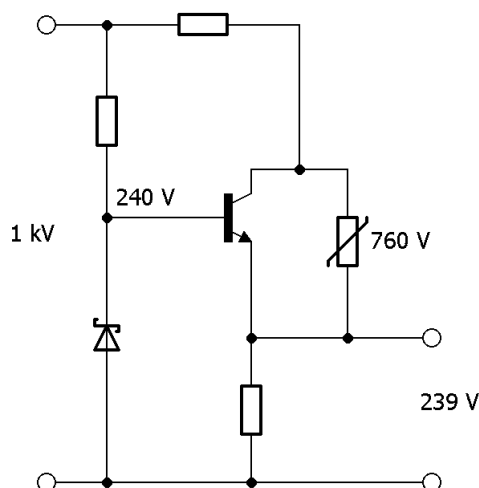
#### 6.1.2. Dělič napětí

Druhé řešení je použití tranzistorového děliče napětí, jako tomu je u desky 1KVSCB. Zde vyvstala otázka, zdali je lepší použít v zapojení jeden, dva nebo tři tranzistory.



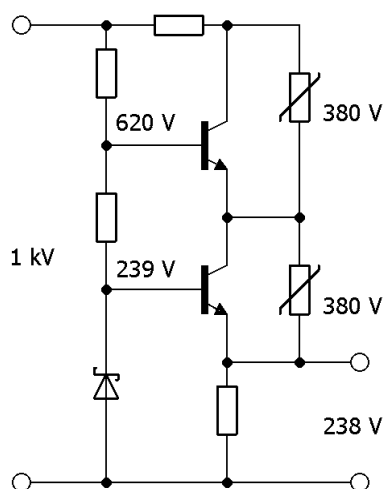
Na první pohled se použití jednoho tranzistoru zdá jako nejlepší cesta. Na desce to zabere málo místa a menší počet součástek vyjde levněji. Na tranzistoru sice bude vyšší úbytek napětí a dojde tak k většímu zahřívání, ale to se dá vyřešit správnou volbou chladiče.

Po návrhu zapojení se ale ukázalo, že úbytek je příliš velký a sehnat vysokonapěťový varistor pro napětí 760 V a zároveň sehnat chladič schopný jej uchládit v uzavřeném prostoru, kde s největší pravděpodobností není žádná cirkulace vzduchu, není snadné. Na obrázku 15 jsou vidět odhadnutá napětí, která by se v obvodu vyskytla.



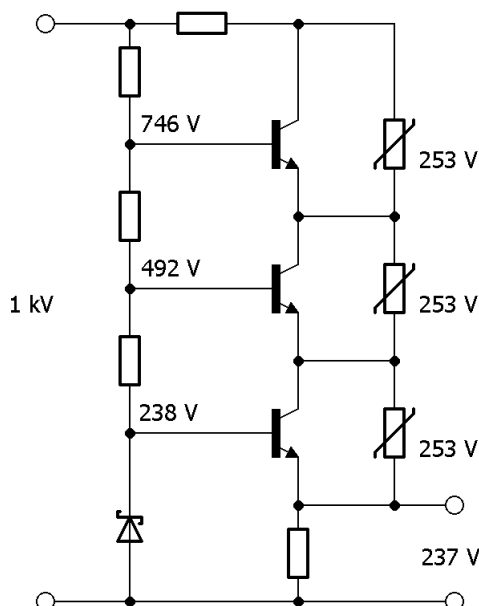
**Obrázek 15: Tranzistorový dělič s použitím 1 tranzistoru (odhad napětí v obvodu)**

Zapojení s použitím jednoho tranzistoru pro snížení napětí na hodnotu okolo 230 V je ve výsledku z praktického hlediska nepoužitelné. Při použití dvou tranzistorů, jak je na obrázku 3 vidět, již nároky pro varistory značně poklesly. Úbytek napětí se rozdělí do více součástek a ubude tak stres na každé z nich. Prodlouží se tím jejich životnost a zároveň je jednodušší se vypořádat s jejich teplotou.



**Obrázek 16: Tranzistorový dělič s použitím 2 tranzistorů (odhad napětí v obvodu)**

Zapojení se třemi tranzistory (obrázek 17) zvýší výhody předchozího zapojení, ale výsledek v porovnání s cenou součástek je ve výsledku stejný. Součástky pro obě zapojení jsou dostupné a to i v nízké cenové kategorii.



**Obrázek 17: Tranzistorový dělič s použitím 3 tranzistorů (odhad napětí v obvodu)**

Další důležitou součástí tohoto zapojení je dostatečně výkonný tranzistor, který si poradí s tak vysokým napětím. Zde se vyskytuje problém, neboť se musí vybírat tranzistor nejen pro vysoké napětí, ale i s velkým proudovým ziskem. Malý zisk znamená velké ztráty.

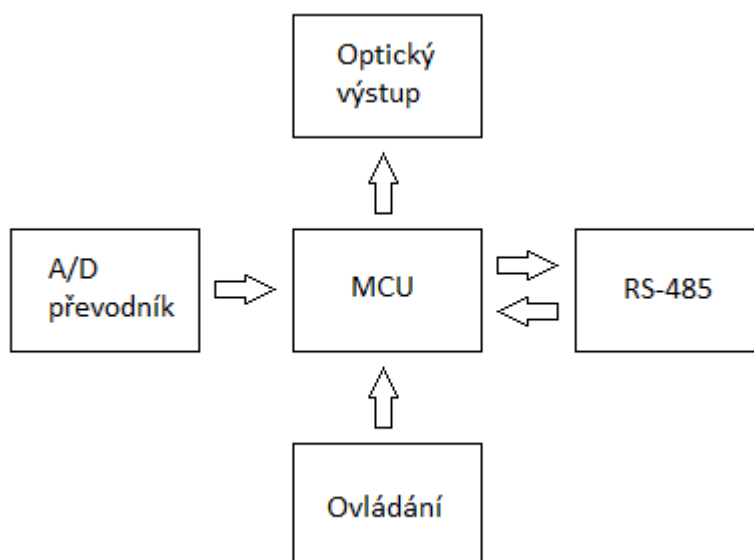
Jako náhrady místo použitého STP03D200 se vybraly tranzistory BU205 a STN0214 na základě jejich dostupnosti. Oba mají vysoké napětí kolektor emitor, konkrétně 700 V a 1k2 V, jenže jejich proudové zesílení  $h_{FE}$  je pouze 2 a 3, zatímco při návrhu zapojení se počítalo s Darlingtonovým tranzistorem, který má proudové zesílení  $h_{FE}$  až 400. Cena těchto tranzistorů se ale pohybuje o 2 řády jinde a tak je není možné použít. Pokud se podíváme na obrázek 17, můžeme si u prvního tranzistoru všimnout napětí vyššího než 700 V, což znamená, že tranzistor BU205 by nebylo možné použít.

Problém nastává u realizace této desky. Použití zmíněných tranzistorů by vyžadovalo obrovský chladič kvůli ztrátám vlivem malého proudového zesílení, což by bylo konstrukčně náročné. Deska se tedy nakonec nerealizovala. I když by šlo o cenově výhodné řešení, konstrukčně a výkonově by nevylepšila stávající řešení použité na desce 1KVSCB.

## 6.2. Řídící jednotka pro desku PECONT

Řídící jednotka pro tuto desku je black box, o kterém není dostatek informací. Na pracovišti vedoucího práce není k dispozici žádná dokumentace a nepovedlo se dohledat ani jiné informace k tomuto modulu. Proto bylo rozhodnuto navrhnout novou desku, která nahradí tuto část.

Prvním krokem návrhu je vytvořit blokové schéma desky, co všechno bude potřeba při výrobě zahrnout. Jádrem desky je mikrokontrolér (MCU), jehož funkcí bude zpracovávat naměřená data a komunikovat například s PC (předávání dat). Vstupní data jsou naměřené hodnoty z A/D převodníku, ovládací tlačítka (reset apod.) a data z RS-485. Výstupem je opět komunikace po RS-485 a optický výstup stavu desky.



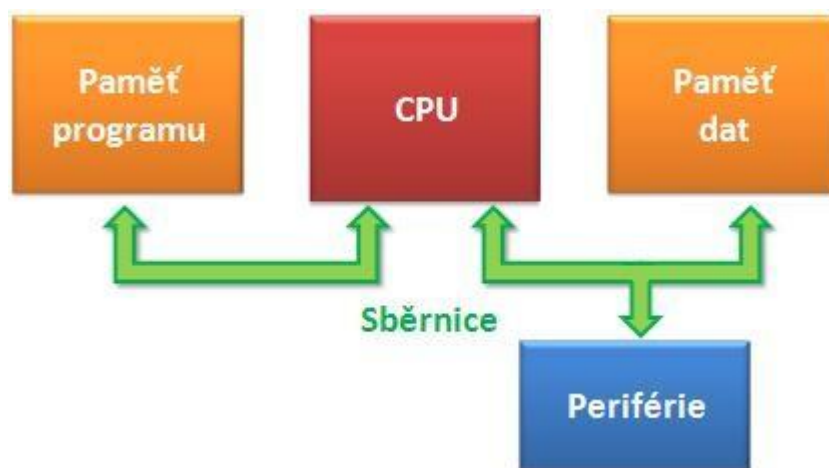
Obrázek 18: Blokové schéma zapojení pro řídicí jednotku desky PECONT

### 6.2.1. Mikrokontrolér

Pro tuto desku byl zvolen mikrokontrolér STM32F411RE vyráběný společností STMicroelectronics. Jádro mikrokontroléru je Cortex-M4 pracující na frekvenci až 100 MHz. Tento typ jádra používá Harvardskou architekturu s RISC instrukční sadou. Obsahuje integrovaný 12bitový A/D převodník s až 16 kanály. Dále nabízí šest 16bitových časovačů, dva 32bitové, dvakrát watchdog časovač a SysTick časovač (určený pro přerušení systémem softwaru). Dále nabízí 512 kilobytů paměti Flash a 128 kilobytů paměti SRAM [6].

Architektura počítače představuje konkrétní realizaci počítače na základě návrhu. Jako referenční architektura se dnes považuje Von Neumannova architektura publikovaná roku 1945. Definuje strukturu počítače, jenž je řízen programem, který je uložen v paměti.

Pro dnešní počítače není Von Neumannova architektura příliš vhodná kvůli společné paměti pro data a program, a tak se častěji používá architektura Harvardská. Hlavní rozdíl Harvardské architektury je oddělení paměti programu a dat. Výhodou je paralelizace, kdy je možné přistupovat k oběma pamětím zároveň. Zároveň to ochraňuje program, neboť nemůže přepsat sám sebe, a navíc může mít každá paměť jinou velikost. Nevýhodou je náročnější režie řízení procesorem nebo ne zcela využitá paměť. Kvůli oddělení pamětí nemůže být využita paměť dat pro program a naopak.



**Obrázek 19: Harvardská architektura [11]**

Architektura RISC (Reduced Instruction Set Computer) je označení pro procesory s redukovanou instrukční sadou. Nabízí jednoduchou a optimalizovanou sadu strojových instrukcí oproti architektuře CISC (Complex Instruction Set Computer), kde instrukce pokrývají širokou škálu funkcí, které lze nahradit již existujícími jednoduchými instrukcemi. Příkladem je například násobení (CISC), které lze nahradit sčítáním (RISC).

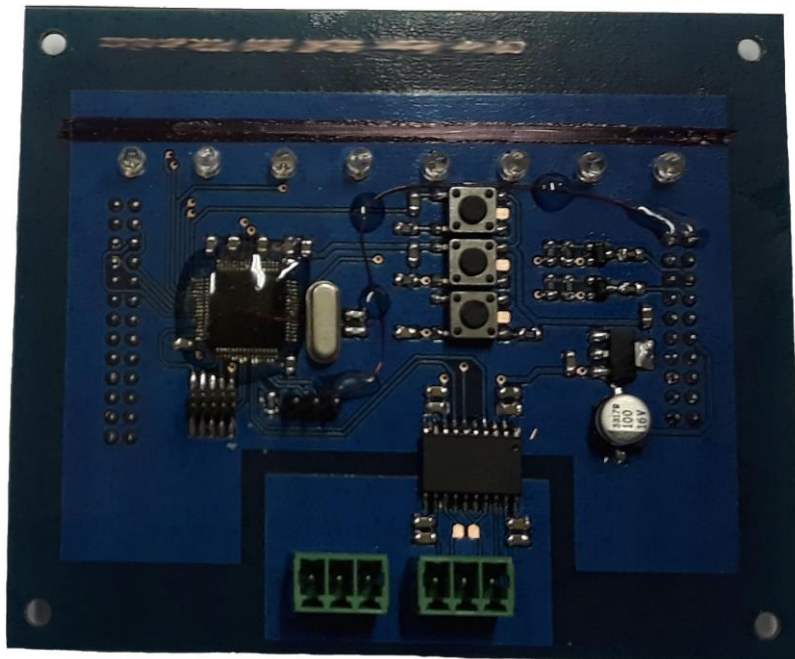
### 6.2.2. Realizace desky

Samotný návrh schéma zapojení a následná výroba desky probíhala pod dozorem vedoucího projektu v rámci kolaborace s dalšími studenty, neboť tato část nespadá pod specializaci oboru IT. Návrh desky byl založen na blokovém schéma, kde nakonec odpadlo zapojení A/D převodníku. Při podrobnějším prostudování desky PECONT bylo zjištěno, že A/D převodník nebyl součástí původní řídicí jednotky, jak bylo z počátků domníváno, ale je přímo součástí desky. Tedy externí ani integrovaný A/D převodník není použit a deska byla tomuto faktu přizpůsobena.

Pro ovládání desky byly přidány 3 tlačítka pro vstup do nastavení, změnu režimu módu, potvrzení výběru a reset mikrokontroléru.

Optický výstup stavu desky je řešen zapojením 8 LED diod. Tyto diody signalizují zapnutí/vypnutí desky, zvolený režim, přenos dat a podobně.

Sériová komunikace po sériové lince (sběrnice MODBUS) je galvanicky oddělena od zbytku zapojení, aby nedošlo k poškození hlavního jádra desky.



**Obrázek 20: Realizovaná řídící jednotka pro desku PECONT**

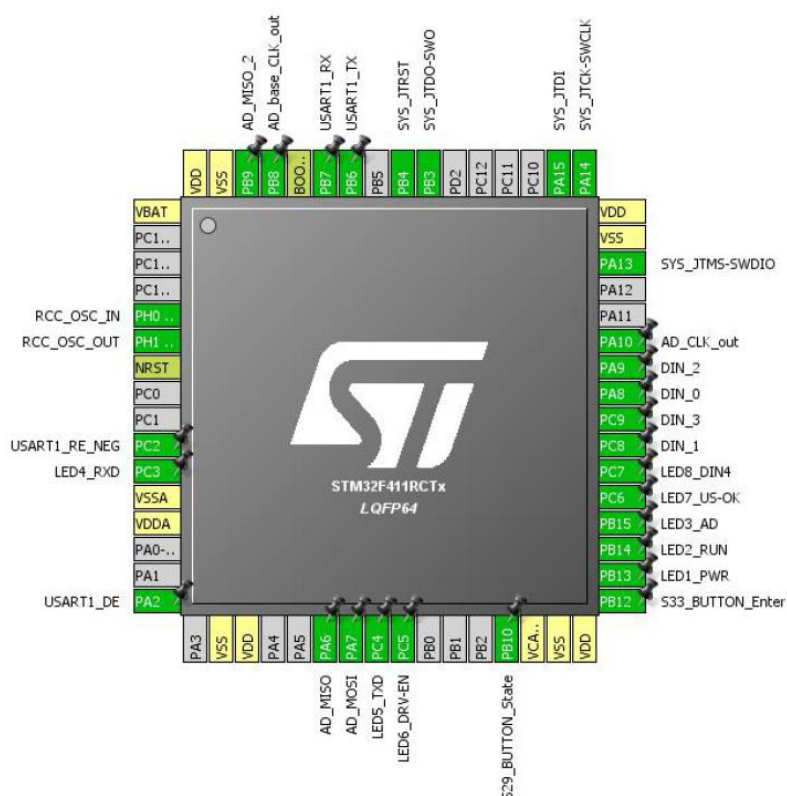
## 7. Program řídící jednotky

Program k této desce vznikl, stejně jako tvorba desky, v rámci kolaborace s vedoucím projektu. Mé zaměření bylo naprogramovat komunikaci po sběrnici MODBUS a výsledný projekt rozdělit na jednotlivé moduly za účelem zlepšení přehlednosti kódu v projektu.

### 7.1. STM32CubeMX

Projekt byl vytvořen v programu STM32CubeMX. Jde o volně dostupný program od společnosti STMicroelectronics pro konfigurování mikrokontrolérů. Tento program byl vytvořen za účelem usnadnění práce a ušetření času vývojářů. Nabízí grafické rozhraní pro snadnou konfiguraci mikrokontroléru v jazyce C. Pro programování samotného kódu je nutno využít programů třetí strany, jako například Visual Studio firmy Microsoft.

Při vytváření projektu se zvolí mikrokontrolér, pro který budeme program vytvářet. Program nám nabídne grafické zobrazení zvoleného mikrokontroléru s možností nastavení jednotlivých pinů (Pinout view).



Obrázek 21: Příklad nastavení pinů pro UART v programu STM32CubeMX

Na obrázku 21 si lze všimnout různých obarvení pinů. Žluté a světle zeleně zabarvené piny jsou neprogramovatelné piny. Jedná se o piny pro napájení, BOOT apod. Šedě zbarvené

piny reprezentují programovatelné piny, kterým nebyla přiřazena konkrétní funkce. Zelené zbarvení mají piny, kterým byla nastavena funkce, jež je také popsána vedle pinu, a jsou povolené (Enabled). V příkladu na obrázku 21 je to například externí oscilátor (piny PH0, PH1). Další obarvení, které není na obrázku vidět, je oranžové. To znázorňuje varování, že pin má přiřazenou hodnotu, ale není povolen (Disabled). Nastane to například v případě, když pinu nastavíme funkci USART (PB6, PB7), ale ještě nevybereme mód (asynchronní apod.).

Výstupem je hlavní soubor programu main.c, který kromě základní konfigurace obsahuje samotný program aplikace. Vnitřně je dělen na 2 části – systémovou a uživatelskou.

Systémová část je vytvořena automaticky a obsahuje nastavení pinů, inicializace, systémové funkce a podobné. Do této části kódu by programátor neměl zasahovat. To je z důvodu funkčnosti wizardu pro konfiguraci mikrokontroléru, protože se kdykoli můžeme vrátit zpět a piny přenastavit. V tuto chvíli program snaže vše mimo uživatelskou část a znovu nastaví. Tedy pokud by programátor zasáhl do systémové části kódu a například doplnil nějaké vlastní funkce, tímto přenastavením by o ně přišel.

Uživatelská část je již chráněna proti automatickému smazání při změně konfigurace. Zde se deklarují vlastní proměnné, funkce a píše se hlavní program projektu. V kódu je tato část jasně oddělena komentáři s hesly USER CODE BEGIN pro začátek uživatelské části a USER CODE END pro ukončení uživatelské části.

```

/* Private variables ----- */
TIM_HandleTypeDef htim3;
TIM_HandleTypeDef htim4;
TIM_HandleTypeDef htim5;

UART_HandleTypeDef huart1;
DMA_HandleTypeDef hdma_usart1_rx;
DMA_HandleTypeDef hdma_usart1_tx;

/* USER CODE BEGIN PV */
/* Private variables ----- */
uint8_t board_ADDRESS;
uint8_t board_SN_ID[6]="L00999";
int16_t board_Currents[8];
uint8_t board_DigitalInputs[4];
uint8_t board_COMMAND;

/* USER CODE END PV */

```

} Systémová část

} Uživatelská část

Obrázek 22: Ukázka deklarace proměnných - systémová a uživatelská část

## 7.2. Komunikace po sběrnici MODBUS

Pro komunikaci byla vybrána sběrnice MODBUS v režimu ASCII. Z toho důvodu je potřeba vytvořit pomocnou funkci pro převod 1 bytu na 2 oddělené byty, aby bylo možné

sestavit rámec pro komunikaci. Adresa, funkce a tak dále jsou stále tvořeny jedním bytem, ale v komunikaci je tento byte reprezentován dvěma znaky. Dále je potřeba vytvořit funkci pro výpočet kontrolního součtu LRC. Dle domluvy s vedoucím projektu nejsou kódy v této práci uvedeny.

### 7.2.1. Funkce rozdělení bytu na 2 ASCII znaky

Základním datovým typem, se kterým se v programu pracuje, je byte. Výhodou je, že s ním lze pracovat jako s datovým typem char. Abychom jej převedli na 2 ASCII znaky pro MODBUS komunikaci, je potřeba získat hexadecimální hodnotu bytu jako dva samostatné znaky. Toho se docílilo vytvořením 2 proměnných typu int (iH a iL) pro vyšší a nižší hodnotu bytu. Na vstupní znak se pomocí funkce bitového AND aplikuje maska 0000 1111 a uloží se do proměnné iL pomocí přetypování. Pro vyšší hodnotu bytu se aplikuje maska 1111 0000, na výsledek se aplikuje rotace vpravo o 4 místa a uloží se do proměnné iH.

Tyto získané hodnoty zatím nerepresentují skutečné znaky pro přenos, ale pouze desítkovou hodnotu hexadecimálního znaku. Založíme si tedy konstantu typu char\* pro řetězec znaků „0123456789ABCDEF“. Znaky na pozici iL a iH jsou hexadecimální reprezentace požadovaného znaku. Do nových proměnných typu char cL a cH tedy uložíme tyto znaky a dosáhli jsme rozdělení bytu na 2 ASCII znaky

Rámec se skládá v pořadí vyšší | nižší byte, tedy vytvoříme pole znaků o velikosti N jako výstup funkce, do kterého se uloží na pozici 0 hodnota cH a na pozici 1 hodnota cL. Jelikož je zbytečné volat tuto funkci pro každý byte zvlášť, je vstupním argumentem řetězec bytů, které se postupně ve smyčce rozdělí a uloží do výstupního pole znaků. Velikost tohoto pole se určí na základě velikosti vstupního řetězce.

### 7.2.2. Funkce výpočtu LRC

LRC součet je nezbytné vypočítat na původních datech před rozdělením. Sestavíme si tedy zprávu skládající se z adresy, kódu funkce a dat, které odesíláme. Vytvoříme si dvě pomocné proměnné b1 a b2. Do proměnné b1 uložíme vždy výsledek sčítání (v prvním kroku je rovna prvnímu bytu) a do proměnné b2 uložíme následující byte pro operaci sčítání (v prvním kroku jako druhý byte). Tyto dvě proměnné sečteme a výsledek uložíme opět do proměnné b1. V rámci jednoduchosti přetypujeme byty na datový typ uint8\_t a operaci sčítání provedeme nad celými čísly. Na výsledný součet provedeme negaci dvojkovým doplňkem, tedy provedeme bitovou negaci a přičteme hodnotu 1. Výslednou hodnotu tedy



vrátíme jako výsledek funkce. Můžeme provést ověření správnosti výpočtu sečtením výsledku LRC a sumy původní zprávy. Pokud je součet těchto dvou hodnot 0, pak je výpočet v pořádku.

### 7.2.3. Sestavení rámce a odeslání odpovědi

Před odesláním odpovědi je nezbytné sestavit rámec. Vytvoříme buffer, kam uložíme znak dvojtečky jako počátek zprávy. Poté pomocí funkce pro rozdělení zprávy na 2 ASCII znaky přidáme do bufferu adresu, kód funkce, případná data a nakonec LRC součet. Na závěr přidáme znaky 0x0D a 0x0A (CR a LF) a pomocí systémové funkce pro odeslání dat po sériové lince rámec odešleme.

### 7.2.4. Funkce pro převod dvou ASCII znaků na byte

Tento převod podobný jako v opačném případě. Naším vstupem jsou 2 ASCII znaky typu char, pojmenované jako c1 a c2. Stejně jako u funkce pro rozdělení na 2 ASCII znaky si tyto proměnné přetypujeme na datový typ integer a odečteme od nich hodnotu 48. Pokud bude hodnota vyšší než 15, odečteme hodnotu 7. Těmito hodnotami odstraníme všechny znaky kromě námi požadovaných znaků používaných v hexadecimální soustavě. Proměnnou c1 posuneme rotací vlevo o 4 místa do druhé vyšší poloviny bytu a uložíme zpět do c1. Poté provedeme logickou operaci OR nad proměnnými c1 a c2, kterou uložíme do proměnné c3 typu char. Tato proměnná je požadovaný znak.

Abychom opět nemuseli spouštět funkci nad každými dvěma ASCII znaky, je vstupem funkce řetězec znaků, kde se cyklicky bere vždy dvojice znaků pro převod na 1 byte, který se poté uloží do výstupního řetězce.

### 7.2.5. Funkce pro příjem požadavku

Zde je použita systémová funkce, která pomocí časovače stále přijímá data. Jakmile přijde zpráva, provede se kontrola počátečního znaku. Pokud je přijatý znak dvojtečka (začátek vysílání), začne se zpráva zpracovávat. V opačném případě se data ignorují a čeká se na další příjem znaku dvojtečky. Jakmile přijde znak dvojtečky, vezmou se první dva následující znaky (adresa) a pomocí funkce pro převod dvou ASCII znaků na 1 byte se zjistí adresa příjemce. Pokud se adresa liší od adresy příjemce, zpráva se zahodí a čeká se na další data začínající znakem dvojtečky. Pokud se adresa shoduje, dekoduje se zbytek zprávy, provede se kontrolní LRC součet a předají se data hlavnímu programu.

### 7.3. Modularizace

Program vznikl za spolupráce s vedoucím projektu, a tedy kód tvořilo více než jeden člověk. Každý programátor může mít jiný programovací styl a výsledný projekt se pak stává zamotaným klubkem. Proto se často na pracovištích vyžaduje jistá sémantika, nejčastěji podle „best practises“ neboli neoficiální, ale obecně zavedené praktiky.

Tento projekt se stal tím špatným případem. Nebyla předem vytvořena žádná struktura programu. Postupně se do programu přidávaly nové funkce bez ohledu na jejich logické zařazení. To nakonec vedlo k hlavnímu souboru o 1 500 řádcích. Cílem tohoto bodu tedy je provést revizi programu a rozdělit jej do různých logických bloků za účelem zvýšení přehlednosti programu a sjednocení stylu zápisu kódu. Program byl rozdělen do celkem sedmi samostatných bloků podle funkcionality kódu:

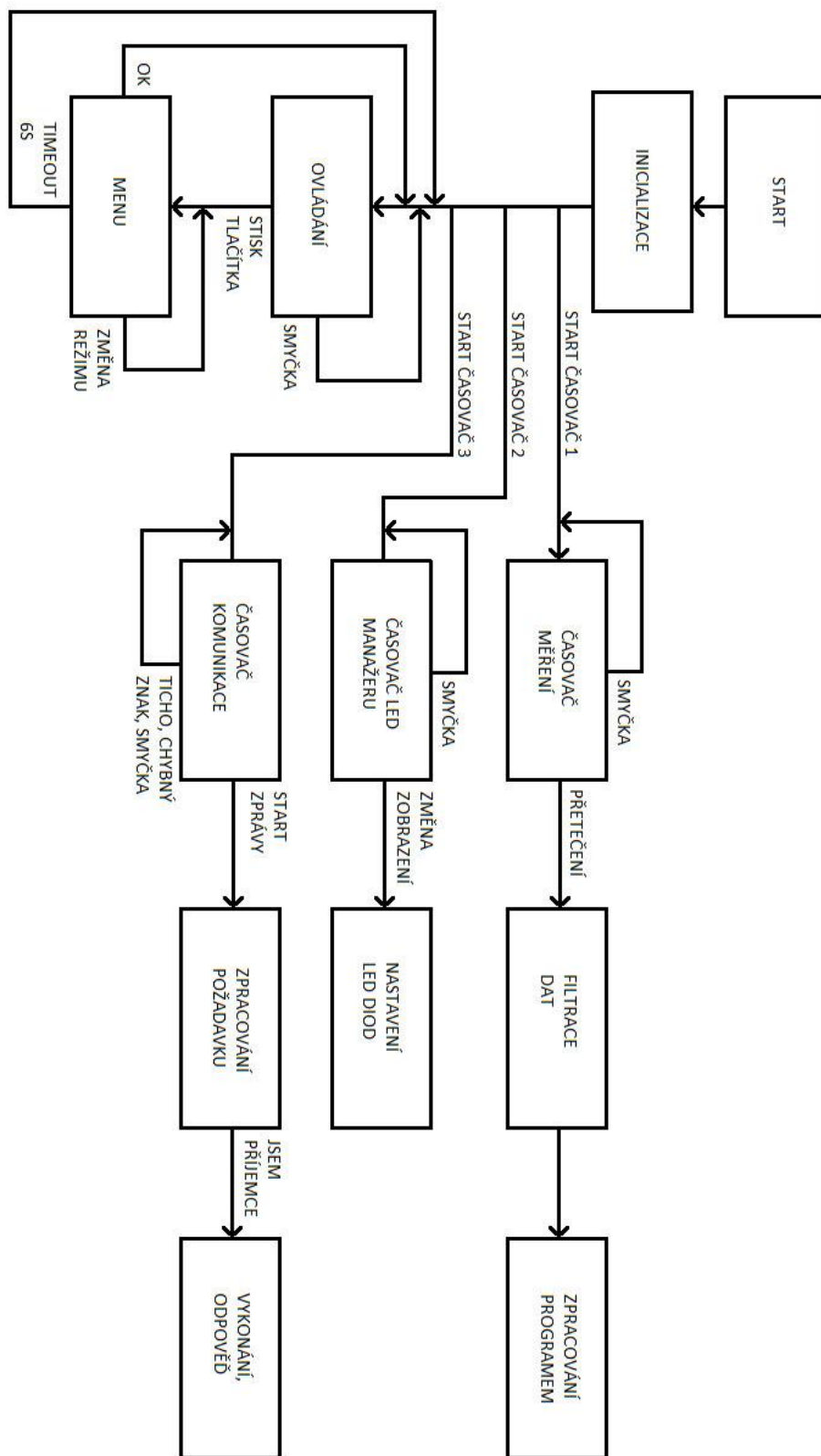
- DIN\_Manager – správa digitálních vstupů
- FIR\_functions – funkce a nastavení FIR filtru
- Flash\_functions – funkce pro práci s pamětí Flash
- Helper\_functions – pomocné funkce (časovače, převody apod.)
- LED\_Manager – správa výstupu LED diod
- main – hlavní program projektu
- Serial\_communication – funkce pro sériovou komunikaci (MODBUS)

V rámci revize kódu došlo nejen k úpravě stylu zápisu kódu, ale také k úpravám obsahu. Přebytné returny na konci funkcí a zbytečné proměnné byly odstraněny a některé podmínky (if) byly nahrazeny přepínači (switch), kde to princip kódu dovoľoval (například výběr v menu).

### 7.4. Shrnutí celého programu

Nekonečná smyčka hlavního programu obsahuje pouze funkce pro ovládání desky. Jde o nastavení režimu měřicí jednotky. Na stisk tlačítka (vyjma tlačítka pro reset) se spustí menu. Jedním tlačítkem se vybírá režim a druhým se výběr potvrzuje. Pokud 6 s po spuštění menu není stisknuto žádné tlačítko, menu se ukončí a program se vrátí k předchozí akci.

Všechny ostatní funkce jako zajištění komunikace (příchozí i odchozí), výpočet FIR filtru u naměřených dat po A/D převodu a ovládání led diod při jednotlivých operacích je řízeno pomocí časovačů.



Obrázek 23: Vývojový diagram programu

## 8. Závěr

Jelikož zadání nebylo zcela určeno pro obor IT, lehce se po domluvě s vedoucím práce v průběhu projektu upravilo. Cílem již nebylo vytvořit novou desku od základů, ale navrhnout pouze moduly, které by nahradily stávající řešení na deskách.

První navrženou deskou je modul pro napájení měřicí desky přímo ze solárních panelů. String solárních panelů produkuje napětí až 1 000 V, a tak je nezbytné navrhnout systém, který toto napětí zredukuje na pracovních 5 V s minimálními ztrátami a zahříváním. Jak se v průběhu návrhu ukázalo, lepší, levnější nebo dostupnější řešení se navrhnout nepodařilo.

Druhou navrženou deskou je náhrada řídicí (měřicí) jednotky pro desku s interním názvem PECONT. K této desce nebyly žádné dostupné materiály, a tedy se zvolilo celou jednotku nahradit novou. Realizace desky proběhla na základě návrhu v této práci, desku plošných spojů již ale navrhnul a realizoval vedoucí práce.

Program k této desce vznikl za spolupráce s vedoucím projektu. Mé zaměření bylo naprogramovat sběrnici MODBUS pro komunikaci v režimu ASCII s dalším přístrojem. Jsou zde popsány jednotlivé funkce pro tuto komunikaci, ale na žádost vedoucího nejsou zveřejněny žádné kódy. Další zaměření pro práci s tímto programem pro měřicí jednotku bylo rozdělit jej do vícero modulů na základě jejich funkcionality. Původně projekt vznikl jako jeden velký soubor, který obsahoval všechny funkce najednou. V případě programování mikrokontrolérů takový přístup ve většině případech nevádí, neboť se jedná po poměrně malé kódy. Ve chvíli, kdy se na projektu podílí více lidí, se ale kód stává nepřehledným. Rozdělení programu do vícero funkcí nejen zpřehlednilo celý projekt, ale také dalo jednodušší možnost provést revizi a optimalizaci kódu.

Samotný program s poklady k jednotlivým měřicím deskám, se kterými se v tomto projektu pracovalo, je u vedoucího projektu.

## 9. Reference a odborná literatura

- [1] SOLÁRNÍ ENERGIE. *Skupina ČEZ* [online]. Praha: ČEZ, ©2017 [cit. 2017-09-16]. Dostupné z: <https://www.cez.cz/edee/content/microsites/solarni/f8.htm>
- [2] Přeměna sluneční energie v energii elektrickou. *Fyzikální ústav Akademie věd ČR* [online]. Praha: Fyzikální ústav AV ČR, ©2017 [cit. 2017-09-16]. Dostupné z: <https://www.fzu.cz/popularizace/premena-slunecni-energie-v-energii-elektrickou>
- [3] Fotovoltaické elektrárny Ralsko a Mimoň. *Skupina ČEZ* [online]. Praha: ČEZ, ©2017 [cit. 2017-09-16]. Dostupné z: <https://www.cez.cz/cs/vyroba-elektriny/obnovitelne-zdroje/slunce/provozovane-fotovoltaicke-elektrarny/fotovoltaicka-elektrarna-ralsko.html>
- [4] GROSMAN, Josef. *ŘPS - MODBUS* [přednáška]. Liberec: Fakulta mechatroniky, informatiky a mezioborových studií, TUL, [cit. 2017-09-16].
- [5] BARRY, Richard. *Using the FreeRTOS Real Time Kernel*. 1. vyd. London: Real Time Engineers Ltd., 2010. ISBN 978-1-4461-7030-4
- [6] STM32F7x2 - STMicroelectronics. *STMicroelectronics* [online]. Ženeva: STMicroelectronics, ©2017 [cit. 2017-09-16]. Dostupné z: <https://www.st.com/en/microcontrollers/stm32f7x2.html?querycriteria=productId=LN1903>
- [7] Princip paměti FLASH. *České vysoké učení technické v Praze* [online]. Praha: České vysoké učení technické v Praze, ©2015 [cit. 2017-09-16]. Dostupné z: <http://noel.feld.cvut.cz/vyu/scs/prezentace2003/Flash-Intel/>
- [8] OLIVKA, Petr a Martin KRČMÁŘ. *Paměti počítačů: Studijní materiál pro předmět Architektury počítačů* [online]. Ostrava, 2010 [cit. 2017-09-16]. Dostupné z: <http://poli.cs.vsb.cz/edu/apps/down/pameti.pdf>. Technická univerzita Ostrava.
- [9] Mezinárodní vesmírná stanice. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2017-09-16]. Dostupné z: [https://cs.wikipedia.org/wiki/Mezin%C3%A1rodn%C3%AD\\_vesm%C3%ADrn%C3%A1\\_stanice](https://cs.wikipedia.org/wiki/Mezin%C3%A1rodn%C3%AD_vesm%C3%ADrn%C3%A1_stanice)
- [10] Modbus: Úvod do protokolu. In: *Lukáš Vlček: Osobní blog* [online]. Liberec: Vlček, ©2016-2017 [cit. 2017-09-16]. Dostupné z: <https://www.lukas-vlcek.cz/iot/modbus-uvod-do-protokolu/>
- [11] Harvardská architektura. In: *MediaWiki SPŠ a VOŠ Písek* [online]. Písek: Střední průmyslová škola a Vyšší odborná škola Písek, 2010 [cit. 2017-09-16]. Dostupné z: [http://wiki.sps-pi.cz/index.php/Harvardsk%C3%A1\\_architektura](http://wiki.sps-pi.cz/index.php/Harvardsk%C3%A1_architektura)