České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídící techniky

# BAKALÁŘSKÁ PRÁCE

## Palubní počítač pro motocykl

Vypracoval: Zdeněk Táborský
Vedoucí práce: Ing. Pavel Němeček

Praha 2006

# Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne …….  …….
podpis

# Poděkování

Rád bych poděkoval všem, kteří mi jakýmkoli způsobem pomáhali při vzniku této bakalářské práce. Děkuji především vedoucímu mé práce Ing. Pavlu Němečkovi za trpělivé vedení a cenné rady během celé realizace. Dále děkuji rodičům za trpělivost a podporu.

# Anotace

Tématem této bakalářské práce je návrh a realizace zařízení pro měření a interpretaci fyzikálních veličin na spalovacím motoru. Konkrétní provedení má podobu palubního počítače pro motocykl. Práce se zabývá nejen hardwarovým řešením, ale i návrhem základního softwaru.

Celé zařízení je rozděleno do dvou částí, z nichž každá je řízena vlastním mikroprocesorem. Jedna část slouží ke zprostředkování styku s okolím. Obsahuje veškeré vstupní i výstupní obvody a prvky pro realizaci požadovaných měření a výpočtů. Druhá část umožňuje získané informace interpretovat pomocí zobrazovacích prvků. Předávání dat mezi oběma částmi probíhá pomocí meziprocesorové komunikace.

# Annotation

The subject of this bachelor work is a design and a realization of arrangement for measurement and interpretation of physical values on combustion engine. Concrete rendering is a board computer for motorcycle. The work deals not only with a resolution of hardware but also with a proposal of fundamental software.

The whole arrangement consists of two parts. Both of them are managed by its own microprocessor. The first part serves for intervention of contact with the environment. It includes all input and output circuits and elements for realization of measurements and calculations. The second part enables to interpret acquired information using imaging elements. Data are transfered along a bus between both microprocessors.

# Obsah

**1 Úvod ... 1**
1.1 Požadavky na řešení ... 2
**2 Návrh hardwaru ... 3**
2.1 Kritéria a podmínky návrhu ... 3
2.2 Hardwarové funkční bloky ... 4
2.2.1 Mikroprocesor AT90CAN128 ... 4
2.2.2 Mikroprocesor AVR ATmega16 ... 8
2.2.3 Napájecí zdroj ... 9
2.2.4 Digitální vstupy ... 10
2.2.5 Analogové vstupy ... 11
2.2.6 Obvody výstupů ... 11
2.2.7 Externí paměť ... 12
2.2.8 Sériové rozhraní RS232 ... 13
2.2.9 Budič sběrnice CAN ... 14
2.2.10 Obvod reálného času ... 15
2.2.11 Display ... 16
2.2.12 Krokový motor a jeho budič ... 18
2.2.13 Stupnice z LED diod ... 21
2.2.14 Ostatní hardwarové části ... 22
2.3 Konstrukční řešení ... 23
2.3.1 Plošné spoje ... 23
2.3.2 Pouzdra součástek ... 23
2.3.3 Mechanické uspořádání a konektory ... 24
**3 Volba vhodných senzorů ... 26**
3.1 Čidlo tlaku ... 26
3.2 Teplotní čidlo ... 27
3.3 Snímání zubů na klikovém hřídeli ... 27
3.4 Senzor rychlosti ... 27
**4 Softwarové řešení ... 29**
4.1 Programovací jazyk na vyšší úrovni ... 29
4.2 Kompilátor GCC v balíku WinAVR ... 30
4.3 Software pro horní desku ... 31
4.3.1 Struktura softwaru ... 31
4.3.2 Knihovna pro display ... 33
4.3.3 Knihovna pro LED diody ... 34

4.3.4 Knihovna pro krokový motor ..... 37
4.3.5 Ostatní knihovny ..... 37
4.3.6 Hlavní funkce Main() ..... 38
4.4 Software pro spodní desku ..... 38
4.4.1 Popis demonstračního softwaru ..... 38
4.4.3 Použité knihovny ..... 40
**5 Závěr ..... 41**

**Seznam použitých zdrojů ..... 42**
**Přílohy ..... 43**
A Příloha – Schéma zapojení ..... 43
B Příloha – Desky plošných spojů ..... 45
C Příloha – Rozmístění součástek ..... 46
D Příloha – Seznam použitých součástek ..... 47
E Příloha – Obsah přiloženého CD ..... 49

# Seznam obrázků

Obr. 2.1: Blokové schéma hardwaru palubního počítače ..... 4
Obr. 2.2: AT90CAN128 - 64 TQFP - Rozmístění vývodů ..... 4
Obr. 2.3: Struktura ALU mikroprocesoru AT90CAN128 ..... 5
Obr. 2.4: Možné propojení jednotlivých zařízení na sběrnici $I^2C$ ..... 7
Obr. 2.5: Průběh logických úrovní na vodičích SDA a SCL ..... 8
Obr. 2.6: ATmega16 - 44 TQFP - Rozmístění vývodů ..... 8
Obr. 2.7: Zapojení napájecího zdroje ..... 10
Obr. 2.8: Zapojení jednoho digitálního vstupu ..... 10
Obr. 2.9: Zapojení jednoho analogového vstupu ..... 11
Obr. 2.10: Zapojení univerzálního a výkonového výstupu ..... 12
Obr. 2.11: DataFlash AT45DB041B ..... 12
Obr. 2.12: Datový rámec sériového přenosu RS232 ..... 14
Obr. 2.13: Zapojení obvodu MAX232 ..... 14
Obr. 2.14: Zapojení budiče sběrnice CAN ..... 15
Obr. 2.15: Zapojení obvodu reálného času ..... 16
Obr. 2.16: Rozměry a rozmístění vývodů displeje HCMS-2975 ..... 17
Obr. 2.17: Znázornění posuvného registru pro jeden znak displeje HCMS-2975 ..... 18
Obr. 2.18: Vnitřní struktura obvodu L6208 ..... 19
Obr. 2.19: Zapojení obvodu L6208 ..... 20
Obr. 2.20: Zapojení LED diodové stupnice ..... 22
Obr. 2.21: Podoba hotového zařízení (bez krokového motoru) ..... 25
Obr. 3.1: Čidlo tlaku MPX4115AP ..... 26
Obr. 3.2: Teplotní čidlo LM135 ..... 27
Obr. 3.3: 1GT101DC - senzor snímání zubů na klikovém hřídeli ..... 27
Obr. 4.1: Struktura softwaru pro horní desku ..... 32
Obr. 4.2: Struktura řídícího registru ..... 33
Obr. 4.3: Princip míchání barev u RGB LED diod ..... 35
Obr. A.1: Schéma zapojení horní desky ..... 43
Obr. A.2: Schéma zapojení spodní desky ..... 44
Obr. B.1: Horní deska plošných spojů – obě strany ..... 45
Obr. B.2: Spodní deska plošných spojů – obě strany ..... 45
Obr. C.1: Rozmístění součástek na horní desce plošných spojů – obě strany ..... 46
Obr. C.2: Rozmístění součástek na spodní desce plošných spojů – obě strany ..... 46

# Seznam tabulek

Tab. 2.1: Přehled významných použitých obvodů a jejich pouzder ..... 24
Tab. 4.1: Ukázka možného zaplnění pole *CLED*..... 35
Tab. E.1: Seznam součástek použitých na horní desce..... 47
Tab. E.2: Seznam součástek použitých na spodní desce..... 48

# 1 Úvod

Cílem této bakalářské práce bylo vytvořit univerzální zařízení, které bude umět snímat požadované veličiny na spalovacím motoru. Zařízení by mělo disponovat dostatečným počtem univerzálních analogových a digitálních vstupů, vhodnými a dostačujícími možnosti jak naměřené veličiny zobrazovat a několika výstupy pro případné ovládání externích zařízení. Samotné zobrazování naměřených hodnot by však pro účely důkladné diagnostiky nepostačovalo. Nedílnou součástí by tedy měla být dostatečná paměť pro uchování naměřených hodnot a možnost komunikovat s okolím přes vhodné rozhraní. Kvůli charakteru prostředí, v kterém bude pracovat, je potřeba alespoň částečné odolnosti vůči rušivým vlivům a mechanickému namáhání např. vibracím.

Vzhledem k tomu, že v samotném zadání bakalářské práce nebyla podoba zařízení blíže specifikována, měl jsem možnost navrhnout konkrétní řešení podle vlastního uvážení. Rozhodl jsem se, realizovat ho jako palubní počítač pro motocykl. V tomto případě jsem měl pevně stanovené maximální rozměry a rozložení některý zobrazovacích prvků. Funkčně však bylo zařízení navrženo pro použití s jakýmkoli spalovacím motorem, důraz byl kladen na univerzálnost.

Práce je tématicky členěna do tří hlavních částí. První část je věnována návrhu hardwaru a konstrukčnímu řešení. V druhé části jsou uvedeny vhodné senzory, které mohou se zařízením spolupracovat. Poslední část popisuje implementovaný software, který slouží k demonstraci funkčnosti základních bloků.

## 1.1 Požadavky na řešení

V této části jsou uvedeny konkrétní požadavky na celé zařízení. Při stanovování těchto kritérií bylo uvažováno o veškerých potřebách, které mohou při měření na spalovacím motoru nastat.

**Vstupy a výstupy:**

- 10 digitálních vstupů
- 6 analogových vstupů
- 1 analogový diferenční vstup
- 2 universální výstupy
- 1 výkonový výstup

**Možnosti zobrazování:**

- zobrazování hodnot pomocí stupnice z LED diod
- zobrazování hodnot na displeji
- zobrazování hodnot pomocí ručičky a krokového motoru

**Další funkce:**

- možnost komunikace přes RS232
- možnost komunikace po sběrnici CAN
- flash paměť pro případné uchovávání naměřených hodnot
- možnost využívat informace o reálném čase

# 2 Návrh hardwaru

Tato kapitola se zabývá návrhem a popisem realizace hardwaru palubního počítače. V první části je uvedeno podle jakých kritérií bylo konkrétní řešení navrhováno. Jsou zde zmíněny požadavky na jednotlivé části. Druhá část se detailně zabývá návrhem a popisem jednotlivých funkčních bloků. Dále popisuje výběr a funkci vhodných součástí a integrovaných obvodů. Třetí část pojednává o konstrukci obou desek plošných spojů.

## 2.1 Kritéria a podmínky návrhu

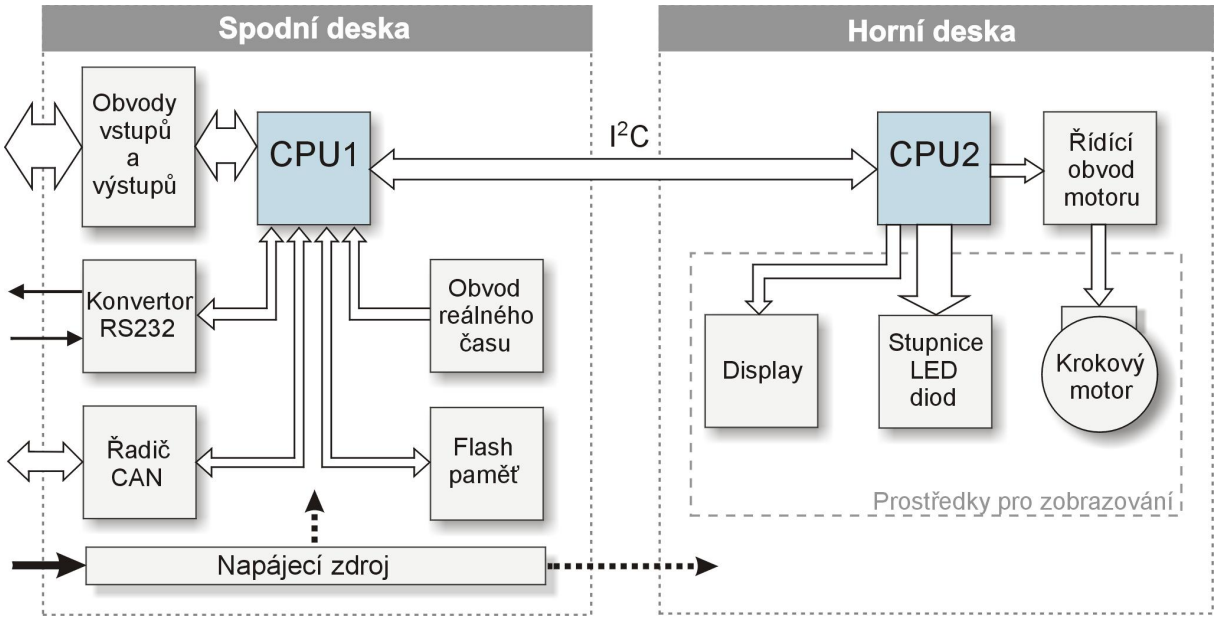
Maximální rozměry palubního počítače byly limitovány obalem, do kterého měl být umístěn. Tento vymezený prostor má kruhovou základnu o průměru 77 mm a výšku cca 50 mm. Na základě těchto parametrů bylo evidentní, že se veškeré součástky nevejdou na jednu desku plošných spojů. Z tohoto důvodu se od začátku počítalo se dvěma deskami plošných spojů uloženými nad sebou. Pro zvýšení spolehlivosti a přehlednosti byl na každou desku umístěn jeden mikroprocesor. Propojení obou desek bylo omezeno na vodiče pro komunikaci mikroprocesorů a napájecí vodiče. Jednotlivým deskám byl ohraničen obor funkčnosti a přiděleny následující bloky:

**Spodní deska** – výkonný procesor, který bude zpravovat všechny vstupy a výstupy, bude zprostředkovávat komunikaci po sběrnicích (RS232, CAN), přistupovat k paměti DataFlash a získávat informace z obvodu reálného času.

**Horní deska** – méně výkonný procesor, který bude realizovat veškeré možnosti zobrazování (stupnice z LED diody, display, ručička na krokovém motoru).

Komunikace mikroprocesorů bude probíhat po sběrnici $I^2C$.

**Blokové schéma:**



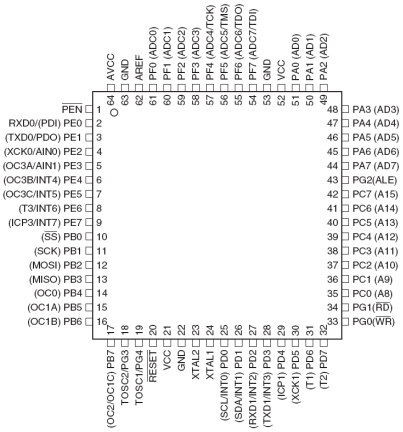

Obr. 2.1: Blokové schéma hardwaru palubního počítače

Schéma znázorňuje propojení základních funkčních bloků. Jejich podrobný popis se nachází v následující části.

## 2.2 Hardwarové funkční bloky

Tato část je věnována návrhu a popisu jednotlivých hardwarových funkčních bloků.

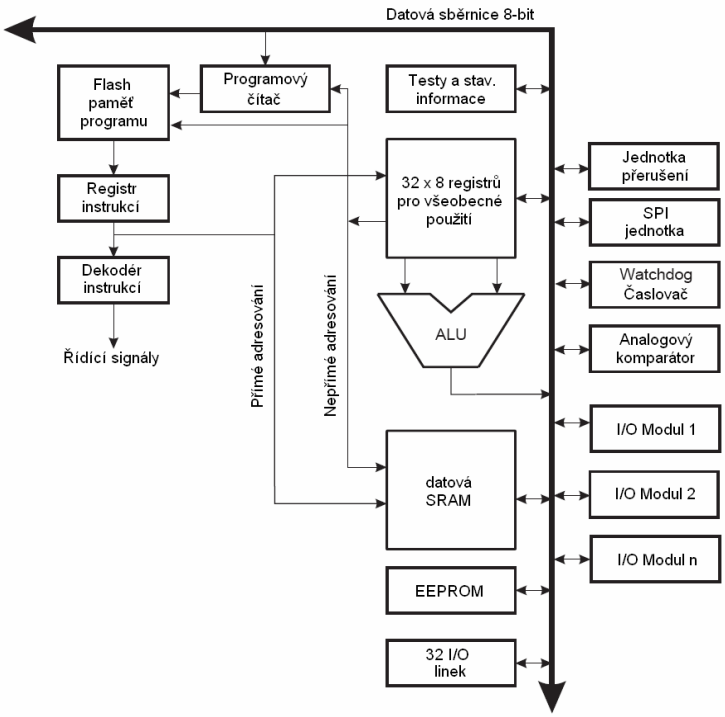
### 2.2.1 Mikroprocesor AT90CAN128




Obr. 2.2: AT90CAN128 - 64 TQFP - Rozmístění vývodů

Označení AT90CAN128 v sobě skrývá, že jde vlastně o mikroprocesor ATmega128, který je rozšířen o řadič sběrnice CAN. AT90CAN128 byl navržen speciálně pro automobilový průmysl. Tento nízkopříkonový 8-bitový mikroprocesor je založený na architektuře AVR RISC a je vyrobený technologií CMOS. Díky provádění silných instrukcí v každém hodinovém taktu se jeho výkonnost blíží 1MIPS/MHz..

Jádro AVR spojuje bohatou instrukční sadu (133 instrukcí) s 32 hlavními pracovními registry. Všechny tyto registry jsou přímo připojeny k výpočetní aritmeticko-logické jednotce (ALU), což dovoluje, aby byly dva registry nezávisle přístupné během jedné instrukce vykonávané v jediném cyklu. Struktura ALU je patrná na Obr. 2.3.




Obr. 2.3: Struktura ALU mikroprocesoru AT90CAN128

Rodina mikroprocesorů AVR disponuje velkou vnitřní pamětí typu Flash, kterou je možné programovat přímo v aplikaci (In-System Programming - ISP). Mikroprocesor má za tímto účelem rozhraní SPI (Serial Peripheral Interface) pro sériové programování. Tímto způsobem lze programovat i vnitřní paměť EEPROM, kterou je rovněž vybaven. Kromě výše uvedených pamětí je k dispozici ještě paměť SRAM, která slouží běžící aplikaci v mikroprocesoru k odkládání “nepotřebných” dat.

**Přehled základních vlastností AT90CAN128:**

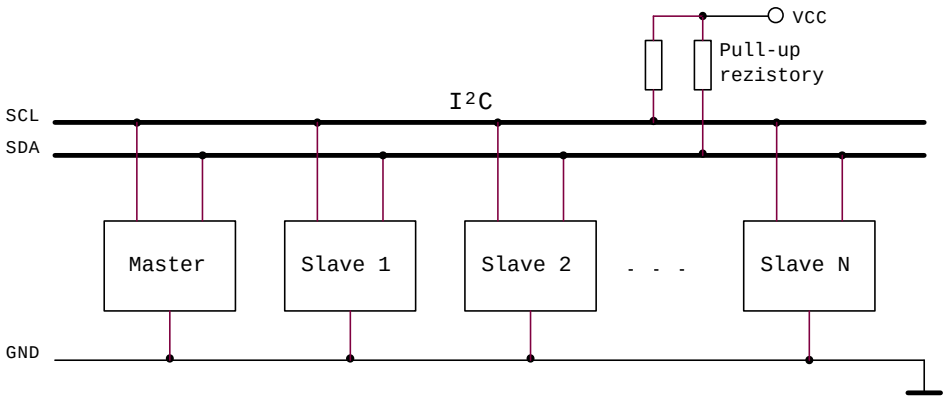
- Nezávislá programová a datová paměť:
  - 128 kB Flash paměti – In-System, Self-Programmable (až 10 000 cyklů zápis/výmaz)
  - Boot Code sekce s nezávislými Lock Bity
  - 4 kB EEPROM (100 000 cyklů zápis/výmaz)
  - 4 kB interní SRAM
  - až 64 kB volitelné externí paměti
  - programovací zámek pro bezpečnost softwaru
- JTAG (IEEE std. 1149.1)
- CAN 2.0A a 2.0B
- Periferie
  - programovatelný Watchdog s On-chip oscilátorem
  - 8-bitový synchronní časovač/čítač (Timer/Counter-0): 10-bitová předdělička, externí Event Counter, výstup Compare nebo 8-bitový PWM výstup
  - 8-bitový asynchronní časovač/čítač (Timer/Counter-2): 10-bitová předdělička, externí Event Counter, výstup Compare nebo 8bitový PWM výstup, 32KHz oscilátor pro RTC operaci
  - duální 16-bitové synchronní čítače/časovače: 10-bitová předdělička, vstup Capture s rušením šumu, externí Event Counter, 3 výstupy Compare nebo 16bitový PWM výstup, modulace výstupu Compare
  - 8-kanálový 10-bitový SAR ADC
  - Two-wire sériové rozhraní ($I^2C$)
  - duání programovatelný sériový USART
  - Master/Slave sériové rozhraní SPI
  - On-chip analogový komparátor
- Speciální vlastnosti MCU:
  - Power-on Reset a programovatelná detekce Brown-out
  - Interní kalibrovaný RC oscilátor
  - 8 externích zdrojů přerušení
  - pět režimů spánku: Idle, ADC Noise Reduction, Power-save, Power-down, Standby
  - Global Pull-up Disable
- I/O: 53 programovatelných I/O linek
- Pracovní napětí: 2.7 až 5.5V
- Rozsah teplot -40 až 125°C
- Maximální frekvence: 8MHz při 2.7V, 16MHz při 4.5V

Veškeré podrobnější údaje o mikroprocesoru AT90CAN128 jsou uvedeny v katalogovém listu výrobce [3].

Již v počátcích bylo rozhodnuto, že oba procesory spolu budou komunikovat po sběrnici $I^2C$. Vlastnosti a funkce $I^2C$ jsou tedy pro navrhování palubního počítače důležité. Z toho důvodu je v následující části tato sběrnice popsána.

**Popis sběrnice $I^2C$:**

$I^2C$ bus je zkratka která vznikla z IIC (Inter Integrated Circuit) bus. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. V dnešní době je značně rozšířená a podporuje ji řada integrovaných obvodů nejen autorské firmy Philips. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - SDA (sériová data) a SCL (sériové hodiny). Na jednu sběrnici může být připojeno více integrovaných obvodů, počet je omezen teoreticky možnostmi adresování. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je rovná 100kb/s. V režimu *fast* lze dosáhnout rychlosti až 400kb/s. V režimu *high-speed* lze dosáhnout rychlosti až 3,4Mb/s (verze 2,0). Rychlost přenosu musí být pochopitelně přizpůsobena nejpomalejšímu čipu na sběrnici. Oba vodiče musí být implicitně v logické „1“ to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4,7 kΩ.

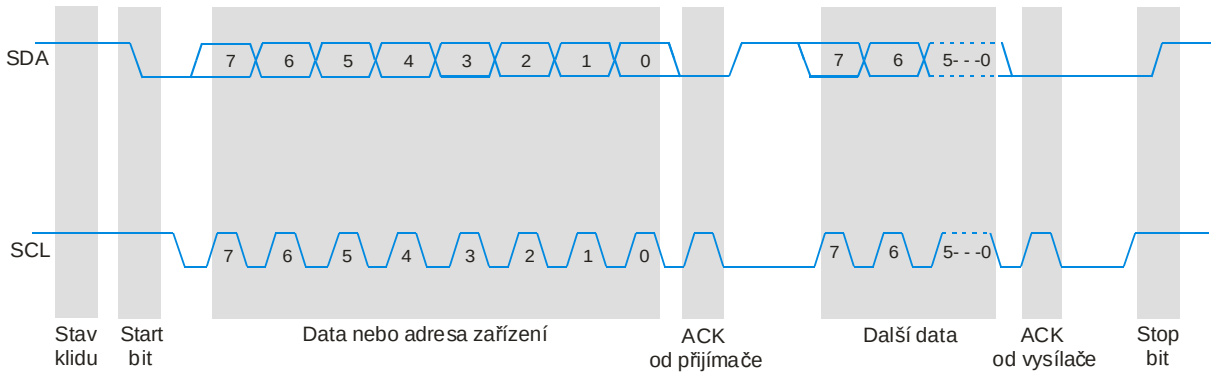



Obr. 2.4: Možné propojení jednotlivých zařízení na sběrnici $I^2C$

**Princip přenosu po $I^2C$:**

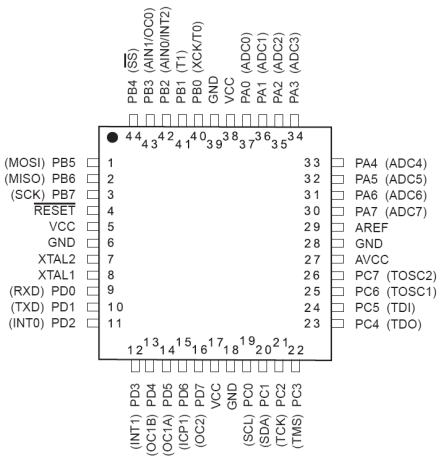
Zařízení, které pracuje jako MASTER, řídí sběrnici. Pokud si zařízení MASTER přeje zahájit komunikaci, vyšle nejprve adresu cílového zařízení. Zařízení SLAVE poslouchá na sběrnici a pokud identifikuje svou adresu, potvrdí její přijetí pomocí signálu ACK (přidrží SDA ve stavu L) po dobu devátého hodinového impulzu. Adresa je 7-bitová, osmý bit R/W (*read/write*) je použit pro signalizaci režimu komunikace. Pokud je tento bit nastaven na „1“,

bude se ze zařízení SLAVE číst. V opačném případě bude proveden zápis. Na začátku každé komunikace vysílá zařízení MASTER start sekvenci, na konci komunikace následuje stop sekvence. Data jsou přenášena po 1Byte. Při každém pulzu na SCL je přenesen jeden bit. Stav vodiče SDA je možné měnit pouze tehdy, pokud je vodič SCL ve stavu L. Je-li tedy vodič SCL ve stavu H, považuje se stav na vodiči SDA za platná data. Výjimku z tohoto pravidla tvoří pouze start a stop sekvence. Pokud dojde při komunikaci dvou zařízení k tomu, že jedno je zaneprázdněno a nemůže pokračovat v komunikaci, přidrží stav vodiče SCL na úrovni L a tím signalizuje stav *wait*. Komunikace je pozastavena a pokračuje se až po uvolnění vodiče SCL.




Obr. 2.5: Průběh logických úrovní na vodičích SDA a SCL

## 2.2.2 Mikroprocesor AVR ATmega16




Obr. 2.6: ATmega16 - 44 TQFP - Rozmístění vývodů

ATmega16 patří do stejné rodiny mikroprocesorů jako ATmega128, proto má stejné základní rysy a strukturu. Tento charakter určuje architektura AVR RISC, o které bylo již zmíněno v souvislosti s AT90CAN128. Protože jde o nižší řadu, liší se především velikostí paměti a možnostmi periferií. Pro konkrétní představu o rozdílech mezi těmito mikroprocesory je zde uveden přehled hlavních vlastností, které jsou od ATmega128 odlišné. Podrobnější údaje jsou uvedeny v katalogovém listu výrobce [4].

**Přehled rozdílných vlastností ATmega16:**

- Nezávislá programová a datová paměť:
  - 16 kB Flash paměti
  - 512 B EEPROM
  - 1 kB interní SRAM
- Bez CAN 2.0A a 2.0B
- Periferie
  - dva 8-bitové časovače/čítače se samostatnou předděličkou a módem Compare
  - 16-bitový čítač/časovač se samostatnou předděličkou, módem Compare a módem Capture
- I/O: 32 programovatelných I/O linek
- Pracovní napětí: 4.5 až 5.5V

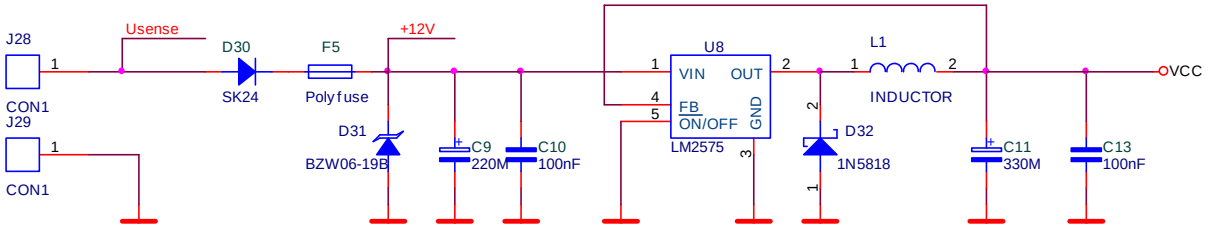
### 2.2.3 Napájecí zdroj

Palubní napětí motocyklu je 12V. Většina obvodů a součástek však vyžaduje napájecí napětí 5V, proto musí být součástí palubního počítače napájecí zdroj. Celé zařízení má být umístěno v plechovém obalu, v důsledku toho není zaručené dostatečné chlazení aktivních součástí zdroje. Při návrhu byl kladen největší důraz na rozměry a účinnost. Na základě těchto podmínek byl zamítnut zdroj se spojitým stabilizátorem (např. 7805) a byl použit spínaný stabilizátor, jehož účinnost je větší a tím pádem jsou menší i tepelné ztráty.

Pro funkci spínaného stabilizátoru byl zvolen obvod LM2575T. Tento obvod pracuje se vstupním napětím 7 až 45V, na frekvenci 52kHz a je schopen do zátěže trvale dodávat 1A. Jeho velkou předností je potřeba minimálního počtu externích součástek. Pro správnou funkci potřebuje pouze Schottkyho diodu D32, cívku L1 (330μH) a kondenzátor C11 (330μF). Zapojení a hodnoty těchto součástek byly převzaty z katalogového listu výrobce [5].

Celý zdroj byl doplněn o vstupní ochranu. Napájecí napětí je přivedeno na konektory J28 a J29. Za vstupní svorkou je umístěna dioda D30, která chrání obvod proti přepólování.

Za touto diodou následuje vratná pojistka PolySwitch na proud 1,1A. K další ochraně slouží transil D31, který nepropustí napětí větší než 22V. Vývod „$U_{SENSE}$“ připojený na analogový vstup mikroprocesoru umožňuje získat informaci o aktuálním palubním napětí.

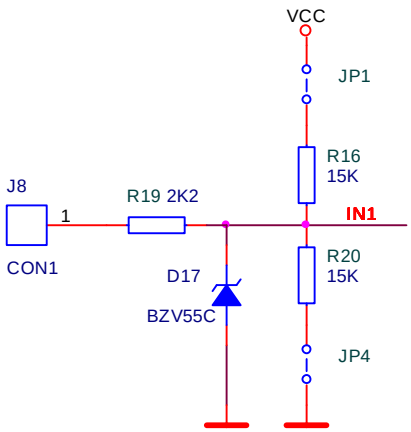



Obr. 2.7: Zapojení napájecího zdroje

## 2.2.4 Digitální vstupy

Jelikož se jedná o zařízení, které má především měřit fyzikální veličiny, je důležité vhodně navrhnout vstupní obvody. K docílení co největší všestrannosti je třeba dostatečný počet vstupů. Z toho důvodu bylo navrženo deset digitálních vstupů s možností univerzálního použití. Zapojení všech vstupních obvodů je identické.

Vnější signál je přiváděn na svorku J8, za níž následuje ochranný rezistor R19 (2,2kΩ). D17 je Zenerova dioda na 4,7V, která nepropustí dále k procesoru vyšší napětí. Digitální vstupy mají možnost nastavení klidové úrovně na VCC nebo GND. K tomu slouží pull-up rezistor R16 (15kΩ) a propojka JP1 resp. pull-down rezistor R20 (15kΩ) s propojkou JP4. Spoj „IN1“ vede k mikroprocesoru.

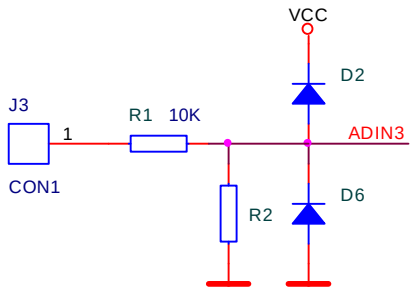



Obr. 2.8: Zapojení jednoho digitálního vstupu

### 2.2.5 Analogové vstupy

Pro účel měření analogových veličin bylo navrženo osm vstupních analogových obvodů. Dva jsou modifikovány na diferenční vstup pro měření rozdílového napětí.

Signál přivedený na vstupní svorku J3 je upraven odporovým děličem R1 a R2. Hodnoty rezistorů byly voleny tak, aby bylo možné měření širokého spektra napěťových úrovní se zachováním velikého vstupního odporu. Hodnota odporu R1 = 10kΩ je u všech analogových vstupních obvodů stejná. Hodnota odporu R2 se mění v závislosti na rozsahu hodnot vstupního napětí. Pro vstupní signál s úrovní do 15V je vhodné použít R2 = 2kΩ. Pokud bude vstup určen pro napětí do 5V, je vhodná hodnota R2 = 10kΩ. D2 a D6 jsou rychlé Schottkyho diody, které plní funkci ochrany. Návěští „ADIN3“ je spojeno s mikroprocesorem.




Obr. 2.9: Zapojení jednoho analogového vstupu

Jeden z analogových vstupů je trvale připojen na napájecí napětí ještě před napájecím zdrojem. Tento vstup má sloužit k získávání informace o aktuálním palubním napětí.

Vstupní obvody diferenčního vstupu se liší pouze ve vstupních rezistorech. Napěťový dělič je nahrazen jediným rezistorem s hodnotou odporu 10 kΩ, který je zapojen v podélné větvi. Tento rezistor má pouze ochranný význam.

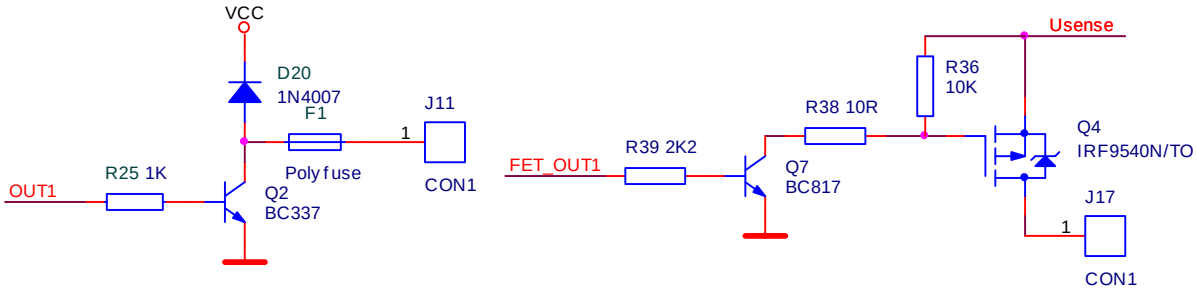
### 2.2.6 Obvody výstupů

Palubní počítač by neměl pouze měřit potřebné veličiny, ale měl by umožňovat do některých dějů i zasáhnout. Pro ovládání ostatních zařízení v motocyklu byly navrženy dva univerzální výstupy a jeden výkonový výstup s MOSFET tranzistorem.

Obě zapojení univerzálních výstupů obsahují běžný NPN tranzistor BC337, od toho se odvíjí charakter jejich použití. Tento tranzistor je schopen pracovat s kolektorovým proudem do 0,5A. Zapojení dále obsahuje ochranou diodu D1, která má význam při použití indukční

zátěže (např. relé), při jejíž rozpínání vznikají záporné napěťové špičky. K zamezení proudového přetížení výstupu slouží vratná pojistka PolySwitch F1 s hodnotou 0,5A.

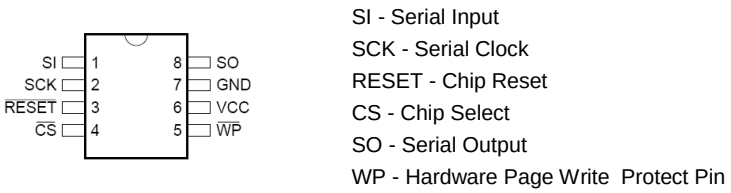
Výkonový výstup má sloužit k řízení elektronických částí motocyklu napájených palubním napětím (12V). Hlavním prvkem obvodu je P-MOSFET Q4, který umožňuje pracovat s proudy do 19A. Buzení Q4 zajišťuje univerzální NPN tranzistor Q7. Rezistor R38 = 10Ω, zapojený mezi oba aktivní prvky, omezuje proudovou špičku při rozepnutí Q4, která vzniká v důsledku kapacity řídící elektrody. Celé zapojení je oproti běžným zapojením s MOS tranzistory velmi jednoduché, to má za následek omezenou spínací rychlost výstupu.




Obr. 2.10: Zapojení univerzálního a výkonového výstupu

## 2.2.7 Externí paměť

Pro uchovávání dat naměřených a zpracovaných palubním počítačem je třeba dostatečná externí paměť. Při jejím výběru bylo uvažováno kromě typu a kapacity i se způsobem přístupu, rychlosti záznamu a čtení atd. Jako vhodná alternativa byla vybrána DataFlash AT45DB041B od firmy Atmel.




Obr. 2.11: DataFlash AT45DB041B

AT45DB041B je DataFlash se sériovým rozhraním, která vyžaduje napájecí napětí pouze 2,7V. Disponuje 4.325.376 paměťovými bity, které jsou organizovány do 2048 stránek. Každá ze stránek má rozsah 264 bytů. Kromě této hlavní paměti také obsahuje dva SRAM datové buffery, každý s kapacitou 264 bytů. Buffery umožňují jak přijímání dat zatímco se

programuje stránka v hlavní paměti, tak kontinuální čtení nebo zápis dat. DataFlash podporuje SPI sériové rozhraní (mód 0 a mód 3), což usnadňuje návrh hardwaru, zvyšuje spolehlivost systému, minimalizuje náchylnost na rušení a redukuje počet potřebných pinů. Obvod je schopen pracovat s frekvencí hodinových impulsů do 20 MHz s typickou spotřebou 4mA při čtení. AT45DB041B nepotřebuje vysoké vstupní napětí pro programování, postačuje 2,7V až 3,6V pro operace čtení i zápisu. Komunikace s pamětí je probíhá po 3-vodičovém rozhraní (sériový vstup SI, sériový výstup SO, hodinový signál SCK) za současného zpřístupnění signálem CS.

I přes to, že obvod pracuje s 3,3V, toleruje 5-Voltovou TTL logiku na všech svých signálových vstupech (SI, SCK, CS, RESET a WP). V opačném případě je i procesor schopný pracovat s úrovní logické „1“ okolo 3V. Za těchto podmínek není třeba vstupní a výstupní signály pro DataFlash nijak konvertovat, veškeré řídící signály jsou tedy přímo přivedeny k mikroprocesoru.

Z důvodu potřeby napájení napětím 3,3V (2,7 až 3,6V), je nutné zajistit snížení napětí z 5V. K tomuto účelu je před vlastní paměť zařazen stabilizátor napětí LE33, který zajišťuje na výstupu stabilizované napětí 3,3V.

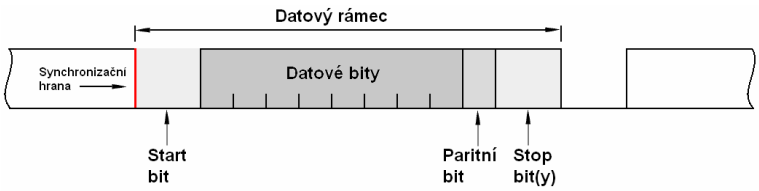
**LE33** je Low Drop stabilizátor, to znamená, že k získání určitého výstupního stabilizovaného napětí mu stačí vstupní napětí o 0,2V vyšší. Dovolené vstupní napětí však může mít hodnotu až 18V. Jmenovitý stabilizační proud je 100mA, rozsah pracovních teplot od -40 do 150°C. Obvod má vnitřní proudovou i tepelnou ochranu. Pro jeho správnou funkci je vhodné připojit na výstup kondenzátor 100nF.

### 2.2.8 Sériové rozhraní RS232

Aby byl palubní počítač schopen komunikovat s okolím (např. s PC) a bylo možné z něho získávat naměřené údaje, případně dělat servisní či diagnostické operace, musí být vybaven patřičným rozhraním. Jedním z nejběžnějších standardů je sériový asynchronní přenos RS232.

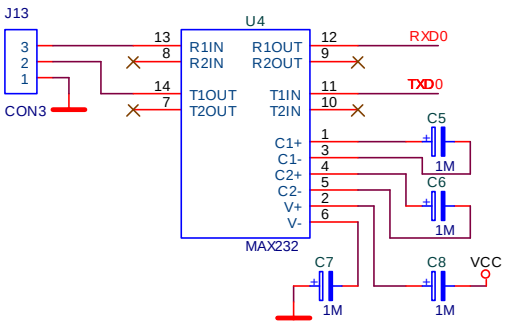
**RS232** používá dvě napěťové úrovně, kde logická „1“ je někdy označována jako „*marking state*“, logická „0“ jako „*space state*“. Pro větší odolnost proti rušení je informace po propojovacích vodičích přenášena větším napětím než standardním 5V. Log. „1“ je realizována záporným napětím, zatímco logickou „0“ představuje kladná úroveň. Nejčastěji se pro generování těchto napětí používá napěťový zdvojovač z 5V a invertor. V takovém případě

jsou logické úrovně reprezentovány takto: log „0“ +10V, log. „1“ –10V. Standardem RS232 jsou dovolené úrovně pro log. „0“ v rozmezí +3 až +15V a pro log. „1“ v rozmezí –3 až -15V. Přenos informací probíhá asynchronně, pomocí pevně nastavené přenosové rychlosti a synchronizace sestupnou hranou start bitu. Za ní již následují data.




Obr. 2.12: Datový rámec sériového přenosu RS232

Jako konvertor mezi úrovní TTL a RS232 byl použit známý obvod **MAX232** od firmy Maxim Integrated Circuits, který obsahuje dva oddělovače pro převod z RS232 do TTL a dva oddělovače pro převod opačný. Vyšší napětí pro RS232 se získává pomocí nábojové pumpy, záporné napětí vytváří invertor. Pro funkci těchto dvou částí je potřeba čtyř kondenzátorů s hodnotou 1µF. Jejich zapojení bylo převzato z katalogového listu výrobce [7]. Kondenzátor C5 slouží k násobení napětí, kondenzátor C6 k inverzi napětí. Od kvality použitých kondenzátorů se odvíjí skutečné úrovně napětí na RS232.

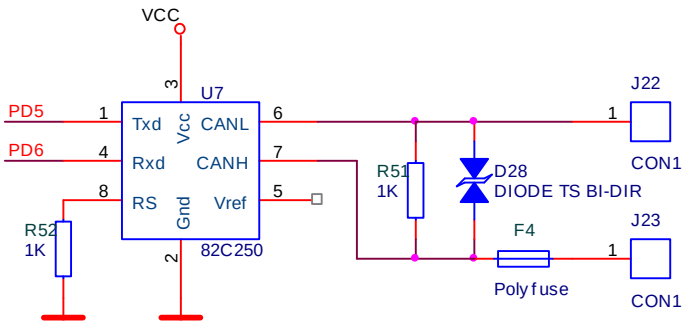



Obr. 2.13: Zapojení obvodu MAX232

## 2.2.9 Budič sběrnice CAN

*Controller Area Network* (CAN) je sériový komunikační protokol, který byl vyvinut firmou Bosch pro nasazení v automobilech. V tomto odvětví je v dnešní době hojně rozšířen. Pro svou spolehlivost, nízkou cenu, vysokou přenosovou rychlost a snadnou rozšiřitelnost se dnes používá i v různých průmyslových aplikacích. Možnost komunikovat po sběrnici CAN tedy může být pro palubní počítač v budoucím využití velkým přínosem.

Firma Atmel nabízí některé své procesory s integrovaným řadičem CAN. Toho bylo využito při výběru hlavního procesoru. Zvolené AVR s označením AT90CAN128 umí komunikovat po CAN sběrnici, bylo tedy pouze nutné připojit vhodný budič této sběrnice. Výrobců těchto obvodů je veliké množství a jejich produkty se liší v různých parametrech (rychlost, ESD–odolnost vůči elektrostatickým pulzům, napájecí napětí, spotřeba, rozměry atd.). Při volbě konkrétního typu hrála podstatnou roli dostupnost, proto byl zvolen produkt firmy Philips s označení PCA82C250. Jedná se o běžný budič sběrnice CAN, s napájecím napětím 5V, nízkou spotřebou a dostatečnou rychlostí a odolností proti rušení. Konkrétní parametry PCA82C250 jsou uvedeny v katalogovém listu výrobce [8].




Obr. 2.14: Zapojení budiče sběrnice CAN

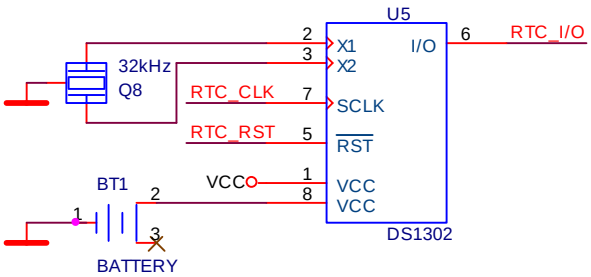
Obvod je k mikroprocesoru připojen přes piny 1 ($T_{XD}$) a 4 ($R_{XD}$). Zapojení je doplněno o rezistor R52 = 1kΩ, sloužící k volbě jednoho ze tří pracovních režimů: *high-speed*, *slope control* a *standby*. Volbou tohoto rezistoru se budiči nastavil režim *slope kontrol*, v kterém je regulována rychlost náběžné resp. sestupné hrany mezi výstupy CANH a CANL. Přestože PCA82C250 má na těchto výstupech určitou ochranu, byla tato část ještě rozšířena. Ochrannou funkci plní jednak paralelní připojení rezistoru R51 = 1kΩ a bipolárního transilu D28, a také vratná pojistka PolySwitch zapojená ve větvi CANH.

### 2.2.10 Obvod reálného času

Během provozu palubního počítače bude možné ukládat různé naměřené hodnoty do paměti. Bude-li třeba s těmito hodnotami později operovat a získat z nich užitečné informace, je nutné mít údaje o reálném čase. Proto je součástí zařízení obvod reálného času DS 1302 od firmy Dallas Semiconductor.

Obvod **DS 1302** obsahuje hodinový obvod s kalendářem a 31 bytů paměti SRAM. Komunikace je realizována přes jednoduché sériové rozhraní. DS 1302 poskytuje informace o aktuální sekundě, minutě, hodině, dnu, týdnu, měsíci a roku. Tyto informace jsou automaticky upravovány v závislosti na počtu dní v měsíci či přechodném roce. Hodiny mohou pracovat v 24-hodinovém režimu nebo 12-hodinovém režimu s indikací AM/PM. Obvod také obsahuje detektor výpadku napájecího napětí s automatickým přepnutím na záložní akumulátor. Navíc umožňuje tento akumulátor udržovat v pohotovostním stavu volitelným nabíjecím proudem. Spotřeba samotného obvodu je velmi nízká, tj. méně než 300nA (při UCC = 2V).

Zapojení obvodu bylo přejato z katalogového listu výrobce [9]. Vývody 2 a 3 slouží k připojení krystalového oscilátoru 32.768kHz. Na pin 1 je přiváděno napájecí napětí (2 až 5,5V), pin 8 je určen pro záložní baterii (zde použitá má kapacitu 180mAh). Vývody 5 (RST), 6 (I/O) a 7 (SCLK) tvoří rozhraní pro komunikaci.

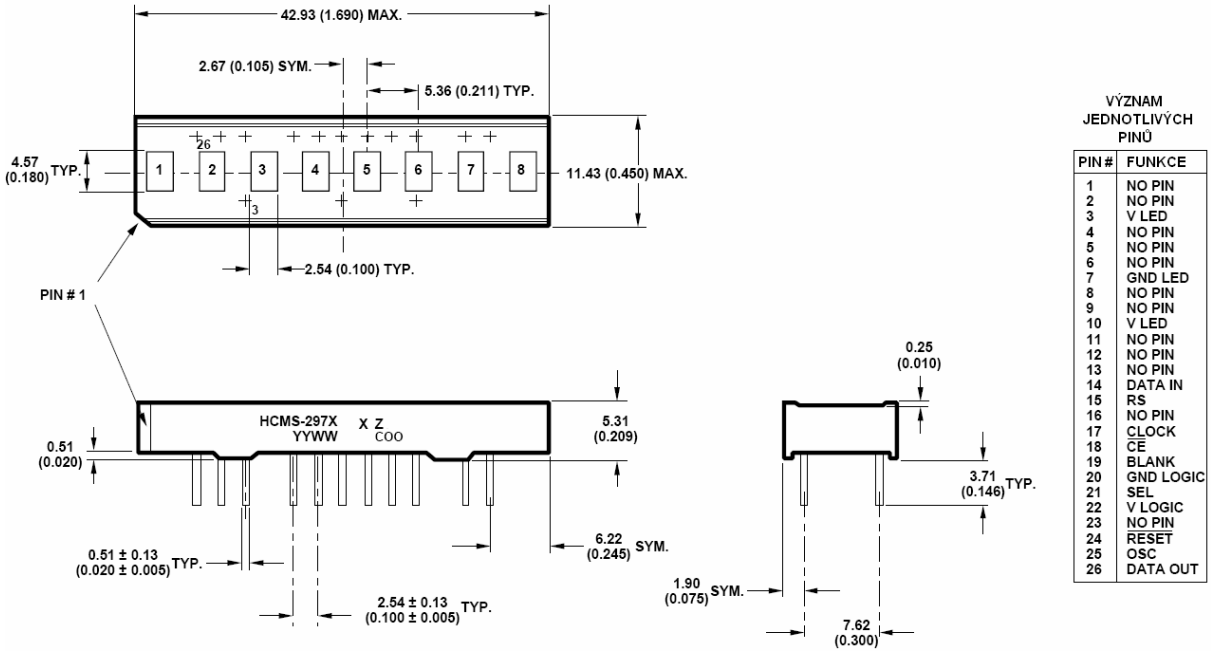



Obr. 2.15: Zapojení obvodu reálného času

## 2.2.11 Display

Stěžejním zobrazovacím prvkem celého zařízení je bezpochyby display, proto byla volba konkrétního typu velice důležitá. Při jeho výběru musela být brána v potaz různá omezení, jedním z největších byl vhodný rozměr. Palubní počítač, jak již bylo řečeno, měl pevně stanovené rozměry desek plošných spojů. Poloha displeje byla také předem určena. V důsledku toho mohly být maximální rozměry cca 45mm x 15mm. Zobrazování pouze číslic by k dostatečnému vyjádření informací nestačilo, proto musel být display buď 16-segmentový alfanumerický nebo v lepším případě maticový. Dalším aspektem výběru byl fyzikální princip zobrazování. V tomto ohledu lze volit mezi LED, LCD a dalšími méně tradičními způsoby. Po zvážení světelných podmínek, v kterých má pracovat, byl jedinou alternativou typ zobrazující LED technologií s požadavkem na vysokou svítivost.

Jako velmi vhodný produkt byl vybrán display od firmy Agilent Technologies s označením HCMS-2975.



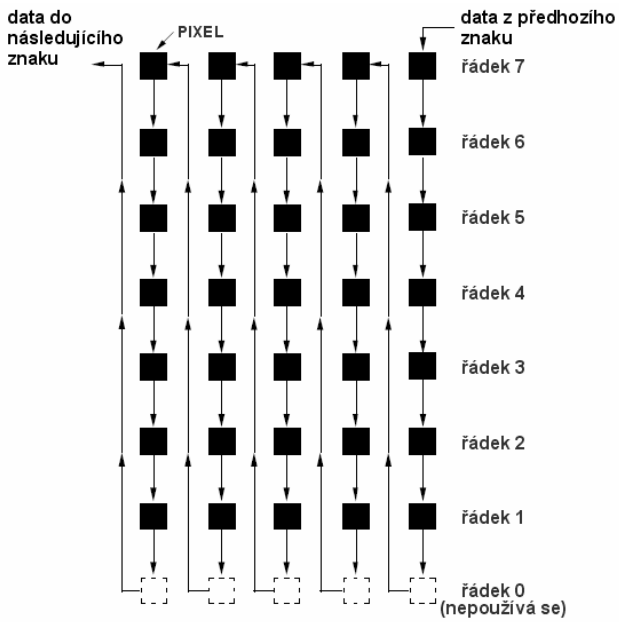

| PIN # | FUNKCE |
|---|---|
| 1 | NO PIN |
| 2 | NO PIN |
| 3 | V LED |
| 4 | NO PIN |
| 5 | NO PIN |
| 6 | NO PIN |
| 7 | GND LED |
| 8 | NO PIN |
| 9 | NO PIN |
| 10 | V LED |
| 11 | NO PIN |
| 12 | NO PIN |
| 13 | NO PIN |
| 14 | DATA IN |
| 15 | RS |
| 16 | NO PIN |
| 17 | CLOCK |
| 18 | CE |
| 19 | BLANK |
| 20 | GND LOGIC |
| 21 | SEL |
| 22 | V LOGIC |
| 23 | NO PIN |
| 24 | RESET |
| 25 | OSC |
| 26 | DATA OUT |

Obr. 2.16: Rozměry a rozmístění vývodů displeje HCMS-2975

Jde o vysoce výkonný CMOS 8-místný alfanumerický display, kde na jeden znak připadá 5x7 bodů. Toto rozlišení nabízí bezproblémové zobrazení malých i velkých písmen ASCII tabulky. Díky CMOS technologii má nízkou spotřebu (max. 2,4W), zároveň je kompatibilní s TTL. Velikou výhodou je jednoduché sériové rozhraní. Display obsahuje sériový 320-bitový posuvný registr, kde každý bit (kromě každého osmého) ovládá jeden LED pixel. Na každou z osmi zobrazovacích částí tedy připadá 40 bitů v posuvném registru. Počet LED diod v matici každého znaku je 35 (7x5), proto je každý osmý bit nevyužit. Způsob nahrávání a funkci jednotlivých bitů znázorňuje část posuvného registru na Obr. 2.17. Veškeré podrobnější informace a parametry jsou uvedeny v katalogovém listu výrobce [10].

Zapojení displeje nevyžaduje téměř žádné externí součástky. Ke komunikaci s displejem slouží signály DATA IN (pin 14) CLOCK (pin 17) REGISTER SELECT (pin 15) CHIP ENABLE a BLANK (pin 19). DATA IN je sériový vstup pro nahrávání dat. Náběžné hrany hodinových impulsů na vstupu CLOCK způsobují posun nahrávaných dat. Signál REGISTER SELECT slouží k výběru do jakého registru se budou data nahrávat. Pokud je na RS logická „0“ zapisují se data do posuvného registru pro zobrazování, je-li RS v logické „1“ dochází k zápisu do řídícího registru. Na vstupu CHIP ENABLE musí být při zápisu dat

logická „0“, při návratu do logické “1“ za současné nízké úrovně na CLOCK se data zapíší do požadovaného registru. Signál BLANK umožňuje regulovat jas displeje.




Obr. 2.17: Znázornění posuvného registru pro jeden znak displeje HCMS-2975

Display má vyveden signál RESET (pin 24), kterým lze resetovat vnitřní obvody. Reset nastane při logické „0“, přičemž dojde k vynulování celého řídícího registru. Obsah posuvného datového registru se nezmění. Signál RESET je spojen se signály RESET obou procesorů o ostatních obvodů. Dalšími vývody jsou OSCILLATOR SELECT (pin 21), který slouží k volbě externího nebo vnitřního oscilátoru. OSCILLATOR (pin 25) je určený pro případné připojení externího oscilátoru nebo výstup z interního. Pomocí rezistoru R1 = 4,7kΩ, přes který je připojen pin 21 na VCC, byl zvolen interní oscilátor. Výstup DATA OUT, sloužící pro pokračování dat v případě kaskádního řazení displejů se nevyužívá.

### 2.2.12 Krokový motor a jeho budič

U palubního počítače může být vhodnou alternativou pro přehledné zobrazování některých veličin ručička se stupnicí. Za účelem realizace této myšlenky byl do struktury zařízení přidán krokový motorek. V souvislosti s tím bylo třeba vyřešit jeho řízení.

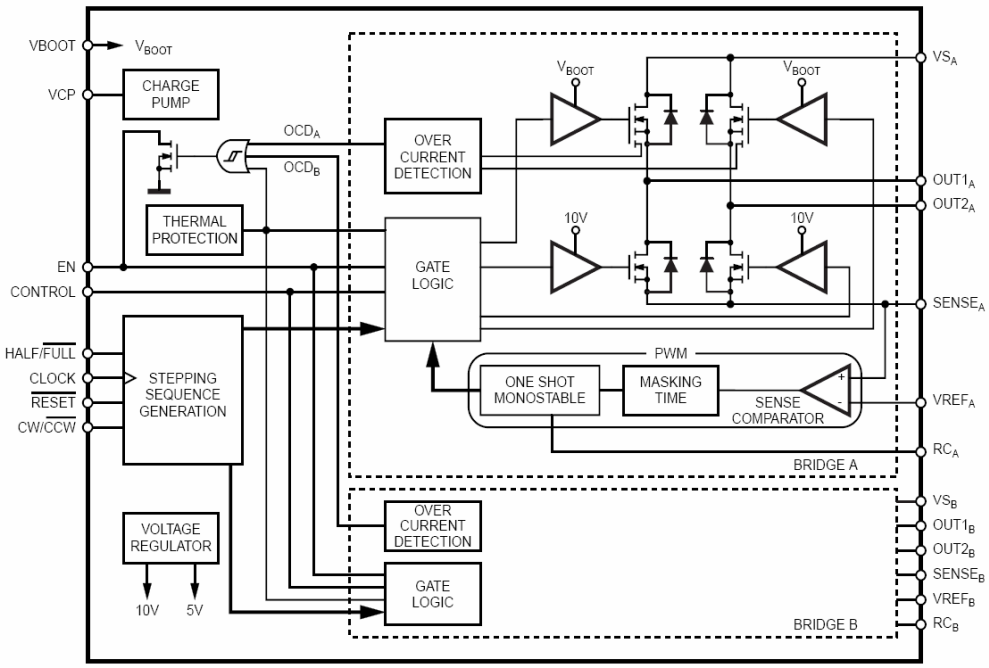
K pootáčení ručičky postačuje jen velmi malá síla resp. výkon, a proto bylo zapotřebí použít krokový motor co nejmenších rozměrů a odběru. Tyto požadavky splňují krokové motory v různých mechanikách počítače (CD, FDD), proto byl použit miniaturní krokový

motorek z 3,5“ disketové mechaniky. Jde o motor bez vyvedení středů vinutí, musí být tedy řízen bipolárně. Bipolární řízení má ve srovnání s unipolárním řadu výhod, ovšem je nutná značně složitější řídící logika a větší počet koncových spínacích členů. K plnému bipolárnímu řízení krokového motoru jsou zapotřebí dva úplné H-můstky, pro každou větev jeden (dohromady 8 tranzistorů a 8 ochranných diod). Skládat řídící obvod z jednotlivých součástek rozměry celého zařízení nedovolovaly. Tato alternativa by současně byla pro tak nenáročné použití velmi neefektivní. Protože se jedná o motor s velice malým odběrem, bylo řešení hledáno mezi integrovanými obvody. Takovéto obvody většinou nejsou běžně nabízeny, proto byla při výběru podstatná dostupnost.

Byl použit budič **L6208** od firmy STMicroelectronics. **L6208** je DMOS plně integrovaný budič krokového motoru s nadproudovou, tepelnou a podpěťovou ochranou. Technologie MultiPower-BCD kombinuje na jednom chipu DMOS výkonové tranzistory s CMOS bipolární logikou. Obvod umožňuje dvoufázové bipolární řízení s plným i polovičním krokem pomocí H-můstků s regulací proudu PWM modulací.

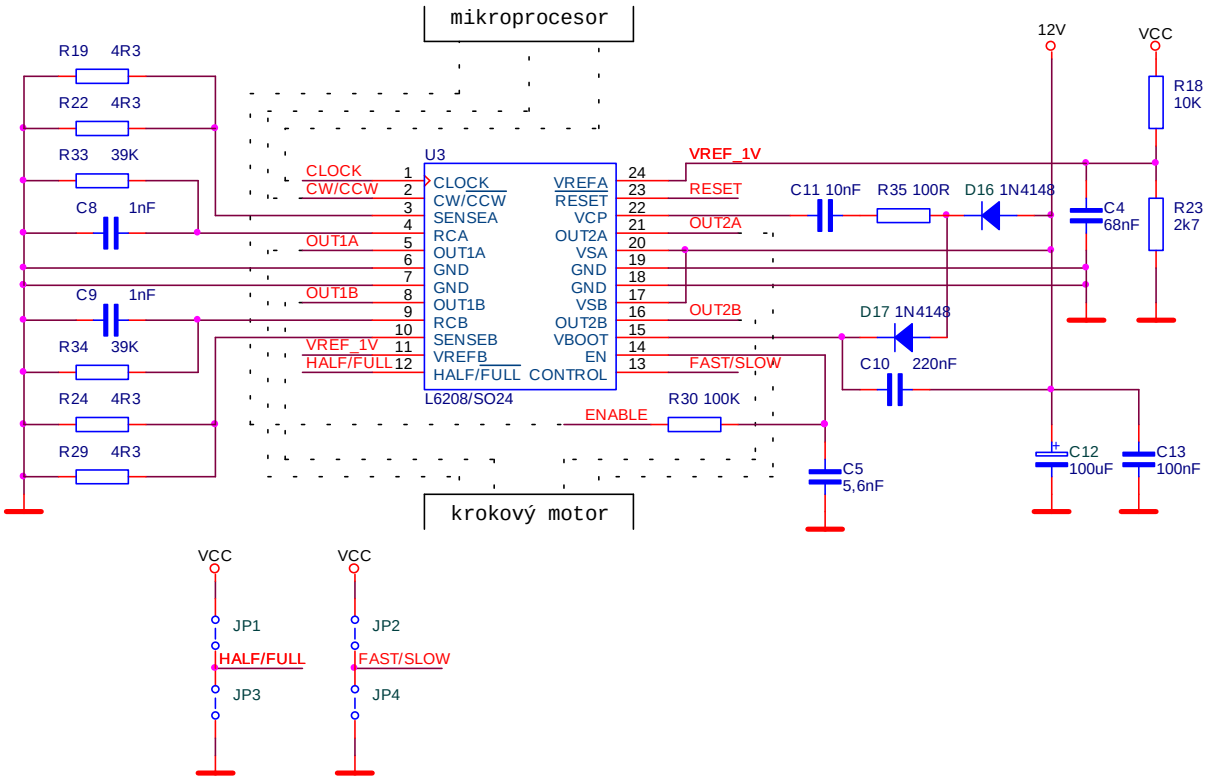
**Základní parametry obvodu:**

- Rozsah pracovních napětí: 8 až 52V
- Maximální výstupní proud: špičkový 5,6A (efektivní 2,8A)
- Odpor $R_{DS}$ otevřených výkonových tranzistorů: 0,3Ω při T = 25°C
- Pracovní frekvence až 100KHz




Obr. 2.18: Vnitřní struktura obvodu L6208

Zapojení obvodu je shodné s doporučeným zapojením v katalogovém listu výrobce [11]. Paralelní spojení rezistorů R19 a R22 umožňuje vetší proudové zatížení větve $SENSE_A$. Rezistor v této větvi slouží k získávání zpětné vazby a tím k regulaci řídícího proudu. Jeho hodnotou se tedy nechá nastavit požadovaná velikost budícího proudu. Rezistory R24 a R29 mají stejný význam, pouze se vztahují k druhému H-můstku. Vstupy VREF_1V (pin 11 a pin 24) slouží k připojení referenčního napětí 1V. K získání reference je určen napěťový dělič složený z rezistorů R18 = 10kΩ a R23 = 2,7kΩ napájený z 5V. Diody D16, D17, rezistor R35 a kondenzátory C10 a C11 tvoří nábojovou pumpu. Vyšší napětí získané z nábojové pumpy je určeno k buzení výkonových MOS tranzistorů s N-kanálem. Výstupy jednotlivých můstků jsou značeny OUT1A, OUT2A, OUT1B, OUT2B.




Obr. 2.19: Zapojení obvodu L6208

K řízení obvodu slouží signály CLOCK, CW/CCW, ENABLE, HALF/FULL a FAST/SLOW. Vysoká úroveň na vstupu ENABLE je nutná k povolení řízení, v této úrovni jsou aktivovány výkonové MOSFETy obou H-můstků. Signálem CW/CCW se určuje směr otáčení, logická „1“ znamená otáčení ve směru hodinových ručiček, logická „0“ opačný směr. Pomocí signálu na vstupu CLOCK se stanovuje kolik kroků má motor vykonat. Signál HALF/FULL slouží k volbě plného čí polovičního krokování. Pomocí FAST/SLOW se

nastaví rychlost a návaznost přepínání koncových tranzistorů, vlastnosti obou módů jsou důkladně popsány v katalogovém listu [11]. Signály HALF/FULL a FAST/SLOW není třeba během provozu měnit, proto nejsou připojeny na port mikroprocesoru, ale mají možnost nastavení pomocí propojek JP1, JP2, JP3 a JP4. Vstup RESET je spojen se signály RESET ostatních obvodů.

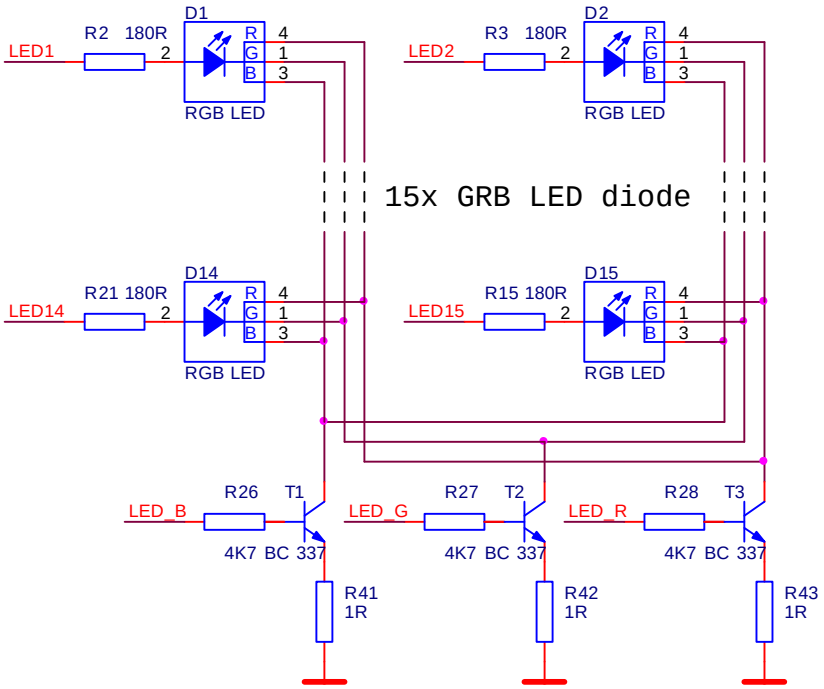
### 2.2.13 Stupnice z LED diod

Skupina LED diod je další možností jak zprostředkovat naměřené čí vypočtené údaje uživateli. Všechny diody mohou tvořit stupnici, která bude znázorňovat hodnotu jedné veličiny. Mohou však také sloužit jako indikace (kontrolky) k více podnětům najednou. Tyto předpoklady vedly k úsudku, že je vhodné, používat zobrazování pomocí více barev. Pro stupnici byly zvoleny LED diody, které emitují trojici základních barev (RGB). Jedná se o diody se společnou anodou, to znamená, že jednotlivé barvy mají samostatně vyvedenu katodu. V zapojení je počítáno s patnácti LED diodami.

Při návrhu zapojení bylo uvažováno s možností zobrazovat libovolnou barvu spektra sloučením dílčích složek RGB. V pouzdře jsou jednotlivé barvy realizovány jako samostatné LED. Emisi rozdílných barev způsobuje polovodičový materiál pro ně použitý. Různé materiály však mají za následek i rozdílné napětí potřebné k rozsvícení červené, zelené a modré barvy. Z toho vyplývá, že nemohou být najednou rozsvíceny dvě barvy v jedné LED a míchání barev se musí řešit multiplexem.

Odběr celé LED diody se tedy bude rovnat odběru jediné barvy nebo bude nižší. Proudové zatížení jednotlivých pinů vstupních a výstupních portů mikroprocesoru je asi 20mA, a proto nejsou diody řízeny přes tranzistory ale přímo připojeny k mikroprocesoru. Pomocí tranzistorů jsou spínány pouze větve jednotlivých barev, pro případ rozsvícení jedné barvy u všech LED najednou. Na anody diod je přiváděno napětí přes rezistory s hodnotou odporu 180Ω, což zajišťuje odběr každé z nich do 10mA. V emitorech tranzistorů spínajících jednotlivé větve s barvami, jsou zařazeny rezistory s hodnotou 1Ω. Tuto hodnotu lze částečně upravit pro případ, že bude třeba vyrovnat jasy jednotlivých barev. Regulace je však omezená z důvodu proudu, který těmito rezistory v krajních případech může téci. Je-li uvažováno s odběrem 10mA jedné LED a svítí-li všechny najednou, prochází editorovým rezistorem proud 150mA. Pokud bude hodnota odporu 10Ω, bude se na rezistoru ztrácet téměř čtvrt

Wattu. Je evidentní, že 10Ω vzhledem v 180Ω připojeným na anodu, nebude mít na změnu jasu zásadní vliv.




Obr. 2.20: Zapojení LED diodové stupnice

### 2.2.14 Ostatní hardwarové části

Kromě výše uvedených hardwarových bloků zařízení obsahuje několik funkčních částí, které pro svou jednoduchost nebyly popisovány.

Na spodní desce se jedná o podpěťový resetovací obvod MPC120, jehož výstupní signál je veden přes propojku, aby jej bylo možné vyřadit.

Horní deska navíc obsahuje LED diody pro podsvícení a jednoduchý zdroj proudu pro jejich napájení. Zdroj je signálově spojen s mikroprocesorem, je tedy umožněno spínání čí regulace jasu řízených LED diod. Dalším obvodem na horní desce je zapojení s fotorezistorem, které může sloužit k automatické regulaci jasu všech světelných zobrazovacích prvků.

## 2.3 Konstrukční řešení

Následující tři podkapitoly uvádějí, jakým způsobem bylo řešeno konkrétní konstrukční provedení. Jsou zde uvedeny vlastnosti desek plošných spojů a kritéria, podle kterých byla volena pouzdra součástek. V poslední části je popsaná podoba hotového zařízení.

### 2.3.1 Plošné spoje

Obě desky plošných spolů byly navrhovány v návrhovém systému OrCad 10 a realizovány jako dvoustranné. Z důvodu co největší odolnosti proti mechanickému poškození, byly jednotlivé cesty voleny co možná nejširší. Pochopitelně byl brán zřetel na proudové zatížení jednotlivých spojů. Bylo dodrženo oddělení výkonové, digitální a analogové země (GND). Obě desky plošných spojů mají na obvodu čtveřici otvorů (průměr 3,2mm) pro přišroubování na distanční můstky. Tyto otvory jsou symetricky rozmístěné, aby bylo možné vzájemné pootočení desek o 90°, 180° a 270° oproti původně uvažované poloze. Do horní desky byl uprostřed vyfrézován otvor (průměr 6,5mm) pro osazení krokového motoru a dva otvory (průměr 2,5mm) pro šrouby na připevnění tohoto členu.

Obrázky plošných spojů, rozmístění součástek, fotografie vyrobených desek plošných spojů a výrobní podklady jsou umístěny v příloze.

### 2.3.2 Pouzdra součástek

Pouzdra veškerých součástek byla volena tak, aby zabrala co nejméně místa. Převážně se jedná o provedení SMD. V některých případech bylo vhodnější použít klasické součástky, protože měly menší základnu a jejich výška nebyla limitující. Všechny rezistory a kondenzátory mají velikost 0805, pokud nebylo nutné použití větší. Tranzistory pro spínání LED diod a tranzistory výstupů byly kvůli velikosti spínaného proudu zvoleny v pouzdře TO92. Téměř všechny součástky napájecího zdroje také nejsou SMD. Obvody LE33 (stabilizátor 3,3V) a MPC120 (podpěťový resetovací obvod) jsou nabízeny v provedení SO-8 resp. SOIC, ale v této verzi zabírají na plošném spoji více místa, proto byla zvolena též pouzdra TO92. Přehled pouzder použitých obvodů je uveden v následující tabulce:

| Označení | Popis | Pouzdro |
|---|---|---|
| AT90CAN128 | Hlavní mikroprocesor | TQPF-64 |
| ATmega16 | Sekundární mikroprocesor | TQPF-44 |
| LM 2575T | Spínaný zdroj 5V | TO 220 |
| AT45DB041B | DataFlash 4MB | 8-SOIC |
| LE33 | Stabilizátor 3,3V | TO92 |
| MAX232 | Konvertor RS232 | SO-16 |
| PCA82C250 | Budič sběrnice CAN | SO-8 |
| DS 1302 | Obvod reálného času | 8-SOIC |
| L6208 | Budič krokového motoru | SO-24 |
| MPC120 | Podpěťový resetovací obvod | TO92 |

Tab. 2.1: Přehled významných použitých obvodů a jejich pouzder

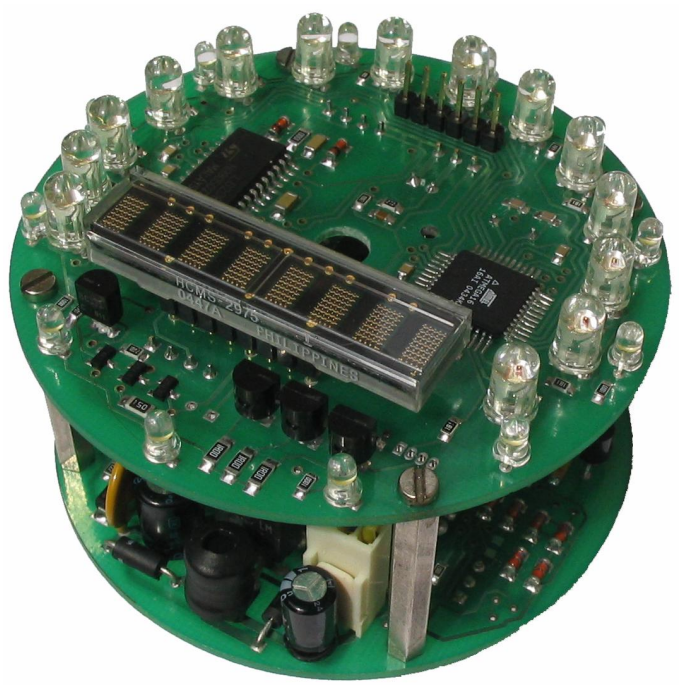
Cívka (L1), kterou obsahuje napájecí zdroj, byla k dostání pouze ve variantě s nedostatečným zatěžovacím proudem. Tento problém byl vyřešen rozebráním cívky a jejím opětovným namotáním s využitím silnějšího vodiče. V zapojení se také vyskytuje množství propojek (JP1 až JP21) pro nastavování režimů a připojování pull-up či pull-down rezistorů. Klasické jumpery jsou příliš veliké a jejich propojení s ohledem na vibrace nespolehlivé, bylo proto přistoupeno na efektivnější řešení. Propojky jsou realizovány jako plošky na desce plošných spojů umístěné blízko sebe. Spojení a rozpojení je tedy nutné provádět pájkou.

Kompletní seznam použitých součástek je uveden v příloze.

### 2.3.3 Mechanické uspořádání a konektory

Obě desky plošných spojů jsou spojeny distančními sloupky s délkou 30mm. Do prostoru mezi nimi směřují veškeré rozměrnější součástky z horní i spodní části. Propojení obou desek je provedeno jedním plochým 4-žilovým kabelem (pro signály) a jedním plochým 3-žilovým kabelem (pro napájení). Signálový vodič je zakončen běžnými plochými konektory CONPC-SPK-4 s roztečí 2,54mm. Na napájení byly zvoleny konektory PSH02-03PG resp. PSF02-03P s možností zatížení vyšším proudem (cca 2,5A) a zámkem polohy. Konektory pro programování tvoří jednoduchá 6-pinnová lišta. Pro připojení vnějších vodičů byly použity svorky AKR RM s roztečí 3,5mm, které jsou umístěné na spodní straně dolní desky a jsou uspořádány do dvou řad po dvanácti. Jedinou výjimkou je konektor (PSH02-03PG) sběrnice RS232, jež je řešen samostatně. Označení konektorů je převzato z katalogu GM Electronic [17].

Krokový motor je k horní desce plošných spojů přichycen pomocí příruby, která byla vyrobena speciálně k tomuto účelu. Tento díl tvoří též vedení osy motoru, protože samotný motor nemá žádná ložiska.



Obr. 2.21: Podoba hotového zařízení (bez krokového motoru)

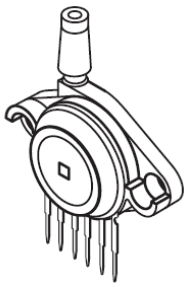
# 3 Volba vhodných senzorů

Palubní počítač pro motocykl nabízí využití mnoha vstupů, které byly navrženy tak, aby vyhovovaly co nejširší oblasti možných požadavků. Pro měření „elektrických“ veličin by měly postačovat možnosti samotného zařízení.

Pro měření ostatních veličin je třeba zvolit vyhovující senzory. Fyzikálních veličin, které se dají na spalovacím motoru měřit, je opravdu mnoho. V následující části tedy nejsou popsána veškerá možná čidla a snímací obvody, ale pouze pár zajímavých senzorů, s jejichž použitím se od začátku počítalo. I přes uvedení pouze těchto senzorů je možné spolupracovat s téměř libovolnými. Nebudou-li u nich zajištěny vhodné výstupní signály, je nutné vyhotovit speciální externí obvody, které tento požadavek zaručí.

## 3.1 Čidlo tlaku

Důležitou hodnotou pro správnou diagnostiku je údaj o podtlaku v sacím potrubí resp. karburátoru(ech). Vhodné podtlakové čidlo je např. MPX4115AP od firmy Motorola.



Obr. 3.1: Čidlo tlaku MPX4115AP

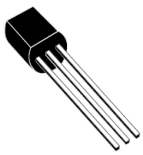
Obvod v sobě integruje vlastní čidlo, výstupní zesilovač a obvody pro teplotní kompenzaci. Citlivost je 45,9mV/kPa, výstupní napětí se určí podle vztahu:

$$V_{OUT} = VCC \cdot (0{,}009 \cdot P - 0{,}094) \pm chyba$$

Pro střední hodnotu tlaku 1000kPa vychází při napájecím napětí VCC = 5,1V výstupní signál s úrovní $V_{OUT}$ = 4,1V. Bližší parametry a funkce tohoto čidla jsou popsány v katalogovém listu výrobce [14].

## 3.2 Teplotní čidlo

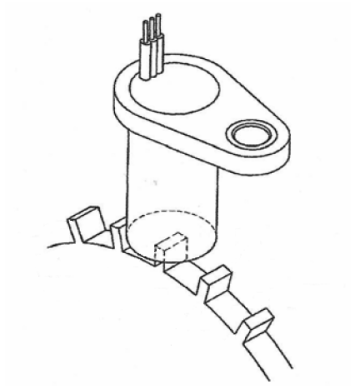
Teplotních senzorů je na trhu nabízeno skutečně mnoho. Příkladem možného jednoduchého čidla je LM135 od firmy SGS-THOMSON Microelectronics, které je běžně k dostání v pouzdře TO92. LM135 má lineární charakteristiku v celém rozsahu, teplotní rozsah -55 až +150°C a umožňuje snadnou kalibraci. Podrobnější specifikaci lze najít v katalogovém listu výrobce [15].



Obr. 3.2: Teplotní čidlo LM135

## 3.3 Snímání zubů na klikovém hřídeli

Za účelem počítání zubů na klikovém hřídeli bylo vybráno čidlo 1GT101DC pracující na principu Hallova jevu. Na výstupu tohoto senzoru je možné získávat digitální signál, jehož frekvence odpovídá počtu sejmutých zubů za jednotku času (sekundu). 1GT101DC vyžaduje napájecí napětí 4,5V až 24V a má typický odběr 10mA. Konkrétnější popis lze najít v katalogovém listu výrobce [16].



Obr. 3.3: 1GT101DC - senzor snímání zubů na klikovém hřídeli

## 3.4 Senzor rychlosti

Pro zjišťování rychlosti motocyklu je použit jednoduchý a velmi rozšířený snímač. Jedná se o mechanický jazýčkový spínač reagující na magnetické pole. Malý magnet

umístěný na kole způsobí, že při každém průchodu kolem snímače (při každé otáčce) dojde ke spojení obou vstupních vodičů. Z informace o počtu otáček kola, jeho obvodu a času je možné rychlost vypočítat.

# 4 Softwarové řešení

V této části je pojednáno o jazyku a prostředí, ve kterém byl software pro palubní počítač napsán. Podstatná část této kapitoly však popisuje právě tento základní software a jeho části. Vytvořený software demonstruje pouze zlomek možností celého zařízení.

## 4.1 Programovací jazyk na vyšší úrovni

Při psaní jednoduchých programů pro mikroprocesory je možné využít assembler. U složitějších a rozsáhlejších programů, kde se využívají stále se rozvíjející schopnosti „jednočipů“, je vhodnější využít nějaký vyšší programovací jazyk. Proto byly pro tyto účely vytvořeny překladače z vyšších programovacích jazyků. Velké obliby dosáhl zejména jazyk C, což je dané tím, že má nejen vlastnosti, které se očekávají od vyšších programovacích jazyků, ale i vlastnosti očekávané spíše u assemblerů. Z vyšších programovacích jazyků má jazyk C "nejblíže" k hardwaru.

Programování v C je oproti assembleru výrazné zjednodušení, protože například dělení dvou 32-bitových čísel lze bez problémů napsat na jeden řádek. Toto „příjemnější“ programování má i svoje nevýhody, neboť program je přirozeně větší a pomalejší. Nicméně výsledný kód programu je mnohem kratší a jeho vývoj podstatně rychlejší.

Jazyk C používaný pro mikroprocesory je od klasického jazyka C nepatrně odlišný. Při psaní je vhodné oprostit se od složitých konstrukcí podmínek a nepoužívat složité zápisy. Tím je myšleno to, že se v jediném řádku provede jak sčítání či umocnění na základě vyhodnocení podmínky podle vstupu a celý tento příkaz je v cyklu. Pokud je psán kód pomalu a srozumitelně, je tím téměř jisté, že vytvořený kód bude skutečně prováděn v mikroprocesoru tak, jak bylo požadováno.

Struktura mikroprocesorů AVR je navržena tak, aby co nejvíce vyhovovala překladačům vyšších programovacích jazyků, zejména již zmíněného jazyka C. Velice dobrým a rozšířeným kompilátorem pro programování AVR v jazyce C je kompilátor GCC, který je obsažen v balíku WinAVR. Hlavní předností kompilátoru je jeho zcela volná šiřitelnost. Je vydáván pod licencí GNU což znamená že ho lze šířit pokud je uveden pouze jeho původní zdroj.

## 4.2 Kompilátor GCC v balíku WinAVR

**WinAVR** je programový balík pro Windows postavený na kompilátoru GCC. Jeho instalace se dá volně stáhnout, stále se objevují nové aktualizace. Distribuce WinAVR obsahuje kompletní nástroje pro programování: editor, kompilátor GCC, programátor UISP, knihovny *avr-libc* a další utility pro programování atd. Kompletní informace o balíku jsou uvedeny v WinAVR User Manual, který je také součástí instalace [18].

**Kompilátor GCC** jazyka C vznikl v rámci projektu GNU. Tento projekt se zabývá vývojem alternativních volně šiřitelných programů především pro systémy UNIXového typu. Projekt vznikl díky velkému množství nezávislých programátorů z celého světa a svou kvalitou je schopen konkurovat i produktům velkých komerčních firem.

Vše začalo dopisem odeslaným 27. září 1983 ve 12:35:59 Richardem Stallmanem z MIT AI Lab v Cambridgi. Tento autor editoru Emacs a mnoha dalších interpretrů, kompilátorů, grafických knihoven atd. v dopisu píše, že se rozhodl vytvořit alternativu komerčních implementací UNIXu. Jeho základní pravidlo vyžaduje: „Používám-li nějaký program, který se mi líbí, musím mít právo ho vylepšit a poskytnout i ostatním, kterým se líbí“. V současné době se o správu projektu GNU stará nezisková organizace pro software v obecném zájmu Free Software Foundation, Inc.

Přesto, že se vlastní systém GNU příliš nerozšířil, umožnila existence volného programového vybavení vznik operačních systémů jako je FreeBSD, Hurd a Linux. Veškeré programy jsou napsány s maximální snahou o přenositelnost ve zdrojové formě na většinu UNIXových systémů včetně IBM OS/2 (projekt EMX). Velké množství je možné provozovat i v relativně nekompatibilních prostředích jako je DOS (projekt DJGPP) nebo Win 95 a Win NT (projekt CygWin32).

Následující nainstalované programy jsou nutné k přípravě GNU vývojového prostředí kompilací ze zdrojových textů:

**make -** verze GNU programu make, který podle času modifikace jednotlivých souborů a databáze možných způsobů jejich překladu rozhoduje o tom, které soubory a jak je nutné přeložit a poté spojit do nových verzí výsledné aplikace

**gcc -** GNU nebo i jiná verze překladače jazyka „C“

**as -** překladač assembleru (nejlépe též verze GNU)

**ld -** linker, program pro spojování částí strojového kódu do výsledného programu

**ar -** knihovní program pro správu fragmentů strojového kódu

**nm** - program pro výpis symbolických jmen použitých ve fragmentu strojového kódu

**bison** nebo **lex** - lexikální analyzátor pro překlad lexikálního popisu do jazyka „C“, jeho přítomnost není absolutně nutná, protože GCC je již dodáváno jak se zdrojovými texty pro bison tak s přeloženými soubory v jazyce „C“.

**m4** - makro expander, jeho přítomnost není absolutně nutná

**makeinfo** - systém pro přípravu dokumentace do hypertextového formátu info - jeho přítomnost je také volitelná

Překladač GCC je navržen tak, aby umožňoval i snadnou vlastní přenositelnost na systémy, na kterých zatím nebyl provozován a i pro nové systémy, pro které zatím jiný překladač neexistuje. Tato vlastnost je velmi výhodná i pro vývoj řídících aplikací, kde se implementace samotného kompilátoru GCC na cílové platformě nepředpokládá.

Kvalita výsledného kódu v assembleru je přinejmenším srovnatelná s většinou komerčních překladačů. Překladač je postupně optimalizován a rozšiřován o další cílové (target) a hostitelské (host) platformy.

Popis GCC a GNU byl převzat ze stránek Ing. Pavla Píši [19].

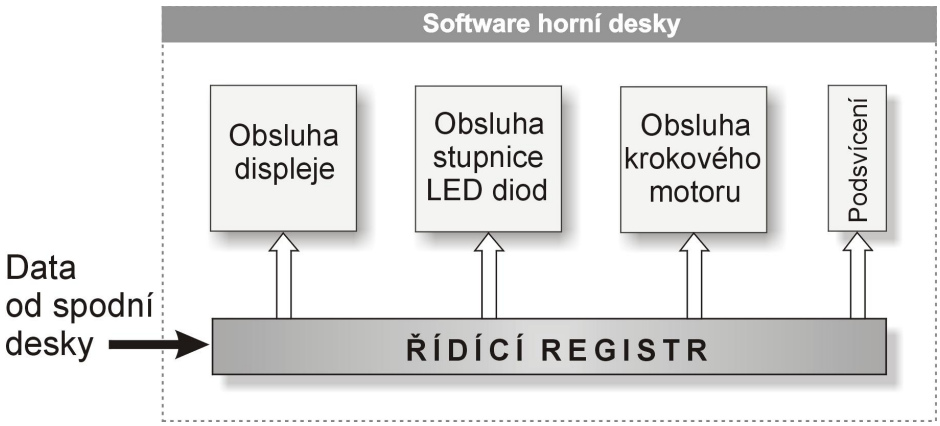
## 4.3 Software pro horní desku

Mikroprocesor na horní desce obstarává, jak již bylo mnohokrát uvedeno, pouze veškeré řízení prostředků pro zobrazování. Konkrétně se jedná o display, stupnici z LED diod a krokový motor. Druhotnou úlohou tohoto mikroprocesoru je ovládání podsvicovacích LEDek a regulace jasu všech použitých světelných zdrojů v závislosti na okolním osvětlení. V souvislosti s těmito úkoly byl navržen software, který řeší veškeré hlavní funkce horní desky. Software obsahuje několik knihoven, jejichž vzájemná struktura a popis se nachází v následující části.

### 4.3.1 Struktura softwaru

Na Obr. 4.1. je znázorněno základní členění částí softwaru pro horní desku. Úkolem horní desky je interpretovat to, co požaduje deska spodní. Aby nebylo nutné při komunikaci mezi těmito dvěma částmi řešit složité dorozumívání o tom, co se má kde zobrazit, je součásti softwaru horní desky řídící registr. Tento registr má pevný počet buněk,

kde každá poskytuje informace pro konkrétní zobrazovací prvek. Jednotlivé obsluhy zobrazovacích prvků tedy vědí, kde v registru se nachází data právě pro ně.




Obr. 4.1: Struktura softwaru pro horní desku

Zprávy přijímané od spodní desky jsou uspořádány podle jednoduchého protokolu. Každá přijatá zpráva obsahuje na prvním Byte adresu, na kterou mají být data uložena v řídícím registru. Tím je jednoznačně určeno, kterému zobrazovacímu prvku informace náleží. Další Byte je již první datový. Samotná zpráva neobsahuje údaj o délce zprávy, neboť tato informace se přenáší na základě principu nejnižší úrovně komunikace po sběrnici $I^2C$ a lze jí využít. Jsou-li tedy data přijata, uloží se na patřičné místo do řídícího registru. Odtud si je mohou jednotlivé obsluhy podle potřeby načíst.

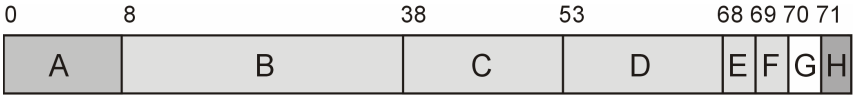
Pro displej, stupnici z LED diod i krokový motor byly napsány speciální knihovny, které umožňují jejich snadné používání. Stručný popis těchto knihoven je uveden v následujících kapitolách.

**Popis řídícího registru:**

Pro snadný popis řídícího registru a významu jeho jednotlivý částí je třeba uvést režimy, v kterých může stupnice z LED diod pracovat a pro které byla koncepce registru navržena.

Jednotlivé módy stupnice jsou:

- zobrazování hodnoty pomocí sloupce rozsvícených elementů
- zobrazování hodnoty pomocí bodu pohybujícím se po stupnici
- funkce kontrolek, každá LED signalizuje stav něčeho jiného




Obr. 4.2: Struktura řídícího registru

Oblast „A“ na Obr. 4.2 má velikost 8 Byte. Každý Byte obsahuje jeden znak, který má být zobrazen na displeji. Oblasti „B“, „C“, „D“, „E“ a „F“ slouží pro správu stupnice z LED diod. Barva každé diody je určena 16-bitovým slovem, spodní 3 bity (0, 1, 2) určují podíl červené ve výsledné barvě. Bity 3, 4, 5 znamenají podíl zelené a bity 6, 7 a 8 příspěvek modré barvy. Protože se stupnice skládá z 15ti LED, kde každá potřebuje 2 Byte pro určení odstínu, je pro údaje o barvě vyhrazen prostor 30 Byte (oddíl „B“). Části „C“ a „D“ se využívají v případě, že LED diody fungují jako kontrolky. V „C“ je pro každou LED vyhrazen 1 Byte pro hodnotu, která znamená, po jakou dobu má svítit. Oblast „D“ má stejný rozsah ale určuje po jakou dobu je LED zhasnutá. Toho se dá využít v případě, kdy je požadováno blikání. Pokud má kontrolka nepřetržitě svítit, musí mít odpovídající Byte v „D“ nulovou hodnotu. Zbytek řídícího registru („E“, „F“, „G“, „H“) je členěn po 1 Byte. Pomocí „E“ je volen požadovaný režim LED diod. V bloku „F“ se nachází hodnota, kterou má stupnice zobrazit. „G“ slouží pro předání pozice, na kterou se má přesunout krokový motor a „H“ obsahuje informaci o požadovaném jasu LED pro podsvícení.

### 4.3.2 Knihovna pro display

Tato knihovna obsahuje základní funkce pro obsluhu displeje, jehož vlastnosti a popis zapojení byly uvedeny v kapitole 2.2.11. Některé významné funkce jsou popsány v následující části.

**void IO_DispInit (void)**

Součástí knihovny je samozřejmě tato funkce pro inicializaci, která musí být volána před všemi operacemi s displejem. Je bezparametrická a provádí zápis převážně do řídícího registru displeje. Při jejím zavolání dojde k nastavení všech potřebných parametrů.

**void Display (char* str)**

Jde o stěžejní funkci celé knihovny, která zajišťuje samotné zobrazování. Parametrem je ukazatel na řetězec znaků, které se mají zobrazit.

Displej se chová jako 320-bitový posuvný registr, uspořádání a význam jednotlivých bitů je popsán v kapitole 2.2.11, konkrétně na Obr. 2.17. Funkce tedy nejprve zajistí převod vstupních ASCII znaků na posloupnost bitů, které po zapsání do displeje vytvoří požadovaný znak. Za tímto účelem je ve funkci obsaženo pole *Pixels* 8-bitových slov s délkou 40, které se po naplnění odešle na display. K převodu znaků na posloupnost bitů a plnění zmíněného pole se využívá funkce *OutChar*.

**void OutChar (u08* pix, u08 pos, u08 ch)**

Funkce slouží k zaplnění odpovídající části výše uvedeného pole posloupností bitů, která odpovídá převodu jednoho znaku. Odpovídající bitovou kombinaci načítá z dalšího pole *FontLookup*, které obsahuje k veškerým znakům ASCII potřebnou bitovou posloupnost.

**void DispClear (void)**

Zavoláním této funkce se do posuvného registru displeje zapíší samé „0“, což má za následek jeho kompletní vymazání.

Knihovna dále obsahuje funkci *Display2*, která plní stejnou úlohu jako funkce *Display*. Jediným rozdílem funkcí jsou vstupní parametry, do *Display2* vstupuje 8 proměnných typu *char*.

### 4.3.3 Knihovna pro LED diody

Knihovna nabízí využití LED diodové stupnice v širokém rozsahu možností. Poskytuje funkce pro práci LED jako sloupcový či bodový bargraph, nebo jejich jednotlivému využití k signalizaci různých stavů (kontrolky). Dále umožňuje nastavit libovolné LED diodě libovolnou barvu (teoreticky maximálně 512 odstínů) nebo každou diodu rozblikat libovolnou frekvencí.

Jak již bylo v kapitole 2.2.13 uvedeno, musí být míchání barev z jednotlivých složek (RGB) prováděno multiplexováním, což je vysvětleno v hlavní funkci *LedLight*. Tato a další důležité funkce jsou popsány v následujícím textu.

**void IO_LED_Init (void)**

Provede inicializaci veškerých pinů resp. portů mikroprocesoru, které se k řízení LED využívají.

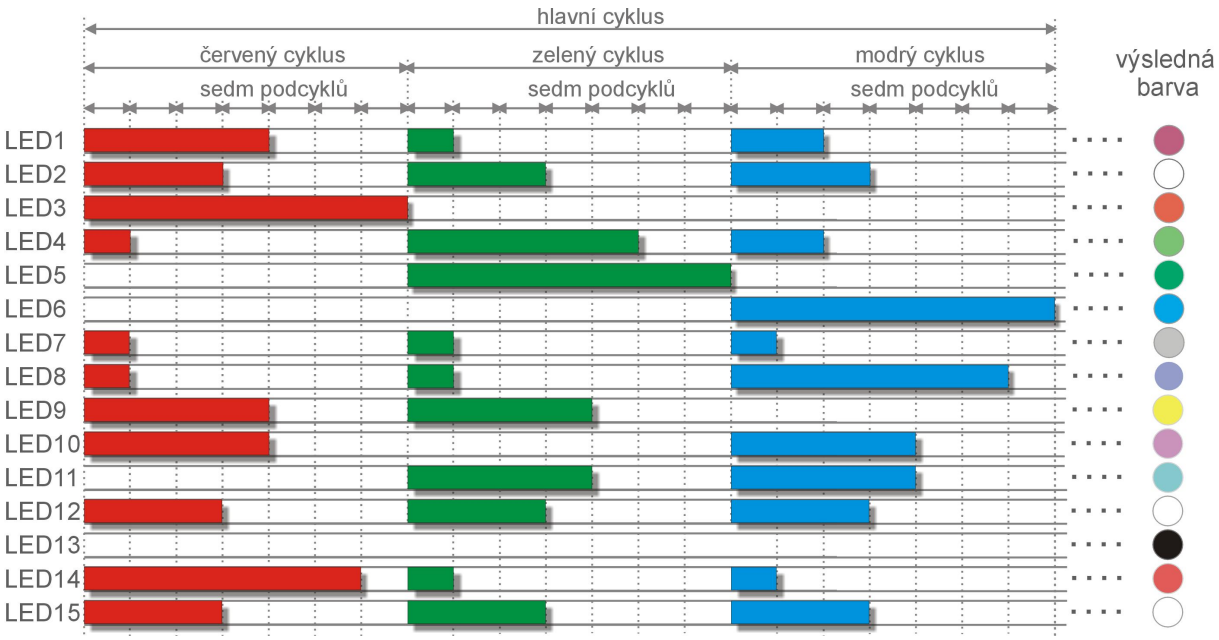
**void LedLight (void)**

Funkce slouží obnovování a míchání požadovaných barev. Volá se pomocí přerušení od časovače každých 70µs, aby nebylo znatelné blikání diod v rámci multiplexu. Informace podle kterých se reguluje barva se berou z dvourozměrného pole *CLED*. Toto pole má 15 řádků (pro 15 LED) a tři sloupce, kde každý je pro jednu barvu z RGB. Hodnoty pole mohou být v rozsahu 0 až 7, přičemž 0 znamená, že tato barva nebude ve výsledné obsažena. Hodnota 7 určuje maximální možný podíl příslušné barvy.

Princip použitého multiplexu je naznačen na Obr. 4.3. v souvislosti s Tab. 4.1, která představuje možný obsah pole *CLED*.

| | Červená | Zelená | Modrá |
|---|---|---|---|
| LED1 | 0x04 | 0x01 | 0x02 |
| LED2 | 0x03 | 0x03 | 0x03 |
| LED3 | 0x07 | 0x00 | 0x00 |
| LED4 | 0x01 | 0x05 | 0x02 |
| LED5 | 0x00 | 0x07 | 0x00 |
| LED6 | 0x00 | 0x00 | 0x07 |
| LED7 | 0x01 | 0x01 | 0x01 |
| LED8 | 0x01 | 0x01 | 0x06 |
| LED9 | 0x04 | 0x04 | 0x00 |
| LED10 | 0x04 | 0x00 | 0x04 |
| LED11 | 0x00 | 0x04 | 0x04 |
| LED12 | 0x03 | 0x03 | 0x03 |
| LED13 | 0x00 | 0x00 | 0x00 |
| LED14 | 0x06 | 0x01 | 0x01 |
| LED15 | 0x03 | 0x03 | 0x03 |

Tab. 4.1: Ukázka možného zaplnění pole *CLED*




Obr. 4.3: Princip míchání barev u RGB LED diod

**void BarGraph1 (u08 number)**

Tato funkce umožňuje zobrazování ve stylu sloupcového bargraphu. Nastavuje obsah *CLED* tak, aby rozsvícený počet diod odpovídal hodnotě uložené v řídícím registru (část „F“ 70. Byte). Hodnota je funkci předávána parametrem *numer*. Podle této hodnoty nastaví odpovídajícímu počtu řádků v *CLED* barvu, která je obsažena v řídícím registru (část „B“). Následující řádky zaplní nulami.

**void BarGraph2 (u08 number)**

Již z názvu je patrné, že tato funkce má určitou souvislost s předcházející. *BarGraph2* se liší pouze ve způsobu zobrazení požadované hodnoty, což je realizováno formou bodu pohybujícím se po stupnici. Funkce nastaví barvu pouze LED diodě s pořadovým číslem shodným s hodnotou v *numer*. Požadovaná barva se opět načte z řídícího registru. Ostatní řádky pole *CLED* se vynulují.

**void Indicator (void)**

Funkce se využívá v režimu kontrolek. Obnovuje dvourozměrné pole *ELED*, které stanovuje, zda má konkrétní LED dioda svítit či ne. Hodnoty tohoto pole určuje na základě cyklické dekrementace hodnot uložených v řídícím registru v oddílech „C“ a „D“. Má-li konkrétní dioda podle *ELED* svítit, nastaví do *CLED* požadovanou barvu. Nemá-li v danou dobu svítit, je příslušný řádek v *CLED* vynulován. Tento princip umožňuje nezávislé blikání každé diody jinou frekvencí.

**void RGB (u08 rgb)**

Nastavuje výstupní port mikroprocesoru tak, aby byla aktivní pouze jedna barevná větev. Požadovaná barva je obsažena v parametru *rgb*.

**void NoLightAllLed (void)**

Funkce, která vynuluje celé pole *CLED* a tím způsobí zhasnutí všech LED diod.

**void SetOneLed (u08 number, u16 color)**

Tato funkce je volána většinou z výše uvedených funkcí. Umožňuje samotné nastavení požadovaného řádku v poli *CLED* tzn. nastavit konkrétní LED diodě požadovanou barvu.

**void SetELED (u08 row, u08 col, u08 value)**

**void DecELED (u08 row, u08 col)**

Tyto funkce umožňují práci s polem *ELED*, *SetELED* nastavuje hodnotu konkrétní buňce, *DecELED* tuto buňku dekrementuje.

### 4.3.4 Knihovna pro krokový motor

Funkce této knihovny umožňují, aby ručička krokového motoru sledovala hodnotu nastavenou v řídícím registru.

**void IO_Motor_Init (void)**

Provede inicializaci veškerých výstupních pinů mikroprocesoru, které se k řízení krokového motoru používají.

**void SetStep (u08 value)**

Vypočítá směr a počet kroků, které je třeba udělat, aby se dosáhlo požadované pozice v parametru *value*.

**void M_Step (void)**

Tato funkce ovládá motor v závislosti na počtu potřebných kroků, které byly vypočítány funkcí *SetStep*. Protože vyžaduje čekací smyčky a je volána pomocí přerušení, není řešena s použitím cyklů. Při každém jejím zavolání se inkrementuje stavová proměnná, která určuje, která část kódu se bude při následujícím zavolání vykonávat.

### 4.3.5 Ostatní knihovny

Jedinou významnou knihovnou, která nebyla napsána ale byla přejata, je knihovna pro komunikaci po sběrnici $I^2C$. Knihovna byla vytvořena Pascalem Stangem [21] a nabízí kompletní funkce pro řízení veškerých dějů na zmíněné sběrnici. Je využívána v původní verzi, nemusela být nijak modifikována.

### 4.3.6 Hlavní funkce Main()

Ve funkcí *Main* se po zavolání všech inicializačních funkcí rozběhne nekonečná smyčka. Během ní se vždy obnoví display, dopočtou kroky pro krokový motor a na základě módu nastaveného pro LED diody se volají potřebné funkce. Do běhu programu zasahují dvě přerušení. První, od časovače, způsobuje obnovení LED diod a krokování motoru. Druhé je vyvoláváno při přijetí dat po $I^2C$. V tomto přerušení se data uloží do řídícího registru.

## 4.4 Software pro spodní desku

Software pro spodní desku, narozdíl od softwaru horní desky, byl napsán pouze jako demonstrace funkčnosti základních částí spodního bloku. Software tedy nemá strukturu, která by zahrnovala obsluhu veškerých hardwarových částí. Je v něm využit pouze zlomek možností, které spodní deska poskytuje.

### 4.4.1 Popis demonstračního softwaru

Program spodní desky se skládá z počáteční inicializace a následné nekonečné smyčky. Tato smyčka je rozdělena do několika hlavních bloků, které se cyklicky opakují. Přepínání mezi nimi je řízeno jedním digitálním vstupem, jehož stav se ve všech částech smyčky neustále testuje. Pro konkrétnější představu je v následujícím textu stručně popsáno, jaké informace a z kterých zdrojů jsou získávány. Následně je uvedeno jaký zobrazovací prvek na horní desce byl pro interpretaci těchto hodnot použit. Díky tomu, že se v některých případech stejné informace zobrazují různými způsoby, je možné porovnat, jak je tento způsob pro hodnoty s určitým charakterem vhodný.

**První blok:**

Na stupnici z LED diod se objeví přechod od červené přes zelenou až po modrou barvu, což má ukázat schopnost využívat libovolnou barvu. Display zobrazí pouze uvítací pozdrav.

**Druhý blok:**

V této části je na displeji i na stupnici z LED diod zobrazováno napájecí napětí celého zařízení. To je umožněno díky připojení napájecího napětí na jeden vstup mikroprocesoru

s A/D převodníkem. Uvedené propojení je realizováno spojem přímo na desce plošných spojů, což bylo zmíněno již v kapitole 2.2.3 a 2.2.5. V této části je LED diodám nastaven režim *bargraph2*, což je zobrazování pomocí jednoho pohybujícího se bodu.

**Třetí blok:**

Tato část nastiňuje jednak možnost použití vnějšího přerušení a také využívá další A/D převodník. Na vstup s externím přerušením EXTINT3 je přiveden signál který obsahuje řádově stovky impulsů za sekundu. Zpracováváním tohoto signálu a následným přepočítáváním je simulován proces, při kterém jsou snímány zuby na klikovém hřídeli motoru. Při každém vyvolání přerušení se inkrementuje obsah proměnné a čtyřikrát za sekundu se dopočítávají otáčky. Tato hodnota se posílá jednak na krokový motor a také na display. Na stupnici z LED diod je zobrazena přepočítaná hodnota z A/D převodníku. V této fázi pracují LED diody v režimu *bargraph1*, to znamená, že zobrazují sloupcem rozsvícených elementů.

**Čtvrtý blok:**

Tento blok slouží převážně k demonstraci použití LED diod jako kontrolek. Je v něm demonstrována možnost trvalého rozsvícení kontrolky či signalizace blikáním. Současně jsou na LED diodách opět ukázány různé zobrazovatelné barvy.

V každém z uvedených bloků jsou volány funkce, které připravují data k odeslání po sběrnici. Převážně jde o připojení adresy, na kterou se bude v řídícím registru horní desky zapisovat. Další úlohou těchto funkcí je naformátování a přepočítání získaných hodnot před odesláním. Příkladem takové části kódu může být funkce:

**void SendTextToDisp (char* text, char* str)**

Vstupními parametry jsou dva ukazatele na řetězce. V proměnné *text* je odkaz na řetězec, který bude zobrazen na displeji, *str* je pointer na řetězec, do kterého bude celá zpráva před odesláním uložena. V této funkci se tedy k požadovanému textu pouze přidá adresa odpovídající zobrazení na displeji a výsledná posloupnost se odešle.

### 4.4.3 Použité knihovny

Vytvoření softwaru pro spodní desku nevyžadovalo napsání žádné speciální knihovny. Několik potřebných knihoven však bylo přejeto. Všechny pochází od Pascala Stanga [21]. Jejich obsah nebyl měněn, jsou zcela vyhovující i originální verzi. Konkrétně se jedná o knihovny pro externí přerušení, A/D převodník a komunikaci po $I^2C$. Všechny tyto knihovny poskytují snadnou inicializaci částí, pro které jsou určeny. Knihovna pro externí přerušení umožňuje jednoduchou konfiguraci a obsluhu přerušení. Funkce v knihovně pro A/D převodník nabízí příjemnou správu této periferie.

# 5 Závěr

Úkolem této práce byl návrh a realizace zařízení pro snímání fyzikálních veličin na spalovacím motoru. Konkrétní podoba zařízení byla zvolena jako „Palubní počítač pro motocykl“, což mělo vliv pouze na vzhled a celkové uspořádání, funkčnost se touto volbou nijak neovlivnila. Zařízení bylo navrženo velice univerzálně, aby bylo schopno při svém nasazení splňovat široké spektrum možných požadavků. Důraz byla kladen především na všestrannost, spolehlivost a minimální rozměry. Podařilo se vytvořit kompletní hardware a napsat veškerý software, který byl požadován. Zařízení bylo odzkoušeno a je plně funkční, je tedy možné konstatovat, že se podařilo úspěšně splnit veškeré požadavky.

V první části práce byl popsán návrh hardwaru. Byla uvedena kritéria podle kterých byly voleny součástky, zapojení jednotlivých bloků a použití speciálních integrovaných obvodů. Hardware obsahuje veškeré náležitosti, o kterých bylo od začátku uvažováno. Druhá část byla věnována výběru vhodných senzorů, které je možné při provozu zařízení využívat. V třetí části bylo pojednáno o návrhu vhodného softwaru.

Při testování zařízení se ukázalo, že některé předpokládané vlastnosti a funkce se částečně liší od reálných. Při práci s krokovým motor bylo zjištěno, že tento typ nedisponuje příliš jemným krokem, proto je pohyb ručičky znatelně skokový a rozlišení hodnot, které dokáže zobrazovat, není velké. Tento drobný nedostatek lze vyřešit použitím jiného „preciznějšího“ krokového motoru. Další věcí, která nebyla tak ideální jak se předpokládalo, je míchání barev v GRB LED diodách. Při pohledu na diodu jsou částečně patrná samostatná centra záření jednotlivých barev. Pro dokonalejší skládání složek spektra je třeba před diodu umístit rozptylovací filtr.

Je třeba také uvést, že tato práce se neobešla bez drobných omylů. Během návrhu vznikly dvě nezávažné chyby na desce plošných spojů (jedna byla zapříčiněna nedokonalostí návrhového systému), které byly dodatečně odstraněny.

## Seznam použitých zdrojů


[1] ATMEL: <http://www.atmel.com/>

[2] HW server: <http://www.hw.cz/>

[3] AT90CAN128 - AVR microcontroler, katalogový list, Rev. 4250H–CAN–05/06, <http://www.atmel.com/>

[4] ATmega16 - AVR microcontroler, katalogový list, Rev. 2466E–AVR–10/02, <http://www.atmel.com/>

[5] LM2575 - 52kHz Simple 1A Buck Regulator, katalogový list, M9999-042205, <http://www.micrel.com/>

[6] AT45DB041B - 4-megabit DataFlash, katalogový list, Rev. 1938F–DFLSH–10/02, <http://www.atmel.com/>

[7] MAX232 - Multichannel RS-232 Drivers/Receivers, katalogový list, 19-4323; Rev 10; 8/01, <http://www.maxim-ic.com/>

[8] PCA82C250 - CAN controller interface, katalogový list, 2000 Jan 13; 9397 750 06609 <http://www.semiconductors.philips.com/>

[9] DS1302 - Trickle-Charge Timekeeping Chip, katalogový list, REV: 110805, <http://www.maxim-ic.com/>

[10] HCMS-29xx - High Performance CMOS Alphanumeric Displays, katalogový list, July 14. 2004; 5988-4161EN, <http://www.agilent.com/semiconductors/>

[11] L6208 - DMOS Driver For Bipolar Stepper Motor, katalogový list, September 2003, <http://www.st.com/>

[12] MPC120 - Microcontroller Supervisory Circuit with Open Drain Output, katalogový list, 01/30/01; DS11184D, <http://http://www.microchip.com/>

[13] LE33 - Very Low Drop Voltage Regulator, katalogový list, Rev. 6; July 2004, <http://www.st.com/>

[14] MPX4115AD - Integrated Silicon Pressure Sensor, katalogový list, REV 4. 2001, <http://www.motorola.com/semiconductors/>

[15] LM135 - Precision Temperature Sensor, katalogový list, October 1997, <http://www.st.com/>

[16] 1GT101 - Hall Effect Gear Tooth Sensors, katalogový list, <http://www.sstsensing.com/>

[17] GM Electronic - Katalog součástek pro elektroniku: <http://www.gme.cz/>

[18] WinAVR: <http://winavr.sourceforge.net/>

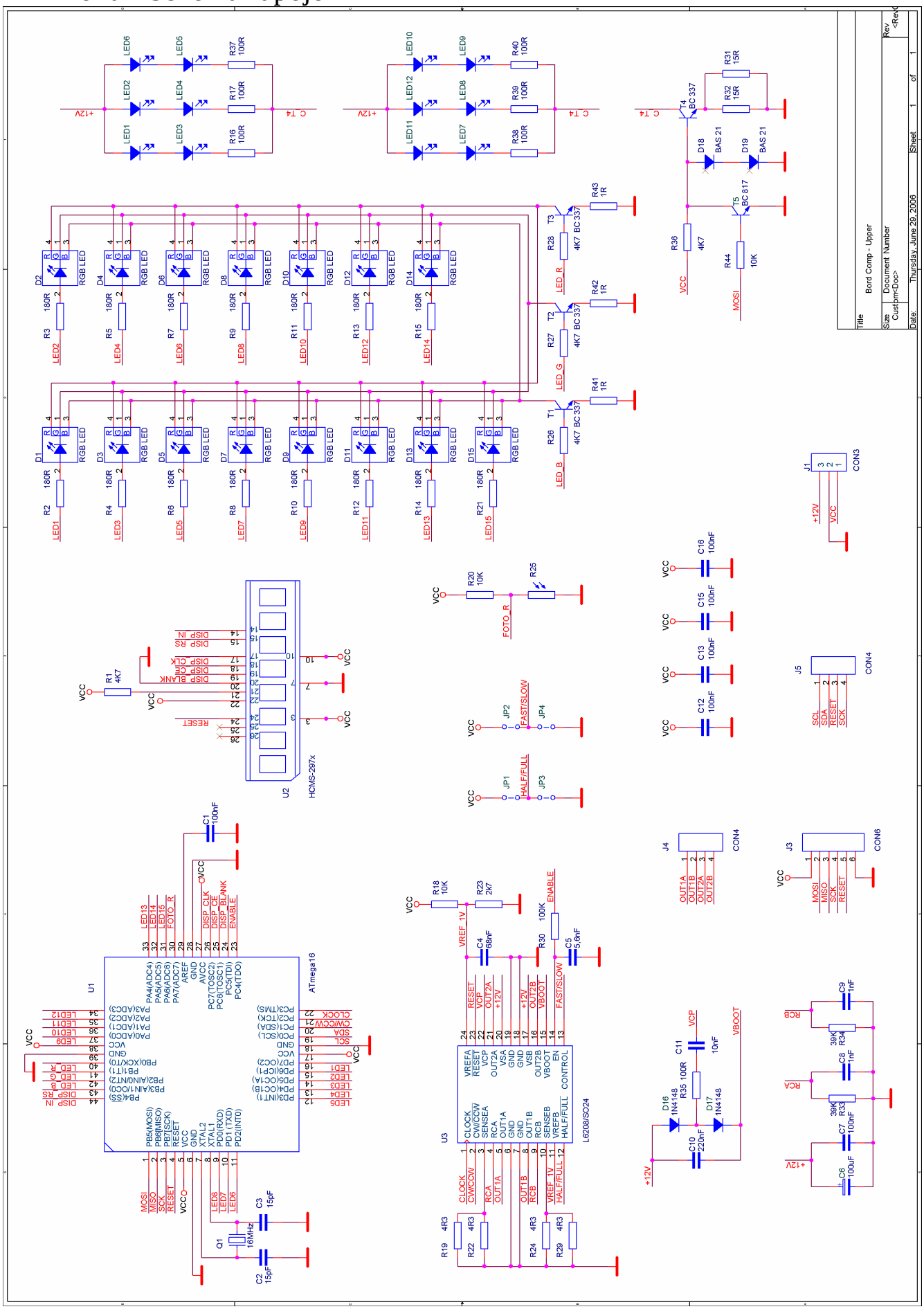
[19] PonyProg: <http://www.lancos.com/>

[19] GNU vývojový řetězec a m68332: <http://cmp.felk.cvut.cz/~pisa/>

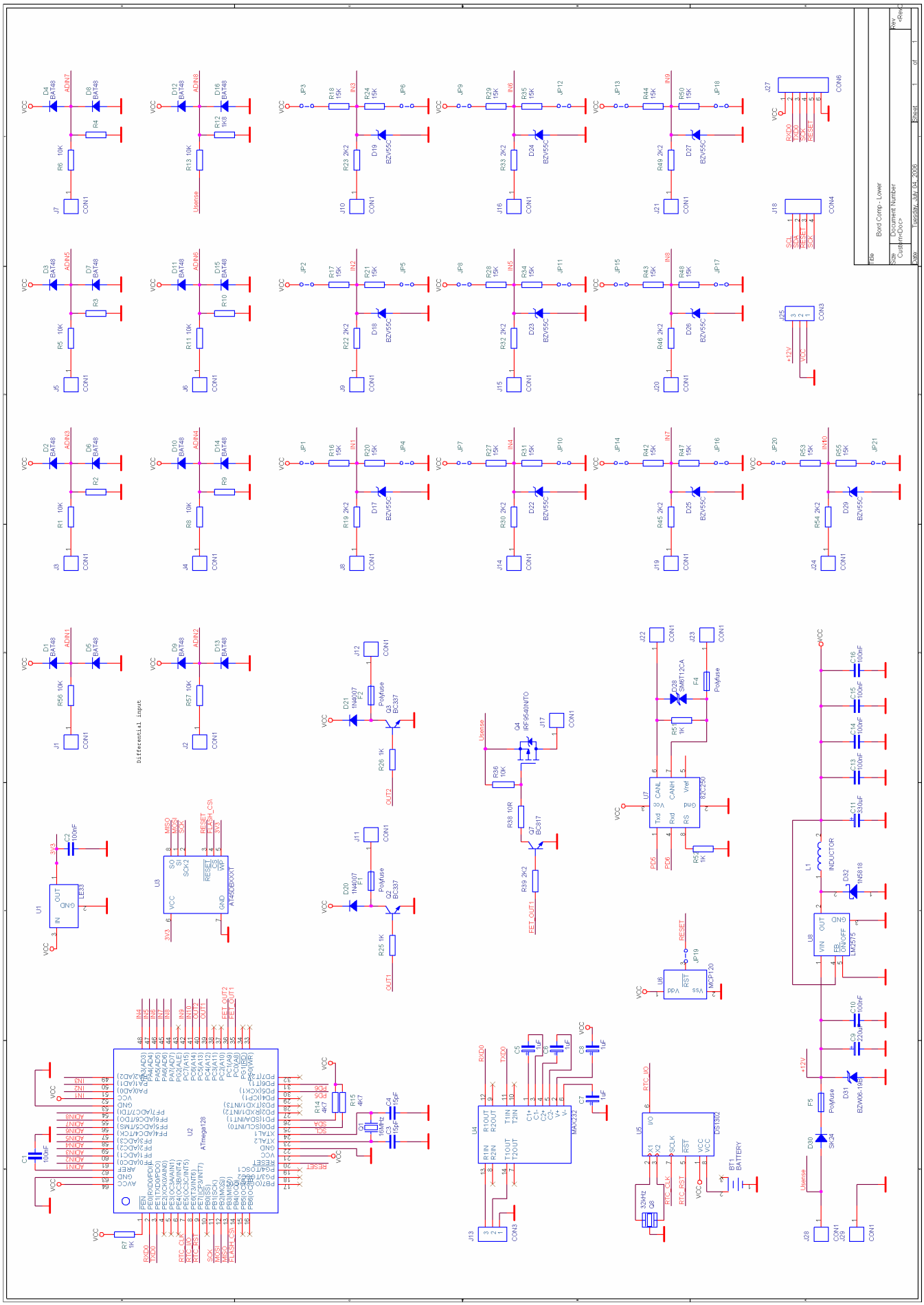
[20] Sdělovací technika - Zajímavé aplikace mikrokontrolérů AVR firmy Atmel: < http://www.stech.cz/>

[21] Procyon AVRlib: <http://hubbard.engr.scu.edu/embedded/avr/avrlib/>

# A Příloha – Schéma zapojení

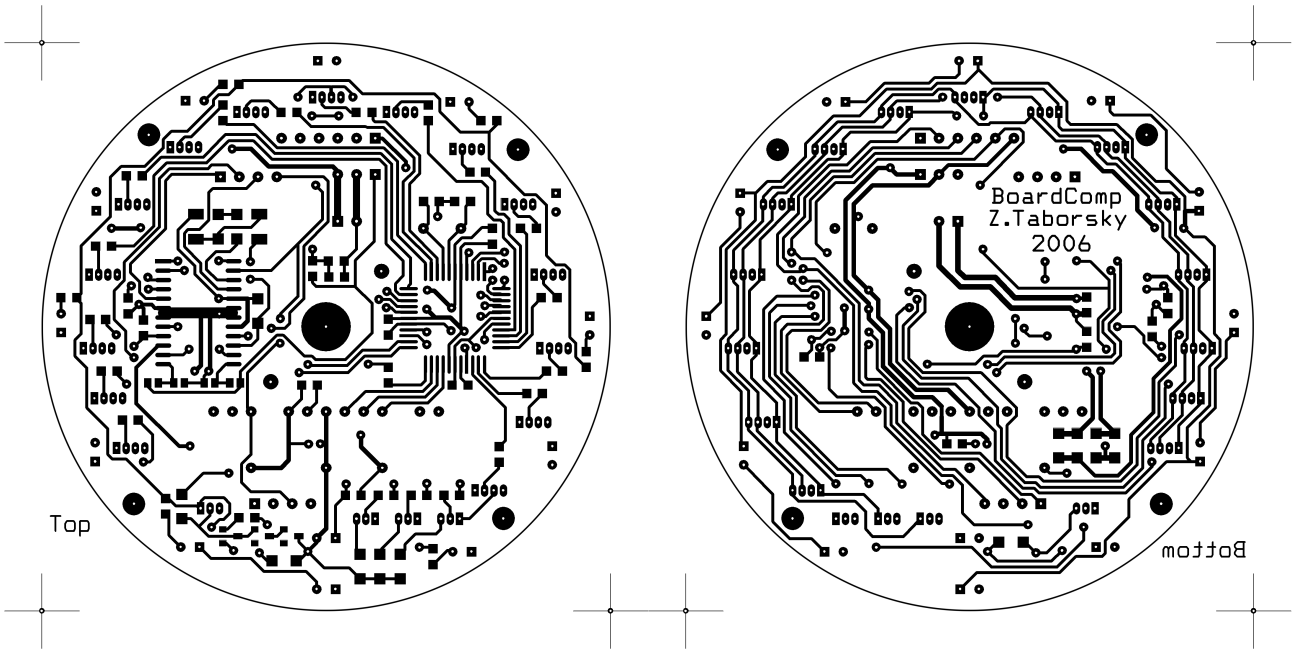



Obr. A.1: Schéma zapojení horní desky

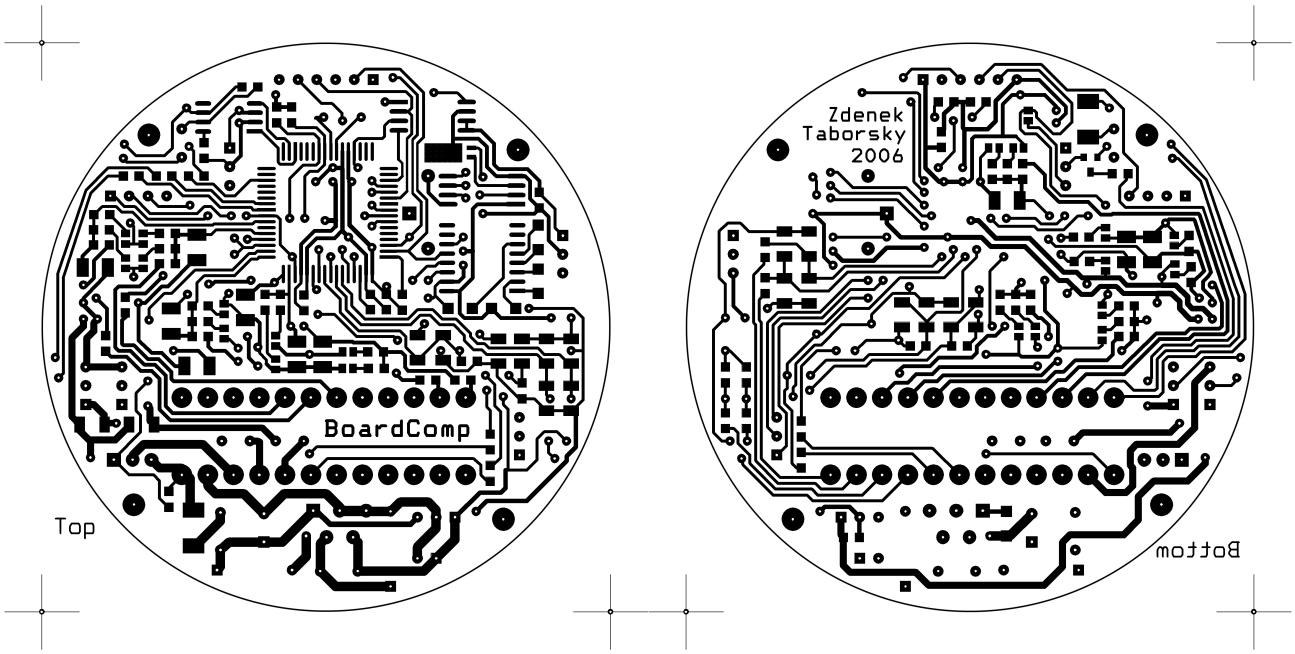



Obr. A.2: Schéma zapojení spodní desky

# B Příloha – Desky plošných spojů

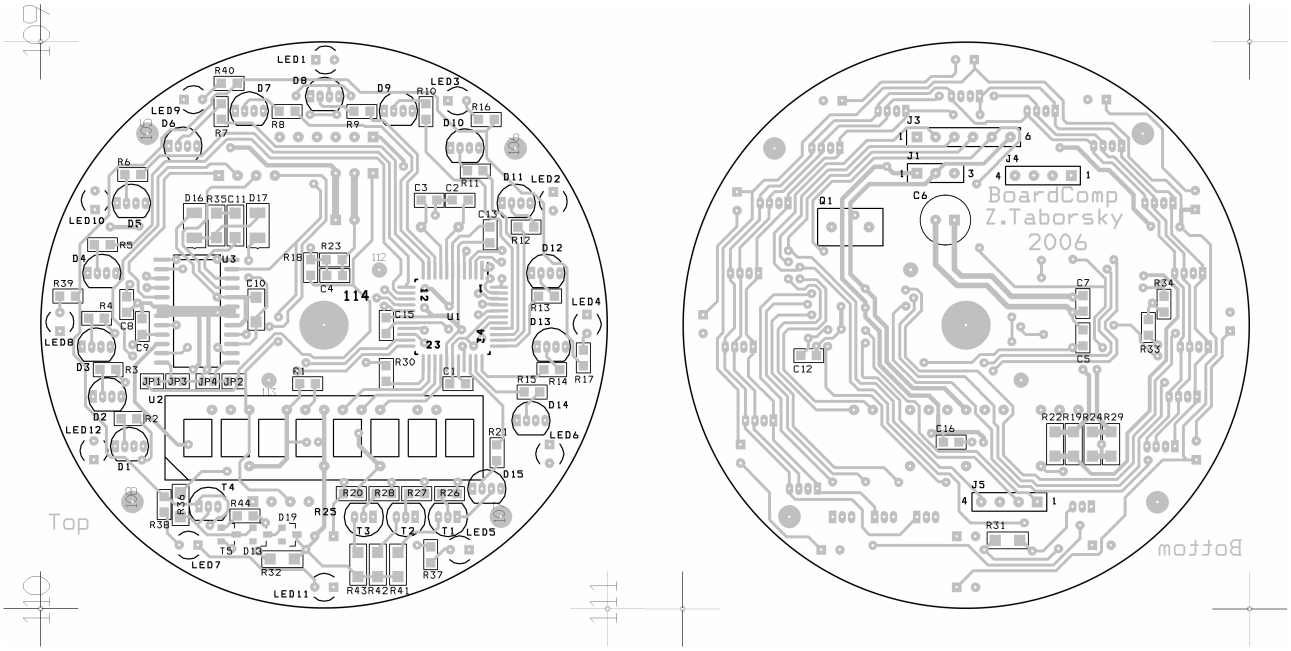



Obr. B.1: Horní deska plošných spojů – obě strany

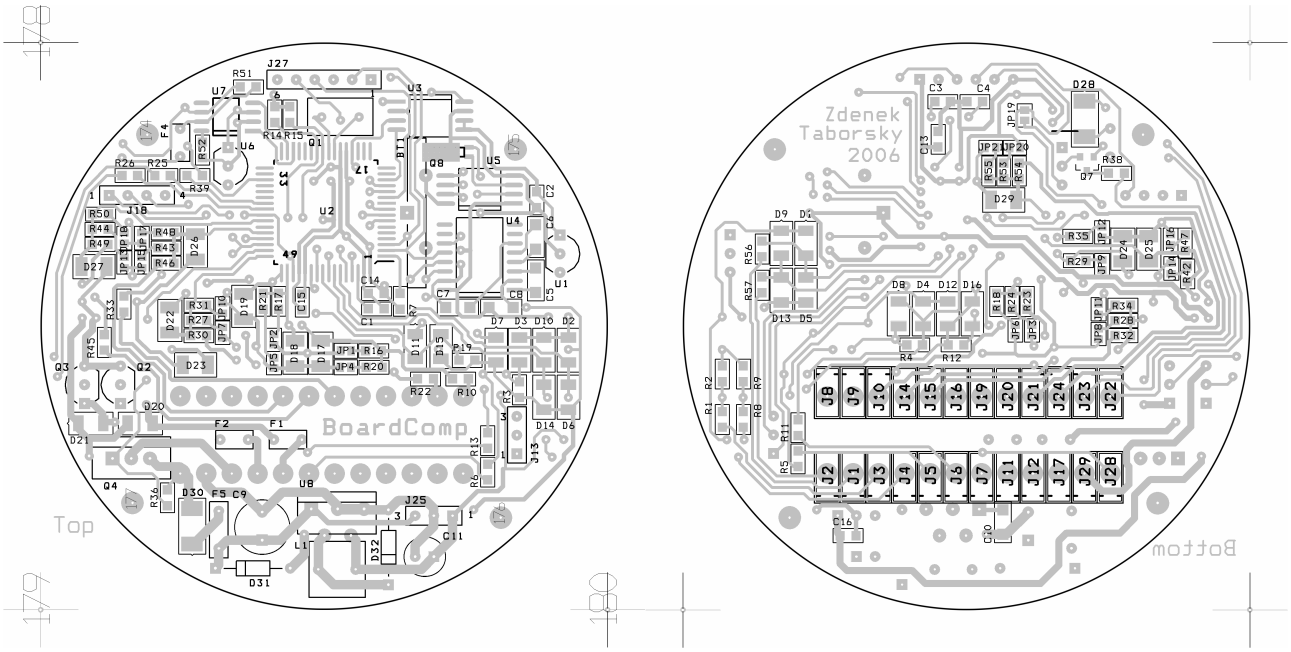



Obr. B.2: Spodní deska plošných spojů – obě strany

# C Příloha – Rozmístění součástek




Obr. C.1: Rozmístění součástek na horní desce plošných spojů – obě strany




Obr. C.2: Rozmístění součástek na spodní desce plošných spojů – obě strany

# D Příloha – Seznam použitých součástek

| Označení | Hodnota | Popis | Počet |
|---|---|---|---|
| R1, R26 - R28 | 4K7 | Rezistor, SMD 0805 | 4 |
| R2 - R15, R21 | 180R | Rezistor, SMD 0805 | 15 |
| R16, R17, R37 - R40 | 100R | Rezistor, SMD 0805 | 6 |
| R18, R20, R44 | 10K | Rezistor, SMD 0805 | 3 |
| R19, R22, R24, R29 | 4R3 | Rezistor, SMD 1206 | 4 |
| R23 | 2k7 | Rezistor, SMD 0805 | 1 |
| R25 | | Foto-rezistor | 1 |
| R30 | 100K | Rezistor, SMD 0805 | 1 |
| R31, R32 | 15R | Rezistor, SMD 1206 | 2 |
| R33, R34 | 39K | Rezistor, SMD 0805 | 2 |
| R35 | 100R | Rezistor, SMD 1206 | 1 |
| R36 | 4K7 | Rezistor, SMD 1206 | 1 |
| R41 - R43 | 1R | Rezistor, SMD 1206 | 3 |
| C1, C7, C12 - C16 | 100nF | Kondenzátor, SMD 0805 | 6 |
| C2, C3 | 15pF | Kondenzátor, SMD 0805 | 2 |
| C4 | 68nF | Kondenzátor, SMD 0805 | 1 |
| C5 | 5,6nF | Kondenzátor, SMD 0805 | 1 |
| C6 | 100µF | Elektrolitický kondenzátor, radiální | 1 |
| C8, C9 | 1nF | Kondenzátor, SMD 0805 | 2 |
| C10 | 220nF | Kondenzátor, SMD 1206 | 1 |
| C11 | 10nF | Kondenzátor, SMD 1206 | 1 |
| D1 - D15 | RGB LED | RGB LED dioda, 5mm | 15 |
| D16, D17 | 1N4148 | Dioda, SMD 1406 | 2 |
| D18, D19 | BAS 21 | Dioda, SMD SOT23 | 2 |
| LED1 - LED12 | LED | Bílá LED dioda, 3mm | 12 |
| Q1 | 16MHz | Krystal, HC49U/S | 1 |
| U1 | ATmega16 | Mikroprocesor, TQFP-44 | 1 |
| U2 | HCMS-2975 | LED display | 1 |
| U3 | L6208 | Budič krokového motoru, SO24 | 1 |
| T1 - T4 | BC 337 | NPN tranzistor, TO92 | 4 |
| T5 | BC 817 | NPN tranzistor, SOT23 | 1 |
| JP1- JP4 | JUMPER | Plošky pro SMD 0504 | 4 |
| J1 | CON3 | Konektor 3-pinový, 2,54mm | 1 |
| J4, J5 | CON4 | Konektor 4-pinový, 2,54mm | 2 |
| J3 | CON6 | Konektor 6-pinový, 2,54mm | 1 |

Tab. E.1: Seznam součástek použitých na horní desce

| Označení | Hodnota | Popis | Počet |
|---|---|---|---|
| R1, R5, R6, R8, R11, R13, R36, R39, R56, R57 | 10K | Rezistor, SMD 0805 | 10 |
| R2 - R4, R7, R9, R10, R25, R26, R51, R52 | 1k | Rezistor, SMD 0805 | 10 |
| R12 | 1K8 | Rezistor, SMD 0805 | 1 |
| R14, R15 | 4K7 | Rezistor, SMD 0805 | 2 |
| R16 - R18, R20, R21, R24, R27 - R29, R31, R34, R35, R42 - R44, R47, R48, R50, R53, R55, | 15K | Rezistor, SMD 0805 | 20 |
| R19, R22, R23, R30, R32, R33, R45, R46, R49, R54 | 2K2 | Rezistor, SMD 0805 | 10 |
| R38 | 10R | Rezistor, SMD 0805 | 1 |
| C1, C2, C13 - C16 | 100nF | Kondenzátor, SMD 0805 | 6 |
| C3, C4 | 15pF | Kondenzátor, SMD 0805 | 2 |
| C5 - C8 | 1µF | Kondenzátor, SMD 1206 | 4 |
| C9 | 220µF | Elektrolitický kondenzátor, radiální | 1 |
| C10 | 100nF | Kondenzátor, SMD 1206 | 1 |
| C11 | 330µF | Elektrolitický kondenzátor, radiální | 1 |
| D1 - D16 | BAT48 | Schottkyho dioda, SMD 1406 | 16 |
| D17 - D19, D22 - D27, D29 | BZV55C | Zenerova dioda 4,7V, SMD 1210 | 10 |
| D20, D21 | 1N4007 | Dioda, SMD 1406 | 2 |
| D28 | SM6T12CA | BIpolární transil 12V | 1 |
| D30 | SK24 | Dioda, SMD DO213 | 1 |
| D31 | BZW06-19B | Unipolární transil 22V | 1 |
| D32 | 1N5818 | Schottkyho dioda | 1 |
| Q1 | 16MHz | Krystal, HC49U/S | 1 |
| Q2, Q3 | BC337 | NPN tranzistor, TO92 | 2 |
| Q4 | IRF9540N | MOSFET tranzistor, TO220 | 1 |
| Q7 | BC817 | NPN tranzistor, SOT23 | 1 |
| Q8 | 32kHz | Krystal, 2x6mm | 1 |
| U1 | LE33 | Stabilizátor 3,3V, TO92 | 1 |
| U2 | ATmega128 | Mikroprocesor, TQFP-64 | 1 |
| U3 | AT45DB041B | DataFlash 4MB, 8-SOIC | 1 |
| U4 | MAX232 | Konvertor RS232, SO-16 | 1 |
| U5 | DS1302 | Obvod reálného času, 8-SOIC | 1 |
| U6 | MCP120 | Podpěťový resetovací obvod TO92 | 1 |
| U7 | 82C250 | Budič sběrnice CAN, SO-8 | 1 |
| U8 | LM2575 | Spínaný zdroj 5V, TO220(T) | 1 |
| F1, F2 | 0,5A | Vratná pojistka 0,5A | 2 |
| F4 | 0,1A | Vratná pojistka 0,1A | 1 |
| F5 | 1,1A | Vratná pojistka 1,1A | 1 |
| L1 | 330µH | Radiální tlumivka | 1 |
| BT1 | 180mAh | Baterie CR2025 | 1 |
| JP1 - JP21 | JUMPER | Plošky pro SMD 0504 | 21 |
| J1 - J12, J14 - J17, J19 - J24, J28, J29 | CON1 | Konektor 1-pinový, 3,5mm | 24 |
| J13, J25 | CON3 | Konektor 3-pinový, 2,54mm | 2 |
| J18 | CON4 | Konektor 4-pinový, 2,54mm | 1 |
| J27 | CON6 | Konektor 6-pinový, 2,54mm | 1 |

Tab. E.2: Seznam součástek použitých na spodní desce

# E Příloha – Obsah přiloženého CD

Přiložené CD obsahuje text této bakalářské práce a veškeré materiály s ní spojené. Data jsou členěna do těchto adresářů:

/**bp**/ -obsahuje tuto práci ve formátu PDF

/**katalogove_listy**/ - obsahuje veškeré katalogové listy a dokumentaci

/**orcad**/ - obsahuje veškeré materiály pro návrh a výrobu desek plošných spojů, které vznikly v návrhovém systému OrCad

/**zdrojove_kody**/ - obsahuje veškeré zdrojové kódy v jazyce C, které byly pro demonstraci funkčnosti zařízení napsány:

/**zdrojove_kody**/**atmega16**/ - zdrojové kódy pro mikroprocesor ATmega16

/**zdrojove_kody**/**atmega128**/ - zdrojové kódy pro mikroprocesor ATmega128

/**programy**/ - obsahuje editor JFE, v kterém byly napsány zdrojové kódy, programátor PonyProg, kompilátor GCC v balíku WinAvr

/**obrazky**/ - obsahuje veškeré použité obrázky ve formátu PNG

/**zaloha**/ - obsahuje komprimovanou kopii všech dat na tomto CD