



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘENOS STATICKÉHO OBRAZU Z CCD KAMERY S MIKROKONTROLÉREM

MICROCONTROLLER BASED SYSTEM FOR ACQUIRING AND TRANSFER OF STATIC PICTURE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN VLADAŘ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ZDENĚK BRADÁČ, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Martin Vladař

ID: 120514

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Přenos statického obrazu z CCD kamery s mikrokontrolérem

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci mikrokontrolérového systému pro sejmutí a přenesení statického obrazu z CCD kamery pro komunikační systémy s extrémně omezeným komunikačním datovým tokem. Systém navrhněte jako miniaturní systém vybavený mikrokontrolérem a nezbytnými rozhraními. Navrhněte elektroniku, realizujte DPS, osadte a oživte. Navrhněte demonstrační úlohu pro jeho využití. Vybavte programovým vybavením, otestujte a předvedte funkčnost.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6
Dle pokynů vedoucího práce.

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

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

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Bakalářská práce se zabývá problematikou přenosu statického obrazu z CCD kamery při extrémně omezeném komunikačním datovém toku. Základ systému tvoří mikroprocesor, který s periferiemi komunikuje převážně přes sériovou sběrnici. Obraz je získáván PC programem pro operační systém Microsoft Windows, který umí pořídít barevný obraz a obraz v odstínech šedi. Vadou systému je podle očekávání pomalá sériová linka.

Klíčová slova

Mikroprocesor, mikrokontrolér, ATmega128A, kamera, CCD, ITM-C-328, statický obraz, Bluetooth, FRAM, FM25H20, RS-232, UART, USART, SPI, pořizování obrazu, 320x240, RGB565, Grayscale, C, C#, Eagle, Makefile, Objektově orientované programování, AVR ISP, AVR Dragon, USB2UART&BT

Abstract

This thesis deals with the transmission of still images from the CCD camera at extremely limited communication data flow. The base system consists of a microprocessor, which communicates with the peripherals mostly via the serial interface. The image is obtained by PC program for Microsoft Windows that can capture color images and grayscale images. A defect of the system is as expected slow serial line.

Keywords

Microprocessor, microcontroller, ATmega128A, camera, CCD, ITM-C-328, static picture, Bluetooth, FRAM, FM25H20, RS-232, UART, USART, SPI, picture capturing, 320x240, RGB565, Grayscale, C, C#, Eagle, Makefile, Object oriented programming, AVR ISP, AVR Dragon, USB2UART&BT

Bibliografická citace:

VLADAŘ, M. Přenos statického obrazu z CCD kamery s mikrokontrolérem. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 47s. Vedoucí bakalářské práce byl doc. Ing. Zdeněk Bradáč, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Přenos statického obrazu z CCD kamery s mikrokontrolérem jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **27. května 2013**

.....
podpis autora

Poděkování

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

V Brně dne: **27. května 2013**

.....
podpis autora

OBSAH

Úvod.....	10
1 LITERÁRNÍ REŠERŠE	11
2 Koncepce systému	12
2.1 Mikroprocesor	14
2.2 Kamera.....	15
2.3 Paměť.....	16
2.4 Bluetooth modul	17
2.5 Doplnkové obvody	18
3 Návrh obvodového schématu	19
3.1 ISP konektor	20
3.2 CAM konektor.....	21
3.3 BT konektor	22
3.4 FM25H20 konektor	23
3.5 Indikace stavu baterie	24
3.6 Reset a krystal	25
4 Návrh desky plošných spojů	26
4.1 Konektory	27
4.2 Rozložení součástek.....	28
4.3 Osazení a oživení DPS.....	29
5 PROGRAM MIKROPROCESORU.....	30
5.1 Seznam a popis funkcí	31
5.2 Parsované příkazy mikroprocesoru.....	34
5.2.1 Příkaz „A“	34
5.2.2 Příkaz „B“	34
5.2.3 Příkaz „C“	34
5.2.4 Příkaz „D“	35
6 PC PROGRAM	36
6.1 Grafické uživatelské prostředí Capture.....	36
6.1.1 Picture box.....	37
6.1.2 UART port.....	37
6.1.3 Colour type.....	38
6.1.4 Capture new.....	38
6.1.5 Capture last.....	38

6.1.6	Battery status	38
6.1.7	Save image	39
6.2	Pořízení obrazu.....	40
7	Závěr.....	42
	Literatura.....	43
	Seznam příloh.....	44

SEZNAM OBRÁZKŮ

Obrázek 1:	Koncepce systému	13
Obrázek 2:	ATMEGA128A pinout [1]	14
Obrázek 3:	Kamera TIM-C-328 náhled, pinout a rozměry [2].....	15
Obrázek 4:	Pinout FRAM FM25H20 s popisem pinů [3]	16
Obrázek 5:	Bluetooth modul	17
Obrázek 6:	USB2UART&BT modul	17
Obrázek 7:	Návrhové schéma obvodu	19
Obrázek 8:	Programátor mikroprocesorů AVR Dragon.....	20
Obrázek 9:	Zapojení kamery k mikroprocesoru	21
Obrázek 10:	Schéma zapojení konektoru pro Bluetooth nebo USB2UART&BT modul	22
Obrázek 11:	F-RAM pinout [3]	23
Obrázek 12:	Schéma zapojení napájení pro A/D převodník [1]	24
Obrázek 13:	Zapojení krystalu a tlačítka reset s pull-up rezistorem	25
Obrázek 14:	TOP (vlevo) a BOTTOM (vpravo) vrstvy desky plošných spojů	26
Obrázek 15:	Šroubovací svorkovnice	27
Obrázek 16:	Konektor 2x3 pro programátor AVR Dragon	27
Obrázek 17:	Návrh desky plošných spojů	28
Obrázek 18:	Vzhled kompletního zapojení	29
Obrázek 19:	Úvodní okno programu Capture 1.0.....	36

Obrázek 20: Skladba 16-bit RGB (RGB565)	37
Obrázek 21: Výběr sériového portu	37
Obrázek 22: Náhled na místo uložení souboru a náhled samotného souboru	39
Obrázek 23: Pořízení obrazu v odstínech šedi	40
Obrázek 24: Pořízení barevného obrazu (16-bit RGB)	40

SEZNAM TABULEK

Tabulka 1: Popis pinů F-RAM [3]	23
Tabulka 2: Příkazy pro nastavení kamery [2]	33

ÚVOD

Před realizací samotné bakalářské práce jsme si museli položit otázku, k čemu vlastně bude zařízení sloužit. Projekt je zadán pouze pro realizaci jako bakalářská práce a s největší pravděpodobností nebude nikdy nasazen do praxe. Proto jsme si teoretické využití vymysleli. Po konzultaci přišel nápad, že kamera bude umístěna na helmě záchranáře a bude sloužit k jednorázovému sejmutí obrazu v případě, že bude záchranář v nouzové situaci a nebude moci podat informaci o své poloze. Tento snímek potom bude sloužit pro ulehčení jeho lokalizace především v budovách, kde je obtížné nebo nemožné použít například GPS lokátor.

Samotná kamera bude umístěna na přední straně helmy a na zadní straně bude námi navržený obvod s bateriemi a modulem pro komunikaci.

V případě uvíznutí záchranáře si nechá technická podpora poslat snímek ze záchranářovy helmy a díky němu může navigovat ostatní záchranáře k postiženému. Snímek také může sloužit pro identifikaci podezřelých objektů jako jsou výbušniny nebo chemické zbraně.

1 LITERÁRNÍ REŠERŠE

Námi navržené zařízení má sloužit jako bezpečnostní opatření pro záchranáře, který bude mít zvýšenou šanci na záchranu například při ztrátě vědomí. Pokud dojde ke ztrátě vědomí, záchranný tým si vyžádá obraz z kamery umístěné na helmě nebo jiné části záchranářova vybavení. Výhodou je samostatná komunikace pomocí bluetooth, takže se dá zařízení použít i při zničení vysílačky nebo jinak způsobené ztrátě komunikace.

V praxi jsme na podobnou koncepci systému využívanou záchranáři narazili pouze od firmy HORNET, která ovšem dělá termovizní kamery pro požární zásah a jsou primárně určeny k orientaci za snížené viditelnosti. Rádiový modul, kterým se dají vybavit je nadstandardní, a primárně tedy není zařízení určeno pro zpětnou vazbu od záchranáře.

Dále je na trhu široká škála kamer k připevnění na helmu, které jsou ale určeny pro pořizování záznamu při sportu nebo při jízdě autem či na motorce. Žádný z těchto modelů však nenabízí posílání dat z kamery v reálném čase.

K posílání dat v reálném čase jsou na druhou stranu určeny bezpečnostní kamery, zejména IP kamery, které pro účely záchranářů nemohou být použity pro své rozměry a absenci baterií.

Téměř všechny kamery používané záchranáři jsou infrakamerami. Důvodem je nutnost orientace v prostorách se sníženou viditelností kvůli dýmu nebo špatnému osvětlení. Infrakamera také odhalí vysokou teplotu předmětů v místnosti, které mohou být pro záchranáře nebezpečné, ale na první pohled je neodhalí.

2 KONCEPCE SYSTÉMU

Pro realizaci systému budeme potřebovat programovatelný mikroprocesor, který bude dostatečně rychlý, levný a je jej možné programovat v jazyce C. Mikroprocesor bude posílat požadavek na obraz do CCD kamery, která bude posílat obraz. Mluvíme tedy o obousměrném provozu na sériové lince RS-232, která je dána kamerou, kterou budeme používat a která je vlastně jedinou pevně danou periferií.

Tato kamera umí snímat obraz v rozlišení až 640x480 s 24-bitovou barevnou hloubkou. To odpovídá velikosti 900kB na jeden snímek bez komprese v surovém formátu. Tato obrovská datová náročnost se nedá v rozumném časovém horizontu zpracovávat. Snížili jsme tedy nároky a to jak na rozlišení tak na barevnou hloubku snímků, abychom zaručili rozumnou dobu pořizování snímku.

Představu o tom, co je ještě rozumná velikost pro přenos obrazu pomocí sériové sběrnice, nám udala také velikost použité paměti. Rozhodli jsme se pro paměť RAMTRON FM25H20, která má velikost 2Mb a komunikuje pomocí SPI sběrnice. Osazovat obvod větší paměti je kvůli rychlosti zbytečné.

Pro přenos fotografie z paměti do PC bude sloužit BT modul, který je používán pro bezdrátový přenos dat opět sériovou linkou UART a opět na něm bude probíhat obousměrný provoz, kvůli poslání požadavku na pořízení snímku a pro přenos samotného snímku druhým směrem.

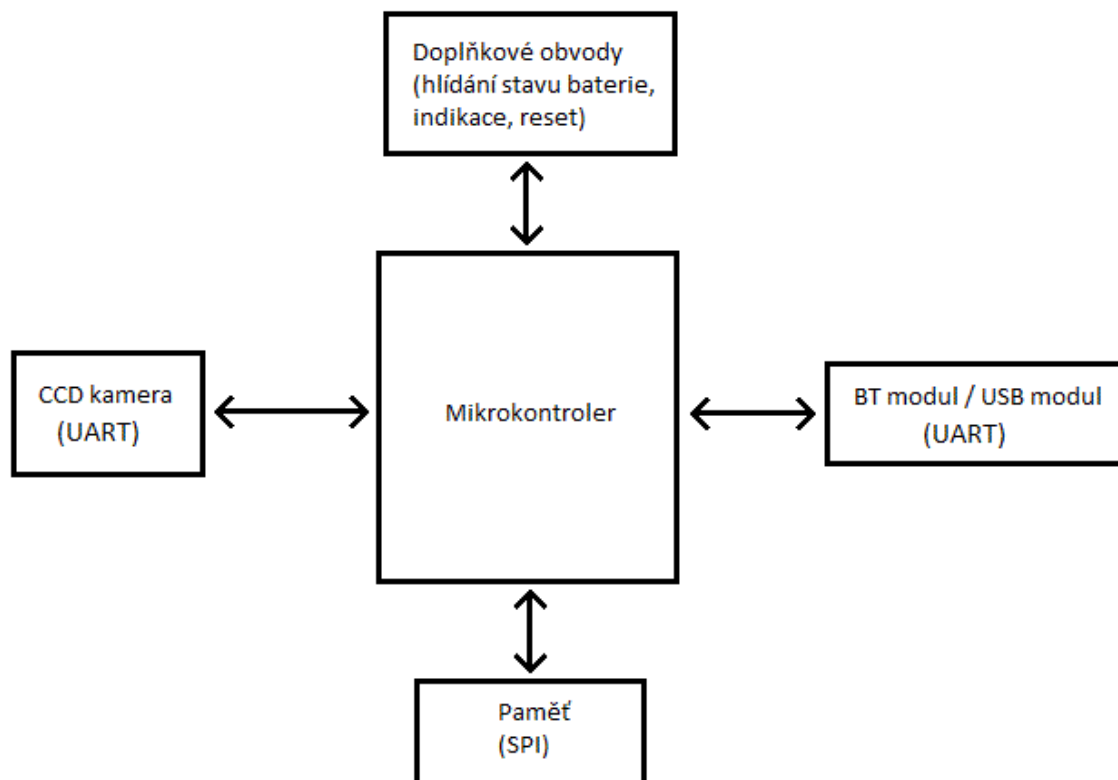
Další věcí, kterou by zařízení, které má zachraňovat životy mělo umět je hlídání stavu baterie. To je řešeno odporovým děličem napětí baterie, jehož výstupní napětí budeme srovnávat s referenčním napětím mikroprocesoru. Stav baterie bude indikovat dioda na zařízení, samotné zařízení bude posílat stav baterie do PC pomocí zmíněného BT modulu a pořízený obraz bude obsahovat symbol baterie s indikací nabití v době pořízení snímku.

Celé zařízení musí být malé a bude se napájet dvěma tužkovými bateriemi (AA nebo AAA), takže všechny jeho části musí umět pracovat na napětí (2x1,5V) 3V.

Programové vybavení mikroprocesoru bude realizováno v jazyce C. Bude kompilováno a nahráváno do mikroprocesoru pomocí Makefile, který bude mimo jiné obsahovat programy avr-gcc a avr-dude používané pro programování mikroprocesorů na unixových systémech.

Jako alternativa je možné použít program AVR Studio, popřípadě Atmel Studio pro programování mikroprocesorů firmy Atmel dodávaný přímo výrobcem.

PC program pro pořízení a zobrazení obrazu bude mít jednoduché uživatelské rozhraní a budeme jej realizovat ve Visual Studiu v jazyce C#. Bude jistě obsahovat ovládací prvky pro otevření sériové komunikace a pro pořízení a vykreslení obrazu.



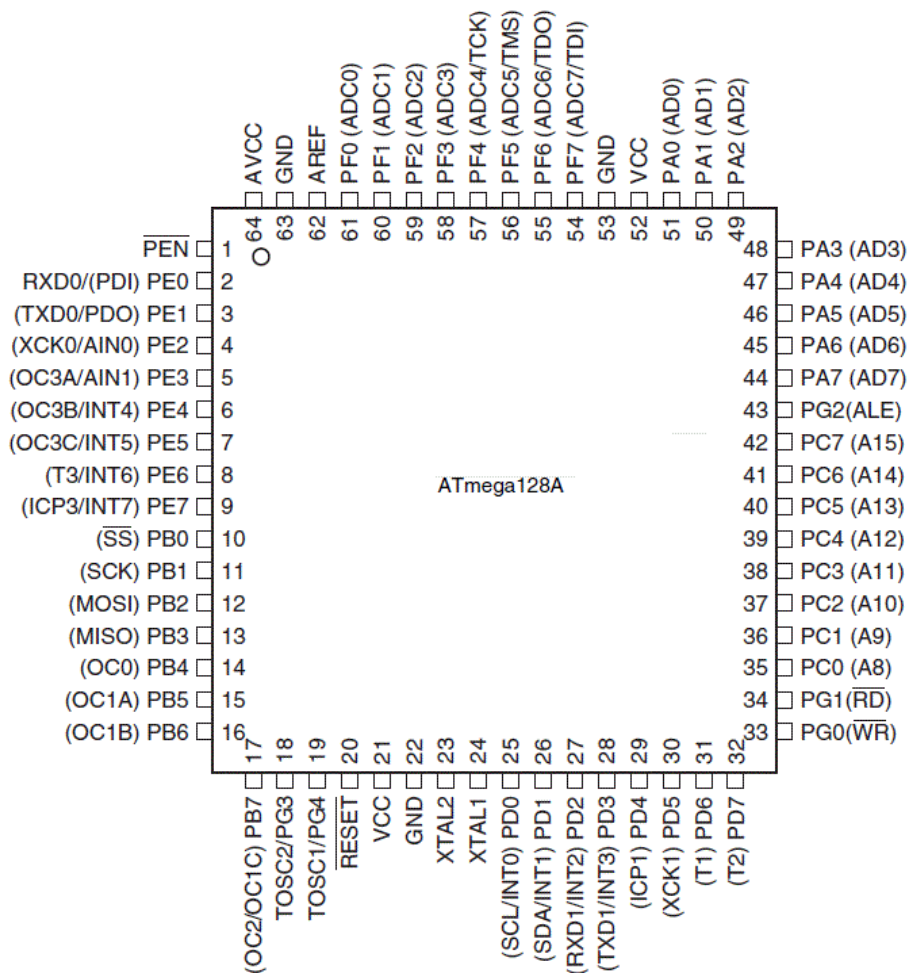
Obrázek 1: Koncepce systému

2.1 Mikroprocesor

Výběru procesoru jsme věnovali hodně pozornosti. Potřebovali jsme takový mikroprocesor, který umí obsluhovat všechny typy sběrnic v dostatečném počtu, je dost rychlý, aby sám nezpomaloval komunikaci alespoň po nejpomalejší z použitých sběrnic.

Potřebujeme tedy procesor, který je schopen obsluhovat sběrnici SPI pro paměť, dvě sběrnice UART pro kameru a BT modul, má zabudovaný A/D převodník a několik programovatelných vstupů/výstupů pro diody.

Zvolili jsme 8mi-bitový mikroprocesor od firmy Atmel, který umí pracovat na frekvenci až 16MHz, umí obsluhovat všechny potřebné sběrnice a pracuje i na námi požadovaném napětí 3V. Všem kritériím odpovídá zvolený procesor ATMEGA128A AU 1240.



Obrázek 2: ATMEGA128A pinout [1]

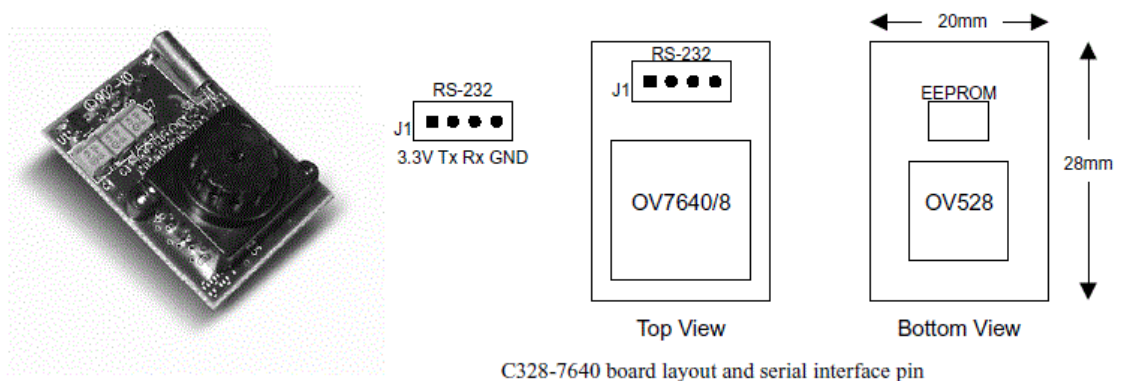
2.2 Kamera

Kamera byla základním stavebním kamenem projektu a celý návrh se odvíjel právě od ní. Jedná se o CCD kameru od firmy Intertec Components ITM-C-328.

Tato kamera umí snímat obraz v rozlišení až 640x480 s 24-bitovou barevnou hloubkou. Má zabudovaný JPEG codec a barevný převodník, takže umí jak černobílý (obraz v stupních šedi), tak i barevný obraz. Kamera komunikuje přes UART sběrnici, přes kterou se nastavuje a ovládá pomocí 6ti-bajtových příkazů. Umí posílat náhled v nízkém rozlišení (do 160x120) a obraz v rozlišení větším (až 640x480). Pro naši aplikaci nám bude stačit rozlišení 320x240 kvůli extrémně omezenému datovému toku při použití UART sběrnice. Jak jde vidět na obrázku 3, jedná se o 3-vodičové zapojení sériové sběrnice, tedy asynchronní přenos dat.

Samotná kamera se ovládá pomocí 6ti-bajtových příkazů a komunikuje až na rychlosti 115200 BAUD, čehož se také budeme snažit využít pro zrychlení přenosu obrazu. Její hlavní nevýhodou je způsob synchronizace, který spočívá v posílání SYNC příkazu maximálně 60krát za sebou a podle datasheetu se obvykle připojí na 25. pokus. Takovéto řešení neshledáváme moc šťastným. Mohlo by spíše začít komunikaci speciálním bajtem při určené rychlosti přenosu.

Také samotný datasheet a jeho popis pro nastavení rozlišení a barevné hloubky není tak docela pravdivý a kamera pracuje v režimu, který v datasheetu není vůbec uveden a určili jsme jej více méně experimentálně.



Obrázek 3: Kamera TIM-C-328 náhled, pinout a rozměry [2]

2.3 Paměť

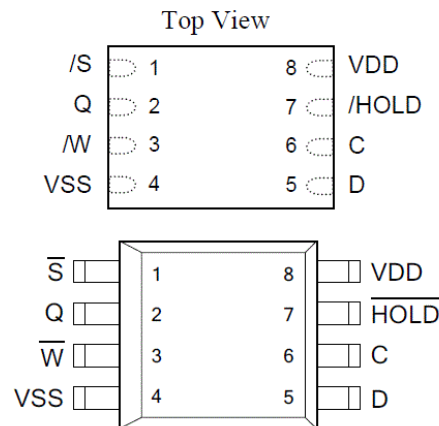
Po původním uvažování o paměti komunikující po TWI sběrnici jsme se nakonec přiklonili k řešení s pamětí RAMTRON FM25H20, která má velikost 2Mb, a komunikuje po SPI sběrnici. Tato paměť se vyrábí pouze v pouzdrech 8-pin TDFN nebo 8-pin RIAJ SOIC, která se obtížně letují na DPS, a proto ji osadíme na redukci na pouzdro DIL8. Tuto paměť i redukci na DIL8 dodal náš konzultant a její realizace není našim dílem.

V rámci bakalářské práce jsme se starali pouze o programové vybavení a obsluhu paměti.

Výhoda Feritové paměti je v tom, že zachovává uložená data i po odpojení napájení. Díky této vlastnosti můžeme s kamerou realizovat jednoduchý fotoaparát s pamětí na jeden snímek a díky tomu, že snímek z kamery pořídíme v surovém maticovém formátu, můžeme do pořízeného obrazu „kreslit“, protože orientace v matici není žádný problém. Maticí rozumějme sice jednorozměrné pole, u kterého ale známe počet řádků a sloupců, takže s ním pracujeme jako s maticí (dvourozměrným polem).

Tato paměť podporuje jak funkci Hold, tak i Write protection, které my ale nepoužíváme. Hold je zbytečná úplně, protože používáme pouze jednu paměť a Write protection zde shledáváme také zbytečným, protože paměť bude sloužit právě tak ke čtení jako k zápisu, není potřeba žádná data blokovat pouze pro čtení.

Pin Configuration



Pinout is equivalent to other SPI F-RAM devices.

Pin Name	Function
/S	Chip Select
/W	Write Protect
/HOLD	Hold
C	Serial Clock
D	Serial Data Input
Q	Serial Data Output
VDD	Supply Voltage (2.7 to 3.6V)
VSS	Ground

Obrázek 4: Pinout FRAM FM25H20 s popisem pinů [3]

2.4 Bluetooth modul

Pro realizaci komunikace s PC jsme zvolili bluetooth modul osazený Wireless Bluetooth V2.0 RS232 TTL Transceiver Module. Modul kromě samotného bluetooth modulu obsahuje ještě signalizační LED a má vyveden pinout na 4-pinovou lištu do DPS. Díky tomu se dá použít jako zásuvný modul a je možné jej vyměnit za jiný se stejným (nebo podobným) pinoutem. Toho jsme také využili při komunikaci se stolním PC, který není vybaven bluetooth modulem a nahradili jsme jej USB2UART modulem. Bluetooth modul je programovatelný a je možné u něj měnit rychlost až na 115 200BAUD, což je také maximální rychlost, kterou zvládá CCD kamera. Máme tedy stejnou rychlost obou sériových linek, i když ani nemusíme, protože stejně budeme používat paměť, která nám odstraňuje problém synchronizace rychlostí těchto UART linek.



Obrázek 5: Bluetooth modul



Obrázek 6: USB2UART&BT modul

2.5 Doplnkové obvody

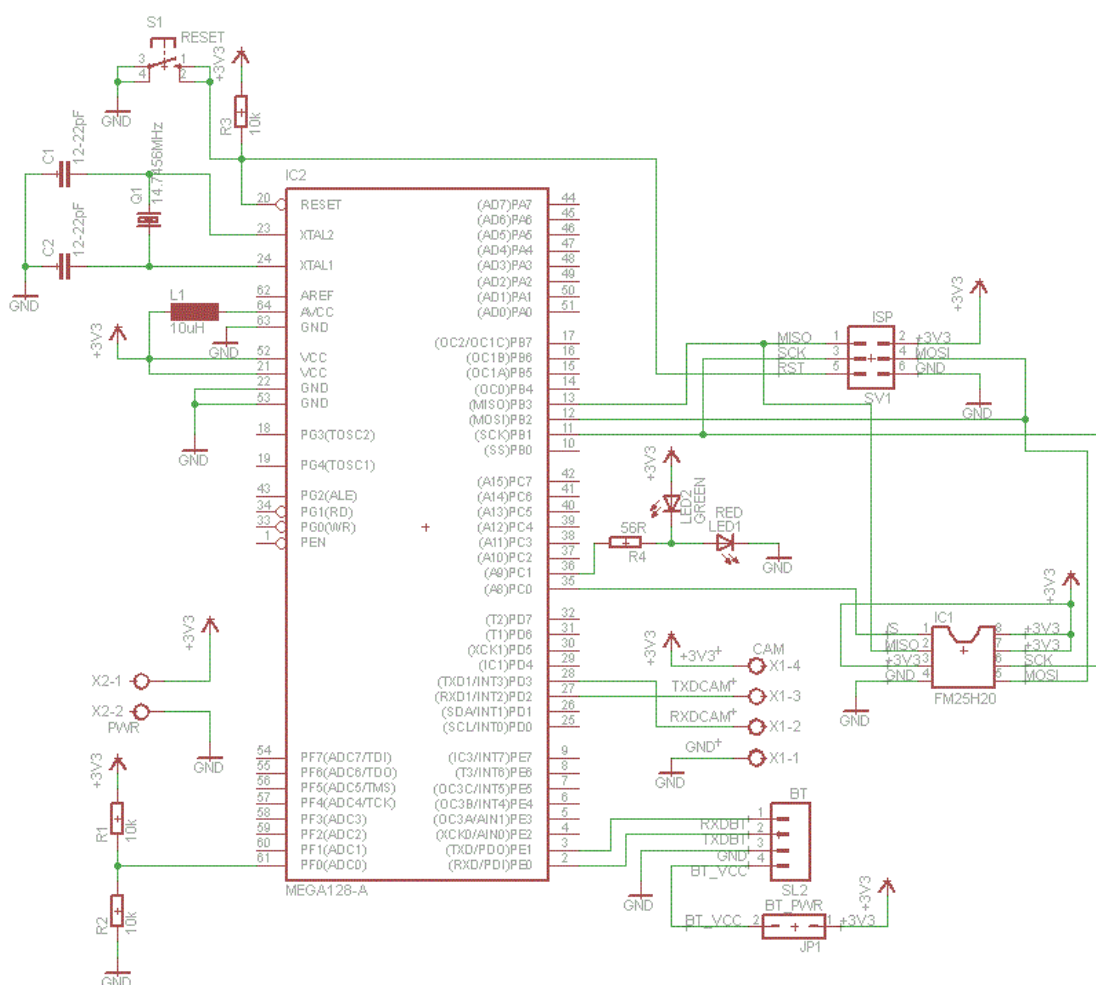
K mikroprocesoru je dále připojeno ještě několik obvodů. Jedná se o reset, který je nutností při odladování, dále odporový dělič připojený k A/D převodníku pro hlídání stavu baterie a signalizační LED diody zapojeny proti sobě, ať vždy svítí pouze jedna (červená/zelená). Krystal o frekvenci 7,3728MHz jsme zvolili kvůli větší přesnosti UART komunikace (viz kapitola 3.6).

Všechny doplňkové obvody jsou detailněji popsány kapitole 3 u návrhu desky plošných spojů.

3 NÁVRH OBVODOVÉHO SCHÉMATU

Návrh schématu jsme realizovali v programu Eagle 6.3.0 Light. Vzhledem k tomu, že nepotřebujeme vytvořit desku plošných spojů větší než 10x10cm, stačila nám bezplatná verze programu.

Samotné schéma se skládá z několika částí, které budou popsány samostatně v této kapitole.



Obrázek 7: Návrhové schéma obvodu

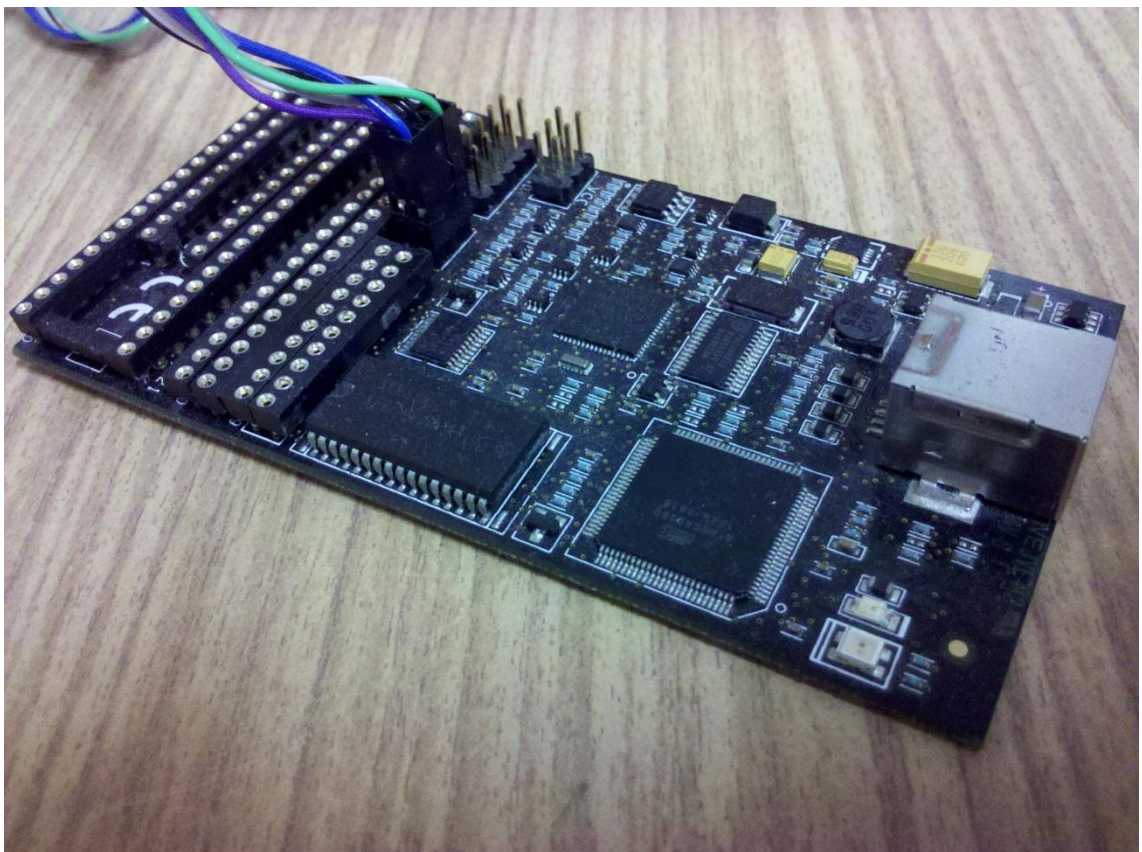
3.1 ISP konektor

Pro programování mikroprocesoru budeme používat programátor AVR Dragon, který je schopen programovat mikroprocesory pomocí ISP, JTAG a Paralelního programování.

Toho se dalo využít pro debugging a používat JTAG. My ale máme k dispozici bluetooth modul a po sériové lince tedy komunikujeme s PC, takže v případě komplikací při programování můžeme kontrolovat průchod programem a jeho větvení pomocí výpisů do terminálu, s čímž už máme zkušenosti z dřívějších projektů. Proto jsme zvolili programování přes SPI sběrnici typu ISP (In-System Programming).

Programátor AVR Dragon podporuje jak unixové operační systémy, tak operační systémy společnosti Microsoft.

Jedná se o velice silný nástroj, ať už jde o možnost využít všechny způsoby programování, které mikroprocesory Atmel nabízejí, tak třeba o možnost aktualizace firmware. Tomu také odpovídá jeho pořizovací cena.

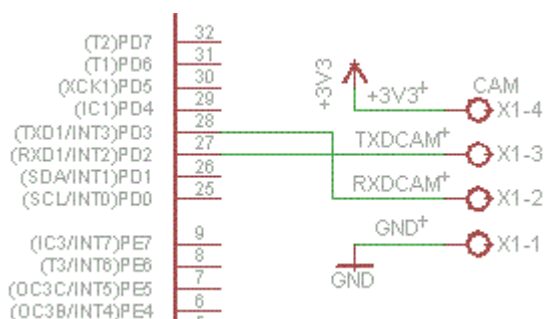


Obrázek 8: Programátor mikroprocesorů AVR Dragon

3.2 CAM konektor

Tento konektor slouží pro komunikaci po sériové sběrnici UART s kamerou. Pro naši aplikaci potřebujeme zapojit piny RXD a TXD, které se připojují do kříže (obdobně jako křížený síťový kabel pro komunikaci dvou PC přímým propojením u starších síťových karet) a pin GND. Samozřejmě samotná kamera se neumí napájet z datových vodičů, jak je možné vidět u některých snímačů v automatizaci, takže ji napájíme 3,3V (respektive 3V z baterií), podle datasheetu kamery [2].

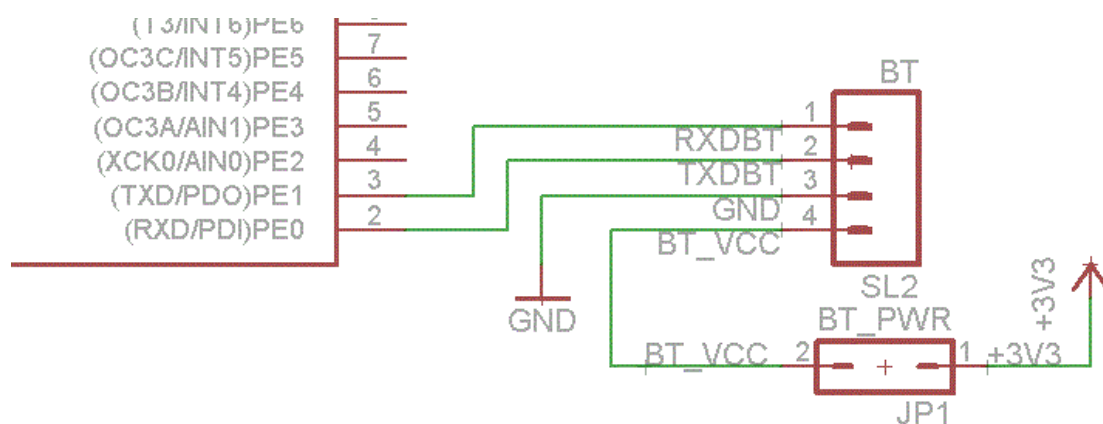
Jedná se tedy o tří-vodičové zapojení sériové linky a asynchronní přenos dat.



Obrázek 9: Zapojení kamery k mikroprocesoru

3.3 BT konektor

Tento konektor slouží k připojení bluetooth modulu nebo USB2UART&BT převodníku pro komunikaci s PC podle přítomnosti nebo absence bluetooth rozhraní na PC. Protože ze zkušenosti víme, že testování pomocí bluetooth modulu je časově náročnější kvůli párování (zejména neustálému zadávání hesla) a není součástí základních desek, což vede k jeho absenci na stolních PC, přidali jsme možnost odpojit napájení tohoto konektoru klasickým „jumperem“, aby se dalo s mikroprocesorem komunikovat přes USB pomocí USB2UART&BT převodníku, který je napájen z USB (země musí být spojeny). Kromě odpojitelného napájení se jedná o kopii CAM konektoru.



Obrázek 10: Schéma zapojení konektoru pro Bluetooth nebo USB2UART&BT modul

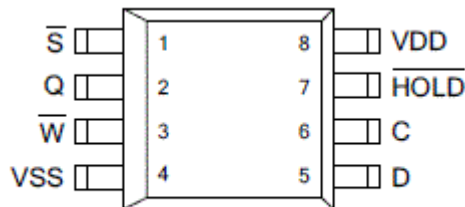
3.4 FM25H20 konektor

V tomto konektoru je zapojena paměť RAMTRON FM25H20, která komunikuje s mikroprocesorem po SPI sběrnici. Paměť má vyvedené konektory pro MISO, MOSI, SCK a Chip Select, které nás zajímají. Zbylé Hold a Write protection nebudeme používat a protože jsou negované, připojili jsme je na napětí 3V. Tady dochází k malé kolizi, kdy máme na SPI sběrnici připojen programátor a paměť. Při tomto zapojení může dojít k dvěma případům.

Jeden možný případ nastane, pokud při programování bude na pinu PC0 mikroprocesoru při programování logická jednička. V takovém případě se nestane vůbec nic, protože bude paměť neaktivní a nebude se na ni zapisovat.

Druhý případ počítá s logickou nulou na výstupu pinu PC0 a paměť je tedy aktivní a dochází k nahrání programu do mikroprocesoru i do paměti, což ale nepovažujeme za problém, protože získaný obraz do paměti budeme zapisovat od jejího začátku a přepíšeme její obsah. Případně paměť v inicializaci smažeme a vyřešíme problém softwarově.

Jako řešení problému nás ještě napadlo použít pull-up rezistor, který by fungoval, pokud by pin PC0 byl nastaven jako vstupní a bylo na něj zapisováno. Pokud by ale byl nastaven jako výstupní, na paměťovém pinu Chip selectu by se stejně mohla objevit logická nula a došlo by k zápisu do paměti. Proto jsme tento jev neřešili, pouze si uvědomujeme možnost jeho vzniku.



Pinout is equivalent to other SPI F-RAM devices.

Obrázek 11: F-RAM pinout [3]

Pin Name	Function
/S	Chip Select
/W	Write Protect
/HOLD	Hold
C	Serial Clock
D	Serial Data Input
Q	Serial Data Output
VDD	Supply Voltage (2.7 to 3.6V)
VSS	Ground

Tabulka 1: Popis pinů F-RAM [3]

3.5 Indikace stavu baterie

Stav baterie zjišťujeme pomocí A/D převodníku. Protože mikroprocesor bude pracovat na napětí 3V, jeho referenční napětí budeme volit vnitřních 2,56V.

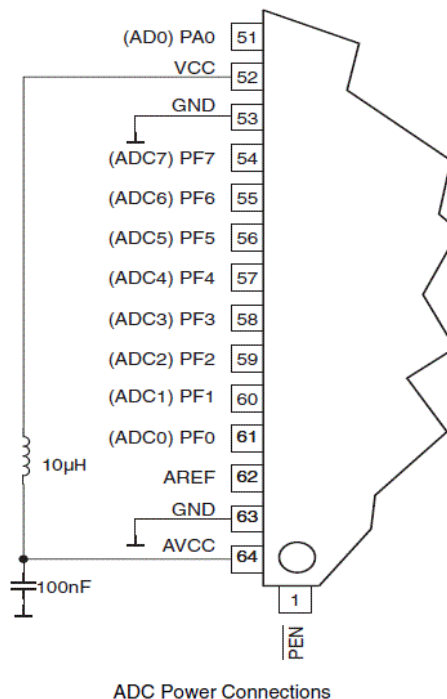
Na vstup A/D převodníku potom přivedeme napětí, jehož hodnota nesmí překročit naši referenční hodnotu 2,56V. To jsme vyřešili jednoduše odporovým napěťovým děličem s poměrem odporů R1 a R2 1 ku 1, což znamená poloviční napětí na výstupu děliče. Díky tomu na vstup A/D převodníku dostaneme polovinu napětí baterií, tedy 1,5V. Tato hodnota je teoretická a hodnota napětí baterií se může lišit. Takto volený napěťový dělič nám zaručí funkčnost i při velkých odchylkách napětí baterií, a to až do napětí $2 \cdot 2,56V$, tedy 5,12V, což by nevydržela jak kamera, tak paměť, takže nefunkčnost indikace baterie by byl ten nejmenší problém.

Práh, kde je baterie ještě nabitá a vybitá zjistíme experimentálně, podle schopnosti obvodu ještě bezchybně fungovat. Viz kapitola 5.2.3

V obvodu jsou 2 LED diody zapojeny proti sobě tak, aby se daly ovládat jedním pinem mikroprocesoru. Jedna je červená a druhá zelená. Vzhledem k tomu, že mají stejné parametry proudu a napětí, jsou připojeny na jeden stejný odpor, který se stará o úbytek napětí aby na diody nešlo 3V, ale pouze 1,88V ($3 - (0,02A \cdot 56\Omega)V$).

Tyto diody se budou starat kromě chyb komunikace a synchronizace právě i o indikaci vybitých baterií.

Pro napájení A/D převodníku slouží napájecí pin AVCC mikroprocesoru, který má být připojen k VCC přes nízko-frekvenční propust' podle datasheetu [1].

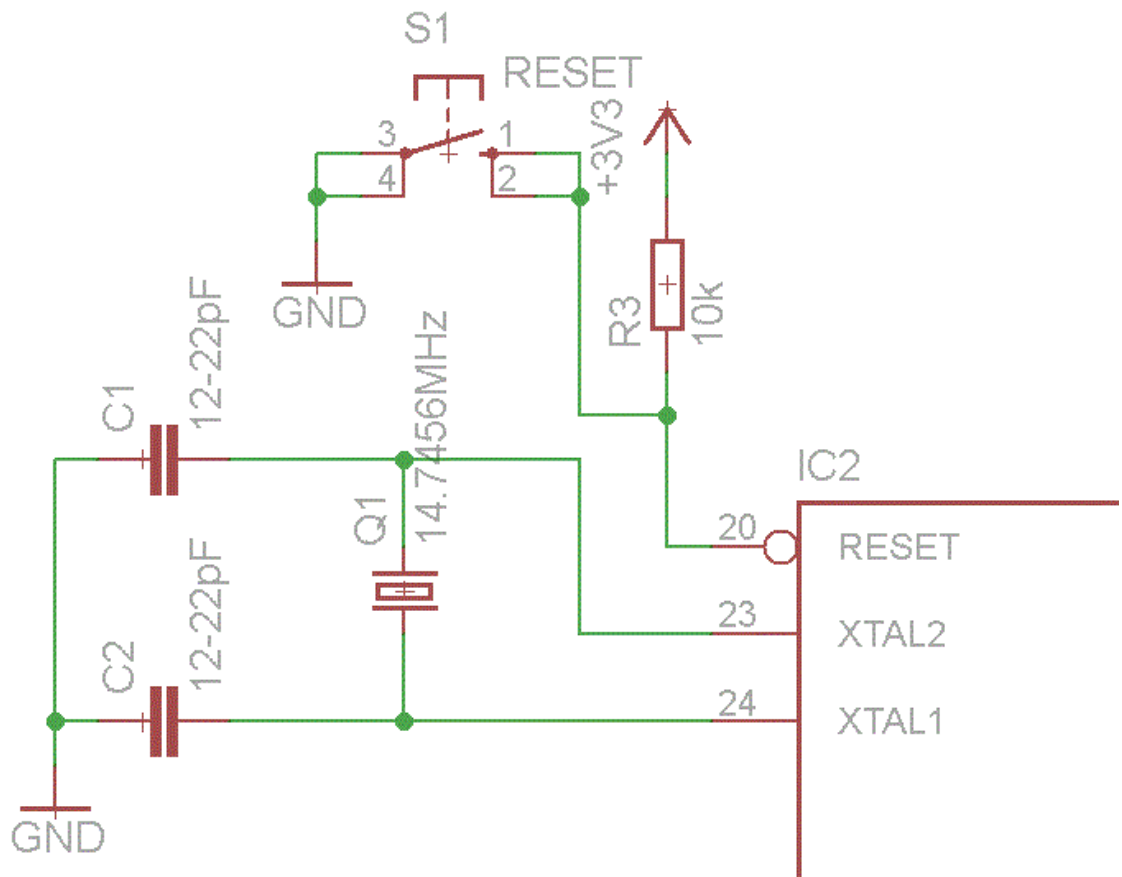


Obrázek 12: Schéma zapojení napájení pro A/D převodník [1]

3.6 Reset a krystal

Tlačítko reset je elementární záležitost, kterou je osazena většina vývojových desek. Je nutností při ladění programu, zejména pokud chceme resetovat pouze mikroprocesor a ne všechna zařízení na desce, což se dá udělat odpojením napětí.

Krystal jsme zvolili externí kvůli možnosti zvolit hodnotu 7,3728 MHz. Tuto hodnotu jsme zvolili kvůli převážnému využití desky k sériové komunikaci při rychlosti 115 200 BAUD, u níž při frekvenci 16 MHz vzniká chyba -3,5%, zatímco při frekvenci 7,3728MHz je chyba 0,0%.



Obrázek 13: Zapojení krystalu a tlačítka reset s pull-up rezistorem

4 NÁVRH DESKY PLOŠNÝCH SPOJŮ

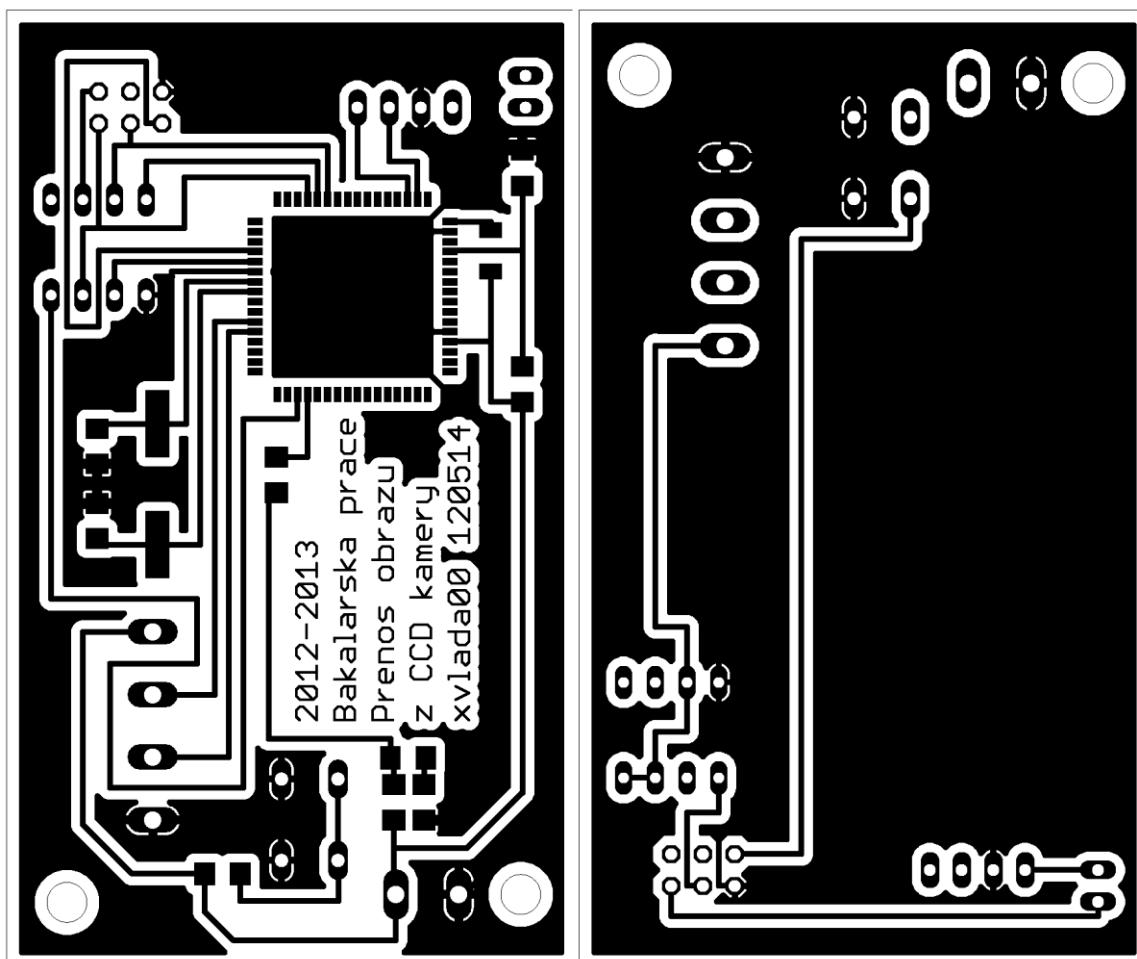
Desku plošných spojů jsme tvořili také v programu Eagle 6.3.0 který, jak jsme psali u návrhu obvodového schématu, pro velikost takto velké desky není nutno kupovat a můžeme použít Light licenci.

Při tvorbě jsme kladli důraz na malé rozměry, protože má být umístěna na helmě.

Další požadavek byla možnost zasunout bluetooth modul nebo USB2UART&BT modul bez omezení funkčnosti tlačítka reset a bez zastínění indikačních diod.

DPS se nám povedla realizovat bez použití prokovení, čímž se usnadní její výroba, je dvoustranná a je osazena jak klasickými součástkami, tak i součástkami pro povrchovou montáž.

Dokumentace návrhu desky plošných spojů je v příloze.



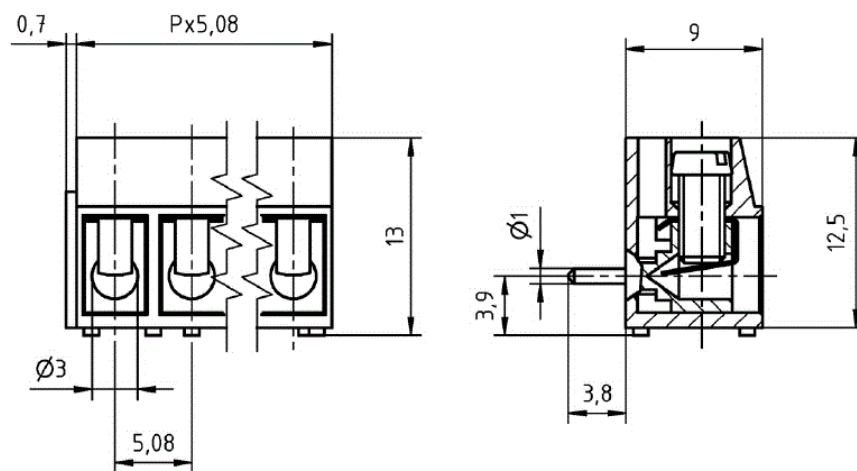
Obrázek 14: TOP (vlevo) a BOTTOM (vpravo) vrstvy desky plošných spojů

4.1 Konektory

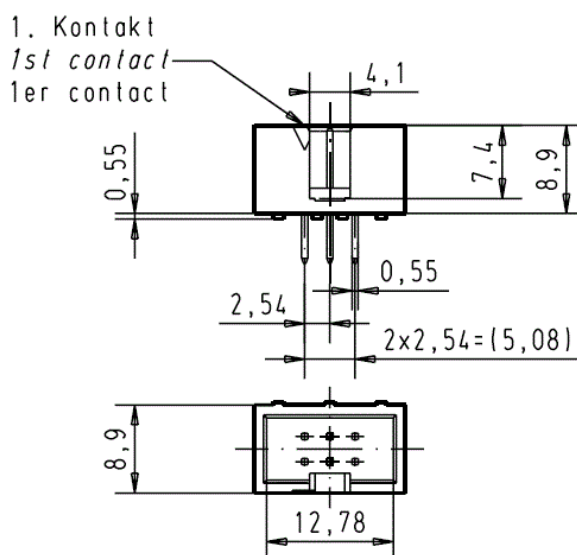
Jak je vidět ve schématu (Obrázek 7), i když ve schématu mají konektory BT a CAM velmi podobné zapojení, na desce plošných spojů se liší. Konektor BT je 4-pinová dutinková lišta s roztečí 2,54mm, do které se zasouvá BT modul, zatímco CAM konektor jsou 2 dvouslotové šroubovací svorkovnice s dvojnásobnou roztečí, tedy 5,08mm. Kamera nám byla dodána se 4 dráty, které nejsou zakončeny konektorem, a proto jsme se rozhodli použít šroubovací svorkovnice.

Tytéž svorkovnice jsme využili také pro napájecí vodiče, díky čemu se také zvyšuje ochrana před přepólováním, protože když se něco musí šroubovat, žádá si to větší pozornost než pouhé nastrčení na protikus.

Pro SPI programátor jsme použili přímý konektor 2x3 o roztečí 2,54mm s výřezem proti zasunutí špatným směrem.



Obrázek 15: Šroubovací svorkovnice



Obrázek 16: Konektor 2x3 pro programátor AVR Dragon

4.2 Rozložení součástek

Vzhledem k tomu, že na ATMEGU jde nejvíce datových vodičů z konektoru programátoru a paměti, volili jsme jejich pozici jako první, rozumně vedle sebe tak, ať nepřekáží modulu, protože kabely k programátoru i paměť, která je osazena na redukci jsou poměrně vysoké a nevešly by se pod bluetooth modul, který bude zapojen v patici vpravo dole na desce plošných spojů (Obrázek 17).

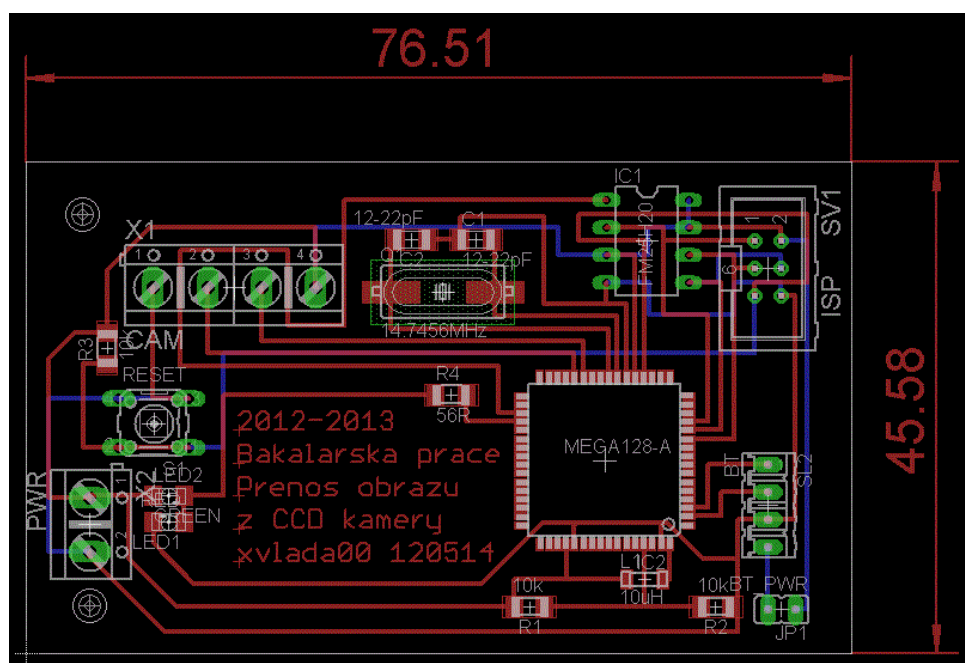
To samé se dá říct i o šroubovací patici napájení a pro připojení kamery. O dosažitelném tlačítku reset a viditelných LED diodách jsme se již zmiňovali.

Dalo by se namítnout, že součástky klasické montáže mohly být na jedné straně desky plošných spojů a bluetooth modul na straně součástek pro povrchovou montáž. To je sice pravda, ale jen pokud se použije bluetooth modul. Pro programování a ladění se ale počítá s využitím jak bluetooth modulu, tak modulu USB2UART&BT, který, jak jde vidět na obrázcích 4 a 5, má jiný pinout a je počítáno s přepájením dutinkové lišty na druhou stranu.

Jumper pro připojení napájení pro bluetooth modul je v pravém dolním rohu obvodu (Obrázek 17).

Vzhledem k malým rozměrům desky počítáme pouze se dvěma otvory pro uchycení. Samotný bluetooth modul je nevodivě chráněn smršťovací průhlednou bužirkou, takže je vhodným kandidátem pro nalepení na helmě a deska by mohla držet pomocí dutinkové lišty na jednom konci a na druhém konci by se „zaklapla“ do pinů na helmě. Případně je možné vyvrtat ještě jeden otvor mezi BT a ISP konektory.

Nesmíme zapomínat, že celý obvod bude napájen dvěma tužkovými bateriemi typu AA, případně AAA, takže konstrukčních variant bude několik a bude záležet na nás, jak se rozhodneme jednotlivé moduly rozmístit.



Obrázek 17: Návrh desky plošných spojů

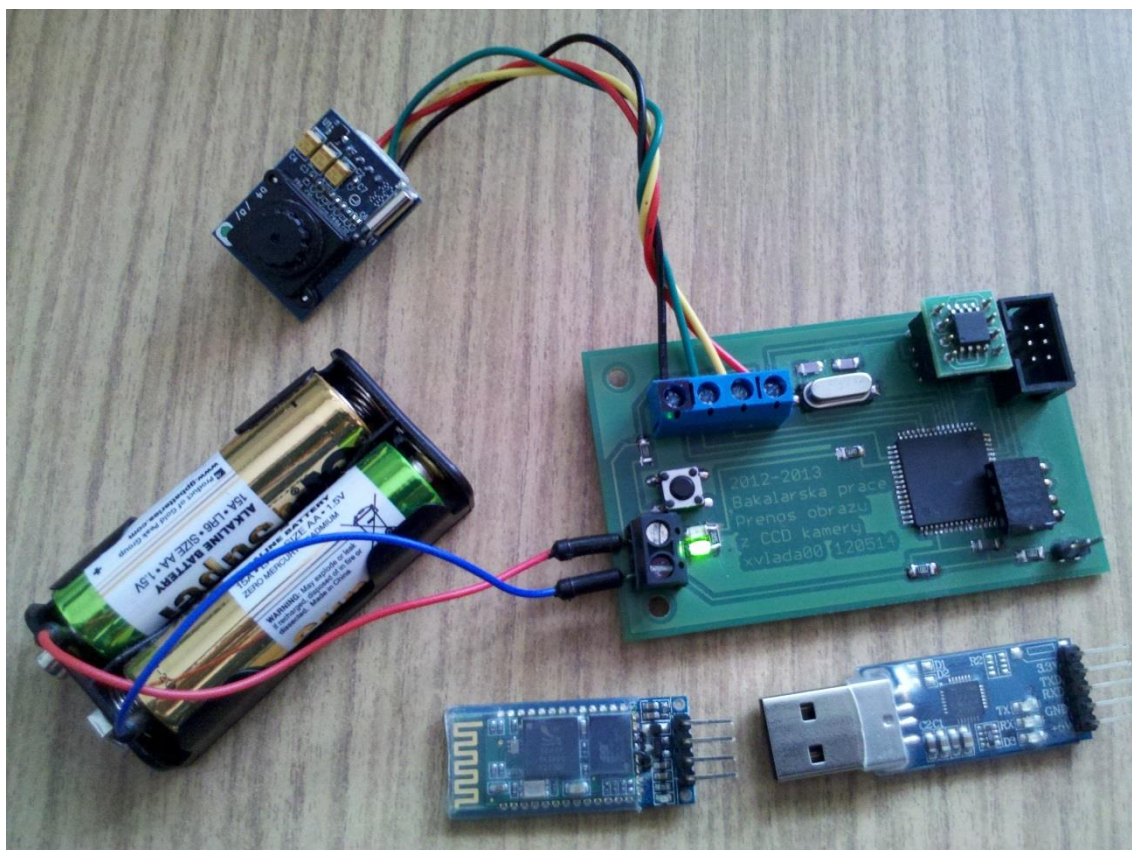
4.3 Osazení a oživení DPS

Díky jednoduchosti DPS bylo osazení a oživení triviální záležitostí. Deska se skládá z více elementárních částí, které všechny fungovaly podle našich představ ihned po osazení.

Otázkou byla pouze skutečná napěťová náročnost a fakt, že kamera podle katalogových listů pracuje s minimálním napětím 3V, což se ukázalo jako naddimenzovaná hodnota a kamera bez problémů funguje i při napájecím napětím 2,8V, které jsme v indikaci baterie označili jako hodnotu 0% a nepočítáme s možnou nižší hodnotou napětí baterií.

Na obrázku 18 vidíme kompletní zapojení systému pro snímání obrazu. Nahoře kamera, vlevo baterie. Na výběr máme z Bluetooth modulu (dole uprostřed) nebo USB2UART&BT modulu (vpravo dole), které po zasunutí do 4-pinové patice nad mikroprocesorem obstarávají komunikaci s PC.

Na DPS bakalářské práce můžeme vidět svítit zelenou LED diodu signalizující dostatečné nabití baterií.



Obrázek 18: Vzhled kompletního zapojení

5 PROGRAM MIKROPROCESORU

Pro mikroprocesor jsme psali program v jazyce C v textovém editoru Kate a kompilovali jsme jej pomocí Makefile na systému Linux Ubuntu "Precise Pangolin" 12.04 LTS. Program jsme psali v anglickém jazyce.

Samotný textový editor zná příkazy jazyka C a umí je v textu zvýraznit. Alternativou je přímo výrobcem mikroprocesorů ATMEL doporučené vývojové prostředí AVR Studio a novější Atmel Studio, které je ovšem navrženo pouze pro operační systémy Microsoft Windows, což nám pro návrh nevyhovovalo.

Tvorba Makefile je jednoduchá a pro uživatele unixových prostředí notoricky známá. O zkompilování programu se staral program `avr-gcc` a o nahrávání *.hex souboru do mikroprocesoru program `avrdude`.

V našem případě používáme 2 soubory s kódem a jeden hlavičkový soubor. Jedná se o soubory `bakalarka.c`, `FM25H20.C` a `FM25H20.h`. Dva poslední jmenované jsme pouze modifikovali, vycházeli jsme ze souborů, které nabízí výrobce paměti volně k užití a šíření.

Programy pro mikroprocesory mají pár specifických vlastností. Jednou z nich je absence parametrů programů. Je to patrné na první pohled ze vzhledu funkce `main`, která neobsahuje typické parametry `argv[]` a `argc`, které se objevují v programech psaných pro PC, které se spouštějí z konzole. Program mikroprocesoru se spouští při startu mikroprocesoru a proto je možnost spustit program jinak, tedy s jinými parametry omezena.

Další vlastností, která s předešlou vlastností úzce souvisí je používání globálních proměnných. I když se v PC programech používání globálních proměnných vyhýbáme a používáme je opravdu jen v krajních případech, zde si je můžeme dovolit, protože celý program je v podstatě degradován pouze na jakousi funkci, která na mikroprocesoru běží a nemůže být změněna, volána s jiným parametrem nebo z jiného místa.

5.1 Seznam a popis funkcí

Pro lepší orientaci v celkem rozsáhlém kódu o téměř pěti stech řádcích jsme volili členění do co možná nejvíce funkcí. Zde si popíšeme alespoň ty nejdůležitější z nich.

```
static int uart_putchar(char c, FILE *stream);
static int camera_putchar(char c);
```

Obě tyto funkce se staraly o posílání znaku přes sériovou sběrnici UART.

```
static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL,
_FDEV_SETUP_WRITE);
stdout = &mystdout;
```

Díky těmto řádkům bylo možné používat funkci printf() pro posílání dat na UART, který teď nesl funkci standardního výstupu, což zjednodušovalo práci zejména při ladění programu.

```
void adc_init(void);
void uart_init(void);
void SPI_MasterInit(void);
```

Názvy funkcí pro inicializaci periférií.

```
void cam_cmd(unsigned char A, unsigned char B, unsigned char C,
unsigned char D, unsigned char E, unsigned char F);
```

Tato funkce pouze poslala 6 Bajtů, které představovaly příkazy pro nastavení a řízení do kamery.

```
void cam_SYNC(void);
void cam_ACK(void);
void cam_BAUD(void);
void cam_RESET(void);
void cam_SNAP_uncopressed(void);
void cam_GET_snap(void);
void cam_data(void);
void cam_INIT(void);
```

Seznam příkazů pro nastavování a ovládání kamery. Hodnoty nastavení vychází z datasheetu kamery ITM-C-328.

```
void green(void);  
void red(void);
```

Obsluha indikace pomocí LED diod. Pro změnu svitu LED je potřeba maskovat PORTC, protože jeho zbylá část přísluší SPI sběrnici, na kterou je připojena FRAM paměť. Z tohoto důvodu byly napsány tyto funkce pro lepší přehlednost.

```
void capture_start(void);
```

První funkce volaná při požadavku na pořízení snímku.

```
ISR (USART0_RX_vect);  
ISR (USART1_RX_vect);
```

Obsluha přerušení vyvolaného přijetím znaku přes UART. Pro zvýšení rychlosti přenosu po sériové sběrnici jsme využívali přerušení při obdržení znaku pro obě využívané sériové linky.

Při přijetí znaku dojde k přerušení, jehož obsluha je vykonána ve funkci ISR (<název_vektoru_přerušení>).

```
void send_FRAM(void);
```

Funkce pro poslání obsahu paměti (obrázku) do PC přes UART.

```
void send_BAT(void);
```

Zjistí stav baterie, převede jak do procent, tak i do formátu pro masku baterie v rohu fotografie a posílá do PC procentuální formát.

```
void make_stamp(void);  
void write_dot_to_picture(uint32_t stamp_addr);  
void write_line_to_picture(uint32_t stamp_addr);
```

Funkce make_stamp tiskne obrázek baterie do levého horního rohu pořízeného obrázku. Kreslení probíhá tak, že procentuální formát stavu baterie převede dělením 10 do formátu, který udává, jak moc bude vyplněn obrázek baterie a to v deseti krocích. Samozřejmě je, že nebudeme psát zvlášť funkci pro grayscale a RGB obrázek. Funkce je univerzální a funguje na oba typy obrazu, jako parametr si bere hodnotu colour, která je globální proměnnou.


```

void write_dot_to_picture(uint32_t stamp_addr);
void write_line_to_picture(uint32_t stamp_addr);
jsou podfunkcemi funkce
void make_stamp(void);

int main(void)

```

Samotný main() neobsahuje téměř nic, protože celý program je řízen příkazy (jedno-bajtový), posílanými přes UART do mikroprocesoru. Každý takto poslaný příkaz donutí UART vyvolat přerušení v jehož obsluze se rozhoduje o vývoji programu. Parsování příkazů je také velmi užitečné při ladění programu.

Command	ID Number	Parameter1	Parameter2	Parameter3	Parameter4
Initial	AA01h	00h	Color Type	Preview Resolution	JPEG Resolution
Get Picture	AA04h	Picture Type	00h	00h	00h
Snapshot	AA05h	Snapshot Type	Skip Frame Low Byte	Skip Frame High Byte	00h
Set Package Size	AA06h	08h	Package Size Low Byte	Package Size High Byte	00h
Set Baudrate	AA07h	1st Divider	2nd Divider	00h	00h
Reset	AA08h	Reset Type	00h	00h	xxh*
Power Off	AA09h	00h	00h	00h	00h
Data	AA0Ah	Data Type	Length Byte 0	Length Byte 1	Length Byte 2
SYNC	AA0Dh	00h	00h	00h	00h
ACK	AA0Eh	Command ID	ACK counter	00h / Package ID Byte 0	00h / Package ID Byte 1
NAK	AA0Fh	00h	NAK counter	Error Number	00h

Tabulka 2: Příkazy pro nastavení kamery [2]

5.2 Parsované příkazy mikroprocesoru

Mikroprocesor obsahuje program, který je schopen kameru synchronizovat, pořizovat z ní obraz, nastavovat parametry pro pořízení obrazu, získávat hodnotu napětí baterií a tyto informace posílat do PC, které si o ně ale musí říci. Proto jsme naparsovali 4 příkazy, pomocí kterých může mikroprocesor reagovat.

5.2.1 Příkaz „A“

Pokud mikroprocesor obdrží tento příkaz, provede synchronizaci s kamerou, nastaví rozlišení obrazu a barevnou hloubku na 320x240 a 8-bit grayscale a vyšle příkaz pro pořízení obrazu. Po sejmutí obrazu se obraz z kamery po UART sběrnici přenesení do FRAM paměti. Poté se volá zjištění stavu baterie, což je aproximovaná a normalizovaná hodnota A/D převodníku napětí na baterii. Tato hodnota se poté tiskne do obrazu jako bílý symbol baterie v levém horním rohu obrazu. Výsledný obraz s maskou se poté pošle do PC. K tomu používá šesti-bajtové příkazy popsané v tabulce Tabulka 2.

5.2.2 Příkaz „B“

Při obdržení znaku „B“ na UART0 mikroprocesor podobně jako u znaku „A“ provede synchronizaci s kamerou a nastaví rozlišení a barevnou hloubku obrazu pomocí inicializace kamery. V tomto případě bude požadován RGB565 obraz a rozlišení 320x240. Podobně jako u příkazu Příkaz „A“ i tady poté nahrajeme obraz do paměti FRAM, aplikujeme masku baterie podle jejího stavu získaného z A/D převodníku a obraz pošleme do PC. Příkazy jsou v tabulce Tabulka 2

5.2.3 Příkaz „C“

Tento příkaz spustí měření napětí na baterii, jehož výsledek dostane aproximací padesáti hodnot z A/D převodníku. Aproximace je prostý aritmetický průměr, který je postačujícím a při opakovaném měření napětí baterie se hodnoty liší řádově v jednotkách procent, což pro indikaci orientačního stavu baterie bohatě stačí. Nakonec tiskneme na standardní výstup, který v našem případě představuje UART1, tedy sériová linka komunikující s PC.

Naměřené hodnoty napětí je však nutno normalizovat do tvaru, který odpovídá procentuálnímu stavu a je tedy 0% - 100%. Tento přepoččet byl založen na předpokládaném nejnižším a nejvyšším napájecím napětí. Tento rozsah jsme si zvolili 2,8V - 3,3V. Napětí ze zdroje bylo na A/D převodník připojeno přes napěťový dělič s poměrem R1:R2 = 1:1, z čehož vyplývá poloviční výstupní napětí, tedy 1,4V – 1,65V. To při 10-ti bitovém rozlišení A/D převodníku a referenčním napětí 2,56V dává maximální a minimální hodnotu A/D převodníku.

$$ADC_{MAX} = \frac{1024}{2,56} 1,4 = 560$$

$$ADC_{MIN} = \frac{1024}{2,56} 1,65 = 660$$

V takovém případě stačí pouze odečíst 560 a dostaneme výsledek 0-100, což přímo odpovídá procentům.

V praxi jsme však museli rovnici trochu zkomplikovat, zejména kvůli nepřesnému děliči napětí.

Již experimentálně jsme tedy zjistili vylepšenou přepoččetní rovnici tvaru

$$Battery(\%) = \frac{4}{5} ADC - 450$$

Rovnice se nám kvůli přesnosti lehce zkomplikovala. Teď nám vychází minimální a maximální hodnoty tak, aby dávaly smysl. Při napájení tvrdým zdrojem na 3,3V se pohybujeme kolem 90% a na baterie 2x1,5V kolem 40%, což odpovídá nefunkčnosti zařízení u napětí kolem 2,8V. I když se snažíme vyhnout používání slov typu „kolem“, jedná se o orientační indikaci a budeme ji potřebovat zejména pro zakreslení indikace baterie do obrazu, která bude mít pouze 10 hodnot, a proto je pro nás tato přesnost dostačující.

5.2.4 Příkaz „D“

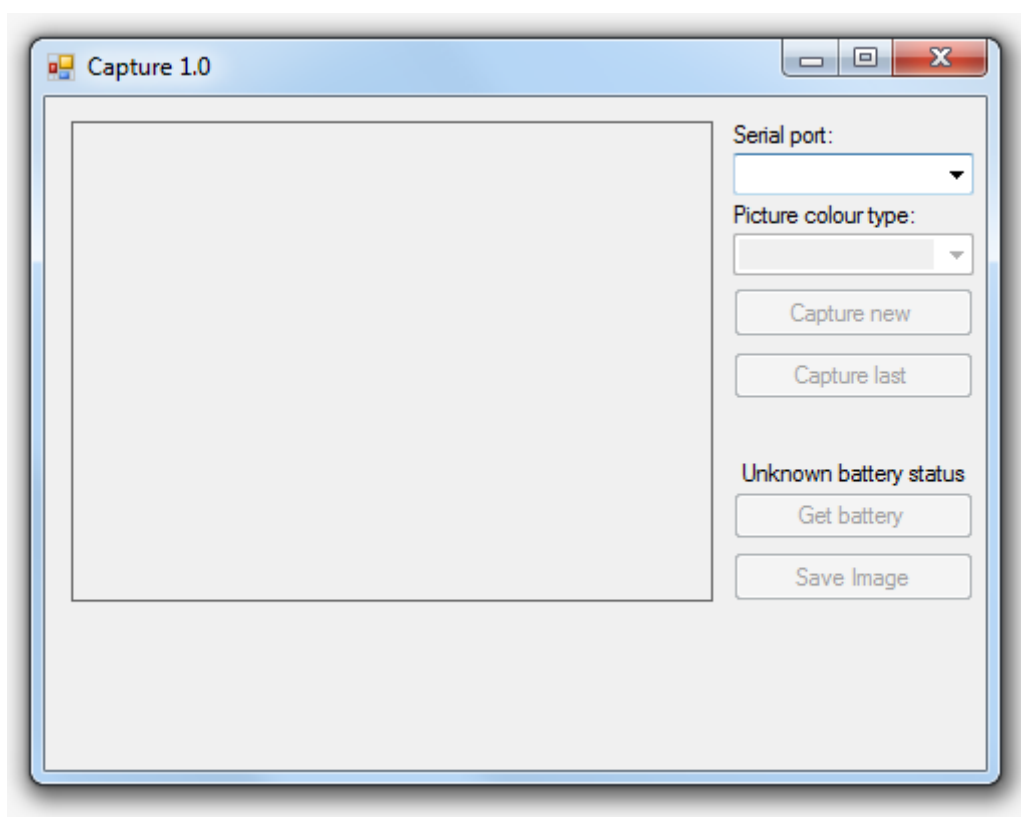
Tímto příkazem pouze pošleme obraz, který je již v paměti FRAM uložen do PC přes UART1. Jedná se o posláni posledního pořizého obrazu s indikací baterie z doby, kdy byl obraz pořizen. Tato funkce slouží k vyvolání poslední fotografie z paměti a to i v případě, že mezi pořizemím a vyvoláním snímku z paměti byl mikroprocesor i s kamerou odstaven od napájení, takže zařízení plní funkci fotoaparátu s pamětí na jeden (poslední) snímek.

6 PC PROGRAM

Pořízení obrazu jsme kontrolovali programem Matlab, který pro jeho jednoduchou práci s maticemi a tedy i s obrazem byl nejjednodušší a nejrychlejší možností. Samotný program s grafickým uživatelským rozhraním byl však realizován ve vývojovém prostředí Microsoft Visual Studio 2010 v jazyce C#. Samotný program obsahuje pouze pár tlačítek a provedení je co nejjednodušší, aby bylo uživatelsky co nejvíce přívětivé. Samotné Visual Studio obsahuje přímo metody pro práci se sériovým rozhraním, což výrazně ulehčuje práci programátora.

Snad jediným problémem, na který jsme během programování narazili, byl přepočít barevného obrazu z tvaru RGB565 na tvar, kde každé barvě přísluší 8bitů, tedy RGB24.

6.1 Grafické uživatelské prostředí Capture

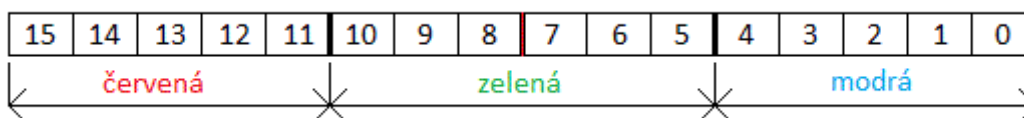


Obrázek 19: Úvodní okno programu Capture 1.0

6.1.1 Picture box

Tento prvek má stabilně nastavenou velikost okna na 320x240 pixelů, protože program s jiným rozlišením pracovat nebude. Obraz se do něj nahrává po převedení obrazu z formátu RGB565 do formátu RGB24, protože přímo RGB565 zobrazit neumí. Přepočítání na formát RGB24 probíhalo tak, že jsme si separovali vždy 2 bajty z obrazu, které představují RGB565 informaci (16 bitů) a ty rozdělili na jednotlivé složky červené, zelené a modré barvy a následovně provedli normalizaci pomocí roztažení 5-ti a 6-ti bitových hodnot na 8-mi bitové ekvivalenty.

```
colR = col / 2048;  
colG = (col % 2048) / 32;  
colB = col % 32;  
  
colR = colR * 255 / 31;  
colG = colG * 255 / 63;  
colB = colB * 255 / 31;
```

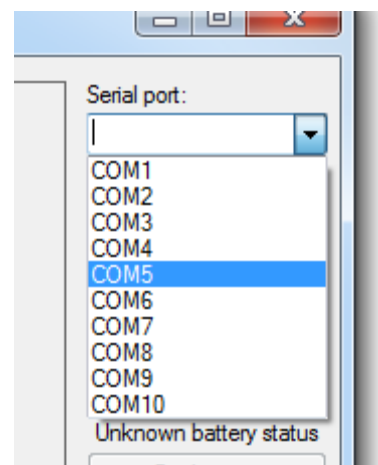


Obrázek 20: Skladba 16-bit RGB (RGB565)

Takto upravené hodnoty jsme již použili jako matice pro toolbox picture box.

6.1.2 UART port

Program obsahuje 2 combo boxy, z nichž jeden slouží pro výběr barevné hloubky obrázku a druhý k výběru portu, na kterém chceme otevřít sériovou linku. Vzhledem k tomu, že se nacházíme v systému Microsoft Windows, jedná se o porty COM1 až COM10. Záleží na nastavení v systému, který port nám přidělí pro náš USB2UART&BT převodník nebo BT modul. V našem případě to byl port COM5.



Obrázek 21: Výběr sériového portu

6.1.3 Colour type

Další combo box slouží k výběru barevné hloubky námi požadovaného obrazu. Tímto vybíráme, jestli mikroprocesoru pošleme příkaz „A“ nebo „B“ jak je popsáno v kapitole 5.2.

6.1.4 Capture new

Tímto tlačítkem, které je použitelné pouze pokud je otevřen sériový port a je zvolena barevná hloubka obrazu, který se má pořídit, se pošle znak „A“ nebo „B“ do mikroprocesoru. Pořídí se nový obraz, a pokud v paměti FRAM již nějaký obraz byl, přemaže se.

6.1.5 Capture last

Tato volba slouží k vyvolání posledního uloženého obrazu v paměti FRAM bez ohledu na to, kdy byl pořízen, a to i v případě, že se tak stalo před odpojením napájení mikroprocesoru a paměti FRAM. Mikroprocesor pro tuto operaci obdrží od PC příkaz „D“.

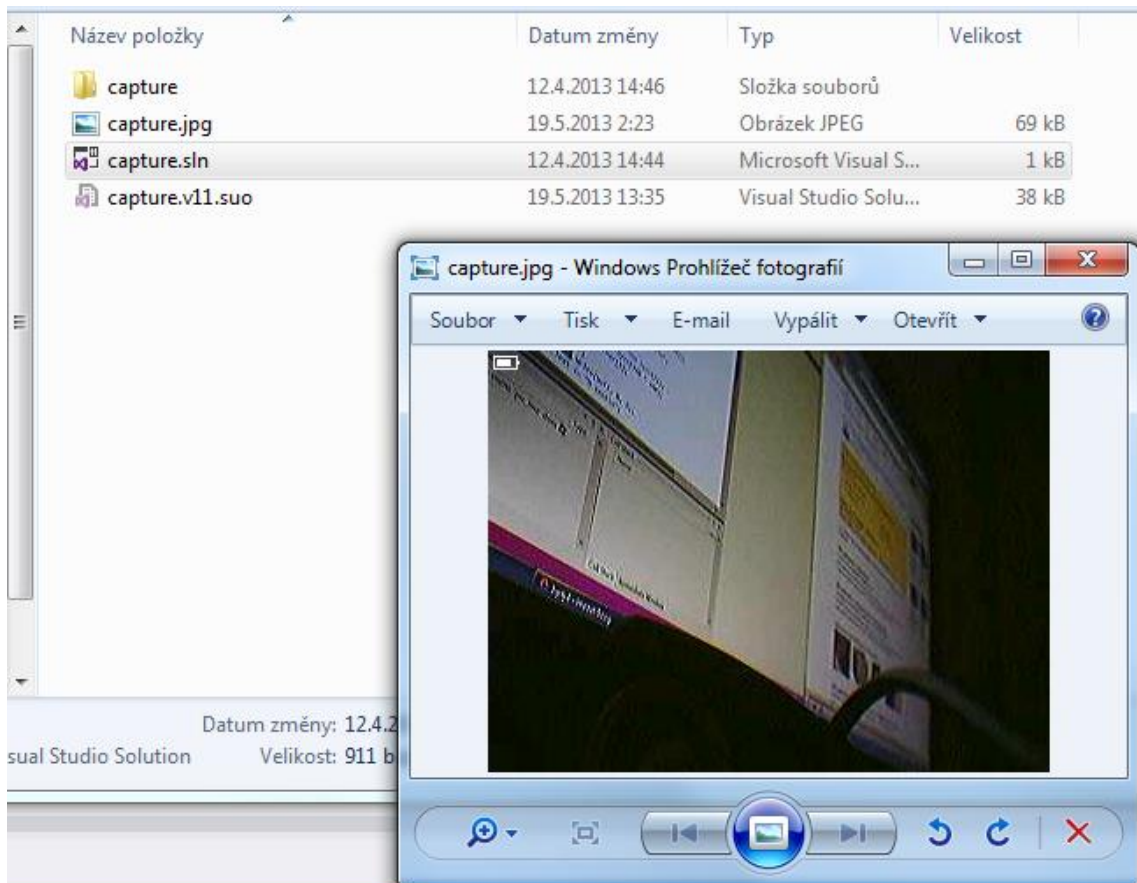
6.1.6 Battery status

Tlačítkem „Get battery“ se do mikroprocesoru pošle příkaz „C“ a mikroprocesor pošle zpátky procentuální stav baterie, který spočítá aproximací aritmetickým průměrem. Tato funkce se volá také automaticky při otevření sériového portu, tedy při navázání komunikace s mikroprocesorem.

Získávání procentuálního rozsahu nabití baterie je popsán v kapitole 5.2.3.

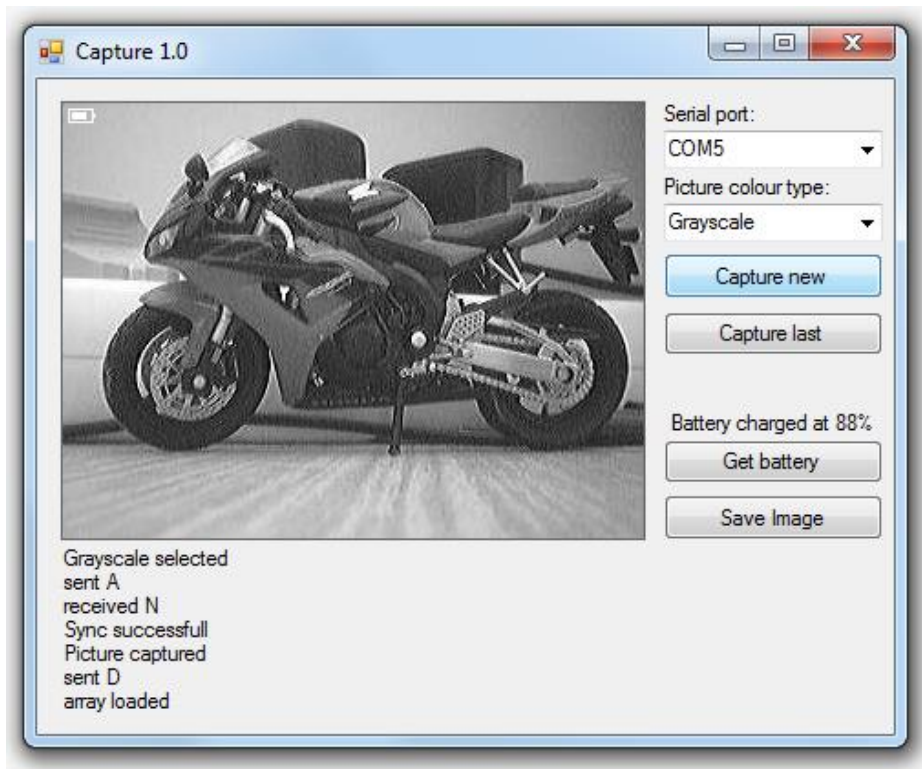
6.1.7 Save image

Tlačítko „Save Image“ slouží k jednoduchému uložení souboru na místo na disku, ze kterého je program spuštěn.

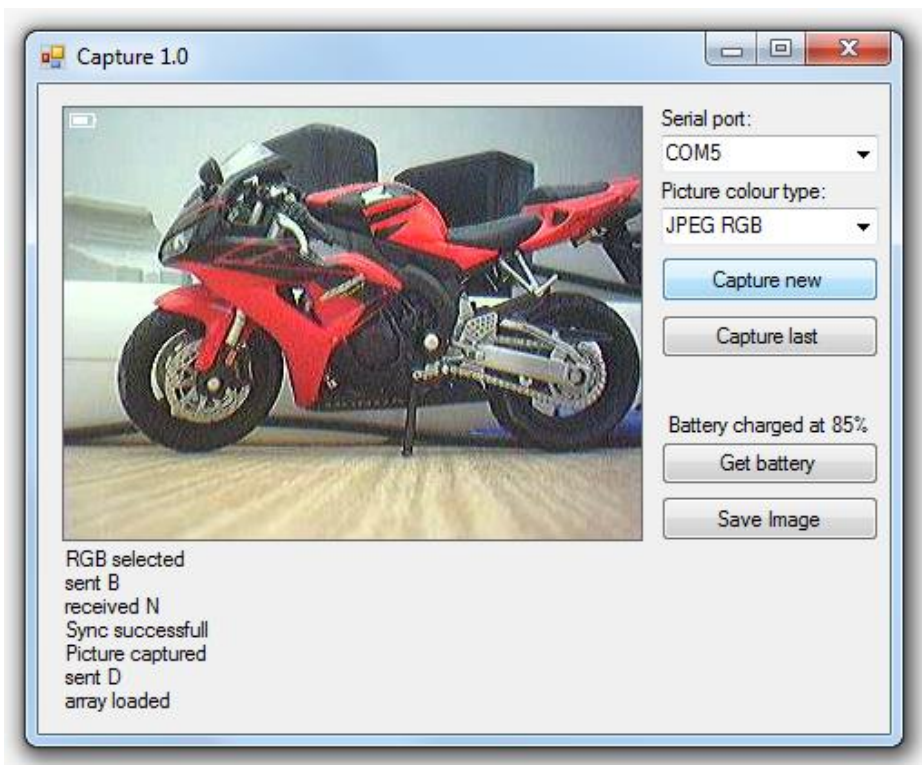


Obrázek 22: Náhled na místo uložení souboru a náhled samotného souboru

6.2 Pořízení obrazu



Obrázek 23: Pořízení obrazu v odstínech šedi



Obrázek 24: Pořízení barevného obrazu (16-bit RGB)

Celým cyklem pořízení obrazu z kamery nás provází log, do kterého jsou vpisovány základní události, které se na sběrnici udály.

Protože jsme psali program pro mikroprocesor v anglickém jazyce, také program v C# má ovládací prvky a zdrojový kód v angličtině.

Vzhledem k tomu, že kamera podporuje JPEG kompresi a my ji k pořízení a přenesení obrazu nepoužíváme i při tak vysokém rozlišení, musíme počítat s velkou časovou náročností přenosu obrazu.

U snímku v odstínech šedi se pohybujeme kolem 7-8 vteřin pro přenos snímku z kamery do paměti FRAM a dalších 7-8 vteřin pro poslání snímku z paměti do PC. To je dohromady 15 vteřin pro pořízení nového snímku a 7-8 vteřin pro získání snímku staršího.

16-bitový RGB obraz je právě 2x datově náročnější a tedy i časy jsou dvojnásobné, tedy 30 vteřin pro nový snímek a 15 vteřin pro pořízení snímku staršího.

7 ZÁVĚR

Realizace obvodového schématu byla bezproblémová, protože jsme ji rozdělili na více elementárních obvodů, které jsou odděleny a spojuje je pouze mikroprocesor, tedy až jeho softwarové vybavení.

Tvorba desky plošných spojů byla komplikovanější, kvůli zajištění místa pro bluetooth modul a jeho rozumné velikosti. U schématu samozřejmě otázku velikosti řešit nemusíme.

Programové vybavení byla nejkomplicovanější část práce. Jeho řešení a odladění bylo zejména časově náročné. Vytvořili jsme program, který je schopen automaticky pracovat s kamerou a zpracovávat obraz z ní pořízen.

Pro PC obsluhu jsme vytvořili jen jednoduchou aplikaci, díky které je možné komunikovat s kamerou, získávat její data a ty zobrazit.

Řešení pomocí bluetooth je docela nešťastné, kvůli jeho nízkému dosahu, zejména v budovách, kde se bude záchranář pohybovat. Jako vylepšení do budoucna by se dal použít například modul BeeWi, ale jak jsme psali již v úvodu, projekt se v praxi nebude používat, takže by to jen zvýšilo náklady na realizaci bakalářské práce.

Dalším vylepšení by mohlo být použití akumulátoru namísto klasických baterií. V našem případě nejdou použít nabíjecí baterie pro jejich nízké napětí 1,2V oproti alkalickým bateriím 1,5V a použít 3 kusy baterií zase zvyšuje hmotnost a rozměry. Například Li-ion akumulátor 3,7V používaný v modelářství pro létající stroje s napětovým stabilizátorem (ideálně zenerova dioda pro low-drop) by byl ideálním řešením.

Funkčnost zařízení by se dala výrazně vylepšit použitím infrakamery. Tohle vylepšení by bylo sice zdaleka nejdražší položkou, ale posunulo by to zařízení na mnohem vyšší úroveň.

Celou koncepci ale nejvíce shazuje rychlost přenosu kvůli použití pomalé sériové sběrnice. Rychlost 115200BAUD není dostačující pro přenos tak velkého objemu dat v rozumném časovém intervalu. Časy přenosů jsou rozebrány v kapitole 6.2. Koncepce by se dala vylepšit použitím kamery, která podporuje větší rychlost na sériové sběrnici nebo přímo vyměnit sériovou sběrnici za sběrnici rychlejší.

LITERATURA

- [1] ATMEL: 8-bit Atmel Microcontroller with 128KBytes In-System Programmable Flash ATmega128, ATmega128L datasheet [online], Revize 2467X–AVR–06/11. 386s. [cit. 2008-12-22].
Dostupné na URL: <http://www.atmel.com/Images/doc2467.pdf>

- [2] Intertec Components: ITM-C-328 Manual [online]. 12s. [cit. 2008-12-22].
Dostupné na URL: http://www.intertec-components.de/common/db/pdf/visuals/ITM-C-328_Manual.pdf

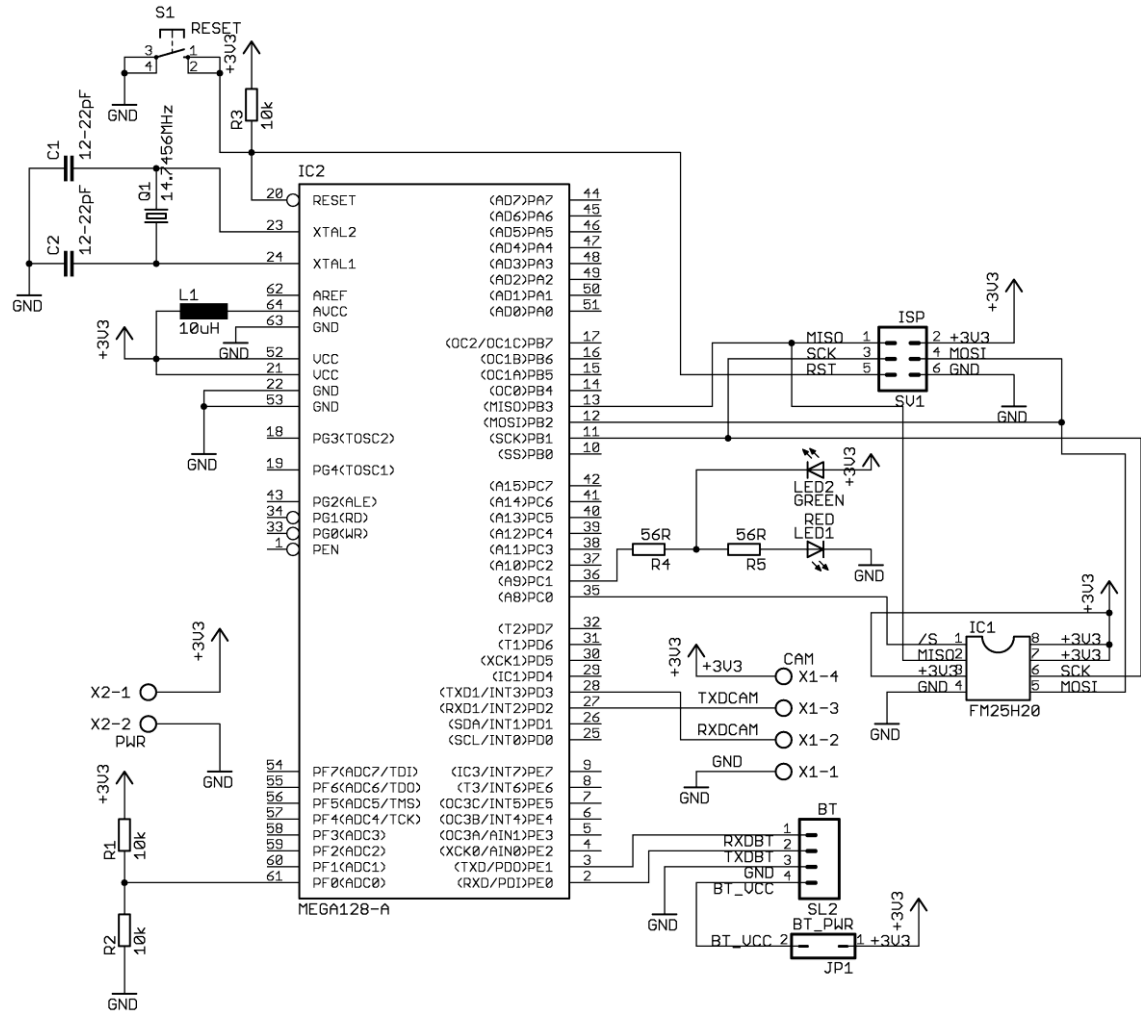
- [3] RAMTRON: FM25H20 - 2Mb SPI FRAM datasheet [online]. Revize 2.2: září 2010. 15s. [cit. 2008-12-26].
Dostupné na URL: http://www.ramtron.com/files/datasheets/FM25H20_ds.pdf

- [4] CadSoft: EAGLE Easily Applicable Graphical Layout Editor Manual, 2nd Edition [online]. Version: 4.1: Copyright 2004. 248s [cit. 2012-12-26].
Dostupné na URL: <http://www.cadsoft.de/wp-content/uploads/2011/05/manual-eng.pdf>

SEZNAM PŘÍLOH

- Příloha 1: Schéma zapojení DPS
- Příloha 2: DPS TOP vrstva (1:1)
- Příloha 3: DPS BOTTOM vrstva (1:1)
- Příloha 4: Osazovací výkres DPS (1:1)
- Příloha 5: Seznam materiálu pro DPS

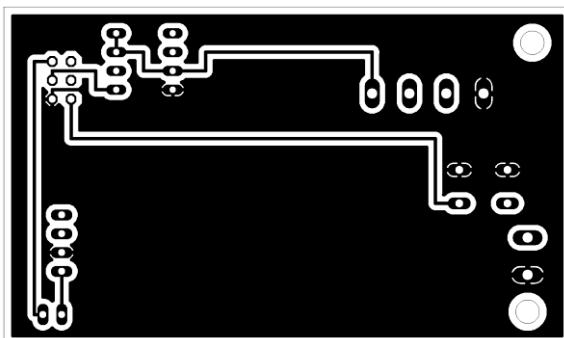
Příloha 1:



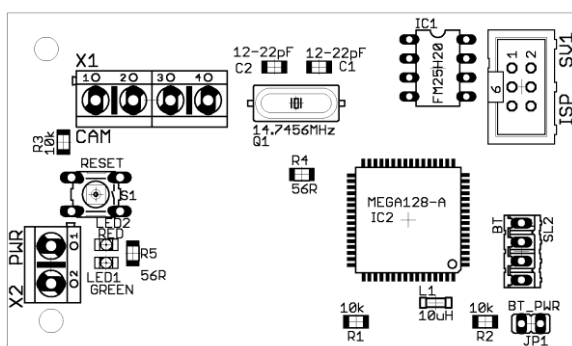
Příloha 2:



Příloha 3:



Příloha 4:



Příloha 5:

partlist exported from C:/Users/wlady/Documents/eagle/bakalarka/bakalarka.sch at 24.5.2013 20:08:41

Part	Value	Device	Package	Description	MF MPN	OC_FARNELL	OC_NEWARK
C1	12-22pF	C-EUC1206	C1206	CAPACITOR, European symbol]			
C2	12-22pF	C-EUC1206	C1206	CAPACITOR, European symbol			
IC1	FM25H20	DIL8	DIL08	Dual In Line / Socket			
IC2	MEGA128-A	MEGA128-A	TQFP64	MICROCONTROLLER	ATMEGA128-16AU	9171118	95K7364
JP1	BT_PWR	JPIQ	JPI	JUMPER		unknown	unknown
L1	10uH	SM-1206	SM-1206	INDUCTOR			
LED1	RED	LEDSMT1206	1206	LED			
LED2	GREEN	LEDSMT1206	1206	LED			
Q1	14.7456MHZ	CRYSTALHC49UP	HC49UP	CRYSTAL		unknown	unknown
R1	10k	R-EU_R1206	R1206	RESISTOR, European symbol]			
R2	10k	R-EU_R1206	R1206	RESISTOR, European symbol			
R3	10k	R-EU_R1206	R1206	RESISTOR, European symbol]			
R4	56R	R-EU_R1206	R1206	RESISTOR, European symbol]			
R5	56R	R-EU_R1206	R1206	RESISTOR, European symbol]			
S1	RESET	10-XX	B3F-10XX	OMRON SWITCH	B3F-1000	176432	36M3542
SL2	BT	M04	04P	AMP QUICK CONNECTOR		unknown	unknown
SV1	ISP	ML6	ML6	HARTING		unknown	unknown
X1	CAM	AK500/4	AK500/4	CONNECTOR		unknown	unknown
X2	PWR	AK500/2	AK500/2	CONNECTOR		unknown	unknown