

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

JEDNODUCHÁ VZDÁLENÁ SIGNALIZACE
PLC SYSTÉMŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

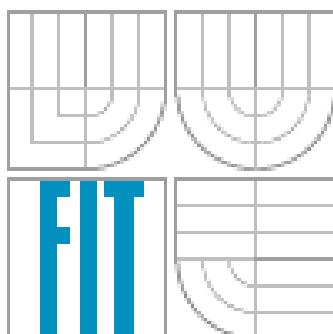
AUTOR PRÁCE
AUTHOR

JAN KACÁLEK

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

JEDNODUCHÁ VZDÁLENÁ SIGNALIZACE PLC SYSTÉMŮ

SIMPLE REMOTE SIGNALLING OF PLC SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN KACÁLEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. VÍT TŘÍSKA

BRNO 2014

ZADÁNÍ

Abstrakt

Cílem této bakalářské práce je návrh a zhotovení hardwarového a softwarového zařízení sloužícího pro demonstraci komunikace s PLC systémy. Dále pak práce shrnuje možnosti této komunikace a částečně implementuje protokol MODBUS přes sériovou linku RS485. Hardwarové zařízení je zhotoveno na desce plošných spojů a ze součástek běžně dostupných v obchodech s elektronikou. Obsahuje LCD displej a LED diody pro zobrazení stavu PLC systému. Softwarová aplikace je napsána v jazyce Java a implementuje stejnou funkčnost jako hardwarové zařízení.

Abstract

The aim of this thesis is the design and manufacture of hardware and software facilities used for the demonstration of communication with PLC systems. Furthermore, this paper summarizes the possibilities of communication and partially implements the MODBUS protocol via RS485 serial line. The hardware device is fabricated on the PCB and the components commonly available in electronics stores. Includes LCD display and LEDs for status display PLC system. The software application is written in Java and implements the same functionality as the hardware device.

Klíčová slova

Bakalářská práce, PLC systémy, MODBUS, PROFIBUS, RS-485, RS-232, USB, SimplyModbusMaster, SimplyModbusSlave, PureJavaComm, PLC protokoly, POE, LCD displej, DPS, Atmega16, transceiver SN75176

Keywords

Bachelor's thesis, PLC systems, MODBUS, PROFIBUS, RS-485, RS-232, USB, SimplyModbusMaster, SimplyModbusSlave, PureJavaComm, PLC protocols, POE, LCD display, PCB, ATmega16, transceiver SN75176

Citace

Jan Kacálek: Jednoduchá vzdálená signalizace PLC systémů, bakalářská práce, Brno, FIT VUT v Brně, 2014

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Víta Třísky. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Kacálek
20. května 2014

Poděkování

Děkuji svému vedoucímu Ing. Vítu Třískovy za odborné vedení a poskytnuté rady, podněty a připomínky, které mi pomohly vypracovat tuto práci.

© Jan Kacálek, 2014

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1.	Úvod	- 9 -
2.	Co je to PLC?	- 10 -
2.1.	Cyklické vykonávání programu	- 10 -
2.2.	Softwarové vybavení.....	- 11 -
2.3.	Komunikace s PLC	- 12 -
2.4.	Komunikační protokoly	- 13 -
3.	Zobrazovací panel	- 15 -
3.1.	Konstrukce	- 15 -
3.2.	Řídící procesor.....	- 15 -
3.3.	LCD displej.....	- 16 -
3.4.	Zvukový signál	- 17 -
3.5.	Programování a ladění	- 18 -
3.6.	Napájení panelu	- 18 -
3.7.	Přenos signálu	- 18 -
4.	Náhrada za PLC	- 21 -
5.	Deska plošných spojů a konstrukční řešení	- 22 -
6.	Bližší pohled na linku RS-485	- 23 -
7.	Firmware výrobku	- 24 -
7.1.	Implementace protokolu Modbus	- 24 -
7.2.	Obslužný program vysílače	- 25 -
7.3.	Obslužný program panelu.....	- 27 -
8.	Program pro PC.....	- 29 -
8.1.	GUI aplikace	- 29 -
8.2.	Připojení k USB portu	- 30 -
8.3.	Vnitřní logika	- 31 -
8.4.	RS485-USB převodník.....	- 33 -
9.	Testování.....	- 34 -
9.1.	Testování vysílače	- 34 -
9.2.	Testování panelu	- 35 -
10.	Závěr.....	- 36 -
	Literatura	- 37 -

A.	Schémata zapojení	- 38 -
B.	Seznam použitých součástek	- 40 -
C.	Vzhled výsledných produktů	- 41 -
D.	Manuál	- 43 -
E.	Obsah CD.....	- 44 -

1. Úvod

Cílem této práce bylo navrhnout a zrealizovat zobrazovací panel, komunikující s programovatelným logickým automatem, dále jen PLC.

PLC se v dnešní době používají zejména v automatizaci, řízení procesů výroby nebo shromažďování dat. Při použití v řízení výrobního procesu neboli ovládání strojů v továrnách je někdy důležité zavčas zareagovat na vzniklý problém nebo neočekávaný stav. Aby nebylo nutné mít jednoho pracovníka u každého stroje po celou dobu výrobního procesu, je vhodné mít stroje propojené s nějakým typem panelu, který informuje obsluhu na tento neočekávaný stav.

Dle návrhu bude panel schopen přijímat informace z PLC systému a zobrazovat několik typů chyb. K tomu bude sloužit signalizace pomocí LED diod, zobrazení informací na LCD displeji a zvukové upozornění.

Jelikož by bylo obtížné ladit aplikační program přímo za běhu nějakého procesu, počítá se v návrhu s vytvořením zařízení, které bude simulovat vznik chyby v nějakém ze strojů a pomocí vhodného přenosového média a protokolu přeneset tuto informaci do panelu. Možnost otestování panelu přímo v procesu bude diskutováno s firmou OEZ s.r.o. zabývající se výrobou jističů.

Dalším cílem této práce je vytvoření softwarové varianty zařízení běžícího pouze v PC, ale schopného zpracovávat stejné vstupní informace jako výše zmiňovaný produkt.

2. Co je to PLC?

Programmable logic controller neboli programovatelný logický automat je digitální počítač používaný převážně pro automatizaci v průmyslové výrobě. Je vhodný jak pro řízení jednotlivých strojů, jednoduchých procesů a výrobních linek tak i řízení podsystémů velkých technologických procesů.

PLC používá programovatelnou paměť pro uložení uživatelsky orientovaných instrukcí sloužících k implementaci specifických funkcí, jako jsou logické funkce, funkce pro vytváření sekvencí, funkce pro časování, sčítání a aritmetické výpočty.

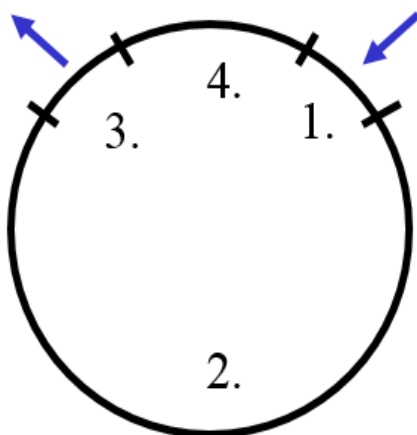
Základními komponentami logických automatů jsou PLC procesory, I/O (vstupně/výstupní) moduly, šasi, napájení, programovatelný software běžící v PC.

Ve svých začátcích byly tyto programovatelné automaty programovány pomocí takzvané "ladder" logiky, neboli jazyku žebříčkových diagramů, tento jazyk se nejvíce podobá schématickým diagramům reléové logiky, která byla užívána v automatizaci před příchodem prvních PLC. V dnešní době je možné automaty programovat jak pomocí "ladder-logic" tak pomocí strukturovaný jazyků jako jsou například BASIC nebo C (podle standardu IEC-1138).

PLC můžeme dělit dle několika kritérií. Například dle provedení (kompaktní, modulární, se zabudovaným operátorským panelem, vanové), velikosti (mikro PLC, malá PLC, střední a nejvyšší výkonnostní třída) nebo počtu procesorů (jednoprocesorová, víceprocesorová).

Jádrum PLC je CPU (Central processor unit), která určuje jeho výkonnost a může být jednoprocesorová a víceprocesorová (matematické koprocessory, komunikační procesory, I/O procesory). Liší se pro každý typ automatu a operační rychlost je posuzována podle doby cyklu (doba zpracování 1000 logických instrukcí) většinou 10^{-1} až 10^1 milisekund.

2.1. Cyklické vykonávání programu



Obrázek 2.1.1 – Cyklické vykonávání programu v PLC [9]

1. Čtení vstupů

- Detekuje stav všech vstupních zařízení, které jsou připojeny k PLC

2. Zpracovávání programu

- Spustí uživatelem definovaný program

3. Vysílání výstupů

- Spustí nebo vypne všechny výstupní zařízení připojené k PLC

4. Režie systému, (aktualizace systémových proměnných, komunikace)

Na obrázku 2.1.1 je zobrazen cyklus logických automatů. Při návrhu programu je nutné dávat pozor na několik zásad a to zejména že reakční doba na změnu hodnoty senzoru je až dvojnásobek doby pro vykonávání cyklu. Záleží na pořadí instrukcí nad stejnými proměnnými a na výstup se posílá vždy naposledy zpracovaná proměnná.

Paměťový prostor CPU se dělí na několik částí.

- Systémová paměť – systémový program (neboli operační systém PLC), je uložena v paměti EPROM.
- Paměť dat – systémové registry, uživatelské registry, zápisníkové registry (marker, flagy), čítače a časovače, vyrovnávací registry pro obrazy vstupů a výstupů. Uložena v paměti RAM (RWM).
- Uživatelská paměť – uživatelský program (soubor tzv. procesů), uložena v paměti EPROM (popř. EEPROM).

2.2. Softwarové vybavení

Systémový program (operační systém) je hardwarově závislý – každý výrobce má jiný systém. Zpracovávání uživatelského programu probíhá v jedné úloze a bez přerušení. Některé typy mají možnost více úlohového zpracování (tzv. multitasking) na jednom procesoru případně i s přerušením. Pro správné využití možností určitého OS je nutné dodržovat specifická pravidla pro:

- vzájemné spouštění a odstavování programů (procesů, úloh)
- pro předání dat

Přerušení je velmi účinný prostředek, který umožňuje zkrátit odezvu systému na kritické situace. Kterýkoliv z běžných procesů smyčky může být přerušen některým z přerušovacích procesů (přerušení od vzniklé chyby, přerušení od periferního vstupu). Instrukce rozpracovaná v okamžiku přerušení se dokončí a po ní je provedena první instrukce přerušujícího procesu, který ale nemění stav žádné úrovně aktivního zásobníku.

Pro naprogramování PLC se používají specializované jazyky. Tyto jazyky jsou opět u každého výrobce trochu odlišné a přenositelnost programů je tak možná pouze mezi systémy stejného výrobce. Norma IEC 1131-3 definuje 4 typy jazyků.

- Jazyk mnemokédů

Strojově orientovaný jazyk podobný assembleru, poskytuje symbolické označení návěstí pro cíle skoků a volání, pro pojmenování vstupních a výstupních proměnných a pro automatické přidělování paměti pro uživatelské programy.

- Jazyk konstantních (reléových) schémat

Jedná se o grafický jazyk, což znamená, že program se zobrazuje ve formě symbolů specifických pro kreslení schémat. Instrukce, které nejsou obvyklé v kontaktní symbolice, se obvykle zobrazují jako dvojice závorek. Tento jazyk je výhodný zejména při programování jednodušších operací.

- Jazyk logických schémat

Základní logické operace popisuje pomocí obdélníkových značek. Své značky mají nejen logické operace ale i funkční bloky jako jsou například čítače, časovače, posuvné registry nebo paměťové členy. Tento jazyk je vhodný pro uživatele zvyklé na kreslení logických schémat pro zařízení s integrovanými obvody.

- Jazyk strukturovaného kódu

Obdoba vyšších programovacích jazyků pro PC nebo mikroprocesory. Vhodný zejména pro uživatele zvyklé programovat právě pro mikroprocesory.

2.3. Komunikace s PLC

Komunikaci s PLC provádí komunikační jednotky, které umožňují nejen komunikaci s rozhraním ale i mezi jednotkami vstupů a výstupů, s podsystémy, se souřadnými a nadřizenými systémy a s operátorskými panely. K dispozici jsou nejen jednotky komunikující po standardním drátovém vedení ale i moduly umožňující dálkové přenosy dat přes modem, radiomodem nebo telefonní síť.

Sériová linka RS-232

Sériová linka nebo sériový port. Používá se jako komunikační rozhraní osobních počítačů a další elektroniky. K přenosu používá dvě napěťové úrovně. Logickou 1 a 0. Logická 1 označuje klidový stav a je indikována zápornou úrovní. Logická 0 je nazývána space state a je přenášena kladnou úrovní výstupních vodičů. Pro komunikaci se u PLC používá k propojení dvou bodů na maximální vzdálenost 15 metrů. Pro přenos potřebuje vyšší napětí než je obvyklé a to až 12V. V drátové komunikaci má nejvyšší přenosovou rychlost až 57600kbps.

Sériová linka RS-422

Založena na přenosu dat pomocí dvou vodičů, přičemž na přijímací straně se při zjišťování hodnoty bitu rozlišuje rozdíl (polarita) napěťových potenciálů mezi těmito vodiči (diferenciální přenos dat). Pro rozeznání logické jedničky či nuly postačuje rozdíl potenciálů dosahující hodnoty pouze 200mV. Při základním zapojení sběrnice, po nichž se vysílají data s diferenciálním kódováním, lze přenos provádět až na vzdálenost 1200 metrů při přenosové rychlosti do 500kbps. Při kratších vzdálenostech se může přenosová rychlost zvýšit až stonásobně. Opět slouží k propojení 2 bodů, vysílač by měl udržovat napěťové úrovně na datových vodičích na hodnotách +6 nebo -6 V.

Sériová linka RS-485

Pomocí této sběrnice může komunikovat maximálně 32 vysílačů a 32 přijímačů, ale pouze jedno zařízení na sběrnici může v dané čase pracovat jako řadič (vysílač). Používá se diferenciálního kódování dat. Mezi oběma napěťovými potenciály se používá rozdílů 5 V, 7 V či 12 V. Maximální vzdálenost dosažitelná za běžných podmínek je cca 1200 metrů, přenosová rychlost může dosáhnout hodnot až 10Mbit/s.

USB

Neboli Univerzální sériová sběrnice. Jedná se o externí sběrnici, ve které jsou jednotlivá komunikující zařízení propojena systémem point to point a ne skutečnou sběrnici tvořenou společnými vodiči. Topologické schéma USB je založeno na struktuře víceúrovňové hvězdice, přesněji řečeno stromové struktuře. Kořenem stromu je zařízení, které komunikaci řídí (master), ostatní zařízení (slave) naslouchají a může jich být až 127. V případě použití u PLC se používá spojení pouze mezi dvěma body a to na maximální vzdálenost 5m. Přenosová rychlost je max. 12Mbit/s. Použití nachází většinou u programování a servisu PLC.

Ethernet

Standardizován jako IEEE 802.3, používá kroucenou dvoulinku, optické kabely a koaxiální kabely. V naprosté většině je vybaven konektorem RJ-45. U PLC slouží k propojení mezi více účastníky a to na velmi velké vzdálenosti. Přenosová rychlost je velmi vysoká, až 10Mbit/s.

Radiomodem

Komunikace pomocí Radiomodemů může probíhat obousměrně na vzdálenost až 5 km ve volném prostoru. V průmyslové zástavbě se tato vzdálenost zkracuje. Radiomodemy pracují v různých přenosových pásmech, nejčastěji na frekvencích 433MHz, 868MHz, 1,9GHz nebo 2,4GHz. Může pracovat v režimu poloduplex (halfduplex, tedy režim střídavé obousměrné komunikace, v daném okamžiku může probíhat pouze v jednom směru) nebo fulduplex (komunikace může probíhat oboustranně).

2.4. Komunikační protokoly

Při komunikaci přes sériovou linku, Ethernet nebo USB se používají pro výměnu dat speciální jazyky neboli protokoly. Komunikační protokoly pracují v reálném čase (čas pro odesílání dat z jednoho uzlu do druhého je rozhodující). Bývají imunní vůči typickým poruchám v průmyslovém prostředí. Mezi nejčastěji používané komunikační protokoly patří:

MODBUS

Jedná se o master-slave protokol běžící na aplikační vrstvě modelu ISO/OSI. Vytvořen již v roce 1979 a je používán především při přenosu dat na sériových linkách RS-232, RS-422/485. Definuje strukturu zprávy na úrovni protokolu PDU (Protokol data unit) nezávisle na typu komunikační vrstvy. Pro kódování dat používá tzv. „big-endian“ reprezentaci dat (při posílání datových položek delších než 1 byte je jako první posílán nejvyšší byte)

ProfiBUS

Jeden z nejrozšířenějších protokolů v oblasti průmyslové automatizace. Využívá fyzickou, spojovou a aplikační vrstvu modelu ISO/OSI. Poprvé vytvořen v roce 1987 a v dnešní době je k dispozici ve třech formách (DP-V0, DP-V1 a DP-V2). Využívána hlavně na lince RS-485 a optickém vlákně.

ProfiNET

Neboli průmyslový Ethernet. Otevřený protokol s přenosovou rychlostí až 10Mbit/100Mbit. Možnost připojení decentralních periférií, komunikace v reálném čase. Přenos dat ve standardizovaném stavu (HTML, XML). Možnost integrace a propojení se stávajícími průmyslovými sběrnici. Existuje ve více variantách (DP – nejpoužívanější), FMS a PA. Jako přenosová média používá RS-485, optické vlákno, IEC 1158-2.

3. Zobrazovací panel

3.1. Konstrukce

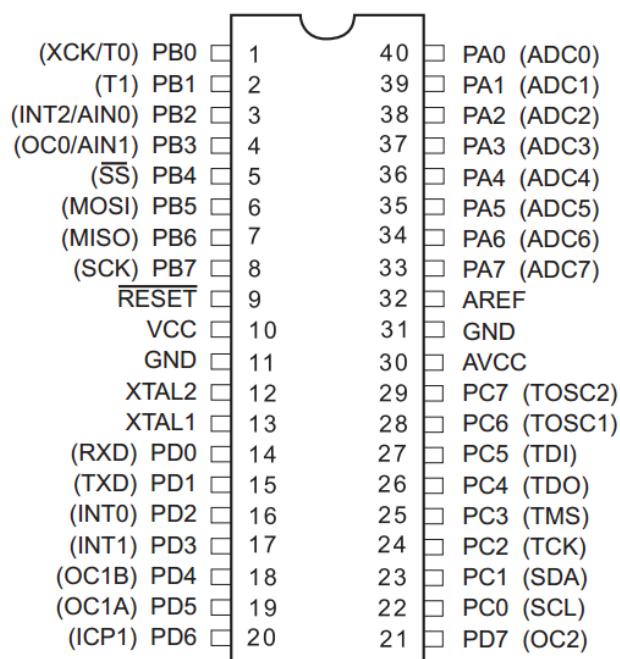
Panel je konstruován do konstrukční krabičky KP18 a jeho předpokládané umístění je v kanceláři technika hlídajícího provoz jednoho či více strojů řízených PLC. Pro zobrazení chyby či nějaké změny stavu slouží LED diody a zároveň výpis chybové informace na zobrazovacím LCD panelu. Zároveň je použit i zvukový signál.

Pro přenos dat mezi PLC a zobrazovacím panelem je použita linka RS-485 s komunikačním protokolem Modbus. Linka je realizována ethernetovým kabelem a konektivita je na obou stranách zajištěna konektorem RJ-45.

3.2. Řídící procesor

Pro řídicí mikroprocesor bylo rozhodováno mezi mikroprocesory rodiny AVR nebo PIC. Nakonec byl jako řídicí procesor vybrán model od AVR ATMEGA162-16PU. Hlavními kritérii pro výběr tohoto mikroprocesoru byla možnost programovat v jazyce C, 28 programovatelných I/O pinů a znalost z předchozího projektu. Na obrázku 3.2.1 jsou vidět vývody pinů použitého mikroprocesoru.

Procesor je použit ve 40 pinovém PDIP pouzdře s operačním napětím 4.5 – 5.5 V. Dále obsahuje 16kB ISP FLASH paměti, 1kB RAM, 512B EEPROM, 2x8b čítač/časovač s PWM jednotkou, 1x16b JTAG rozhraní pro ladění aplikací. Instrukční sada obsahuje celkově 120 instrukcí. Architektura procesoru je typu RISC a lze jej programovat jak paralelně tak sériově přímo v systému.



Obr. 3.2.1 - Vývody mikroprocesoru ATmega16 [10]

3.3. LCD displej

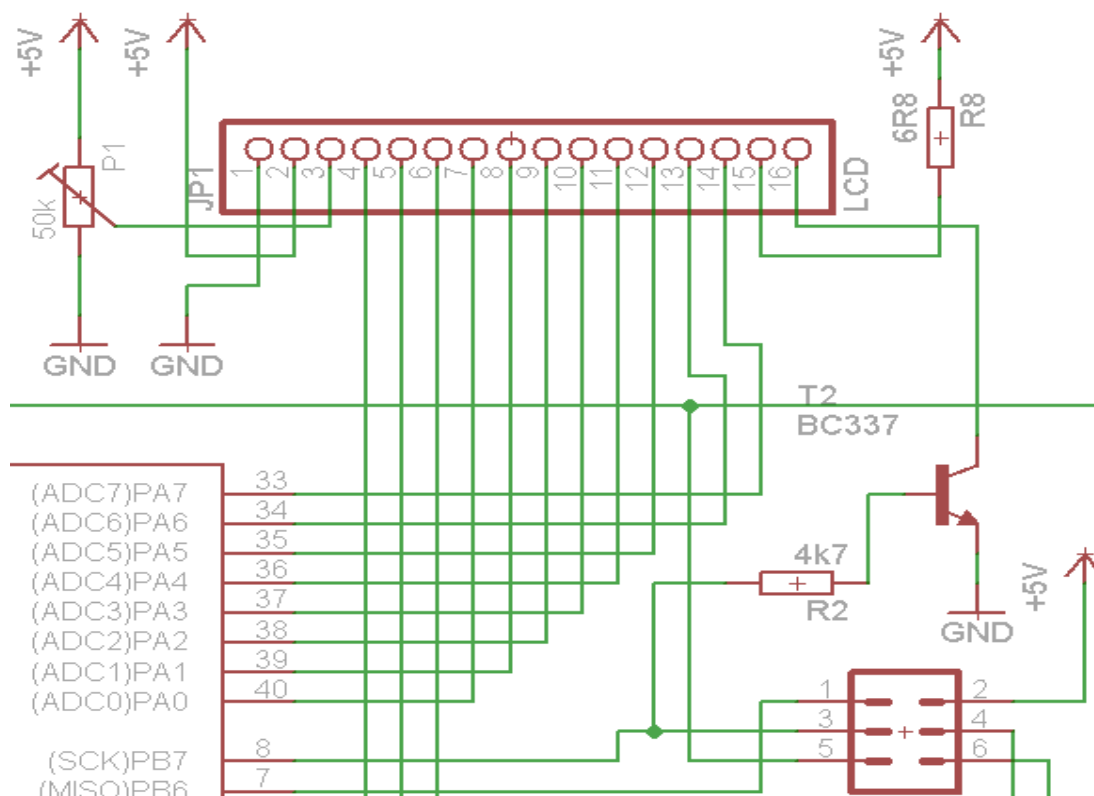
Pro zobrazení informací o vzniklé chybě, nebo neočekávaném stavu slouží LCD displej MC-2004E-SYL od společnosti EVERBOUQUET. K mikroprocesoru je připojen v osmibitovém režimu pro snazší manipulaci a výpis dat na jednotlivé řádky. U displeje je možné ovládat kontrast pomocí trimeru, podsvětlení displeje je ovládáno pomocí mikroprocesoru.

LCD používá řadič HD44780 a obsahuje 4 řádky po 20 znacích. Obvyklé zapojení vývodů displeje je znázorněno na obrázku 3.3.1. Řadič obsahuje běžně používanou znakovou sadu HD44780A00, která obsahuje znaky anglické a řecké abecedy, německé abecedy a některé speciální znaky. U čtyřřádkových displejů začíná první řádek na adrese 0x00h, druhý na 0x40h, třetí 0x14h a poslední řádek na adrese 0x54h.

Vývod	Název	Funkce
1	VSS	GND
2	VCC	Napájení +5V
3	VEE	nastavení kontrastu
4	RS	0 - instrukce, 1 - data
5	R/W	0 - zápis, 1 - čtení
6	E	hodinový vstup
7	DB0	Data 0
8	DB1	Data 1
9	DB2	Data 2
10	DB3	Data 3
11	DB4	Data 4
12	DB5	Data 5
13	DB6	Data 6
14	DB7	Data 7
15	LED+	Anoda podsvětlení
16	LED-	Katoda podsvětlení

Tab. 2.3.1 - Vývody LCD displej

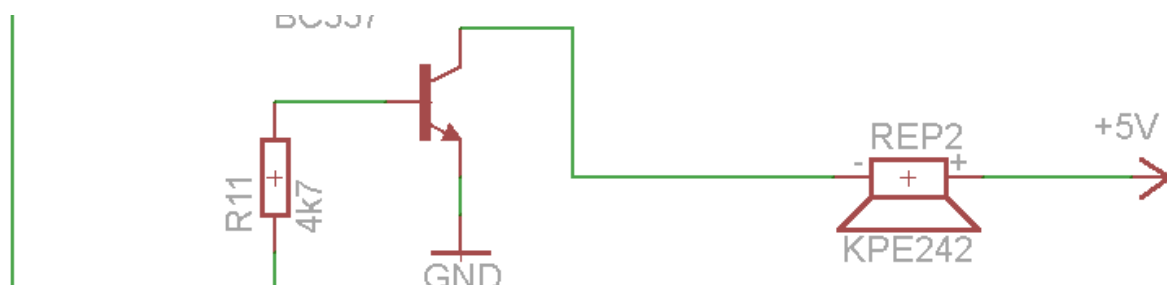
Na obrázku 3.3.2 Je poté zobrazeno připojení displeje k mikroprocesoru, přívod napájení a ovládání kontrastu displeje. Pin 5 – funkce pro čtení a zápis by mohl být trvale připojen na GROUND, protože z displeje se nebudou číst žádné informace, ale z důvodu možného rozšíření je tento pin také připojen k mikroprocesoru.



Obrázek 3.3.2 – Zapojení LDC displeje

3.4. Zvukový signál

Pro generování zvukového signálu byl vybrán piezzo reproduktor KP242. Siréna má stálý tón o frekvenci $4,1 \pm 0,5$ kHz. Reproduktor je připojen přes NPN tranzistor k mikroprocesoru na pin PB3, který je schopný generovat PWM (Pulse Width Modulation). PWM lze velmi dobře použít pro generování různých zvuků a tím docílit rychlého upozornění na chybu. Schéma zapojení je viditelné na obrázku 3.4.1



Obrázek 3.4.1 – Zapojení piezzo reproduktoru

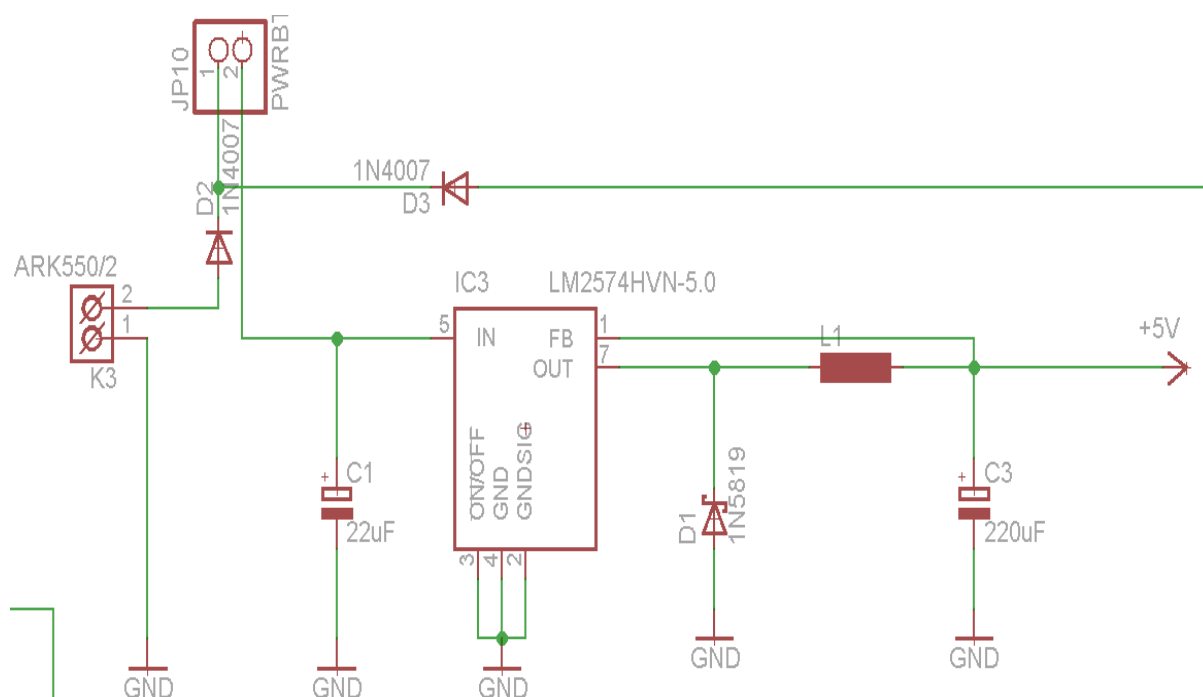
3.5. Programování a ladění

Pro naprogramování a ladění mikroprocesoru je přítomné rozhraní ISP i JTAG, při nahrávání programu bylo použito ISP. S použitím tohoto přístupu k programování je možné programovat mikroprocesor sériově přímo v zařízení.

3.6. Napájení panelu

K napájení panelu je možné použít buď baterii připojenou do svorkovnice K3 nebo rozhraní společně s linkou RS-485. Panel je možné zapnout a vypnout pomocí vypínače připojeného na konektor JP10. K regulaci napětí na požadovaných 5V je použit lineární regulátor napětí LM2574HVN 5.0V, který zároveň produkuje proud o hodnotě 0.5A. Schéma zapojení je zobrazeno na obrázku 3.6.1. Maximální vstupní napětí je až 63V.

K regulátoru bude připojena 9V baterie, která by měla podávat dostatečný výkon k napájení obou zařízení.



Obrázek 3.6.1 – Zapojení lineárního regulátoru napětí

3.7. Přenos signálu

Jak již bylo zmíněno na začátku kapitoly pro přenos signálu mezi PLC a zobrazovacím panelem se používá linka RS-485 s protokolem Modbus. Linka je realizována síťovým kabelem a zakončena standardním ethernetovým konektorem RJ-45.

Převod přijatých a odeslaných dat zajišťuje diferenciální bus transreciever SN75176B. V návrhu byl ještě převodník MAX485, ale z důvodu ceny a dostupnosti dostal přednost již zmiňovaný SN75176B od společnosti Texas Instruments v pouzdře DIP8. Tento transreciever zajišťuje převod dat ze sériové sběrnice na formát TTL používaný mikrokontroléry a je určen pro symetrická vedení. Splňuje ANSI TIA/EIA-422-B a TIA/EIA-485-A a doporučení ITU V. 11 a X. 27.

Dle funkční tabulky zobrazené v 3.7.1 mohou vysílač a přijímač pracovat v režimech aktivní-HIGH nebo aktivní-LOW což umožňuje vhodné řízení směru vysílání.

DRIVER			
INPUT D	ENABLE DE	OUTPUTS	
		A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

RECEIVER		
DIFFERENTIAL INPUTS A-B	ENABLE RE	OUTPUT R
$V_{ID} \geq 0.2\text{ V}$	L	H
$-0.2\text{ V} < V_{ID} < 0.2\text{ V}$	L	?
$V_{ID} \leq -0.2\text{ V}$	L	L
X	H	Z
Open	L	?

Tabulka 3.7.1 – Tabulka režimů transrecieveru SN75176B [11]

Na obrázku 3.7.2 je uvedeno zapojení transrecieveru. Dále je viditelné rozložení pinů v konektoru a přívod signálů do něj. Na piny 3 a 6 jsou přivedeny signály SDA (3) a SCL (6) které se přenáší společně s daty a napětím. U zařízení je tedy možné implementovat dodatečnou funkčnost s využitím těchto sběrnicových linek. V projektu však zůstávají nevyužity.

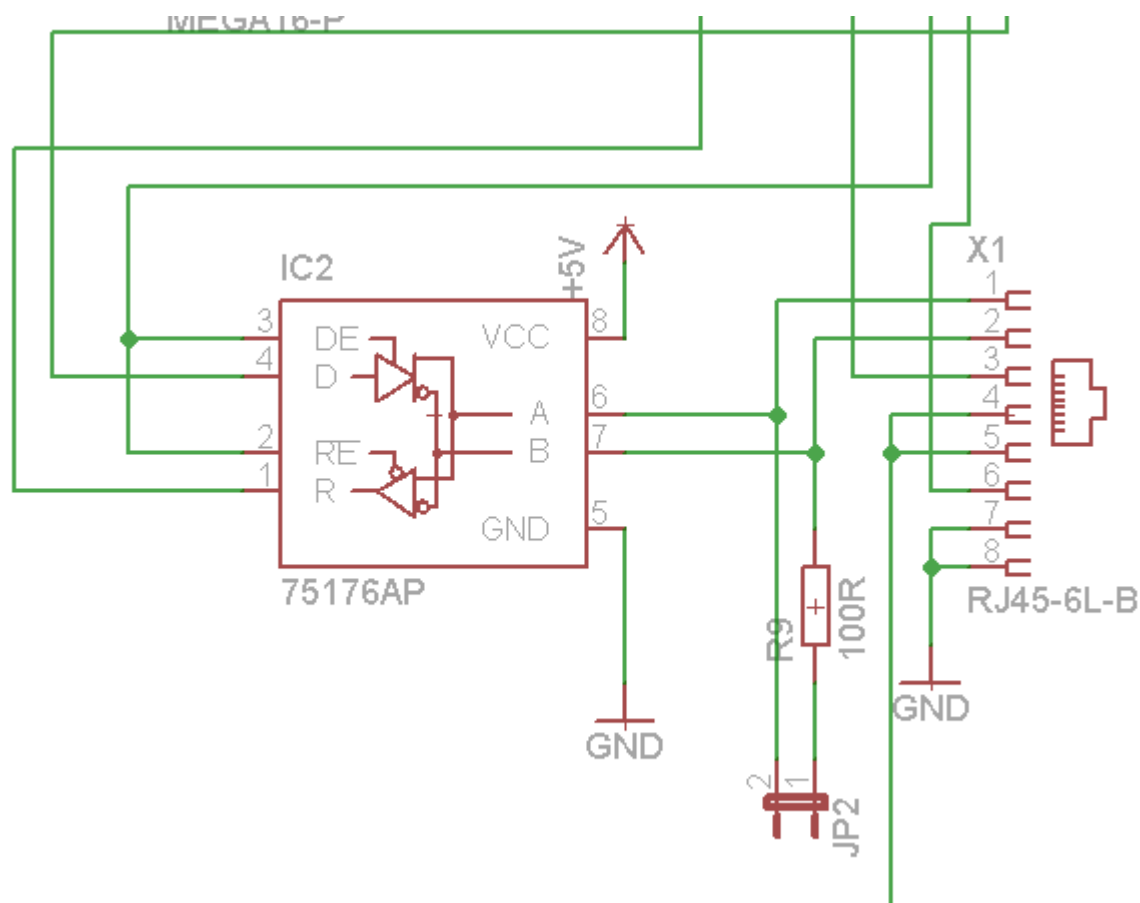
Datové (D, R) piny transrecieveru jsou připojeny na piny RXD (PDO) a TXD (PD1) mikroprocesoru, piny ovládající režim komunikace ($DE, \overline{RE}$) jsou připojeny na pin PB4. Výstupní piny (A, B) jsou připojeny na první respektive druhý pin konektoru. Mezi oběma výstupními vodiči se nachází 120 Ω rezistor, možnost jeho připojení dává jumper JP10.

Spolu s daty je v síťovém kabelu přenášeno i napětí, tak aby nebylo nutné mít v panelu zdroj napájení i když je zde možnost mít připojenu záložní 9V baterii. Jelikož je přenos řešen standardními síťovými komponentami, bylo nutné přenášet napětí dle standardu PoE, aby nedošlo k případnému poškození ostatních zařízení omylem připojených k panelu.

Standard PoE, z anglického Power over Ethernet, definuje jak přenášet napětí po síťovém kabelu. Pro přenos se využívá dvou režimů. Režim ‚A‘, kdy se napájení přenáší mezi dvojicí aktivní (datových) párů vodičů (1,2 a 3,6). A režim ‚B‘, tedy napájení po volných, nevyužitých párech (4,5 a 7,8).

Pro PoE existuje několik norem. Nejnovější norma IEE802.3af definuje aktivní PoE, rozsah přenášeného napětí je 44 – 57 V a maximální odběr na jedno zařízení 350 mA. Druhou používanou normou je takzvané pasivní PoE, které je nekompatibilní s normou IEE802.3af. Tato norma určuje pro přenos kladného napětí 4 a 5 vodič a záporného napětí 7 a 8 vodič.

V návrhu desky se počítá s použitím pasivního PoE a režimu B pro přenos napětí do panelu. Pro ochranu proti přepólování slouží polovodičová dioda.



Obrázek 3.7.2 – Zapojení transrecieveru a konektoru RJ-45

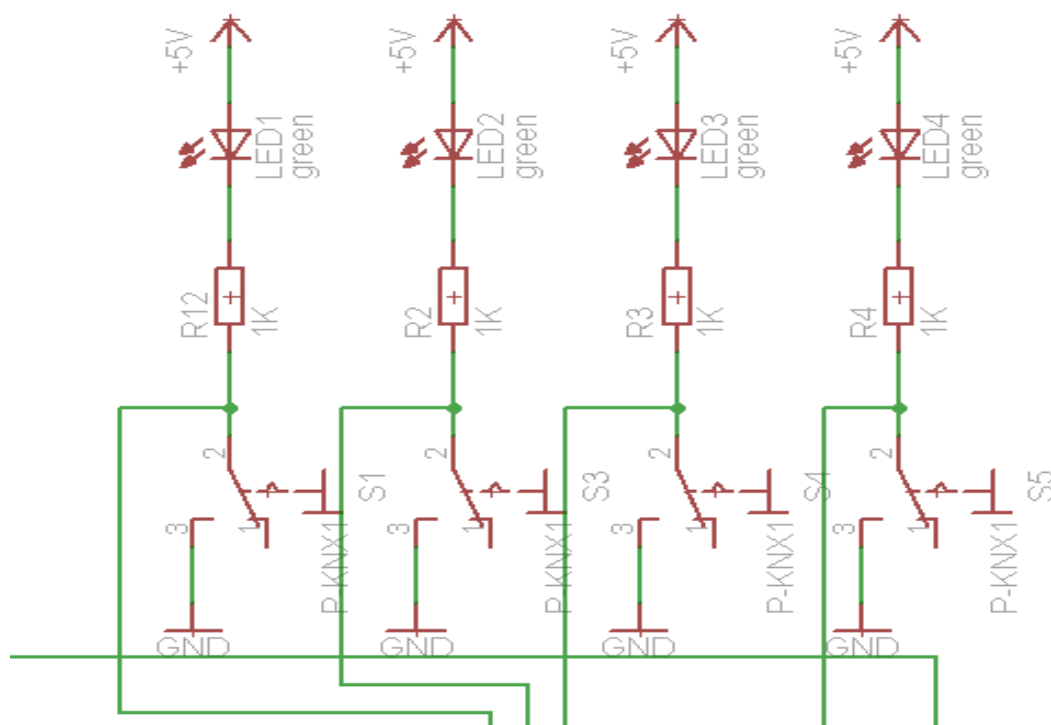
4. Náhrada za PLC

Při tvorbě této bakalářské práce nebyl k dispozici žádný programovatelný automat, proto bylo přistoupeno k návrhu a sestrojení desky simulující PLC. Deska nemá za účel nahrazovat popřípadě kopírovat chování nebo vlastnosti nějakého automatu, ale pouze pomocí zvoleného protokolu a přenosové linky zasílá informace o změně stavu (v tomto případě přepnutí vypínače).

Při návrhu simulátoru se celkově vycházelo z návrhu zobrazovacího panelu. Proto je většina jeho částí stejná (součástky pro přenos signálu, schéma napájení, periferie k programování a ladění). Výraznější změnou je zapojení vypínačů a LED indikátorů, jejíž schéma je uvedeno na obrázku 4.1.1.

Pro řízení byl vybrán stejný typ mikroprocesoru jako na desce, i když by zde mohl být použit menší například 28 pinový procesor. Množství nevyužitých I/O pinů, bylo vyvedeno na lišty tak, aby se mohli použít například pro sběr informací z jiných periférií (což je pro účely projektu nepotřebné). Proto lze tento výrobek použít i v budoucnu pro nějaké další projekty, nebo jako menší vývojový kit.

Na desku je tedy plánováno umístit 4 LED diody a 4 spínače, každá LED dioda je umístěna na jednom pinu mikroprocesoru společně se spínačem, aby bylo při vývoji zřetelně vidět, zda je spínač sepnutý nebo ne.



Obrázek 4.1.1 - Zapojení LED diod a spínačů

Jako přepětová ochrana, proti neočekávanému stavu napětí na vedení zde slouží trubičková pojistka. Schéma zapojení celého simulátoru je uvedeno v příloze A.

5. Deska plošných spojů a konstrukční řešení

Obě desky mají rozměry 80x60 mm a jsou na nich umístěny všechny potřebné součástky. Mikroprocesor, regulátor ani transceiver nejsou přímo uchyceny na DPS, ale každá je umístěna do DIP modulu pro snazší výměnu a údržbu.

Desky byly z časových důvodů zadány k výrobě externí firmě, která zvládla výrobu na profesionální úrovni. Pájení bylo prováděno v domácí dílně s použitím mikropájký.

Konstrukční krabičky jsou pro účely předvedení zhotoveného výrobku opracovány ručně v domácích podmínkách, z toho důvodu samozřejmě nedosahují požadovaného profesionálního vzhledu. V případě nasazení do výroby a použití například v továrnách nebo kancelářích je nutné předat k návrhu specialistovy.

Deska pro panel je umístěna v krabičce o rozměrech 125x120x57 mm. Na vrchní straně se nachází LCD displej pro zobrazení informací, 4 LED diody pro signalizaci chybového stavu a jeden spínač. Na boku se pak ve spodní polovině výrobu nachází RJ-45 konektor pro připojení přenosové linky.

V původním návrhu byly LED diody i spínače připevněny na samotné desce, ale při sestrojování bylo zjištěno, že umístění součástek takto blízko u sebe by v celkovém vzhledu budilo špatný dojem, proto byly všechny viditelné součástky přesunuty na krabičku a připevněny pomocí vhodných držáků. S deskou jsou poté spojeny drátovými propojkami. Na desce jsou potřebné piny vyvedeny na lišty tak, aby se dali lehce připojit pomocí standardních spojek. Rozteč vývodů na liště je 2,54 mm.

Simulátor PLC je umístěn do konstrukční dvoudílné krabičky KP58 o rozměrech 124x78x42mm. Na vrchní straně se nachází 5 spínačů a 4 LED diody. Jeden spínač slouží k zapnutí a vypnutí simulátoru a další 4 spínače slouží k simulaci nějakého chybového stavu. Při sepnutí spínače se rozsvítí příslušná LED dioda pro signalizaci a kontrolu, zda byl spínač sepnut.

Stejně jako u panelu jsou viditelné součástky umístěny na krabičce. Jedinou výjimku tvoří RJ-45 konektor, který je umístěn přímo na desce. Z toho důvodu bylo nutné umístit desku co nejbližší některému z obvodových stran krabičky, čímž vzniknul dostatečný prostor pro umístění baterie dovnitř obalu.

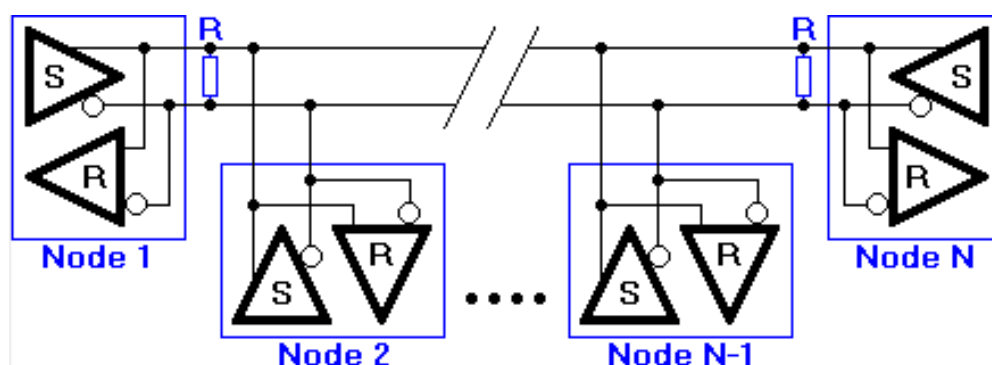
Výsledné podoby výrobků jsou uvedeny v příloze C.

6. Bližší pohled na linku RS-485

Jak již bylo zmíněno pro přenos signálu je použita linka RS-485, a to pro její dobré vlastnosti jako je například vyšší odolnost proti rušení. Linka RS-485 nespecifikuje typy zapojení konektorů nebo přenosový protokol, specifikuje pouze použití kabeláže a elektrické vlastnosti.

RS485 nevztahuje při přenosu signály vůči signálové zemi, ale signály jsou plovoucí, tedy že přenášená data jsou odesílána po trase sig+ i sig-. Příjímač poté porovnává rozdíly mezi oběma signály. Tento způsob ve většině případů eliminuje vznik zemních smyček.

Přenosová rychlost je přímo úměrná vzdálenosti přenosu. Například při vzdálenosti 1200m je přenosová rychlost 100kbit/s, při vzdálenosti 120m až 1Mbit/s a rychlostí 10Mbit/s je linka schopna přenášet na vzdálenost do 15 m.



Obr. 6.1 – Zapojení zařízení na lince RS-485 [8]

Na obrázku 6.1 je znázorněno zapojení více přijímačů/vysílačů na lince RS485. Pro delší vzdálenosti a větší rychlosti je nutné na koncích přidat ukončovací rezistory tzv. rezistory a to o hodnotě 100 – 120 Ω . Tyto rezistory slouží k tomu, aby eliminovaly všechny případné odrazy, které se mohou při přenosu vyskytnout.

Linka RS-485 je standardně použitelná tak, že obsahuje jedno zařízení typu master a několik zařízení typu slave. Jednotky jsou definovány jako slave nebo master na nejvyšší úrovni protokolu. Slave jednotky čekají na příjem požadavků od master zařízení a poté vykonají požadavek. Při tomto rozvržení je šířka přenášeného pásma využita až na 100 procent.

Jiným způsobem je, že kterékoliv ze zařízení může kdykoliv započít vysílání dat, v tom případě je ale nutné implementovat detekci a korekci dat způsobenou například započítáním vysílání dvou zařízení najednou. Rapidně se také sníží efektivita pásma a rychlost přenosu a to až na 37 procent.

7. Firmware výroby

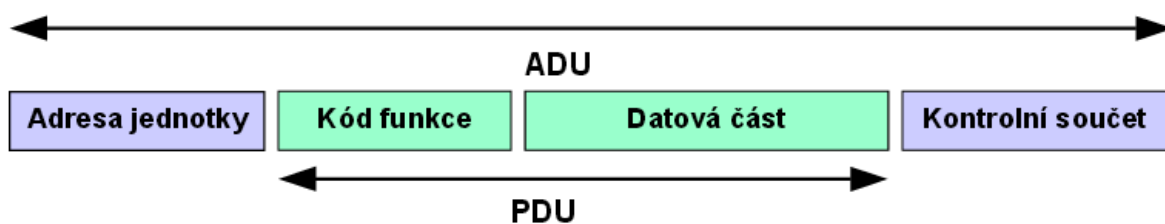
Firmware byl do obou výrobků nahrán přes standardní 6pinové rozhraní ISP pomocí programátoru USBasp [3] a program Atmel Studio 6.1. Pro debugging programu bylo připraveno také rozhraní JTAG, ale při programování nebylo potřeba jej využít.

U obou mikroprocesorů byly pomocí programu AVRDUDE nastaveny hodnoty Fuse bits na LOW Bits – 0xFF a HIGH Bits 0x89. Tedy povolení programování SPI a JTAG a externí krystal v rozmezí 8-12 MHz.

7.1. Implementace protokolu Modbus

Při zkoumání protokolu MODBUS a možnostech jeho implementace jsem narazil na několik open source knihoven, některé byly pro možnosti projektu příliš náročné a rozsáhlé a některé zase nesplňovali správné požadavky na implementaci.

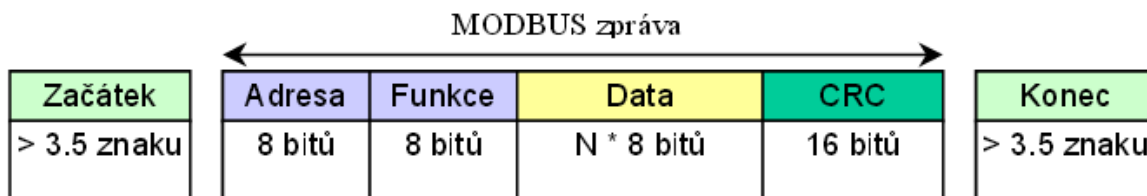
Na obrázku 7.1.1 je formát aplikační zprávy po sériové lince. Zpráva obsahuje Modbus PDU, adresu jednotky a kontrolní součet.



Obrázek 7.1.1 – Formát zprávy MODBUS RTU [4]

Protokol definuje dva vysílací režimy a to RTU, který je povinný pro každé zařízení a ASCII, jehož implementace je dobrovolná. Všechny zařízení na lince musí být ve stejném vysílacím režimu. Tento režim definuje především, v jakém formátu jsou data vysílána.

V režimu RTU je formát bytu následovný: 1 start bit, 8 datových bitů, 1 bit parita, 1 stop bit. Jednotky musí podporovat sudou paritu, a pokud není použita, je nahrazena druhým stop bitem. Formát rámce je znázorněn na obrázku 7.1.2. [4]



Obrázek 7.1.2 – Formát rámce MODBUS RTU [4]

Typický Modbus packet má následující tvar:

Režim Master: 02 03 00 08 00 01 05 FB

kde první dvojice značí adresu slave jednotky, druhá dvojice značí požadavek '03 - Čti uchovávací registry (Read Holding Registers)'. Další dvě dvojice reprezentují adresu slave jednotky ze které se bude číst jedno slovo (00 01) a poslední dvojice je datový součet zprávy.

V projektu je zadáno, že výsledný výrobek může být rozmístěn ve více kusech na různých místech, a proto je zde zvolena komunikace typu unicast. V případě MODBUS protokolu to znamená, že pro adresu je použita hodnota 0x00.

Protokol MODBUS definuje několik funkcí pro práci s daty a registry které definuje. Každá funkce po provedení odesílá odpověď, pokud proběhla v pořádku nebo chybu pokud tomu tak nebylo. Pro náš projekt je důležitá a použitá funkce 6 (0x06) – Zapiš jeden registr (Write Single Register), která slouží k zapsání hodnoty jednoho registru.

Formát funkce číslo 6 je patrný z obrázku 7.1.3. V případě, že by se nejednalo o broadcast zprávu, byla by odpověď na požadavek kopie přijaté zprávy, odeslaná po vykonání funkce.

Požadavek

Kód funkce	1 byte	0x06
Adresa registru	2 byty	0x0000 až 0xFFFF
Hodnota registru	2 byte	0x0000 až 0xFFFF

Obrázek 7.1.3 – Formát zprávy 0x06 – Zapiš jeden Registr [4]

7.2. Obslužný program vysílače

Na straně vysílače, teda simulátoru PLC pro projekt, bylo nutné implementovat pouze vysílání MODBUS protokolu. Obslužný program tedy obsahuje inicializační funkci, která nastaví pinům příslušejícím ke spínačům, tedy piny PD4 – PD7, hodnotu LOW.

A dále nastaví hodnotu pinu PB4, který je připojen k transrecieveru, na hodnotu HIGH. Tím se nastaví transreciever na funkci odesílání a tato hodnota se již v průběhu běhu programu nemění, protože není nutné, aby simulátor přijímal data.

Následně je volána funkce pro nastavení sériové komunikace a nastavení registrů potřebných pro odesílání dat.

```
UCSRB = (1 << RXEN) | (1 << TXEN) | (1 << TXCIE);
```

- RXEN – povolení příjmu
- TXEN – povolení odesílání
- TXCIE – povolení přerušení od vysílače

```
UCSRC = (1 << UCSZ0) | (1 << UCSZ1);
```

- UCSZ0, UCSZ1 – určují počet datových bitů, v programu 8b

```
UBRRH = (BAUD_PRESCALE >> 8); UBRL = BAUD_PRESCALE;
```

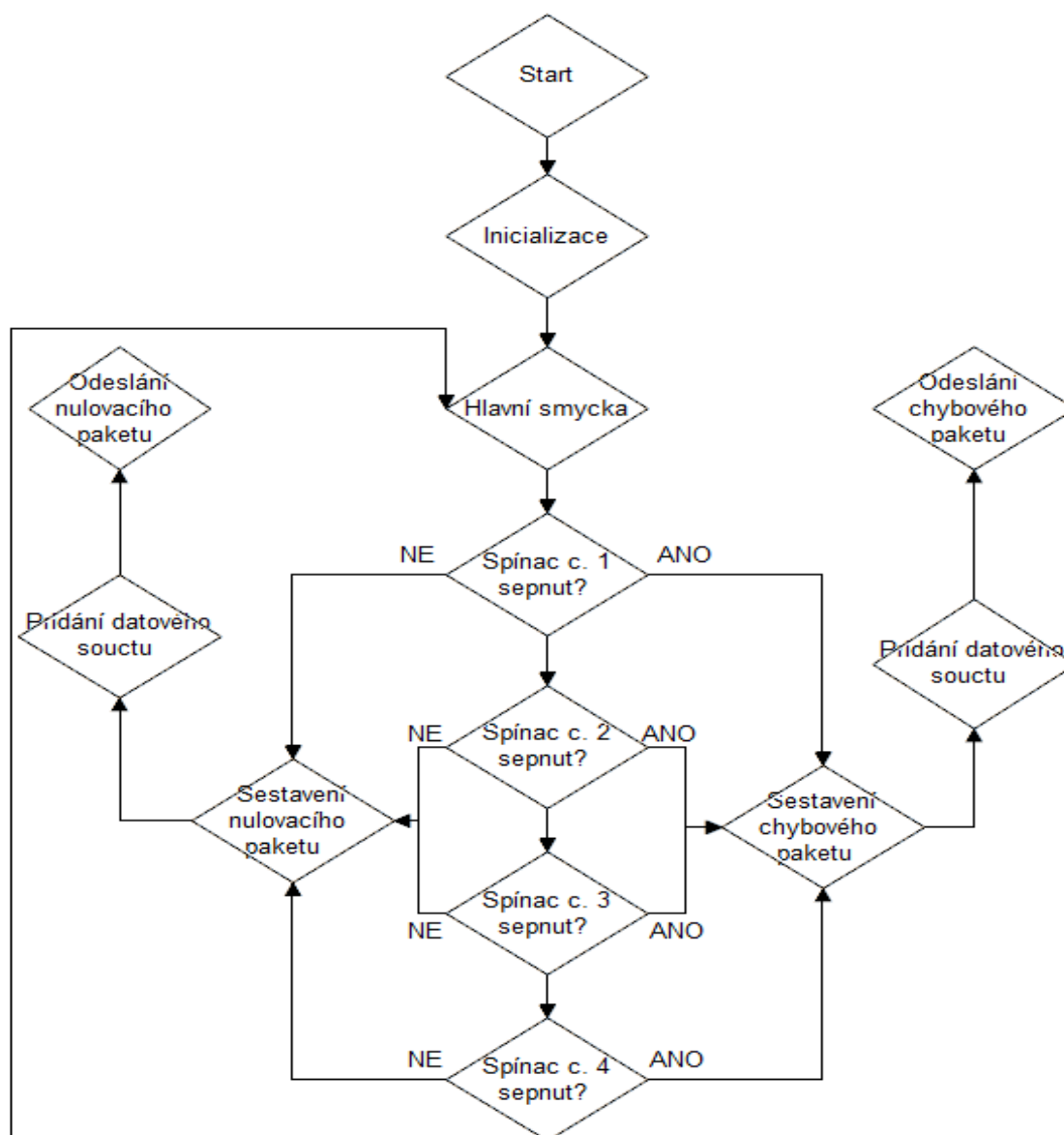
- UBRRH, UBRL – tyto registry slouží k nastavení přenosové rychlosti

```
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)
```

- Přenosová rychlost je vypočítána pomocí vzorce:

$$f_{BR} = \frac{f_0}{16 * (UBRR + 1)}$$

Chod hlavního programu popisuje obrázek 7.2.1. Program čeká na stisk spínače reprezentující chybu a poté volá funkci pro sestavení a odeslání MODBUS paketu. Těsně před odesláním paketu je volána funkce pro sestavení kontrolního součtu paketu.



Obrázek 7.2.1 – Diagram vysílače

Ke kontrolnímu součtu se používá 16bitové CRC (Cyklický redundantní součet) s generujícím polynommem $x^{16} + x^{15} + x^2 + 1$. Počáteční hodnota CRC je nastavena na hodnotu 0xFFFF a generující polynom je 0xA001.

Pakety obsahující chybový stav se odesílají každých 4-5 vteřin, z důvodu možné ztráty nebo poškození paketu. Pakety obsahující informaci o tom, že je vše v pořádku, jsou odesílány jednou za 25 vteřin. Odesílání paketu probíhá postupně po dvou bytech. Odeslání prvního bajtu je voláno z funkce pro sestavení paketu, zbytek bytů je odeslán postupně při přerušení TxC, dokud není odeslán celý paket.

7.3. Obslužný program panelu

Na straně panelu je MODBUS protokol pouze přijímán, proto bylo nutné naimplementovat pouze příjem paketů. Dle specifikace MODBUS protokolu odesílá slave zařízení odpověď na master požadavek. A to buď požadovaná data v případě „read“ požadavku anebo kopii v případě „write“ požadavku. Pokud se při zpracování požadavku vyskytne chyba, odesílá slave zařízení informaci o výskytu chyby a kód chyby.

Typický algoritmus zpracování paketu je znázorněn na obrázku 7.3.1. Pokud master odešle požadavek typu broadcast slave jednotky neodesílají odpověď ani zápornou odpověď, aby nedošlo k zahlcení přenosové linky. Proto se zprávy typu broadcast používají jen v případě „write“ požadavku.

Na začátku obslužného programu v mikroprocesoru panelu, dochází stejně jako v případě simulátoru, k inicializaci sériové komunikace a nastavení registrů pro MODBUS protokolu. Dále proběhne nastavení pinů PD4-PD7, které ovládají LED diody na HIGH. A nakonec před spuštěním hlavní smyčky proběhne inicializace LCD displeje.

V hlavní smyčce programu probíhá stále kontrola holding registru, který udržuje informace o chybách. Pokud je do registru uložena chyba, rozsvítí se příslušná LED dioda a zpráva o chybě se vypíše na LCD display. Ve stávajícím řešení je zahrnut pouze jeden kód chyby, v případném rozšíření je zde možnost zapsat více chyb a na displej vypisovat i chybovou hlášku.

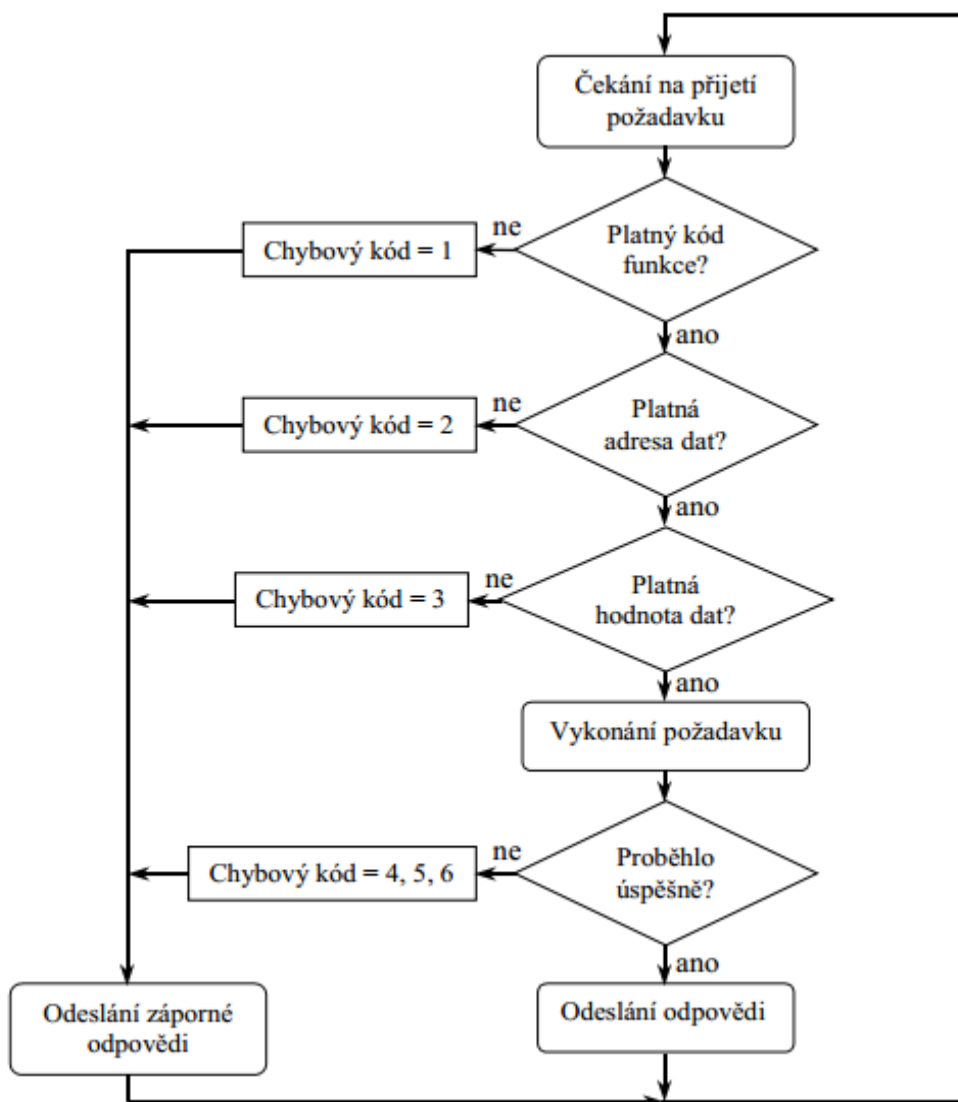
Zároveň s rozsvícením LED diody a zapsáním informace na LCD displej zazní zvuk, který upozorní obsluhu na vzniklou situaci. Zvuk zní po celou dobu kdy je chyba zaznamenávána panelem. Zvuk je generován piezzo reproduktorem, který je připojen na pin PB3 mikroprocesoru. Ke generování zvuku je použita, jak již bylo uvedeno, piezzo siréna se stálým tónem.

Při příjmu znaku dojde k přerušení, kde dojde ke startu časovače a nastavení jeho hodnoty a uložení znaku na do proměnné. Pokud se jedná o první znak zprávy, je proměnná vynulována a další přijaté znaky se ukládají za přijatý první znak. To jestli se jedná o první znak, zjišťuje nastavený časovač. Dle specifikace musí být mezi zprávami mezera delší než 3,5 znaku.

Pro rychlost 9600 baudů v 11 bitovém paketu je přeneseno 872 znaků za sekundu. Pro jeden znak je to $\frac{1000(ms)}{872 (znaků)} = 1,146 ms/znak$. Mezera mezi jednotlivými znaky ve zprávě nesmí být delší než 1,5 znaku tedy $1,5 * 1,146 = 1,719 ms$. A mezera mezi jednotlivými zprávami musí být delší než 3,5 znaku tedy $3,5 * 1,146 = 4,011 ms$. Na začátku programu je definován interní časovač, který sepne jednou

za 10 μ s. Od toho je odvozena hodnota pro ukončení zprávy 179 tiků časovače a pro ukončení rámce a přesunutí do obslužné funkce pro uložení hodnot do registru 401 tiků.

V obslužné funkci se nejprve zkontroluje, zda se jedná o broadcast zprávu, která je jediná podporovaná, zda souhlasí kontrolní součet zprávy, tedy je-li zpráva úplná a bez chyb a poté se zkontroluje, zda se jedná o zprávu s funkcí 6, tedy o zprávu „Zapiš hodnotu do uchovávacího registru“. Pokud je vše splněno, je daná hodnota zapsána do registru a program čeká na přijetí další zprávy.



Obrázek 7.3.1 – Zpracování MODBUS požadavku [4]

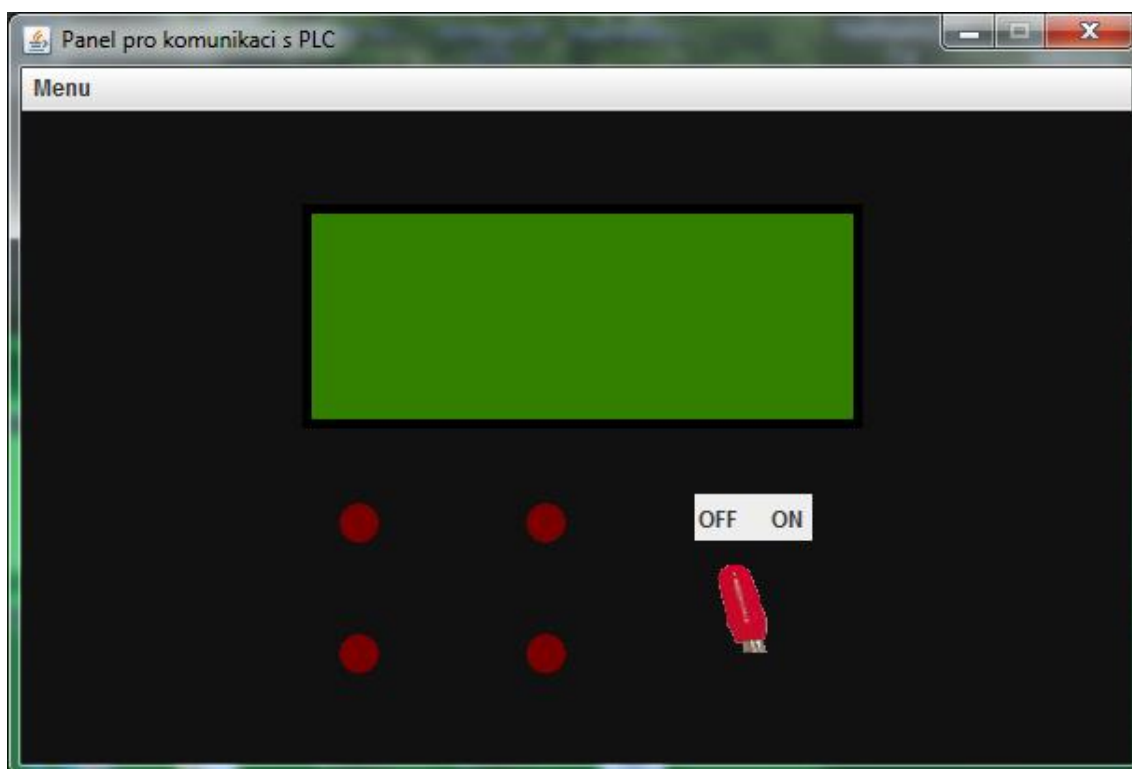
8. Program pro PC

Pro program do počítače jsem zvolil programovací jazyk Java, z důvodu znalostí toho to jazyka z jiných projektů. Jako GUI aplikace jsem zvolil JFC/Swing. Vzhled aplikace je podobný jako fyzické zařízení, obsahuje tedy LCD display pro výpis informací, čtyři LED diody pro zobrazení chybového stavu a vypínač.

8.1. GUI aplikace

LCD display je reprezentován čtveřicí komponent typu `JLabel`. Ty jsou poté umístěny v komponentě `JPanel`. Pozadí panelu je nastaveno na barvu `#338000`, která je podobná barvě monochromatické displeje. Tato barva značí vypnutý displej. Po zapnutí displeje je barva nastavena na hodnotu `#53A000`.

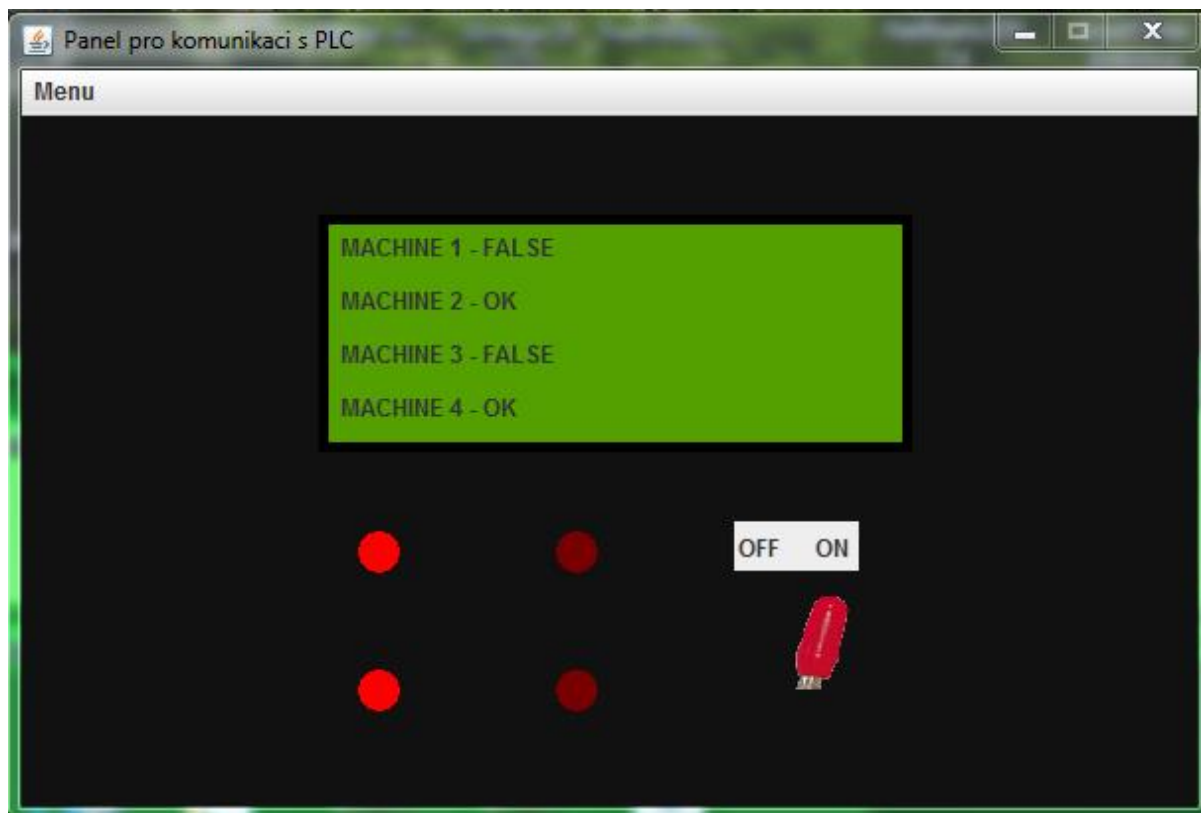
LED diody jsou znázorněny pomocí komponenty `JPanel`, na kterou jsou pomocí funkcí `drawOval(int, int, int, int)` a `fillOval(int, int, int, int)` vykresleny čtyři kruhy. Vypnutá LED dioda má nastavenou barvu `#780000`, a sepnutá dioda `#FF0000`. Pro zapnutí a vypnutí vnitřní logiky aplikace slouží komponenta `JPanel`, ve které je vykreslován obrázek spínače. Ten se mění tak aby byla patrná změna stavu a bylo viditelné, kdy je panel zapnutý a kdy je vypnutý.



Obrázek 8.1.1. – Panel ve stavu ‚vypnuto‘

Po zapnutí se inicializuje display s informacemi o stavu jednotlivých strojů a spustí se naslouchání na zvoleném PC portu.

Na obrázku 8.1.1 je vidět vzhled aplikace ve stavu vypnuto a na obrázku 8.1.2 ve stavu zapnuto se zobrazenou jednou chybou na stroji 1. Z obrázků je patrný rozdíl indikace chyby a stavu kdy je vše vypnuto.

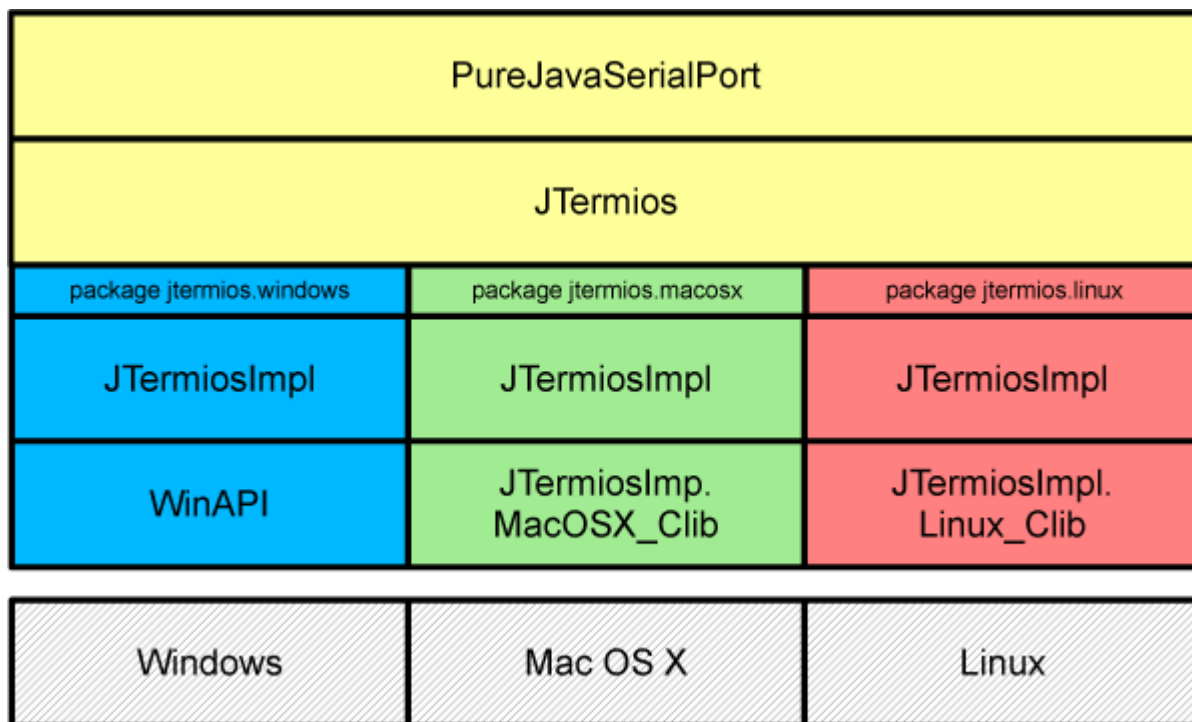


Obrázek 8.1.2 – Panel ve stavu zapnuto a zobrazující chyby

8.2. Připojení k USB portu

Před spuštěním panelu je nutné inicializovat spojení s USB portem, kam jsou přenášena data z vysílače (PLC, simulátor). Jelikož jazyk Java neumožňuje přímé spojení s USB, byla zvolena nativní knihovna pro připojení. Na internetu jsem narazil na několik takovýchto knihoven, například `JavaCommunication Api` neboli `javax.comm`. Tato knihovna bohužel potřebuje k fungování ještě další knihovnu, která je závislá na platformě, což velmi stěžuje přenositelnost.

Nakonec byla zvolena knihovna `PureJavaComm`, která využívá knihovnu `JTermios` a `JNA library`, zajišťující přístup k portům na nižší úrovni. Knihovna `PureJavaComm` implementuje stejné funkce a rozhraní jako zmíněná `JavaCommunication Api`, takže přechod mezi těmito knihovnami je velmi jednoduchý. Na obrázku 8.2.1 je znázorněn architekturu přístupu pro každou platformu.



Obrázek 8.2.1 – Architektura knihovny PureJavaComm [6]

Funkce pro komunikaci a připojení k portu jsou obsaženy ve třídě `USBCOM` implementující `SerialPortEventListener`. V konstruktoru této třídy se nejdříve načtou všechny porty dostupné v hostitelském PC. Poté se tento list projde a pokusí se připojit ke zvolenému portu, pokud se to nepodaří, je vyhozena výjimka a uživatel se musí pokusit připojit znovu ke stejnému nebo k jinému portu.

V případě, že se připojení podaří je nastavena rychlost přenosu, počet datových bytů, typ parity a počet stop bitů. Dále je nastaven `EventListener` pro poslech příchozích paketů. V případě příchodu paketu je zavolána funkce `serialEvent(SerialPortEvent event)` a začnou se načítat data.

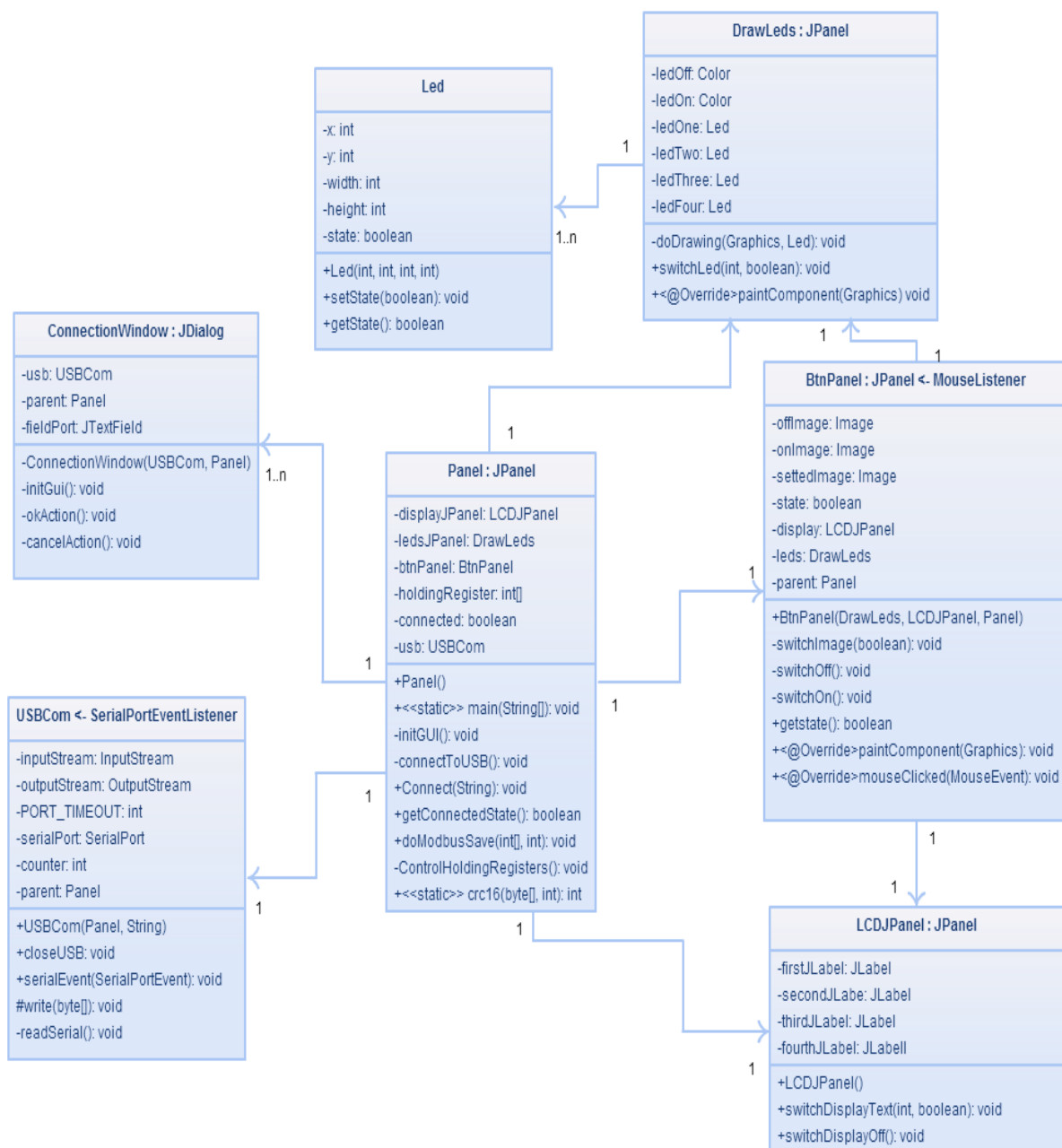
8.3. Vnitřní logika

Po příjmu je u paketu spočítán kontrolní součet stejně jako v případě panelu a vysílače. Kontrolní součet je porovnán s CRC příchozím s paketem. Pokud souhlasí, je zkontrolováno číslo funkce a ID slave zařízení. Pokud je vše v pořádku proběhne uložení přijatých dat do `holdingRegistru`, který uchovává informace o stavu chyb.

Po každém přijetí a uložení dat je volána funkce pro kontrolu `holdingRegistru`. Funkce kontroluje pouze první čtyři hodnoty v tomto registru, protože tam jsou uloženy informace o stavu

chyby. Pokud je na místě uložena hodnota 0x02, indikuje to, že byla přijata zpráva o chybě a je nutné na to upozornit obsluhu panelu.

Na obrázku 8.3.1 je znázorněn diagram tříd pro aplikaci. Modifikátory přístupu + (public), - (private) a # (protected). Znak značí : (dědičnost) a <- (rozhraní).



Obrázek 8.3.1 – Diagram tříd softwarové aplikace

8.4. RS485-USB převodník

Pro převod dat přijatých od PLC je nutný USB-RS485 převodník, který po připojení vytváří virtuální COM port pro komunikaci s obslužným softwarem v PC. Při vývoji byl použit převodník USBto485_v1_1 od firmy CUTTER Systems. K počítači je připojen pomocí USB konektoru typu B. Na druhé straně pak obsahuje vývody pro připojení datových linek A a B, napájení +5V a GND.

Tento převodník byl zapůjčený od vedoucího práce Ing. Vít Tříška. Bohužel k němu nebyla přidána dokumentace a při hledání na internetu nebyly nalezeny žádné relevantní výsledky. Nicméně převodník fungoval dle očekávání a data byla přijímána v PC, tak jak byla odeslána z vysílače.

Ovšem problém nastal při přenosu dat na delší vzdálenosti než 20 metrů. Zde se již objevovalo větší množství poškozených paketů. Dle informací z jiných levnější převodníků je tato funkčnost také očekávána, proto je nutné při nasazení do provozu použít některý z výkonnějších převodníků aby nedocházelo ke ztrátám informací.

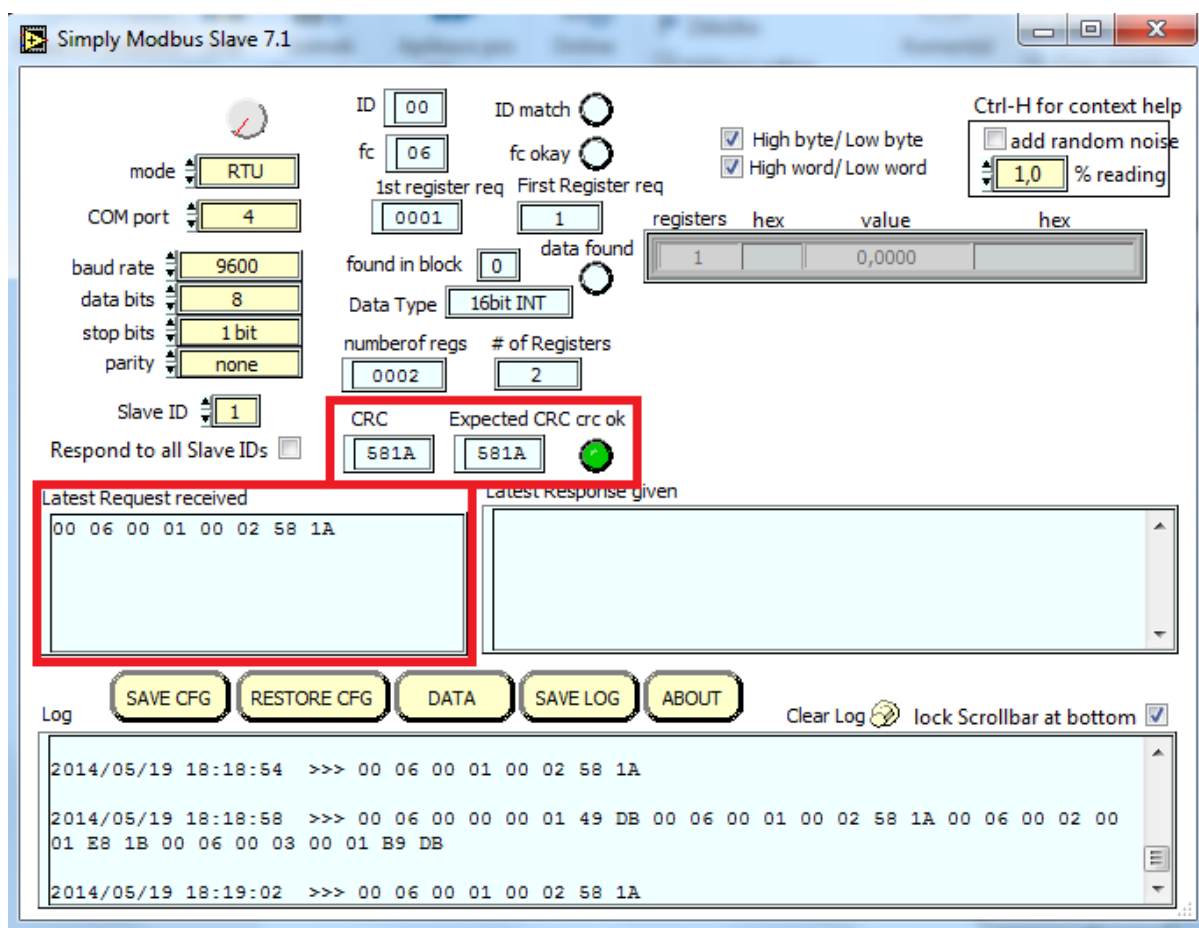
9. Testování

Jak již bylo zmíněno v průběhu textu, během vývoje nebylo k dispozici žádné PLC. Domluvené testování ve firmě OEZ s.r.o. nakonec nebylo možné kvůli nedostatku času obsluhy PLC systému. I když bylo možné ke konci práce zapůjčit jeden PLC systém, tak z důvodu nedostatku znalostí s programováním těchto systémů, bylo od zapůjčení ustoupeno.

9.1. Testování vysílače

Systém byl tedy testován pouze s použitím prostředků dostupných z internetu. S využitím programu *SimplyModbusSlave*, který umožňuje odesílat MODBUS pakety, a převodníku používaného při vývoji byla zjištěna dostatečná úroveň odesílaných paketů z vysílače.

Na obrázku 9.1.1 je zvýrazněn přijatý formát zprávy, který odpovídá formátu zprávy předpokládaného při použití funkce číslo 6. Z obrázku je dále patrné, že vysílač správně počítá datový součet a že nedochází ke ztrátám paketů.



Obrázek 9.1.1 – Výstup z programu *SimplyModbusSlave*

9.2. Testování panelu

Panel byl testován současně se softwarovou aplikací. A to nejdříve aplikací `SimplyModbusMaster`, která je obdobou programu použitého při testování vysílače. A také vysílačem samotným. Oba testované způsoby dávali stejné výsledky, čímž byla v rámci testování dostupnými prostředky ověřena správná funkčnost panelu. Softwarová verze dávala ve stejném čase a tudíž se stejnými testovacími hodnotami totožné výsledky jako samotné hardwarové zařízení.

Výše uvedené testy probíhali na poměrně malou vzdálenost, délka kabelu necelé dva metry. Proto byl proveden ještě jeden test na vzdálenost 100 metrů. Softwarová aplikace již nedávala tak přesné výsledky, z důvodu poškození nebo ztráty paketu. Tato možnost byla očekávána z důvodu použitého převodníku.

Hardwarová verze na tuto vzdálenost dávala stejné výsledky jako na kratší vzdálenost, i když s větším zpožděním.

10. Závěr

Cílem této práce bylo seznámit se s možnostmi komunikace se PLC systémy, navrhnout a vyrobit zařízení, které s nimi bude schopno komunikovat a bude tak demonstrovat vybranou možnost komunikace. Zvolenou možností komunikace byla linka RS485 s protokolem MODBUS pro přenos dat po této lince.

Během vývoje bylo nutné navrhnout náhradu za PLC systém tak aby bylo možné výsledný výrobek otestovat. Výsledný výrobek za PLC systém částečně nahrazuje PLC ale jen pro účely testování. Po otestování je možné vyjmout vyrobené zařízení vyjmout z krabičky a použít ho například jako vývojový kit pro komunikaci skrz linku RS485. Proto jsou nevyužité piny mikroprocesoru vyvedeny na samostatnou lištu pro možnost připojení externích zařízení.

Samotný signalizační panel je schopný přijímat a dekódovat přijatá data dle protokolu MODBUS. Tento protokol je velmi rozšířený v komunikaci s PLC systémy, proto je možné panel použít s většinou těchto systémů a záleží pak na vnitřní implementaci PLC aby posílal data, která je panel schopný dekódovat a zobrazit.

Dalším cílem práce bylo vytvořit softwarovou verzi signalizačního panelu s využitím převodníku z RS485 do USB. Výsledný program je schopný přijímat a dekódovat stejná data jako hardwarové zařízení, a také dává stejné výsledky jako zmiňované zařízení.

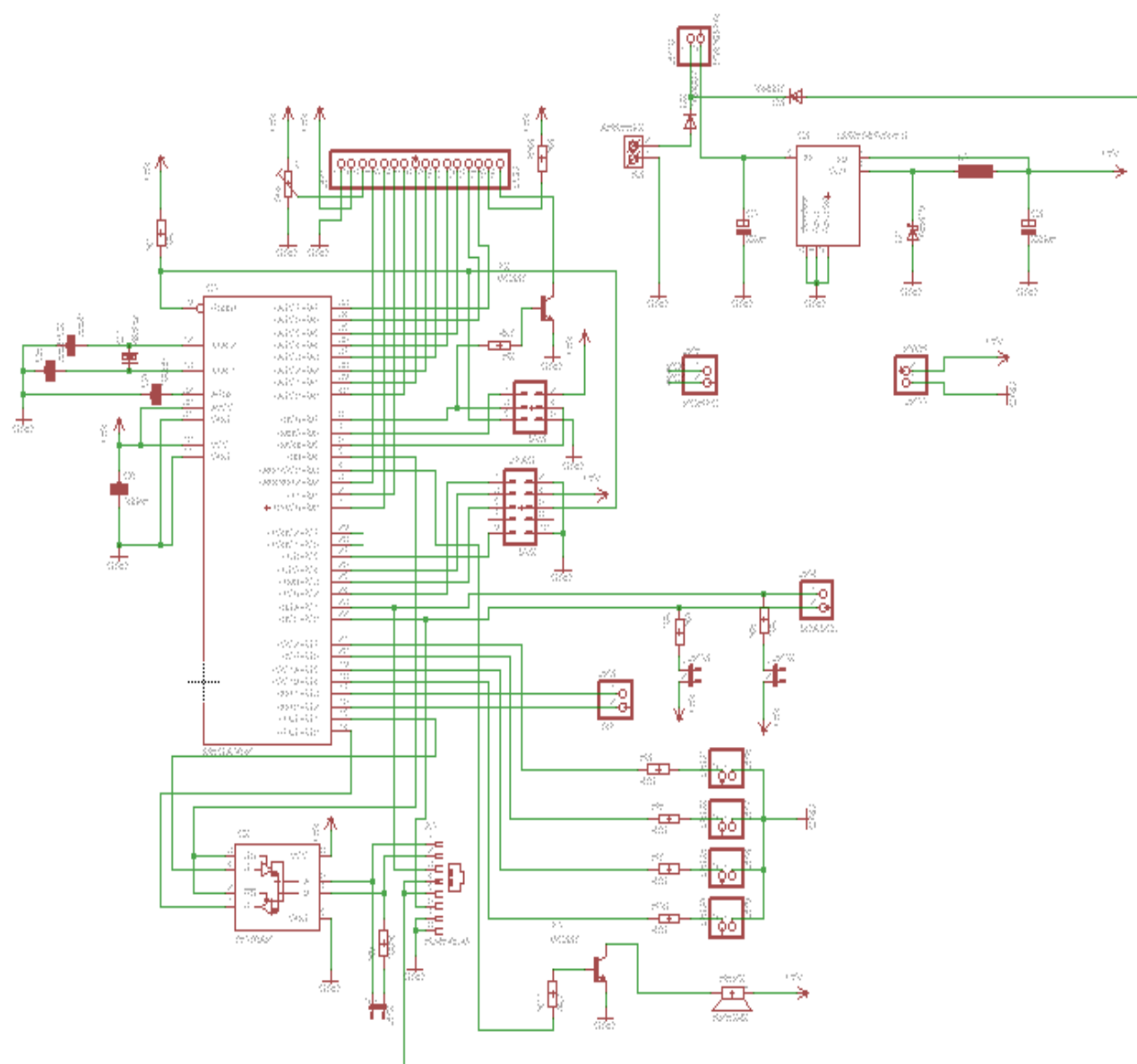
Během testování se objevilo několik nedostatků v návrhu a to především nemožnost použití jedné 9V baterie pro napájení vysílače i zobrazovacího panelu, tato možnost by se ale v případě použití PLC pravděpodobně nevyskytla. Dalším nedostatkem je určitě vzhled výsledného výrobku, který byl vyráběn v domácích podmínkách a tak při použití ve výrobním procesu by bylo nutné provést úpravu vzhledu.

Možností dalšího vývoje jak panelu, tak softwarové aplikace je několik, v první řadě větší podpora MODBUS protokolu, tedy možných funkcí, které protokol poskytuje. Dále pak výpis více informací o vzniklé chybě na displeji. Jednou z dalších možností rozšíření je přidání bezdrátového vysílače a přijímače, tak aby nebylo nutné používat linku RS485.

Literatura

- [1] ŠMEJKAL, L.; MARTINÁSKOVÁ, M.: *PLC a automatizace 1. Základní pojmy, úvod do programování*, Praha: BEN – technická literatura, 2003, ISBN 80-86056-58-9
- [2] MANN, B.: *C pro mikrokontroléry*, Praha: BEN – technická literatura, 2004, ISBN 80-7300-077-6
- [3] FISCHL, T.: *USBasp – USB programmer for Atmel AVR controllers* [online]. Dostupné z <http://www.fischl.de/usbsp/>
- [4] RONEŠOVÁ, A. Ing.: *Přehled protokolu MODBUS* [online]. Květen 2005. Dostupné z <http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>
- [5] MODBUS-IDA.org: *MODBUS over Serial line, Specification and Implementation Guide* [online]. 2006-12-20. Dostupné z http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf
- [6] NYHOLM, K.: *PureJavaComm with JTermios* [online]. 2011-04-05. Dostupné z <http://www.sparetimelabs.com/purejavacomm/purejavacomm.php>
- [7] Simply Modbus: *Data communication test software* [online]. Dostupné z <http://www.simplymodbus.ca/index.html>
- [8] BIES, L.: *RS485 serial information* [online]. Dostupné z <http://www.lammertbies.nl/comm/info/RS-485.html>
- [9] PLC – programovatelné automaty [online]. Dostupné z [iat.fs.cvut.cz/109/files/psar/- 37 -řednášky_2007/plc_vznik_hw_sw.ppt](http://iat.fs.cvut.cz/109/files/psar/-37-řednášky_2007/plc_vznik_hw_sw.ppt)
- [10] ATMEL Corporation: *ATMEGA16 Datasheet (PDF)* [online]. Dostupné z <http://www.alldatasheet.com/datasheet-pdf/pdf/78532/ATMEL/ATMEGA16.html>
- [11] Texas Instruments: *SN75176 Datasheet (PDF)* [online]. Dostupné z <http://www.alldatasheet.com/datasheet-pdf/pdf/28608/TI/SN75176.html>
- [12] TIŠNOVSKÝ, P.: *Sběrnice RS-422, RS-423 a RS-485* [online]. 2008-12-18. Dostupné z <http://www.root.cz/clanky/sbernice-rs-422-rs-423-a-rs-485/>

A. Schémata zapojení



Obrázek A.1 – Schéma zapojení zobrazovacího panelu

B. Seznam použitých součástek

PLC Panel		PLC simulátor	
C1	kondenzátor (22μF)	C1	kondenzátor (22μF)
C2, C5	kondenzátor (33pF)	C2	kondenzátor(220μF)
C3	kondenzátor(220μF)	C6 , C7	kondenzátor(100nF)
C4, C6	kondenzátor(100nF)	C8, C9	kondenzátor (33pF)
D1	Zenerova dioda (1N5819)	D1	Zenerova dioda (1N5819)
D2, D3	dioda (1N4007)	D2	dioda (1N4007)
IC1	μC - AtMega16-16PU	F2	pojistka (0031.8201)
IC2	SN75176	IC1	μC - AtMega16-16PU
IC3	LM2574HVN-5.0	IC2	SN75176
JP1	1x16 Pinhead	IC4	LM2574HVN-5.0
JP2, JP12, JP13	Jumper	JP1, JP2, JP3, JP4	1x2 pinhead
JP3, JP4, JP5, JP6	1x2 Pinhead	JP5, JP6, JP7, JP8	
JP7, JP8, JP9, JP10		JP9, JP11, JP12	
JP11		JP13, JP14	
K3	ARK550/2	JP10, JP15, JP16	Jumper
L1	Cívka (330μH)	K1	ARK500/2
P1	Trimr (50kΩ)	L1	Cívka (330μH)
Q1	Krystal (8MHz)	Q2	Krystal (8MHz)
R1, R4, R6	Rezistor (10kΩ)	R1	Rezistor (10kΩ)
R2, R11	Rezistor (4,7kΩ)	R2, R3, R4, R12	rezistor (470Ω)
R3, R5, R7, R10	rezistor (470Ω)	R5, R6, R9	rezistor (100Ω)
R8	rezistor (6,8Ω)	SV1	2x3 pinhead
R9	rezistor (100Ω)	SV3	2x5 pinhead
REP2	reproduktor (KPE242)	SV4	1x8 pihead
SV2	5x2 Pinhead	SV5	1x4 pinhead
SV3	2x3 Pinhead	X1	Konektor RJ-45
T1, T2	NPN Tranzistor (BC337)	Spínač	5x KNX1
X1	Konektor RJ-45	LED dioda	4x zelená
LED dioda	4x červená	DIP40	1x pouzdro
LCD displej	1x monochromatický (4x20)	DIP8	2xpouzdro
Spínač	1x KNX-1		
DIP40	1x pouzdro		
DIP8	2xpouzdro		

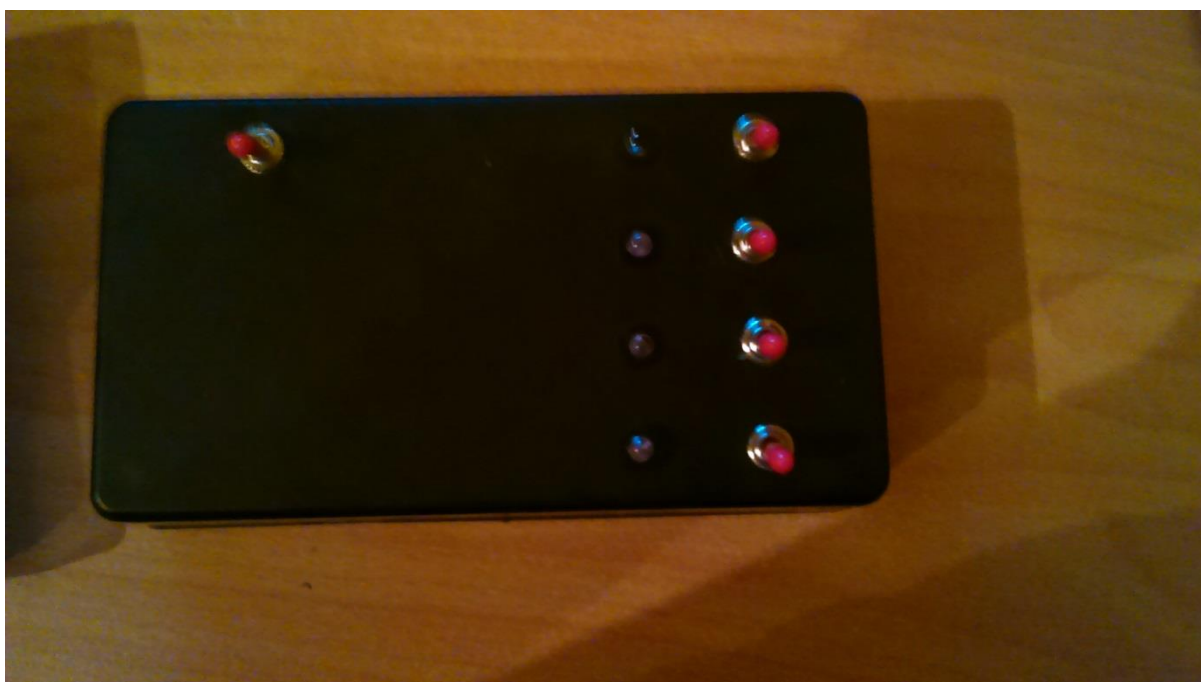
C. Vzhled výsledných produktů



Obrázek C.1 – Zobrazovací panel ve stavu vypnuto



Obrázek C.2 – Zobrazovací panel ve stavu zapnuto, nepřijímající chybu



Obrázek C.3 – Vysílač ve stavu vypnuto



Obrázek C.4 – Vysílač simulující chybu a panel zobrazující přijatou chybu

D. Manuál

V této příloze je popsán postup spuštění a odzkoušení všech zkonstruovaných nebo naimplementovaných částí

D.1. Hardwarová část

Obě hardwarové části (vysílač a panel) jsou vybaveny 9V baterií, která zajišťuje napájení, a také obsahují spínač, který zařízení spouští a vypíná. U panelu je to jediný spínač, umístěný ve spodní části napravo. U vysílače je to spínač umístěný v levém horním rohu. Pro komunikaci stačí připojit ethernetový kabel do konektorů na bocích krabičky.

Pro vyslání zprávy o chybě stačí u panelu sepnout jeden ze čtyř spínačů po pravé straně.

D.2. Softwarová část

Pro spuštění softwarové aplikace je nutné mít nainstalováno Java JDK. Spustit lze příkazem `java -jar SignalizationPanel` v adresáři `dist` na CD.

Po spuštění je nutné připojit se k portu, který je závislý na používaném OS (vyvíjeno v OS Windows 7 – port COM4). Tento port virtuálně vytváří RS485 převodník. A pro úplné zapnutí je nutné přepnout spínač ze stavu OFF na ON.

E. Obsah CD

- dist - obsahuje spustitelný soubor v Java pro softwarovou verzi
- doc - obsahuje pdf a docx soubor s teoretickou částí BP
- eagle - obsahuje schémata a nákresy desek plošných spojů pro program Eagle
- SignalizationPanel - obsahuje zdrojové kódy softwarové aplikace v Java
- PLCPanel - obsahuje zdrojové kódy pro mikroprocesor v signalizačním panelu
- PLCSimulation - obsahuje zdrojové kódy pro mikroprocesor ve vysílači