



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Zobrazovací jednotka stavových veličin řízená z PLC

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické, informační a řídicí systémy

Autor práce: **Jiří Kulich**

Vedoucí práce: Ing. Zbyněk Mader, Ph.D.





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

The Display Unit of State Variables Controlled by a PLC

Bachelor thesis

Study programme: B2612 – Electrotechnology and informatics

Study branch: 2612R011 – Electronic, Information and control systems

Author: **Jiří Kulich**

Supervisor: Ing. Zbyněk Mader, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří Kulich**
Osobní číslo: **M15000100**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Zobrazovací jednotka stavových veličin řízena z PLC**
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 PLC Mitsubishi FX5U a mikrořadičem řady PIC32, vyberte vhodnou sběrnici pro datové přenosy.
2. Navrhněte elektronické zařízení zobrazovací jednotky s mikrořadičem PIC32, které umožní zpracovat a zobrazit na LCD displeji data zasláná z PLC po sběrnici. Zařízení dále umožní přenos dat do PLC.
3. Návrh realizujte do podoby funkčního vzorku včetně SW vybavení mikrořadiče a PLC, funkčnost proveďte na demonstračním modelu.

Rozsah grafických prací: Dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30-40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] AXELSON, Jan. Serial Port Complete. Second Edition, Lakeview Research LLC Madison, WI 53704, ISBN 978-1931448-07-9
- [2] IBRAHIM, Dogan. PIC32 microcontrollers and the digilent chipKIT: introductory to advanced projects. First edition. Kidlington, Oxford: Newnes is an imprint of Elsevier, 2015. ISBN 9780080999340
- [3] DI JASIO, Lucio. Programming 32-bit microcontrollers in C: exploring the PIC32. Burlington, MA: Newnes, 2008. ISBN 9780750687096

Vedoucí bakalářské práce:

Ing. Zbyněk Mader, Ph.D.

Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: 19. října 2017

Termín odevzdání bakalářské práce: 14. května 2018

prof. Ing. Zdeněk Plíva, Ph.D.
děkan



prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou 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 bakalářské práce pro vnitřní potřebu TUL.

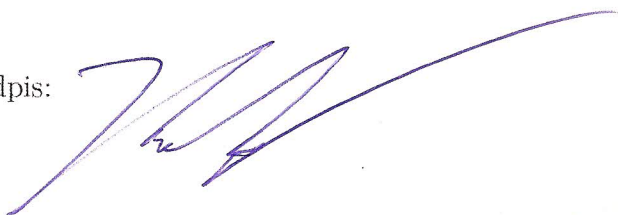
Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

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

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

Datum: 9.5.2018

Podpis:



Abstrakt

Cílem této práce je návrh a realizace zobrazovací jednotky komunikující přes sběrnici s nadřazeným PLC. Jednotka je řízena mikrokontrolérem PIC od společnosti Microchip. Hlavní funkcí zařízení je komunikovat po sběrnici s PLC a zobrazovat požadované údaje na displeji. Součástí práce je též komunikační protokol a program mikrokontroleru, který umožňuje obousměrnou komunikaci. Dosažené výsledky jsou demonstrovány na jednoduchém modelu.

Klíčová slova:

PIC32MX, mikrokontroler, znakový LCD displej, RS485, komunikační protokol, PLC, zobrazovací jednotka

Abstract

The aim of this work is to design and implement a display unit communicating over a bus with a PLC. The unit is controlled by a Microchip PIC microcontroller. The main features of the device are to communicate with the PLC and display the required data on the display. Part of the thesis is also a communication protocol and a program of microcontroller, which allows bidirectional communication. Achieved results are demonstrated on a simple model.

Key words: PIC32MX, microcontroller, character LCD display, RS485, communication protocol, PLC, display device

Poděkování

Za odborné rady, otevřený přístup a volné pole působnosti patří můj dík panu Ing. Zbyňku Maderovi, Ph.D. Děkuji mu také za poskytnuté téma BP, které přesně odpovídalo mým představám.

Obsah

Seznam zkratek	11
1 Úvod	12
2 Řídící systém	14
2.1 Sestava PLC	15
3 Mikrokontroler zobrazovací jednotky	16
3.1 Programování mikrokontroleru	17
4 RS-485	21
4.1 Komunikační protokol	23
4.1.1 Paket	23
4.1.2 Obsluha protokolu v PLC	24
5 Zapojení zobrazovací jednotky	26
5.1 Napájení	26
5.2 Mikrokontroler	27
5.3 Displej	28
5.3.1 Testování grafického displeje	29
5.4 Adresa	29
5.5 RS-485	31
5.6 Teploměr, EEPROM	31
5.7 DPS	33

6	Program mikrokontroleru	34
6.1	Inicializace	34
6.2	Smyčka programu	37
6.3	Zpracování	38
6.4	Funkce	39
6.4.1	Zobrazování na displej	39
6.4.2	Měření teploty	40
6.4.3	Zapisování do paměti EEPROM	40
6.4.4	Čtení z paměti EEPROM	40
7	Testování	41
7.1	Výsledek	43
8	Závěr	45
	Seznam použité literatury	46
	Přílohy	47
A	Seznam příloh na CD	47
B	Seznam použitých součástek	48
C	Schéma - mikrořadič	49
D	Schéma - zobrazování	50
E	Schéma - adresace	51
F	Schéma - teploměr, EEPROM, RS485	52
G	Schéma - napájení	53

Seznam obrázků

1.1	Blokové schéma	12
2.1	PLC Mitsubishi FX5U [7]	14
2.2	Sestava PLC	15
3.1	Rozložení pinů mikrokontroléru PIC32MX250F128B [6]	17
3.2	Základní schéma	18
3.3	Diagram časovacích obvodů [6]	19
4.1	Sběrnice RS-485	21
4.2	Graf závislosti délky vedení na přenosové rychlosti [1]	22
4.3	Datová struktura paketu	23
4.4	Datová struktura ACK paketu	24
5.1	Schéma napájení [Příloha G]	26
5.2	Schéma zapojení mikrokontroleru [Příloha C]	27
5.3	Schéma zapojení displeje [Příloha D]	28
5.4	Schéma obvodu nastavení adresy [Příloha E]	30
5.5	Svorkovnice pro připojení sběrnice RS485 [8]	31
5.6	Schéma řevodníku RS485 [Příloha F]	31
5.7	Teplotní senzor TCN75 [9]	32
5.8	Schéma zapojení teplotního čidla a paměti EEPROM [Příloha F]	33
5.9	Návrh DPS	33
7.1	Testovací vzorek zobrazovací jednotky	41

7.2	První obrazovka - měření teploty	42
7.3	Druhá obrazovka - graf teploty	42
7.4	Třetí obrazovka - práce s pamětí EEPROM	43

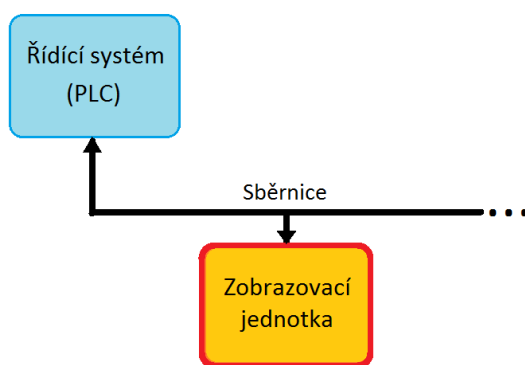
Seznam zkratek

TUL	Technická univerzita v Liberci
BP	Bakalářská práce
PLC	Programmable Logic Controller, programovatelný logický automat
RISC	Reduced Instruction Set Computing, architektura procesoru s omezenou instrukční sadou
MIPS	Microprocessor without Interlocked Pipeline Stages, architektura procesoru s možností zpracování více instrukcí najednou
RAM	Random-Access Memory, paměť s náhodným přístupem
DMA	Direct Memory Access, přímý přístup do paměti
UART	Universal Asynchronous Receiver and Transmitter, asynchronní sériové rozhraní
SPI	Serial Peripheral Interface, sériové periferní rozhraní
I2C	Inter-Integrated Circuit, seriová sběrnice
PPS	Peripheral Pin Select, výběr periférních pinů
CTMU	Charge Time Measurement Unit, jednotka pro měření kapacit a generování signálů
USB	Universal Serial Bus, univerzální sériová sběrnice
PMP	Parallel Master Port, hlavní paralelní port
THT	Through-hole technology, technologie montáže s drátovými vývody
SMD	Surface Mount Device, součástka pro povrchovou montáž

1 Úvod

Tuto bakalářskou práci jsem si zvolil proto, že svým tématem poměrně dobře uceluje dosavadní získané znalosti. Obsaženou problematikou spojuje elektroniku a programování, což jsou obory, které mě provází celým mým studijním životem (pominu-li základní školu) a jelikož mě osobně baví, chtěl bych v nich pokračovat i ve svém budoucím profesním životě. Jsem rád, že se obsahem práce stalo i programování PLC. Rozšíření dosavadních znalostí a zkušeností i z tohoto portfolia je pro mě velmi vítané a jsem za tuto možnost vděčný.

Samotný námět této práce vyšel přímo z požadavků praxe. Ve složitějších a komplexnějších aplikacích je potřeba informovat obsluhu o určitých stavech či událostech řízeného procesu. To je ovšem u prostorově rozsáhlých struktur problematické, jelikož rozvaděč řídicího PLC není vždy tam, kde je patřičná informace potřebná. Protože by byl dotykový operační panel pro stručnou informaci zbytečně veliký a drahý, je nahrazen jednodušší a vhodnější zobrazovací jednotkou se znakovým displejem.



Obrázek 1.1: Blokové schéma

Při sběrnicové komunikaci je možné takovýchto zobrazovacích jednotek osadit

vícero. Lze tedy informovat o stavech daného procesu přímo v místě potřeby. Displej samotný je obsluhován mikrokontrolerem, který naslouchá na sběrnici a dle komunikačního protokolu plní příslušnou funkci. Tou ovšem nemusí být pouze zobrazování textových řetězců. Jeho součástí může být i senzor či akční prvek. To zařízení umožňuje univerzálnější použití. Na trhu je k dostání mnoho variant takovýchto jednotek. Jejich cena začíná někde na tisícikoruně. Tato práce popisuje návrh a sestavení takové zobrazovací jednotky dle specifických požadavků včetně komunikačního protokolu.

2 Řídící systém

Jako řídicí systém je použito PLC Mitsubishi FX5U. Jedná se o kompaktní PLC ze série MELSEC iQ-F od společnosti Mitsubishi. Jelikož je kompaktní, obsahuje integrovaný síťový zdroj, sběrnici RS-485 a ethernet, analogové stupy a výstupy, polohovací výstupy, slot pro SD kartu a možnost připojení rozšiřujících modulů, což mu propůjčuje jistý potenciál z modulárních PLC. Zde jsou shrnuty základní informace.

- Integrované vstupy/výstupy: 16/16
- Napájení: 100 – 240 V AC
- Typ výstupů: Transistorový
- Rozhraní: RS-485, Ethernet
- Analogový vstup: 2x (0 – 10 V), vstupní odpor 115,7 k Ω
- Analogový výstup: 1x (0 – 10 V)



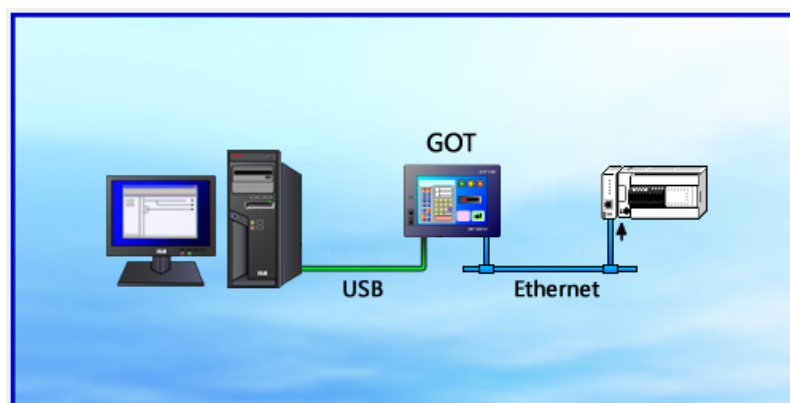
Obrázek 2.1: PLC Mitsubishi FX5U [7]

K tomuto PLC byl na univerzitě k dispozici dotykový grafický displej GOT GS21, jakožto tzv. rozhraní člověk-stroj (HMI). Tento TFT displej, celým názvem GOT

GS2107 – WTBD, má úhlopříčku 7” a rozlišení 800x480 obrazových bodů. Jako komunikační rozhraní s PLC byl zvolen Ethernet.

2.1 Sestava PLC

Před samotným programováním a uvedením PLC do provozu je nutné definovat sestavu, ve které je PLC zapojeno. Možností je mnoho. V tomto případě, kdy je k PLC připojen grafický dotykový panel skrze ethernetové rozhraní, je tento panel definovaný jako master. Komunikace počítače s PLC je tedy zprostředkována přes sběrnici USB grafického panelu.



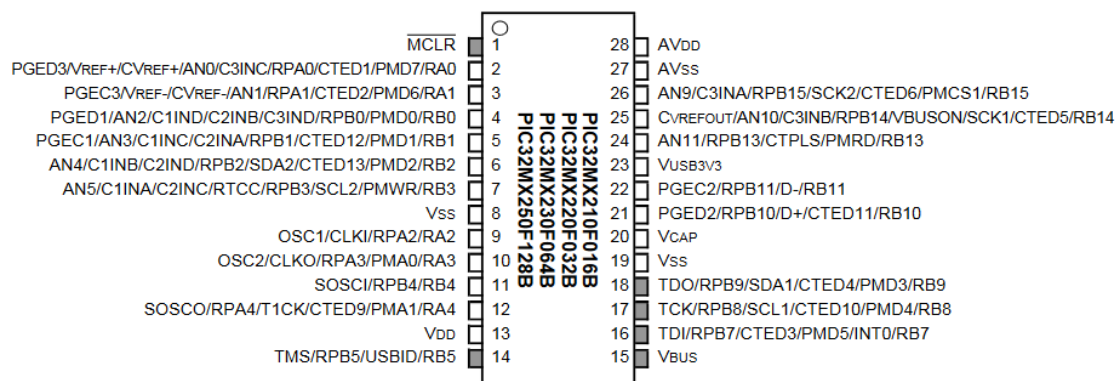
Obrázek 2.2: Sestava PLC

Při tomto sestavení je nejprve pro displej definována jeho IP adresa a adresa PLC jako station 2. Ve chvíli, kdy by měl displej stejné číslo stanice jako PLC, komunikace a tedy i celá sestava bude nefunkční. Poté je i na straně PLC zadána příslušná IP adresa a vybrána správná CPU jednotka, popřípadě i rozšiřující moduly (v tomto případě je PLC bez nich). Po správném definování sestavy je PLC i GOT připraven na nahrávání uživatelského programu.

3 Mikrokontroler zobrazovací jednotky

Celá jednotka je řízena mikrokontrolerem PIC32MX250F128B. Jedná se o 32-bitový mikrokontroler od společnosti Microchip postavený na jádře MIPS32 z rodiny RISC (Reduced Instruction Set Computer). Jádra tohoto typu mají menší počet jednodušších instrukcí, což značně zrychluje jejich zpracování. Jádra MIPS navíc umožňují zpracovávat více instrukcí najednou. Řada PIC32MX je svými vlastnostmi předurčena pro náročné aplikace. Zde je výčet jeho parametrů.

- Maximální taktovací frekvence: 50 MHz
- Velikost programové paměti: 128 KB
- RAM: 32 KB
- Pomocná Flash: 3 KB
- Provozní napětí: 2,3 V až 3,6 V
- 4x DMA kanály
- 2x UART
- 2x SPI
- 2x I2C
- CODEC interface (I2S, AC97)
- Obvod reálného času
- Možnost remapování (PPS)
- Jednotka CTMU
- USB interface
- 9kanálový A/D převodník s rozlišením 10 bitů
- 5 PWM
- 5x 16-bitový Timer
- Paralelní port PMP
- 3 komparátory
- Vnitřní oscilátory: 8 MHz, 32 KHz
- 21 I/O pinů
- 28 pinů celkem

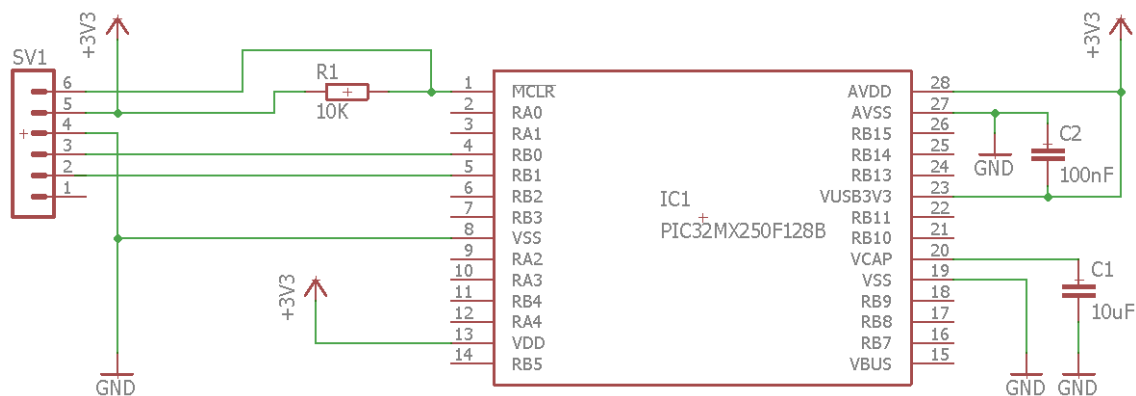


Obrázek 3.1: Rozložení pinů mikrokontroleru PIC32MX250F128B [6]

Společnost Microchip poskytuje volně stažitelné vývojové prostředí MPLAB pro programování mikrokontrolerů PIC v jazyku C/C++. Toto vývojové prostředí umožňuje nejen editaci, kompilaci a nahrávání zdrojového kódu, ale také pokročilé funkce ladění, jako je krokování programu, sledování hodnot proměnných a funkčních registrů i jejich editace během vykonávání programu. Neobejde se ovšem bez obvodového debugger/programátoru. Při práci bylo použito prostředí MPLAB X IDE ve verzi 3.40 s kompilátorem XC32-gcc (v1.42).

3.1 Programování mikrokontroleru

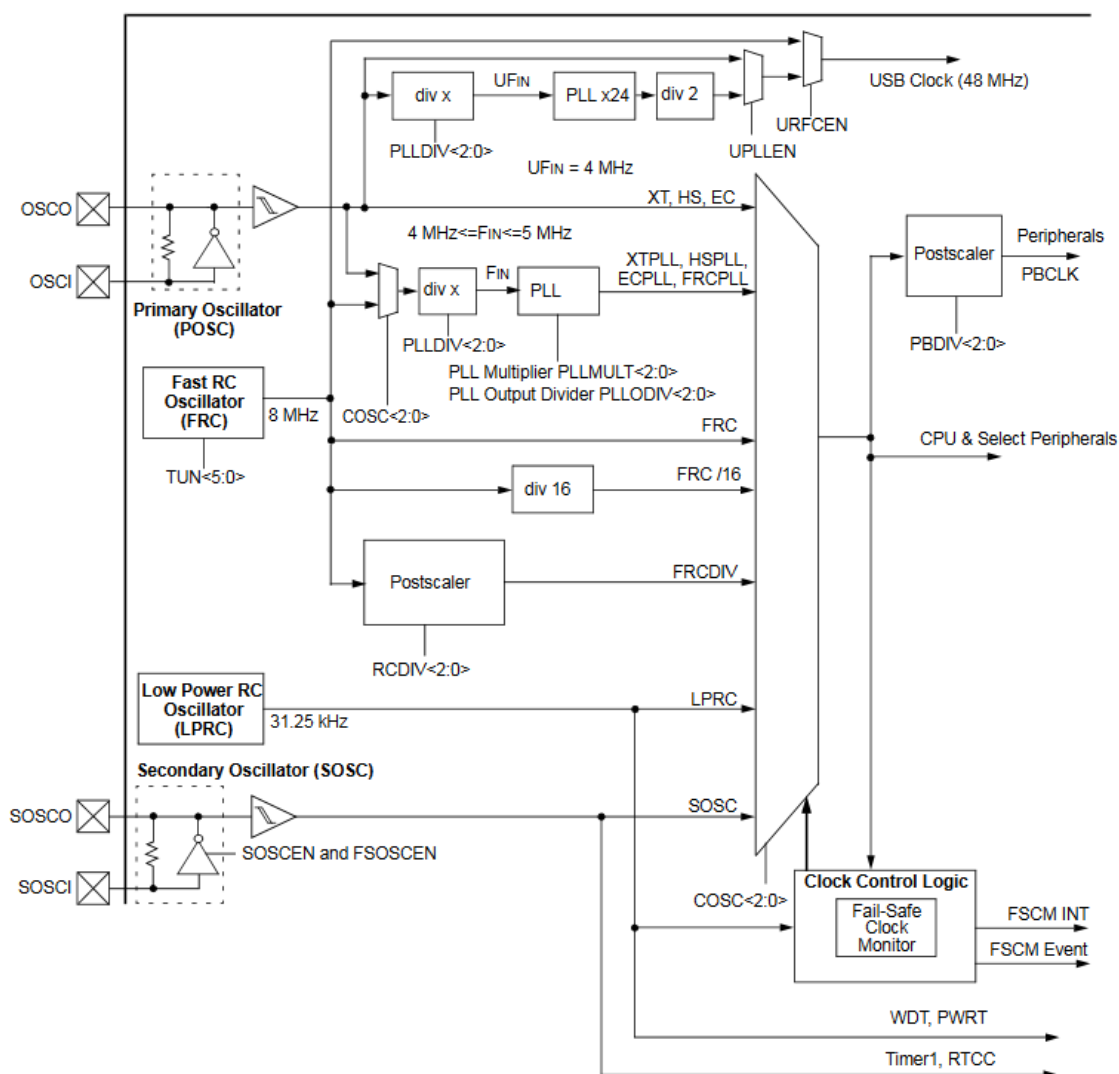
Aby bylo možné mikrokontroler programovat je nutné, aby byl obvod programátoru (v tomto případě chipKIT Programmer) zapojen na pinech k tomu určeným. Tím je dvojice pinů PGED/PGEC, která se vyskytuje na mikrokontroleru celkem třikrát. Je tedy pouze na nás, kterou dvojici si vybereme. Při programování byla využita dvojice PGED1/PGEC1 na pinech 4 a 5. Základní schéma umožňující programování vypadá tedy následovně.



Obrázek 3.2: Základní schéma

Pull-up rezistor R1 ošetřuje resetovací pin MCLR. Kondenzátor C1 má mít dle katalogového listu hodnotu 10uF (tantalový nebo keramický). Kondenzátor C2 slouží pro pokrytí proudových špiček a jeho hodnota je volitelná. Piny AVss a AVdd slouží jako reference pro analogový převodník.

Jelikož má mikrokontroler vnitřní 8MHz oscilátor označovaný jako FRC (Fast RC Oscillator), nemusí se připojovat externí (na piny OSC1 a OSC2). Mikrokontroler obsahuje také obvod PLL (Phase-Locked Loop), který umožňuje měnit výslednou frekvenci pomocí děliček a násobiček. Jejich hodnoty jsou určeny uživatelem a nastaveny při nahrávání programu.



Obrázek 3.3: Diagram časovacích obvodů [6]

Chceme-li, aby byla výsledná frekvence rovna 20MHz je potřeba definovat několik parametrů. Nejprve je nutné zvolit zdroj hodinového signálu. To udává parametr FNOSC, kterému přiřadíme hodnotu FRCPLL. Použije se tedy vnitřní oscilátor FRC spolu s obvodem PLL, který umožní frekvenci měnit. PLL se skládá ze vstupní děličky, násobičky a výstupní děličky. Násobička pracuje s frekvencí v rozsahu 4MHz až 5MHz. Vstupní signál 8MHz tedy vydělíme 2, tím získáme frekvenci 4MHz pro násobičku. Násobičku nastavíme na 20, čímž signál vynásobíme a získáme frekvenci 80MHz. Výstupní dělička nastavená na hodnotu 4 následně signál vydělí

a získáme frekvenci 20MHz. To vše sdělíme překladači následujícími příkazy.

```
#pragma config FNOSC    = FRCPLL    // zdroj pro SYSCLK
#pragma config FPLLIDIV = DIV_2     // DIV IN
#pragma config FPLLMUL  = MUL_20    /* 4MHz <= Freq <= 5MHz
MAX 40MHz */
#pragma config FPLLODIV = DIV_4     // DIV OUT
```

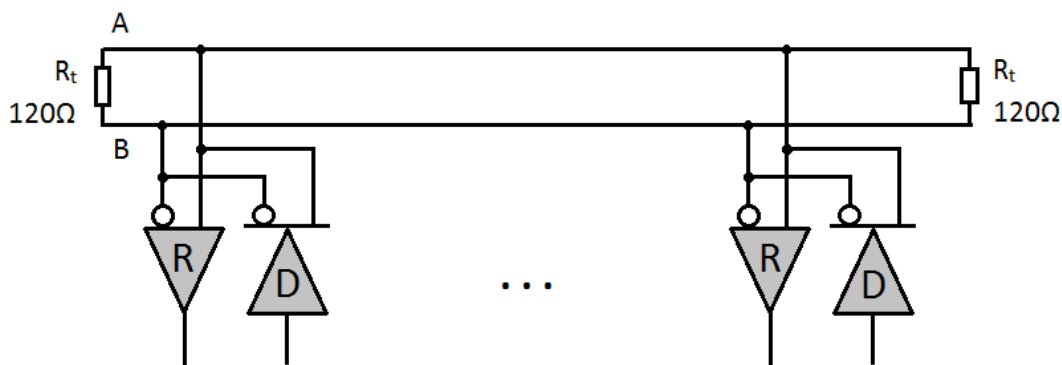
Direktiv **#pragma** je více. Lze nastavit ochranu kódu, piny pro ladění (použito ICS_PGx1, tedy piny PGED1/PGEC1, viz výše), diagnostika JTAG, watchdog atd. To ovšem nebudu blíže rozebírat.

Toto je tedy nutný základ, který se musí před samotným programováním splnit. Nyní je vše připravené a je možné začít psát zdrojový kód.

4 RS-485

Jako komunikační sběrnice byla vybrána sběrnice RS-485. Je pro jednoduchou výměnu informací mezi zařízeními master a slave nejvhodnější.

Princip fungování sběrnice RS-485 je založen na vyhodnocování napěťového rozdílu (a polarity) mezi dvěma vodiči (diferenciální přenos). Logický stav tedy není určován vůči nulovému vodiči (GND). To umožňuje větší odolnost k rušení a tedy i možnost vyšších přenosových rychlostí na větší vzdálenost. V nejjednodušším případě postačí jeden kroucený pár. Pro správnou funkčnost vysílače a přijímače je však potřeba společný potenciál (zemní rozdíl nesmí být vyšší než 7V). Stačí každé zařízení připojit na společnou zem. V případě použití třetího zemního vodiče, je třeba při větších zemních potenciálech použít sériový odpor (cca 100Ω) pro eliminaci nežádoucích proudů.



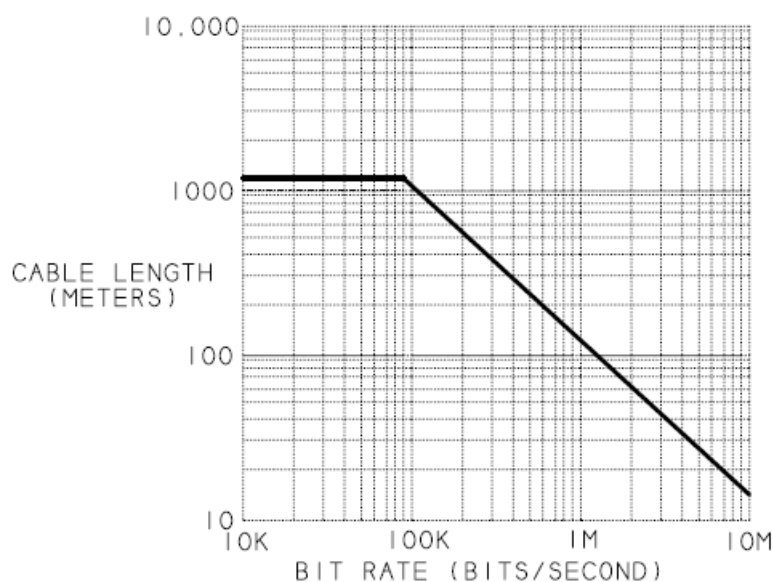
Obrázek 4.1: Sběrnice RS-485

Komunikační vodiče jsou označeny jako „A“ a „B“ (někdy též „-“ a „+“). Tyto vo-

diče jsou vysílačem buzeny v protifázi. V logickém stavu 0 je napětí na vodiči „A“ kladnější než na „B“. Vyhodnocovací úroveň je $\pm 200\text{mV}$. Pokud ovšem žádné zařízení nevysílá, jsou všechna ve stavu vysoké impedance, a tudíž není úroveň na sběrnici definována. V tomto stavu je náchylná na indukované napětí. Z tohoto důvodu je vhodné sběrnici opatřit napěťovým děličem, který stanoví úroveň v klidovém stavu.

Z důvodu vysoké vstupní impedance komunikačních obvodů se sběrnice chová jako nezakončené vedení. To vede k nežádoucím odrazům signálu, které mohou v nejhorším případě úplně znemožnit komunikaci. Aby se tomu předešlo, připojují se na konce vedení tzv. terminátory. Jedná se o obyčejný odpor, který vedení mírně zatíží, čímž dojde k maření případných odrazů. Nejčastěji má hodnotu 110Ω nebo 120Ω (někdy uváděno 150Ω [5]). Je-li vedení kratší než 50m, terminátory se nemusí používat, jelikož by docházelo k zbytečnému zatěžování sběrnice.

Maximální délka sběrnice je 1200m (někdy uváděno i 1600m při 65pF/m [4]). Záleží na kapacitě vedení a na přenosové rychlosti. Čím je přenosová rychlost vyšší, tím je maximální možná délka nižší ($9600\text{Bd} - 1000\text{m}$ [5]). Maximální počet připojených zařízení je 32.



Obrázek 4.2: Graf závislosti délky vedení na přenosové rychlosti [1]

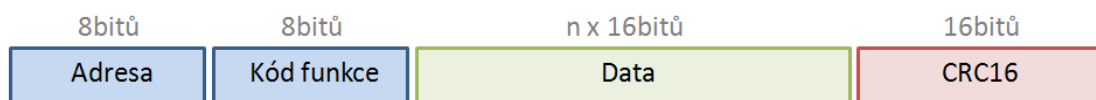
4.1 Komunikační protokol

Aby bylo možné komunikovat po sběrnici, je nutné určit způsob, jakým tato komunikace bude probíhat. To definuje komunikační protokol. Jelikož sběrnice RS-485 nemá vyvedené řídicí signály, musí se o řízení komunikace starat samotný protokol, aby nedocházelo ke kolizím. Musí také zabezpečit adresaci paketu, aby se zpráva dostala k určenému zařízení. V neposlední řadě také musí obsahovat kontrolní mechanismus. Poškozený paket (např. vlivem rušení) musí být nalezen a vyřazen.

Pro sběrnici RS-485 je nejpoužívanějším průmyslovým protokolem protokol MODBUS. Jde o otevřený komunikační protokol typu Master - Slave. Z něj se také při tvorbě protokolu vycházelo. Ovšem komunikačních protokolů na sběrnici RS-485 je více. Za zmínku stojí protokol SPINEL české společnosti Papouch, která vyrábí průmyslovou elektroniku. Mezi jejich výrobky patří i podobné komunikační jednotky, jako je navržena v této bakalářské práci.

4.1.1 Paket

Mezi zařízením master a slave se komunikuje pomocí paketů, tedy datových bloků (balíčků). Tento datový blok musí mít předepsanou datovou strukturu.



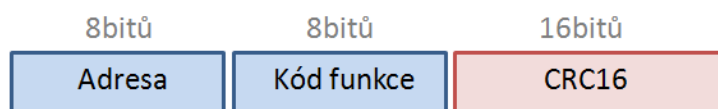
Obrázek 4.3: Datová struktura paketu

Hlavička, která musí být v každém paketu, obsahuje 8bitovou adresu slave zařízení a kód funkce. Adresa je v rozsahu 0 až 255, kde 0 reprezentuje broadcast paket, čili paket který přijde všem stanicím. Kód funkce je číslo, které reprezentuje o jaká data či požadavek se jedná. Chceme-li například zobrazit text na displej, jako kód funkce přiřadíme konstantu pro zobrazování textu (tedy 0xAA) a připojíme textový řetězec jako data.

Oblast dat je omezena 32byty (16 x word). Oblast je vyjádřena 16bitovou strukturou, jelikož PLC operuje právě s datovým typem word. To výrazně zjednodušuje práci s daty.

Na konci každého paketu je kontrolní 16bitová hodnota CRC16. CRC (cyklický redundantní součet) má řídicí polynom 0x8005, který odpovídá polynomu použitému v protokolu MODBUS.

Tímto způsobem komunikuje nadřazená stanice s podřízenou a naopak. Pokud paket neobsahuje žádná data, jako například ACK (Acknowledgement) paket, tedy potvrzovací paket informující o vykonání daného příkazu, vypadá následovně.



Obrázek 4.4: Datová struktura ACK paketu

4.1.2 Obsluha protokolu v PLC

Pro zjednodušení komunikace se zobrazovacími jednotkami byly v PLC vytvořeny dvě funkce, které obsluhují navržený protokol. Obě jsou psány v jazyce ST (Structured text).

Funkční blok Packet_generate generuje paket dle vstupních parametrů do zvolené paměťové oblasti PLC. Vstupem je adresa požadované jednotky (Adr), kód funkce (F_code), data (Data_h), délka dat v bytech (Data_len) a obslužné signály funkce (Enable, Process). Výstupem je vytvořený paket (Packet) a jeho délka v bytech (Packet_len). Hodnota CRC16 je generována a umístěna v paketu automaticky. Výstup ACK indikuje, že je paket hotový a je možné ho odeslat. Tento výstup lze použít pro následnou obsluhu odeslání paketu nebo obnovení vstupních parametrů funkce.

Program funkčního bloku Packet_generate:

```
IF (Enable) THEN
  IF (Process) THEN
    F_code_ := F_code;
    RCL(TRUE, 8, F_code_);
    Packet[0] := (F_code_ OR Adr);
    DIV_U(TRUE, Data_len, 2, Data_len_div);
    BMOV(TRUE, Data_h[0], (Data_len_div[0] + Data_len_div[1]),
      Packet[1]);
    CRC(TRUE, Packet[0], (((Data_len_div[0] + Data_len_div[1]) * 2)
      + 2), Packet[(Data_len_div[0] + Data_len_div[1] + 1)]);
    Packet_len := (Data_len_div[0] + Data_len_div[1] + 2) * 2;
    ACK := TRUE;
  ELSE
    ACK := FALSE;
  END_IF;
END_IF;
```

Funkce Packet_check slouží pro kontrolu přijatého paketu. Jako vstupní hodnota je datové pole s přijatou zprávou (Data_head), jeho délka (Data_n) a obslužné vstupy (Enable, Process). Funkce spočítá CRC16 z celého paketu a pokud je korektní (tedy výsledek je 0), je aktivován výstup ACK. Spolu s kontrolou je navíc separována adresa (Adr) a kód funkce (F_C) z hlavičky paketu pro zjednodušení následné práce s přijatými daty.

Program funkce Packet_check:

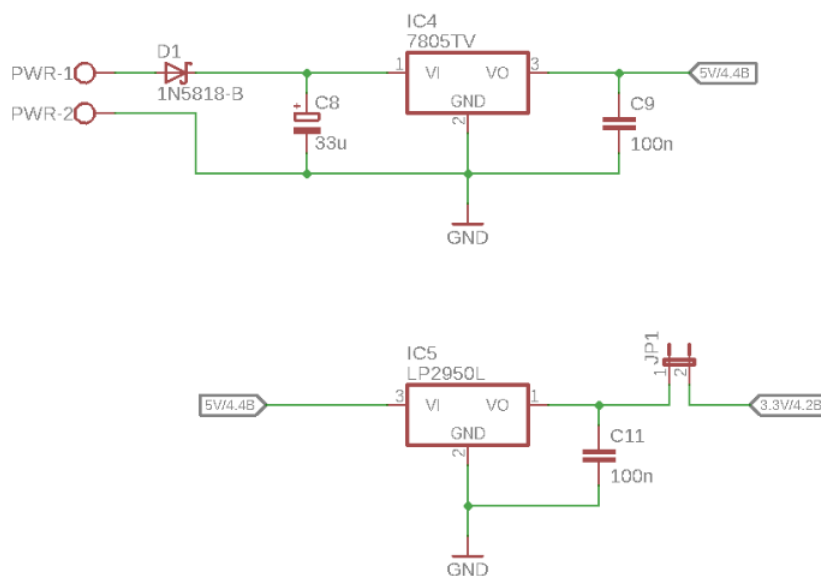
```
IF (Enable) THEN
  IF (Process AND (Data_n > 1)) THEN
    CRC(TRUE, Data_head[0], Data_n, CRC_);
    IF (CRC_ = 0) THEN
      Adr := (Data_head[0] AND 16#00FF);
      F_C := (Data_head[0] AND 16#FF00);
      SWAP(true, F_C);
      ACK := TRUE;
    ELSE
      ACK := FALSE;
    END_IF;
  ELSE
    ACK := FALSE;
  END_IF;
ELSE
  ACK := FALSE;
END_IF;
```

5 Zapojení zobrazovací jednotky

Návrh celého zařízení je rozdělen do jednotlivých dílčích celků, které jsou níže popsány a vysvětleny.

5.1 Napájení

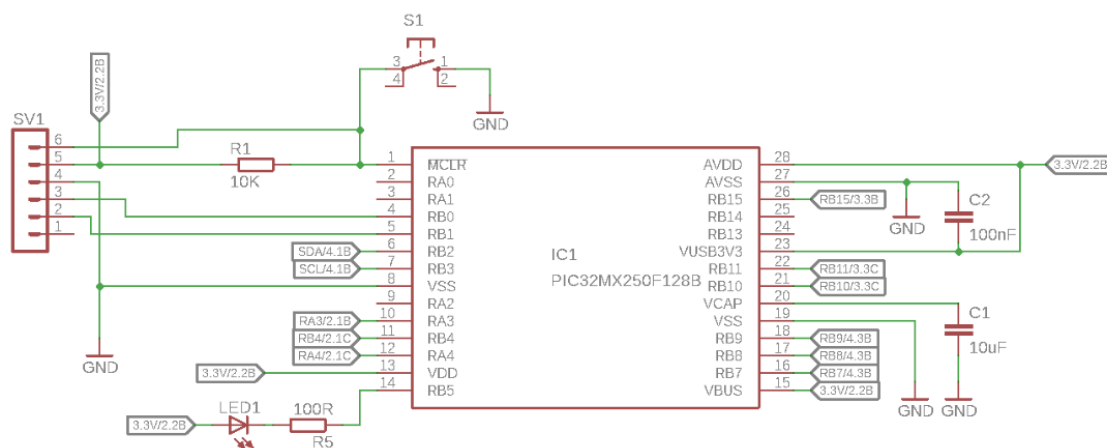
Zařízení je napájeno stejnosměrným napětím v rozsahu 8 až 12V. Vstupní stabilizátor LM7805 sice má maximální vstupní napětí 25V, ale jeho výkonová ztráta by byla obtížně chladitelná. Vstup je opatřen diodou 1N5818, která zařízení chrání před obrácenou polaritou. Napětí 5V je použito pro napájení displeje, jeho posuvného registru, IO rozhraní RS-485 a stabilizátoru LP2950L na 3.3V, který napájí ostatní obvody.



Obrázek 5.1: Schéma napájení [Příloha G]

Navržený prototyp zařízení má možnost připojit programátor bez nutnosti vypojovat mikrokontroler. Z tohoto důvodu je napájení od stabilizátoru 3,3V opatřeno jumpem, který se v případě programování odpojí a mikrokontroler je napájen z programátoru.

Jádrem celé zobrazovací jednotky je mikrokontroler PIC32MX250F128B. Schéma zapojení je podobné základnímu s tím rozdílem, že je navíc opatřen resetovacím tlačítkem a LED diodou, která signalizuje stav zařízení. Pomalé blikání je standardním stavem. Jiné je chybovou zprávou.

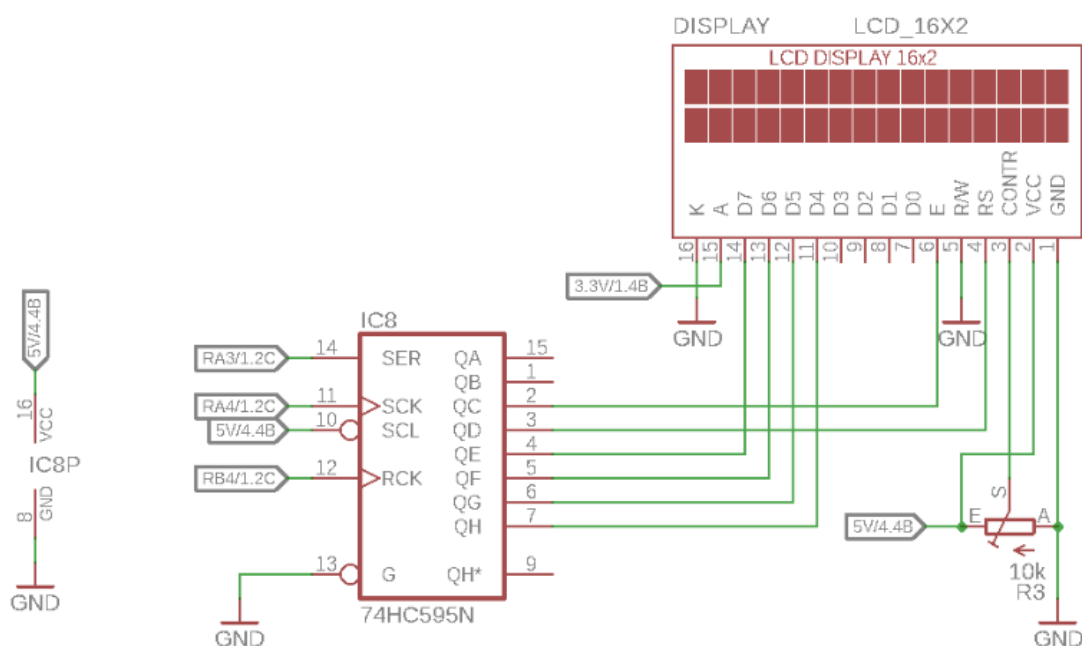


Obrázek 5.2: Schéma zapojení mikrokontroleru [Příloha C]

5.3 Displej

Pro zobrazování údajů z PLC je použit znakový displej (RC1602B-GHW-ESX) o velikosti 16x2 znaků. Tento typ je určen pro 5V logiku. Jelikož mikrokontroler PIC32MX pracuje na 3,3V, je nutné komunikaci s displejem přizpůsobit. Navíc samotná komunikace vyžaduje minimálně 6 pinů mikrokontroleru. Řešení těchto problémů je použití posuvného registru 74HC595. Nejen že se tím sníží počet obslužných pinů na tři, což je u takto malých mikrořadičů velmi vítané, ale také převede logikou úroveň 3,3V na výstupních 5V[3].

Pro řízení posuvného registru jsou použity piny RA4 pro hodinový signál (SCK), RA3 pro sériový vstup (SER) a RB4 pro obnovu výstupů (RCK). Pro komunikaci s displejem přes posuvný registr byla pro tuto práci vytvořena knihovna `lcd_shreg.h`. Displej disponuje podsvícením, které je napájeno napětím 3,3V ze stabilizátoru LP2950L.



Obrázek 5.3: Schéma zapojení displeje [Příloha D]

5.3.1 Testování grafického displeje

K dispozici mi byl i grafický TFT displej HY28A s dotykovým panelem. Komunikovat s ním lze buď sériovou linkou SPI nebo paralelním 16bitovým portem. Jelikož mikrokontroler PIC32MX250F128B nedisponuje takovým množstvím vstupně výstupních pinů, byl pro ovládání použit vývojový kit uC32 společnosti Digilent. Postaven je na mikrokontroleru Microchip PIC32MX340F512H. Je tedy ze stejné rodiny PIC32MX.

Jelikož se jedná o barevný grafický displej, je jeho obsluha nesrovnatelně složitější. Bez použití dostupných knihoven se tedy nešlo obejít. To byl ovšem problém, jelikož podpora ze strany čínského výrobce byla, jak se dalo čekat, mizivá. Nakonec byla použitelná knihovna nalezena. Byla ovšem určena na jiný procesor a bylo tedy třeba menších úprav.

Po zapnutí displeje a nahrání zkušebního textu bylo okamžitě patrné, že komunikační rychlost s displejem je natolik nízká, že zobrazování jednotlivých řádků pixelů je okem viditelné. Obnova celé obrazovky trvala téměř sekundu. To i v případě nejvyšší možné taktovací frekvenci procesoru, tedy 80Mhz. Problém byl nejspíše v použité knihovně, která nebyla určena na tento typ procesoru. Dotykový panel, který byl na obsluhu jednodušší, a tedy nebylo třeba knihoven, fungoval bez problémů. Po korekci souřadnic byl i poměrně přesný.

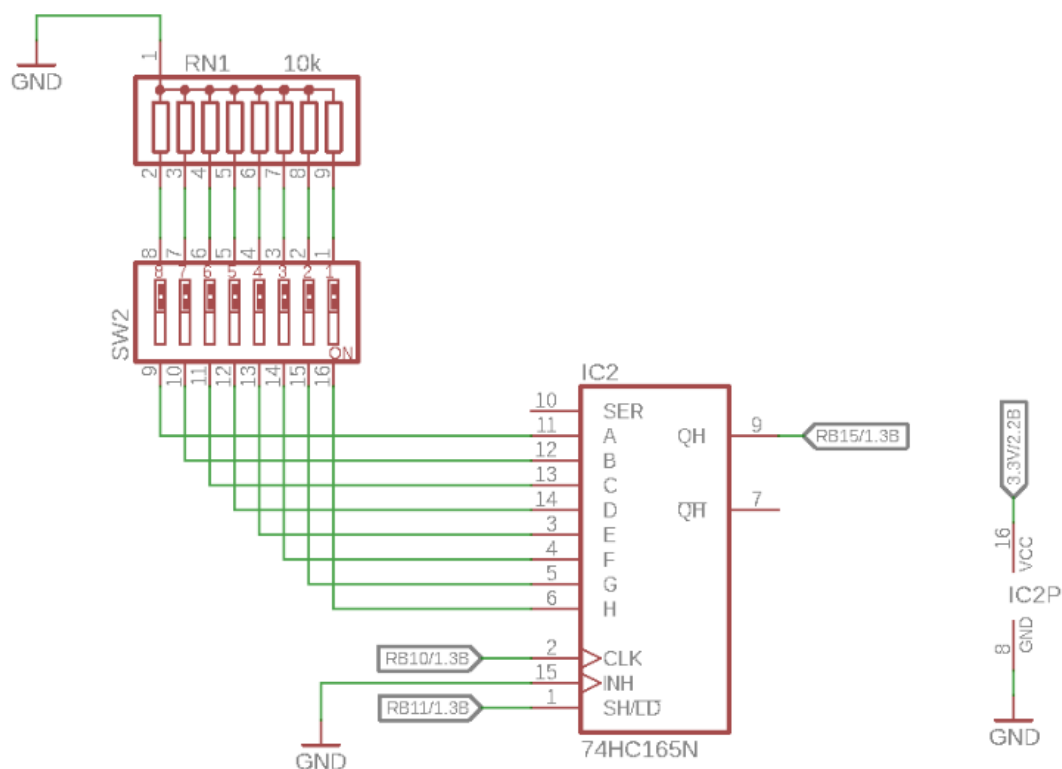
Z důvodu pomalé obnovy obrazovky nebyl displej použit. Navíc by to komplikovalo obsluhu z PLC, jelikož by to znamenalo práci s grafikou. Náročnost takového řešení ze strany protokolu a jeho obsluhy z PLC by byla nad rámec dostupného času. Od tohoto řešení bylo tedy odstoupeno.

5.4 Adresa

Aby bylo možné rozlišit jednotlivá zařízení připojená na sběrnici, je nutné definovat adresu pro každé z nich. Tato adresa musí být v dané sběrnici unikátní. Uživatel má možnost určit si adresu zařízení sám a to pomocí DIP přepínače na plošném spoji.

Je tedy na uživateli, aby zvolená adresa již nebyla použita.

Adresa je vyčtena po každém spuštění (restartu) zařízení. Pomocí posuvného registru SN74HC165 je paralelní informace o nastavené adrese odeslána sériově do mikrokontroleru. Posuvný registr má zdroj hodinového signálu (CLK) na pinu RB10. Pin RB11 určuje, zdali se bude adresa paralelně načítat nebo sériově posílat do mikrokontroleru (SH/LD). Pin RB15 je sériový výstup (QH).



Obrázek 5.4: Schéma obvodu nastavení adresy [Příloha E]

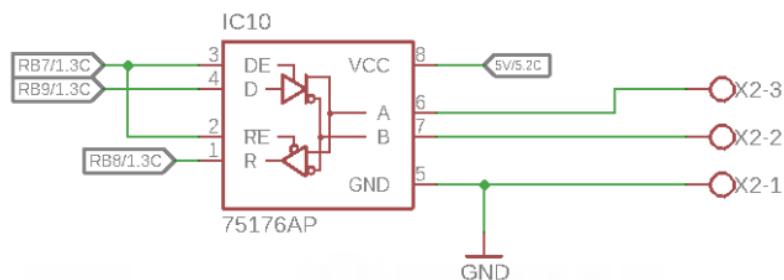
Adresa 0 je rezervovaná pro tzv. broadcast packet, tedy packet, který přijme každá stanice. Zadá-li uživatel adresu 0 do DIP přepínače, je tato adresa vyhodnocena jako chybná a jednotka hlásí rychlým blikáním LED chybu.

5.5 RS-485

Pro převod sběrnice RS485 na sériový datový signál je použit IO SN75176AP. Sériová komunikace je přivedena na piny RB9 (Tx) a RB8 (Rx) obsluhující UART2. Tyto piny jsou tolerantní na 5V, se kterými operuje převodník komunikace. Pin RB7 určuje, zda se mají data přijímat nebo vysílat. Jelikož sběrnice RS485 nemá vyvedeny signály řízení provozu komunikace, musí řízení zabezpečit použitý komunikační protokol. Sběrnice jako taková nemá definovaný konektor ani fyzické vedení. Pro připojení vodičů sběrnice byla zvolena svorka TB-5.0-P-3P.



Obrázek 5.5: Svorkovnice pro připojení sběrnice RS485 [8]

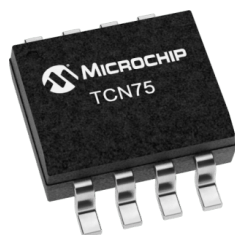


Obrázek 5.6: Schéma řevodníku RS485 [Příloha F]

5.6 Teploměr, EEPROM

Zařízení umožňuje snímat teplotu okolí. Teplotní senzor TNC75 komunikuje skrze datovou sběrnici I2C. Měří v teplotním rozsahu od -55°C až $+125^{\circ}\text{C}$ s rozlišením

0,5°C. Komunikaci obsluhují piny RB2 (SDA) a RB3 (SCL).

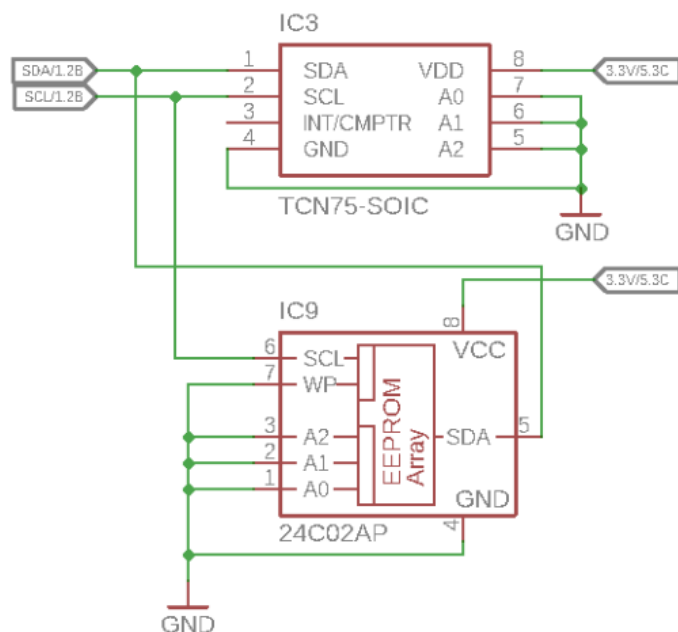


Obrázek 5.7: Teplotní senzor TCN75 [9]

Každé zařízení na sběrnici I2C musí mít svoji unikátní adresu. Adresa senzoru TCN75 je ve formátu „1001 $A_2A_1A_0$ “, kde poslední tři bity jsou určeny logickou úrovní na pinech A_0 , A_1 a A_2 . Pro jednoduchost zapojení jsou všechny adresové piny uzemněny. Adresa senzoru je tedy „1001000“.

Na stejné sběrnici je i paměť EEPROM o kapacitě 256B. Ta slouží k zapamatování dat i po odpojení napájecího napětí. Je využita pro kalibrační hodnotu teplotního čidla. Uživatel má k paměti plný přístup. PLC z ní může číst a zapisovat nová data. Kalibrační hodnota je na pozici 0 paměti EEPROM.

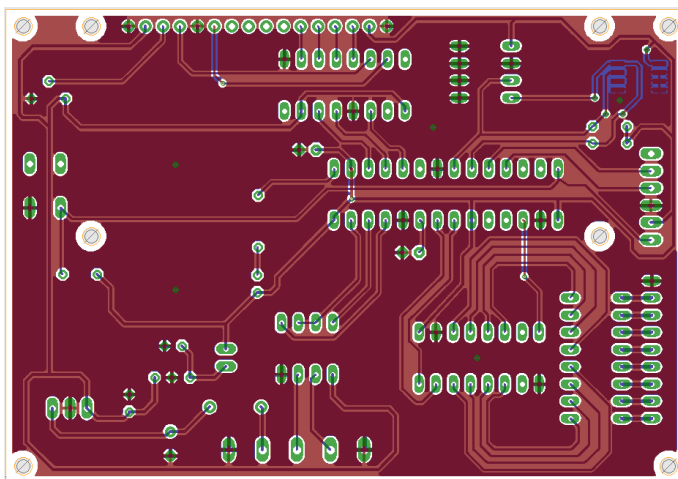
Stejně jako teplotní senzor i paměť EEPROM má svou adresu. Jeho adresa je ve formátu „1010 $A_2A_1A_0$ “. Volitelné adresové piny jsou taktéž zemněny. Výsledná adresa je „1010000“.



Obrázek 5.8: Schéma zapojení teplotního čidla a paměti EEPROM [Příloha F]

5.7 DPS

Pro zobrazovací jednotku byl navržen plošný spoj v softwaru EAGLE. Jedná se o oboustranný plošný spoj o rozměrech 100 x 70mm. Osazený je THT součástkami a jedním SMD pouzdrem teploměru TCN75. Displej je připevněn na distančních šroubech. [Příloha CD - DPS]



Obrázek 5.9: Návrh DPS

6 Program mikrokontroleru

Zdrojový kód mikrokontroleru je psán v jazyce C. Jako programovací software bylo použito vývojové prostředí od společnosti MPLAB X IDE ve verzi 3.40 s kompilátorem XC32-gcc. Celý program i s použitými knihovnami, které nejsou součástí prostředí MPLAB, jsou zahrnuty v příloze bakalářské práce. [Příloha CD - Program mikrokontroleru]

Popis programu je rozčleněn do jednotlivých tematických skupin. Není zde uváděn celý kód, ale pouze jména funkcí, popř. úseky programu, které řeší danou problematiku.

6.1 Inicializace

Ihned po spuštění je jako první ve funkci `main()` nastaven registr `ANSELA` a `ANSELB` na 0. Registr `ANSEL` určuje, jestli se má daný port (pin) chovat jako digitální I/O pin, nebo jako analogový vstup. Hodnota 0 definuje digitální I/O pin.

Dále následuje funkce `UART_Init()`, která připraví komunikaci `UART2`. Ta obstarává datové přenosy na sběrnici `RS485`. Mikrokontroler umožňuje definovat na jakých pinech bude daná hardwarová periferie naslouchat. To se nastavuje v registru `PPS` (Peripheral Pin Select).

```
CFGCONbits.IOLOCK = 0;  
CFGCONbits.PMDLOCK = 0;  
U2RXR = 0b0100;      // U2RX na RPB8  
RPB9R = 0b0010;      // U2TX na RPB9  
CFGCONbits.IOLOCK = 1;  
CFGCONbits.PMDLOCK = 1;
```

Tímto jsou piny RB8 a RB9 přiděleny periférii UART2. Následuje nastavení samotné asynchronní komunikace. Rychlost přenosu je nastavena na 9600Bd. To je zajištěno vypočítanou hodnotou z taktovací frekvence (20Mhz) a zvolenou děličkou (16 pro BRGH = 0). Výsledná hodnota (129) je uložena do registru U2BRG. Výsledná frekvence je 9615,4Mhz (chyba 0.2%).

```
U2MODEbits.BRGH = 0;           // High baud rates vypnut
unsigned long int baudRateDivider= ((SYS_FREQ/(16*9600)) - 1);
U2BRG = baudRateDivider;       // nastaven BRG (=129)
```

Dále je aktivován příjem (URXEN = 1) a nastaveno přerušování při příjmu (U2RXIE = 1). Priorita přerušování je nastavena na 1 (U2IP = 0b001). Zde je uvedeno celé nastavení komunikace.

```
U2MODEbits.IREN = 0;
U2MODEbits.RTSMD = 1;
U2MODEbits.UEN = 0b01;
U2MODEbits.ABAUD = 0;
U2MODEbits.PDSEL = 0b00;
U2STAbits.ADM_EN = 0;
U2STAbits.ADDEN = 0;
U2STAbits.URXISEL = 0b00;
U2MODEbits.LPBACK = 0;
    // UART configurace
U2MODEbits.ON = 1;
U2STAbits.UTXEN = 0;
U2STAbits.URXEN = 1;
    // UART nastavení preruseni
IFS1bits.U2RXIF = 0;
IFS1bits.U2TXIF = 0;
INTCONbits.MVEC = 1;
IEC1bits.U2RXIE = 1;
IEC1bits.U2TXIE = 0;
IEC1bits.U2EIE = 1;
IPC9bits.U2IP = 0b001;
```

Obsluha přerušování UARTInt() postupně přenáší obsah registru U2RXREG pomocí funkce ReadUART2() z knihovny uart_legacy.h do pole UINT8 zprava[64]. Navíc integruje hodnotu zp_i, která má po dokončení příjmu hodnotu počtu přijatých bytů. Dále nastaví proměnou zp_a na true, což signalizuje smyčce hlavního programu dostupnost nových dat.

```

void __ISR(_UART2_VECTOR, IPL1) UARTInt(void)
{
    IFS1bits.U2RXIF = 0;
    IFS1bits.U2TXIF = 0;
    IFS1bits.U2EIF = 0;

    if (zp_a == false)
    {
        zp_i = 0;
    }

    if(DataRdyUART2())
    {
        zprava[zp_i] = ReadUART2();
        zp_i++;
        zp_a = true;
    }
}

```

V programu následuje nastavení použitých pinů. Registr TRIS určuje, zdali je daný port (pin) výstupní nebo vstupní. Hodnota 0 je pro výstupní a 1 pro vstupní. Dále pak je nastaven časovač Timer3, který v přerušení bliká LED diodou. Funkcí Initialize_LCD() je nastavena komunikace s displejem přes posuvný registr. Jako poslední nastavenou periferií je sběrnice I2C.

Po kompletním nastavení se funkcí ReadAddress() sejme adresa zadaná na DIP přepínači. Pokud je vyčtená adresa rovna 0, je program zastaven a zařízení hlásí chybu rychlým blikáním LED. Adresa 0 je rezervovaná pro broadcast packety a tedy nemůže být použita. Pokud je nastavená adresa použitelná, program pokračuje dále a LED bliká pomocí nastaveného časovače Timer3 pomalu (perioda cca 1s).

```

MyAddress = ReadAddress();
if (MyAddress == 0)
{
    //chybna adresa
    OpenTimer3(T3_ON | T3_PS_1_256, 0x1fff);
    ConfigIntTimer3(T3_INT_ON | T3_INT_PRIOR_6);
    while(1);
}
OpenTimer3(T3_ON | T3_PS_1_256, 0xffff);
ConfigIntTimer3(T3_INT_ON | T3_INT_PRIOR_6);

```

Následně je vyčtena kalibrační hodnota teploměru uložená v paměti EEPROM. Tato 8bitová hodnota reprezentuje teplotu, o kterou se má naměřená teplota změnit.

Jelikož má teploměr rozlišení 0,5°C, musí být kalibrační hodnota také s přesností na 0,5. Hodnota je tedy ve formátu: 1.b = 0,5; 2.b = 1; 3.b = 2; 4.b = 4; 5.b = 8; 6.b = 16; 7.b = 32; 8.b = znaménko.

```
UINT8 kal = readEeprom(0);

if(bit_test(kal,6)){kal_T = kal_T + 32.0;}
if(bit_test(kal,5)){kal_T = kal_T + 16.0;}
if(bit_test(kal,4)){kal_T = kal_T + 8.0;}
if(bit_test(kal,3)){kal_T = kal_T + 4.0;}
if(bit_test(kal,2)){kal_T = kal_T + 2.0;}
if(bit_test(kal,1)){kal_T = kal_T + 1.0;}
if(bit_test(kal,0)){kal_T = kal_T + 0.5;}

if(bit_test(kal,7))
{
    kal_T *= -1.0;
}
```

Dále je zapnut Watchdog Timer s periodou cca 2s, který chrání zařízení před chybovými stavy programu (např. vlivem poklesu napájecího napětí atp.). Program následně pokračuje do hlavní nekonečné smyčky vyhodnocování příkazů.

6.2 Smyčka programu

V nekonečné smyčce while program čeká na příjem komunikace z UART2, tedy převodníku sběrnice RS-485. Pokud jsou nějaká data přijata, je v přerušení UARTInt() nastavena proměnná zp_a, která signalizuje nově přijatá data. Následně je každý přijatý byte uložen do pole UINT8 zprava[64]. Proměnná zp_i je postupně inkrementována s každým přijatým bytem. Na konci přenosu tedy obsahuje celkový počet přijatých bytů. To je velice důležitá informace pro správné zpracování paketu.

Aby nedocházelo ke zpracování ještě nekompletního paketu, je příjem prohlášen za ukončený, až když je nastaven příznak RIDLE (Receiver Idle bit). Paket je zpracován následující otočkou cyklu.

Zde je celý hlavní cyklus.

```
while (1)
{
    ClearWDT();
    if (zp_a_ok)
    {
        U2STAbits.URXEN = 0;
        U2STAbits.UTXEN = 0;
        delay(2);
        zp_a = 0;
        zpracovani();
        zp_a_ok = false;
        U2STAbits.URXEN = 1;
        U2STAbits.UTXEN = 0;
    }
    if (zp_a && U2STAbits.RIDLE == 1)
        zp_a_ok = true;
    Text();
}
```

6.3 Zpracování

Funkce zpracovani() obstarává zpracování přijatého paketu.

```
void zpracovani()
{
    bool NotBroadcast = true;
    if (CRC16(zprava, (zp_i)))
    {
        //chyba CRC
        Communication_error_count++;
        return;
    }
    Bus_Message_Count++;
    if (zprava[0] != MyAddress && zprava[0] != 0x00)
    {
        //spatna adresa
        Slave_Message_Count++;
        return;
    }

    // kod pokracuje //
```

Nejprve je paket zkontrolován funkcí CRC16(puchMsg, usDataLen) [5], která provede cyklický redundantní součet s řídicím řetězcem protokolu MODBUS. Pokud je výsledek výpočtu 0, je paket prohlášen za korektní a pokračuje se ve zpracování.

V opačném případě je paket ignorován a je inkrementován počet chybných paketů. Dále je porovnávána adresa zařízení s adresou v hlavičce paketu. Je-li adresa shodná nebo je rovna 0 (broadcast) pokračuje se ve zpracování. V opačném případě je paket taktéž ignorován a je inkrementován celkový počet přijatých paketů. Následuje zjištění kódu funkce a spuštění příslušné části programu.

6.4 Funkce

Zařízení v současném stavu, který je prohlášen za ukončený, obsluhuje několik funkcí, které lze z PLC vyvolat. Všechny jsou dále uvedeny a popsány.

6.4.1 Zobrazování na displej

Hlavní funkce celého zařízení má kód 0xAA a 0xAB (psáno hexadecimálně). Každý řádek má tedy vlastní kód funkce. Lze ji vyvolat i jako broadcast. Maximální délka textu zobrazeného na displeji je 31 znaků (32. znak je znak ukončení řetězce).

Funkce, která obsluhuje zadávání textu na displej je `New_Text(UINT8 row)`. Operátor `row` určuje, na jaký řádek se má text zapsat. Displej je dvouřádkový a každý řádek má své pole pro uchování zobrazovaného řetězce (pole `char Text1[32]` a `Text2[32]`). Funkce `New_Text(UINT8 row)` zkopíruje datový obsah paketu do příslušného pole. Cyklus kopírování ukončí znak „null“ (ukončení řetězce) nebo dosažení maximálního počtu znaků, čili 32.

Po ukončení funkce `New_Text(UINT8 row)` je v hlavní smyčce spuštěna funkce `Text()`, která zobrazuje řetězce `Text1[32]` a `Text2[32]` na displej. Aby nedocházelo ke zbytečnému neustálému zobrazování textu na displeji každou otočkou cyklu, je text obnovován periodou 500ms. V případě rolování textu je začátek a konec řetězce zobrazen na dobu 1s.

6.4.2 Měření teploty

Zařízení umožňuje měřit teplotu okolí pod kódem 0xB0. Tuto funkci nelze vyvolat broadcast paketem, jelikož by došlo k požadavku o čtení od všech stanic.

Obsluhující funkcí je `ReadTemperature()`. Přes rozhraní I2C vyčte z teploměru TCN75 údaj z převodníku, ze kterého je vypočítána naměřená teplota. Ta je přičtena ke kalibrační hodnotě a odeslána jako datový typ float funkcí `SendMessage(UINT8 F_code, unsigned char* Data, UINT8 DataLen)`. Ta vytvoří paket spolu s kontrolou CRC16, vypočítanou funkcí `unsigned short CRC16(puchMsg, usDataLen)` a odešle přes UART2 do převodníku sběrnice RS-485.

6.4.3 Zapisování do paměti EEPROM

Funkce zapisování do paměti EEPROM má kód 0xA1. Je obsluhována funkcí `writeEeprom(UINT8 pointer, UINT8 dataB)` a je dostupná přes broadcast paket.

Obsluhující funkce má dva operátory. Operátor `pointer` udává pozici bytu v paměti EEPROM a `dataB` hodnotu, která se na danou pozici uloží. Paměť je velikosti 256B. Na pozici 0 je uložena hodnota pro kalibraci teploměru.

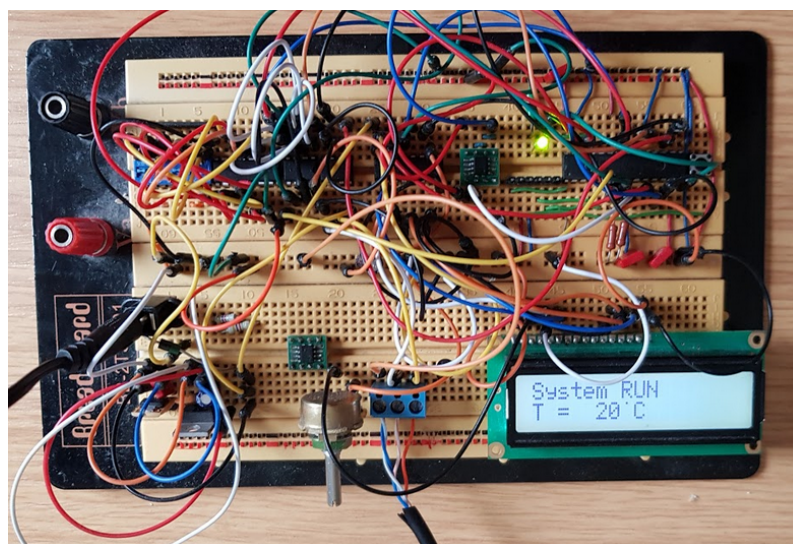
6.4.4 Čtení z paměti EEPROM

Funkce čtení z paměti EEPROM má kód 0xA2. Obsluhována je funkcí `UINT8 readEeprom(UINT8 pointer)`. Jelikož vrací hodnotu uloženou v paměti, nemůže být použita jako broadcast.

Operátorem funkce je proměnná `pointer`, která určuje pozici bytu určeného ke čtení.

7 Testování

Zobrazovací jednotka byla testována na nepájivém poli. Svým zapojením i použitými součástkami je identická od návrhu DPS (vyjma použitého potenciometru a redukce SMD).

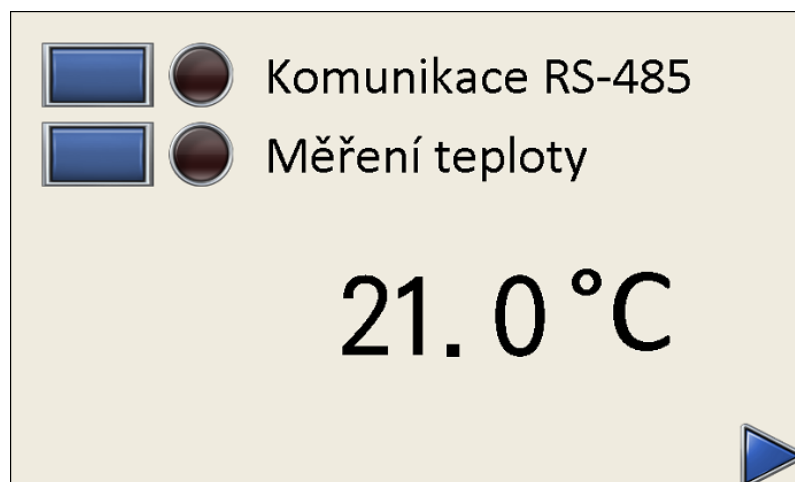


Obrázek 7.1: Testovací vzorek zobrazovací jednotky

Pro ověření funkčnosti zařízení byl vytvořen program v PLC, který využije všechny dostupné funkce navržené jednotky. PLC tedy zobrazuje údaje na displeji, vyčítá naměřenou teplotu a pracuje s daty v paměti EEPROM. [Příloha CD - Program PLC]

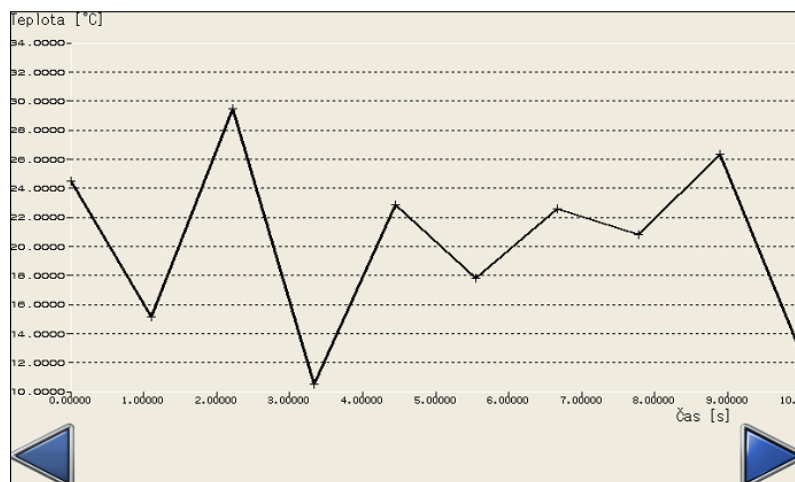
Na první obrazovce operačního panelu PLC je možné spustit komunikaci po sběrnici RS-485. Po její aktivaci je v PLC spuštěna funkce RS2, která obstarává Non-protocol komunikaci a aktivována navržená funkce pro obsluhu protokolu. Poté je možné spustit odesílání požadavků na měření teploty. Je-li měření teploty aktivová-

no, na 1. řádku displeje jednotky je zobrazen text „System RUN“ a v pravidelných intervalech je zobrazována naměřená teplota, a to jak na operačním panelu PLC, tak i na 2. řádku displeje jednotky, kde je daná teplota zobrazena celočíselně. Po dalším stisku ovládacího tlačítka je měření vypnuto a na displeji jednotky je zobrazena zpráva „System STOP – by user“.



Obrázek 7.2: První obrazovka - měření teploty

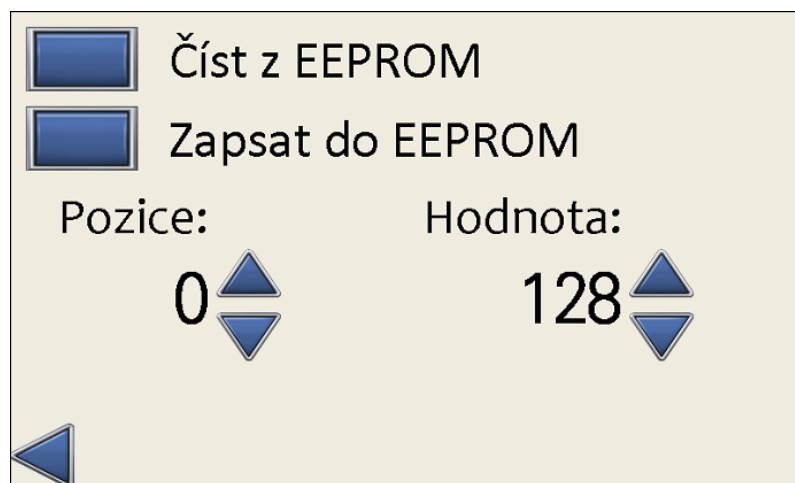
Na druhé obrazovce jsou naměřené údaje graficky zpracovány. Na ose x je zobrazen čas stárí vzorku. Pro lepší znázornění rychlých změn je zobrazeno pouze 10s.



Obrázek 7.3: Druhá obrazovka - graf teploty

Třetí obrazovka slouží pro čtení a zápis do paměti EEPROM. Je možné zadat pozici

v paměti a hodnotu, které se má uložit. Při čtení z paměti je pro přijatá hodnota zobrazena v proměnné pro nastavení hodnoty. V PLC je při aktivaci měření teploty přístup k EEPROM zobrazovací jednotky blokován.



Obrázek 7.4: Třetí obrazovka - práce s pamětí EEPROM

7.1 Výsledek

Při stisknutí tlačítka pro aktivaci komunikace je rozsvícena signalizační kontrolka. Poté, při stisku tlačítka „Měření teploty“, se na displeji jednotky okamžitě zobrazí zpráva „System RUN“. Po uplynutí intervalu 1,5s je zobrazena na druhém řádku naměřená teplota bez desetinných míst. V každém následujícím intervalu 1,5s je hodnota obnovována. Ta samá hodnota je i na grafickém panelu PLC. Při opětovném stisku tlačítka „Měření teploty“ je měření teploty ukončeno a na 1. řádku displeje jednotky je zobrazena hláška „System STOP - by user“. Tato zpráva je záměrně delší, než je rozsah displeje (16 znaků). Text je tedy rolován a zobrazován postupně. Po opětovném stisku se znovu zobrazuje měřená teplota. Při deaktivaci „Komunikace RS-485“ je komunikace zastavena a tlačítko „Měření teploty“ je nečinné.

Komunikace s jednotkou tedy funguje a lze zobrazovat údaje na displeji, a to i takové, které se na něj celé nevejdou. Také teplotu lze vzdáleně měřit a vyčítat.

Na druhé obrazovce je při měření teploty zobrazena historie přijatých hodnot. Při přidržení prstu na teplotním senzoru je okamžitě vidět nárůst teploty. Údaje v grafu jsou při opuštění obrazovky smazány a graf je vyčištěn, to je způsobeno tím, že údaje jsou ukládány do vyrovnávací paměti displeje GOT, která je využita vždy pro danou obrazovku.

Při čtení z paměti EEPROM zobrazovací jednotky z adresy 0 je zobrazena hodnota 137. Binárně je to „10001001“, což znamená, že kalibrační hodnota je záporná (8. bit je 1) a má hodnotu -4,5 (4. a 1. bit je 1). Když je zapsána na adresu 0 jiná hodnota, je po restartu jednotky změněna i hodnota naměřené teploty. Po přepsání hodnoty na jiné adrese je při opětovném čtení skutečně změněna. Čtení a zápis do paměti EEPROM tedy funguje korektně.

8 Závěr

Dle bodů zadání a dostupných komponentů bylo navrženo a zprovozněno zařízení, které dokáže komunikovat po sběrnici RS-485 a zobrazovat údaje na znakový displej. Navíc bylo opatřeno teplotním čidlem, které umožňuje vzdáleně měřit teplotu v okolí zařízení. Pro potřeby komunikace byl vytvořen komunikační protokol po vzoru protokolu MODBUS.

Od začátku návrhu byl kladen důraz na univerzálnost výsledného zařízení. Zobrazovací jednotku lze libovolně adresovat. To umožňuje zapojení většího množství jednotek na jednu sběrnici RS-485. Navržený protokol je možné jednoduše rozšířit o další funkce, které by eventuálně mohly být obsaženy v dalších verzích tohoto zařízení. Na tuto práci by mohlo být v budoucnu navázáno implementací protokolu MODBUS RTU a rozšířením jednotky o další periferie, např. relé, analogové vstupy, měření teploty externím teploměrem (např. Pt100) atd. Velice zajímavé by bylo zařízení přepracovat na modulární systém. To by zákazníkovi umožnilo sestavit si z jednotlivých modulů zařízení dle svých požadavků.

V průběhu této práce jsem si ověřil a místy i zdokonalil své dosavadní znalosti v programování mikrořadičů PIC v jazyku C. Přínosem byla i práce se sběrnici RS-485, která se v průmyslu stále využívá. Tvorba použitého protokolu mě podrobně seznámila s problematikou komunikace po této sběrnici. Samozřejmě nemohu pominout využití znalostí z oboru elektroniky při návrhu zapojení a volby vhodných součástek. Velkým přínosem pro mě byla práce s daty a obsluha komunikace v PLC.

Literatura

- [1] *AXELSON, Jan. Serial Port Complete COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems* [online].
2nd ed. Madison: Lakeview Research, 2007 [cit. 2018-05-06].
ISBN 978-193-1448-079. Dostupné z: [http://www.singlix.com/trdos/archive/serial/%5BElectronics%5D%20Serial%20Port%20Complete%20-%20Com%20Ports,%20Usb%20Virtual%20Com%20Ports,%20And%20Ports%20For%20Embedded%20Systems,%20Second%20Edition%20\(2009\).pdf](http://www.singlix.com/trdos/archive/serial/%5BElectronics%5D%20Serial%20Port%20Complete%20-%20Com%20Ports,%20Usb%20Virtual%20Com%20Ports,%20And%20Ports%20For%20Embedded%20Systems,%20Second%20Edition%20(2009).pdf)
- [2] *DI JASIO, Lucio. Programming 32-bit Microcontrollers in C.*
Newnes, 2008. ISBN 978-075-0687-096.
- [3] *MATOUŠEK, David. Aplikace mikrokontrolérů PIC32MX.*
BEN - technická literatura, Praha 2014. ISBN 978-807-3005-047
- [4] *Papouch*[online].2018 Web společnosti Papouch.
Dostupné z: <https://www.papouch.com/cz/website/mainmenu/mainpage/>
- [5] *MODBUS.ORG* [online].2002 **Specification & Implementation guide.**
Dostupné z: http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf
- [6] *[datasheet]: PIC32MX1XX/2XX* [online].
Microchip Technology, 2011 [cit. 2018-05-06].
Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/61168D.pdf>
- [7] [PLC Mitsubishi FX5U] [online]. In: . [cit. 2018-05-06]. Dostupné z: <https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/images/56af62f425704856947e65d80ab544b9FX5U.jpg?W=560>
- [8] *[Svorkovnice]* [online]. In: . [cit. 2018-05-06]. Dostupné z: http://www.avelmak.sk/media/img_items_d/2/28/284/28443.jpg
- [9] *[TCN75]* [online]. In: . [cit. 2018-05-06]. Dostupné z: http://www.microchip.com/_images/ics/medium-TCN75-SOIC-8.png

A Seznam příloh na CD

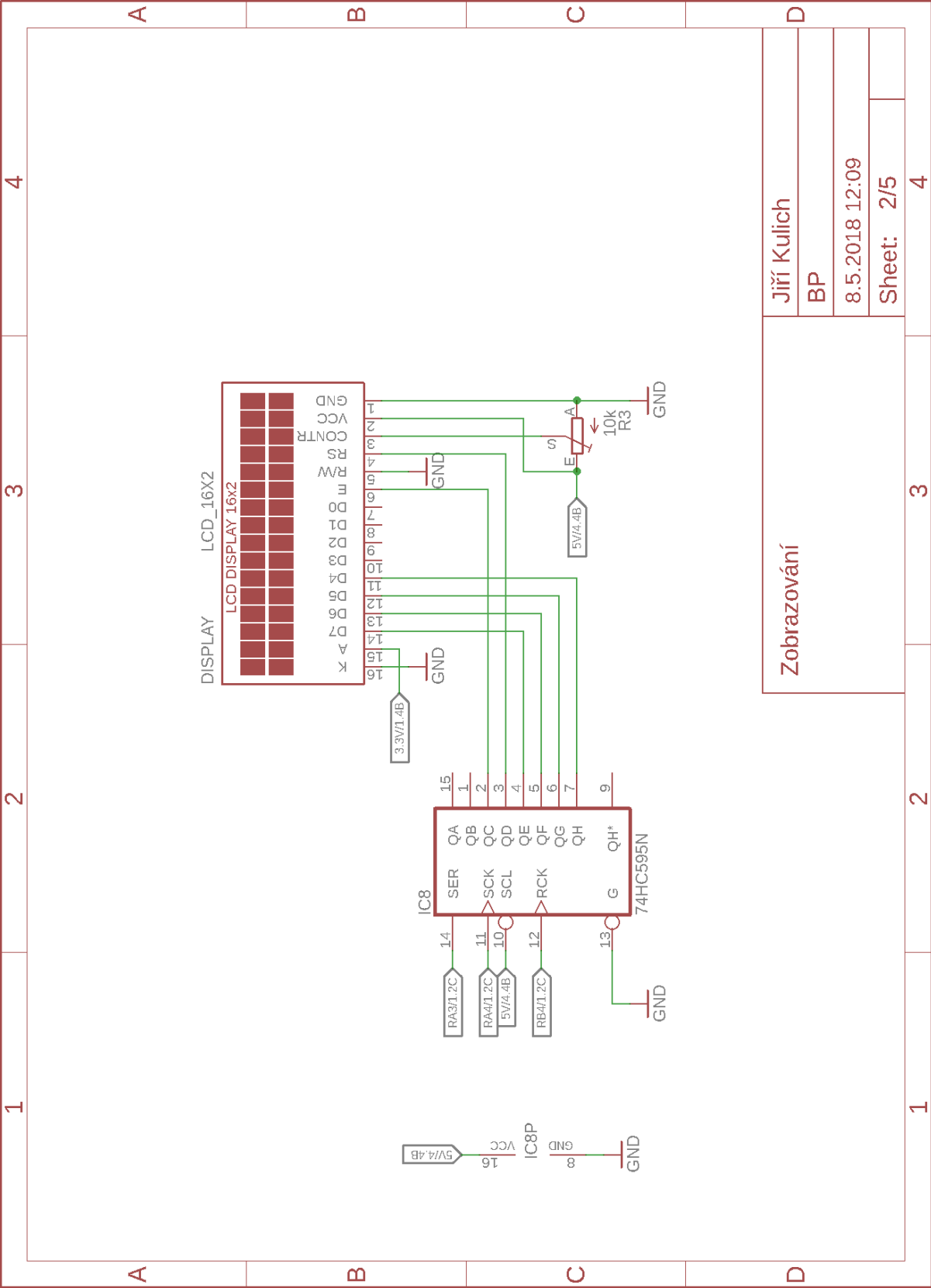
- DPS
 - BP.brd
 - BP.lbr
 - BP.sch
- Fotografie
 - Animace zobrazení.gif
 - Obrazovka 1.jpg
 - Obrazovka 2.jpg
 - Obrazovka 3.jpg
 - Zobrazovací jednotka.jpg
- Katalogové listy
 - 24C02.pdf
 - LM7805.pdf
 - lp2951-n.pdf
 - PIC32MX.pdf
 - sn74hc165.pdf
 - sn74hc595.pdf
 - sn65176b.pdf
 - TCN75.pdf
- Program mikrokontroleru
 - CRC_Generator.c
 - CRC_table.h
 - lcd_shreg.c
 - lcd_shreg.h
 - Main.c
- Program PLC
 - PLC.pdf
 - GOT.pdf
- Text bakalářské práce
 - Bakalarska_prace_2018_Jiri_Kulich.pdf
 - Kopie_zadani_bakalarske_prace_2018_Jiri_Kulich.pdf

B Seznam použitých součástek

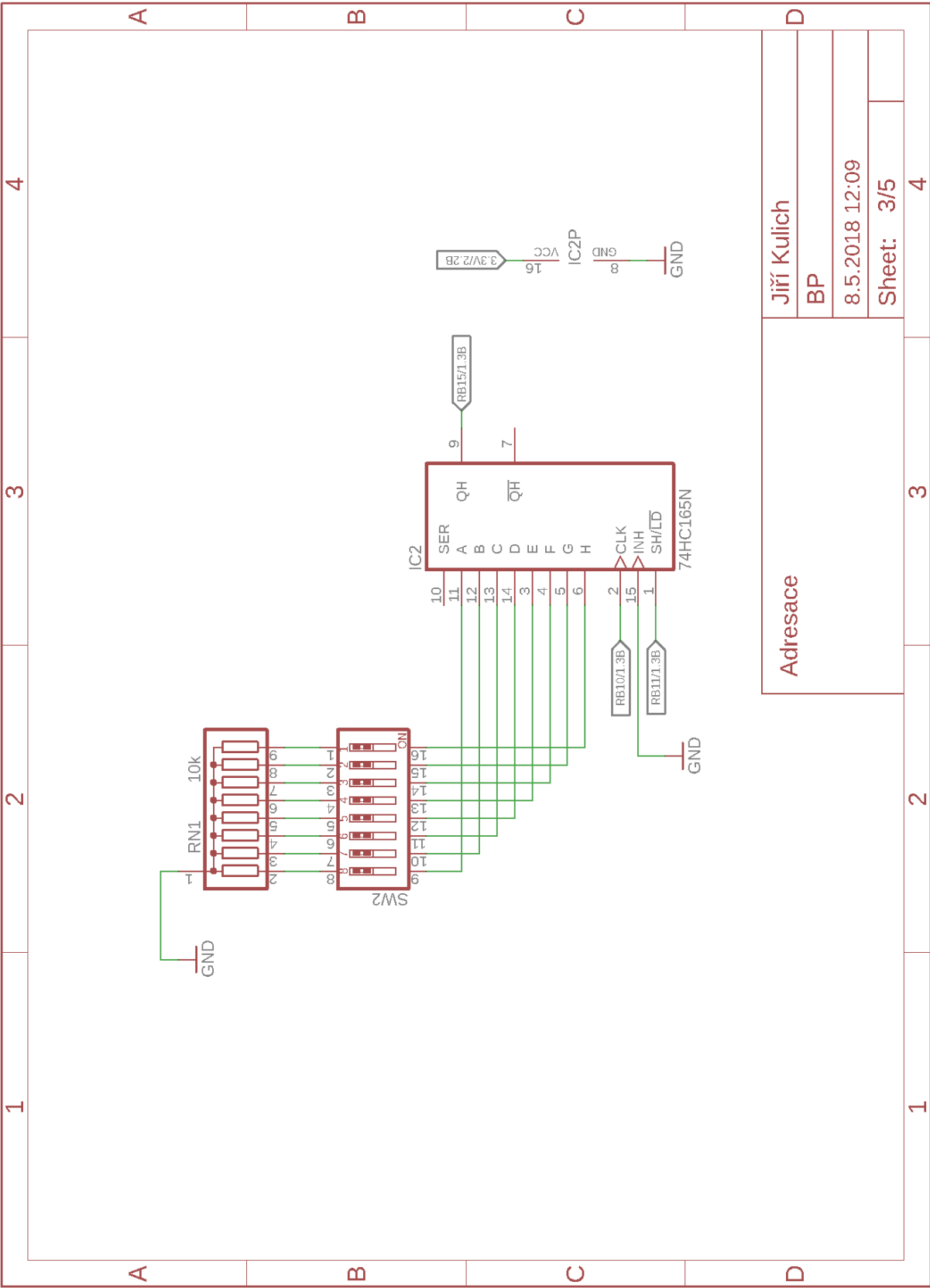
- PIC32MX250F128B
- RC1602B-GHW-ESX
- LM7805
- LP2950L
- 1N5818
- 74HC595
- SN74HC165
- TCN75
- EEPROM M24C02
- TB-5.0-P-3P
- EDG108S
- LED L-934LID
- RRA 8x10k
- Mikrotlačítko 6x6mm
- CA14NV12,5-10KA2020
- 6Pin 2,54mm (M)
- 16Pin 2,54mm (M)
- Rezistor 100R
- Rezistor 4k7 2x
- Rezistor 10k
- 33uF/35V
- CCK-100N 3x
- T350E106K025AT

[illegible]

D Schéma – zobrazování



E Schéma – adresace



Jiří Kulich

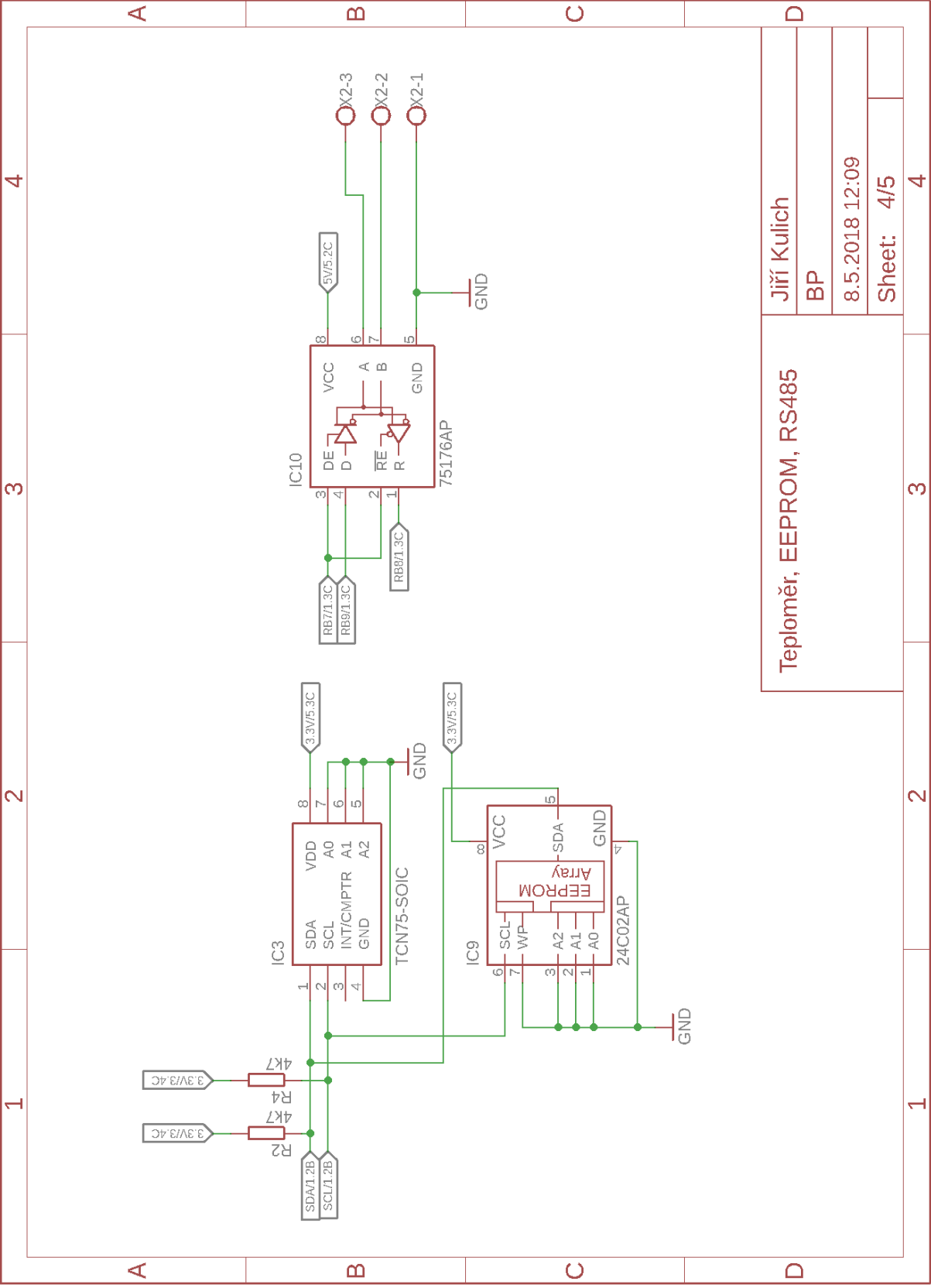
BP

8.5.2018 12:09

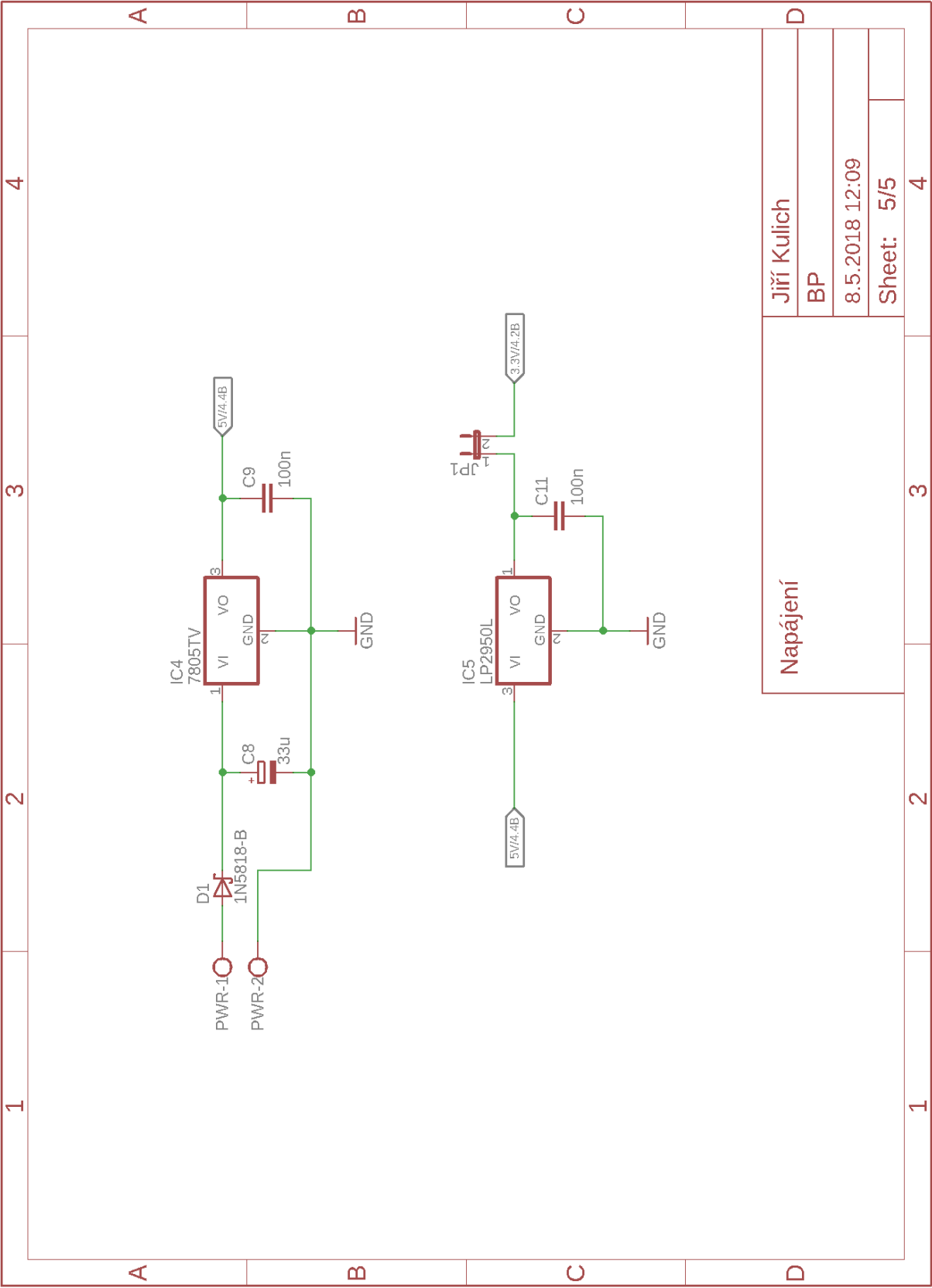
Sheet: 3/5

Adresace

F Schéma - teploměr, EEPROM, RS485



G Schéma - napájení



Napájení	Jiří Kulich
	BP
	8.5.2018 12:09
	Sheet: 5/5