

Technická univerzita v Liberci

Fakulta mechatroniky, informatiky a mezioborových studií
Studijní program: Elektronické informační a řídicí systémy

Bakalářská práce



Matěj Kolář

On-line monitoring geofyzikálních dat

Ústav mechatroniky a technické informatiky

Vedoucí práce: Ing. Miloš Hernych
Konzultant práce: Doc. Ing. Milan Hokr, Ph.D.

Liberec, 2010

Prohlášení

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

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

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

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

Datum: 20. 5. 2010

Podpis:

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Milošovi Hernychovi za odborné vedení a především trpělivost při konzultování jednotlivých problémů.

Abstrakt

Práce jako celek pojednává o kompletní realizaci elektronického zařízení. Výsledkem je systém umožňující on-line monitoring geofyzikálních dat. Popsána je teorie komunikační sběrnice RS-485, popis protokolu FINET firmy FIEDLER. Základní přehled rodiny mikrokontrolérů AVR firmy Atmel, která byla v zařízení použita. Je zde popsán budič sběrnice RS-485, GSM/GPRS modul Cinterion TC65i a obvod FTDI. Schéma a popis zapojení zařízení. Software je napsán v jazyce C. Zahrnuje obsluhu GSM/GPRS modulu pomocí AT příkazů a komunikaci po sběrnici RS-485 protokolem FINET.

Klíčová slova: AVR, RS-485, FINET, jazyk C, GPRS, AT příkazy

Abstract

This thesis deals about complete realization electronic devices. This device allows on-line monitoring geophysics. In work is theory of communication bus RS-485, details of protocol FINET by FIEDLER. There are basic knowledge about AVR microcontrollers. There controllers are in the device. There are details of driver RS-485 bus, GSM/GPRS module Cinterion TC65i and integrated circuit FTDI. There is schematic and details about connections device. Software is written in C. Include services GSM/GPRS module by AT commands and communications through RS-485 bus with FINET protocol.

Keywords: AVR, RS-485, FINET, language C, GPRS, AT commands

Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt	5
Obsah	6
Slovník pojmů.....	9
Seznam obrázků.....	10
1 Úvod.....	11
2 Stávající infrastruktura	13
2.1 Probíhající výzkum	13
2.2 Sběrnice RS-485	13
2.2.1 Dvou vodičová verze RS-485.....	14
2.2.2 Čtyřvodičová verze RS-485.....	14
2.3 Finet.....	14
2.3.1 Komunikační rámec	15
2.3.2 Znaky komunikačního rámce.....	15
2.3.3 Adresace.....	16
2.3.4 CONNECT.....	17
2.3.5 Služba READN	17
3 Prostředky pro realizaci zařízení.....	19
3.1 Atmel AVR.....	19
3.1.1 Architektura AVR	19
3.1.2 ISP Programátor.....	20
3.1.3 AVR Dragon	20
3.1.4 AVR Studio.....	20

3.1.5	ISP, JTAG, debugWIRE	20
3.2	Cinterion TC65i	21
3.2.1	Napájení.....	21
3.2.2	SIM karta	23
3.2.3	Komunikační rozhraní	23
3.2.4	Java	24
3.3	HW prostředky RS-485.....	24
4	Vývoj zařízení - hardware.....	26
4.1	Měřicí modul.....	26
4.1.1	Výběr procesoru.....	26
4.1.2	Napájení.....	27
4.1.3	Varianta 1, s ATmega8	27
4.1.4	Varianta 2, s ATmega48	27
4.1.5	Varianta 3, s ATmega162	28
4.2	Řídící jednotka.....	29
4.2.1	Napájení.....	30
4.2.2	Procesor.....	34
4.2.3	Cinterion TC65i.....	36
4.2.4	RS-485.....	39
4.2.5	USB.....	40
4.2.6	Konstrukční řešení.....	42
5	Vývoj software	44
5.1	Jazyk C	44
5.2	Projekt	44
5.2.1	Moduly	46
5.2.2	main.c	46

Obsah

5.2.3	init.c	46
5.2.4	usart.c	46
5.2.5	rs485.c	47
5.2.1	err.c	48
6	Závěr	49
	Přílohy	52
	Literatura	53

Slovník pojmů

MIPS – jednotka udávající počet miliónů instrukcí za sekundu

JTAG – programovací rozhraní

debugWIRE – programovací rozhraní

FLASH – datová paměť, určená pro uložení firmware

SRAM – operační paměť

EEPROM – datová paměť, lze s ní pracovat, data zůstanou i bez napájení

ISP – programovací rozhraní

SPI – komunikační rozhraní

MASTER – prvek navazující komunikaci

SLAVE – prvek pouze odpovídající

GPRS – mobilní internet

1-Wire – jednovodičová sběrnice firmy DALLAS

USART – Univerzální synchronní a asynchronní přijímač a vysílač

TTL – Napěťové úrovně 0 V pro logickou 0 a 5 V pro logickou 1

AT příkazy – Soubor příkazů umožňujících nastavování parametrů modemů, navazování a řízení spojení v telefonní síti (GSM)

DPS – Deska plošných spojů

M2M – Machine2Machine – bezdrátová výměna dat mezi dvěma zařízeními

URC – Zpráva informující o změně určitých parametrů bez nutnosti opakovaných dotazů

BaudRate – Komunikační rychlost jednotky USART

Seznam obrázků

<i>OBRÁZEK 1 - ISP KONEKTOR</i>	<i>21</i>
<i>OBRÁZEK 2 - JTAG KONEKTOR</i>	<i>21</i>
<i>OBRÁZEK 3 - ODEBÍRANÝ PROUD BĚHEM VYSÍLÁNÍ PŘI FREKVENCI 850 A 900 MHZ.</i>	<i>22</i>
<i>OBRÁZEK 4 - ODEBÍRANÝ PROUD BĚHEM VYSÍLÁNÍ PŘI FREKVENCI 1800 A 1900 MHZ.</i>	<i>22</i>
<i>OBRÁZEK 5 – SCHÉMA ZAPOJENÍ MAX1483 ŘÍDÍCÍ JEDNOTKY</i>	<i>25</i>
<i>OBRÁZEK 6 – SCHÉMA ZAPOJENÍ NAPÁJECÍHO ZDROJE ŘÍDÍCÍ JEDNOTKY</i>	<i>33</i>
<i>OBRÁZEK 7 - SCHÉMA ZAPOJENÍ PROCESORU ŘÍDÍCÍ JEDNOTKY</i>	<i>36</i>
<i>OBRÁZEK 8 - SCHÉMA ZAPOJENÍ MODULU TC65I ŘÍDÍCÍ JEDNOTKY</i>	<i>38</i>
<i>OBRÁZEK 9 – SCHÉMA ZAPOJENÍ MAX1483 ŘÍDÍCÍ JEDNOTKY</i>	<i>40</i>
<i>OBRÁZEK 10 – SCHÉMA ZAPOJENÍ FTDI ČIPU ŘÍDÍCÍ JEDNOTKY</i>	<i>41</i>
<i>OBRÁZEK 11 - SCHÉMA ZAPOJENÍ SMD KONEKTORU A SVORKOVNICE</i>	<i>43</i>
<i>OBRÁZEK 12- NEOSAZENÁ DPS, MODUL CINTERION TC65I A SIM KARTA</i>	<i>50</i>
<i>OBRÁZEK 13- ŘÍDÍCÍ JEDNOTKA V KRABÍČCE, BEZ ČELNÍHO PANELU</i>	<i>51</i>
<i>OBRÁZEK 14 - POHLED DO TUNELU</i>	<i>51</i>

1 Úvod

Úkolem práce je návrh a fyzická realizace vzorového systému sloužícího ke sběru a archivaci geofyzikálních dat. Zařízení je nutné integrovat do stávajícího a souběžně vyvíjeného systému pro geofyzikální výzkum ve štolě přivaděče vody pro úpravnu v Bedřichově.

V době zadávání práce byla instalována infrastruktura datové sítě sestávající se ze dvou kroucených dvojlinek, střídavého napájení 24 V a průmyslových krabic instalovaných po cca 50 metrech. Délka celé instalace je cca 1000 metrů. Předpoklad je použití sběrnice RS-485 a komunikačního protokolu FINET firmy Fiedler-Mágr (v podstatě obdoba ProfiBusu) s možnou modifikací. Měření ve zbývající části tunelu má být řešeno bezdrátovou komunikací a napájeno z baterií. Nedílnou částí systému bude i komunikace s webovým serverem, který umožní přístup k datům pomocí internetové sítě.

Mým úkolem je návrh a realizace řídicí jednotky, která bude z jednotlivých geofyzikálních senzorů prostřednictvím komunikační sběrnice RS485 a výše uvedeného protokolu (FINET) vyčítat naměřené hodnoty a pomocí GPRS připojení k internetu tyto hodnoty ukládat na FTP server do souboru. Soubor by poté měl zpracovávat systém instalovaný mimo štolu a již není předmětem mé práce.

Roli geofyzikálních senzorů zde sehrávají moduly s A-D převodníkem a rozhraním RS485 nebo moduly disponující místo A-D převodníku konkrétním komunikačním protokolem různých sofistikovanějších senzorů. Příkladem může být komunikační sběrnice 1-Wire firmy Dallas, kterou používají teploměry řady DS18B20. Případně koupené senzory firmy Fiedler-Márg, které komunikují protokolem.

Já osobně jsem se pro realizaci zařízení rozhodl použít jednočipové RISC mikropočítače firmy Atmel s architekturou AVR. Pro první pokusy jsem sestavil

triviální ISP programátor do sériového portu, ale po určitém čase jsem zakoupil vývojový nástroj od Atmelu AVR Dragon. Firmware píšu v jazyce C a ladím jej ve freewarovém prostředí Atmelu AVR Studio a balíčku WinAVR, jehož součástí je GNU překladač GCC jazyka C a velké množství doplňujících modulů obecného jazyka C i konkrétně pro potřeby AVR, včetně pojmenování registrů všech procesorů AVR. Pro návrh desky plošných spojů jsem zvolil software Eagle.

Sběrnice RS485 je snadno vyřešena budičem MAX1483 a pro GPRS komunikaci používám M2M modul firmy Cinterion, TC65i. Modul je určen pro přímou montáž na DPS pomocí B2B konektoru. Lze jej ovládat pomocí AT příkazu po sériové lince nebo programovat v jazyce JAVA jako samostatný procesor. Já modul ovládám pomocí mikrořadiče AVR.

V průběhu realizace vzniklo několik modulů, kromě řídicí jednotky s GPRS rozhraním také podřízený modul s A-D převodníkem a modul pro převod textového nebo neodpovídajícího komunikačního protokolu senzorů Fiedler-Mágr na protokol používaný na sběrnici ve škole.

2 *Stávající infrastruktura*

2.1 *Probíhající výzkum*

Cílem 4. etapy výzkumu granitového masivu v bedřichovském tunelu bude zřízení výzkumné stanice k dlouhodobému sledování geologických jevů v reálném prostředí granitů Českého masivu pro bezpečnostní výpočty a pro návrh požadavků, indikátorů a kritérií na výběr vhodného geologického prostředí pro HÚ v ČR. Tato nová fáze výzkumu a pozorování si vyžádá zvýšené náklady na konstrukci automatizovaného monitoringu v tunelu instalované přístrojové techniky a technologické vybavení řídicího centra na Technické univerzitě v Liberci.

2.2 *Sběrnice RS-485*

RS-485 (stejně tak RS-422) se vyznačuje dvou vodičovým propojením jednotek. Označují se písmeny A a B. Maximální délka sběrnice je až 1200 m, maximální počet vysílačů a přijímačů je typicky 32. Lze použít přijímače s větším vstupním odporem, které umožní použití až 256 jednotek. Maximální přenosová rychlost je 10 Mb/s, ale záleží na délce vedení. Sběrnice musí být zakončena terminátory. Vhodné je také použít velké odpory na připojení kladnějšího vodiče k napájecímu napětí (5V) a zápornějšího vodiče k zemi. Je to mnohdy důležitější než terminátory (nejsou na krátké vzdálenosti nutné), protože v situaci kdy žádný z účastníků nevysílá, může být na sběrnici v podstatě cokoliv.

Logické stavy jsou reprezentovány rozdílným napětím mezi oběma vodiči. Je to rozdíl oproti RS-232, kde se úrovně stavů vztahují k zemi. Rozdílným napětím mezi oběma vodiči je výhodný zejména kvůli eliminaci naindukovaného rušivého napětí. Logický stav „1“ je reprezentován rozdílovým napětím mezi vodiči A a B menší než -300 mV, logický stav "0" rozdílovým napětím větším než

+300 mV. Správný vysílač by měl na výstupu generovat napětí +2 V (příp. -2 V), správný přijímač by měl na vstupu rozlišit napětí ± 200 mV.

2.2.1 Dvou vodičová verze RS-485

Přenos dat se uskutečňuje pomocí 7 nebo 8 bitových rámců se startbitem, jedním nebo více stopbity a případně i paritním bitem. Přenos je poloduplexní a proto se vyžaduje řízení přenosu dat. Výhodou je, že pomocí dvou vodičové linky RS-485 je možné vytvořit více prvkovou komunikační síť.

2.2.2 Čtyřvodičová verze RS-485

Čtyřvodičová verze RS-485 umožňuje obousměrnou komunikaci, v tomto režimu není nutné řízení směru přenosu dat. Používá se jako přímá náhrada RS232 na větší vzdálenosti. V síti s více účastníky se spíše nepoužívá, nicméně je možné použít síť s jednou stanicí MASTER, pro kterou je první linka přijímací a druhá vysílací. Pro účastníky typu SLAVE je první linka vysílací a druhá přijímací. MASTER se o svou vysílací linku nemusí dělit, účastníci typu SLAVE se dělí o jednu vysílací linku. Předpokladem u SLAVE je možnost odpojovat vysílací kanál. Komunikace je rychlejší protože MASTER nemusí čekat před vysíláním a nemůže se stát, že se kvůli selhání jednoho účastníka zablokuje vysílací linka.

2.3 Finet

FINET je binární protokol s pevným rámcem (úvodní a ukončovací znak) určený pro přenos po duplexních i poloduplexních komunikačních kanálech. Komunikaci vždy navazuje nadřazený systém na principu dotaz - odpověď. Stanice odpovídá na každý dotaz pro ni určený. Důležitým parametrem je doba označená jako klid na lince. Ta je podmínkou pro rozeznání začátku bloku dat. Obvykle je nastavena na trojnásobek doby potřebné k odeslání jednoho bytu. Tato prodleva je nutná jak na straně SLAVE mezi přijetím dotazu a odesláním

odpovědi, tak i na straně MASTER po přijetí odpovědi před vysláním dalšího dotazu. Druhým časovým parametrem je maximální doba mezi jednotlivými byty zprávy. Po jejím vypršení je zbytek zprávy ignorován a čeká se na příchod nové zprávy (SLAVE) nebo se vyšle opakovaný dotaz (MASTER). Obvyklé nastavení komunikačního portu je 19200 b, 8 bitů, žádná parita, 1 stop bit a žádné řízení toku dat.

2.3.1 Komunikační rámec

Směr MASTER -> SLAVE - Zpráva (dotaz) bez datového pole

SD1	DA	SA	FC	FCS	ED
-----	----	----	----	-----	----

Směr MASTER -> SLAVE - Zpráva (dotaz) s datového pole

SD2	LE	LER	SD2R	DA	SA	FC	DATA	FCS	ED
-----	----	-----	------	----	----	----	------	-----	----

Směr SLAVE -> MASTER - Odpověď bez datového pole - krátké potvrzení

SACK

Směr SLAVE -> MASTER - Odpověď bez datového pole

SD1	DA	SA	FC	FCS	ED
-----	----	----	----	-----	----

Směr SLAVE -> MASTER - Odpověď s datovým polem

SD2	LE	LER	SD2R	DA	SA	FC	DATA	FCS	ED
-----	----	-----	------	----	----	----	------	-----	----

Směr SLAVE -> MASTER - Negativní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	ER1	ER2	FCS	ED
-----	----	-----	------	----	----	----	-----	-----	-----	----

2.3.2 Znaky komunikačního rámce

SD1 – úvodní znak 1, pevná hodnota \$10

SD2 – úvodní znak 2, pevná hodnota \$68

LE – délka dat, vlastní délka dat + 3 (DA,SA,FC)

LER – opakovaná délka dat

2 Stávající infrastruktura

SD2R – opakovaný úvodní znak 2, pevná hodnota \$68

DA – cílová adresa zprávy

SA – zdrojová adresa zprávy

FC – řídicí byte rámce – definuje službu rámce protokolu FINET

\$00 – odpověď CONNECT

\$08 – odpověď s daty

\$0C – negativní odpověď

\$69 – dotaz CONNECT

\$63 – dotaz zápis dat do jednotky

\$6C – dotaz čtení dat z jednotky

ER1 – kód negativní odpovědi

\$F1 – neznámý kód FC

ostatní kódy jsou určeny pro vyšší vrstvu

ER2 – rozšiřující data negativní odpovědi (pro vyšší vrstvu)

DATA – vlastní předávaná data (pro vyšší vrstvu)

FCS – kontrolní součet jako bytový součet DA+SA+FC+ DATA modulo 256.

ED – koncový znak, pevná hodnota \$16

SACK – krátké potvrzení, pevná hodnota \$E5

2.3.3 Adresace

Zdrojová adresa je adresa odesílatele dat, cílová jeho příjemce. Pokud je cílovou adresou 0 odpoví na dotaz všechny zařízení na sběrnici, v odpovědi se pak objeví skutečná adresa tohoto zařízení. Měřicí přístroje FIEDLER nepodporují kontrolu kolize na lince, proto není možné, aby v síti více zařízení

2 Stávající infrastruktura

používalo adresu 0. Komunikace může, ale nemusí začínat krátkým rámcem s FC = \$69 (CONNECT) a DA = 0. Odpověď pak obsahuje skutečnou adresu SLAVE přístroje. Nebo lze CONNECT použít pro zmapování sítě postupnou inkrementací DA.

2.3.4 CONNECT

Slouží pro iniciaci spojení. Dotaz tvoří rámec bez datového pole s FC = \$69 a pozitivní odpověď tvoří rámec bez datového pole s FC = \$00. Pokud je v dotazu použita DA = 0, je v odpovědi skutečná adresa zařízení, které odpovědělo.

Dotaz:

SD1	DA	SA	69	FCS	16
-----	----	----	----	-----	----

Odpověď:

SD1	DA	SA	00	FCS	16
-----	----	----	----	-----	----

2.3.5 Služba READN

Čtení obsahu datových registrů. Přímý přístup k paměti přístroje.

Dotaz:

FC=\$6C

LE=4+4 x n (n je počet čtených bloků dat)

Datové pole tvoří kód služby \$0B a seznam požadovaných registrů:

\$0B	TR1	AL1	AH1	LR1	...	TRn	ALn	AHn	LRn
------	-----	-----	-----	-----	-----	-----	-----	-----	-----

TRn - typ registru, pro jednotku vždy = 3 (registr R)

ALn - dolní byte adresy prvního bytu n-tého čteného registru

AHn - horní byte adresy prvního bytu n-tého čteného registru

LRn - počet čtených bytů n-tého čteného registru

2 Stávající infrastruktura

Odpověď:

FC=8

LE=3+LR1+LR2+...+LRn

Popis datového pole odpovědi:

Data z jednotlivých registrů v pořadí a o velikosti tak jak byly uvedeny v dotazu.

3 Prostředky pro realizaci zařízení

3.1 Atmel AVR

3.1.1 Architektura AVR

Jádro AVR je rodina 8bitových mikropočítačů typu RISC s harvardskou architekturou od firmy Atmel. Skládá se z 32 stejných 8bitových registrů, které mohou obsahovat jak data, tak adresy. Posledních 6 registrů můžeme ve dvojici použít jako ukazatele adresy pro nepřímé adresování paměti dat.

Za zrodem AVR stojí studenti Alf-Egil Bogen a Vegard Wollan. V roce 1997 koncepci koupila firma Atmel. Hlavním konkurentem AVR je firma Microchip s procesory PIC.

V současné době Atmel nabízí několik variant mikroprocesorů AVR od nejmenších tinyAVR. Disponují velikostí paměti FLASH velikostech od 0,5 – 16 KB, paměti EEPROM od 64 – 512 B a paměti SRAM od 32 do 512 B, maximální taktovací frekvencí až 20 Mhz (20 MIPS), až třemi časovači (dva 8bitový, jeden 16bitový) a nejvýše jednou jednotkou USART. Stejně výkonné, ale s větší pamětí a lepší výbavou HW periférií, jsou megaAVR. Disponují až 256 KB FLASH, 4096 B EEPROM a 8 KB SRAM. Nejvyšší modely disponují i 6 časovači a 4 jednotkami USART. Vrcholem v nabídce pak je řada XMEGA, lze je taktovat až na 32 Mhz, disponují maximálně 384 KB paměti FLASH, 4 KB EEPROM a až 32 KB SRAM. Obsahují i 8 časovačů a až 8 jednotek USART. Všechny řady mají zástupce s 10bitovým vícekanálovým A/D převodníkem (XMEGA 12bitovým). Atmel nabízí různé modifikace těchto základních řad určených pro specializované aplikace, řada Automotive AVR (automobilový průmysl), Battery Management AVR (obsahují periférie vhodné pro kontrolu stavu a dobíjení akumulátorů) nebo například modifikace megaAVR s integrovanou RF bezdrátovou komunikací. Existují i procesory řady AVR32, jedná se 32bitovou modifikaci AVR, zařízení disponují například USB i Ethernetem, dosahují rychlosti až 100 MIPS.

3.1.2 ISP Programátor

Pro první nahrání firmware do procesoru jsem zvolil program PonyProg a k němu doporučené triviální zapojení ISP (In System Programmer) programátoru do sériového portu. Programátor fungoval bez problémů, nicméně neposkytoval žádné možnosti ladění.

3.1.3 AVR Dragon

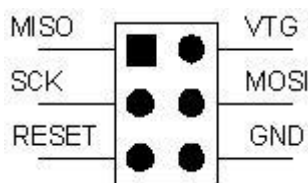
Z výše uvedených důvodů jsem si koupil programátor a debugger AVR Dragon. Jedná se o relativně levné, nicméně velmi účelné řešení. Podporuje ISP, JTAG i programovací a ladící rozhraní Atmelu DebugWIRE. Všechny uvedené rozhraní umožňují programování v aplikaci, přičemž JTAG a DebugWIRE umožňuje i ladění.

3.1.4 AVR Studio

Freewarové vývojové prostředí firmy Atmel. Umožňuje psaní, ladění i simulaci programu. Podporuje plnou integraci GCC překladače.

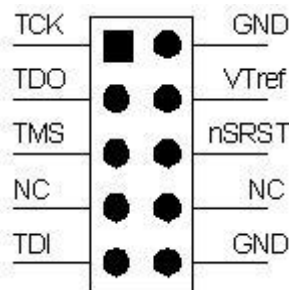
3.1.5 ISP, JTAG, debugWIRE

ISP programovací rozhraní je základní způsob programování většiny procesorů AVR. Využívá šesti vodičů, tří vodičů komunikačního rozhraní SPI, reset, zem a napájecího napětí v případě napájení aplikace pomocí programátoru. Bohužel neumožňuje ladění aplikace. DebugWIRE rozhraní využívá pouze resetovacího pinu a nutné je připojení země a případného napájení. Pomocí něj lze aplikaci bez problémů ladit. Problém je, že nenaprogramovaný procesor má debugWIRE deaktivovaný a je nutné ho zapnout pomocí ISP, čímž se ISP deaktivuje, aktivní může být pouze jedno rozhraní současně. V praxi to znamená, že pro obě rozhraní potřebujeme stejný konektor o šesti pinech.



Obrázek 1 - ISP konektor

JTAG rozhraní funguje samostatně, využívá 10 pinový konektor, umožňuje programování i ladění aplikací. V procesorech AVR je spíše na ústupu kvůli debugWIRE. Procesor má vyhrazené piny přímo pro JTAG (TCK, TDO, TMS, TDI) bez dalších alternativních funkcí.



Obrázek 2 - JTAG konektor

3.2 Cinterion TC65i

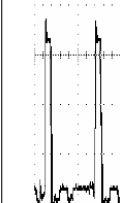
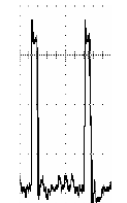
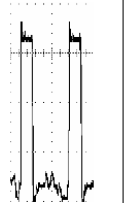
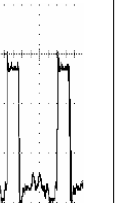
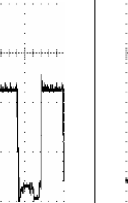
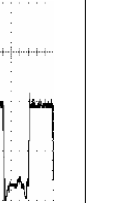
Cinterion TC65i je GSM/GPRS modul. Jde o DPS, která má rozměry 33 x 35 mm a do cílové aplikace se připojuje 80 pinovým SMD konektorem, jehož protikus je příslušenství k modulu. DPS navíc obsahuje konektor pro připojení anténního koaxiálního kabelu. Vše ostatní je třeba připojit pomocí SMD konektoru z cílové aplikace.

3.2.1 Napájení

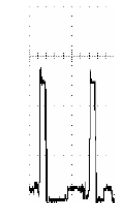
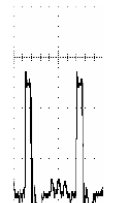
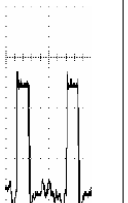
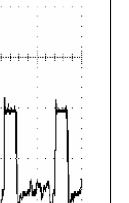
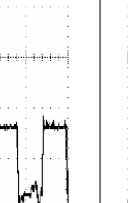
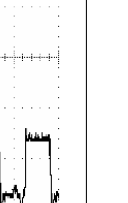
Rozsah napájecího napětí je 3,2 - 4,5 V. Vysílání má poměrně velké proudové nároky. Během tzv. transmit burstu může odebíraný proud dosahovat až 1,6 A, který pouze po dobu vysílacího time-slotu a je závislý na počtu vysílacích time-slotů, na maximálním vysílacím výkonu a vysílací frekvenci (*viz*

3 Prostředky pro realizaci zařízení

Obrázek 3 - Odebíraný proud během vysílání při frekvenci 850 a 900 Mhz., str. 22 a
Obrázek 4 - Odebíraný proud během vysílání při frekvenci 1800 a 1900 Mhz., str. 22).

Mode	GSM call	GPRS Class 8	GPRS Class10		GPRS Class 12	
Timeslot configuration	1Tx / 1Rx	1Tx / 4Rx	2Tx / 3Rx		4Tx / 1Rx	
RF power nominal	2W (33dBm)	2W (33dBm)	2W (33dBm)	1W (30dBm)	1W (30dBm)	0.5W (27dBm)
Radio output power reduction with AT^SCFG, parameter <ropr>	<ropr> = 1 ... 3	<ropr> = 1 ... 3	<ropr> = 1	<ropr> = 2 or 3	<ropr> = 1	<ropr> = 2 or 3
Current characteristics						
Burst current @ 50Ω antenna (typ.)	1600mA	1600mA	1600mA	1300mA	1100mA	880mA
Burst current @ total mismatch	1600mA	1600mA	1600mA	1300mA	1100mA	880mA
Average current @ 50Ω antenna (typ.)	250mA	260mA	480mA	400mA	570mA	480mA
Average current @ total mismatch	250mA	260mA	480mA	400mA	570mA	480mA

Obrázek 3 - Odebíraný proud během vysílání při frekvenci 850 a 900 Mhz.

Mode	GSM call	GPRS Class 8	GPRS Class10		GPRS Class 12	
Timeslot configuration	1Tx / 1Rx	1Tx / 4Rx	2Tx / 3Rx		4Tx / 1Rx	
RF power nominal	1W (30dBm)	1W (30dBm)	1W (30dBm)	0.5W (27dBm)	0.5W (27dBm)	0.25W (24dBm)
Radio output power reduction with AT^SCFG, parameter <ropr>	<ropr> = 1 ... 3	<ropr> = 1 ... 3	<ropr> = 1	<ropr> = 2 or 3	<ropr> = 1	<ropr> = 2 or 3
Current characteristics						
Burst current @ 50Ω antenna (typ.)	1000mA	1000mA	1000mA	850mA	700mA	620mA
Burst current @ total mismatch	1000mA	1000mA	1000mA	850mA	700mA	620mA
Average current @ 50Ω antenna (typ.)	200mA	200mA	320mA	280mA	400mA	370mA
Average current @ total mismatch	200mA	200mA	320mA	280mA	400mA	370mA

Obrázek 4 - Odebíraný proud během vysílání při frekvenci 1800 a 1900 Mhz.

V jednom GSM rámcí je celkem 8 time-slotů, vysílací time-slot (během ostatních je odebíraný proud minimální) je 1 – 4 (viz přechozí tabulky). Budeme uvažovat pouze jeden vysílací time-slot v každém rámcí a frekvenci 1800 Mhz. Jeden time-slot je dlouhý 0,577 ms a rámec 4,6 ms, tzn., že vysoký odběr proudu

3 Prostředky pro realizaci zařízení

je jen jednu osminou vysílací periody a průměrný odběr proudu je tedy pouze $1,6 / 8 +$ odběr při přijímacím time-slotu (viz kolonka *average current*, Obrázek 4 - *Odebíraný proud během vysílání při frekvenci 1800 a 1900 Mhz., str. 22*). Zdroj se zatížitelností přes 1,6 A není potřeba. Vysílací burst lze pokrýt kondenzátorem s vyšší kapacitou (dle výpočtu, řádově jednotky mF). Zatížitelnost zdroje pak stačí pouze taková, která pokryje právě *average current*.

Napěťový rozsah umožňuje přímé napájení z Li-ion baterie. Zařízení navíc obsahuje nabíjecí obvod těchto baterií, vyžaduje pouze připojení ochranné diody s nízkým úbytkem, spínacího FET tranzistoru, snímacího odporu a termistoru pro kontrolu teploty baterie. Pro napájení nabíjecího obvodu je třeba 5 V (nabíjecí obvod funguje jako DC-DC měnič). S použitím baterie odpadá potřeba kondenzátorů, proudová špička se bez problémů pokryje. Maximální proud odebíraný z 5 V zdroje nepřesáhne *average current*.

3.2.2 SIM karta

Slot pro SIM kartu se připojuje pomocí šestice pinů na SMD konektoru, doplněním o několik kondenzátorů. Jeden z pinů slouží ke zjištění stavu vložení SIM karty. Modul podporuje 3 V i 1,8 V logiku SIM karet. Veškerou komunikaci se SIM, stejně jako registraci do sítě GSM obstarává modul automaticky.

3.2.3 Komunikační rozhraní

Modul je možné ovládat pomocí sady AT příkazů posílaných na jedno ze dvou asynchronních sériových rozhraní (ASC0 a ASC1). Dále obsahuje SPI, I2C a USB. Modul obsahuje i 10 GPIO, A-D a D-A převodník. Umožňuje také připojení analogového reproduktoru a mikrofonu, případně pak digitální zvukové rozhraní.

3.2.4 Java

TC65i má vlastní rozhraní pro program napsaný v jazyce JAVA. Software lze do modulu nahrát pomocí ASC nebo USB rozhraní. Výše uvedené rozhraní modulu je možné používat pro komunikaci mezi cílovou aplikací a programem napsaným v JAVĚ (kromě USB). Variantu s programem nahraným přímo v modulu nevyužívám z důvodů popsaných v části 4.2.2 *Procesor, str. 34*.

3.3 HW prostředky RS-485

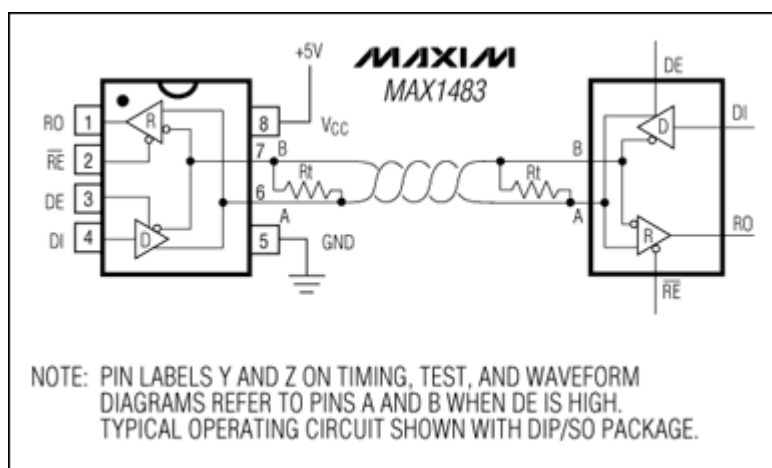
Pro realizaci komunikačního rozhraní sběrnice RS-485 nám stačí zařízení, které bude mít jeden pin pro příjem a druhý pro vysílání sekvence změn logických úrovní, reprezentujících sériovou komunikaci a druhé zařízení, které logické úrovně (LOW a HIGH) převede na napěťovou diferenci vodičů A a B.

První funkci, tedy sériovou komunikaci zajistí procesor s jednotkou USART. Jedná se HW periférii procesoru, která se postará o rychlost a správné časování komunikace, startbity, stopbity a paritu. V programu pak jednotce USART stačí předat jeden byte a ta zajistí, že bude odeslán, přijatý byte zase předá programu.

Druhé zařízení, které zajistí převod 2-stavového signálu na napěťovou diferenci a případně na stav vysoké impedance, zajistí budič sběrnice RS-485. Jako první jsem zvolil levný SN75176 od Texas Instruments. Bohužel jeho kvalita byla velice nízká. Proto jsem převodník nahradil typem MAX1483CSA+ od firmy MAXIM, tento převodník v klidu odebírá řádově 20 μA a umožňuje až 256 současně připojených stanic. Obvod má 8 vývodů, kromě napájení a připojení vodičů A a B má RO – Receiver output, RE – Receiver enable, DE – driver enable, DI – driver input. Receive output je výstup z přijímače (předává data ze sběrnice do procesoru), připojuje se k Rx (vstupní) pinu procesoru. Driver input je vstup do budiče (přijímá data z procesoru a předává je na sběrnici), připojuje se k Tx (výstupní) pinu procesoru. Pro řízení komunikace slouží piny RE – zapíná

3 Prostředky pro realizaci zařízení

přijímač a DE – zapíná vysílač. Vypínání přijímače má význam pouze při vysílání, ale není nutné (osobně jsem příjem ze sběrnice při současném vysílání s úspěchem využil pro zpětnou kontrolu toho, co na linku skutečně dorazí, více v části o software). Vypínat vysílač je nezbytné (jedná se o výše zmíněný stav vysoké impedance), současně na lince může být aktivní pouze jeden a to nejlépe maximálně po dobu nezbytně nutnou pro odeslání zprávy.



Obrázek 5 – Schéma zapojení MAX1483 řídicí jednotky

Sběrnice RS-485 pracuje na velké vzdálenosti a obecně je vhodné zařadit ochranu proti přepětí na vodičích. Vhodné jsou transily se zlomovým napětím 6,8 V zapojené mezi zem a vodiče A a B.

4 Vývoj zařízení - hardware

4.1 Měřicí modul

Pro návrh plošných spojů jsem používal software EAGLE. Vznikly celkem tři varianty měřicího modulu. Měřicí moduly popíšu jen ve zkratce, nejsou předmětem této práce. Vznikly zejména pro účely dalšího vývoje nebo na základě požadavků probíhající výstavby systému. Nicméně periférie jako A-D převodník a interface sběrnice RS485 byly odladěny na těchto modulech a do finální řídicí jednotky jen přeneseny.

4.1.1 Výběr procesoru

Pro první pokusy jsem zvolil procesor ATmega8. Disponuje 8 KB FLASH a 1 KB SRAM. Maximální výkon je 16 MIPS. Jedná se o starý procesor, který není doporučován pro nové aplikace. Navíc lze programovat pouze pomocí ISP, a proto není možnost aplikaci ladit, krokovat ani zařadit breakpointy. ATmega8 jsem použil v první variantě měřicího modulu, programoval jsem ho vlastním ISP programátorem.

Nicméně přístup typu „nahraju program a čekám, co se bude dít“ se mi zdál značně nepohodlný, zejména v situaci, kdy je třeba něco rychle ověřit a hlavně, když něco nefunguje, což nebyla neobvyklá situace. S AVR Dragon byla situace značně pohodlnější, ale musel jsem procesor ATmega8 nahradit novějším s podporou JTAG nebo debugWIRE. Vzhledem k nenáročnosti na paměť i na výkon jsem vybral nejnižší model z řady megaAVR ATmega48PA. Má 4 KB FLASH a 512 B SRAM. Dokáže až 20 MIPS. Procesor jsem použil ve druhé variantě modulu.

Ve třetí variantě jsem musel použít procesor se dvěma jednotkami USART. Zvolil jsem ATmega162 s 16 KB FLASH a 1 KB SRAM. Bohužel nemá A/D

převodník, ale modul jsem koncipoval tak, že lze doplnit externím A/D převodníkem na samostatné desce.

4.1.2 Napájení

K napájení integrovaných obvodů stačí 5 V stejnosměrných. V tunelu je instalováno napájecí napětí 24 V střídavých. První a druhou variantu modulu lze připojit přímo na střídavých 24 V. O usměrnění se stará usměrňovací můstek a 5 V lineární stabilizátor 78L05 s proudovou zatížitelností maximálně 100 mA. Třetí varianta modulu musí zajistit také napájení senzorů FIEDLER, které je v rozsahu 12 V do 24 V stejnosměrných. Proudová zátěž, zvláště při více senzorech, je na použití lineárního stabilizátoru trochu vyšší. Byla možnost použít stabilizátor 7812 a 7824 s vyšší zatížitelností, ale po dohodě s vedoucím práce jsme použili již zhotovený napájecí zdroj na samostatné desce. Zdroj má vstupní napětí 24 V střídavých a využívá DC-DC měničů a bude pro napájení ideální.

4.1.3 Varianta 1, s ATmega8

Modul vznikl s myšlenkou, že bude zcela univerzální a určen k zasazení do konkrétní aplikace, například desky s externím A/D převodníkem. Součástí modulu by byl pouze procesor, budič RS-485, stabilizátor, usměrňovací můstek, svorkovnice (napájení a RS-485) a dva konektory slučující univerzální vstupy, programovací piny ISP rozhraní a výstup napájení (24 V střídavých a 5 V stejnosměrných). Součástí ISP rozhraní je i sběrnice SPI, která potenciálně umožňuje připojení externího A/D převodníku k procesoru. V modulu jsem otestoval komunikaci po RS-485 a sběrnici 1-Wire.

4.1.4 Varianta 2, s ATmega48

Druhá varianta už je samostatný funkční modul se vstupy vyvedenými na svorkovnici. Deska je připravena pro zabudování do krabičky určenou na DIN lištu. Takže se jedná o kompletní zařízení, které stačí umístit na DIN lištu a

připojit. Součástí krabičky je svorkovnice, na kterou je vyvedené napájení, vodiče A a B a 6 univerzálních vstupů. Vstupy jsou přivedeny na piny procesoru, které mohou sloužit jako vstup pro integrovaný A/D převodník, nebo jako vstup logických úrovní. Na desce je konektor s programovacím rozhraním a piny se vstupy ze svorkovnic, pro použití externího A/D převodníku, který lze umístit přímo do krabičky. Na vstup převodníku se lze připojit pomocí svorkovnice. Pro upřesnění: každý vstup je na svorkovnici přiveden na pin A/D převodníku procesoru a zároveň vyveden na konektor, kterým lze připojit externí převodník. Komunikaci mezi A/D převodníkem a procesorem je opět možné realizovat pomocí sběrnice SPI, která je součástí ISP rozhraní. Navíc je potřeba počítat s použitím vstupů pro digitální senzor s TTL logikou. Zde je vhodné, někdy i nutné, použití pull-up nebo pull-down rezistorů. Problém je, že pro A/D převodník by rezistor způsobil zkreslení, proto jsem na desku umístil piny, které lze propojit zkratovací propojkou a tím odpor připojit. Pro každý ze šesti vstupů je jeden odpor. Napájení je řešeno pomocí usměrňovacího můstku, tantalového kondenzátoru a stabilizátoru. Je odolný proti přepólování. Na vstup napájecího napětí je zařazena tavná mini pojistka do DPS. V modulu jsem použil ještě budič sběrnice SN75176, který bych v případě reálného použití této varianty vyměnil za MAX1483. Pro ochranu proti přepětí jsem použil transily.

Modul je připraven pro snímání analogových hodnot, komunikace na jedné sběrnici RS-485. Po dohrání konkrétního algoritmu je možné připojit libovolný číslicový měřicí senzor pracující v TTL logice. Prozatím je realizována pouze 1-Wire sběrnice, jako určitý standard v jednoduchých číslicových teploměrech.

4.1.5 Varianta 3, s ATmega162

V průběhu řešení vznikl požadavek na použití senzorů firmy FIEDLER-MÁGR. Konkrétně jde o senzor pH a ultrazvukový snímač hladiny, ty komunikují pomocí RS-485. Teoreticky by mohlo jít připojit senzory přímo na hlavní sběrnici, ale je to spíše nežádoucí a u ultrazvukového senzoru je to

dokonce nemožné, protože nepodporuje protokol FINET a data vysílá pouze v textovém režimu. Proto je potřeba vytvořit most mezi sběrnicemi jednotkou, která bude kontinuálně vyčítat aktuální hodnoty jednotlivých měřených veličin několika senzorů současně připojených na jediné sběrnici a bude je ukládat do své paměti. Zároveň je nutno zajistit, že jednotka bude okamžitě vracet hodnoty na dotaz z hlavní sběrnice. Tím vzniká potřeba přítomnosti dvou jednotek USART v procesoru. Zcela jistě šel tento nedostatek řešit softwarovou emulací jednotky USART, nicméně bylo nutné upravit plošný spoj s ohledem na dvě sběrnice RS-485 a proto jsem rovnou přistoupil k použití procesoru se dvěma hardwarovými USARTy.

Rozdíl mezi třetí a druhou variantou je tedy ve dvojici dvou sběrnic RS-485 a nepřítomností A/D převodníku. Pro napájení se zde počítá se zdrojem 5 V na samostatné desce. Z toho důvodu není zařazen usměrňovací prvek a je použit pouze stabilizátor. Napájení navíc není vyvedeno na svorkovnici, ale je pouze drátky propojeno s napájecím zdrojem uvnitř krabičky. Na svorkovnici jsou pouze vodiče A a B obou sběrnic. Nutná je samozřejmě přítomnost dvou budičů, použitý je MAX1483. Na desku jsem umístil jak konektor ISP, tak JTAG. Programovat a ladit budu pomocí JTAG. ISP je pro případ, že by JTAG nějakým způsobem nefungovalo (ještě jsem s ním nepracoval), ale zejména kvůli SPI rozhraní. Tím je zařízení připraveno i na externí A/D převodník.

4.2 Řídící jednotka

Vývoj DPS řídící jednotky prošel pouze první, ladící variantou, na jejímž základě vznikla již finální deska. Nebudu první variantu detailně popisovat. V následujícím popisu druhé varianty pouze upozorním na opravené chyby z první verze. Centrem řídící jednotky je procesor XMEGA16A4, umožňuje komunikaci pomocí GPRS a sběrnice RS-485, navíc lze jednotku připojit přes USB k počítači, ze kterého je napájen a po kterém lze komunikovat přímo s TC65i. Zahrnuje vlastní li-ion baterii a slot pro SIM kartu. Jednotka je doplněna o čtyři

univerzální vstupy (analogové nebo logické) a výstupy. Napájí se z 9 - 24 V (stejnosemých nebo střídavých). Celá jednotka je v krabici na din lištu v šíři jednoho modulu.

4.2.1 Napájení

Napájení řídicí jednotky má svá úskalí. Je třeba napájet procesor XMEGA16A4, modul TC65i a budič MAX1483. Pro napájení TC65i jsem zvolil variantu s Li-ion baterií, která 5 V zdroj s proudovou zatížitelností kolem 0,5 A. Ze stejného zdroje lze napájet i MAX1483. Pro napájení procesoru lze s úspěchem použít napětí V_{ext} z modulu TC65i, které je 3 V a lze zatížit až do 50mA. Bohužel je zde několik problémů.

Vstupní napájecí napětí dosahuje hodnoty až 24 V střídavých, které se po usměrnění promění v hodnotu až 34 V stejnosměrných. Odběr 0,5 A a výstupní požadované napětí 5 V dělá použití lineárního stabilizátoru značně neefektivní, teoretický ztrátový výkon dosáhne hodnot až $(34 - 5) \times 0,5 = 14,5 \text{ W}$. V praxi bude sice odběr 0,5 A zcela výjimečný a střídavý průběh napětí způsobí menší rozdíl napětí, nicméně i desetinový odběr by potřeboval malý chladič a nebude efektivní. Přistoupil jsem proto na řešení napájení pomocí step-down spínaného zdroje. Napájecí rozsah se tak dostal na hodnotu 9 - 40 V a díky usměrňovacímu můstku lze použít i střídavého napětí. Vstupní napájení je do DPS přivedeno ze svorkovnice, hned na vstupu je SMD pojistka a transil se zlomovým napětím 36V. Deska je navržena tak, že lze osadit buď usměrňovací můstek, nebo ochranou diodu a propojovací odpor s nulovým odporem. Můstek se osadí v případě použití střídavého napětí, zatímco dioda v případě stejnosměrného napětí, pokud požadujeme společnou zem A-D převodníku a zem napájecího zdroje. Spínaný zdroj jsem realizoval pomocí obvodu MC34063, doplněný o několik pasivních součástek dle katalogového listu. Výstupní napětí 5 V je použito pro napájení nabíjecího obvodu li-ion baterie a budiče sběrnice RS-485.

Při výpadku nabíjecího¹ napětí nebude sběrnice funkční, což nevadí, neboť se nepočítá se záložní baterií u ostatních zařízení a nebude tedy s čím komunikovat.

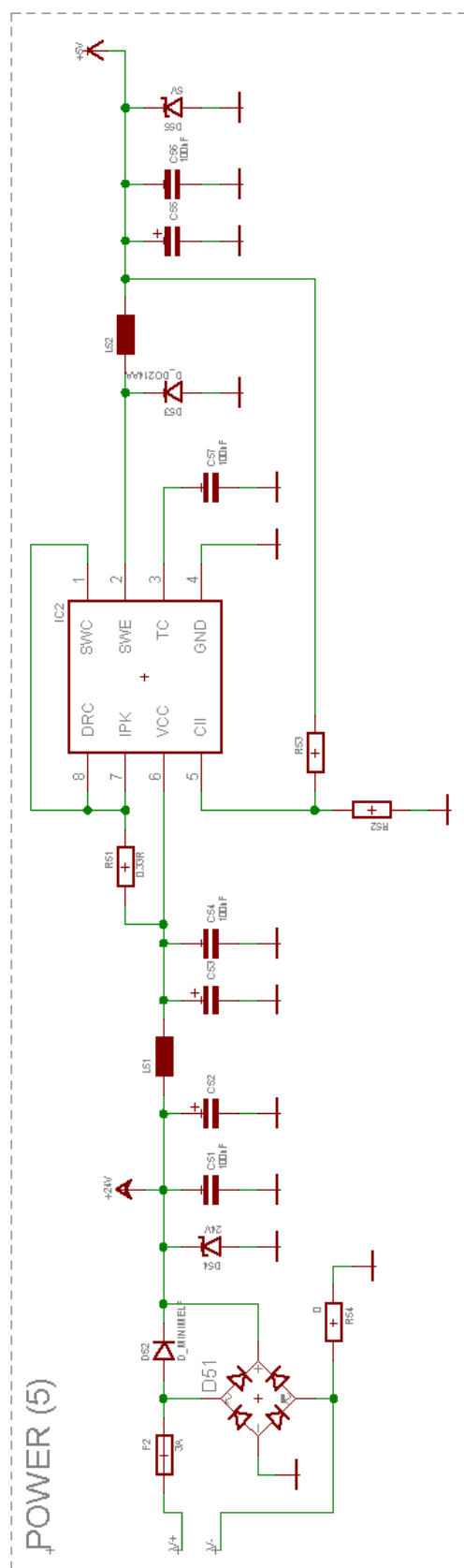
Druhý problém je v napájení procesoru, je nutné ho napájet i v případě výpadku nabíjecího napětí, protože bez CPU je celé zařízení nefunkční a bylo by tedy zbytečné baterii vůbec používat. Navíc, vzhledem k absenci jakýchkoliv tlačítek, musí procesor zajistit i zapnutí modulu TC65i. Zapíná se přesně definovaným impulzem na pin IGT. Napětí poskytované TC65i sice funguje i při výpadku nabíjecího napětí, problém ovšem je, že není k dispozici, pokud je TC65i v tzv. POWER-DOWN MODE, ve kterém se modul nachází, pokud je vypnutý (po připojení baterie, po vybití baterie) a kromě RTC není aktivní žádná funkce. Fakticky to znamená, že modul nepůjde zapnout, bude-li procesor napájen z V_{ext} .

V první variantě DPS jsem procesor připojil přes lineární stabilizátor přímo k baterii (kvůli napěťovým úrovním logiky TC65i, která odpovídá právě $V_{ext} = 3\text{ V}$). Procesor byl tedy aktivní jen s připojenou baterií bez ohledu na stav TC65i. Šlo o chybu, protože stabilizátor² měl spotřebu cca 1 mA a spolehlivě vybil baterii při delším odpojení napájení, i když byl TC65i vypnutý a dál vybíjel i po ochranném odpojení baterie modulem. Elegantní řešení se mi povedlo objevit spíše náhodou, zpětně jsem jej dohledal v datasheetu, ale vzhledem k nepřehlednosti celé dokumentace od Cinterionu se tomu ani nedivím. Pokud je modul v NORMAL MODE (aktivní, včetně přihlášení do GSM sítě), připojením nabíjecího napětí se začne dobíjet baterie. V případě, že TC65i je v POWER-DOWN MODE se po připojení nabíjecího napětí přepne do CHARGE-ONLY MODE, ve kterém se dobíjí baterie, GSM aktivní není, ale je již k dispozici V_{ext} a dokonce i sériová linka, modul pak lze do NORMAL MODE přepnout pouze AT příkazem.

¹ Nabíjecí napětí v dalším textu znamená vstupní napětí dc-dc měniče (9 - 24 V), potažmo nabíjecího obvodu modulu TC65i (5 V výstup z dc-dc měniče).

² přesto, že šlo o low-drop stabilizátor s malou spotřebou

V praxi situace tedy vypadá následně. Řídící jednotka se po připojení baterie dá zapnout pouze připojením nabíjecího napětí (5 V), respektive napájecího napětí (24 V) na svorkovnici zařízení. Modul se přepne do CHARGE-ONLY MODE, připojí se napájení na procesor i na MAX1483. Procesor hned na začátku běhu programu pošle na sériovou linku TC65i AT příkaz pro přepnutí do NORMAL MODE. Odpojením napájecího napětí v CHARGE-ONLY MODE by se TC65i vypnul, čímž by se odpojil i procesor. V NORMAL MODE a s připojeným napájecím napětím je aktivní GSM, procesor i sběrnice RS-485. V případě výpadku (odpojení) napájení nebude aktivní RS-485, ale modul TC65i zůstane v NORMAL MODE, čímž zůstane aktivní i napětí V_{ext} a procesor, dokud nedojde k vybití baterie a bezpečnému přepnutí modulu do POWER-DOWN MODE, které ukončí vybíjení baterie. Jedinou nevýhodou tedy zůstává nemožnost řídící jednotku zapnout bez napájecího napětí přivedeného na svorkovnici, byť baterie může být plně nabita. Vzhledem ke skutečnosti, že baterie je zde pouze záložní (počítá se s trvalým připojením na napájení), je nevýhoda irelevantní.



Obrázek 6 – Schéma zapojení napájecího zdroje řídicí jednotky

4.2.2 *Processor*

První varianta byla pokusná, nepočítal jsem se sběrnici RS-485 a nepotřeboval dvojici USARTů, použil jsem procesor MEGA48. Ve druhé variantě byly dvě jednotky USART nutné. Nechtěl jsem znovu použít procesor MEGA162, jeho cena je neúměrně vysoká, je schopen udělat maximálně 16 MIPS (při 3 V to bude dokonce tak polovina) a při použití v měřicím modulu jsem narazil na problém s dostupností (Ten mimochodem poznamenal celý sortiment firmy Atmel. Poměrně mě zaskočilo, že i v dnešní době může takový problém nastat a je poměrně důležité na to brát ohled, není nijak nereálná situace, kdy konkrétní součástky nelze sehnat třeba půl roku). Použil jsem procesor z řady AVR XMEGA, konkrétně ATxmega16A4. Obvod je za cenu nižší než ATmega162, nicméně disponuje 5 jednotkami USART, 12-bitovým A-D převodníkem a rychlostí až 32 MIPS, navíc má maximální napájecí napětí 3,3 V, takže výkon lze využít naplno s 3 V z Vext.

Hodinový signál je získáván z interního RC oscilátoru, který se automaticky kalibruje s 32,768 khz externím oscilátorem. Procesor disponuje velice rozsáhlým systémem pro distribuci hodinového signálu. Kromě 32Mhz interního oscilátoru má i 2 Mhz, 32 khz a 32,768 khz oscilátor. Je možné připojit kromě externího 32,768 khz hodinového oscilátoru i až 16 Mhz krystal. Disponuje násobičem (zvolený zdroj dokáže 1-31x násobit) a i 11-bitovou předděličkou (dělit až 4096x), navíc je možné hodinový signál dělit dvěma nebo čtyřmi pro jednotlivé skupiny periférií zvlášť.

Ovládání vstupních/výstupních pinů je také značně vylepšeno oproti megaAVR. Stejně jsou registry DDR (definuje pin jako vstupní nebo výstupní), PORT (v režimu výstupního pinu nastavuje jeho logickou úroveň, v případě vstupního připojuje interní pull-up rezistor) a PIN (logická úroveň pinu). Jedná se o 1-bytové registry (8 bitů, 8 pinů, 1 port). AVR XMEGA má registry se stejnou funkcí s názvy DIR (DDR), OUT (PORT), IN (PIN). Jsou však doplněny porty DIRSET, DIRCLR, DIRTGL (OUTSET, OUTCLR, OUTTGL). Zápis jedničky do

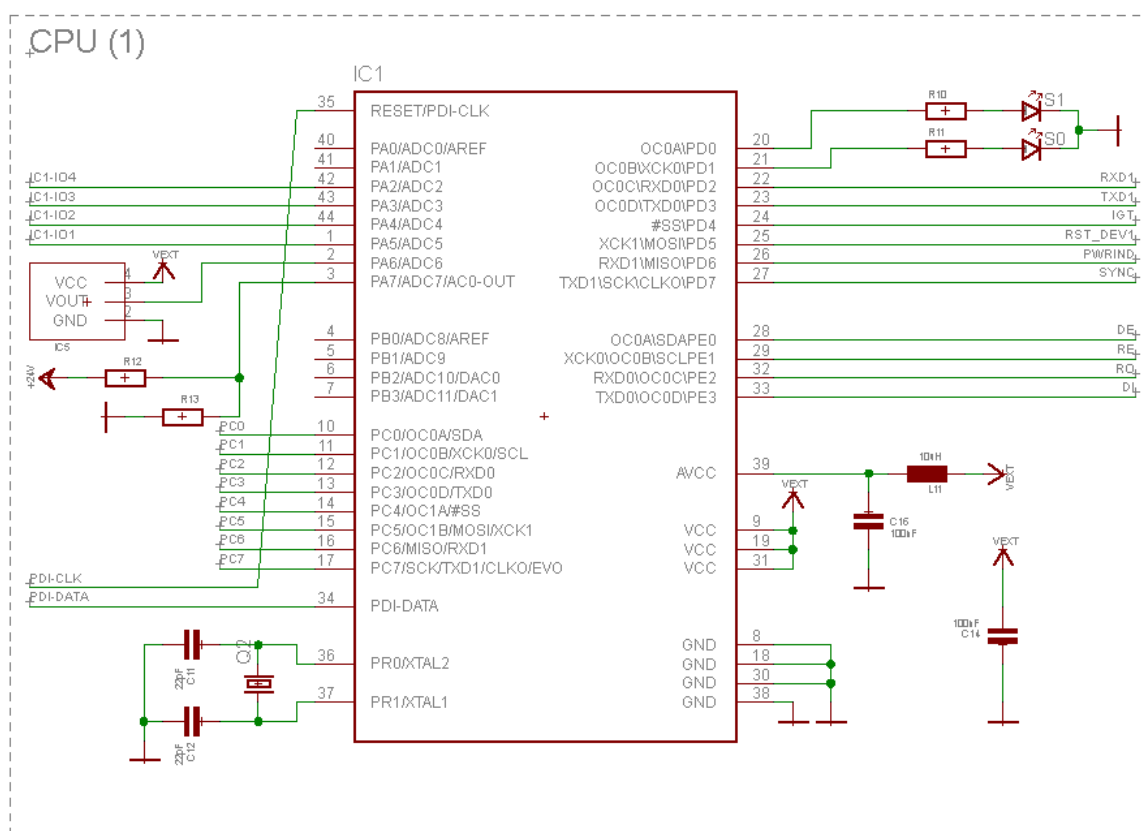
bitu registru OUTSET nastaví odpovídající pin na úroveň HIGH, zápis 0 neprovede nic. OUTCLR naopak při zápisu jedničky příslušný pin nastaví na 0. Zápis do OUTTGL změni úroveň odpovídajícího pinu na opačnou (přepne). V programu je pak možné vynechat logický součin (součet) k nastavení konkrétního bitu v bytové registru. V praxi je ulehčení patrné zejména u přepnutí hodnoty, kdy nemusíme softwarově zjišťovat stav a teprve následně přepínat. Další výhodou je možnost ke každému pinu připojit interní pull-up nebo pull-down rezistor, případně logické zapojení jako bus-holder.

Nebudu dále popisovat jednotlivé části. AVR XMEGA je velmi dobře vybavený 8-bitový MCU. Disponuje víceúrovňovým systémem přerušení s možností nastavení priority, DMA (přímý přístup do paměti), Event System (umožňující vyvolání a reakce jednotlivých periférií bez zásahu SW) nebo i rozhraním pro připojení externí paměti SDRAM.

Na začátku vývoje řídicí jednotky jsem se musel rozhodnout, zda k řízení aplikace použiji externí procesor nebo prostředí pro jazyk JAVA přímo v TC65i. Vždy jsem se přikláněl k procesoru, už proto, že jsem chtěl zachovat možnost jednoduchého přesunu knihoven, které mám již napsané, a věnovat se především mikroprocesorovým aplikacím psaným v jazyce C. Druhou stránkou věci bylo, že možnost ovládat aplikaci přímo pomocí modulu TC65i je úspornější, navíc JAVA v modulu běží na 32-bitové procesoru s frekvencí 400 Mhz. Právě rychlost rozhodla o použití 8-bitového procesoru s desetinovou frekvencí. JAVA by možná měla navrch ve složitém výpočtu, ale při komunikaci po sběrnici RS-485, kde je nutné při změně směru komunikace zapínat a vypínat vysílač budiče pomocí změny napětí na pinu DE (*viz část 2.3 HW prostředky RS-485, str. 24*) bylo přepnutí tak dlouhé, že docházelo ke ztrátě dat. Program nejdříve GPIO pin modulu připojený na DE budiče nastavil na úroveň high, poté začal vysílat data po sériové lince připojené k budiči, ale než se změna úrovně projevila, čímž se vysílač zapnul, několik bajtů již bylo odesláno a na sběrnici se nedostali.

Chvilí jsem uvažoval i o variantě programu v TC65i i v procesoru (HW je proto tedy připraven). Procesor by měl zajistit komunikaci po RS-485 a program v JAVĚ zajistit GPRS komunikaci, nicméně díky IP STACKu ovládanému AT příkazy v TC65i usuzuji, že jednodušší bude vše řídit pouze procesorem.

Napěťové úrovně procesoru a TC65i jsem řešil výše uvedeným způsobem. Logická úroveň high u T65i představuje 3 V, low samozřejmě 0 V. Napěťové úrovně procesoru jsou dány jeho napájecím napětím, v tomto případě tedy $V_{ext} = 3 \text{ V}$.



Obrázek 7 - Schéma zapojení procesoru řídící jednotky

4.2.3 Cinterion TC65i

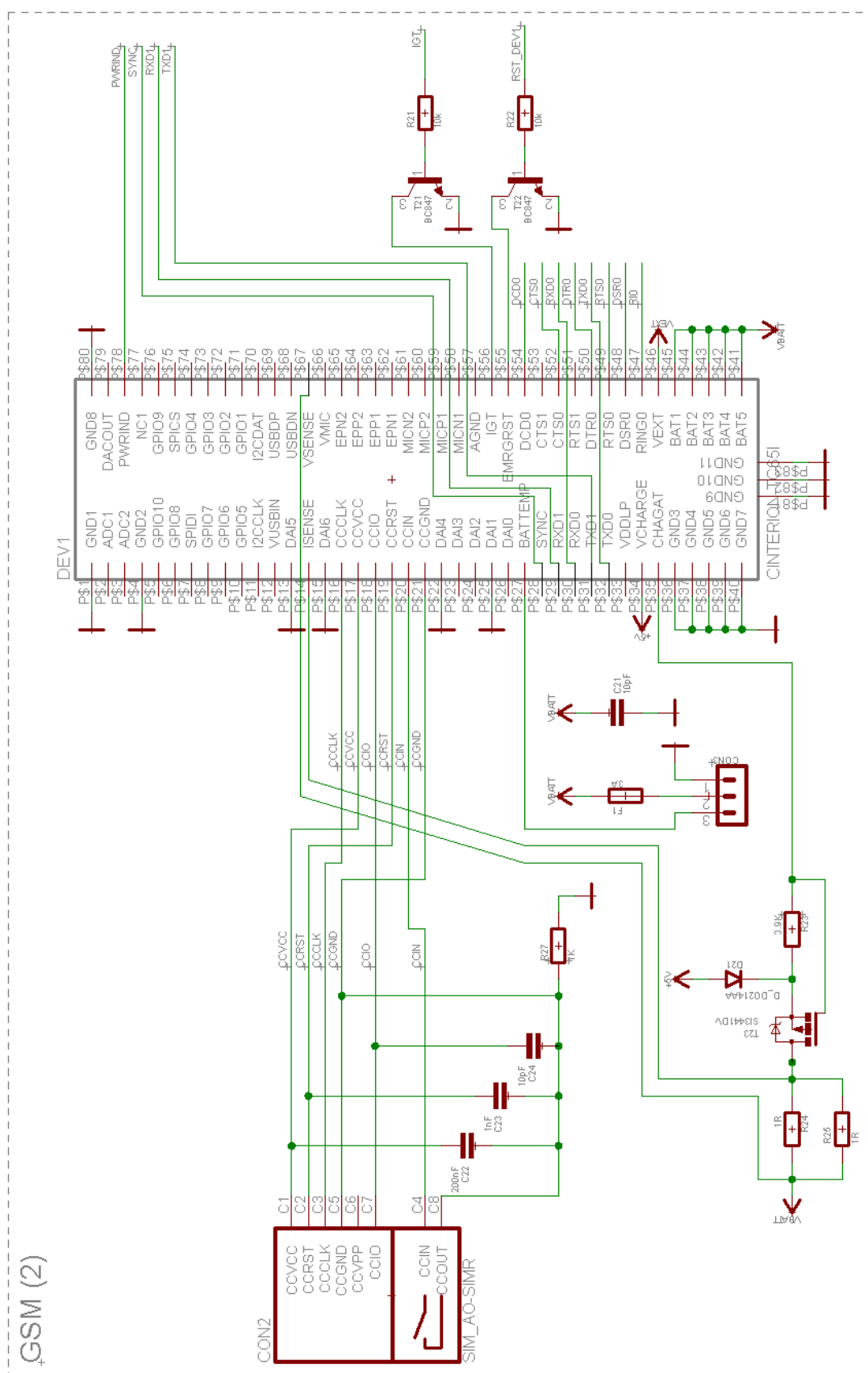
Do DPS je modul přidělán třemi šrouby a konektorem. Kromě připojení periférií jako je CPU a USB je třeba připojit SIM kartu, baterii a anténu.

Jak jsem se již zmínil v popisu TC65i, připojení SIM karty nepředstavuje žádný problém. Je použit slot na SIM s výsuvnou zásuvkou, kartu lze vyndat i zandat při zapnutém zařízení.

Baterie se k modulu připojuje dle katalogového zapojení spolu s externím tranzistorem a snímacím odporem, jehož hodnota zároveň určuje nabíjecí proud. Podrobnosti k napájení viz část 3.2.1 *Napájení*, str. 21. Doplním snad jen, že napájecí (ty, které přivádí napětí z baterie) a zemní piny jsou vždy po pěti vedle sebe a je třeba jich plně využít, protože špičkově může proud jimi tekoucí dosahovat téměř 2 A (viz část 3.2.1 *Napájení*, str. 21). Baterie, kterou jsem zakoupil má bohužel integrovaný termistor s jinou hodnotou než je třeba pro TC65i. Proto jsem termistor se správnou hodnotou přilepil k baterii zvenku. Zapojení je doplněno pojistkou před baterií.

Anténu jsem připojil přes redukci malého konektoru TC65i na standardní konektor GSM antény a použil obyčejnou úhlovou anténou. Úroveň signálu je o cca 30% lepší než u standardního mobilního telefonu s integrovanou anténou.

Dle doporučení v dokumentaci jsem piny IGT a RST připojil přes tranzistor.

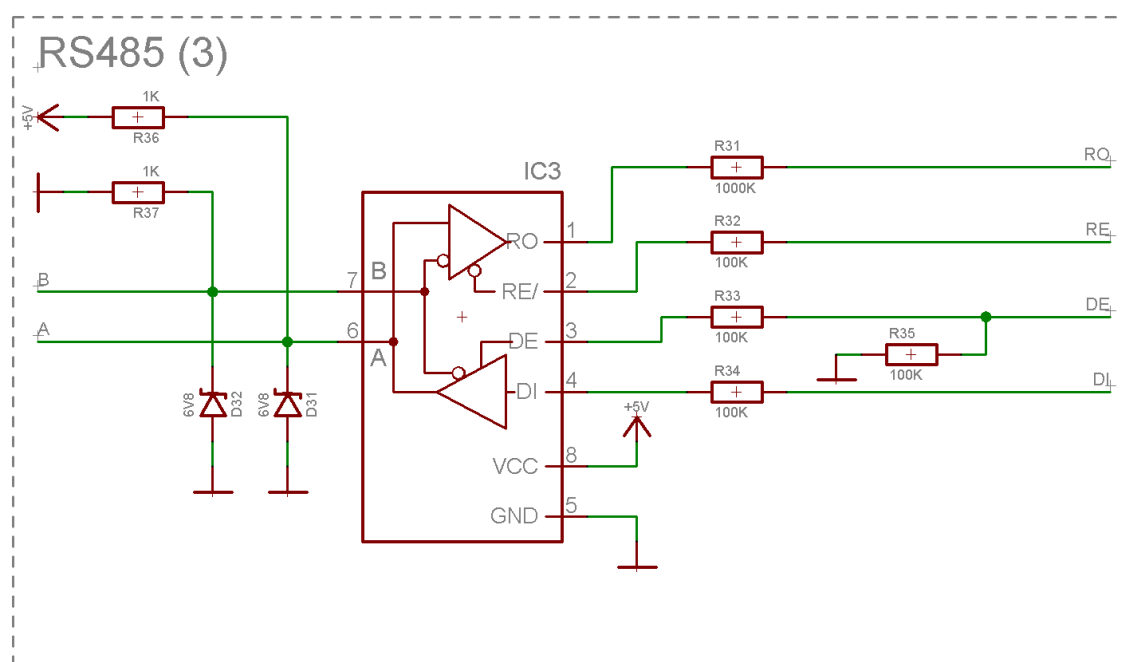


Obrázek 8 - Schéma zapojení modulu TC65i řídicí jednotky

4.2.4 RS-485

Sběrnice RS-485 je, jak z předchozího textu vyplývá řešena budičem firmy MAXIM-IC MAX1483, který je připojen na USART procesoru. Původně jsem chtěl zařízení doplnit o galvanické oddělení sběrnice, nicméně vzhledem ke složitosti řešení jsem od toho upustil. Zařízení navíc bude napájeno ze společného zdroje spolu se všemi ostatními zařízeními, takže by neměl být problém se zemními smyčkami. Zapojení obvodu je doplněno dvojicí transilů pro ochranu pinů A a B a obdobu pull-up a pull-down odporů pro definování logické úrovně na lince v případě kdy žádné zařízení nevysílá (*viz část 2.2 Sběrnice RS-485, str. 13*). Poslední verze zapojení je doplněna i pull-down rezistorem na pin DE, protože v případě nedefinovaného stavu na pinu procesoru (například v důsledku poruchy) by mohlo dojít k zapnutí vysílače a tím zablokování sběrnice.

Napájecí napětí pro MAX1483 je 5 V, již jsem uvedl, že ho napájím z 5 V výstupu DC-DC měniče, tedy nebude funkční při výpadku napájecího (nabíjecího) napětí. Problém ovšem je s napěťovými úrovněmi budiče a procesoru. Nicméně logickou jedničku procesoru, tedy 3 V, budič vyhodnotí jako logickou jedničku a logickou jedničku budiče (5 V) srážím odporem na nižší úroveň.



Obrázek 9 – Schéma zapojení MAX1483 řídicí jednotky

4.2.5 USB

Modul TC65i disponuje vlastním USB rozhraním, které slouží pro připojení modulu k PC. Po připojení je možné nahrát do modulu JAVA program a pomocí terminálu lze s modulem komunikovat pomocí AT příkazů (vhodné při odzkoušení příkazů před implementací do CPU). Komunikaci mezi aplikací v JAVĚ a PC integrované rozhraní neumožňuje. V mém případě nejde o problém, nicméně jsem během návrhu nebyl přesvědčen, že JAVU využívat nebudu a byla by škoda případné USB rozhraní nevyužít i pro komunikaci cílové aplikace s PC (nastavení parametrů).

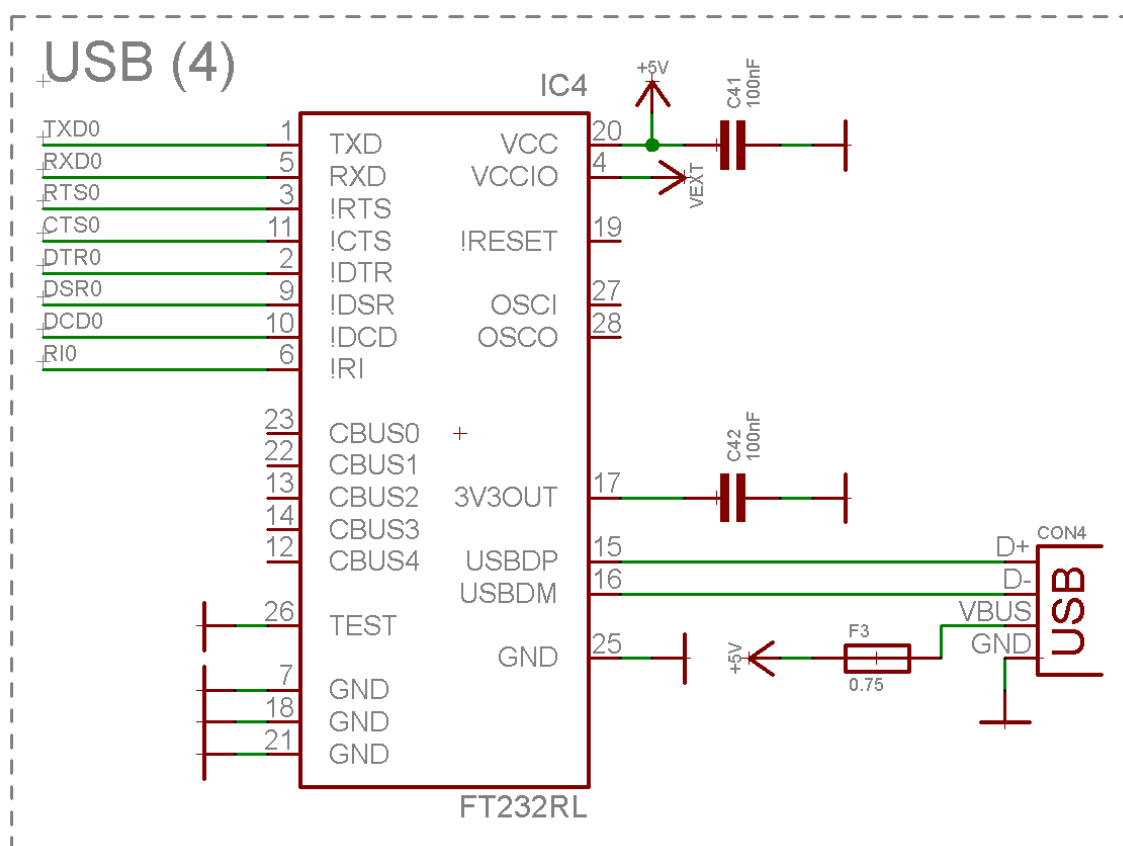
Bohužel jediný způsob bylo využít rozhraní ASC0 nebo ASC1 (jeden z nich bude připojený k procesoru, druhý volný). Jako převodník mezi sériovým rozhraním a USB jsem použil obvod FTDI FT232R. Jde o velice jednoduchý způsob implementace USB rozhraní do mikroprocesorové aplikace. Na DPS se FTDI připojí k procesoru (v mém případě k TC65i) obdobně jako například MAX232, v PC se nainstaluje ovladač dodávaný k čipu a vytvoří se virtuální COM port. V praxi tedy v PC softwarově nepoznáme rozdíl mezi připojením

přes FTDI nebo pomocí portu COM (snadný přístup z PC aplikace), ale řeší to častý problém s absencí COM portu na dnešních PC.

Zapojení je téměř shodné s doporučeným zapojením z katalogového listu. Obvod napájím z 5 V. Výhodné je, že FTDI má samostatné napájení napěťových úrovní, ty napájím z Vext a korespondují s napěťovými úrovněmi TC65i.

Použití USB je v konečném důsledku zbytečné. Modul TC65i neprogramuju, ani nepotřebuji komunikaci modulu s okolím, jediný význam měl při odzkoušení AT příkazů.

Využil jsem i možnost napájení celého zařízení z USB, po připojení zařízení k USB je napětí 5 V z PC spojeno s výstupním napětím dc-dc měniče. Samozřejmě, že v tomto případě je nutné odpojit napájecí napětí zařízení. Je možné napájení z USB přerušit neosazením pojistky.

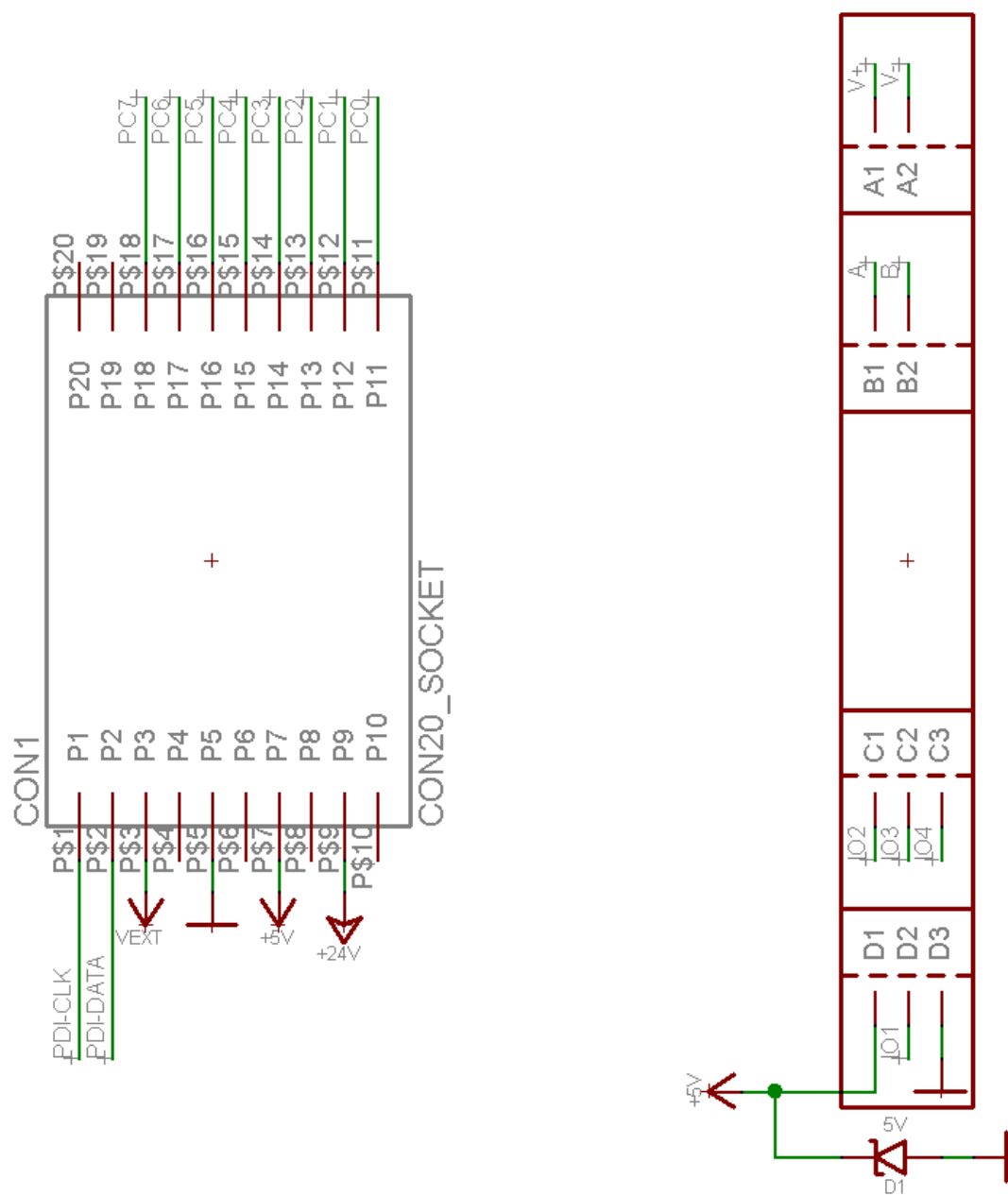


Obrázek 10 – Schéma zapojení FTDI čipu řídicí jednotky

4.2.6 Konstrukční řešení

DPS řídící jednotky, včetně baterie a antény je umístěna v univerzální krabičce na lištu DIN, široké 17,5 mm. Zařízení obsahuje celkem 12 šroubových svorek, přičemž dvě z nich jsou nevyužity kvůli úspoře místa na DPS. Upevnění a umístění desky v krabičce jsem musel řešit atypicky k dosažení větší integrace. Na čelní straně krabičky je slot pro SIM kartu, USB konektor a dvojice LED diod. Schématickou značku svorkovnice jsem nakreslil jako krabičku v půdorysu (Obrázek 11, vpravo). Programátor se k zařízení připojuje pomocí 20-pinového SMD konektoru na desce (Obrázek 11, vlevo). Je nedostupný při zavření krabičky (krabička je na zacvaknutí), není to praktické, ale bylo záměrem schovat konektor. Umožňuje také připojení dalších periférií k procesoru (druhou DPS) a poskytuje všechna napájecí napětí.

Krabička je doslova plná, vše se tam vešlo velmi natěsno. V případě rozšiřující DPS by se musela použít širší krabička. Existují dvě provedení, takže to není problém.



Obrázek 11 - Schéma zapojení SMD konektoru a svorkovnice

5 *Vývoj software*

5.1 *Jazyk C*

Software jsem se rozhodl psát v jazyce C. V podstatě jsem s ním poprvé začal dělat až při této práci, jsem s ním zcela spokojený a myslím, že v průběhu realizace jsem se s ním naučil pracovat na lepší úrovni než jsem doposud pracoval s jazykem PASCAL.

5.2 *Projekt*

Celý program je rozdělen do několika samostatných modulů skládajících se ze zdrojových a hlavičkových souborů. Zdrojový soubor obsahuje vlastní program, hlavičkový předem známé konstanty, které lze snadno změnit bez zásahu do samotného kódu modulu a deklarace proměnných a funkcí přístupných v celém programu i mimo modul. Modul se do výsledného programu vkládá prostřednictvím odkazu na hlavičkový soubor. Zdrojový soubor musí obsahovat odkaz na svůj hlavičkový.

Pro přehlednost jsem zvolil jednoduché pořadí psaní jednotlivých částí kódu. Pro soubor typu .c platí:

- systémové .h soubory
- vlastní .h soubory
- interní globální proměnné
- interní definice konstant a maker
- externí funkce
- interní funkce
- main

Pro soubor typu .h platí:

- systémové .h soubory

- vlastní .h soubory
- externí globální proměnné
- externí definice konstant a maker
- funkční prototypy externích funkcí

Systémové soubory jsou součástí balíčku WinAVR. Vlastní soubory, jsou mé vlastní moduly. Globální proměnné jsou deklarovány na začátku každého souboru a jsou platné po celou dobu existence programu, lokální jsou deklarovány uvnitř funkce a s koncem funkce zanikají. Externí proměnné, definice a funkce jsou dostupné ve všech modulech, které obsahují modul, v němž jsou deklarovány. Interní proměnné, definice a funkce jsou dostupné pouze v rámci modulu, v němž jsou definované. V hlavičkových souborech jsou deklarovány externí proměnné, ale z funkce je zde uveden pouze funkční prototyp (název funkce, vstupní proměnné, datový typ vstupních proměnných, datový typ výstupní hodnoty), celá funkce musí být napsaná v příslušném .c souboru. Mým cílem bylo každý modul koncipovat tak, aby šel kdykoliv vložit do jakéhokoliv projektu. Pro začlenění stačí pouze definovat základní konstanty a vstupní a výstupní funkce a proměnné modulu, kterými se předávají příkazy a parametry (externí) lze snadno vyčíst z .h souboru. Zbytek (interní) mě v situaci, kdy je modul kompletní a odladěný nemusí zajímat.

Hlavní modul je main.c, obsahuje hlavní funkci main, kterou program volá jako první. Funkce musí obsahovat úvodní jednorázovou inicializaci a nekonečnou smyčku typu while(1). Moje představa ideálního programu je stav, kdy v nekonečné smyčce není žádný kód a vše je řešeno pomocí přerušení. Kdykoliv je pak možné doplnit jakýkoliv program do smyčky. Takový stav se mi nepovedlo docílit a v konkrétní situaci mi přidělalo více zbytečné práce, než užítku.

5.2.1 Moduly

V textu nebudu podrobně popisovat zdrojový kód, je přiložen na CD. Stručně shrnu jednotlivé moduly a významnější funkce.

5.2.2 main.c

Tento soubor nemá svůj hlavičkový soubor. Provádí základní inicializaci voláním funkce Init (z modulu init.c).

Po inicializaci zapne přerušení a dostane se do nekonečné smyčky, ve které cyklicky zasílá dotazy na sběrnici rs-485 pomocí funkce BusSend (modul rs485.c) a reaguje na případnou URC zprávu pomocí funkce MobileIdle (modul usart.c).

5.2.3 init.c

V hlavičkovém souboru jsou definovány makra, zjednodušující ovládání některých výstupů. Jsou dostupné i mimo modul.

V základním souboru jsou definovány funkce pro nastavení stavu diod a některých výstupních pinů.

Hlavní funkce je funkce Init(). Definuje výstupní piny, nastavuje zdroj hodinového signálu, jeho předděličku atd. Volá funkce pro inicializaci dvou USARTů (funkce ze souborů usart.c a rs485.c).

5.2.4 usart.c

Hlavní modul celého programu. Kompletně obsluhuje GSM modul pomocí AT příkazů.

Základní funkcí je SendString, které se jako parametr předá řetězec a jeho velikost a funkce se postará o jeho odeslání. Spolupracuje se dvěma zdroji přerušení. USARTD0_DRE_vect se vyvolává při uvolnění zásobníku pro odchozí data (1 byte), v obsluze se odešle další v pořadí. Spolupracuje i s USARTD0_RXC_vect, které je vyvoláno při příchozím bytu. Obsluha tohoto

přerušeni zajišťuje příjem zprávy a změnu proměnné oznamující novou zprávu, při vysílání obstarává kontrolní příjem (přijímá echo a porovnává s odeslanými daty).

Funkce `SendCommand` zajistí kompletní odeslání příkazu s čekáním na odpověď a potvrzením výsledku svou návratovou hodnotou. Bohužel během čekání na odpověď blokuje běh programu (který ale stejně většinu času čeká v nekonečné smyčce, takže pro tento konkrétní projekt to není na závadu).

Funkce `SendCommandNoReply` je obdobou předchozí funkce, ale svou návratovou hodnotu vrátí již po odeslání příkazu a úspěšném přijmutí echa. Nečeká na odpověď. Rozdíl oproti funkci `SendString` je ten, že funkce čeká na dokončení odeslání a potvrzení příkazu. `SendString` zahájí odesílání, které probíhá samo pomocí přerušeni, takže program pokračuje v běhu ještě před odesláním celého příkazu. Je to výhodné pro efektivitu programu, ale nevýhodné v případě, že musíme čekat na odpověď.

Funkce `FtpInit` nastavuje parametry pro připojení modulu k internetu a FTP serveru. `FtpSend` zajistí odeslání dat na FTP server. `GetTime` je funkce, která z časového serveru vyčte přesný čas.

Poslední důležitou funkcí je `IsUrc()`, které se předá přijatá zpráva a funkce vyhodnotí, zda se jedná o URC zprávu a případně zareaguje. Tímto způsobem se modul i zapíná. Připojení na napětí a následné automatické přepnutí do CHARGE-ONLY MODE vyvolá zprávu `^SYSSTART CHARGE ONLY MODE`, na kterou funkce reaguje odesláním příkazu pro zapnutí.

5.2.5 rs485.c

Modul obstarává komunikaci po sběrnici RS-485. Funkce `BusSend`, které se předá pouze adresa cílového zařízení a obsah zprávy, se postará o sestavení paketu protokolu FINET a jeho zaslání.

Přijaté pakety jsou rozpoznány již v obsluze přerušení příjmu a do výstupní proměnné se předávají pouze čistá data. Na základě přijaté zprávy o teplotě je v obsluze přerušení volána funkce `FtpSend`, která hodnotu ukládá na FTP server.

V případě, že by řídicí jednotka nebyla jako MASTER a mohlo by dojít k přijetí paketu i bez předchozího vyžádání, tedy například při obsluze FTP servu, musela by funkce být volaná až po ukončení obsluhy přerušení, protože do té doby jednotka nepřijímá žádné nové zprávy.

5.2.1 err.c

Modul sloužící k detekci chyb. Obsahuje externí funkci `ErrMsg`, která je volána, všude tam, kde chyba hrozí. Jejím parametrem je kód chyby. Na jehož základě je chyba vyhodnocena. Navíc je kód chyby ukládán do EEPROM a lze jej zpětně vyčíst.

6 Závěr

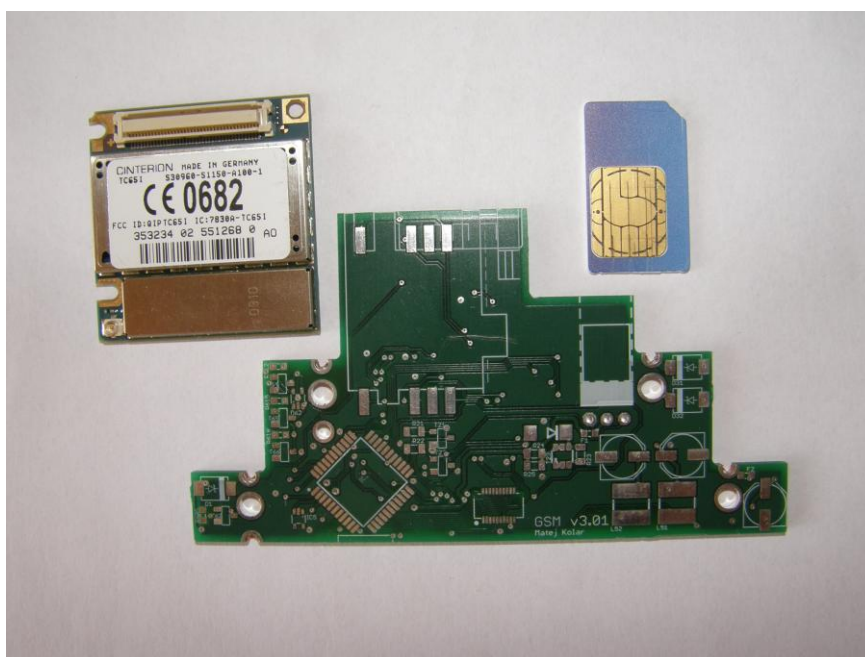
Finální vzorové řešení si vyžádalo několik oprav desky plošných spojů a mnoho způsobů naprogramování.

Hardware bylo poměrně obtížné vměstnat do krabičky a na jednu desku plošných spojů, nicméně se to podařilo. Osazení desky proběhlo bez problémů, pouze u některých z variant měřící jednoty jsem použil špatné tavidlo a zařízení spolehlivě přestalo do několika dnů fungovat. Osazování řídicí jednotky bylo mírně náročnější kvůli SMD konektoru modulu Cinterion, ale šlo to zvládnout. Během jednoho osazení jsem si tavidlem zanesl mechanismus výsuvu SIM karty, ale šlo odstranit ethanolem.

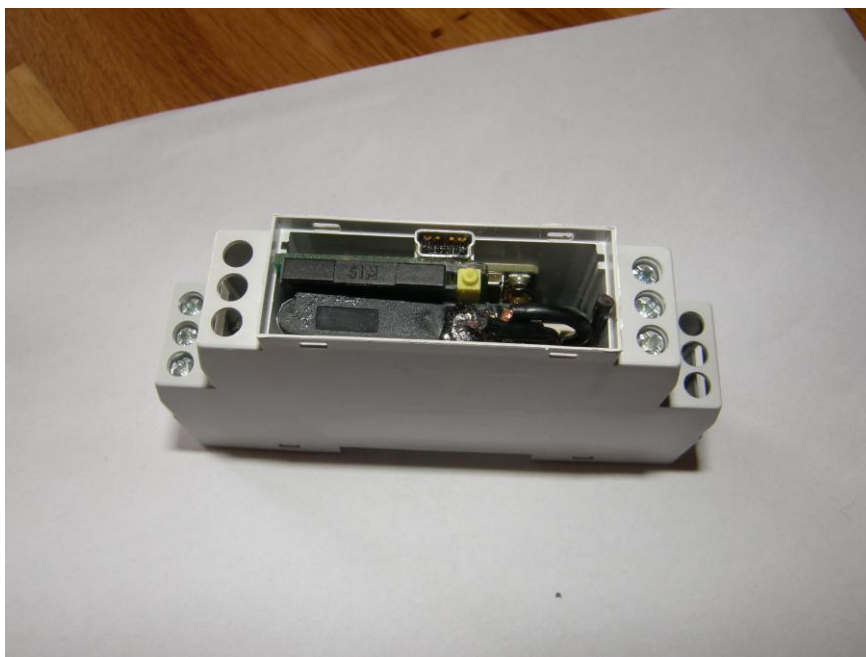
Psaní SW jsem původně pojal příliš obecně. Chtěl jsem udělat co nejuniverzálnější řešení, přenositelné do jakéhokoliv hotového programu, tak aby do stávajícího SW nebylo nutné téměř zasahovat. Po určité době jsem, ale dospěl k závěru, že než jedno takové univerzální řešení je časově náročnější než několik samostatných projektů řešených jednoduše a přímo. Nemluvím ani o nepřehlednosti takového řešení.

Vznikl program splňující požadavky obsluhy GSM modulu pomocí AT příkazů a možnost komunikace po sběrnici RS-485. Nelíbí se mi, že například během čekání na reakci na AT příkaz, program čeká mimo modul main v nekonečné smyčce. Program tak vlastně stojí a nic nedělá a dělat nemůže. V této aplikaci je problém vyřešen poměrně krátkou obsluhou svého druhého úkolu (RS-485) pomocí přerušení. Složitější program, s nutností obsloužit dvě podobně náročné (spíše nevhodné pro MCU) komunikace (zasílání a příjem AT příkazů), by se už nedal řešit dlouhým čekáním v nekonečné smyčce na dvou místech. Situaci bych pravděpodobně řešil algoritmem obdobným z operačního systému, případně systému reálného času. Místo nekonečné smyčky by bylo možné uprostřed nekonečné smyčky uložit stav zásobníků a z funkce vyskočit, přerušení by pak příslušné data obnovilo a vrátilo do funkce.

Vzorové řešení řídicí jednotky jsem odzkoušel pomocí provizorního měřicího modulu a veřejného FTP serveru. V protokolu FINET jsem implementoval pouze odesílání (z měřicího modulu) a příjem (řídicí jednotka) teploty ostatní veličiny jsou vyčítány téměř stejně, je ale možné že by se musel algoritmus vyčtení dat z paketu mírně pozměnit (upravit velikost proměnné). Každý údaj ukládaný na server je doplněn časovou značkou z časového serveru. Ve finální podobě jednotka fungovala bez problémů.



Obrázek 12- Neosazená DPS, modul Cinterion TC65i a SIM karta



Obrázek 13- Řídící jednotka v krabičce, bez čelního panelu



Obrázek 14 - Pohled do tunelu

Přílohy

Na CD je přiloženo

- Schéma zapojení řídicí jednotky v programu EAGLE
- Motiv desky plošných spojů řídicí jednotky v programu EAGLE
- Kusovník v MS EXCEL
- Tabulka pro výpočet BAUDRATE v MS EXCEL
- Kompletní zdrojový kód

Literatura

- [1] *Katalogový list ATMEL ATxmega 16A4*
http://www.atmel.com/dyn/resources/prod_documents/doc8069.pdf
- [2] *Katalogový list ATMEL AVR XMEGA A*
www.atmel.com/dyn/resources/prod_documents/doc8077.pdf
- [3] *Cinterion TC65i Datasheet*
- [4] *TC65i Hardware Interface Description*
- [5] *TC65i AT Command Set*
- [6] *Application Note 07: Rechargeable Lithium Batteries in GSM*
- [7] *Application Note 26: Power Supply for GSM Applications*
- [8] *Katalogový list MAX1483*
<http://datasheets.maxim-ic.com/en/ds/MAX1482-MAX1483.pdf>
- [9] *Katalogový list FTDI FT232R*
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [10] *Katalogový list MC34063*
http://www.datasheetcatalog.org/datasheets/2300/501448_DS.pdf
- [11] <http://www.atmel.com>
- [12] <http://winavr.sourceforge.net/>
- [13] <http://www.maxim-ic.com/>
- [14] <http://cs.wikipedia.org/wiki/RS-485>
- [15] <http://hw.cz/docs/rs485/rs485.html>
- [16] <http://www.fiedler-magr.cz/>