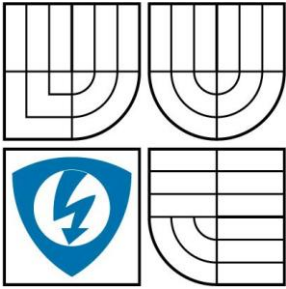


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## MODUL DIGITÁLNÍCH VSTUPŮ S ROZHRANÍM MODBUS

MODULE OF DIGITAL INPUTS WITH MODBUS COMMUNICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

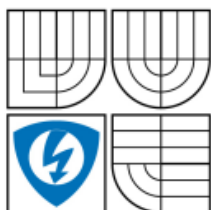
TOMÁŠ NOŽKA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ZDENĚK BRADÁČ, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí  
techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Tomáš Nožka  
**Ročník:** 3

**ID:** 72998  
**Akademický rok:** 2008/2009

## NÁZEV TÉMATU:

**Modul digitálních vstupů s rozhraním Modbus**

## POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte modul digitálních vstupů vybavený mikrokontrolérem I51. Vybavte jej standardní komunikační sběrnici RS232/485 s protokolem Modbus. Oživte a otestujte funkčnost. Sestavte plně komentované programové vybavení v C. Sepište bakalářskou práci.

## DOPORUČENÁ LITERATURA:

Dle pokynů vedoucího práce.

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 1.6.2009

**Vedoucí práce:** doc. Ing. Zdeněk Bradáč, Ph.D.

**prof. Ing. Pavel Jura, CSc.**  
*předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

**ANOTACE:**

Bakalářská práce se zabývá základní specifikací protokolu Modbus a jeho využitím pro komunikaci mezi modulem vzdálených digitálních vstupů a zařízením, které využívá informace získané ze vzdálených vstupů. V bakalářské práci je dále popsán návrh hardwarového a softwarového vybavení modulu. Celé řízení obstarává mikroprocesor řady i8051. Jednotka se chová jako zařízení slave. Komunikace se vzdálenými zařízeními probíhá přes sériové rozhraní (RS-485 a RS-232) pomocí jednoho ze dvou volitelných režimů komunikace – Modbus RTU nebo Modbus ASCII. Vzdáleným zařízením jsou pomocí Modbus zpráv především poskytovány informace o stavu digitálních vstupů. Práce také obsahuje informace o některých profesionálních modulech vstupů a výstupů komunikujících pomocí Modbus protokolu.

**Klíčová slova:** Modbus RTU, Modbus ASCII, I51, RS-232, RS-485, digitální vstupy

**ABSTRACT:**

Bachelor's thesis deals with the basic specification of Modbus protocol and its use for communication between the remote digital inputs device module and device that uses information obtained from remote inputs. The thesis proposal is to further describe hardware and software for the module. Processor i8051 series provides the function of the whole procedure. Unit acts as a slave device. Communication with the remote device is done via serial interface (RS-485 and RS-232) using one of two modes of communication either Modbus RTU or Modbus ASCII. Primarily information on the status of digital inputs are being provided to the remote devices. The work also includes information about some professional modules of inputs and outputs communicating via Modbus protocol.

**Key words:** Modbus RTU, Modbus ASCII, I51, RS-232, RS-485, digital inputs

## **Bibliografická citace**

NOŽKA, T. *Modul digitálních vstupů s rozhraním Modbus*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 71 s.  
Vedoucí semestrální práce doc. Ing. Zdeněk Bradáč, Ph.D.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Modul digitálních vstupů s rozhraním Modbus jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **1. června 2009**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce doc. Ing. Zdeňku Bradáčovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **1. června 2009**

.....  
podpis autora

## OBSAH

<b>1. ÚVOD .....</b>	<b>11</b>
<b>2. PROTOKOL MODBUS .....</b>	<b>12</b>
2.1 Popis protokolu .....	12
2.1.1 Kódování dat.....	14
2.1.2 Datový model.....	14
2.1.3 Kategorie kódů funkcí .....	15
2.1.4 Definice Modbus transakce .....	16
2.2 Modbus na sériové lince .....	18
2.2.1 Adresovací pravidla .....	18
2.2.2 Vysílací režimy .....	18
<b>3. PROFESIONÁLNÍ MODULY S ROZHŘANÍM MODBUS.....</b>	<b>21</b>
3.1 Příklady modulů vstupů a výstupů s Modbus RTU .....	21
3.1.1 Moduly řady M-7000.....	21
3.1.2 Jednotky řady MicroUnit.....	22
3.1.3 Moduly Quido RS 10/1, 30/3, 60/3 a 100/3 .....	22
3.2 Řídící jednotky s protokolem Modbus TCP .....	24
<b>4. POPIS ZAPOJENÍ MODULU .....</b>	<b>25</b>
4.1 Zdroj a napájení modulu .....	25
4.1.1 Popis zapojení zdroje napětí pro modul.....	26
4.2 Energetická rozvaha.....	26
4.3 Mikroprocesor a jeho periferie.....	27
4.3.1 Mikroprocesor AT89C51ED2 .....	27
4.3.2 Popis zapojení mikrokontroléru a periferií .....	31
4.4 Rozhraní RS-232 a RS-485.....	33
4.4.1 Specifikace RS-232 .....	33
4.4.2 Specifikace RS-485 .....	34
4.4.3 Popis zapojení rozhraní RS-232 .....	35
4.4.4 Popis zapojení rozhraní RS-485 .....	36
4.5 Galvanické oddělení digitálních vstupů.....	37
4.5.1 Popis zapojení galvanického oddělení digitálních vstupů .....	37

4.6	Konstrukční uspořádání .....	39
4.7	Deska plošných spojů .....	40
<b>5.</b>	<b>POPIS PROGRAMOVÉHO VYBAVENÍ.....</b>	<b>41</b>
5.1	Struktura programu .....	41
5.2	Komunikace pomocí režimu Modbus RTU .....	47
5.2.1	Popis jednotlivých stavů v RTU módu.....	48
5.2.2	Podporované kódy funkcí pro RTU mód.....	51
5.3	Komunikace pomocí režimu Modbus ASCII .....	62
5.3.1	Popis komunikačních stavů v ASCII módu.....	63
5.3.2	Podporované kódy funkcí pro režim ASCII .....	65
5.4	Přepínání směru toku dat pro RS-485 .....	66
<b>6.</b>	<b>ZÁVĚR.....</b>	<b>67</b>
<b>7.</b>	<b>LITERATURA .....</b>	<b>68</b>
<b>8.</b>	<b>SEZNAM ZKRATEK.....</b>	<b>70</b>

## SEZNAM OBRÁZKŮ

Obr. 2.1	Struktura a rozsah současné verze protokolu Modbus.....	12
Obr. 2.2	Základní tvar Modbus zprávy [5] .....	13
Obr. 2.3	Kategorie funkčních kódů [5] .....	15
Obr. 2.4	Obecný postup zpracování požadavku slave jednotkou [5].....	17
Obr. 2.5	RTU rámeček zprávy [5] .....	19
Obr. 2.6	Pořadí odesílaných znaků CRC [17].....	19
Obr. 2.7	ASCII rámeček zprávy [5].....	20
Obr. 2.8	Pořadí odesílaných znaků LRC pro hodnotu 0x61 [17].....	20
Obr. 3.1	Modul digitálních vstupů M-7051-G [2] .....	21
Obr. 3.2	I/O modul Quido RS 10/1 [19] .....	23
Obr. 3.3	Modulární jednotka I-8431-MTCP s rozšiřující kartou [3] .....	24
Obr. 4.1	Blokové schéma modulu.....	25
Obr. 4.2	Schéma zapojení zdroje napětí pro modul.....	25
Obr. 4.3	Vnitřní bloková struktura procesoru 8051 [20] .....	27
Obr. 4.4	Schéma zapojení mikroprocesoru a jeho periférií .....	31
Obr. 4.5	Synchronizace dat na RS-232 .....	34
Obr. 4.6	Schéma zapojení rozhraní RS-232 a RS-485.....	35
Obr. 4.7	Zapojení galvanického oddělení dig. vstupu č. 1.....	37
Obr. 4.8	Použitý box pro uložení DPS [12] .....	39
Obr. 4.9	Mechanický výkres krabičky RAILBOX 22.5 [12].....	40
Obr. 5.1	Vývojový diagram výpočtu CRC.....	45
Obr. 5.2	Stavový diagram pro slave jednotku.....	46
Obr. 5.3	Stavový diagram pro režim Modbus RTU.....	49
Obr. 5.4	Vývojový diagram režimu RTU a správy čítačů – 1. část.....	58
Obr. 5.5	Vývojový diagram režimu RTU a správy čítačů – 2. část.....	59
Obr. 5.6	Vývojový diagram režimu RTU a správy čítačů – 3. část.....	60
Obr. 5.7	Stavový diagram pro režim Modbus ASCII .....	63

## SEZNAM TABULEK

Tab. 2.1 Datový model Modbus [5] .....	14
Tab. 2.2 Modbus chybové kódy [5] .....	16
Tab. 2.3 Adresní prostor.....	18
Tab. 4.1 Datové signály RS-232 .....	33
Tab. 5.1 Požadavek funkce 0x02 .....	51
Tab. 5.2 Odpověď funkce 0x02 .....	51
Tab. 5.3 Záporná odpověď (chyba) funkce 0x02.....	52
Tab. 5.4 Požadavek funkce 0x03 .....	52
Tab. 5.5 Odpověď funkce 0x03 .....	52
Tab. 5.6 Záporná odpověď (chyba) funkce 0x03.....	52
Tab. 5.7 Význam uchovávacích registrů.....	53
Tab. 5.8 Požadavek funkce 0x06 .....	54
Tab. 5.9 Odpověď funkce 0x06 .....	54
Tab. 5.10 Záporná odpověď (chyba) funkce 0x06.....	55
Tab. 5.11 Požadavek funkce 0x08 .....	55
Tab. 5.12 Odpověď funkce 0x08 .....	56
Tab. 5.13 Záporná odpověď (chyba) funkce 0x08.....	56
Tab. 5.14 Podporované kódy podfunkcí pro funkci 0x08.....	56
Tab. 5.15 Požadavek funkce 0x0B.....	60
Tab. 5.16 Odpověď funkce 0x0B.....	60
Tab. 5.17 Záporná odpověď (chyba) funkce 0x0B .....	61
Tab. 5.18 Použité objekty funkce 0x2B.....	61
Tab. 5.19 Požadavek funkce 0x2B.....	61
Tab. 5.20 Odpověď funkce 0x2B.....	62
Tab. 5.21 Záporná odpověď (chyba) funkce 0x2B .....	62
Tab. 5.22 Požadavek funkce 0x02 pro mód ASCII .....	65
Tab. 5.23 Odpověď funkce 0x02 pro mód ASCII .....	65
Tab. 5.24 Požadavek funkce 0x06 pro mód ASCII .....	66
Tab. 5.25 Odpověď funkce 0x06 pro mód ASCII .....	66
Tab. 5.26 Záporné odpovědi (chyby) funkcí 0x02 a 0x06 pro mód ASCII.....	66

## 1. ÚVOD

Navržený modul digitálních vstupů využívá ke komunikaci s různými typy zařízení protokolu Modbus RTU, který je díky otevřenému standardu v oboru průmyslové automatizace velmi rozšířený. Mezi zařízení využívající tento protokol patří především programovatelné logické automaty (PLC) s přímou podporou tohoto standardu a také systémy PC, které využívají vývojová prostředí s již zabudovaným protokolem Modbus RTU. Tyto systémy mohou využívat jednotku jako svůj vzdálený vstup bez nutnosti instalace nějakých ovladačů.

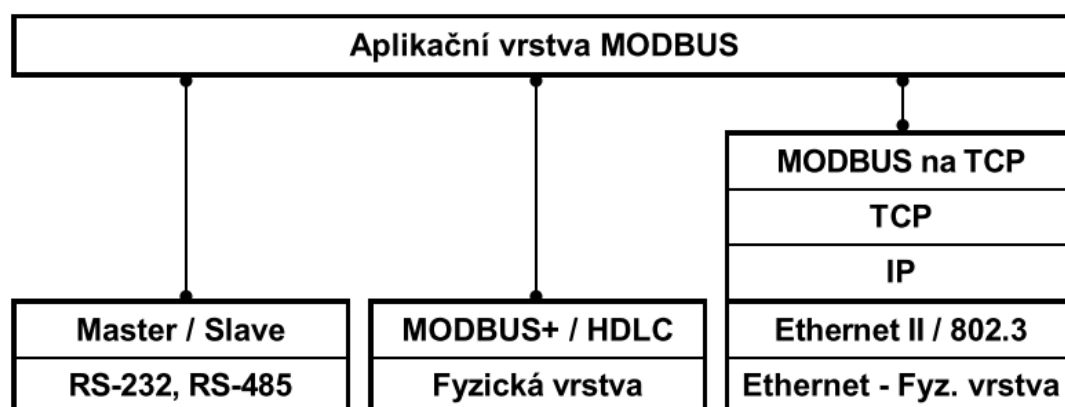
Celý systém řídí mikroprocesor řady I51, konkrétně byl vybrán typ AT89C51ED2. Pro komunikaci s uvedenými zařízeními je jednotka vybavena sériovými komunikačními sběrnicemi RS-232 a RS-485. Jednoduchým přepnutím dvou jumperů lze zvolit, jaké z těchto dvou rozhraní bude právě využíváno pro vysílání a příjem dat. Informace je možné získávat z osmi digitálních vstupů. Všechny tyto vstupy jsou galvanicky odděleny od ostatních obvodů modulu. Pokud je na kterémkoliv vstupu log. 1, je tento stav indikován LED diodou. Výhodou této jednotky je, že je napájena průmyslovým standardem napětí +24 V a má ochranu proti přepětí a přepólování napájení.

Programové vybavení pro mikroprocesor je napsáno v jazyku C. Jednotka dokáže komunikovat pomocí jednoho ze dvou implementovaných komunikačních režimů Modbus. Jedná se o režimy RTU a ASCII. Mód RTU, který je průmyslem více využíván, tedy obsahuje větší množství podporovaných kódů funkcí než mód ASCII. Pro přenos dat je možné zvolit si rychlost komunikace a typ parity. Modul také dokáže uchovávat historii třiceti posledních stavů všech digitálních vstupů. Rozmezí času mezi dvěma uloženými hodnotami stavů je volitelné od 50 milisekund do 65,5 sekundy.

## 2. PROTOKOL MODBUS

Modbus je komunikační protokol, který byl vytvořen v roce 1979 firmou MODICON a postupně se z něj stal průmyslový standard. Pracuje na úrovni aplikační vrstvy ISO/OSI modelu a umožňuje komunikaci typu klient-server mezi různými zařízeními. Komunikace probíhá na principu žádost-odpověď a požadavek je specifikován pomocí tzv. kódu funkce.

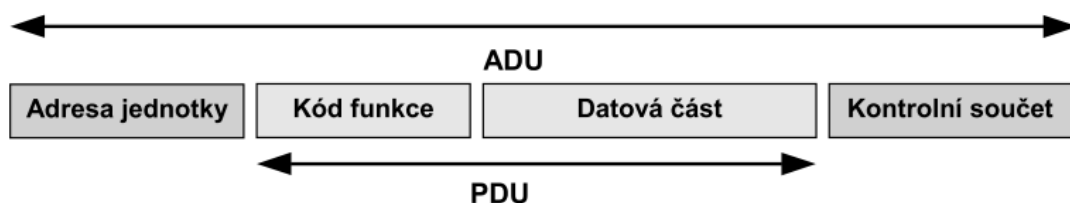
Modbus umožňuje komunikaci po mnoha typech komunikačních rozhraní, kterými jsou například asynchronní sériový přenos (RS-232, RS-422, RS-485...), TCP/IP Ethernet, Modbus + (vysokorychlostní síť), Modbus RTPS (pro systémy reálného času) [5].



Obr. 2.1 Struktura a rozsah současné verze protokolu Modbus

### 2.1 POPIS PROTOKOLU

Protokol Modbus určuje strukturu zprávy na úrovni protokolu (PDU – Protocol Data Unit) nezávisle na typu komunikační vrstvy. Podle typu použité sítě, která využívá protokolu, je PDU rozšířena o další části a vytváří tak zprávu na aplikační úrovni (ADU – Application Data Unit) [5].



Obr. 2.2 Základní tvar Modbus zprávy [5]

Kód funkce udává serveru jaký druh operace má provést. Rozsah kódů je 1 až 255, přičemž kódy 128 až 255 jsou vyhrazeny pro oznámení záporné odpovědi (chyby). Některé kódy funkcí obsahují i kód podfunkce, který upřesňuje požadovanou operaci. Obsah datové části zprávy poslané masterem slouží slave jednotce k uskutečnění operace určené kódem funkce. Některé funkce nepotřebují pro provedení operace žádná další data. Pokud tomu tak je, může datová část ve zprávě úplně chybět. Jestliže se při provádění požadované operace neobjeví chyba, odpoví server zprávou, která má vložen v poli *Kód funkce* kód provedené funkce, což indikuje úspěšné vykonání požadavku. V datové části odpovědi předá slave jednotka masteru požadovaná data (pokud nějaká jsou).

Jestliže při vykonávání požadované operace dojde k chybě, je v poli *Kód funkce* vrácen kód požadované funkce s nastaveným nejvyšším bitem indikujícím neúspěch (exception response). V datové části je vrácen chybový kód (exception code) upřesňující důvod neúspěchu.

Z důvodu možné ztráty požadavku nebo odpovědi, je žádoucí mít na straně mastera časový limit pro přijetí odpovědi, aby master nečekal na odpověď, která kvůli ztrátě nemusí přijít.

Maximální velikost PDU je 253 *Bytů*. Tato velikost je zděděna z první implementace Modbusu na sériové lince RS-485, kde byla maximální velikost ADU 256 *B* (253 *B* PDU + 1 *B* adresa + 2 *B* CRC) [5].

Protokol Modbus definuje 3 základní typy zpráv (PDU):

- **Požadavek (Request PDU)**

- 1 Byte - *Kód funkce*
- *n* Bytů - *Datová část požadavku*

- **Odpověď (Response PDU)**

- 1 Byte - *Kód funkce* (kopie z požadavku)
- *m* Bytů - *Datová část odpovědi*

- **Záporná odpověď (Exception Response PDU)**

- 1 Byte *Kód funkce* + *80h* (indikace neúspěchu)
- 1 Byte *Chybový kód* (identifikace chyby)

### 2.1.1 Kódování dat

Modbus používá tzv. „Big-endian“ reprezentaci dat. To znamená, že při posílání datových položek delších než 1 Byte je jako první poslán nejvyšší Byte a jako poslední nejnižší Byte [5].

### 2.1.2 Datový model

Datový model Modbusu je založen na sadě tabulek, s charakteristickým významem. Definovány jsou čtyři základní tabulky. Mapování tabulek do adresního prostoru je závislé na konkrétním zařízení. Každá z tabulek může mít vlastní adresní prostor nebo se mohou částečně či úplně překrývat [5].

**Tab. 2.1 Datový model Modbus [5]**

Tabulka	Typ položky	Přístup	Popis	Adresa (MODICON)
<b>Diskrétní vstupy</b> ( <i>Discrete Inputs</i> )	1bit	Pouze čtení	Data poskytovaná I/O systémem	10000-19999
<b>Cívky</b> ( <i>Coils</i> )	1bit	Čtení/zápis	Data modifikovatelná aplik. programem	0-9999
<b>Vstupní registry</b> ( <i>Input Registers</i> )	16 bitové slovo	Pouze čtení	Data poskytovaná I/O systémem	30000-39999
<b>Uchovávací registry</b> ( <i>Holding Registers</i> )	16 bitové slovo	Čtení/zápis	Data modifikovatelná aplik. programem	40000-49999

V aplikační paměti zařízení může mít každá tabulka svůj oddělený datový prostor nebo zařízení může obsahovat pouze jeden paměťový blok pro všechny tabulky.

### 2.1.3 Kategorie kódů funkcí

Protokol Modbus definuje tři skupiny kódů funkcí:

#### 2.1.3.1 Veřejné kódy funkcí

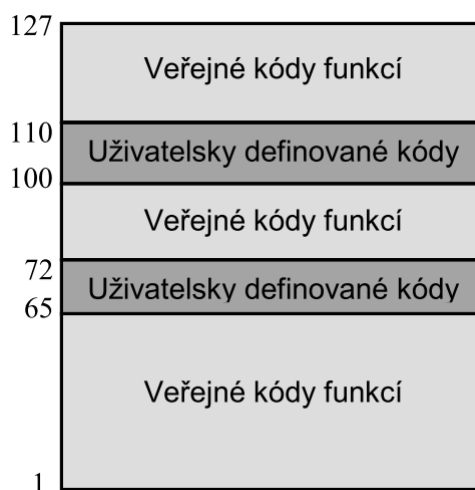
- jsou jasně definované
- je garantována unikátnost
- jsou schvalovány společností MODBUS-IDA.org
- jsou veřejně zdokumentované
- je k nim dostupný test shody
- zahrnují veřejné přiřazené kódy funkcí i nepřiřazené kódy rezervované pro budoucí použití

#### 2.1.3.2 Uživatelsky definované kódy funkcí

- dva rozsahy uživatelsky definovaných kódů funkcí
- umožňují uživateli implementovat funkci, která není definována touto specifikací
- není garantována unikátnost kódů
- lze je po projednání přesunout do veřejných kódů

#### 2.1.3.3 Rezervované kódy funkcí

- jsou v současnosti používány některými firmami a nejsou dostupné pro veřejné použití [5]



Obr. 2.3 Kategorie funkčních kódů [5]

#### 2.1.4 Definice Modbus transakce

Když master posílá slave jednotce požadavek, očekává na něj odpověď. Mohou nastat čtyři situace:

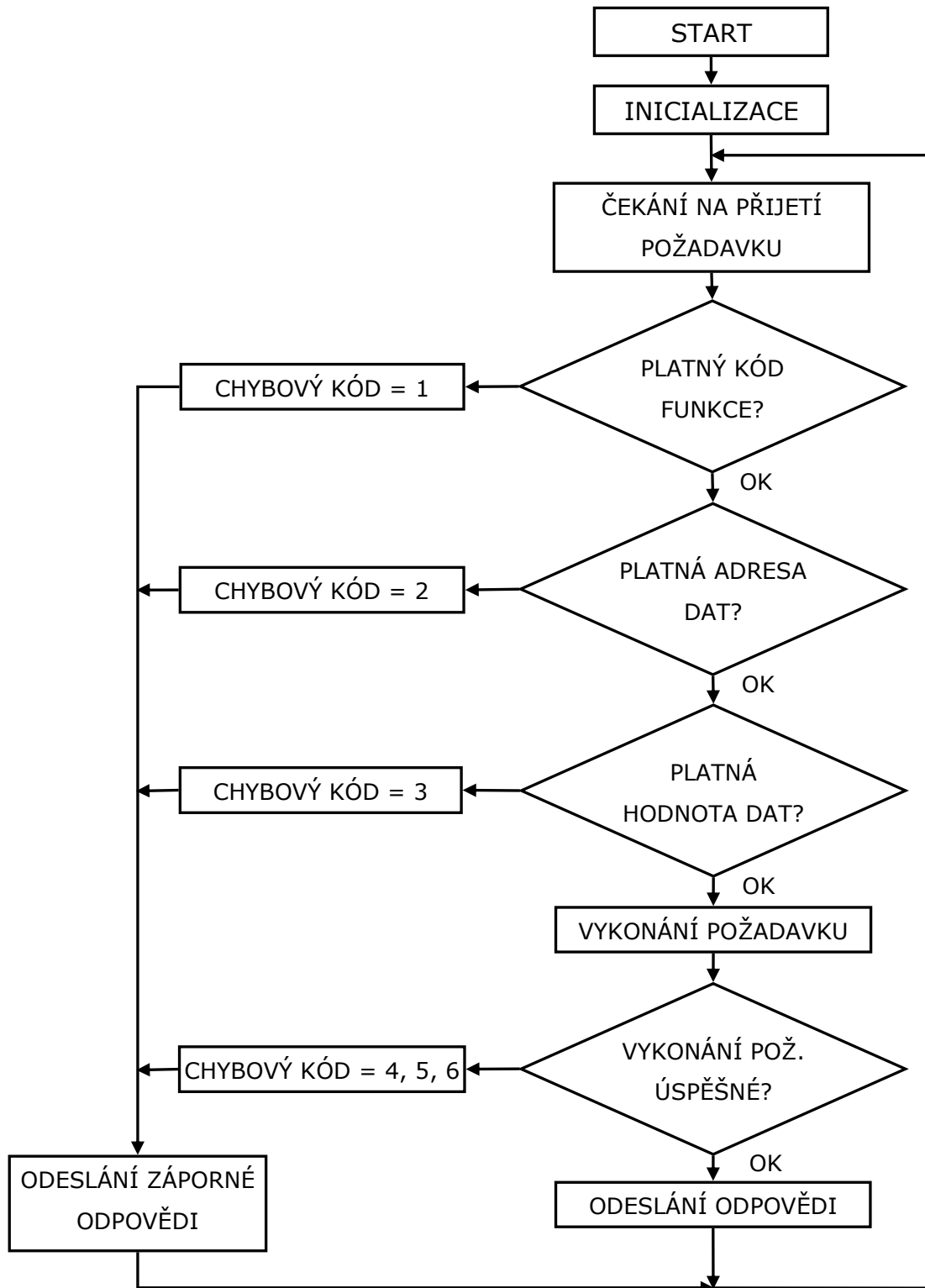
- slave jednotka přijme požadavek bezchybně a je schopna jej normálně zpracovat – masteru je odeslána normální odpověď
- slave jednotka požadavek nepřijme z důvodu komunikační chyby – masteru není odeslána žádná odpověď, dojde tedy k vypršení časového limitu pro příjem odpovědi
- slave jednotka přijme požadavek, ale detekuje komunikační chybu (parita, CRC...) – masteru není odeslána žádná odpověď, dojde tedy k vypršení časového limitu pro příjem odpovědi
- slave jednotka přijme bezchybně požadavek, ale není schopna jej normálně zpracovat – masteru je odeslána záporná odpověď s udáním důvodu neúspěchu

Normální a záporná odpověď se liší nejvyšším bitem kódu funkce. Pokud je bit v log. 0, jedná se o normální odpověď. Je-li bit v log. 1, jedná se o zápornou odpověď. V případě záporné odpovědi obsahuje datová část kód chyby. V Tab. 2.2 je uveden seznam chybových kódů, které mohou nastat [5].

**Tab. 2.2 Modbus chybové kódy [5]**

Kód	Jméno	Význam
01	Ilegální funkce	Požadovaná funkce není podporována
02	Ilegální adresa dat	Zadaná adresa je mimo podporovaný rozsah
03	Ilegální hodnota dat	Předávaná data jsou neplatná
04	Selhání zařízení	Při provádění požadavku došlo k neodstranitelné chybě
05	Potvrzení	Kód určený k použití při programování. Slave jednotka hlásí přijetí platného požadavku, ale jeho vykonání bude trvat delší dobu
06	Zařízení je zaneprázdněné	Kód určený k použití při programování. Slave jednotka je zaneprázdněna vykonáváním dlouho trvajících příkazů.
08	Chyba parity paměti	Kód určený k použití při práci se soubory. Slave jednotka při pokusu přečíst soubor zjistila chybu parity

Ve stavovém diagramu na Obr. 2.4 je možné vidět obecný postup zpracování Modbus požadavku slave jednotkou.



Obr. 2.4 Obecný postup zpracování požadavku slave jednotkou [5]

## 2.2 MODBUS NA SÉRIOVÉ LINCE

Modbus Serial Line protokol je protokol typu Master-Slave, který pracuje na úrovni 2 ISO/OSI modelu. Na fyzické úrovni 0 ISO/OSI modelu lze použít různá sériová rozhraní, například RS-232 nebo RS-485.

Současně může být na sběrnici pouze jeden master a 1 až 247 Slave jednotek. Komunikaci vždy zahajuje master, Slave nesmí nikdy vysílat data bez pověření mastera. Master posílá požadavky Slave jednotkám ve dvou režimech:

- **unicast režim** – master posílá požadavek pouze jedné Slave jednotce pomocí konkrétní adresy a ta vrací odpověď
- **broadcast režim** – master posílá požadavek všem jednotkám, žádná jednotka nevrací odpověď [5]

### 2.2.1 Adresovací pravidla

Master nemá žádnou specifickou adresu, ale každá Slave jednotka musí mít pro celou Modbus síť jedinečnou adresu. [5].

**Tab. 2.3 Adresní prostor**

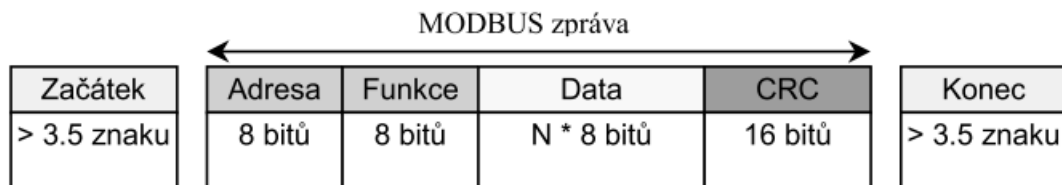
0	1 až 247	248 až 255
Broadcast adresa	Individuální adresa Slave jednotky	Rezervováno

### 2.2.2 Vysílací režimy

Protokol Modbus definuje dva sériové vysílací režimy, Modbus RTU a Modbus ASCII. Režim udává, jaký formátu bude použit na vysílání a dekodování dat. Každá jednotka musí podporovat režim RTU, režim ASCII není povinný. Všechny moduly na jedné sběrnici musejí pracovat ve stejném vysílacím režimu [5].

#### 2.2.2.1 Modbus RTU

V režimu RTU obsahuje každý 8 bitový byte zprávy dva 4 bitové hexadecimální znaky. Při vysílání nesmějí být mezery mezi znaky delší než 1.5 znaku. Začátek a konec zprávy je detekován podle pomlky na sběrnici delší než 3.5 znaku. K detekci chyb slouží 16 bitové CRC. Formát RTU rámce je zobrazen na Obr. 2.5 [5].



Obr. 2.5 RTU rámec zprávy [5]

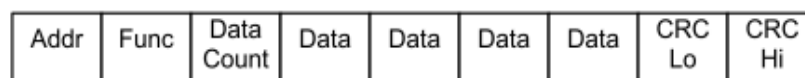
**Formát bytu (11 bitů):** 1 start bit, 8 datových bitů, 1 bit parita, 1 stop bit

Každá jednotka musí podporovat sudou paritu. Pokud není použita parita, je nahrazena druhým stop bitem [5].

#### CRC – Cyklický kontrolní (redundantní) součet

Speciální funkce, používaná k detekci chyb, které mohou vzniknout během přenosu dat. Kontrolní součet je odeslán nebo přijímán společně s daty, při jejichž přenosu by mohlo dojít k chybě. Po příjmu dat je znovu nezávisle vypočítán. Pokud je nezávisle vypočítaný kontrolní součet odlišný od přijatého, je zřejmé, že při přenosu dat došlo k chybě. Jestliže se shoduje, tak téměř jistě k žádné chybě nedošlo. Výpočet CRC je popsán v kapitole 5.1 [2].

Podle protokolu Modbus je 16 bitové CRC rozděleno na dva osmibitové znaky. Spodních osm bitů je přijímáno nebo odesíláno jako předposlední znak a horních osm bitů je přijímáno nebo odesíláno jako znak poslední.



Obr. 2.6 Pořadí odesílaných znaků CRC [17]

#### 2.2.2.2 Modbus ASCII

V režimu ASCII je posílán každý 8 bitový byte jako dvojice ASCII znaků. Mód ASCII je tedy pomalejší, než režim RTU. Umožňuje ale vysílat znaky s mezerami až 1 s. Začátek a konec zprávy je určen odlišně od RTU módu. Začátek zprávy je definován znakem „:“ a konec zprávy dvojicí řídicích znaků CR a LF. K detekci chyb je použito 8 bitové LRC pole. Formát ASCII rámce je zobrazen na Obr. 2.7 [5].

Začátek	Adresa	Funkce	Data	LRC	Konec
znak „:“	2 znaky	2 znaky	0 až 2*252 znaků	2 znaky	2 znaky CR, LF

**Obr. 2.7 ASCII rámeček zprávy [5]**

**Formát bytu (10 bitů):** 1 start bit, 7 datových bitů, 1 bit parita, 1 stop bit

Každá jednotka musí podporovat sudou paritu. Pokud není použita parita, je nahrazena druhým stop bitem [5].

### LRC – Kontrola podélnou paritou

Funkce pro kontrolu chyb vzniklých během přenosu dat, jejíž hodnota je uložena v jednom bytu. LRC je odeslán nebo přijímán společně s daty, při jejichž přenosu by mohlo dojít k chybě. Po příjmu dat je LRC opět vypočítán nezávisle na přijaté hodnotě. Jestliže je vypočítaná hodnota odlišná od přijaté, vznikla chyba při přenosu dat. Jestliže je shodná, pravděpodobně k žádné chybě nedošlo.

Při výpočtu LRC je konečná hodnota reprezentována jako dvojkový doplněk výsledného součtu všech znaků zprávy vyjma znaků „:“, CR a LF. Pokud je při sčítání překročena hodnota 255, byte je nastaven na nulu a výpočet pokračuje dále.

Podle protokolu Modbus je osmibitové LRC rozděleno na dva znaky ASCII. Nejdříve je odeslán či přijímán významnější ASCII znak a následně znak méně významný [17].

„:“	Addr	Func	Data Count	Data	Data	Data	Data	LRC Hi	LRC Lo	CR	LF
								"6" 0x36	"1" 0x31		

**Obr. 2.8 Pořadí odesílaných znaků LRC pro hodnotu 0x61 [17]**

### 3. PROFESIONÁLNÍ MODULY S ROZHRANÍM MODBUS

#### 3.1 PŘÍKLADY MODULŮ VSTUPŮ A VÝSTUPŮ S MODBUS RTU

##### 3.1.1 Moduly řady M-7000

Firma ICP DAS Co., Ltd., která je známá jako výrobce mnoha různých prostředků pro měření, řízení a komunikaci v průmyslu, nabízí moduly řady M-7000. Tyto měřicí a řídicí moduly komunikují po sběrnici RS-485 s podporou protokolu Modbus RTU a textového protokolu DCON. Podpora protokolu Modbus RTU umožňuje jednotkám přímou komunikaci s většinou SCADA / HMI programů, nebo připojení k PLC automatům bez nutnosti instalace dalších ovladačů. Sérii M-7000 v dnes tvoří asi 32 typů modulů [3] [14].

Mezi typické vlastnosti těchto jednotek patří:

- galvanické oddělení mezi vstupy a sběrnici 3000 V DC
- přepět'ová ochrana vstupů do 240 V<sub>rms</sub> (u jednotek s označením R)
- digitální vstupy až do 50 V DC
- ochrana proti zkratu na digitálních výstupech
- vstupní napětí od +10 V do +30 V
- detekce přerušení obvodu u termočlávkových modulů
- Vícestupové analogové moduly, možnost samostatné konfigurace každého kanálu [14]



Obr. 3.1 Modul digitálních vstupů M-7051-G [2]

### 3.1.2 Jednotky řady MicroUnit

Další ze společností, které nabízejí moduly s implementovaným protokolem Modbus RTU, je plzeňská firma TEDIA s jednotkami řady MicroUnit. První moduly této série byly uvedeny na trh před více než deseti lety. Jejich současná nabídka obsahuje velké množství typů modulů analogových či digitálních vstupů a výstupů. Moduly jsou vsazeny do boxů určených pro montáž na lištu DIN. Komunikace probíhá po sériovém standardu RS-485. Rychlost přenosu dat je volitelná v rozsahu od 2 400 Bd do 115,2 kBd (u vybraných jednotek je až 460,8 kBd). Nastavit lze i typ paritního bitu [13].

Osm zabudovaných funkcí protokolu Modbus RTU umožňuje:

- zapisovat data na výstupní periferie (digitální výstupy, analogové výstupy, modulátory PWM apod.)
- číst informace z výstupních periferií
- číst stav vstupních periferií (digitální vstupy, analogové vstupy, čítače apod.)
- zapisovat a číst vnitřní konfigurační registry modulu
- přistupovat registrům určeným pro diagnostiku [13]

### 3.1.3 Moduly Quido RS 10/1, 30/3, 60/3 a 100/3

Moduly jsou produkty společnosti Papouch s.r.o. Vstupně / výstupní jednotka Quido RS 10/1 obsahuje deset digitálních vstupů a jeden výstup. Komunikace probíhá pomocí linek RS-232 nebo RS-485. Na vstupy lze pro detekci stavů připojit napětí nebo spínací kontakt. Výstupem je přepínací kontakt relé. Rozdíly mezi jmenovanými jednotkami jsou pouze v počtu vstupů a výstupů.

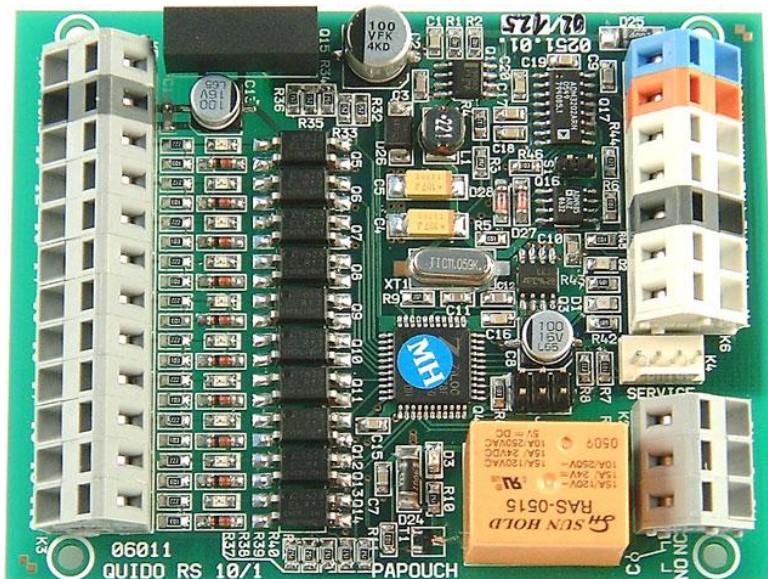
Mezi základní funkce modulu patří čtení aktuálního stavu vstupů, ovládání výstupního relé s přepínacím kontaktem, automatické odesílání informace o změně na vstupech, počítání impulsů na vstupech nebo počítání změn stavu vstupů [19].

### Vlastnosti I/O modulu RS 100/3:

- 100 galvanicky oddělených logických vstupů pro napětí nebo pro kontakt
- tři výstupy s přepínacím kontaktem relé
- automatická reakce zařízení na změnu stavu vstupu
- Komunikace pomocí protokolu Spinel nebo Modbus RTU
- Nastavování výstupů na definovanou dobu
- Široký rozsah napájecího napětí [19]

### Volitelné funkce I/O modulu

- sepnutí výstupů závislé na nastavené kombinaci vstupů
- přizpůsobení pro jiný komunikační protokol
- paměť pro větší počet změn stavů na vstupech
- změna úrovně vstupního napětí
- spínání výstupů na základě „odposlechu“ komunikace dvou zařízení pomocí rozhraní RS-232 nebo RS-485 [19]



Obr. 3.2 I/O modul Quido RS 10/1 [19]

### 3.2 ŘÍDICÍ JEDNOTKY S PROTOKOLEM MODBUS TCP

Programovatelné řídicí jednotky komunikující prostřednictvím protokolu Modbus TCP, který využívá pro komunikaci síť typu Ethernet, jsou nabízeny například společností ICP DAS Co., Ltd. pod označením I-8431-MTCP a I-8831-MTCP. Moduly je možné osazovat rozšiřujícími kartami se sériovou sběrnici řady I-87xxx a paralelní sběrnici I-8xxx. Jednotka I-8431-MTCP poskytuje čtyři a jednotka I-8831-MTCP osm volných slotů pro rozšiřující karty. V současnosti je k dispozici asi 90 rozšiřujících karet s různými kombinacemi a typy digitálních, analogových a čítačových vstupů a výstupů [3].



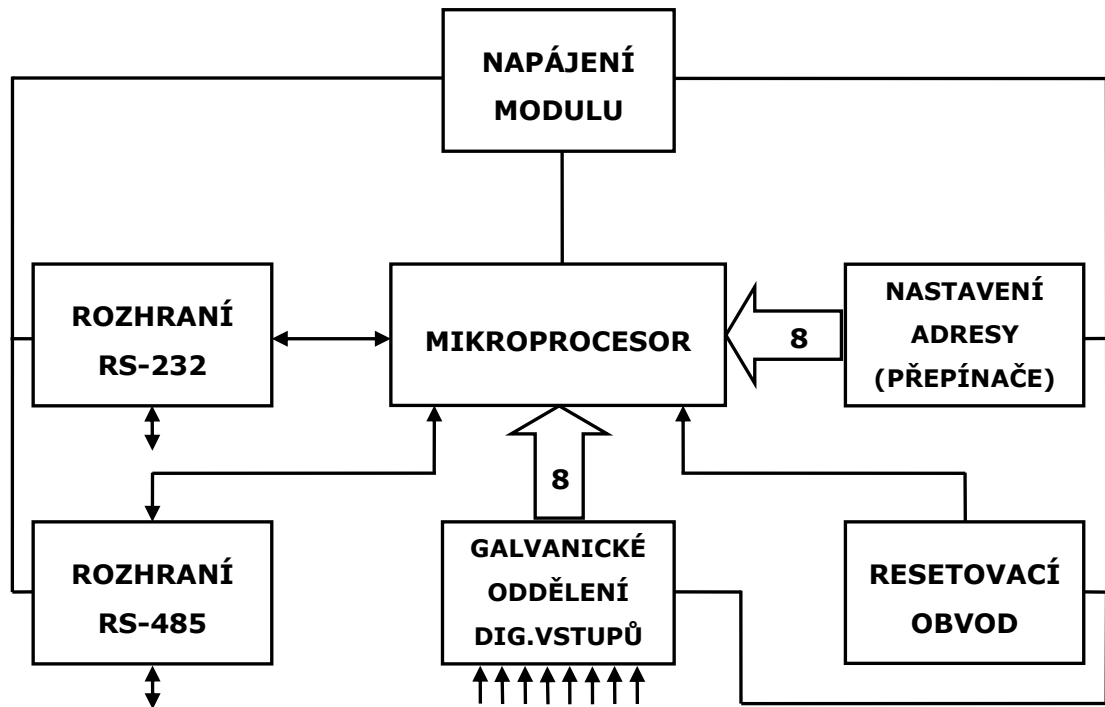
**Obr. 3.3 Modulární jednotka I-8431-MTCP s rozšiřující kartou [3]**

Jednotky obou typů obsahují již hotový software s těmito parametry:

- chovají jako zařízení typu Modbus TCP Slave, ke kterému je možné připojit až osm zařízení typu Modbus TCP Master
- je podporován protokol Modbus TCP pro přístup k vstupům a výstupům rozšiřujících karet
- sériové porty na řídicí jednotce lze nastavit jako virtuální sériové porty v rámci systému Windows
- rozšiřující karty jsou automaticky detekovány a namapovány do registrů vstupů a výstupů

Nevyhovuje-li standardní programové vybavení, je k jednotkám dodáváno vývojové prostředí Modbus SDK, díky němuž může uživatel standardní program modifikovat podle svých požadavků [3].

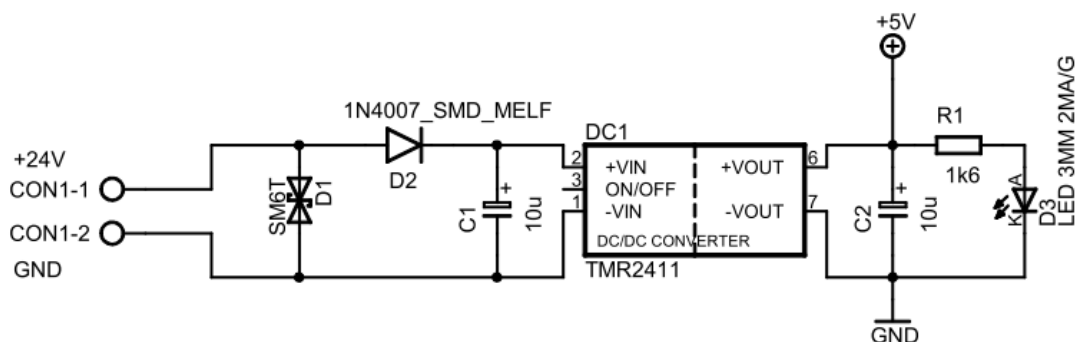
## 4. POPIS ZAPOJENÍ MODULU



Obr. 4.1 Blokové schéma modulu

V následujících kapitolách jsou postupně popisovány jednotlivé bloky uvedené na Obr. 4.1. Celé schéma zapojení modulu je zobrazeno v příloze 1.

### 4.1 ZDROJ A NAPÁJENÍ MODULU



Obr. 4.2 Schéma zapojení zdroje napětí pro modul

#### 4.1.1 Popis zapojení zdroje napětí pro modul

Vstupní napájení modulu je navrženo na napěťový průmyslový standard +24 V. Je ale možné připojit napětí v rozmezí +18 V až +35 V. Napětí 24 V je přivedeno přes diodu  $D_2$  na spínaný zdroj (DC/DC měnič) TMR2411, který ho transformuje na napětí 5 V. Dioda  $D_2$  je použita, aby zabránila přepólování vstupů DC/DC měniče. Jedná se o běžnou usměrňovací diodu 1N4007. Transil SM6T zapojený mezi kladným a záporným vstupem zdroje slouží k ochraně proti zničení vstupu při vysokém napětí (přepětí). Kondenzátory  $C_1$  a  $C_2$  jsou připojeny podle doporučení výrobce DC/DC měniče a jejich hodnoty jsou 10  $\mu$ F. Na výstupu spínaného zdroje je také připojena indikační zelená LED dioda  $D_3$  v sérii s odporem  $R_1$ . Hodnota odporu  $R_1$  1,6 k $\Omega$  byla vypočítána podle (1) na odběr proudu LED diodou 2 mA [6].

$$R_1 = \frac{U_{CC} - U_{D3}}{I_{D3}} = \frac{5V - 1,9V}{0,002A} = 1550 \Omega \doteq 1600 \Omega \quad (1)$$

$U_{CC}$ ..... napájecí napětí [V]

$U_{D3}$  ..... úbytek napětí na diodě  $D_3$  [V]

$I_{D3}$ ..... proud diodou  $D_3$  [A]

#### 4.2 ENERGETICKÁ ROZVAHA

Mikrokontrolér AT89C51ED2	$I_{MAX}$	24 mA
Integrovaný obvod MAX232	$I_{MAX}$	15 mA
Integrovaný obvod LTC1485	$I_{MAX}$	2,3 mA
Nulovací obvod TL7705	$I_{MAX}$	3 mA
Indikační diody (celkem 9)	$I_{MAX}$	18 mA
Ostatní součástky	<u><math>I_{MAX}</math></u>	<u>40 mA</u>
Celkem	$I_{MAX}$	102,3 mA

Z vypočítané hodnoty maximálního proudového odběru vyplývá, že DC/DC měnič TMR 2411 dostačuje. Jeho maximální výstupní proud je 400 mA [9].

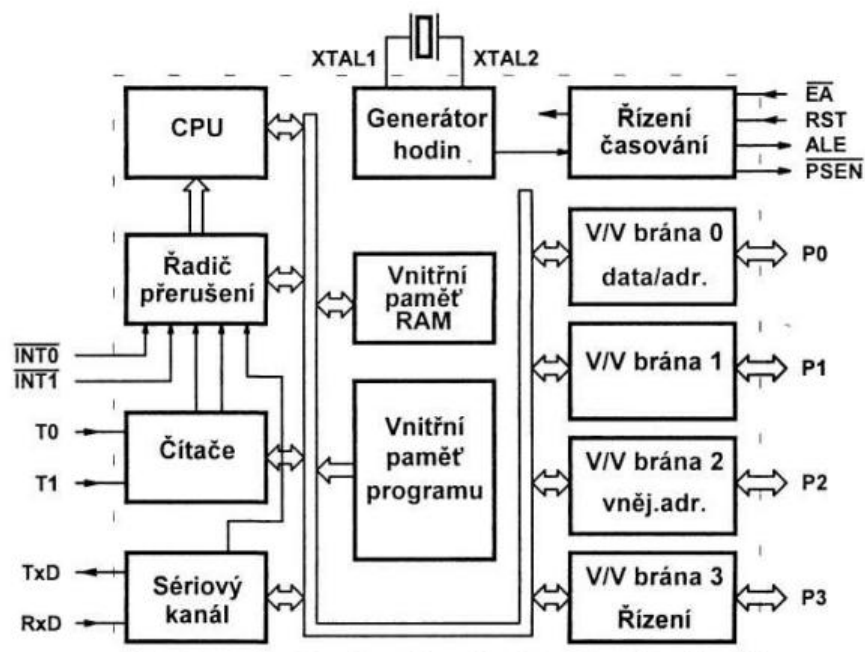
Údaje o maximální spotřebě byly nalezeny v katalogových listech příslušných obvodů a součástek [6] [7] [8] [10] [11].

### 4.3 MIKROPROCESOR A JEHO PERIFERIE

#### 4.3.1 Mikroprocesor AT89C51ED2

Mikroprocesor AT89C51ED2 patří do rodiny osmibitových jednočipových mikroprocesorů řady 8051. Ty jsou založeny na harvardské architektuře, u které je oddělena programová a datová paměť. Tento typ je produktem firmy ATMEL a je vyrobený technologií CMOS.

Podrobné informace o mikrokontroléru AT89C51ED2 jsou uvedeny v [8].



Obr. 4.3 Vnitřní bloková struktura procesoru 8051 [20]

##### 4.3.1.1 Vybrané rysy mikrokontroléru

- založen na procesoru 80C52
- kompatibilní s instrukční sadou Intel 8051
- čtyři 8 bitové vstupně / výstupní porty
- tři 16 bitové čítače / časovače
- sériový kanál UART, rozhraní SPI

- 256 Bytová rychlá paměť RAM, 1792 B paměti XRAM (přístupná přes instrukci MOVX)
- 64 kB programové paměti typu Flash
- 2 kB paměti EEPROM pro data
- devět zdrojů přerušení se čtyřmi úrovněmi priority
- dva registry DPTR
- hardwarový Watchdog
- integrovaný Power Monitor – hlídá kolísání napájecího napětí
- ISP – programování procesoru pomocí Bootloaderu přes sériovou linku, k programování je použito standardního napájecího napětí
- Bootloader umístěn ve zvláštní paměti ROM
- pracovní frekvence až 40 MHz, pro standardní mód
- Napájecí napětí v rozsahu 2.7 V to 5.5 V
- Rozsah pracovních teplot -40 °C až +85 °C [8]

#### 4.3.1.2 Organizace paměti

Mikroprocesor AT89C51ED2 má oddělené adresové prostory programu a dat. Tyto prostory jsou přístupné různými instrukcemi. Paměťový prostor lze dále rozdělit na vnitřní, integrovaný na čipu, a vnější, který lze realizovat pomocí externí paměti. Pro přepínání mezi vnitřním a vnějším programovým adresovým prostorem je procesor vybaven vstupem  $\overline{EA}$ . Je-li vstup  $\overline{EA} = 0$ , potom je programová paměť tvořena celou vnější pamětí (až 64 kB). Je-li vstup  $\overline{EA} = 1$ , instrukce se čtou z 64 kB vnitřní paměti typu Flash a externí paměť je v tomto případě vypnutá. Vnitřní paměť programu je rozdělena do 512 stránek po 128 B.

Vnitřní datová paměť RAM 256 B je určena v rozmezí adres 00h až 07h (128 B) pro ukládání proměnných do registrů a pro zásobník, adresy 08h až FFh ukazují do registru speciálních funkcí (SFR). Pokud je třeba přistupovat do zbylých 128 B vnitřní paměti dat, je třeba využít nepřímého adresování. Vnější datová paměť RAM s kapacitou až 64 kB je přístupná přes registr DPTR [8] [20].

#### **4.3.1.3 Registry speciálních funkcí (SFR)**

Všechny informace, důležité pro činnost mikroprocesoru a jeho periférií integrovaných na čipu, jako jsou čítače, sériový kanál, přerušovací systém atd., jsou uloženy do tzv. registrů speciálních funkcí [20].

#### **4.3.1.4 Čítače / časovače**

Mikrokontrolér AT89C51ED2 obsahuje tři 16 bitové čítače, jejichž obsah je přístupný pomocí paměťově mapovaných registrů TH0, TL0 (čítač 0), TH1, TL1 (čítač 1) a TH2, TL2 (čítač 2). Tyto registry určují vyšších a nižších 8 bitů příslušného čítače. Hodinový synchronizační signál čítačů může být odvozen z vnějšího zdroje, přivedeného na vývody procesoru T0 a T1 nebo z oscilátoru procesoru. Je-li zdrojem signálu vnitřní oscilátor procesoru, potom je čítač ve funkci časovače a inkrementuje se v každém strojovém cyklu. Strojový cyklus je tvořen 12 periodami oscilátoru. Ve funkci čítače vnějších událostí se obsah příslušného čítače (registru) inkrementuje po přechodu signálu z log. 1 do log. 0. Protože zjištění změny na vstupech trvá 2 strojové cykly (24 period oscilátoru), je maximální číтанá frekvence vnějšího signálu 1/24 kmitočtu oscilátoru mikroprocesoru. Logická úroveň číтанého signálu musí zůstat nezměněna vždy alespoň 1 celý strojový cyklus.

Konfiguraci čítače / časovače 0 a 1 zajišťujeme naprogramováním registru TMOD. Pro nastavení čítače / časovače 2 slouží registr T2MOD. Čítače se programově spouští nebo zastavují nastavením nebo vynulováním bitu TR<sub>n</sub> v registru TCON, v případě čítače 2 v registru T2CON [8] [20].

#### **4.3.1.5 Přerušeni**

Pro snazší komunikaci s vnějšími perifériemi je mikroprocesor AT89C51ED2 vybaven přerušovacím systémem s devíti zdroji přerušeni.

Vnější přerušeni INT0 a INT1 mohou být vyvolána buď logickou úrovní (log. 0) nebo změnou logické úrovně (sestupnou hranou z log. 1 do log. 0). Vznikne-li vnější přerušeni, je nastaven příslušný příznak IE0, IE1. Jakmile je vyvolán obslužný podprogram přerušeni, je příznak obvodově automaticky vynulován. Přerušeni od časovače 0, 1 a 2 se vyvolávají nastavením příznaků TF0, TF1 a TF2. Tyto příznaky indikují přetečení příslušného časovače. Jakmile dojde

k přerušení od časovače, odpovídající příznak TF<sub>n</sub> je vynulován při přechodu do obslužného podprogramu.

Přerušení od sériového kanálu se generuje logickým součtem příznaků RI a TI. Aby bylo možné zjistit, jestli bylo přerušení generováno příznakem RI (příjem) nebo TI (vysílání), nejsou příznaky automaticky nulovány při přechodu do obslužného podprogramu. V obslužném programu je nejdříve třeba stanovit příčinu přerušení a poté příslušný příznak programově vynulovat. V obslužném podprogramu pro sériový kanál se také zároveň programově rozhoduje o tom, která žádost (RI nebo TI) bude zpracována dříve a bude tak mít vyšší prioritu.

Další možnosti přerušení tohoto procesoru jsou přerušení od klávesnice KBD IT, od SPI (Serial Port Interface) SPI IT a PCA (Programmable Counter Array - programovatelné čítačové pole) PCA IT.

Všechny příznaky, které generují přerušení, mohou být programově vynulovány nebo nastaveny a to nastavením příslušného bitu v registrech speciálních funkcí IEN0 a IEN1. Pro nastavení priority přerušení slouží registry IPH0, IPL0, IPH1 a IPL1 [8] [20].

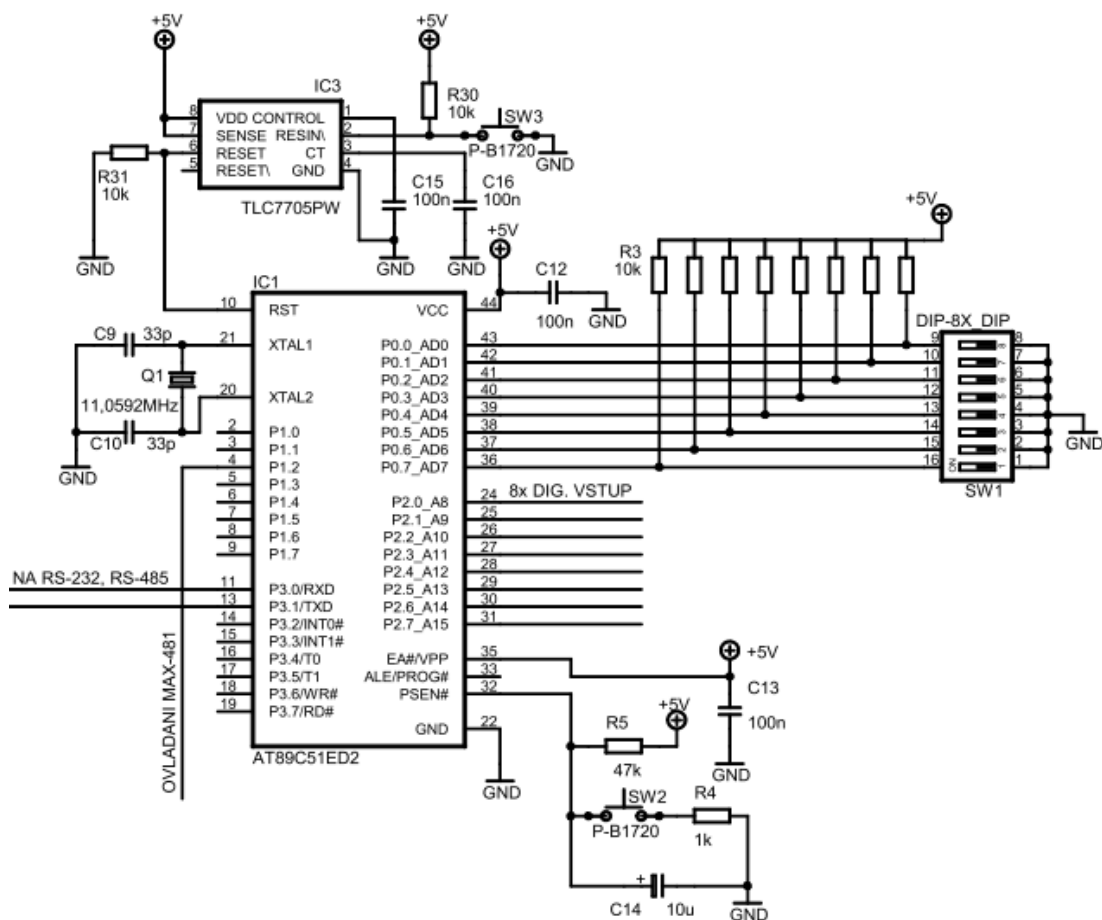
#### **4.3.1.6 Sériový kanál UART**

Další periferií mikropočítače AT89C51ED2 je plně duplexní sériový kanál integrovaný na čipu. Ten umožňuje komunikaci ve standardním 8 a 9bitovém asynchronním režimu nebo 8 bitovém synchronním režimu s pevnou přenosovou rychlostí. Plně duplexní sériový kanál umožňuje současně vysílat i přijímat hodnoty z vývodů RxD (příjem dat) a TxD (vysílání dat). Příjmací kanál je vybaven vyrovnávacím registrem, do kterého je uložena právě přijatá hodnota. Tím je umožněn okamžitý příjem další hodnoty. Přijatá hodnota však musí být převzata dříve, než je dokončen příjem následující hodnoty. Pokud by tomu tak nebylo, nová hodnota by přepsala původní hodnoty. Procesor není vybaven příznaky, které indikují ztrátu přijaté hodnoty, chybu rámce a parity nebo indikaci přerušení, které jsou obvyklé u specializovaných obvodů.

Přijímací i vysílací registr je dostupný na stejné adrese. Označuje se SBUF, což je registr sériového kanálu v prostoru speciálních funkcí. Zápisem se naplňuje vysílací registr, při čtení SBUF přečte hodnotu z vyrovnávacího registru.

Sériový kanál může pracovat ve čtyřech módech, které lze nastavit pomocí registru SCON a nejvyšším bitem v registru PCON [8] [20].

### 4.3.2 Popis zapojení mikrokontroléru a periférií



Obr. 4.4 Schéma zapojení mikroprocesoru a jeho periférií

Srdcem celého modulu je mikroprocesor ATMEL AT89C51ED2 v pouzdře PLCC44, který vychází architektury Intel 8051. Tento byl zvolen především díky velké vnitřní paměti programu (64 kB). Procesor je řízen krystalem  $Q_1$  o frekvenci 11,0592 MHz připojeným na piny XTAL1 a XTAL2. Tato frekvence je zvolena

z důvodu přesnějšího výpočtu přenosové rychlosti sériové linky. Mezi napájení procesoru a zem je zapojen filtrační kondenzátor  $C_{12}$  100 nF, který brání možným zákrmitům vzniklých v DC/DC měniči. Jelikož bude pracováno pouze s vnitřní pamětí mikroprocesoru, je na vývod  $\overline{EA}$  přivedena log. 1 reprezentovaná hodnotou napájecího napětí +5 V. Tento typ procesoru podporuje programování přes sériovou linku pomocí bootloaderu. To lze vyvolat, pokud je v době spádové hrany signálu RESET splněna podmínka, že je vývod  $\overline{PSEN}$  v log. 0,  $\overline{EA}$  v log. 1 a ALE v log. 1 nebo nezapojen. V tu chvíli se začne program vykonávat od adresy FC00h, kde je umístěn bootloader, namísto od adresy 0000h. Proto je na vývod  $\overline{PSEN}$  připojen spínač  $SW_2$ , jenž po stisknutí zkratuje kondenzátor  $C_{14}$ . Tím se přivede  $\overline{PSEN}$  do log. 0. Piny sériové linky jsou převáděny na rozhraní RS-232 a RS-485. Zapojení těchto obvodů jsou popsány v části 4.4 .

Na port P2 je připojeno 8 galvanicky oddělených digitálních vstupů. Popis jejich funkce je popsán v části 4.5. Celý port P0 je ve čtecím režimu a je k němu připojeno 8 přepínačů ( $SW_1$ ), které slouží k nastavení 8 bitové adresy modulu. Pokud je kterýkoliv z přepínačů v levé poloze, na příslušném portu je log. 1. Přepínač nastavený na pravou polohu uzemňuje konkrétní vývod a je detekována log. 0.

Pro vnější reset procesoru je použit nulovací integrovaný obvod TL7705 ( $IC_3$ ). Tento obvod resetuje mikropočítač, jakmile napájecí napětí klesne pod hodnotu přibližně 4,5 V. Po opětovném zvýšení napětí nad 4,5 V přidržetí reset ještě po dobu zvolenou externím kondenzátorem [1]. Tento kondenzátor je připojen mezi vývod  $C_T$  a zem. Zvolená hodnota 100 nF odpovídá podle (2) času 1,3 ms.

$$t = C_T \cdot 13000 \quad [s] \quad (2)$$

t..... doba přidržetí resetu [s]

$C_T$ ..... kapacita kondenzátoru [F]

Mikroprocesor lze přivést do stavu reset také externím tlačítkem  $SW_3$ , které je zapojeno na vývod  $\overline{RESIN}$  obvodu TL7705.

#### 4.4 ROZHRANÍ RS-232 A RS-485

##### 4.4.1 Specifikace RS-232

RS-232 je rozhraní pro přenos informací, vytvořené původně pro komunikaci dvou zařízení do vzdálenosti 15 m. Aby byla vysílaná informace více odolná proti rušení, je přenášena větším napětím, než je klasických 5 V. Standard RS-232 pouze definuje, jak přenést určitou sekvenci bitů a nezabývá se už vyššími vrstvami komunikace. V referenčním modelu ISO/OSI tedy představuje fyzickou vrstvu [18].

##### 4.4.1.1 Napěťové úrovně

RS-232 používá dvě napěťové úrovně log. 1 a log. 0. Log. 1 je označována jako marking state (klidový stav), log. 0 se značí jako space state. Je-li napětí záporné, je detekována log. 1, zatímco log. 0 je přenášena kladným napětím. Povolené napěťové úrovně jsou uvedeny v Tab. 4.1 [18].

**Tab. 4.1 Datové signály RS-232**

Úroveň	Vysílač	Přijímač
Log. 0	+5 V až +15 V	+3 V až +25 V
Log. 1	-5 V až -15 V	-3 V až -25 V
Nedefinovaný stav	-3 V až +3 V	

##### 4.4.1.2 Parametry datového přenosu

Počet datových bitů je volitelný, běžně se používá 8 bitů, lze se také setkat se 7 a 9 bity. Základní tři vodiče rozhraní (příjem RxD, vysílání TxD a společná zem GND) mohou být doplněny ještě dalšími vodiči sloužícími k řízení přenosu. Pro zabezpečení přenosu dat se využívá parity. Ve vysílacím zařízení se sečte počet jedničkových bitů a doplní se paritním bitem tak, aby byla zachována předem dohodnutá podmínka sudého nebo lichého počtu jedničkových bitů (sudá parita – počet bitů v log. 1 + paritní bit = sudé číslo; lichá parita – počet bitů v log. 1 + paritní bit = liché číslo). Aby bylo možné detekovat ukončení datového rámce, používá se tzv. Stop bitu. Ten zároveň zajišťuje určitou prodlevu pro přijímač [18].

#### 4.4.1.3 Synchronizace dat

RS-232 používá asynchronní přenos informací. Každý přenesený Byte konstantní rychlostí je proto třeba synchronizovat. K synchronizaci se používá sestupná hrana tzv. Start bitu. Za ní již následují posílaná data [18].



Obr. 4.5 Synchronizace dat na RS-232

#### 4.4.2 Specifikace RS-485

RS-485 (ISO 8482) dvouvodičové polo-duplexní (komunikující současně pouze jedním směrem) sériové rozhraní, na které lze připojit až 32 jednotek na jedno vedení. Rychlost přenosu dat může být větší než 10 Mbit/s a délky vedení mohou dosáhnout asi 1200 m [4] [21].

##### 4.4.2.1 Přenos dat

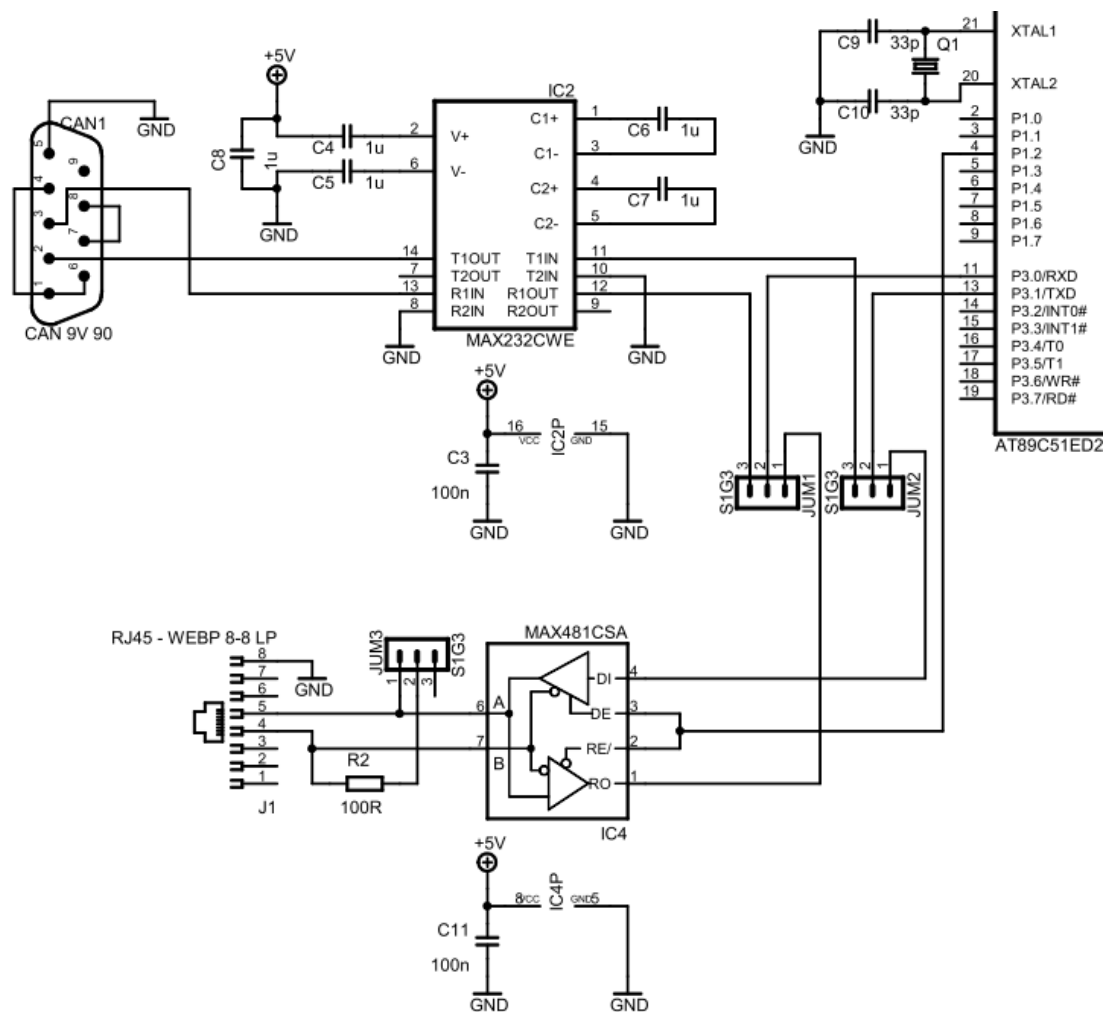
U rozhraní RS-485 se používá twisted pair vedení. Tímto principem se odstraní součtová (aditivní) rušení. Přenos je polo-duplexní, proto je třeba řídit směr komunikace. Aby se zabránilo odrazům na vedení, je nutné ho zakončit odporem (100  $\Omega$ ) na obou koncích. Každý obvod na sběrnici RS-485 musí mít podle standardu vstupní impedanci 12 k $\Omega$  [4] [21].

##### 4.4.2.2 Logické stavy

Vysílač i přijímač jsou sice spojeny zemnicím vodičem GND, ten ale není používán se pro určení logického stavu na vodičích. Logické stavy jsou totiž určeny diferencí mezi dvěma vodiči. Označí-li si jeden vodič jako A druhý jako B, pak je-li signál v log. 0, je rozdíl mezi napětím na A a B kladný (větší než +200 mV). Pokud

je signál v log. 1, je rozdíl mezi napětím na A a B záporný (menší než -200 mV).  
V klidovém stavu je na B vodiči +5 V a na vodiči A 0 V.

RS-485 definuje maximální napěťový rozsah na vstupu přijímače (rozdíl zemních potenciálů + alternující signálové napětí) od -7 V do +12 V [4] [21].



Obr. 4.6 Schéma zapojení rozhraní RS-232 a RS-485

#### 4.4.3 Popis zapojení rozhraní RS-232

Ze specifikace sběrnice RS-232 je zřejmé, že vývody sériové linky procesoru pracující s TTL úrovněmi musejí být převedeny na logické úrovně RS-232 a naopak. K tomu je použito integrovaného obvodu MAX232CWE (IC<sub>2</sub>). Jedná se o obousměrný převodník TTL logiky na logiku rozhraní RS-232. Obsahuje dvě

dvojice konvertorů napěťové úrovně. Dva jsou určeny pro převod RS-232 na TTL a dva pro převod opačným směrem. Obvod má pro převod na RS-232 zabudovaný napěťový násobič a invertor. Násobič používá kondenzátor  $C_6$ , připojený mezi vývody  $C_{1+}$  a  $C_{1-}$ , ke znásobení napětí +5 V na +10 V na kondenzátoru  $C_4$  (výstup V+) a kondenzátor  $C_7$ , připojený mezi vývody  $C_{2+}$  a  $C_{2-}$ , jako invertor k převrácení napětí +10 V na -10 V na kondenzátoru  $C_5$  (výstup V-). V tomto zapojení je použito pětice keramických kondenzátorů (hmota X7R) o hodnotě 1  $\mu\text{F}$  (doporučení výrobce) [7].

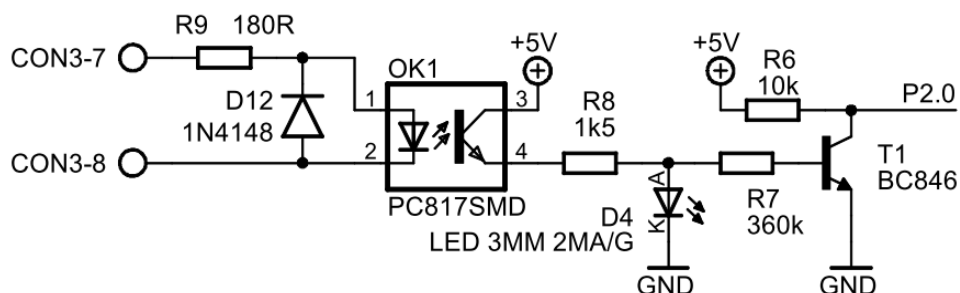
Mezi napájení obvodu a zem je opět zapojen filtrační kondenzátor  $C_3$  100 nF. Výstup vysílače T1OUT a vstup přijímače R1IN jsou připojeny na konektor CAN 9 na vývod 2, respektive 3. Pin č. 5 konektoru CAN 9 je přiveden na společnou zem. Výstup přijímače R1OUT je připojen na jumperový přepínač JUM<sub>1</sub>. Vstup vysílače T1IN je přiveden na jumperový přepínač JUM<sub>2</sub>. Tyto přepínače jsou použity k volbě právě používané sběrnice (RS-232 nebo RS-485). Jejich střední vývody jsou zapojeny na sériovou linku mikroprocesoru (RxD a TxD). Nezapojené vstupy obvodu MAX232CWE jsou uzemněny [7].

#### 4.4.4 Popis zapojení rozhraní RS-485

Pro převod informace ze sériové linky mikroprocesoru na rozhraní RS-485 a nazpět je použit integrovaný obvod LTC1485CS8 (IC<sub>4</sub>). Ten slouží k převodu standardu RS-485 na TTL logiku. Výstup vysílače TxD sériového kanálu je připojen přes přepínač mezi rozhraními RS-232 a RS-485 JUM<sub>2</sub> na vstupní pin DI (Driver Input). Přijímací vývod sériového kanálu mikroprocesoru RxD je spojen skrz jumperový přepínač JUM<sub>1</sub> s výstupním pinem RO (Receiver Output) převodníku. Pro volbu režimu přenosu (příjem či vysílání dat) je využito portu P1.2, propojeného s piny DE a  $\overline{\text{RE}}$ . Pokud je na portu P1.2 nastavena log. 1 je zvolen režim vysílání (obvod převádí TTL na RS-485), naopak při nastavení log. 0 se data přijímají (obvod převádí RS-485 na TTL). Vodiče A a B jsou vyvedeny na konektor (piny č. 4 a č. 5) RJ-45, který je také běžně používán pro sítě typu Ethernet. Pokud by tento modul byl koncovým zařízením, je třeba pomocí jumperového přepínače JUM<sub>3</sub> připojit

zakoňovací odpor  $R_2$  o hodnotě  $100 \Omega$ . Na vývod č. 8 konektoru RJ-45 je připojen zemnicí vodič. Opět je zde použit filtrační kondenzátor mezi napájením obvodu a zemí [11].

#### 4.5 GALVANICKÉ ODDĚLENÍ DIGITÁLNÍCH VSTUPŮ



**Obr. 4.7 Zapojení galvanického oddělení dig. vstupu č. 1**

##### 4.5.1 Popis zapojení galvanického oddělení digitálních vstupů

Každý z digitálních vstupů je galvanicky oddělen přes optočlen PC817SMD s izolačním napětím 5 kV. Ke vstupu optočlenu je zapojena paralelně univerzální dioda 1N4148, která plní funkci ochrany proti opačnému zapojení digitálního vstupu (přepólování). Jestliže bude na vstup optočlenu přivedena log. 1, na portu mikroprocesoru bude detekována log. 0 (otevřený tranzistor BC846 uzemní přes  $10 \text{ k}\Omega$  rezistor napájecí napětí) a opačně. K rozsvícení diody optočlenu je dle katalogového listu třeba proudu 20 mA. Když dioda svítí, je na ní úbytek napětí 1,2 V [10]. Pokud je uvažováno vstupní napětí +5 V (log. 1) je třeba přidat do série odpor o hodnotě  $180 \Omega$ . Na rezistoru je tedy úbytek napětí 3,8 V. Výpočet odporu znázorňuje (3).

$$R_V = \frac{U_{CC} - U_{DO}}{I_{DO}} = \frac{5V - 1,2V}{0,02A} = 190 \Omega \doteq 180 \Omega \quad (3)$$

$R_V$  ..... odpor na vstupu optočlenu [ $\Omega$ ]

$U_{CC}$  ..... napájecí napětí [V]

$U_{DO}$  ..... úbytek napětí na diodě optočlenu [V]

$I_{DO}$  ..... proud diodou optočlenu [A]

Indikace vstupu v log. 1 je vyřešena svítivou zelenou diodou s proudovým odběrem 2 mA a úbytkem napětí 1,9 V, zapojenou paralelně s tranzistorem BC846. Aby byl na diodě úbytek napětí 1,9 V, musí být v sérii s tranzistorem optočlenu zapojen 1,5 kΩ rezistor. Příklad výpočtu odporu je znázorněn v (4).

$$R_{SO} = \frac{U_{CC} - (U_D + U_{CEO})}{I_D + I_B} = \frac{5V - (1,9V + 0,1V)}{0,002A + 3,33 \cdot 10^{-6} A} = 1500 \Omega \quad (4)$$

$R_{SO}$  ..... odpor v sérii s tranzistorem optočlenu [ $\Omega$ ]

$U_{CC}$  ..... napájecí napětí [V]

$U_D$  ..... úbytek napětí na diodě [V]

$I_D$  ..... proud diodou [A]

$I_B$  ..... proud báze tranzistoru BC846 [A]

$U_{CEO}$  ..... úbytek napětí mezi kolektorem a emitorem tranzistoru v optočlenu [V]

Při otevřeném tranzistoru jím protéká proud maximálně 0,5 mA (úbytek na 10 kΩ rezistoru je 5 V). Pro proudový zesilovací činitel  $h_{fe} = 150$  je podle vzorce (5) třeba, aby do báze tranzistoru tekla proud 3,33  $\mu$ A.

$$I_B = \frac{I_C}{h_{fe}} = \frac{0,005A}{150} = 3,33 \mu A \quad (5)$$

$I_B$  ..... proud báze tranzistoru BC846 [A]

$I_C$  ..... proud kolektoru tranzistoru BC846 [A]

$h_{fe}$  ..... proudový zesilovací činitel [-]

Pokud je uvažován proud báze 3,33  $\mu$ A, je nutné zapojení odporu báze o velikosti 360 kΩ. Jeho hodnota je vypočítána podle (6).

$$R_B = \frac{U_{CC} - (U_{R_{SO}} + U_{BET} + U_{CEO})}{I_B} = \frac{5V - (3V + 0,7V + 0,1V)}{3,33 \cdot 10^{-6} A} = 360 k\Omega \quad (6)$$

$R_B$  ..... odpor báze tranzistoru BC846 [ $\Omega$ ]

$U_{CC}$  ..... napájecí napětí [V]

$U_{R_{So}}$  ..... úbytek napětí na odporu  $R_{So}$  [V]

$U_{BET}$  ..... úbytek napětí mezi bází a emitorem tranz. BC846 [V]

$U_{CEO}$  ..... úbytek napětí mezi kolektorem a emitorem tranzistoru  
v optočlenu [V]

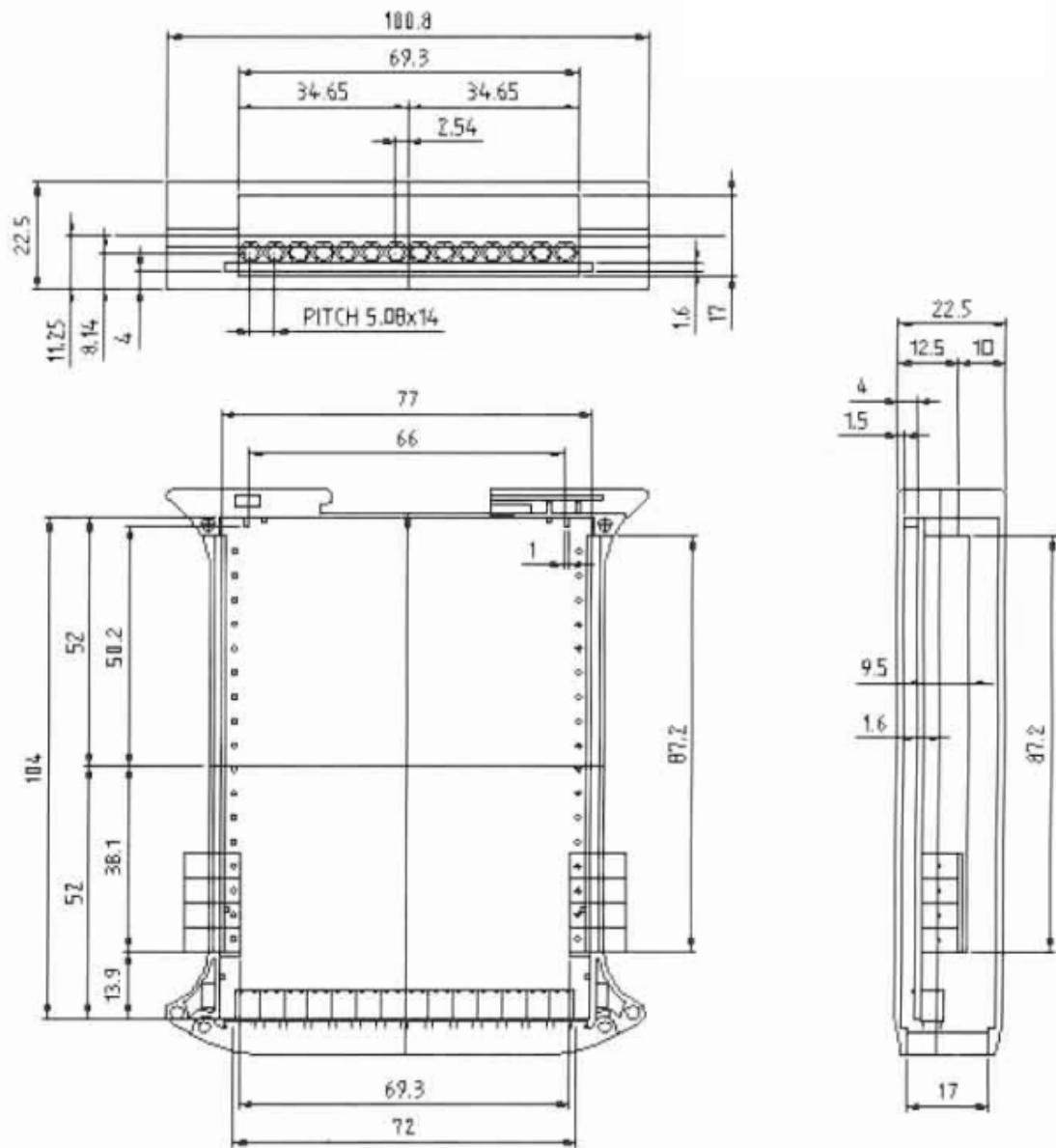
$I_B$  ..... proud báze tranzistoru BC846 [A]

#### 4.6 KONSTRUKČNÍ USPOŘÁDÁNÍ

Pro uložení desky plošných spojů (DPS) je použit černý přístrojový box (RAILBOX 22.5), který lze upevnit na standardní lištu DIN EN 50022. Hlavní výhodou této krabičky je rychlá a snadná montáž DPS bez spojovacích šroubů. Šířka boxu 22,5 mm umožňuje uložit DPS se součástkami vysokými až 15 mm. Indikační diody na desce jsou viditelné skrz transparentní víčko na přední straně boxu. Přístup ke konektorům a svorkovnicím umístěným na desce je možný z obou bočních stran krabičky, kde jsou k tomuto účelu vytvořeny otvory [12].



Obr. 4.8 Použitý box pro uložení DPS [12]



Obr. 4.9 Mechanický výkres krabičky RAILBOX 22.5 [12]

#### 4.7 DESKA PLOŠNÝCH SPOJŮ

Navržená deska plošných spojů je oboustranná a má velikost 104 x 77 mm. Výkresy DPS (pohled ze strany součástek, pohled ze strany spojů a osazovací výkres) jsou zobrazeny v příloze 3.

## 5. POPIS PROGRAMOVÉHO VYBAVENÍ

Naprogramovaný mikroprocesor AT89C51ED2 se chová jako Modbus slave jednotka, která komunikuje s master jednotkou pomocí sériové linky. V programu jsou obsaženy oba režimy komunikace přes sériovou linku definované protokolem (Modbus RTU a Modbus ASCII). Software je napsán v jazyce C pomocí prostředí KEIL  $\mu$ Vision3.

### 5.1 STRUKTURA PROGRAMU

#### Deklarace globálních proměnných

V této části je deklarována většina proměnných použitých v hlavním programu, funkcích a přerušeních. Pro přijímané a odesílané znaky jsou deklarovány pole o 256 položkách typu char. Tato velikost je plně dostačující pro RTU režim, kde je maximální počet přijatých či odesílaných znaků právě 256.

#### Přerušení od časovače 0

```
void timer0 (void) interrupt 1 using 1
```

Při tomto přerušení je navyšována proměnná  $t$ , která je určena pro počítání času mezi příchozími znaky a zprávami v Modbus RTU módu. Časovač 0 pracuje v módu 2 (8 bitový, přednastavený). Pro práci s proměnnými je zde využito první banky registrů.

#### Přerušení od časovače 2

```
void timer2 (void) interrupt 5 using 2
```

Toto přerušení, pokud je povoleno, nastává každých 50 ms a má pro každý režim jinou funkci.

V RTU módu slouží jako časovač, který určuje časové rozmezí mezi čtením stavu všech digitálních vstupů modulu. Všechny přečtené hodnoty jsou po dosažení definované hodnoty času uloženy do zásobníku stavu vstupů (kruhový zásobník).

V módu ASCII je časovač určen pro počítání času jedné sekundy mezi příchozími znaky. Pokud je čas překročen, nastaví se chyba v přijímané zprávě.

Ukázka programu:

```
void timer2 (void) interrupt 5 using 2
{
time++; // pri pretečení se navysí time o 1
/* rutina pro režim RTU */
if (ASCII == 0)
{
// jakmile time dosáhne hodnoty v registru (čas v ms) delene 50,
// podmínka je splněna
if (time == (registers[4]/50))
{
// pokud preteče buffer je citac nastaven zpět na nulu
// (kruhový buffer FIFO)
if (buffer_count == 30)
// nulování citace bufferu
buffer_count = 0;
// nactení hodnot všech vstupu do bufferu stavu dig. vstupu
buffer[buffer_count] = ~P2;
// nulování citace casu
time = 0;
// navýšení citace bufferu stavu digitalních vstupu o 1
buffer_count++;
}
}
/* rutina pro režim ASCII */
else
{
// pokud čas dosáhne 1 s, je nastavena chyba zpravy ASCII
if(time == 20)
{
time = 0; // nulování citace casu
error_frame = 1; // nastavení chyby ve zprávě
}
}
TF2 = 0; // nulování příznaku prer. od časovace 2
}
```

Časovač 2 pracuje v 16 bitovém přednastaveném režimu. Výpočet přednastavených hodnot pro TH2 a TL2 je znázorněn v (7). Pro práci s proměnnými je zde využito druhé banky registrů.

$$\begin{aligned}
 \text{RCAP2H}:\text{RCAP2L} &= 65536 - \frac{f_{osc}}{12} \cdot t = \\
 &= 65536 - \frac{11,0592 \cdot 10^6 \text{ Hz}}{12} \cdot 0,05 \text{ s} = 19456 = 0x4C00
 \end{aligned}
 \tag{7}$$

RCAP2H..... horních 8 bitů přednastavené hodnoty

RCAP2L..... spodních 8 bitů přednastavené hodnoty

$f_{osc}$  ..... frekvence oscilátoru [Hz]

$f_{pož}$  ..... požadovaný čas přetečení časovače – 50 ms [s]

### Přerušení od sériové linky

```
void serial_line (void) interrupt 4 using 3
```

U tohoto přerušení je nejdříve potřebné zjistit, zda bylo vyvoláno přijatým nebo odeslaným znakem. Následně je příslušný příznak vynulován.

Jestliže se jedná o příznak od přijatého znaku, pro oba režimy komunikace dojde k uložení znaku na příslušnou pozici v poli přijatých znaků a kontrole parity. Je kontrolován takový typ parity, který je aktuálně nastaven. Pokud je parita chybná, nastaví se chyba v přijímané zprávě. Dále je zde hlídáno, jestli nepřetekl čítač přijatých dat. Pro každý režim je mechanismus uložení znaku a kontroly parity různý. V ASCII módu jsou zde navíc kontrolovány různé stavy při příjmu znaků „:“, CR a LF podle stavového diagramu, který je znázorněn na Obr. 5.7.

V případě příznaku od odeslaného znaku je pro oba režimy nastavena proměnná `sent` na 1. Operand slouží jako indikace celého odeslaného znaku pro rutinu vykonávající odesílání zprávy. U ASCII módu je zde navíc kontrolován ukončovací znak LF. Pokud je tento znak detekován, je proměnná `sent` nastavena na 2. Pro rutinu vykonávající odesílání zprávy to znamená, že jsou odeslány všechny ASCII znaky.

Pro práci s proměnnými je zde využito třetí banky registrů.

### Funkce pro kontrolu LRC

```
unsigned char LRC (unsigned char *message, unsigned char  
data_len)
```

Funkce vrací programu vypočítanou 8 bitovou hodnotu LRC a pracuje se dvěma vstupními parametry – přijatá či odesílaná zpráva, délka zprávy.

Postup výpočtu LRC:

1. Součet všech znaků zprávy vyjma počátečního znaku „:“ a dvou konečných znaků CR a LF. Všechny překročení hodnoty 255 jsou ignorovány (byte je nastaven na nulu a pokračuje se ve výpočtu).
2. Celkový součet znaků je odečten od 0xFF.
3. K rozdílu je přičtena jednička – vznik dvojkového doplňku.
4. Funkce vrací hodnotu dvojkového doplňku [17].

### Funkce pro kontrolu CRC

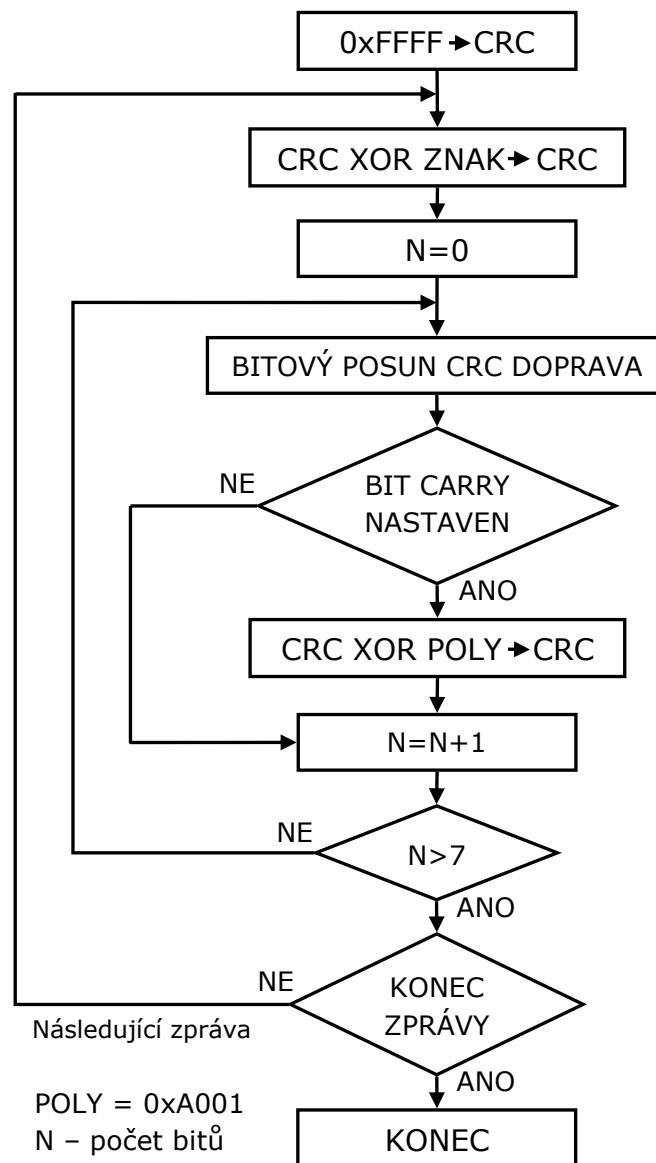
unsigned short **CRC16** (unsigned char \*message, unsigned char data\_len)

Funkce vrací program vypočítanou 16 bitovou hodnotu CRC a pracuje se dvěma vstupními parametry – přijatá či odesílaná zpráva, délka zprávy.

Postup generování CRC:

1. Do 16 bitového CRC registru se uloží hodnota 0xFFFF
2. Výsledek bitové logické funkce XOR mezi hodnotou 8 bitového znaku zprávy (bez parity, start a stop bitu) a CRC registru je uložen zpět do CRC registru. Pořadí znaku je určeno čítačem, který začíná čítat na prvním znaku a je postupně navyšován.
3. Hodnota CRC registru je bitově posunuta o jeden bit směrem k nejméně významnému bitu (LSB). Nejvýznamnější bit (MSB) je nastaven na log. 0.
4. Je kontrolován příznak podtečení (bit carry). Pokud byl LSB log. 0, bit carry není nastaven a je zopakován krok 3 (další posun). Pokud byl LSB log. 1, bit carry je nastaven a do CRC registru je uložen výsledek bitové logické funkce XOR mezi hodnotou CRC registru a CRC polynomem definovaným protokolem (0xA001)
5. Opakují se kroky 3 a 4, dokud není provedeno osm bitových posunů. Jakmile jsou provedeny všechny bitové posuny, je již celý znak zpracován a postupuje se ke kroku 6.
6. Opakují se kroky 2 až 5 pro další 8 bitový znak ze zprávy. Toto se opakuje, dokud nejsou všechny znaky zprávy zpracovány.
7. Konečný obsah CRC registru je hodnota CRC, kterou vrací funkce [17].

Celý postup zobrazuje vývojový diagram na Obr. 5.1.



Obr. 5.1 Vývojový diagram výpočtu CRC

### Funkce pro zápornou odpověď v RTU módu

```
void exception (unsigned char exception_code)
```

Funkce pracuje s jedním vstupním parametrem – kód chyby. Na druhou pozici v poli odesílaných dat je zde ke kódu funkce přičtena hodnota 0x80. Kód chyby je uložen jako následující odesílaný znak po kódu funkce. Nakonec je nastaven celkový počet odesílaných bytů zprávy, který je pro každou zápornou odpověď stejný (5 bytů).

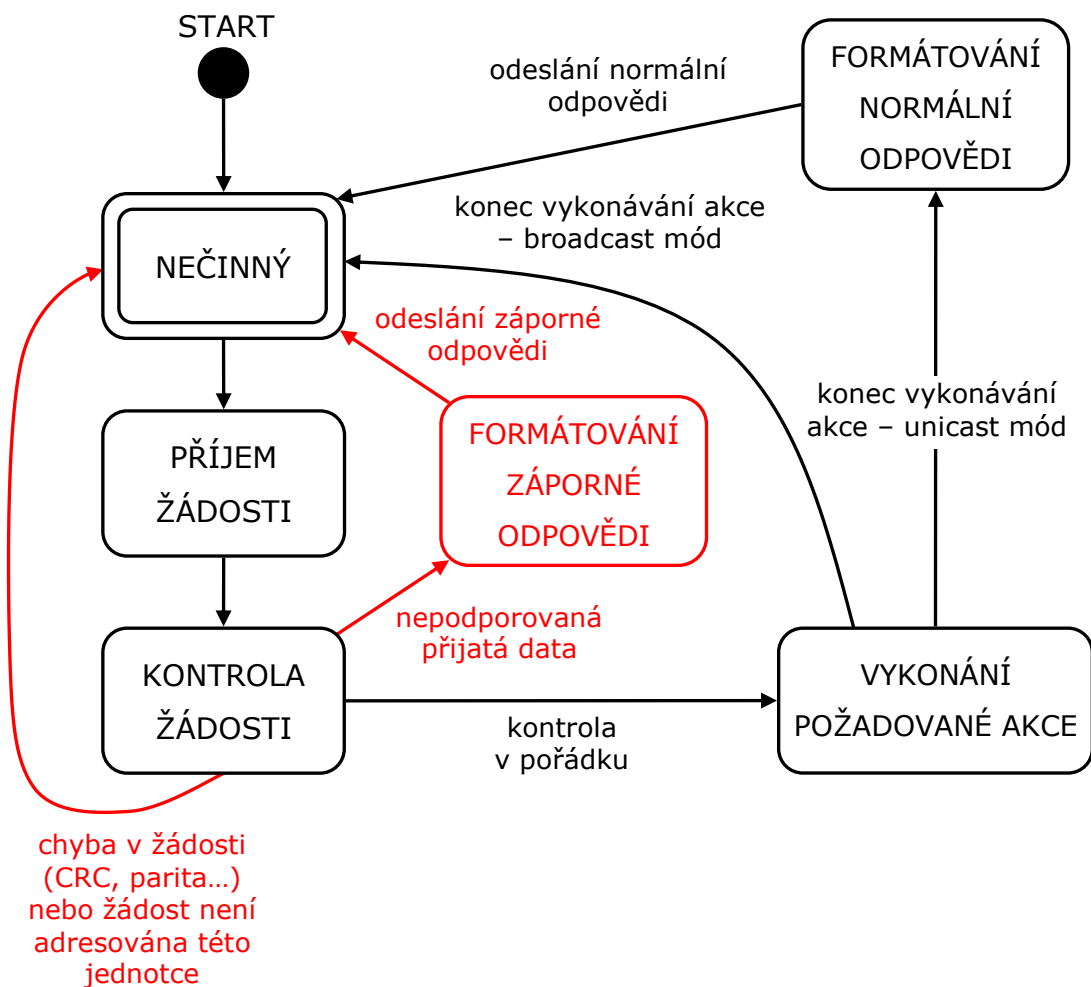
### Funkce pro zápornou odpověď v ASCII módu

```
void exception_ASCII (unsigned char exception_code)
```

Funkce pracuje s jedním vstupním parametrem – kód chyby. Na čtvrté a páté místo v poli odesílaných dat jsou zde ke dvěma znakům kódu funkce přičteny hodnoty 0x08 a 0x00. Kód chyby je uložen do následujících dvou odesílaných znaků. Nakonec je nastaven celkový počet odesílaných bytů zprávy, který je pro každou zápornou odpověď stejný (11 ASCII znaků).

### Hlavní program

```
void main (void)
```



Obr. 5.2 Stavový diagram pro slave jednotku

Celý hlavní program je naprogramován jako stavový automat, v němž jsou obsaženy principy zpracování požadavků od master jednotky pro oba podporované režimy Modbus RTU a Modbus ASCII. Obecně jsou pro slave jednotku jednotlivé stavy znázorněny na Obr. 5.2. Podrobným popisem komunikace pro mód RTU se zabývá kapitola 5.2. Mód ASCII je detailně popsán v kapitole 5.3.

Na začátku funkce main jsou nastaveny režimy třech časovačů a povoleny přerušeni od sériové linky, časovače 0 a časovače 2. Časovač 1 pracuje v módu 2 (osmibitový, přednastavený) a slouží pro definování komunikační rychlosti pro sériovou linku. Pro základní přenosovou rychlost 9600 b/s jsou nastaveny registry TH1 a TL1 na 253 (0xFD). Výpočet hodnoty pro TH1 a TL1 je ukázán v (8). Dále jsou deklarovány lokální proměnné. Většina proměnných je uvedena do počátečního stavu. Parita je po resetu nastavena na sudou (proměnná  $parity=2$ ) a komunikační rychlost na dvojnásobek základní – 19200 b/s (proměnná  $speed=1$ ).

$$TH1 = 256 - \frac{2^{SMOD} \cdot f_{osc}}{32 \cdot 12 \cdot v_{prenos}} = 256 - \frac{11,0592 \cdot 10^6 \text{ Hz}}{384 \cdot 9600 \frac{b}{s}} = 253 = 0xFD \quad (8)$$

SMOD..... nejvýznamnější bit v registru PCON

$f_{osc}$  ..... frekvence oscilátoru [Hz]

$v_{prenos}$  ..... požadovaná přenosová rychlost [b/s]

## 5.2 KOMUNIKACE POMOCÍ REŽIMU MODBUS RTU

Modbus RTU je definován jako hlavní režim komunikace. Po jakémkoliv resetu (hardwarový nebo softwarový uskutečněný Modbus funkcí) nebo při nastavení uchovávacího registru ASCII do nuly jednotka začíná přijímat a odesílat zprávy právě v tomto módu.

Po aktivaci RTU režimu je nastavena sériová linka do módu 3 (9 bitový UART) z důvodu, že je zde byte přenášen pomocí 11 bitů (viz. část 2.2.2.1). Dále je spuštěn časovač 1, povolen příjem dat a spuštěn časovač 0 sloužící pro počítání doby trvání příjmu 1,5 a 3,5 znaku. Časovač 0 pracuje v módu 2 (osmibitový,

přednastavený) a je nastaven tak, aby přetékal (vyvolal přerušení) s dvoutřetinovou frekvencí přenosové rychlosti. Pro čas 1,5 znaku musí být tedy vyvoláno 11 přerušení od časovače 0 a pro čas 3,5 znaku musí být počet přerušení 26. Výpočet hodnoty registrů TH0 a TL0 pro rychlost 19200 b/s je znázorněn v (9).

$$TH0 = 256 - \frac{f_{osc}}{12 \cdot f_{pož}} = 256 - \frac{11,0592 \cdot 10^6 \text{ Hz}}{12 \cdot 12800 \text{ Hz}} = 184 = 0xB8 \quad (9)$$

$f_{osc}$  ..... frekvence oscilátoru [Hz]

$f_{pož}$  ..... požadovaná frekvence – dvě třetiny přenosové rychlosti [Hz]

### 5.2.1 Popis jednotlivých stavů v RTU módu

Celý postup příjmu, zpracování a odesílání zpráv definuje stavový diagram na Obr. 5.3. Tento diagram je mírně modifikován od originálního uvedeného v Modbus specifikaci [17].

#### 5.2.1.1 Počáteční stav

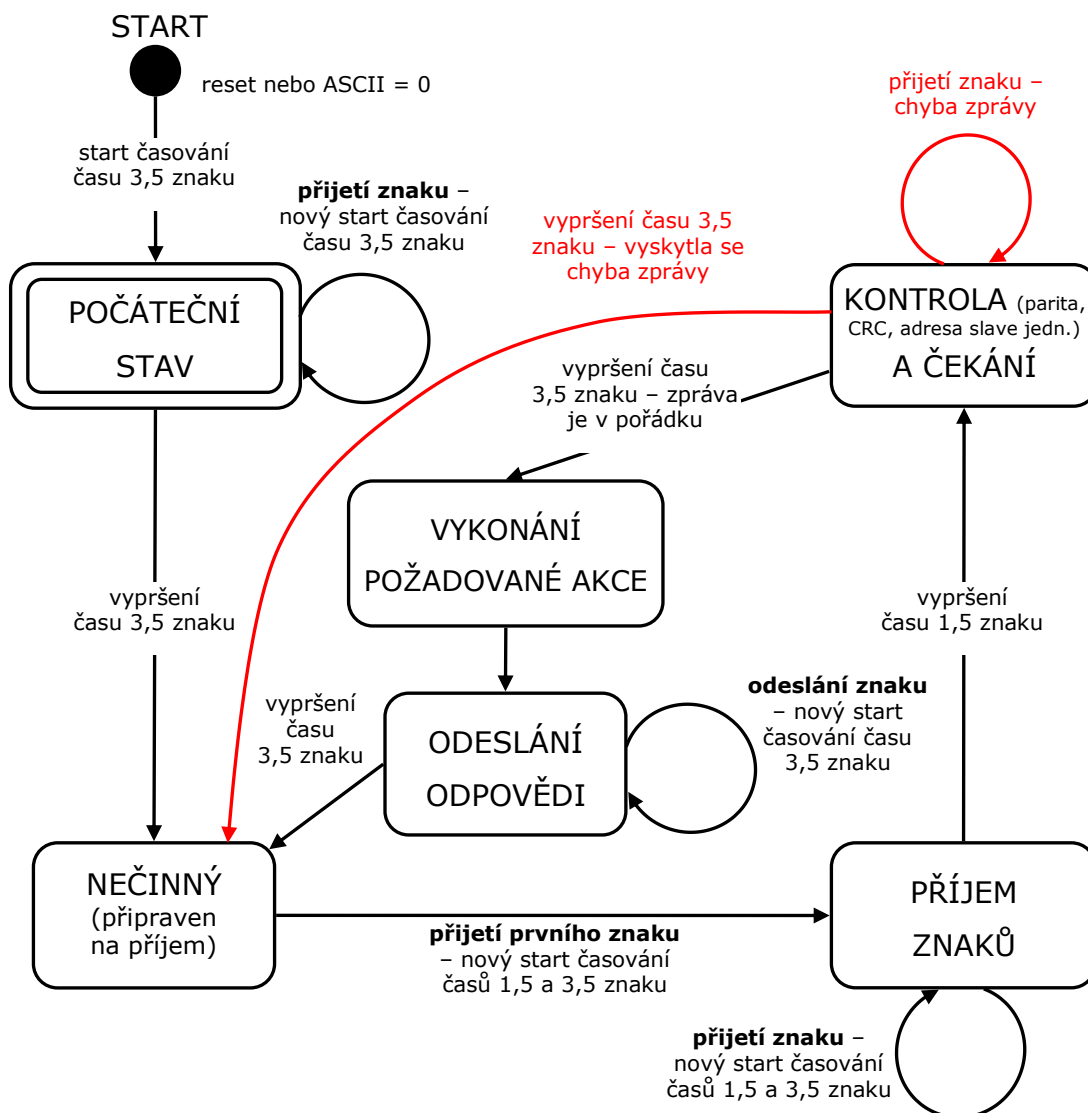
Jednotka je po zapnutí minimálně po dobu trvání příjmu 3,5 znaku v počátečním stavu. Pokud je přijat během této doby nějaký znak, je počítání času 3,5 znaku restartováno. Po uplynutí času přechází modul do stavu „nečinný“.

#### 5.2.1.2 Stav „Nečinný“

V tomto stavu je podle určené rychlosti přenosu dat uložena do registru TH0 příslušná hodnota pro časování doby 1,5 a 3,5 znaku a bit SMOD je nastaven na log. 1 nebo log. 0. Také jsou zde nulovány některé proměnné a všechny přijaté a odeslané znaky.

Podle negované hodnoty portu P0, na který jsou připojeny přepínače, je při každém přechodu do nečinného stavu nastavena adresa modulu. Pro negovanou hodnotu vyšší než 247 nastavena je adresa 247 a pro hodnotu nula je nastavena adresa 1. Toto přiřazení adres je nutné, protože Modbus slave jednotky mají povolený rozsah adres 1 až 247.

Po provedení jmenovaných úkonů přechází mikroprocesor do režimu nízké spotřeby (nastavení bitu IDLE v registru PCON). Zpět do standardního módu se mikropočítač dostane při jakémkoliv přerušení. Jestliže jde o přerušení od přijatého znaku, přechází jednotka do stavu „příjem znaků“.



**Obr. 5.3 Stavový diagram pro režim Modbus RTU**

### 5.2.1.3 Stav „Příjem znaků“

V této části probíhá příjem celé zprávy a je zde počítán čas 1,5 znaku. Jakmile tento čas vyprší, předpokládá se, že je celý požadavek přijat a modul přechází do stavu „kontrola a čekání“

#### **5.2.1.4 Stav „Kontrola a čekání“**

Nejdříve je zkontrolováno, jestli je délka zprávy kratší než čtyři znaky. V případě, že je délka zprávy kratší, vznikla chyba při přenosu. Poté se čeká, dokud neuplyne čas 3,5 znaku. Pokud je během této doby přijat další znak, jedná se také o chybu. Po dosažení času 3,5 znaku je kontrolováno CRC a je zjišťováno, jestli je zpráva adresována tomuto modulu (přímá adresace nebo broadcast). Jestliže je během jakékoliv kontroly nalezena chyba (včetně ověření parity), přechází jednotka do stavu „nečinný“. Proběhne-li kontrola bez chyb, modul začne vykonávat požadovanou akci (přechod do stavu „vykonání požadované akce“).

#### **5.2.1.5 Stav „Vykonání požadované akce“**

Prvním odesílaným znakem je u jakékoliv odpovědi adresa modulu. Na začátku zpracování požadavku je tedy do odesílané adresy nakopírována adresa přijatá.

Podle přijatého kódu funkce je vykonán příslušný požadavek. Detailním popisem všech podporovaných funkcí v RTU módu se zabývá kapitola 5.2.2. Pokud jednotka funkci nezná je odesílána záporná odpověď s chybovým kódem č. 1 (ilegální funkce). U většiny podporovaných funkcí je kontrolováno, jestli odpovídá délka přijaté zprávy délce uvedené v popisu funkce podle Modbus protokolu. Jsou-li délky rozdílné, je odesílána záporná odpověď s chybovým kódem č. 3 (ilegální hodnota dat). Poté je zjišťováno, jestli se nachází adresa dat v definovaném rozsahu. Pokud není adresa platná, odesílá se záporná odpověď s chybovým kódem č. 2 (ilegální adresa dat). Nakonec před samotným zpracováním probíhá kontrola na platnou hodnotu dat. V případě, že jsou data mimo rozsah, odesílá se záporná odpověď s chybovým kódem č. 3 (ilegální hodnota dat).

Na poslední a předposlední místo v odesílané zprávě je uloženo vypočítané CRC. Po vložení CRC je již celá zpráva zpracována a je připravena k odeslání. Jde-li o adresaci typu broadcast nebo je nastaven mód „passive“ (proměnná `passive = 1`), odpověď není odesílána a modul přechází rovnou do stavu „nečinný“. Jestliže je zpráva přímo adresována jednotce, začíná odesílání odpovědi (přechod do stavu „odeslání odpovědi“).

### 5.2.1.6 Stav „Odeslání odpovědi“

V tomto stavu jsou postupně odesílány všechny znaky zprávy. Ke znakům je zde doplňována právě nastavená parita. Detekce všech odeslaných zpráv je realizována pomocí časování doby trvání příjmu 3,5 znaku. Jakmile tento čas vyprší, je celá odpověď odeslána a modul přechází do stavu „nečinný“.

## 5.2.2 Podporované kódy funkcí pro RTU mód

### 5.2.2.1 Čti diskrétní vstupy (*Read Discrete Inputs*) – 02 (0x02)

Primárním úkolem navrženého modulu je čtení stavu implementovaných digitálních vstupů. K tomu je určena tato funkce, která podle specifikace umožňuje číst stav 1 až 2000 diskrétních vstupů. Pro účely použité jednotky má funkce omezení pro čtení maximálně osmi stavů.

V požadavku je definována adresa prvního vstupu a počet čtených vstupů. V odpovědi je upřesněn počet bytů, protože do jednoho bytu je možné uložit stav maximálně 8 vstupů. Stav je indikován jako log. 1 pro aktivní vstup a log. 0 pro vstup neaktivní. Nejméně významný bit (LSB) prvního bytu je stav prvního (adresovaného) vstupu. Pokud není počet požadovaných vstupů dělitelný osmi beze zbytku, zbylá část nejvýznamnějšího bytu je směrem k MSB doplněna nulami [16].

**Tab. 5.1 Požadavek funkce 0x02**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x02
Počáteční adresa	2 byty	0x00 až 0x08
Počet vstupů	2 byty	1 až 8

**Tab. 5.2 Odpověď funkce 0x02**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x02
Počet bytů	1 byte	0x01
Stav vstupů	1 byte	

**Tab. 5.3 Záporná odpověď (chyba) funkce 0x02**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x82
Chybový kód	1 byte	0x02, 0x03

### 5.2.2.2 Čti uchovávací registry (Read Holding Registers) – 03 (0x03)

Tato funkce slouží podle definice v protokolu ke čtení obsahu souvislého bloku až 125 uchovávacích registrů. V navržené jednotce je použito pouze 36 registrů. Lze tedy žádat o maximálně 36 hodnot registrů.

V požadavku je definována adresa prvního registru a počet registrů. V odpovědi odpovídá každému registru dvojice bytů [16].

**Tab. 5.4 Požadavek funkce 0x03**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x03
Počáteční adresa	2 byty	0x00 až 0x08
Počet registrů	2 byty	1 až 36 (0x24)

**Tab. 5.5 Odpověď funkce 0x03**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x03
Počet bytů	1 byte	2 x N
Hodnoty registrů	N x 2 byty	

N – počet registrů

**Tab. 5.6 Záporná odpověď (chyba) funkce 0x03**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x83
Chybový kód	1 byte	0x02, 0x03

Význam registrů a zapsaných hodnot popisuje Tab. 5.7.

Tab. 5.7 Význam uchovacích registrů

Adresa	Význam registru	Hodnota	Význam hodnoty
40000 (0x00)	Režim komunikace	0	Mód RTU
		1	Mód ASCII
40001 (0x01)	Komunikační rychlost	9600	9600 b/s
		19200	19200 b/s
40002 (0x02)	Parita	0	Bez parity
		1	Lichá parita
		2	Sudá parita
40003 (0x03)	Zapnutý / vypnutý záznam stavů digitálních vstupů	0	Vypnuto
		1	Zapnuto
40004 (0x04)	Čas mezi jednotlivými záznamy stavů dig. vstupů	50 až 65500	Čas v ms (dělitelný beze zbytku 50)
40005 (0x05)	Nulování registrů pro záznam stavů dig. vstupů	Pro nulování je nutné zapsat hodnotu 0xFF	
40006 až 40035 (0x06 až 0x23)	Registry pro záznam stavů dig. vstupů	0000000b až 11111111b	Stavy všech dig. vstupů (LSB je vstup č.1)

Velkou výhodou této jednotky je možnost čtení historie stavů digitálních vstupů. Tento záznam je aktivní, je-li v registru na adrese 0x03 uložena hodnota 1.

V registru na adrese 0x06 je uložena, je-li zapnut záznam digitálních vstupů, vždy nejaktuálnější hodnota jednotlivých stavů. Čím vyšší je adresa registru (až po adresu 0x23), tím je hodnota stavů starší. Jelikož je zásobník, do kterého je ukládán záznam stavů, kruhový, je nutné jeho obsah přenést do registrů pomocí níže popsaného algoritmu převodu.

Část programu pro převod:

```
// pokud je již sepnuto časování je prováděn zápis z bufferu stavu vstupu
do příslušných registrů
if (registers[3] == 1)
{
    // pokud je počáteční adresa registru menší než 6
    if (R_DATA[3]<6)

    // cykluje až do počtu poz. registru + nastavené adresy - 6 (registry
stavu vstupu začínají až na 6)
for (aux = 0; aux < (R_DATA[5] - 6 + R_DATA[3]); aux++)
{
```

```

// na prislusne pozice jsou nahrany hodnoty bufferu - na prvni
// pozici je posledni ctena hodnota...
registers [6 + aux] = buffer[buffer_count - aux - 1 + aux2];

// pokud podtece citac bufferu, je prictena k aktualni pozici
// v bufferu hodnota 30
if ((buffer_count - aux - 1 + aux2) == 0)
    // nahrani teto hodnoty do pomocne promenne 2
    aux2 = 30;
}
// pokud je pocatecni adresa registru vetsi nez 6
if (R_DATA[3]>=6)

// cykluje az do poctu poz. registru + nastavene adresy - 6 (registry
// stavu vstupu zacinaji az na 6)
for (aux = (R_DATA[3] - 6); aux < (R_DATA[5] - 6 + R_DATA[3]); aux++)
{
    // pokud podtece citac bufferu, je prictena k aktualni pozici
    // v bufferu hodnota 30
    if ((buffer_count - aux + aux2) <= 0)
        // nahrani teto hodnoty do pomocne promenne 2
        aux2 = aux2 + 30;
    // na prislusne pozice jsou nahrany hodnoty bufferu - na prvni
    // pozici je posledni ctena hodnota...
    registers [6 + aux] = buffer[buffer_count - aux - 1 + aux2];
}
}

```

### 5.2.2.3 Zapiš jeden registr (Write Single Register) – 06 (0x06)

Úkolem této funkce je uložení definované hodnoty do jednoho uchovávacího registru. V požadavku je specifikována adresa registru určeného pro zápis a zapisovaná hodnota. Standardní odpovědí je kopie požadavku, která je vrácena, jakmile je do registru proveden zápis [16].

**Tab. 5.8 Požadavek funkce 0x06**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x06
Adresa registru	2 byty	0x00 až 0x35
Hodnota registru	2 byty	Podle Tab. 5.7

**Tab. 5.9 Odpověď funkce 0x06**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x06
Adresa registru	2 byty	0x00 až 0x35
Hodnota registru	2 byty	Podle Tab. 5.7

**Tab. 5.10 Záporná odpověď (chyba) funkce 0x06**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x86
Chybový kód	1 byte	0x02, 0x03

Pomocí této funkce lze nastavovat hodnoty jednotlivých registrů uvedené v Tab. 5.7. Jakákoliv jiná zapisovaná hodnota je ignorována a je odesílána záporná odpověď s chybovým kódem č. 3 (ilegální hodnota dat).

V modulu je tedy možné volit režim komunikace – Modbus RTU nebo Modbus ASCII, typ použité parity – sudá, lichá nebo žádná a komunikační rychlost – 9600 b/s nebo 19200 b/s. Další možností je zapnutí a vypnutí záznamu stavů digitálních vstupů, nulování těchto záznamů a nadefinování času v milisekundách mezi jednotlivými uloženými stavy (povolené jsou jen hodnoty dělitelné padesáti). Pro nulování všech uložených stavů je nutné poslat požadavek s hodnotou registru 0xFF.

#### **5.2.2.4 Diagnostika (Diagnostics) – 08 (0x08)**

Tato funkce slouží k provedení testu sloužícímu pro kontrolu komunikace mezi Master a Slave jednotkami, ke kontrole záznamů různých interních chybových stavů, nulování záznamů chybových stavů, k restartu komunikace a k nastavení pasivního režimu (modul neodesílá odpovědi). Funkce používá dvoubajtový kód podfunkce, který určuje požadovaný typ testu. Normální odpověď obsahuje kopii požadavku nebo data obsahující informaci o požadovaném chybovém stavu. V Tab. 5.14 jsou uvedeny tímto modulem podporované kódy podfunkce. Pokud obsahuje požadavek nepodporovanou podfunkci je odeslána záporná odpověď s chybovým kódem č. 1 (ilegální funkce) [16].

**Tab. 5.11 Požadavek funkce 0x08**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x08
Podfunkce	2 byty	Viz. Tab. 5.14
Data	N x 2 byty	

**Tab. 5.12 Odpověď funkce 0x08**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x08
Podfunkce	2 byty	Viz. Tab. 5.14
Data	N x 2 byty	

**Tab. 5.13 Záporná odpověď (chyba) funkce 0x08**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x88
Chybový kód	1 byte	0x01, 0x03

**Tab. 5.14 Podporované kódy podfunkcí pro funkci 0x08**

Kód podfunkce		Název
Hex	Dec	
0x00	00	Vrať data požadavku
0x01	01	Restartuj komunikaci
0x04	04	Přejdi do pasivního režimu
0x0A	10	Vynuluj čítače a diagnostický registr
0x0B	11	Vrať počet zpráv – CPT1
0x0C	12	Vrať počet komunikačních chyb – CPT2
0x0D	13	Vrať počet negativních odpovědí – CPT3
0x0E	14	Vrať počet zpracovaných zpráv – CPT4
0x0F	15	Vrať počet nezodpovězených zpráv – CPT5
0x12	18	Vrať počet ztracených znaků (zpráv) – CPT8
0x14	20	Vynuluj čítač ztracených znaků (zpráv)

### Vlastnosti podporovaných podfunkcí

*Vrať data požadavku 0x00* – v odpovědi je vrácena přesná kopie požadavku (kontrola komunikace).

- Hodnota dat v požadavku – jakékoliv data
- Hodnota dat v odpovědi – kopie dat požadavku

*Restartuj komunikaci 0x01* – pomocí této subfunkce je proveden softwarový reset programu. Restart je proveden pouze, obsahují-li data požadavku hodnotu 0x0000 nebo 0xFF00. Při hodnotě 0xFF00 je také vymazán čítač komunikačních událostí, jinak je jeho stav ponechán nezměněn. Před samotným restartem je ještě odeslána odpověď s kopií přijaté zprávy.

*Přejdi do pasivního režimu 0x04* – je-li v datech požadavku uložena hodnota 0x0000, uvede modul do módu pouhého příjmu zpráv (nejsou odesílány žádné odpovědi). Není odeslána ani odpověď na tuto zprávu.

*Vynuluj čítače a diagnostický registr 0x0A* – obsahují-li data požadavku hodnotu 0x0000, jsou smazány všechny čítače interních chyb. V odpovědi je odeslána kopie přijaté zprávy.

*Vrať počet zpráv 0x0B* – je-li v datech požadavku uložena hodnota 0x0000, v datové části odpovědi je odeslán čítač počtu zpráv (CPT1).

*Vrať počet komunikačních chyb 0x0C* – obsahují-li data požadavku hodnotu 0x0000, v datové části odpovědi je odeslán čítač počtu komunikačních chyb (CPT2).

*Vrať počet negativních odpovědí v 0x0D* – je-li v datech požadavku uložena hodnota 0x0000, v datové části odpovědi je odeslán čítač počtu negativních odpovědí (CPT3).

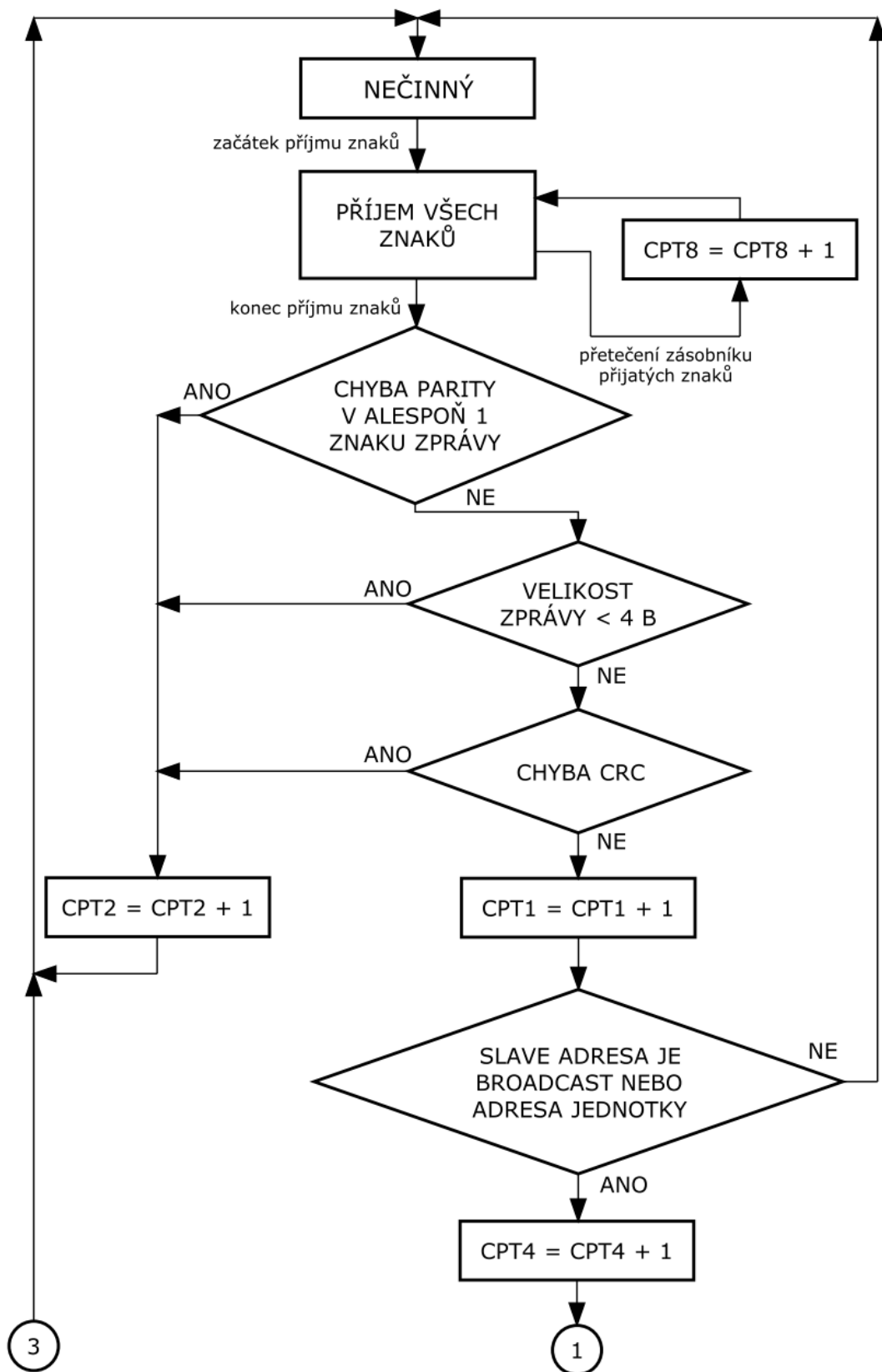
*Vrať počet zpracovaných zpráv 0x0E* – obsahují-li data požadavku hodnotu 0x0000, v datové části odpovědi je odeslán čítač počtu zpracovaných zpráv (CPT4).

*Vrať počet nezodpovězených zpráv 0x0F* – je-li v datech požadavku uložena hodnota 0x0000, v datové části odpovědi je odeslán čítač počtu nezodpov. zpráv (CPT5).

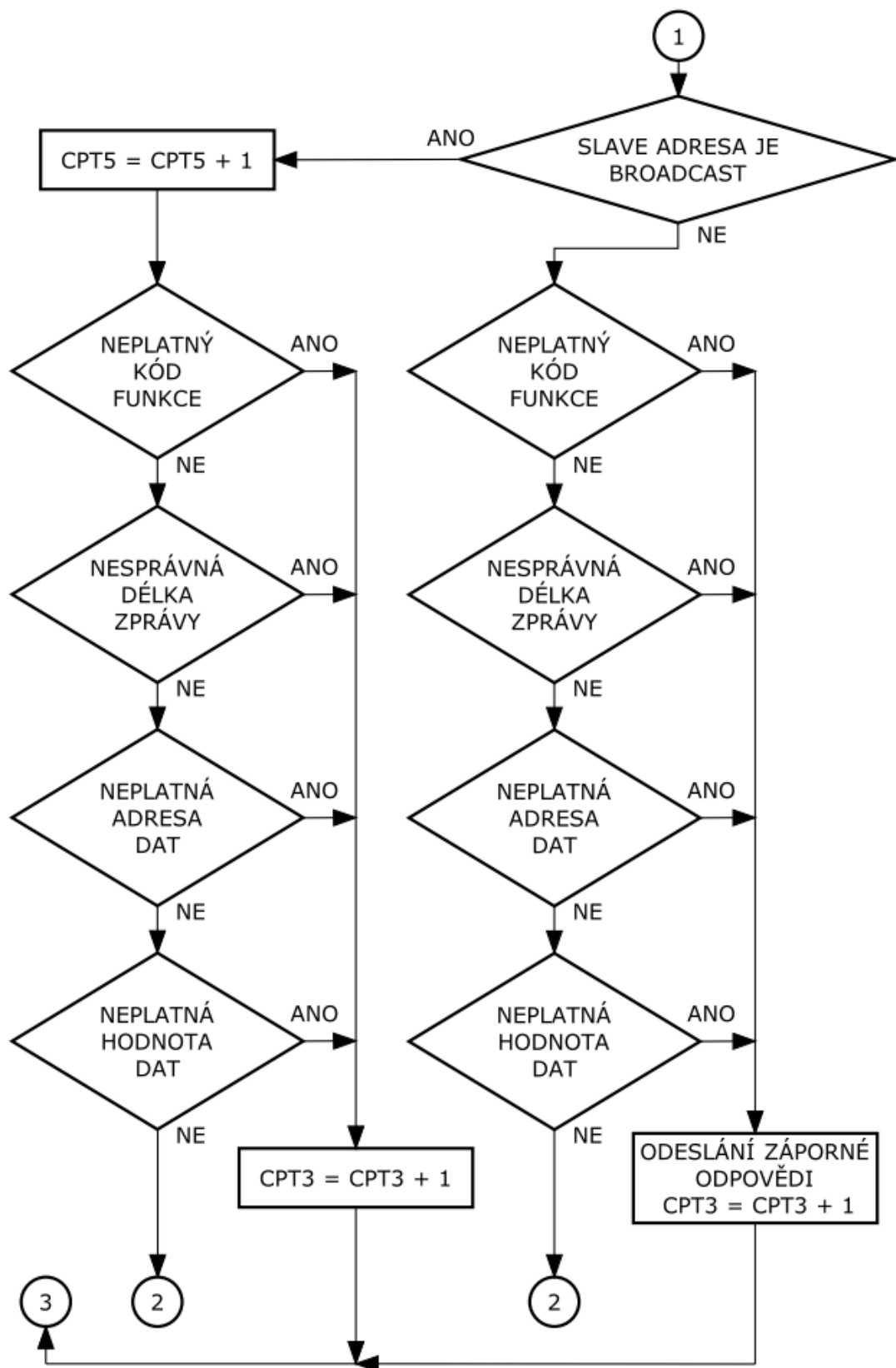
*Vrať počet ztracených znaků 0x12* – obsahují-li data požadavku hodnotu 0x0000, v datové části odpovědi je odeslán čítač počtu ztracených znaků (CPT8).

*Vynuluj počet ztracených znaků 0x14* – je-li v datech požadavku uložena hodnota 0x0000, je smazán čítač počtu ztracených znaků. V odpovědi je odeslána kopie přijaté zprávy [16].

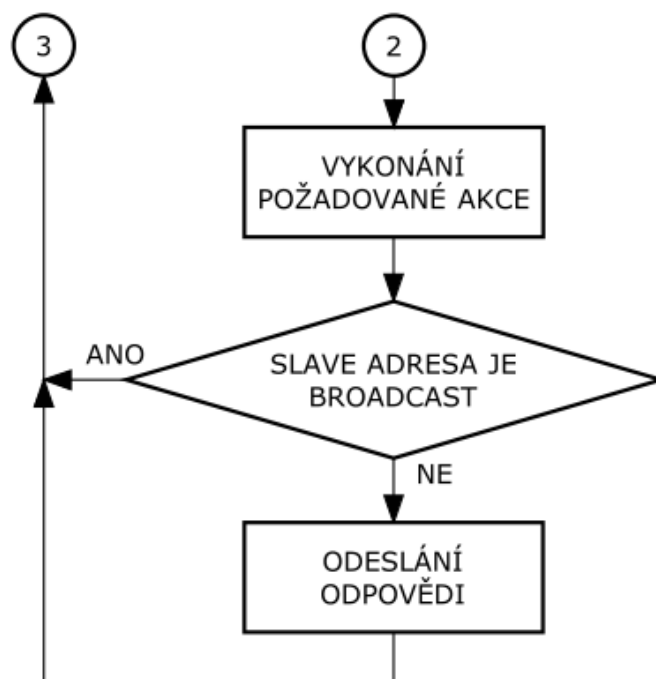
Vlastnosti čítačů (16 bitové) jsou patrné z vývojového digramu, který definuje, v jakých částech programu jsou navyšovány (Obr. 5.4, Obr. 5.5 a Obr. 5.6).



Obr. 5.4 Vývojový diagram režimu RTU a správy čítačů – 1. část



Obr. 5.5 Vývojový diagram režimu RTU a správy čítačů – 2. část



Obr. 5.6 Vývojový diagram režimu RTU a správy čítačů – 3. část

#### 5.2.2.5 Čti čítač komunikačních událostí (*Get Comm Event Counter*) – *11 (0x0B)*

Pomocí této funkce je možné získat stavové slovo a hodnotu čítače komunikačních událostí. Čítač událostí je inkrementován po každém úspěšném dokončení požadavku. Standardní odpověď obsahuje dvoubajtové stavové slovo a dvoubajtový počet událostí. Jelikož nemůže být jednotka nikdy zaneprázdněna, v každé odpovědi je hodnota stavového slova 0x0000 [16].

Tab. 5.15 Požadavek funkce 0x0B

Význam	Délka	Hodnota
Kód funkce	1 byte	0x0B

Tab. 5.16 Odpověď funkce 0x0B

Význam	Délka	Hodnota
Kód funkce	1 byte	0x0B
Stavové slovo	2 byty	0x0000
Čítač událostí	2 byty	0x0000 až 0xFFFF

**Tab. 5.17 Záporná odpověď (chyba) funkce 0x0B**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x8B
Chybový kód	1 byte	0x03

**5.2.2.6 Čti identifikaci zařízení (Read Device Identification) –  
(0x2B / 0x0E)**

Čtení identifikace a dalších údajů upřesňujících popis zařízení umožňuje funkce 0x2B / 0x0E. Identifikace zařízení je rozdělena na tři kategorie objektů.

- Základní identifikace zařízení (povinná)
- Obvyklá identifikace zařízení (nepovinná)
- Rozšířená identifikace zařízení (nepovinná)

Detaily k této funkci jsou k dispozici v [16]. V tomto modulu odesílá funkce v odpovědi pouze objekty ze základní kategorie. Jejich vlastnosti jsou uvedeny v Tab. 5.18.

**Tab. 5.18 Použité objekty funkce 0x2B**

Kategorie	ID objektu	Název objektu / popis	Typ
Základní (ID kód 0x01)	0x00	Název výrobce	ASCII řetězec
	0x01	Kód produktu	ASCII řetězec
	0x02	Revize	ASCII řetězec

V ID objektu 0x00 (název výrobce) je odesílán řetězec „VUT Brno“, v ID 0x01 (kód produktu) řetězec „Modbus\_8\_DIG\_IN“ a v ID 0x02 (revize) řetězec „V1.21“.

**Tab. 5.19 Požadavek funkce 0x2B**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x2B
Typ MEI	1 byte	0x0E
ID kód	1 byte	0x01
ID objektu	1 byte	0x00

**Tab. 5.20 Odpověď funkce 0x2B**

Význam	Délka	Hodnota
Kód funkce	1 byte	0x2B
Typ MEI	1 byte	0x0E
ID kód	1 byte	0x01
Úroveň shody	1 byte	0x01
Pokračování	1 byte	0x00
ID dalšího objektu	1 byte	0x00
Počet objektů	1 byte	0x03
Seznam objektů		
ID objektu	1 byte	
Délka objektu	1 byte	N
Hodnota objektu	N	Závisí na objektu

**Tab. 5.21 Záporná odpověď (chyba) funkce 0x2B**

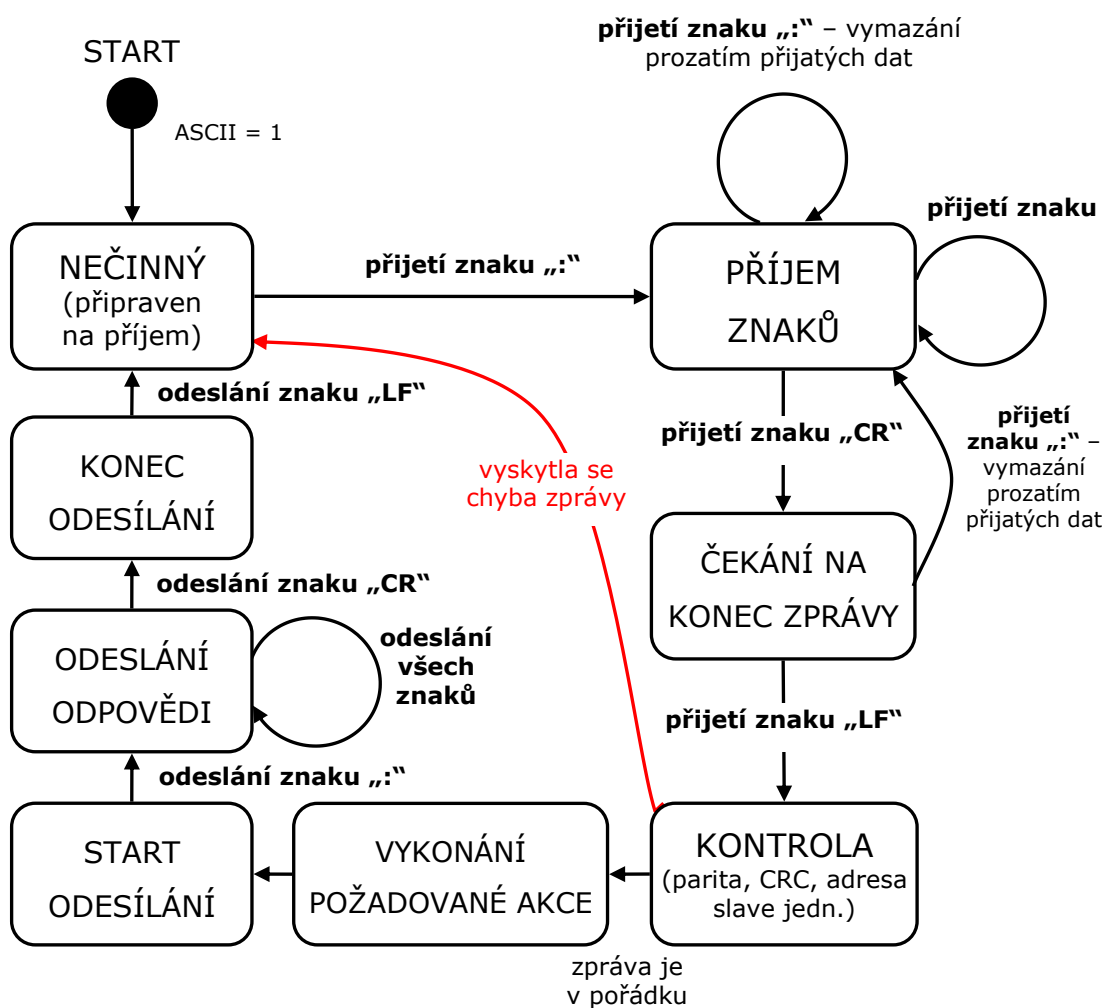
Význam	Délka	Hodnota
Kód funkce	1 byte	0xAB
Chybový kód	1 byte	0x01, 0x02, 0x03

### 5.3 KOMUNIKACE POMOCÍ REŽIMU MODBUS ASCII

Modbus ASCII slouží jako doplňkový mód k Modbus RTU. Jeho aktivace probíhá pouze pomocí zápisu hodnoty 1 do prvního uchovávacího registru (adresa 0x00). Princip komunikace v tomto režimu je znázorněn stavovým diagramem na Obr. 5.7 (upravený diagram z [17]).

Hlavním úkolem modulu je odesílání zpráv se stavy digitálních vstupů. Protože je ASCII módem sekundárním, jsou v něm zahrnuty pouze dvě Modbus funkce, které plně dostačují ke splnění primárního úkolu. Jde o kódy funkcí 0x02 (Čti diskretní vstupy) a 0x06 (Zapiš jeden registr). Kód 0x06 má oproti možným zápisům do uchovávacích registrů v RTU omezení pouze na modifikaci registru prvního. Jeho přepsáním na hodnotu nula je umožněn návrat do RTU režimu bez restartu. Další variantou aktivace RTU je hardwarový reset.

Po aktivaci ASCII režimu je nastavena sériová linka do módu 2 (8 bitový UART), jelikož je zde byte přenášen jako 7 bitový ASCII znak a 1 bitová parita (viz. kapitola 2.2.2.2). Následně je spuštěn časovač 1, časovač 2 a je povolen příjem dat. Typ parity a komunikační rychlost jsou ponechány na poslední nastavené hodnotě. Je-li zapnutý záznam stavů digitálních vstupů, při přechodu do ASCII módu je deaktivován (zastavení běhu časovače 2).



Obr. 5.7 Stavový diagram pro režim Modbus ASCII

### 5.3.1 Popis komunikačních stavů v ASCII módu

Jakmile je dokončeno nastavení sériové linky a časovačů, modul přechází do stavu „nečinný“.

### **5.3.1.1 Stav „Nečinný“**

Zde jsou prováděny stejné úkony jako ve stavu „nečinný“ v RTU režimu (viz. kapitola 5.2.1.6), pouze není spouštěn a nastavován časovač 0 pro počítání času 1,5 a 3,5 znaku. Poté, co je přijat znak „:“, dojde k přesunu do stavu „příjem znaků“. Jakýkoliv jiný přijatý znak je ignorován.

### **5.3.1.2 Stav „Příjem znaků“**

V tomto stavu je postupně přijímána celá zpráva. Je-li v tomto stavu přijat znak „:“, všechna dosud přijatá data jsou vymazána. Příjem po vynulování dat však pokračuje. Maximální doba mezi dvěma příchozími znaky je jedna sekunda (zajišťuje časovač 2). V případě příjmu znaku CR přechází modul do stavu „čekání na konec zprávy“.

### **5.3.1.3 Stav „Čekání na konec zprávy“**

V této části mohou nastat tři případy příjmu znaku:

- Příjem znaku „:“ – přechod do stavu „příjem znaků“ a smazání dosud přijatých dat
- Příjem znaku „LF“ – přechod do stavu „kontrola“ (celý požadavek je již přijat)
- Příjem jiného znaku – tento znak je ignorován

### **5.3.1.4 Stav „Kontrola“**

Nejprve je zde kontrolováno, jestli není délka přijaté zprávy kratší než 9 ASCII znaků. Následně probíhá kontrola CRC a zjišťuje se, je-li zpráva adresována tomuto modulu (přímá adresace nebo broadcast). Pokud není jakákoliv kontrola úspěšná, jednotka přechází do stavu „nečinný“. Proběhne-li kontrola bez chyb, modul začne vykonávat požadovanou akci (přechod do stavu „Vykonání požadované akce“).

### **5.3.1.5 Stav „Vykonání požadované akce“**

Tento stav vykonává podobné úkony jako stav v režimu Modbus RTU (viz. kapitola 5.2.1.5). Rozdílný je pouze počet podporovaných funkcí. Detailním

popisem funkcí pro ASCII mód se zabývá část 5.3.2. Jakmile je akce vykonána, jednotka přechází do stavu „start odesílání“.

### 5.3.1.6 Stav *„Start odesílání“*, *„Odesílání odpovědi“*, *„Konec odesílání“*

Postupně je odesílány znaky „:“, zpráva s odpovědí, CR a LF. Ke každému znaku je přidán jeden bit s příslušným typem parity. Po odeslání všech znaků modul přejde do stavu „nečinný“.

## 5.3.2 Podporované kódy funkcí pro režim ASCII

### 5.3.2.1 Čti diskrétní vstupy (*Read Discrete Inputs*) – 02 (0x02)

Princip této funkce je stejný jako v RTU režimu (viz. část 5.2.2.1). Rozdílný je pouze formát přijímané a odesílané zprávy. Ten je pro různé typy zpráv uveden v Tab. 5.22, Tab. 5.23 a Tab. 5.26.

**Tab. 5.22 Požadavek funkce 0x02 pro mód ASCII**

Význam	Délka	Hodnota
Kód funkce	2 znaky (2 byty)	0,2 (0x02)
Počáteční adresa	4 znaky (4 byty)	0,0,0,0 až 8 (0x00 až 0x08)
Počet vstupů	4 znaky (4 byty)	0,0,0,1 až 8 (0x08)

**Tab. 5.23 Odpověď funkce 0x02 pro mód ASCII**

Význam	Délka	Hodnota
Kód funkce	2 znaky (2 byty)	0,2 (0x02)
Počet bytů	2 znaky (2 byty)	0,1 (0x01)
Stav vstupů	2 znaky (2 byty)	

### 5.3.2.2 Zapiš jeden registr (*Write Single Register*) – 06 (0x06)

Vlastnosti této funkce jsou stejné jako v RTU režimu (viz. kapitola 5.2.2.3) s rozdílem, že v ASCII módu je možné upravovat pouze první registr. Povolený je zápis hodnoty 1 (ponechání ASCII) a hodnoty 0 (přechod do RTU).

Formát přijímané a odesílané zprávy pro ASCII režim je uveden v Tab. 5.24, Tab. 5.25 a Tab. 5.26.

**Tab. 5.24 Požadavek funkce 0x06 pro mód ASCII**

Význam	Délka	Hodnota
Kód funkce	2 znaky (2 byty)	0,6 (0x06)
Adresa registru	4 znaky (4 byty)	0,0,0,0 (0x00)
Hodnota registru	4 znaky (4 byty)	0,0,0,0 nebo 1 (0x00, 0x01)

**Tab. 5.25 Odpověď funkce 0x06 pro mód ASCII**

Význam	Délka	Hodnota
Kód funkce	2 znaky (2 byty)	0,6 (0x06)
Adresa registru	4 znaky (4 byty)	0,0,0,0 (0x00)
Hodnota registru	4 znaky (4 byty)	0,0,0,0 nebo 1 (0x00, 0x01)

**Tab. 5.26 Záporné odpovědi (chyby) funkcí 0x02 a 0x06 pro mód ASCII**

Význam	Délka	Hodnota
Kód funkce	2 znaky (2 byty)	8,2 (0x82) nebo 8,6 (0x86)
Chybový kód	2 znaky (2 byty)	0,2 (0x02) nebo 0,3 (0x03)

#### 5.4 PŘEPÍNÁNÍ SMĚRU TOKU DAT PRO RS-485

Rozhraní RS-485 je poloduplexní, je tedy nutné řídit směr toku dat. Řízení je zajištěno softwarovým nastavováním portu P1.2 na log. 0 nebo na log. 1 (ovládání směru komunikace). Hodnota portu je po restartu nebo ve stavu „nečinný“ u obou režimů komunikace nastavena na log. 0. Je tedy možné přijímat data přes RS-485. Před každým odesláním dat je port přednastaven na log. 1. Směr komunikace je nastaven na odesílání dat.

## 6. ZÁVĚR

Navržený modul digitálních výstupů s rozhraním Modbus splňuje po hardwarové i softwarové stránce všechny požadavky uvedené v zadání. Modul nabízí podobné vlastnosti jako profesionální jednotky digitálních vstupů. Mezi tyto vlastnosti patří především galvanické oddělení vstupů, komunikační rozhraní RS-485 a RS-232 a implementovaný protokol Modbus RTU.

Navržená jednotka dokáže oproti většině profesionálních produktů komunikovat i pomocí režimu Modbus ASCII. Oba režimy podporují pro modul nejdůležitější funkci, kterou je čtení stavů digitálních vstupů. V RTU módu je například navíc možné provádět diagnostiku, číst záznam historie stavů vstupů, nastavovat rychlost komunikace, měnit typ parity nebo je možné se dotázat na identifikaci přístroje.

Při návrhu hardware bylo dbáno na co nejmenší rozměry DPS z důvodu kompaktnosti a také ceny zhotovení desky plošných spojů. Použité součástky jsou tedy většinou zvoleny v SMD provedení.

V průběhu ožívování a testování byly odhaleny nějaké chyby v návrhu DPS, které jsou v nejnovější revizi opraveny. Například byla zcela nově navržena detekce změny stavů digitálních vstupů a jejich světelná indikace. Předchozí indikace a detekce sice byly funkční, jenže byly zbytečně zatěžovány báze tranzistorů. To způsobovalo pomalejší detekci změny stavů digitálních vstupů.

## 7. LITERATURA

- [1] *Bezstarosti Bros: Jednochip* [online]. 2001 [cit. 2008-12-15]. Dostupný z WWW: <<http://www.bezstarosti.cz/elec/jednochip/jednochip.htm>>.
- [2] *Cyklický redundantní součet : Wikipedie* [online]. 2009 , 27.4.2009 [cit. 2009-05-14]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Cyklick%C3%BD\\_redundantn%C3%AD\\_sou%C4%8Det](http://cs.wikipedia.org/wiki/Cyklick%C3%BD_redundantn%C3%AD_sou%C4%8Det)>.
- [3] ING. KAHÁNEK, Michal. Měřicí, řídicí a komunikační převodníky ICP DAS s protokoly Modbus . *Automa* [online]. 2007, č. 3 [cit. 2008-12-15]. Dostupný z WWW: <[http://www.odbornecasopisy.cz/index.php?id\\_document=34302](http://www.odbornecasopisy.cz/index.php?id_document=34302)>.
- [4] ING. POUCHA, Pavel. *RS485 průmyslová komunikace: Přenos dat po linkách RS485 a RS422* [online]. [2008] [cit. 2008-12-15]. Dostupný z WWW: <<http://www.rs485.cz/papouch1.htm>>.
- [5] ING. RONEŠOVÁ, Andrea. *Přehled protokolu Modbus*. [s.l.]: [s.n.], 2005. 20 s. Dostupný z WWW: <<http://home.zcu.cz/~ronesova/bastl/files/Modbus.pdf>>.
- [6] *Katalog GM Electronic 2008*. Praha: GM Electronic spol. s.r.o., 2008. 485 s.
- [7] *Katalogový list +5V Powered, Multichannel RS-232 Drivers/Receivers*. Sunnyvale, USA: MAXIM, 2000. 36 s.
- [8] *Katalogový list 8-bit Flash Microcontroller AT89C51ED2* . [s.l.] : Atmel, 2007. 136 s. Dostupný z WWW: <[www.atmel.com/literature](http://www.atmel.com/literature)>.
- [9] *Katalogový list DC/DC Converters*. Zurich, Switzerland: TRACO POWER, 2007. 3 s.
- [10] *Katalogový list High Reliability Photo Coupler*. [s.l.] : Cosmo, 2002. 2 s. Dostupný z WWW: <<http://www.cosmo-ic.com>>.

- [11] *Katalogový list Linear Technology - Differential Bus Transceiver*. Milpitas, USA: Linear Technology Corporation, 1995. 12 s.
- [12] *Katalogový list RAILBOX*. [s.l.] : Enika, 2006. 4 s. Dostupný z WWW: <[http://www.enika.cz/download/pdf\\_vyrobce/cat\\_railbox\\_8\\_11.pdf](http://www.enika.cz/download/pdf_vyrobce/cat_railbox_8_11.pdf)>.
- [13] LINDA, Martin. Moduly s protokolem Modbus pro vzdálený sběr dat. *Automa* [online]. 2008, č. 8 [cit. 2008-12-15]. Dostupný z WWW: <[http://www.odbornecasopisy.cz/index.php?id\\_document=37718](http://www.odbornecasopisy.cz/index.php?id_document=37718)>.
- [14] *M-7000-série modulů vzdálených I/O s podporou protokolu Modbus RTU* [online]. ELVAC IPC, c1998-2009 [cit. 2009-12-15]. Dostupný z WWW: <<http://www.icpcon.cz/message.asp?message=218>>.
- [15] MATOUŠEK, David. *C pro mikrokontroléry ATMEL AT89S52 : Příklady a aplikace pro C51 ve vývojovém prostředí KEIL µVision3* . 6. díl. Praha: BEN - technická literatura, 2007. 231 s.
- [16] *MODBUS application protocol specification v1.1b*. [s.l.] : Modbus-IDA, 2006. 51 s. Dostupný z WWW: <<http://modbus-ida.org/specs.php>>.
- [17] *MODBUS over Serial Line: Specification and Implementation Guide v1.02*. [s.l.] : Modbus-IDA, 2006. 44 s. Dostupný z WWW: <<http://modbus-ida.org/specs.php>>.
- [18] OLMR, Vít. *Sériová linka RS-232* [online]. HW server s.r.o., c1997-2009 [cit. 2008-12-15]. Dostupný z WWW: <<http://hw.cz/rs-232>>.
- [19] *Quido RS 10/1 – 10 vstupů a 1 výstup* [online]. Papouch s.r.o., c1990-2009 [cit. 2009-05-15]. Dostupný z WWW: <[http://www.papouch.com/shop/scripts/\\_detail.asp?kacislo=0251](http://www.papouch.com/shop/scripts/_detail.asp?kacislo=0251)>.
- [20] SKALICKÝ, Petr. *Procesory řady 8051*. 2. rozš. vyd. Praha: BEN - technická literatura, 2003. 158 s.
- [21] STANĚK, Jan, ŘEHÁK, Jan. *RS 485 & 422* [online]. HW server s.r.o., c1997-2009 [cit. 2008-12-15]. Dostupný z WWW: <<http://hw.cz/docs/rs485/rs485.html>>.

## 8. SEZNAM ZKRATEK

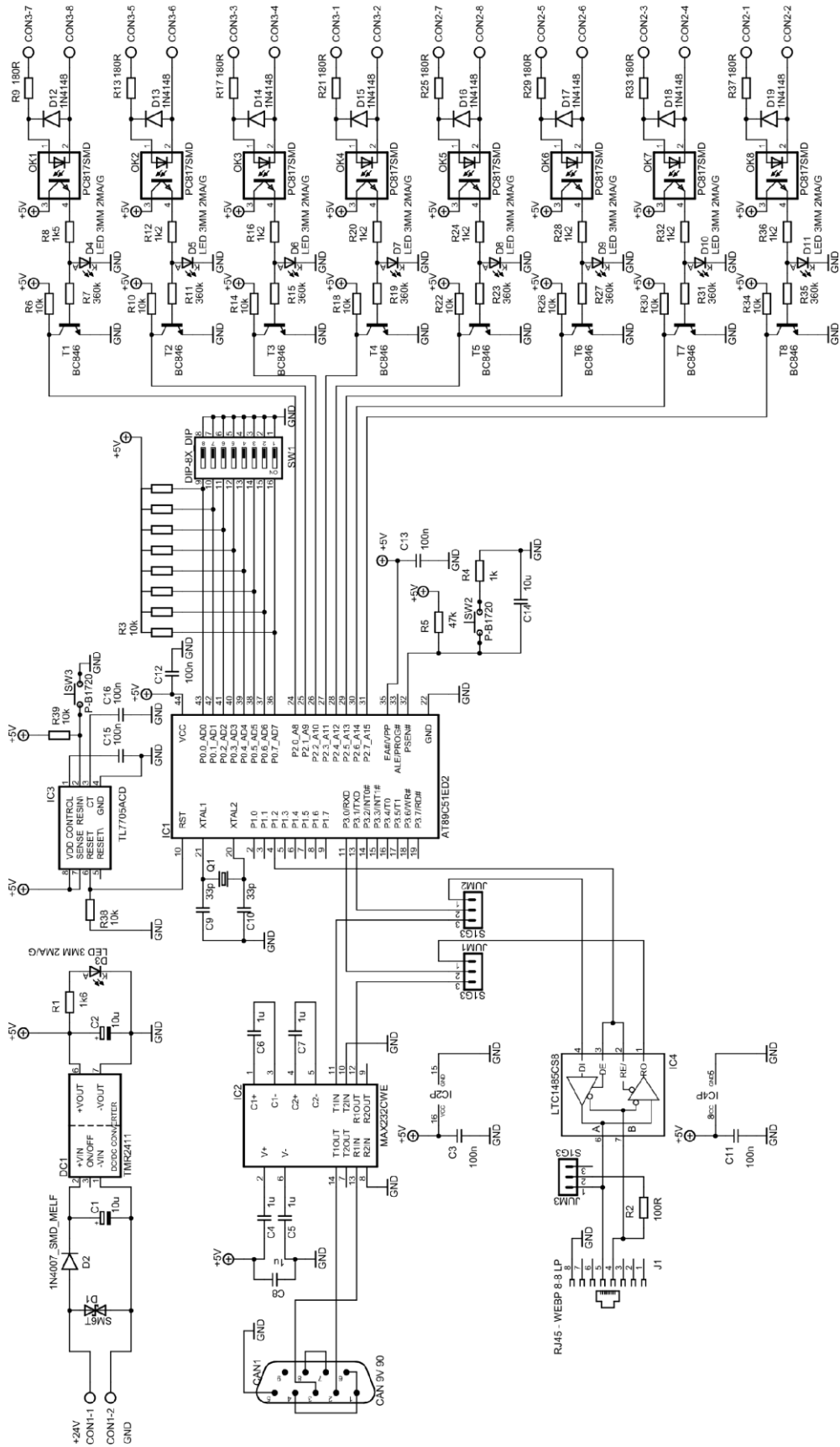
<b>Zkratka</b>	<b>Význam</b>	<b>Popis</b>
ADU	(Application Data Unit)	Aplikační datová jednotka
ALE	(Address Latch Enable)	Povolení zápisu spodní poloviny platné adresy
ASCII	(American Standard Code for Information Interchange)	Kódová tabulka znaků anglické abecedy
CMOS	(Complementary Metal Oxide Semiconductor)	Technologie výroby polovodičových součástek
CRC	(Cyclic Redundancy Check)	Cyklický kontrolní součet
DC	(Direct current)	Stejnoseměrný proud
DIN	(Deutsche Industrie Norm)	Německá průmyslová norma
DPS		Deska plošných spojů
DPTR	(Dual Data Pointer Register)	16 bitový pomocný ukazatel datové paměti
EA	(External Access Enable)	Povolení vnější paměti
EEPROM	(Electrically Erasable Programmable ROM)	Elektricky vymazatelná programovatelná paměť ROM
GND	(Ground)	Společný zemní vodič
HMI	(Human - Machine Interface)	Rozhraní mezi uživatelem a strojem
I/O	(Input / Output)	Vstup / Výstup
IC	(Integrated Circuit)	Integrovaný obvod
IP	(Internet Protocol)	Standardní síťový protokol
ISO	(International Organization for Standardization)	Mezinárodní organizace pro normalizaci

<b>Zkratka</b>	<b>Význam</b>	<b>Popis</b>
LED	(Light Emitting Diode)	Svítivá dioda
LRC	(Longitudinal Redundancy Check)	Kontrola podélnou paritou
LSB	(Least Significant Bit)	Nejméně významný (řádově) bit
MSB	(Most Significant Bit)	Nejvýznamnější (řádově) bit
OSI	(Open Systems Interconnection)	Propojení otevřených systémů
PDU	(Protocol Data Unit)	Datový paket
PLC	(Programmable Logic Controller)	Programovatelný logický automat
PSEN	(Program Strobe Enable)	Povolení čtení z vnější paměti programu
RAM	(Random Access Memory)	Paměť s náhodným přístupem
RMS	(Root Mean Square)	Efektivní hodnota střídavého napětí
ROM	(Read Only Memory)	Paměť pouze pro čtení
SCADA	(Supervisor, Control And Data Acquisition)	Správa, řízení a sběr dat
SDK	(Software Development Kit)	Systémový vývojový nástroj
SFR	(Special Function Registers)	Registry speciálních funkcí
SMD	(Surface Mounted Device)	Součástka pro povrchovou montáž
SPI	(Serial Peripheral Interface)	Sériové rozhraní
TCP	(Transmission Control Protocol)	Přenosový řídicí protokol
TTL	Transistor-Transistor-Logic	Logika integrovaných obvodů
UART	(Universal Asynchronous Receiver-Transmitter)	Univerzální asynchronní přijímač a vysílač
XOR	(Exclusive OR)	Nonekvivalence

## **SEZNAM PŘÍLOH**

- Příloha 1 Schéma zapojení modulu
- Příloha 2 Seznam použitých součástek
- Příloha 3 Výkresy desky plošných spojů

# Příloha 1



Obr. 1 Celkové schéma zapojení

## Příloha 2

Tab. 1 Elektrická rozpiska součástek

Díl č.	Označení	Druh	Typ, Hodnota	Pouzdro, Poznámka
1	IC1	Mikrokontrolér	AT89C51ED2	SMD PLCC44
2	IC2	Převodník na RS-232	MAX232CWE	SMD SO16
3	IC3	Nulovací obvod	TL7705ACD	SMD SO8
4	IC4	Převodník na RS-485	LTC1485CS8	SMD SO8
5	DC1	DC/DC měnič	TMR2411	Traco SIP8
6	Q1	Krystal	11,0592 MHz	HC49US, nízký
7	R1	Rezistor	1k6	SMD 0805; 1 %
8	R2	Rezistor	100R	SMD 0805; 1 %
9	R3	Rezistorová síť	RR 8x10k	SIP 8; 2 %
10	R4	Rezistor	1k	SMD 0805; 1 %
11	R5	Rezistor	47k	SMD 0805; 1 %
12	R6, R10, R14, R18, R22, R26, R30, R34	8 x Rezistor	10k	SMD 0805; 1 %
13	R7, R11, R15, R19, R23, R27, R31, R35	8 x Rezistor	360k	SMD 0805; 1 %
14	R8, R12, R16, R20, R24, R28, R32, R36	8 x Rezistor	1k5	SMD 0805; 1 %
15	R9, R13, R17, R21, R25, R29, R33, R37	8 x Rezistor	180R	SMD 0805; 1 %
16	R30, R31	2 x Rezistor	10k	SMD 0805; 1 %
17	C1	Elektrolyt. kondenzátor	10M / 63V	SMD; 20 %
18	C2	Elektrolyt. kondenzátor	10M / 63V	SMD; 20 %
19	C3	Keram. kondenzátor	100n	SMD 0805; X7R
20	C4, C5, C6, C7, C8	5 x Keramický kondenzátor	1M	SMD 0805; X7R
21	C9, C10	2 x Keramický kondenzátor	33p	SMD 0805;NPO
22	C11, C12, C13	3 x Keramický kondenzátor	100n	SMD 0805; X7R
23	C14	Keram. kondenzátor	10M	SMD 0805; X5R
24	C15, C16	2 x Keramický kondenzátor	100n	SMD 0805; X7R

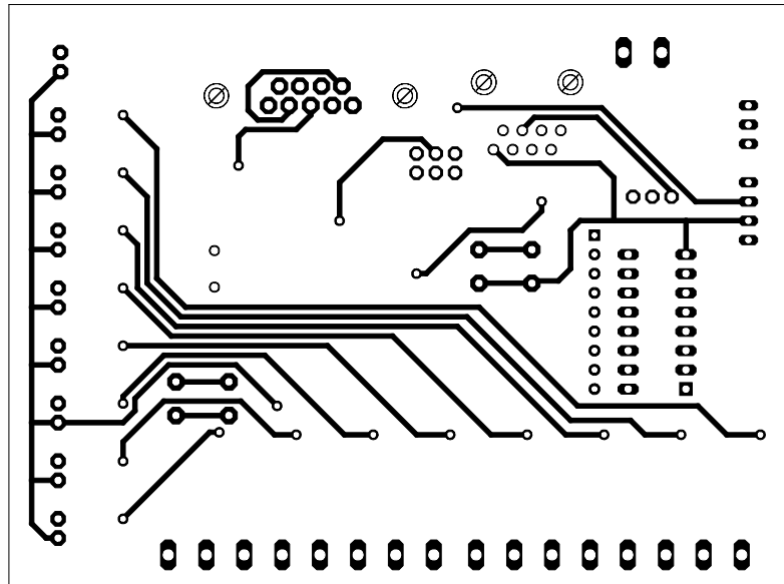
**Tab. 2 Elektrická rozpiska součástek – pokračování**

<b>Díl č.</b>	<b>Označení</b>	<b>Druh</b>	<b>Typ, Hodnota</b>	<b>Pouzdro, Poznámka</b>
25	D1	Transil	SM6T27CA	SMD; 27 V
26	D2	Dioda	1N4007	SMD MELF
27	D3, D4, D5, D6, D7, D8, D9, D10, D11	9 x LED dioda	3MM 2MA/G	3mm, zelená
28	D12, D13, D14, D15, D16, D17, D18, D19	8 x Dioda	1N4148	SMD 0805; 75 V
29	T1, T2, T3, T4, T5, T6, T7, T8	8 x Tranzistor NPN	BC846	SMD SOT23
30	OK1, OK2, OK3, OK4, OK5, OK6, OK7, OK8,	8 x Optočlen	PC817B	SMD
31	SW1	Přepínač	DIP 8x	
32	SW2	Tlačítkový spínač	P-B1720	
33	SW3	Tlačítkový spínač	P-B1720	
34	JUM1	Jumperová lišta	S1G3	3 piny
35	JUM 2	Jumperová lišta	S1G3	3 piny
36	JUM 3	Jumperová lišta	S1G3	3 piny
37	J1	Konektor	WEBP 8-8 LP	RJ 45
38	CAN9	Konektor	CAN 9 V 90	CANON 9
39	CON1	Svorkovnice	AK 1550 / 2	RM = 5mm
40	CON2	4 x Svorkovnice	AK 1550 / 2	RM = 5mm
41	CON3	4 x Svorkovnice	AK 1550 / 2	RM = 5mm
42		Patice PLCC44	PCCL44Z	SMD

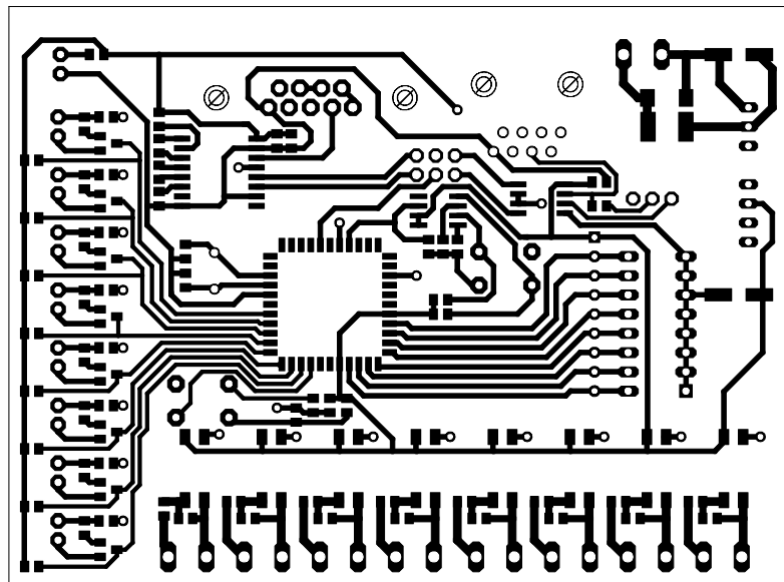
**Tab. 3 Mechanická rozpiska součástek**

<b>Díl č.</b>	<b>Množství</b>	<b>Druh</b>	<b>Typ</b>	<b>Poznámka</b>
43	1	Box do DIN lišty	RAILBOX 22.5	
44	3	Jumper	JUMP-SW	

## Příloha 3



Obr. 2 Deska plošných spojů – pohled ze strany spojů



Obr. 3 Deska plošných spojů – pohled ze strany součástek

