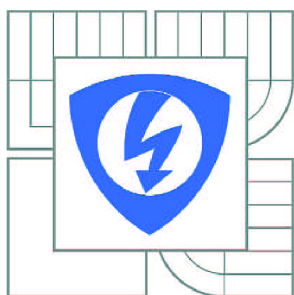


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POUŽITÍ GPRS MODULU PRO PŘENOS JPG SOUBORŮ V REÁLNÉM ČASE

APPLICATION OF GPRS MODULE FOR REAL-TIME TRANSMISSION OF JPG FILES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

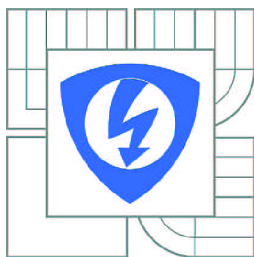
Bc. JAN RUSINSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. OTTO DOSTÁL, CSc.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Rusinský

ID: 78411

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Použití GPRS modulu pro přenos JPG souborů v reálném čase

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte interface mezi kamerou ITM-C-328 a GPRS modulem FastTrack pro zobrazení WWW stránky s aktuálním obrázkem snímaným kamerou. Na stránce bude možné ovládat pohyb servomotoru umožňujícího natáčení připevněné kamery. Zařízení bude osazeno mikrokontrolérem ATMEL a ovládací program bude napsán v jazyce "C". Napájení bude z 12V baterie, mikrokontrolér bude zároveň řídit nabíjení bezúdržbového olověného akumulátoru. Projekt řešte ve spolupráci s pracovníky firmy Honeywell.

DOPORUČENÁ LITERATURA:

[1] Mann, B. : C pro mikrokontroléry, BEN, 2003. ISBN 80-7300-077-6

[2] Kalarakis, M. : Circuit Cellar #228 - Web camera design, July 2009. ISSN 1528-0608, str. 26

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

Vedoucí práce: Ing. Otto Dostál, CSc.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Použití GPRS modulu pro přenos JPG souboru v reálném čase

diplomová práce

Studijní obor: Telekomunikační a informační technika
Student: Jan Rusinský
Vedoucí práce: Ing. OTTO DOSTÁL, CSc.

Abstrakt :

Tento dokument popisuje postup návrhu rozhraní mezi VGA kamerou a GSM/GPRS modemem pro zobrazování JPEG obrázků z kamery na WWW stránce. Rozhraní mezi kamerou a modemem je realizováno osmibitovým mikrokontrolerem ATMEL s aplikací napsanou v jazyce C. Pro přenos obrázků na server je v modemu implementován TCP/IP zásobník.

Dokument je rozdělen na dvě hlavní části. První část popisuje architekturu systému, použité komponenty (kamera, modem, mikrokontroler) a návrh zapojení. Druhá část dokumentu je zaměřena na programové vybavení systému. Nejprve je popsán operační systém modemu, vývojové prostředí OpenAT pro návrh aplikací pro modem a knihovna WIPsoft použitá pro implementaci TCP/IP zásobníku. Dále je popsána aplikace pro mikrokontroler, která řídí kameru a modem. Poslední část dokumentu je věnována návrhu serverové aplikace a webového rozhraní.

Klíčová slova: GPRS, modem, kamera, AT příkazy, OpenAT, WIPsoft, TCP/IP zásobník

Brno University of Technology

Faculty of Electrical Engineering and Communication

Department of Telecommunication

**Application of GPRS module for real-time
transmission of JPG files**

master's thesis

Specialisation of study: Communications and Informatics

Student: Jan Rusinský

Supervisor: Ing. OTTO DOSTÁL, CSc.

Abstract :

This document describes a design procedure of interface between VGA camera and GSM/GPRS modem for displaying JPEG pictures captured by camera on website. Interface is realized by eight-bit microcontroller ATMEL with application written in C language. For picture transfer to server a TCP/IP stack is implemented into modem.

The document is divided into two main parts. The first part describes architecture of the system, used components (camera, modem, microcontroller) and electronic circuit design. The second part of the document is focused on the software for the system. Firstly operating system of modem is described, OpenAT development environment for modem application design and WIPsoft library, which is used to implement TCP/IP stack. Next is described application for microcontroller which controls camera and modem. Last part of this document describes server application and web interface.

Key words: GPRS, modem, camera, AT commands, OpenAT, WIPsoft, TCP/IP stack

Bibliografická citace

RUSINSKÝ, J. Použití GPRS modulu pro přenos JPG souborů v reálném čase.

Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,

2010. 47 s. Vedoucí diplomové práce Ing. Otto Dostál, CSc.

P r o h l á š e n í

„Prohlašuji, že svůj semestrální projekt na téma Použití GPRS modulu pro přenos JPG souboru v reálném čase jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 : 26. května 2010

Podpis autora:

OBSAH

1. ÚVOD	7
2. HARDWARE	8
2.1 KAMERA	9
2.2 MODEM	10
2.3 MIKROKONTROLÉR	11
2.4 NÁVRH ZAPOJENÍ A PLOŠNÉHO SPOJE.....	11
2.4.1 Postup návrhu zapojení.....	11
2.4.2 Popis zapojení.....	12
3. SOFTWARE	14
3.1 NÁVRH KOMUNIKACE	14
3.2 PROGRAMOVÉ VYBAVENÍ MODEMU	15
3.2.1 Operační systém modemu.....	17
3.2.2 Vývojové prostředí OpenAT	18
3.2.3 Aplikace pro přenos dat	18
3.3 PROGRAMOVÉ VYBAVENÍ MIKROKONTROLÉRU	19
3.3.1 Obsluha periférií mikrokontroléru.....	20
3.3.2 Obsluha kamery	21
3.3.3 Obsluha modemu	23
3.3.4 Řízení nabíjení.....	28
3.3.5 Hlavní program.....	29
3.4 PROGRAMOVÉ VYBAVENÍ SERVERU	30
3.4.1 Server pro komunikaci s kamerou	31
3.4.2 Server pro komunikaci s klientem (webserver).....	31
3.5 WEBOVÉ ROZHRANÍ.....	31
3.5.1 Definice HTTP požadavků	32
3.5.2 Automatické znovunačtení obrázku (refresh).....	32
4. NÁVOD K POUŽITÍ	33
5. ZÁVĚR.....	34
6. LITERATURA	35

SEZNAM OBRÁZKŮ

Obr. 1: architektura systému – blokové schéma	8
Obr. 2: blokové schéma modulu kamery (převzato z [16])	9
Obr. 3: zapojení rozšiřujícího konektoru SV1	13
Obr. 3: návrh komunikace uživatele s kamerou.....	16
Obr. 5: blokové schéma jednotlivých modulů aplikace pro mikroprocesor	19
Obr. 6: parametry příkazu COMMAND.....	26
Obr. 7: stavový diagram hlavního programu	29
Obr. 8: aktivační schéma aplikace serveru.....	30
Obr. 9: náhled stránky webového rozhraní	32

1. ÚVOD

Úkolem této semestrální práce je navrhnout rozhraní mezi kamerou a GPRS modemem tak, aby se dal aktuální snímáný obraz zobrazit na webové stránce. V úvodu práce se budu krátce věnovat podobným zařízením na trhu a použité přenosové technologii, dále budou popsány jednotlivé komponenty, ze kterých bude zařízení zkonstruováno a jejich vzájemné propojení, včetně návrhu plošného spoje. Poslední část dokumentu popisuje návrh programového vybavení celého systému.

Podobná zařízení jsou dnes běžně dostupná na trhu v cenách řádově od 5000Kč výše. Některá jsou méně sofistikovaná, kdy se podobají funkcí spíše mobilnímu telefonu, snímané obrázky jsou zasílány formou MMS. Dražší modely mají naopak celou řadu funkcí jako noční vidění, detektory pohybu, snímače otřesu proti neoprávněné manipulaci, integrované web servery apod.

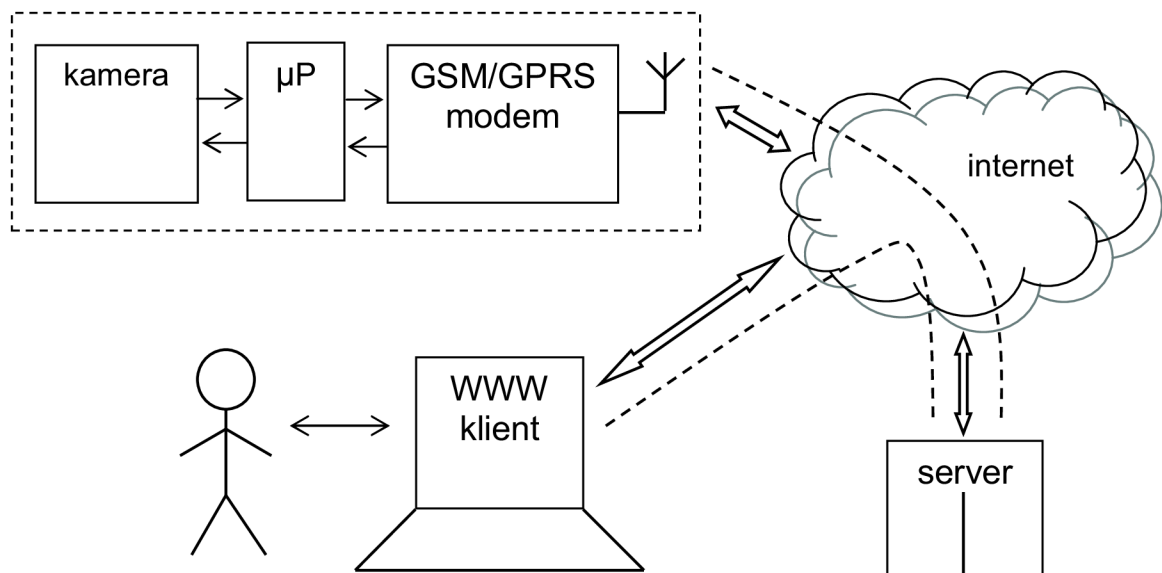
Výhodou použití GPRS technologie pro přenos dat je, že se (obvykle) neplatí za čas připojení, ale za množství přenesených dat. To umožňuje, aby zařízení bylo neustále takzvaně online, navíc s využitím internetových protokolů sady TCP/IP. GPRS je nádstavbou mobilní sítě GSM, která slouží primárně pro přenos hlasu. V dnešní době je však stále větší poptávka po datových přenosech jako videohovory, mobilní připojení k internetu a podobně. Proto bylo nutné vytvořit tuto nádstavbu, než nebudou rozšířeny sítě nové generace, které jsou pro datové přenosy navrženy. Rozšíření GSM sítě pro datové přenosy zajišťují zjednodušeně jen tři prvky. Jedná se o kontrolér paketového přenosu (PCU), který je součástí kontroléru bazových stanic (BSC) a odděluje paketové přenosy od přenosů s přepínáním okruhů. Druhým prvkem je uzel SGSN, který propojuje rádiovou přístupovou síť s hlavní pátevní sítí mobilního operátora a zajišťuje mobilitu zařízení, využívajících datové přenosy v rámci mobilní sítě. Uzel SGSN pak směřuje data k poslednímu uzlu GGSN, což je směrovač, který převádí data ze sítě operátora do veřejné sítě internetu a směřuje je na místo určení.

2. HARDWARE

Následující kapitola popisuje použité komponenty systému, které jsou zobrazeny v blokovém schématu komunikace (čárkovaně ohraničené části). Kameru, mikroprocesor i modem jsem dostal v rámci zadání, takže jsem nemusel provádět jejich výběr.

Kamera komunikuje s mikroprocesorem po sériové lince prostřednictvím definovaných příkazů. Modem je řízen pomocí AT příkazů, jak standardních, tak uživatelských, které jsou podporovány operačním systémem modemu.

Data z modemu jsou přenášena na server přes GPRS rozhraní mobilní sítě s využitím protokolu TCP/IP, který je implementován v modemu. Uživatel komunikuje s kamerou přes webové rozhraní. Komunikace je zprostředkována mezičlánkem v podobě aplikačního serveru. Důvod tohoto řešení je popsán dále. Ve finální verzi bude zařízení napájeno z akumulátoru, takže by mělo být použitelné prakticky kdekoli, podmínkou je pouze signál zvoleného operátora mobilní sítě.

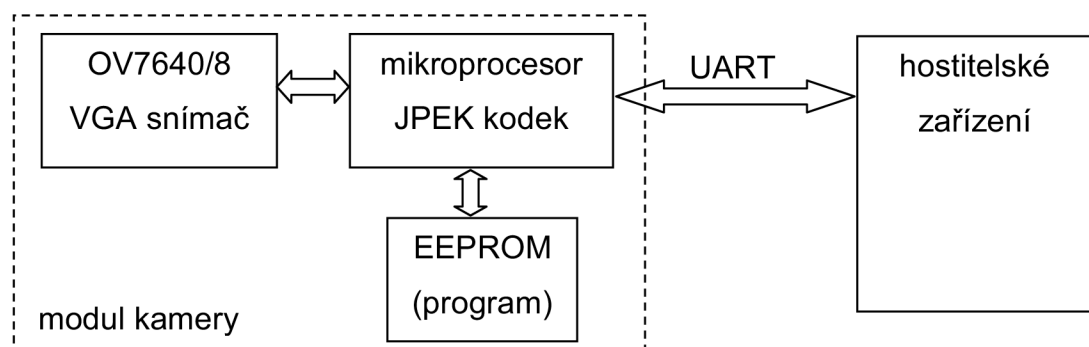


Obr. 1: architektura systému – blokové schéma

2.1 KAMERA

V této podkapitole je popsána kamera ITM-C-328, informace zde uvedené jsou z manuálu kamery [15] a z dokumentace ke zkušební desce kamery [16]. Oba dokumenty jsou volně dostupné na webových stránkách výrobce Intertec Components.

Samotný modul kamery C328 má rozměry přibližně 2 x 3cm. Na desce modulu je integrován snímací senzor, mikroprocesor a EEPROM (viz Obr.2). Snímací senzor je od výrobce OmniVision <<http://www.ovt.com/>> typ OV7640/8. Senzor dokáže snímat obraz v maximálním rozlišení VGA 640 x 480 obrazových bodů s osmibitovou barevnou hloubkou. Mikroprocesor modulu načítá obrazová data ze senzoru a pomocí JPEG kodeku provádí jejich kompresi. Kodek umožňuje komprimovat obraz v několika rozlišeních od 80 x 64 bodů po maximální rozlišení VGA (640 x 480). Modul nevyžaduje připojení externí paměti. Mikroprocesor dále obsluhuje sériovou linku UART, která je zároveň jediné rozhraní, přes které kamera komunikuje se svým okolím. Maximální rychlost sériové linky je 115,2Kbaud/s. Komunikace s kamerou je realizována sadou příkazů s délkou 6 bajtů. Jednotlivé bajty příkazu se vysílají ve standardním formátu: jeden startbit, 8 datových bitů (LSB jako první hned za startbitem) a jeden stopbit. Modul podporuje celkem 11 příkazů, použité příkazy budou vysvětleny dále v kapitole popisující návrh programového vybavení. Modul kamery je připojen čtyřmi vodiči. Dva vodiče jsou napájecí (pin 1 +3,3V a pin 4 GND) a dva vodiče jsou datové linky UARTu. Pin 2 je vysílací (Tx) a pin 3 je přijímací (Rx), viz schéma zapojení příloha A.



Obr. 2: blokové schéma modulu kamery (převzato z [16])

2.2 MODEM

Následující podkapitola je věnována seznámení s použitým modemem Fastrack M1306B. Informace zde uvedené pocházejí z dokumentace k modemu a jeho řídicího procesoru [9] a [10], z internetových stránek výrobce <<http://www.wavecom.com/>> a částečně z další dokumentace [11] až [14].

Modem patří ke starším produktům společnosti wavecom. Jedná se o dvoupásmový (900/1800Mhz) GSM/GPRS modem. Jeho srdcem je modul QUIK Q2406B, výrobcem označovaný jako Wireless CPU®. Řídicí modul obsahuje mikroprocesor, paměť, hodiny reálného času, řadič vstupně/výstupních portů, A/D převodník, řadič SIM karty, rozhraní pro mikrofon a reproduktor, UART a rozhraní pro radiovou část modemu. V modulu je nahrán firmware modemu. Jedná se o RTOS s podporou tzv. OpenAT® prostředí, které umožňuje spuštění uživatelské aplikace napsané v ANSI C přímo v modemu. Více o operačním systému modemu a OpenAT® vývojovém prostředí v kapitole zaměřené na návrh programového vybavení modemu.

Vnější rozhraní modemu tvoří čtyřpinový Micro Fit konektor Molex (pin 1 pro napájení, pin 2 pro GND a dva univerzální vstupy/výstupy), dále patnáctipinový Sub HD konektor, na který jsou vyvedeny signály standardní sériové linky RS232, audio rozhraní (mikrofon a reproduktor) a jako poslední dva jsou bootovací a resetovací vstup. SIM karta se vkládá do výsuvného držáku, který se vysouvá stisknutím miniaturního tlačítka. Připojení antény je zajištěno pomocí SMA konektoru. Posledním důležitým prvkem na těle modemu je červená LED, podle jejíž aktivity lze rozpoznat aktuální stav modemu. Pokud dioda svítí stále, modem je zapnut, ale není registrován v síti GSM. Pokud dioda bliká pomalu, modem je registrován v síti, ale neprobíhá komunikace (režim idle), pokud bliká rychle, modem komunikuje.

Komunikace s modemem je realizována pomocí standardních AT příkazů přes sériovou linku, případně pomocí uživatelských OpenAT® příkazů.

2.3 MIKROKONTROLÉR

Jako rozhraní mezi kamerou a modemem je použit mikrokontrolér ATMEL ATmega164p. Informace dále uvedené pocházejí z datasheetu mikrokontroleru [5]. Atmega164p je osmibitový RISC mikroprocesor s Harwardskou architekturou. V návrhu je použita verze v pouzdru PDIP40. Na čipu je integrováno 16kB paměti programu typu FLASH, 1kB paměti dat SRAM a 512B nonvolatilní EEPROM. Dále jsou v mikrokontroleru integrovány dva synchronní/asynchronní sériové vysílače/přijímače označované jako USART, 3 porty obsahující celkem 32 programovatelných vstupů/výstupů a čtyři čítače/časovače. Na čipu jsou ještě další periferní obvody, které však nejsou v aplikaci využity. Mikrokontrolér je zaměnitelný s Atmega324p/644p, které mají větší paměť a vyšší cenu.

2.4 NÁVRH ZAPOJENÍ A PLOŠNÉHO SPOJE

Na začátku této kapitoly bych měl zmínit, že navržené zařízení není prototypem finálního výrobku, jedná se o vývojovou verzi, která je určena pro další vývoj. Uvedená změna oproti zadání je výsledkem konzultací. Pokud by výsledkem návrhu mělo být finální zařízení pro komerční nasazení, respektive prototyp pro zkušební sérii, bylo by nutné návrh optimalizovat jak z hlediska velikosti (pouzdro MCU, separace nabíječky), tak z hlediska energetické náročnosti (jiný model MCU a GPRS modulu, implementace úsporných režimů).

2.4.1 Postup návrhu zapojení

Vývoj zařízení probíhal v několika fázích. Nejpreve jsem sestavil základní zapojení na nepájivém poli. Toto řešení bylo ale jen krátkodobé, protože působilo problémy se studenými spoji, což byla také pravděpodobně příčina zablokování mikrokontroléru Atmega164p. V další fázi projektu jsem součástky napájel na univerzální plošný spoj s MCU Atmega162, který má podobné parametry jako Atmega164p, ale nemá na čipu integrovány A/D převodníky (nutné pro realizaci nabíječky).

S tímto prototypem jsem prováděl vývoj softwaru prakticky celého systému kromě nabíječky akumulátoru.

V poslední fázi jsem doplnil schéma o obvody pro nabíjení a navrhnul jsem plošný spoj. Vzhledem k tomu, že velikost DPS není nějak kritická, je DPS navržena jako jednostranná s využitím SMD součástek, aby se výroba desky zrychlila a zjednodušila omezením počtu vrtaných děr. Schéma zapojení a výkresy DPS (motiv spoje a osazovací plány) jsou v příloze na konci dokumentu. Elektronická verze návrhu je přiložena na CD. Na hotové desce je provedena oprava, která je ve schematu naznačena (umožňuje měření napětí akumulátoru i mimo režim nabíjení).

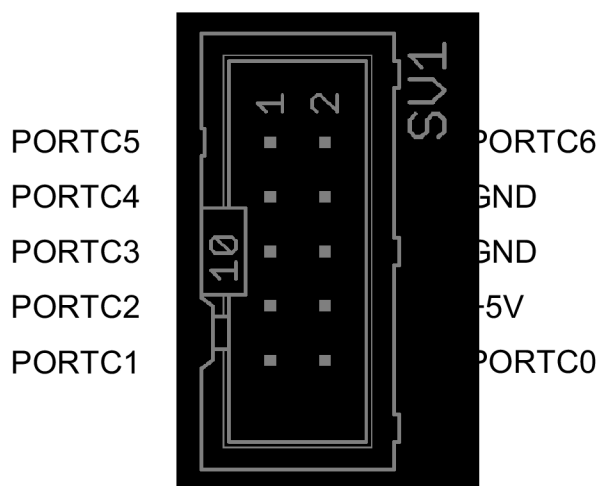
Pro kreslení schéma i návrh DPS byl použit návrhový systém EAGLE (verze: 4.16r2 pro Windows – Light Edition, jen pro neziskové a zkušební účely, CadSoft, <www.cadsoft.de>).

2.4.2 Popis zapojení

Navržené zařízení představuje prakticky dva nezávislé obvody, webkameru a nabíječku, které sdílejí mikrokontrolér a napájecí obvody. Přepnutí mezi těmito dvěma režimy je řešeno páčkovým přepínačem S1. V napájecím obvodu jsou použity integrované stabilizátory. LM1117 [8], který je použit pro napájení kamery a dobře známý LM7805 pro napájení MCU a pomocných obvodů. Kamera je připojena konektorem se zámkem, aby omylem nemohlo dojít k přepólování. Dalším obvodem v zapojení je IO MAX232 obsahující nábojovou pumpu, který přizpůsobuje napěťové úrovně sériové linky USART mikroprocesoru na napěťové úrovně rozhraní RS232. Rozhraní je zakončeno vidlicí CANON9 pro připojení modemu. Převod z CANON9 na patnáctipinový konektor modemu je realizován prostřednictvím propojovacího kabelu, který jsem dostal společně s modemem. K připojení sériového programátoru je zařízení vybaveno desetipinovým AVR ISP kompatibilním konektorem [18]. Programátor jsem sestavil podle návodu uvedeném v [1]. Napájení analogové části obvodu, tedy A/D převodníku a pomocného OZ je od napájecího napětí +5V odděleno tlumivkou L1. Referenční napětí je dodáváno stabilizátorem VR1 (IO TL431). Země pro analogovou část je také oddělena a se společnou zemí je spojena v jediném bodě pod mikrokontrolérem.

Obvod nabíječky akumulátoru je založen na referenčním návrhu [21] od výrobce ATMEL. Oproti originálnímu návrhu jsou pouze použity jiné součástky se stejnými nebo velmi podobnými parametry. Referenční návrh obsahuje podrobný popis funkce, teorii, schéma zapojení a zdrojové kódy obslužného programu včetně vývojových diagramů. Modifikací obslužného softwaru je možné měnit i typ použitého akumulátoru (SLA, NiCD, NiMH, Li-Ion), v případě tohoto projektu se bude asi jednat o zapouzdřený olověný akumulátor (SLA). Vzhledem k tomu, že je nabíječka vyčerpávajícím způsobem popsána v [21], uvedu jen ve stručnosti, že se jedná o obvod na bázi snižujícího DC-DC měniče řízeného výstupem PWM modulátoru MCU. Během nabíjení jsou A/D převodníkem mikrokontroléru monitorovány hodnoty nabíjecího napětí a proudu. Teplota akumulátoru je během nabíjení měřena pomocí NTC termistoru. Nabíjecí obvod je dimenzován na maximální proud 3A. Napájecí zdroj nabíječky by měl mít parametry 15V/1,5A DC.

Zařízení má být doplněno mechanismem pro natáčení směru kamery, k jeho připojení je vyhrazen konektor SV1, na nějž je vyvedeno sedm pinů mikrokontroléru a napájecí napětí (obrázek 3). Aplikační rozhraní je pro připojení mechanismu předpřipraveno, viz dále v kapitole software. Eventuelně je možné konektor využít pro připojení LED diod, pomocí kterých lze pak sledovat průběh programu v mikrokontroléru. Uvedený postup jsem při vývoji hojně využíval, protože bez emulátoru procesoru jde o jedinou možnost, jak trasovat program běžící v MCU.



Obr. 3: zapojení rozšiřujícího konektoru SV1

3. SOFTWARE

Následující kapitola popisuje návrh programového vybavení pro jednotlivé prvky systému. V první části je návrh komunikace mezi klientem a koncovým zařízením. Jsou zde zhodnoceny různé možnosti realizace a zvolená varianta je dále rozvedena a je popsána její implementace na jednotlivých komponentách systému.

Následuje základní popis funkce modemu a jeho OS, vývojového prostředí pro OpenAT a popis aplikace, která bude v modemu obsluhovat obousměrný přenos dat z mikroprocesoru na server.

Další část kapitoly popisuje návrh aplikace pro mikrokontrolér. Jedná se o stěžejní část celého projektu. Nejprve jsou popsány jednotlivé moduly programu, (obsluha kamery, obsluha modemu, nastavení periferií MCU) a poté návrh hlavního programu.

Předposlední část kapitoly je věnována návrhu programového vybavení pro aplikační server a poslední část je věnována webovému rozhraní.

3.1 NÁVRH KOMUNIKACE

Existuje několik možností řešení zadaného úkolu. Zvolený postup je kompromisem mezi rozumnou funkcionalitou a cenou datového tarifu.

Ideální by bylo použít datový tarif s veřejnou statickou IP adresou a implementovat WEB server přímo do mikrokontroléru. Tím by odpadl mezičlánek v podobě serveru, který zprostředkovává komunikaci mezi webem a kamerou, respektive GPRS modemem a je pravděpodobně nejslabším článkem systému.

Standardně lze navazovat datové spojení pouze ve směru mobilní zařízení → internet. Přidělované adresy pro mobilní zařízení jsou totiž v privátní síti operátora chráněné firewallem. Pro příchozí spojení ve směru internet → mobilní zařízení je potřeba aktivovat speciální tarif (aktivace přibližně 600,- Kč, měsíční poplatek za pevnou IP adresu je přibližně 250,- Kč). Podotýkám, že toto řešení by bylo nejdražší.

V tomto projektu byl však požadavek na co nejnižší cenu datového tarifu, navrhnul jsem proto takové řešení, které funguje se standardním datovým tarifem, kdy navázání spojení je možné pouze ve směru mobilní zařízení → internet a IP adresa je přidělována dynamicky.

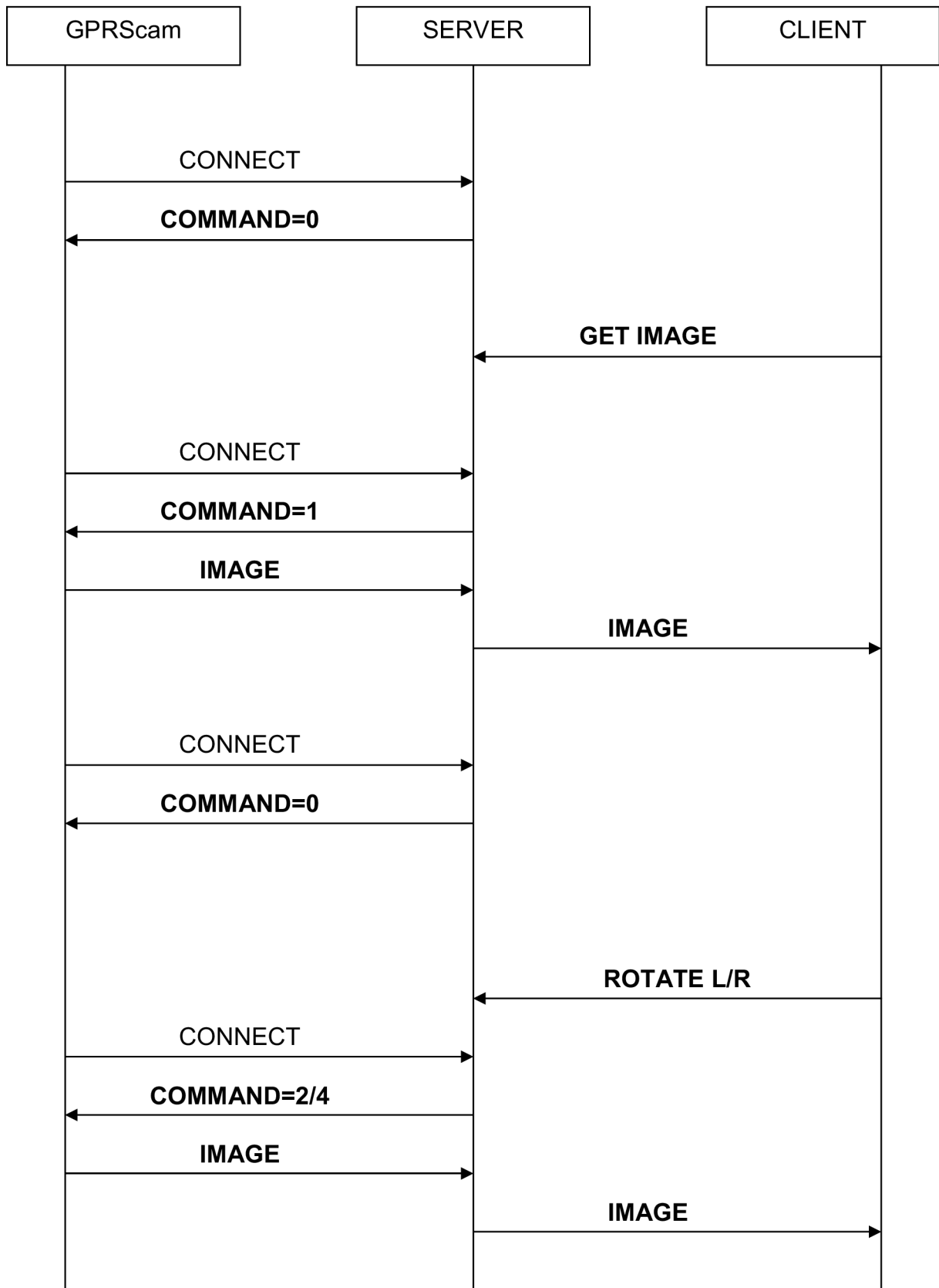
Návrh komunikace je na obrázku č. 4 na následující straně. Kamera periodicky navazuje spojení se serverem a ten ji předá požadavky přijaté od klienta. Požadavky mohou být následující: aktualizace obrázku, rotace kamery vlevo nebo vpravo a změna času pro periodické dotazování kamery na server. Požadavky klienta z webového rozhraní jsou v serveru převedeny do jediného bajtu a při navázání spojení kamery se serverem jsou ji předány. V diagamu komunikace na obrázku 3 jsou požadavky pro kameru označeny jako COMMAND.

Komunikace probíhá na aplikační vrstvě, transportní vrstva je víceméně transparentní, na straně modemu je realizována TCP/IP stackem v podobě OpenAT aplikace a na straně serveru je transportní vrstva implementována pomocí třídy Socket, která je součástí použitého .NET frameworku.

Zkoušel jsem i řešení, kdy byl server součástí webové aplikace (PHP), ale funkce byla velice nestabilní, docházelo k častému „zasekávání a padání“ serveru.

3.2 PROGRAMOVÉ VYBAVENÍ MODEMU

Jak už bylo zmíněno výše v kapitole 2.2, modem má vlastní operační systém s podporou uživatelských OpenAT aplikací. Této možnosti se snažím pokud možno co nejvíce využít a zjednodušit tak návrh aplikace pro mikrokontrolér ATMEL. Informace zde použité pocházejí z internetové encyklopedie [4] a z dokumentace modemu [9] – [17].



Obr. 4: návrh komunikace uživatele s kamerou

3.2.1 Operační systém modemu

Operační systém poskytuje aplikační rozhraní pro hardwarové funkce modemu jako hlasové, nebo datové přenosy. Jakmile je zahájeno nahrávání nové aplikace, vestavěný agent operačního systému se postará o její nainstalování a reinicializaci modemu.

Další výhodou je, že se jedná o RTOS, tedy systém s garantovanou dobou odezvy na události. Tyto vlastnosti umožňují vyvíjet pro modem i aplikace naročné na rychlost obsluhy bez toho, aby se uživatel musel starat o procesy řídicí funkcionality GSM rozhraní.

Operační systém umožňuje spouštět aplikace v jednom ze tří režimů:

1. Samostatná externí aplikace. Modem je řízen pouze pomocí AT příkazů přes sériovou linku z vnějšího zařízení (mikrokontroler, osobní počítač), kdy není spuštěna žádná vestavěná OpenAT aplikace.
2. Samostatná OpenAT aplikace. Modem je řízen pouze nahranou OpenAT aplikací bez komunikace pomocí AT příkazů s vnějším zařízením.
3. Tzv. Kooperativní režim, kdy je modem řízen jak nahranou OpenAT aplikací, tak i AT příkazy z vnějšího zařízení, které může modem řídit přímo pomocí standardních AT příkazů, nebo prostřednictvím vestavěné aplikace přes nestandardní AT příkazy. Tento režim budu využívat v tomto projektu. Umožňuje podstatné zjednodušení aplikace pro mikrokontroler. OpenAT aplikace bude v tomto případě implementací TCP/IP zásobníku (pozn. TCP/IP stack), který bude řízen pomocí nestandardních (uživatelských) AT příkazů zasílaných z aplikace v mikrokontroléru. Podrobnější vysvětlení jednotlivých režimů v [14]. Za zmínku stojí také funkce operačního systému DOTA, která umožňuje nahrávat do modemu nové aplikace přes GSM síť bez nutnosti fyzického přístupu k modemu.

3.2.2 Vývojové prostředí OpenAT

Vývojové prostředí OpenAT se používá k vývoji, ladění, a monitorování aplikací pro modem. Součástí tohto balíku programů je aplikace pro nastavení integrovaného vývojového prostředí, kde uživatel nastavuje parametry projektu (kompilátor, cesty souborů, typ OS v modemu, použité knihovny, atd.), dále program Eclipse (verze 3.2 <<http://www.eclipse.org/platform>>), který slouží jako editor pro psaní vlastní aplikace v jazyce C a správu použitých knihoven. Další součástí je nástroj pro monitorování a ladění aplikace v cílovém zařízení (Target Monitoring Tool), prostředí pro ladění aplikací na osobním počítači bez nutnosti nahrávání do cílového zařízení (Remote Task Environment) a pro pokročilé uživatele ještě příkazový řádek pro rychlejší vývoj aplikace, nastavování parametrů kompilace a používání skriptů bez nutnosti využívat grafické rozhraní. Podrobnější informace o vývojovém prostředí v [12] a [13].

3.2.3 Aplikace pro přenos dat

Po prostudování možností, jak použít OpenAT aplikaci pro přenos dat pomocí GPRS, respektive TCP/IP protokolu jsem vybral knihovnu WIPsoft, což je implementace TCP/IP zásobníku, která je součástí vývojového prostředí. Pokud je použita aplikace WIPsoft, žádná další OpenAT aplikace nemůže být spuštěna.

Před vlastním testováním této aplikace a samotného modemu vůbec jsem musel nahrát nový firmvare, protože stávající verze (6.41) byla zastaralá a nepodporovala WIPsoft. Firmware jsem musel nahrávat postupně, tedy upgradovat nejprve na verzi 6.51, 6.55 a nakonec jsem nahrál současnou verzi 6.57e.

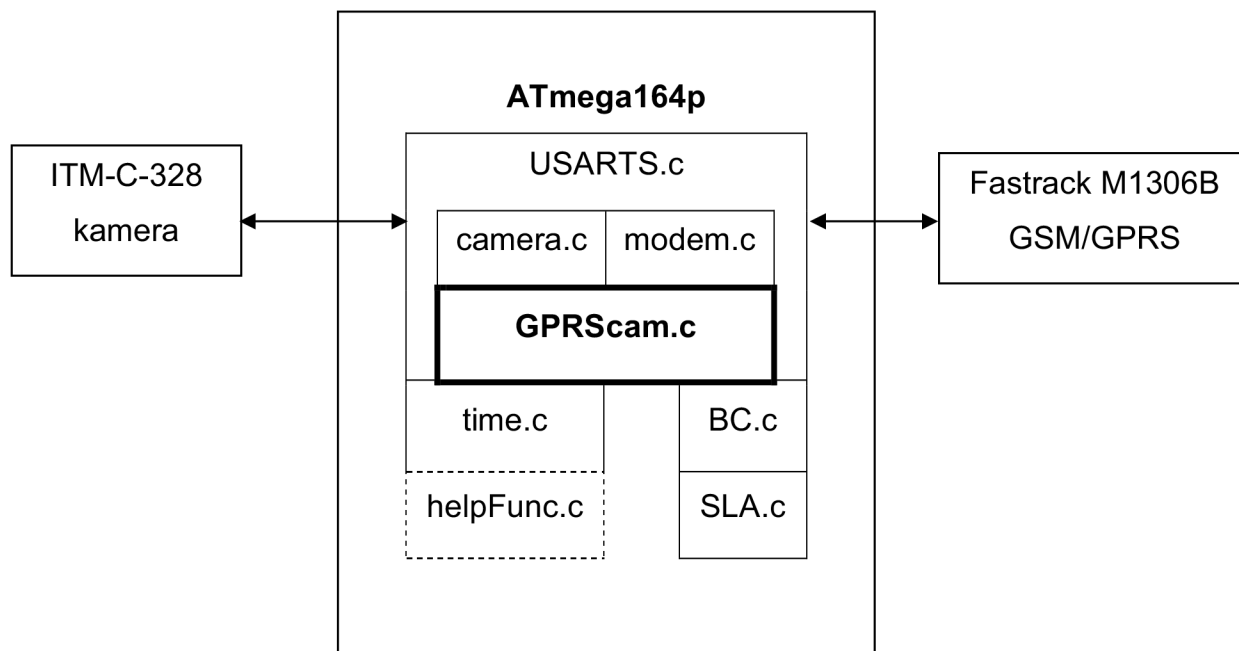
Pokud by nebyl dodržen tento postup, tedy sled jednotlivých verzí firmwaru a downloaderu modem by se mohl zablokovat a nebylo by možné jej zprovoznit standardním způsobem přes sériovou linku, ale musel by se zaslat autorizovanému prodejci/servisu. Tyto informace o upgradování firmwaru jsem získával převážně z internetového fóra výrobce, které slouží ke komunikaci mezi vývojáři a z dokumentace k firmwaru. Přesný postup aktualizace firmwaru je v dokumentaci k jednotlivým verzím.

3.3 PROGRAMOVÉ VYBAVENÍ MIKROKONTROLÉRU

Program pro mikroprocesor je napsán v jazyce C. Pro vývoj jsem použil open source projekt WinAVR <<http://sourceforge.net/projects/winavr/>>. Jedná se o sadu vývojových nástrojů pro RISC mikroprocesory řady AVR pro operační systém Windows. WinAVR obsahuje programy pro kompilaci (avr-gcc), nahrávání programu do mikroprocesoru (avrdude) a ladění aplikací (avr-gdb). Pro nahrávání programu používám program PonyProg - Serial Device Programmer (verze 2.07c Beta, <<http://ponyprog.sourceforge.net>>), který je kompatibilní s výše zmíněným sériovým programátorem.

Pro ladění aplikace v mikrokontroleru, respektive pro komunikaci s PC přes COM port používám volně šiřitelný program Terminal - com port development tool v1.9b stažený z internetových stránek <<http://sites.google.com/site/braypp/terminal>>.

Celý program je modulárně rozdělen na osm zdrojových souborů (viz Obr.5) a k nim náležejících hlavičkových souborů. Rozdělení umožňuje lepší laditelnost celého programu a také přenositelnost na jiný typ mikrokontroléru.



Obr. 5: blokové schéma jednotlivých modulů aplikace pro mikrokontrolér

3.3.1 Obsluha periférií mikrokontroléru

Pro obsluhu sériových linek a časovače jsem vytvořil samostatné zdrojové soubory tak, aby byla zajištěna modularita programu. Funkce, které jsou těmito moduly poskytovány, jsou využívány v ostatních zdrojových souborech.

3.3.1.1 Obsluha sériových linek

Funkce pro obsluhu obou sériových linek jsou umístěny ve zdrojovém souboru **USARTS.c** a jeho hlavičkovém souboru. Implementace je založena na referenčním návrhu [19] výrobce mikrokontroléru. Funkce pro obě sériové linky jsou téměř identické, budou tedy popsány pouze pro USART0. Sériová linka USART0 má navíc pro komunikaci s modemem implementováno hardwarové řízení toku signály RTS/CTS podle standardu RS232.

3.3.1.1.1 Inicializace sériové linky

Pro inicializaci USART0 je určena funkce **USART0_Init(baudrate)**. V této funkci je nastavena přenosová rychlost a parametry přenosu (8 datových bitů, jeden stopbit, bez paritního bitu).

3.3.1.1.2 Přijímání dat

Přijetí dat je ošetřeno asynchronně v obsluze přerušení reagující na přijetí nového bajtu (vektor přerušení USART0 Rx Complete), kde je každý přijatý bajt uložen do kruhového bufferu. Čtení dat z přijímacího bufferu je realizováno voláním funkce **USART0_Receive()**, která vrací nejstarší přijatý bajt z přijímacího bufferu.

3.3.1.1.3 Odesílání dat

K odesílání dat slouží funkce **USART0_Transmit(data)**, které se jako vstupní parametr předává jeden bajt určený k odeslání. Tato funkce zapisuje data k odeslání do odesílacího kruhového bufferu. Data z bufferu jsou pak odesílána obslužnou rutinou přerušení, která reaguje na vyprázdnění odesílacího registru sériové linky (vektor přerušení USART0 Data Register Empty).

3.3.1.1.4 Ověření přijatých dat

Ke zjištění, zda byla přijata data, je určena funkce **DataInReceiveBuffer0()**, která vyhodnocuje, jestli se v přijímacím kruhovém bufferu nacházejí nějaká data. V případě, že se v bufferu nacházejí nějaká data vrací logickou hodnotu true, v případě, kdy je přijímací buffer prázdný, vrací false.

3.3.1.2 *Obsluha časovače*

Obsluha osmibitového časovače 0 je implementována ve zdrojovém souboru **time.c** a jeho hlavičkovém souboru. Tento modul obsahuje funkci pro inicializaci časovače 0 nazvanou **Timer0_init()** a dále dvě funkce pro realizaci zpoždění. Jedna slouží pro realizaci časového zpoždění v řádu sekund **delay_s(sec)**, a druhá pro zpoždění v řádu milisekund **delay_ms(ms)**. Těmito dvěma funkcím se předává jako vstupní parametr požadovaná doba zpoždění v odpovídajících jednotkách. Funkce ke své funkci využívají přerušení při přetečení registru časovače (vektor přerušení Timer/Counter0 Overflow).

3.3.2 **Obsluha kamery**

Obsluha kamery se skládá z několika funkcí pro nastavení kamery a z jedné funkce pro přijímání obrazových dat. Jednotlivé příkazy pro kameru a jejich parametry jsou uvedeny v [16]. Funkce jsou umístěny ve zdrojovém souboru **camera.c** a jeho hlavičkovém souboru.

3.3.2.1 *Inicializace kamery*

Navázání spojení je provedeno následujícím handshakem. Nejprve je nutné opakovaně vysílat příkazu SYNC, dokud kamera nepotvrdí přijetí požadavku příkazem ACK následovaný příkazem SYNC. Od této chvíle je navázána komunikace a je možné zasílat ostatní příkazy. K synchronizaci je určena funkce **camera_connect()**.

3.3.2.2 Nastavení kamery

K nastavení kamery slouží funkce **camera_setup(resolution)**, ve které je provedeno nastavení rozlišení a velikosti paketu pro přenos obrázku. Dále je možno nastavit přenosovou rychlost pro komunikaci s kamerou, ale tato možnost není ve funkci implementována a je použita implicitní rychlost 14400bps. Jako vstupní parametr je funkci předáno požadované rozlišení podle následující tabulky. V tabulce jsou uvedeny také přibližné velikosti obrázku pro dané rozlišení, které je možné využít pro přibližnou kalkulaci objemu přenesených dat a volbu odpovídajícího datového tarifu. Velikosti jsou závislé na komplexnosti snímané scény. Nastavení se provádí příkazem INITIAL a jeho zpracování potvrdí kamera příkazem ACK

vstupní parametr	rozlišení	velikost
0	80x64	3kB
1	160x128	6kB
2	320x240	20kB
3	640x480	50kB

Tabulka č. 1: parametry rozlišení snímku kamery

Velikost paketu určuje po jak velkých blocích je obrázek posílán z kamery do mikroprocesoru. Minimální velikost paketu je 64B a maximální velikost je 512B. Velikost paketu je nastavena staticky přímo v kódu na 64B.

Inicializace a nastavení kamery je provedeno voláním funkce **camera_init()**.

3.3.2.3 Snímání obrázku

Ke snímání obrázku a jejich zasílání mikroprocesoru jsem napsal funkci **camera_snapshot()**. Nejprve je provedena synchronizace zavoláním funkce **camera_connect()**. Poté je vyslán příkaz SNAPSHOT, na který kamera reaguje uložení aktuálního snímaného obrázku do své paměti a odesláním potvrzení ACK.

Následně je kameře zaslán příkaz GET PICTURE. Kamera jeho přijetí potvrdí ACK a nazpět pošle příkaz DATA, který obsahuje velikost obrázku uloženou ve třech bajtech. Velikost je vyčíslena a je zahájen přenos obrázku. Jednotlivé pakety jsou jednoznačně rozlišitelné pomocí identifikačního kódu ID (2B), který obsahuje pořadové číslo paketu. Dále paket obsahuje délku obrazových dat v něm obsažených a kontrolní součet, který ale není ve funkci kontrolován a je zahozen. Přijetí paketu mikrokontrolerem se potvrzuje odesláním příkazu ACK, který zároveň obsahuje ID následujícího očekávaného paketu. Každý přijatý paket je zároveň odeslán na druhou sériovou linku mikrokontroléru do soketu sestaveného spojení se serverem.

3.3.2.4 Přijetí potvrzení od kamery

Jak už bylo zmíněno, úspěšné přijetí a zpracování příkazu kamera potvrzuje odesláním příkazu ACK. K jeho ověření slouží funkce **acknowledge()**. Při inicializaci spojení je použita funkce **acknowledge_sync()**.

3.3.3 Obsluha modemu

Obsluha modemu je založena na referenční aplikaci pro ovládnutí GSM modemů od výrobce ATMEL [20]. Tato aplikace obsahuje funkce pro odesílání AT příkazů a dekodování odpovědí. Jednotlivé AT příkazy jsou uloženy v paměti programu FLASH, kde jsou také okomentovány. Původní aplikaci jsem ale upravil, protože implementace některých funkcí nebyla optimální pro tento projekt. Jednalo se hlavně o dekodování odpovědi modemu přímo v obsluze přerušení a o obsluhu sériové linky, která nebyla řešena pomocí přerušení. Dále budu pro označení jednotlivých AT a OpenAT příkazů používat mnemotechnické názvy, které odpovídají názvu proměnné, ve které je ten který příkaz uložen v paměti (viz **modem.c**). Konkrétní význam jednotlivých příkazů a jejich parametrů je detailně popsán v dokumentaci [11] a [15].

3.3.3.1 Ovládání modemu pomocí AT příkazů

Postup ovládání modemu pomocí AT příkazů je následující (jednotlivé zde zmíněné funkce jsou podrobněji popsány dále).

1. Nejprve je potřeba vymazat a zinicilizovat přijímací buffer sériové linky a vypnout příjem, protože v této chvíli nejsou žádná data od modemu očekávána. Tento proces je implementován ve funkci **COM_rx_reset()**. Pro zpracování neočekávaných dat od modemu by bylo potřeba navrhnout a implementovat prohledávací algoritmus, který by byl schopen dekodovat libovolnou odpověď od modemu na základě databáze všech možných.
2. Dále je potřeba nastavit očekávanou odpověď modemu na daný příkaz funkcí **COM setSearchString(Response)**, které se jako vstupní parametr předá makro (viz dále) obsahující odpověď.
3. Následně se modemu zašle požadovaný AT příkaz voláním funkce **COM_puts(*str)**, které se jako vstupní parametr předá název příkazu, respektive ukazatel na daný příkaz, který je fyzicky uložen v paměti programu.
4. Nakonec se povolí příjem dat ze sériové linky voláním funkce **COM_rx_on()** a porovnáním návratové hodnoty funkce **check_modem_response()** se zjistí zda byl AT příkaz zpracován úspěšně nebo ne. V případě, že byl příkaz zpracován správně, je možné pokračovat dále například zasláním dalšího AT příkazu. Pokud nepřijde očekávaná odpověď, znamená to, že příkaz nebyl modemem zpracován podle očekávání a je potřeba tento chybový stav ošetřit.

3.3.3.2 Inicializace modemu

Modem se inicializuje voláním funkce **modeminit()**. V této funkci je nejprve vypnuto takzvané echo (modem pošle nazpět přijaté AT příkazy), aby se omezila zbytečná komunikace po sériové lince. Echo se vypíná AT příkazem **ATE0**. Na tento příkaz modem odpoví **ATE0 OK**.

Následně se ověří komunikace s modemem zasláním příkazu **AT**, na který modem odpoví **OK**. Nakonec je pomocí příkazu **CPIN** zadán PIN SIM karty a modem se registruje v síti. PIN je přiřazen k příkazu CPIN staticky přímo v kódu, pro jeho změnu je tedy potřeba program zkompileovat a nahrát do mikrokontroléru.

3.3.3.3 Spuštění a zastavení GPRS přenašeče

Spuštění GPRS připojení je realizováno voláním funkce **startbearer()**.

V této funkci je nejprve spuštěn TCP/IP stack příkazem **WIPCFG1**. Potom se inicializuje GPRS přenašeč příkazem **WIPBR1**, který je následován příkazem **WIPBR2** pro zadání názvu přístupového bodu APN. Název APN je přiřazen příkazu staticky stejně jako výše zmíněný PIN. Nakonec je modemu poslán příkaz **WIPBR4**, který spustí vlastní GPRS připojení (GPRS attach procedure). Všechny příkazy jsou potvrzovány odpovědí **OK**.

Funkce **stopbearer()** slouží k odpojení GPRS přenašeče a provádí víceméne stejné kroky v opačném pořadí. Nejprve se tedy odpojí GPRS připojení (GPRS detach procedure) příkazem **WIPBR5**, následně se vypne GPRS přenašeč AT příkazem **WIPBR0** a nakonec se vypne TCP/IP stack zasláním **WIPCFG0**. Úspěšné provedení všech příkazů je opět potvrzováno odpovědí **OK**.

3.3.3.4 Komunikace se serverem

Komunikace se serverem je implementována ve funkci **server_connection()** a probíhá ve třech krocích:

1. Navázání TCP/IP spojení se serverem
2. Výměna dat (přijetí požadavku od klienta a odeslání aktuálního obrázku)
3. Ukončení TCP/IP spojení se serverem

Pro lepší názornost komunikace mikrokontroléru s modemem pomocí AT příkazů je na konci dokumentu přiložen (příloha E) záznam komunikace mikroprocesoru s modem od zapnutí zařízení až po odpojení od serveru i s přibližnými časy vykonávání jednotlivých příkazů.

Červeně jsou označeny AT příkazy zasílané mikrokontrolérem, modře odpovědi modemu. Výměna dat se serverem je naznačena pouze symbolicky. Znaky \r a \n odpovídají ASCII znakům CR a LF.

Navázání spojení se serverem je provedeno AT příkazem **WIPCREATE2**, který vytvoří TCP klienta. Příkaz má staticky nastavenou IP adresu a port (50023) serveru, při změně IP adresy serveru je tedy potřeba znovu zkompilovat a nahrát program.

Po sestavení spojení se sériová linka přepne do datového módu příkazem **WIPDATA**, na který má přijít odpověď **CONNECT_**. V této chvíli je TCP komunikace transparentní, takže data, která pošle server do soketu, přijdou přímo na sériovou linku a naopak data poslaná na sériovou linku může server přijímat z TCP soketu. Mikroprocesor přijme požadavky klienta ve formě příkazu **COMMAND**, jak už bylo zmíněno. Příkaz je tvořen jedním bajtem, jehož struktura a význam jednotlivých bitů je naznačen na obrázku č. 6. V závislosti na obsahu **COMMANDu** se provedou odpovídající akce (odeslání obrázku, rotace kamery, nastavení periody připojení k serveru) a datový mód se ukončí příkazem **WIPDATASTOP**, což je řetězec **+++**, před kterým musí být ochranná doba jedné sekundy. Odeslání obrázku je realizováno voláním funkce **camera_snapshot()**, která přímo odešle do sestaveného soketu obrázek a ten je ukončen dvěma bajty 0x7F (127 dekadicky, ASCII znak DEL), podle kterých pozná serverová aplikace konec obrazových dat. Rotace kamery je realizována voláním funkce **rotate_left()**, **rotate_right()** které pouze v daném směru symbolicky rozsvěčují LED diody připojené na PORTC. Do těchto funkcí je možné implementovat ovládání pohybu kamery pomocí servomotorku nebo krokového motorku. Spojení se serverem se ukončí příkazem **WIPCLOSE**. Všechny AT příkazy kromě **WIPDATA** jsou potvrzovány odpovědí **OK**.

b7	b6	b5	b4	b3	b2	b1	b0
rezerva	refresh 10s	refresh 30s	refresh 60s	rezerva	Rotate right	Rotate left	Get picture

Obr. 6: parametry příkazu COMMAND

3.3.3.5 Pomocné funkce pro komunikaci s modemem

Pro zasílání AT příkazů a dat do modemu a dekodování jeho odpovědi je k dispozici celá řada funkcí, které už byly krátce zmíněny a jsou podrobněji popsány v následujících podkapitolách.

3.3.3.5.1 Funkce pro dekodování odpovědi od modemu

Před odesláním AT příkazu se nastavuje očekávaná odpověď pomocí funkce **COM setSearchString(Response)**. Vstupní parametr Response udává pozici v poli ukazatelů na odpovědi, pro zjednodušení jsou jednotlivým pozicím přiřazeny v souboru **modem.h** makra s mnemotechnickými názvy odpovědi. V této funkci se nejprve zakáže příjem na sériové lince, respektive se zakáže přerušení USART0 Rx Complete. Potom se do ukazatele **searchFor** uloží adresa, na které je textový řetězec očekávané odpovědi a vynuluje se proměnná **rx_i**, která je pak ve funkci pro dekodování odpovědi použita pro procházení přes všechny znaky řetězce v **searchFor**.

Samotná funkce pro dekodování odpovědi modemu je nazvána **check_modem_response()**. V této funkci je použit časovač pro nastavení doby, po kterou se čeká na nastavenou odpověď. Pokud odpověď nepřijde před vypršením nastaveného timeoutu, je vyodnocena jako špatná odpověď. Pro nastavený kmitočet čítače/časovače0 odpovídá jedna sekunda 28mi přetečením (vektor přerušení Timer/Counter0 Overflow) registru čítače. Nastavená hodnota je 280, tedy 10 sekund, což je poměrně dlouhá doba pro přijetí odpovědi a je možné ji zkrátit.

Dále funkce čeká, než přijdou data do přijímacího bufferu sériové linky pomocí funkce **DataInReceiveBuffer0()**. Přijatá data pak porovnává znak po znaku s řetězcem nastaveným v **searchFor**. Pokud přijde očekávaná odpověď v pořádku, vrací funkce 1, v opačném případě vrací 0.

3.3.3.5.2 Funkce pro komunikaci sériovou linkou

Zdrojový soubor **modem.c** obsahuje také několik pomocných funkcí pro řízení sériové linky. Funkce **COM_rx_on()** a **COM_rx_off()** slouží pro zapnutí a vypnutí sériové linky povolením nebo zakázaním přerušení.

Funkce **COM_put_integer(i)** převede vstupní parametr ve formě celého čísla na textový řetězec a odešle jej na sériovou linku. V současné verzi programu není tato funkce použita, ale je velice výhodné ji používat ve fázi ladění programu.

Funkce **COM_puts(*str)** slouží k odeslání řetězce ukončeného znakem null na sériovou linku. Funkce **COM_putsf(*fstr)** je ke stejnému účelu ale pro řetězce uložené v paměti programu FLASH. Tato funkce je používána pro odesílání AT příkazů.

3.3.3.5.3 Funkce pro odesílání SMS

Pokud se zařízení dostane do chybového stavu (např se nepovede inicializovat modem) je možné odeslat chybovou zprávu pomocí SMS na předem stanovené číslo, které je nutno nadefinovat v příkazu SMS v souboru **modem.c**. Vlastní odeslání se provede voláním funkce **send_SMS(char *SMSmsg)**, které se předá jako vstupní parametr textový řetězec obsahující vlastní chybovou zprávu.

3.3.4 Řízení nabíjení

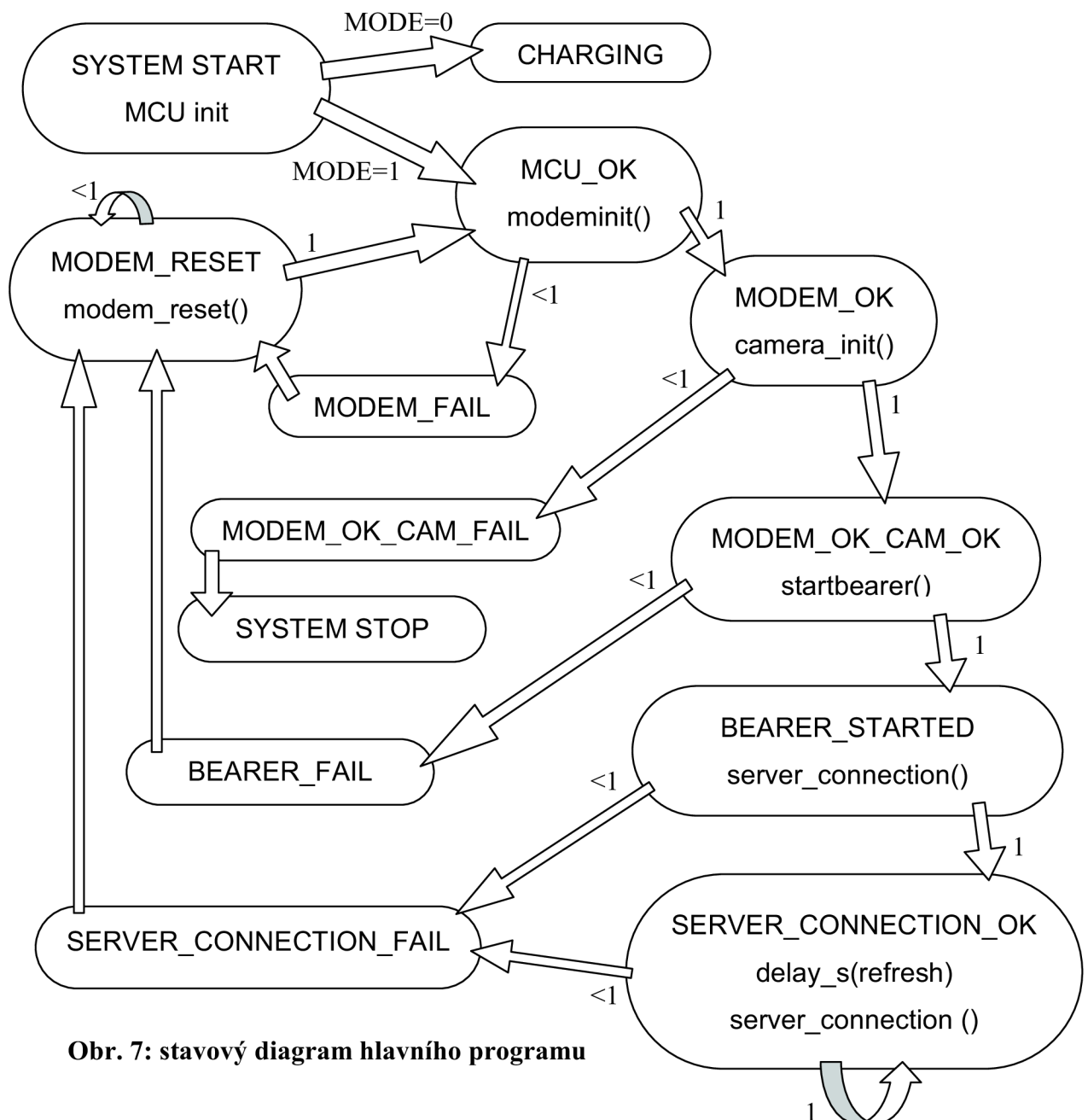
Jak už bylo zmíněno v kapitole o hardwaru, nabíječka akumulátoru je založena na referenčním návrhu [21] výrobce ATMEL. V dokumentaci jsou vývojové diagramy a popis všech funkcí, uvedu zde tedy jen popis použitých modulů a jejich stručný obsah.

BC_def.h	definice typu akumulátoru, napětí a počtu článku, kapacity a koeficientu pro měření A/D převodníkem.
bc.h	obsahuje makra pro definici vstupů a výstupů MCU
BC.c	obsahuje hlavní program nabíječky a funkce pro obslu PWM a ADC
SLA.h	definice parametrů nabíjecího cyklu (max. proud, teplota)
SLA.c	nabíjecí algoritmus pro olověné akumulátory

Program pro nabíjení není odzkoušen a je pouze připraven na odladění. Zařízení jsem provozoval ze síťového napájecího zdroje a s akumulátorem nabylo vůbec testováno. Před testováním nabíjení je nutné nastavit parametry použitého akumulátoru podle dokumentace [21].

3.3.5 Hlavní program

Hlavní program se nachází ve zdrojovém souboru **GPRScam.c**. Program je navržen jako stavový automat. Stavový diagram je na následujícím obrázku č. 7. Stav systému je uložen v proměnné **system_state**. V každém stavu je provedena určitá operace a přechod do dalšího stavu je závislý na výsledku této operace uloženém v proměnné **action_result**. V případě úspěšného provedení operace je výsledek 1, v případě chyby je výsledek 0 nebo záporné číslo signalizující, kde nastala chyba.



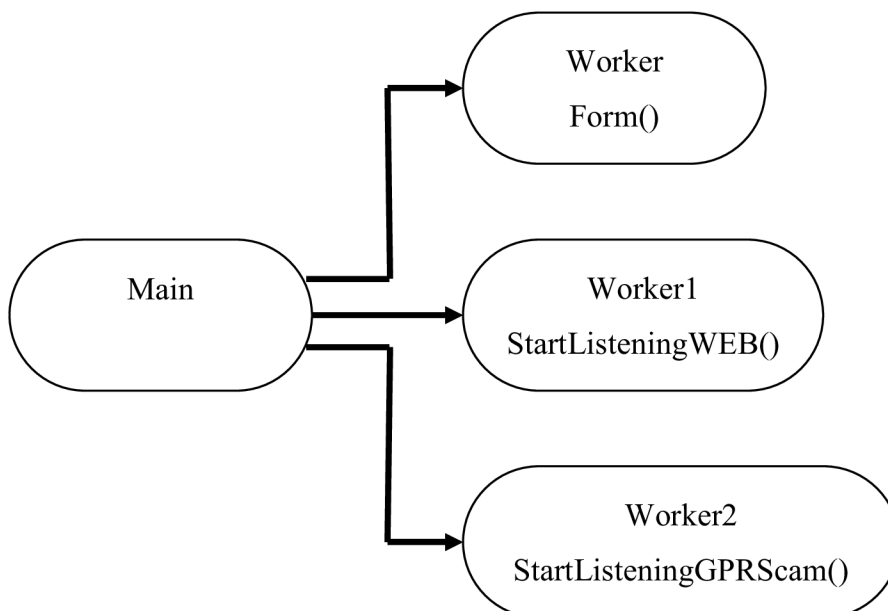
Obr. 7: stavový diagram hlavního programu

Po zapnutí napájení se systém rozhodne na základě parametru MODE zda bude pracovat v režimu nabíječky akumulátoru, nebo jako webkamera. Proměnná MODE je závislá na poloze přepínače S1 (viz příloha A, PORTC.7).

3.4 PROGRAMOVÉ VYBAVENÍ SERVERU

Aplikace pro server je napsána v .NET frameworku v jazyce C# a k jejímu vývoji jsem použil nástroj Microsoft Visual C# 2008 Express Edition. Aplikace je určena pro operační systém Windows.

Jádrem programu jsou dva synchronní TCP servery. Jeden obsluhuje komunikaci s kamerou, druhý pracuje jako jednoduchý WWW server. Servery jsou implementovány třídou **SynchronousSocketListener**. Dále aplikace obsahuje formulář pro zobrazení aktuálního obrázku z kamery. Na obrázku číslo 8 je zobrazeno aktivační schéma jednotlivých procesů (vláken) programu. Procesy jsou realizovány pomocí třídy **Worker**.



Obr. 8: aktivační schéma aplikace serveru

3.4.1 Server pro komunikaci s kamerou

Server naslouchá na portu 50023 a po navázání spojení s kamerou odešle příkaz COMMAND. Potom čeká na přijetí obrázku, který je zakončen dvěma znaky DEL (dekadická jednota 127). Tyto znaky pak z pole přijatých bajtů odstraní a pole přemění na obrázek, který je připraven k odeslání klientovi. Nakonec spojení ukončí a začne naslouchat znovu.

3.4.2 Server pro komunikaci s klientem (webserver)

Webserver naslouchá na portu 80 a dekoduje příchozí požadavky. Jestliže je požadavek v odpovídající formátu (definice požadavků je v následující kapitole) je obsloužen. To znamená, že je zakódován do příkazu COMMAND a klientovi je zaslána odpověď **HTTP/1.0 200 OK** s příslušným obsahem (potvrzení požadavku nebo obrázek). Pokud požadavek neodpovídá žádnému z definovaných, je klientovi odeslána odpověď **HTTP/1.0 404 Not Found**. Následně je spojení ukončeno a server začne opět naslouchat na portu 80.

3.5 WEBOVÉ ROZHRAŇÍ

Webové rozhraní slouží jako HMI pro komunikaci s kamerou. Webová stránka je vytvořena kombinací HTML a JavaScript kódu (viz příloha F). Stránka obsahuje rám (iframe) pro zobrazení obrázku z kamery, tři tlačítka pro nastavení obnovovací periody pro navazování spojení kamery se serverem, dvě tlačítka pro pohyb kamery a rám (iframe), ve kterém se zobrazuje poslední přijatá zpráva od serveru, obvykle potvrzení přijetí požadavku (obrázek č. 9). Součástí stránky je ještě soubor dummy.html, který tvoří pouze pozadí pro druhý rám s odpověďmi serveru.

Stránky s webovým rozhraním musí být dvě, jedna pro počítače ve stejné lokální síti jako server, druhá pro počítače, které se k serveru připojují z vnější sítě. Rozdíl je pouze ve zdrojové/cílové IP adrese požadavků. Možné řešení je v příloze na CD.



Obr. 9: náhled stránky webového rozhraní

3.5.1 Definice HTTP požadavků

1. Při pokusu o znovunačtení obrázku je serveru odeslán požadavek **GET /pic HTTP/1.1**
2. Při stisku některého z tlačítek pro změnu času periodického připojování kamery k serveru je zavolána funkce **set_refresh(time)** a zároveň je odeslán požadavek ve tvaru **GET /?refresh=1 HTTP/1.1**, kde hodnota za refresh nastavuje dobu odezvy kamery (refresh).
3. Stisknutí tlačítka pro rotaci kamery odešle požadavek **GET /?left= HTTP/1.1** nebo **GET /?right= HTTP/1.1**

3.5.2 Automatické znovunačtení obrázku (refresh)

Znovunačítání iframu s obrázkem z kamery je ošetřeno pomocí tří funkcí napsaných v JavaScript kódu. Jedná se o funkci **frame_reload()**, která slouží k samotnému znovunačtení rámu a tím i obrázku, dále funkci **set_refresh(time)**, které využívá funkci **frame_reload()** a jako vstupní parametr funkce je perioda znovunačítání rámce v milisekundách. Poslední funkce **stop_refresh()** je pomocná funkce ke zrušení aktuální nastavené obnovovací periody před nastavením nové a je volána ve funkci **set_refresh(time)**.

4. NÁVOD K POUŽITÍ

Před zprovozněním systému je nutné provést následující kroky. Nejprve je nutné změnit parametry AT příkazů, které jsou určeny pro navazování spojení. Příkazy jsou ve zdrojovém souboru **modem.c**.

1. PIN v příkazu **CPIN**
2. APN v příkazu **WIPBR2**
3. IP adresa serveru v příkazu **WIPCREATE2**

Ukázka pro PIN 5678

```
const unsigned char CPIN[] PROGMEM = "AT+CPIN=5678\r";
```

Ukázka pro APN internet

```
const unsigned char WIPBR2[] PROGMEM = "AT+WIPBR=2,6,11,\"internet\"\r";
```

Ukázka pro IP adresu serveru 217.194.152.248

```
const unsigned char WIPCREATE2[] PROGMEM =  
"AT+WIPCREATE=2,1,\" 217.194.152.248\",50023\r";
```

Dále je nutné odladit program pro nabíjení podle použitého akumulátoru. Postup je uveden v [21]. Nabíjecí obvod je dimenzován na maximální proud 3A. To je nutné brát v úvahu při definování maximálního nabíjecího proudu hlavně u akumulátoru s vyšší kapacitou.

Změny je nutné provést také ve zdrojovém kódu webového rozhraní. Zde se jedná pouze o změnu IP adresy požadavků. Adresy jsou v příloze F zvýrazněny červeně. Pro stránku lokálních počítačů se nastaví privátní adresa serveru, pro stránku „hostů“ se nastaví veřejná adresa počítače serveru, nebo hraničního prvku sítě (routeru), na kterém je pak potřeba nastavit port forwarding pro porty 80 a 50023 směřovaný na lokální IP adresu serveru.

5. ZÁVĚR

Před samotným návrhem řešení jsem se musel seznámit s funkcí jednotlivých komponent zařízení. V případě kamery se nejednalo o velký problém, protože její funkce je jednoznačně popsána pomocí jedenácti příkazů. S mikrokontroléry ATMEL jsem už měl zkušenosti, takže se jednalo spíše o rozšíření znalostí. Jiná situace byla s modemem, respektive jeho operačním systémem a vývojovým prostředím OpenAT. V tomto případě jsem se musel seznámit s poměrně rozsáhlou dokumentací a nakonec jsem zvolil řešení v podobě plug-in modulu (knihovny WIPsoft) dodávaného výrobcem, který se ideálně hodí pro použití v tomto projektu. Návrh a vývoj vlastní OpenAT aplikace by byl časově velmi náročný, ale jednalo by se o velice elegantní řešení bez externího mikrokontroléru. Jak už je v tomto dokumentu zmíněno, navržené zařízení je spíše vývojový kit.

Funkce nabíječky není vyzkoušena, zařízení jsem provozoval pouze ze síťového napájecího zdroje. Obvod je realizován hardwarově a aplikační část pro obsluhu nabíjení je nutné upravit až podle použitého akumulátoru.

Primární funkci webkamery jsem otestoval s pozitivním výsledkem, otázkou je stabilita serverové aplikace při dlouhodobém provozu, která by se dala realizovat lépe, ale s vývojem takovýchto aplikací (myšleno dlouhodobě běžících na PC) nemám dostatek zkušeností.

6. LITERATURA

- [1] Váňa V.: Mikrokontroléry ATMEL AVR Popis procesoru a instrukčního souboru. Technická literatura BEN, Praha 2003.
- [2] Váňa V.: Mikrokontroléry ATMEL AVR Programování v jazyce C. Technická literatura BEN, Praha 2003.
- [3] Mann B.: C pro mikrokontroléry. Technická literatura BEN, Praha 2003.
- [4] Wikipedia The Free Encyclopedia, <wikipedia.org>.
- [5] Atmega162 datasheet. ATMEL Corporation, <www.atmel.com>.
- [6] Application note AVR306: Using the AVR® UART in C, ATMEL Corporation, <www.atmel.com>.
- [7] Zeman V., Rajmic P., Sysel P.: Věcné a typografické pokyny a zásady pro psaní studentských prací. Vysoké učení technické v Brně, Brno 2009
- [8] LM1117/LM1117I 800mA Low-Dropout Linear Regulator datasheet. National Semiconductor, <www.national.com>.
- [9] Fastrack M1306B User Guide. WAVECOM, <www.wavecom.com>.
- [10] Q2406_and_Q2426_Product_Specification, <www.wavecom.com>.
- [11] AT Commands Interface Guide for Open AT® Firmware v6.67e. WAVECOM, <www.wavecom.com>.
- [12] Tutorial for Open AT® IDE 1.04. WAVECOM, <www.wavecom.com>.
- [13] Tools Manual for Open AT® IDE 1.04. WAVECOM, <www.wavecom.com>.
- [14] Basic Development Guide for Open AT® OS v3.14. WAVECOM, <www.wavecom.com>.
- [15] WIP AT COMMANDS USER GUIDE (WIPSOFT V3.11). WAVECOM, <www.wavecom.com>.
- [16] ITM-C-328 Manual. Intertec Components GmbH, <www.intertec-components.de>.
- [17] ITM-EVA-328 Manual. Intertec Components GmbH, <www.intertec-components.de>.
- [18] AVR042: AVR Hardware Design Considerations, <www.atmel.com>.
- [19] AVR306: Using the AVR® UART in C, <www.atmel.com>.

- [20] AVR323: Interfacing GSM modems, <www.atmel.com>.
- [21] AVR450: Battery Charger for SLA, NiCd, NiMH and Li-Ion Batteries, <www.atmel.com>.
- [22] ATmega164PA/324PA/644PA/1284P datasheet. ATMEL Corporation, <www.atmel.com>.

SEZNAM ZKRATEK

Zkratka/Symbol	Popis
A/D	analog/digital
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
AT	Attention
B	bajt, 8bitů
BSC	Base Station Controller
CPU	Central processing unit
CTS	Clear To Send
DOTA	Download Over-the-Air
DPS	Deska plošného spoje
EEPROM	Electrically Erasable Programmable Read-Only Memory
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GND	Ground
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications Groupe Spécial Mobile
HD	High Density
HMI	Human-Machine Interface
ISP	In System Programmable
JPEG	Joint Photographic Experts Group
kB	kilobajt, 1024 bajtů
kodek	koder-dekoder
LED	Light-emitting diode
LSB	least significant bit
MMS	Multimedia Messaging Service
modem	modulator-demodulator

Zkratka/Symbol	Popis
NAT	Network address translation
NTC	Negative Temperature Coefficient
OS	Operating system
OZ	Operační zesilovač
PC	personal computer
PCU	packet control uni
PDIP40	Plastic Dual Inline Package 40pin
PDU	Protocol data unit
PIN	Personal Identification Number
RISC	Reduced Instruction Set Computer
RTOS	Real-time operating system
RTS	Request To Send
SGSN	Serving GPRS Support Node
SIM	subscriber identity module
SLA	Sealed Lead Acid
SMA	SubMiniature version A (konektor)
TCP/IP	Transmission Control Protocol/Internet Protocol
UART	Universal asynchronous receiver/transmitter
uP	mikroprocesor
USART	Universal Synchronous Asynchronous Receiver Transmitter
VGA	Video Graphics Array
WIP	Wavecom Internet Protocol

SEZNAM PŘÍLOH

- Příloha A schéma zapojení
- Příloha B motiv plošného spoje
- Příloha C osazovací výkres, strana plošného spoje
- Příloha D osazovací výkres, strana součástek
- Příloha E záznam komunikace uP s modemem
- Příloha F zdrojový kód webového rozhraní

Přílohy na CD

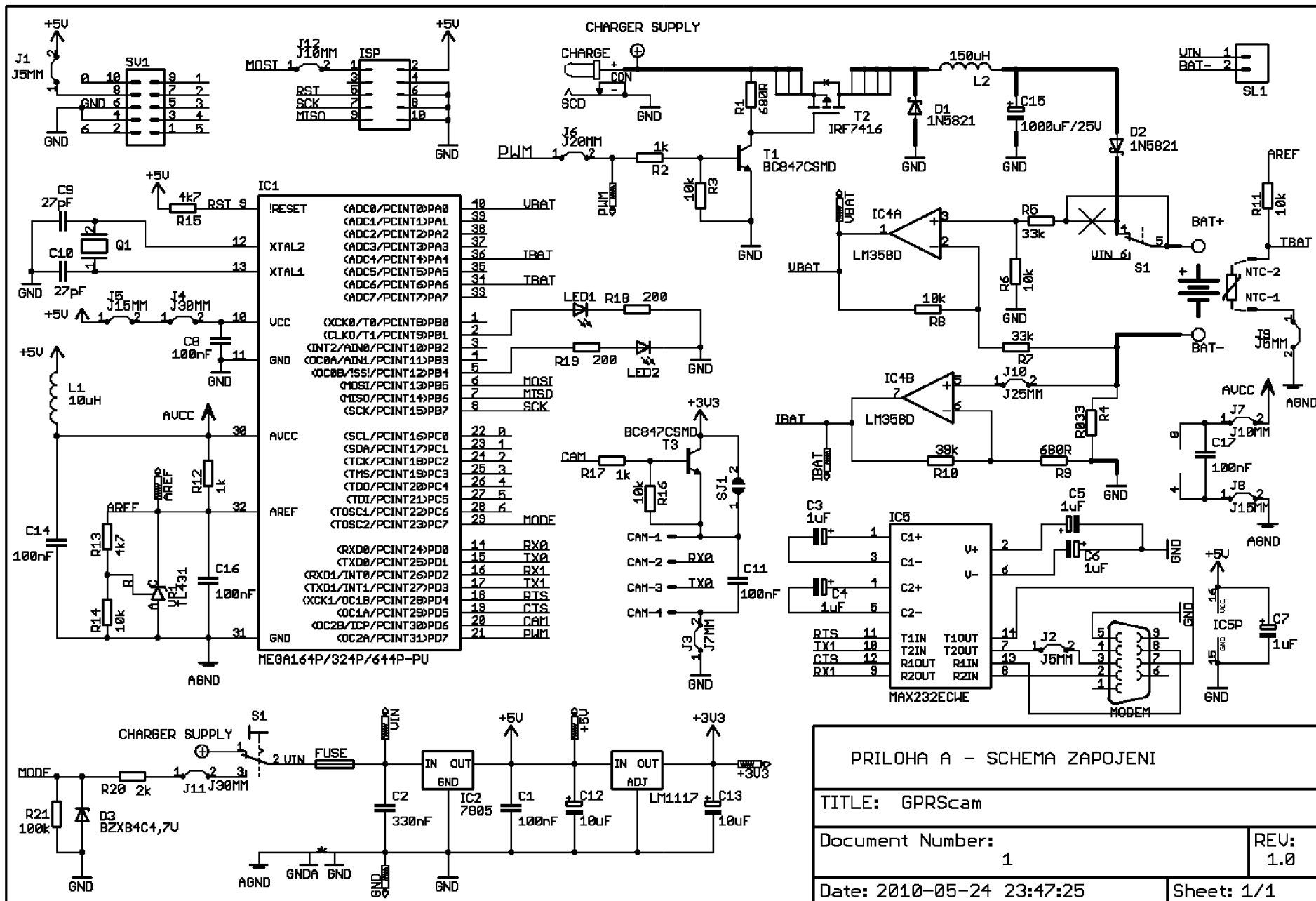
elektronická verze dokumentace

aplikace pro mikrokontrolér

aplikace pro server

webové rozhraní

dokumentace DPS (soubory pro Eagle, seznam materiálu)



PRILOHA A - SCHEMA ZAPOJENI

TITLE: GPRScam

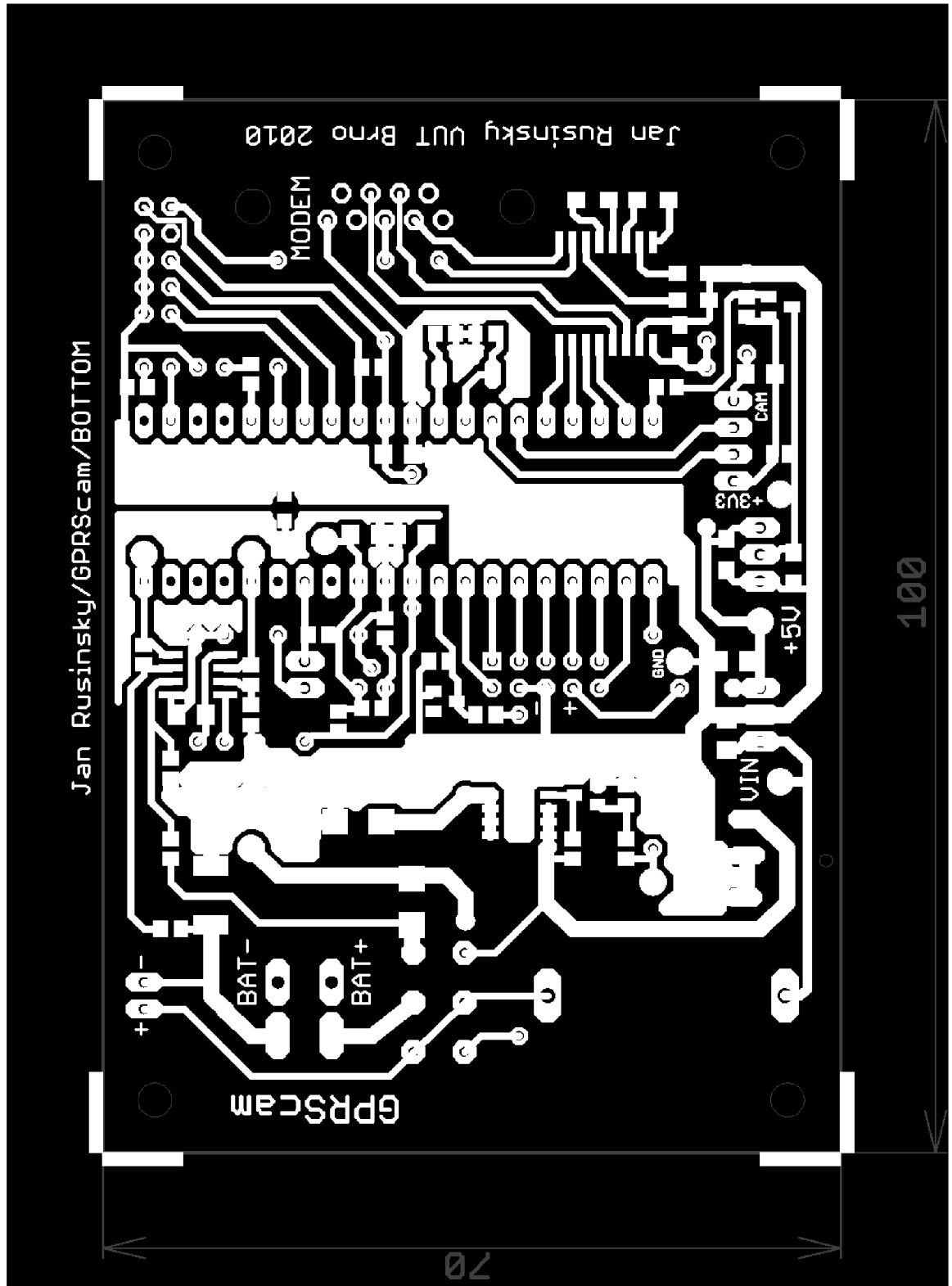
Document Number: 1

REV: 1.0

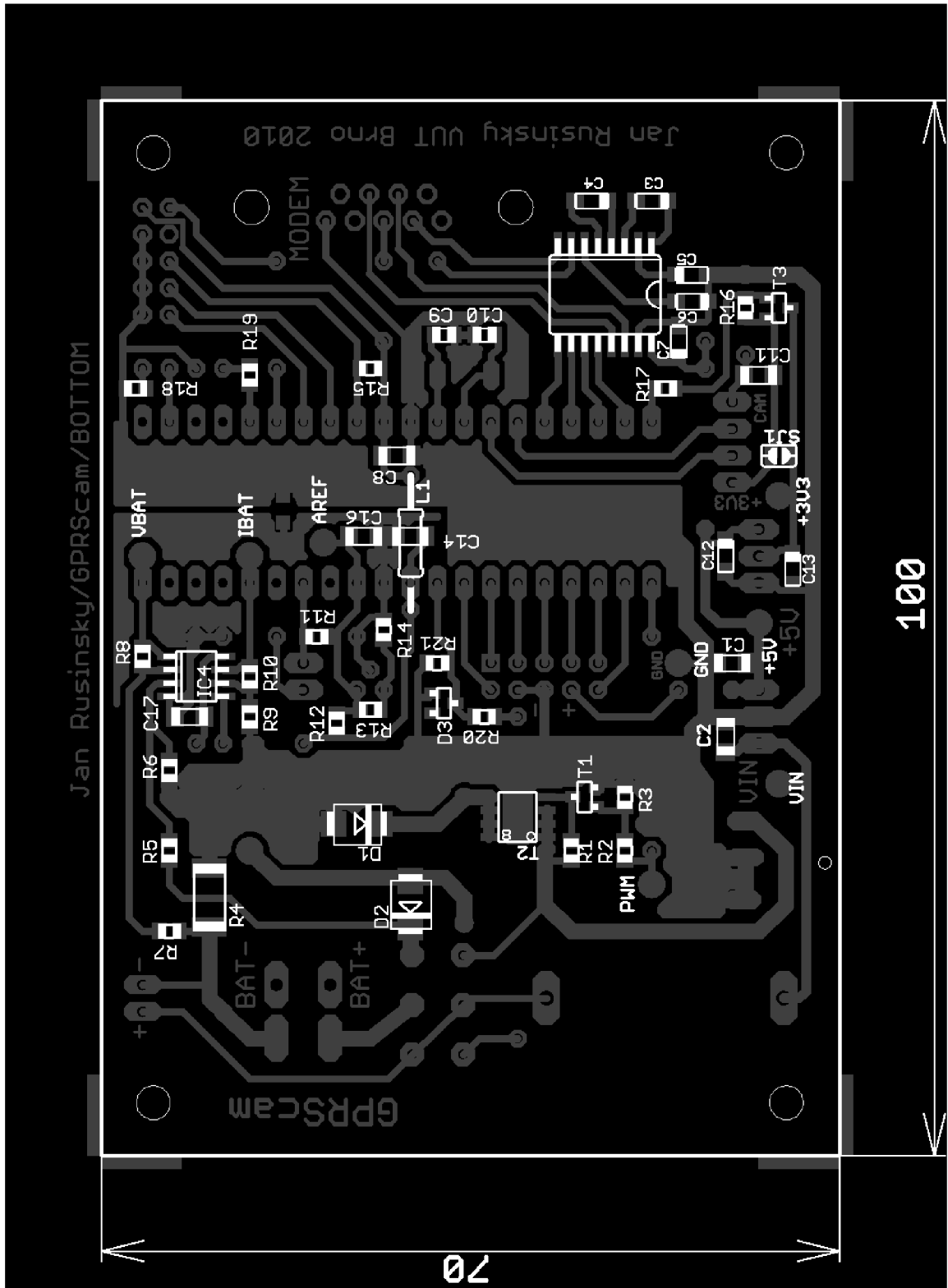
Date: 2010-05-24 23:47:25

Sheet: 1/1

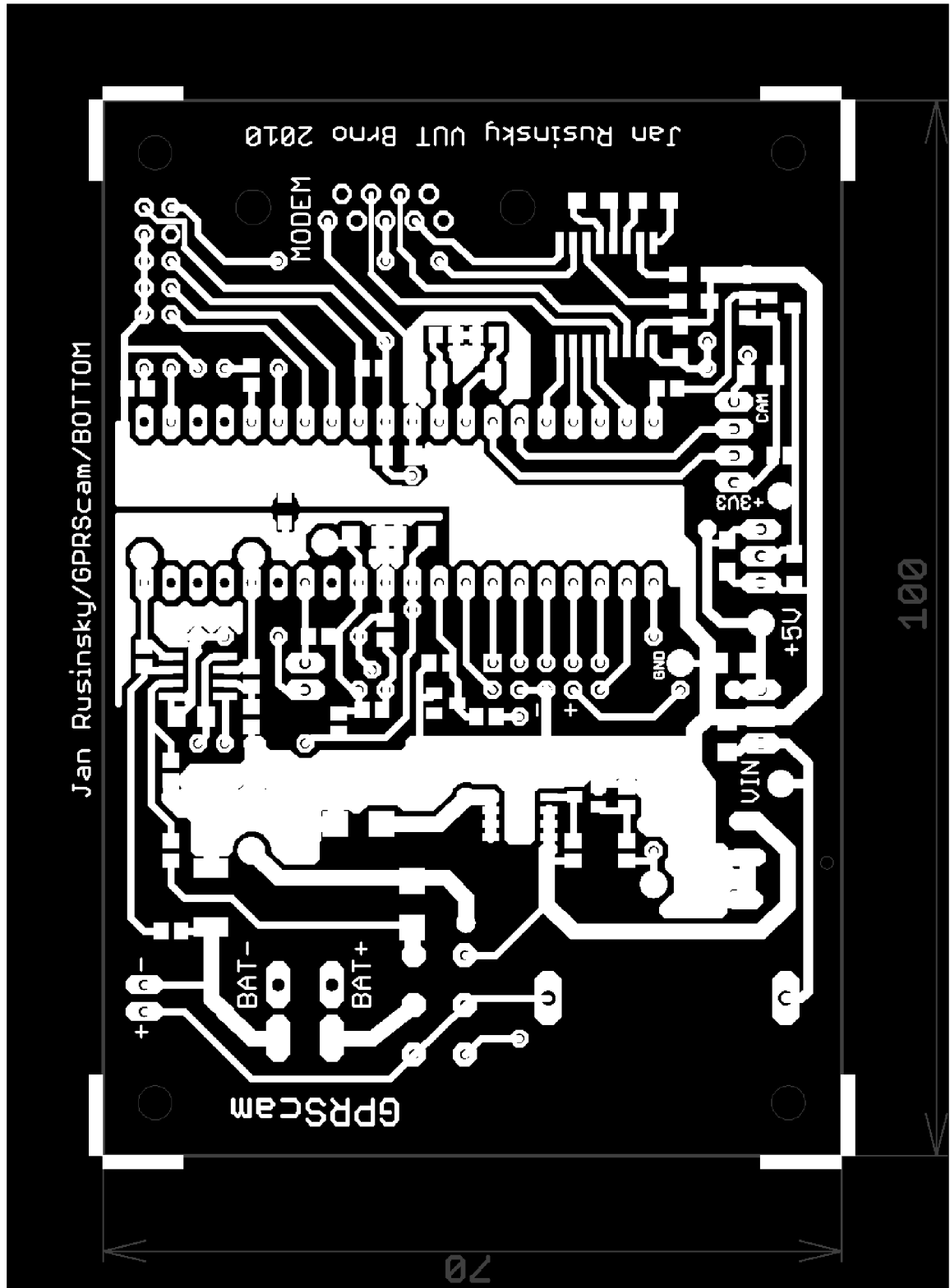
Příloha B – motiv plošného spoje



Příloha C – osazovací výkres, strana plošného spoje



Příloha D – osazovací výkres, strana součástek



Příloha E – záznam komunikace uP s modemem

```
11s { \r\n +WIND: 13 \r\n
      \r\n +WIND: 1 \r\n
      \r\n +WIND: 7 \r\n

      ATE0 \r
      ATE0 \r
      \r\n OK \r\n

      AT \r
      \r\n OK \r\n

      AT+CPIN=5678 \r
      \r\n OK \r\n

16s { \r\n +WIND: 15,1,"O2 - CZ ",2,"O2-CZ
      ",4,"10/02/27,16:26:35+04",6,"0" \r\n
      \r\n +WIND: 16 \r\n
      \r\n +WIND: 4 \r\n
      \r\n +WIND: 10,"SM",1,"FD",1,"ON",1,"EN",0 \r\n
      \r\n +WIND:
      11,"1C2CFC7EB025074D63645FA24C8EB1F5","AC85FFEB70367380F
      BE629BD893AE5CF","5B5745D37FC68FCB9E1914057968C7F3",,"C0
      DD820F17C6AA0932F395CE24268EF3","14BC0493A2ED6C8BA4D579A
      0C0403430" \r\n

      AT+WIPCFG=1 \r
      \r\n OK \r\n

      AT+WIPBR=1,6 \r
      \r\n OK \r\n

      AT+WIPBR=2,6,11,"APNname" \r
      \r\n OK \r\n

5s { AT+WIPBR=4,6,0 \r
      \r\n OK \r\n

1s { AT+WIPCREATE=2,1,"12.34.56.78",50023 \r
      \r\n OK \r\n
      \r\n +WIPREADY: 2,1 \r\n
      \r\n +WIPDATA: 2,1,1 \r\n

      AT+WIPDATA=2,1,2 \r
      \r\n CONNECT \r\n
      COMMAND

3s { pic.jpg+<DEL>+<DEL>
      +++
1s { \r\n OK \r\n

      AT+WIPCLOSE=2,1 \r
      \r\n OK \r\n
```

Červená - AT příkazy
z mikrokontroléru
Modrá - odpovědi od modemu
\r - <CR>
\n - <LF>

Příloha F – zdrojový kód webového rozhraní

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1250">
<meta name="generator" content="PSPad editor, www.pspad.com">
<title>GPRScam web interface</title>
</head>
<body>
<script type="text/javascript">
var casovac=""
function stop_refresh()
{
if(casovac!="")
{
window.clearInterval(casovac)
casovac=""
}
}
function set_refresh(time)
{
stop_refresh()
casovac=window.setInterval("frame_reload()", time)
}
function frame_reload ()
{
}
var frame = document.getElementById('pic_frame');
frame.src = frame.src;
}
</script>
<div align="center">
<br>
<iframe src="http://10.0.0.140/pic" name="photo" align="middle" id="pic_frame" width="640" height="480">
</iframe>
<br><br>
<b> set refresh time</b>
<form action="http://10.0.0.140" method="GET" target="dummy">
<button name="refresh" type="submit" value="3" onclick="set_refresh(60000)"><b>60s</b></button>
<button name="refresh" type="submit" value="2" onclick="set_refresh(30000)"><b>30s</b></button>
<button name="refresh" type="submit" value="1" onclick="set_refresh(10000)"><b>10s</b></button>
</form>
<br>
<b> rotate camera</b>
<form action="http://10.0.0.140" method="GET" target="dummy">
<button name="left" > <b>left</b></button>
<button name="right" > <b>right</b></button>
</form>
<b> server response</b>
<br>
<iframe src="dummy.html" name="dummy" align="center" width="300" height="30">
</iframe>
</div>
</body>
</html>
```