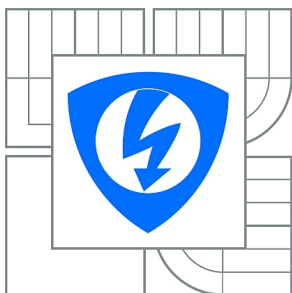




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ELEKTRONICKÝ MONITOROVACÍ SYSTÉM PRO MOTOROVÁ VOZIDLA

ELECTRONIC MONITORING SYSTEM FOR MOTOR VEHICLES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN POSPÍŠIL

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ALEŠ PROKEŠ, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Martin Pospíšil

ID: 106729

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Elektronický monitorovací systém pro motorová vozidla

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte architekturu monitorovacího systému pro motorová vozidla. Od systému je požadováno měření rychlosti vozu, otáček motoru, teploty motoru, teploty vně i uvnitř vozu a dalších veličin. Součástí systému řízeného mikrokontrolérem by měl být GPS modul pro určování polohy, nadmořské výšky, rychlosti a aktuálního času. Kromě řídicí části a senzorů je požadováno použití vhodné zobrazovací jednotky a rozhraní pro ukládání dat.

Realizujte hardware navrženého systému a ověřte jeho funkčnost. Vytvořte softwarové moduly - knihovny pro obsluhu jednotlivých součástí systému tak, aby z nich mohl být snadno sestavován software pro obsluhu systému s různými konfiguracemi periférií.

Vytvořte software pro kompletní obsluhu realizovaného hardware a ověřte jeho funkčnost. Navrhněte a realizujte ukládání naměřených dat do počítače a jejich základní vyhodnocení.

DOPORUČENÁ LITERATURA:

[1] MANN, B. C pro mikrokontroléry, Praha: BEN - technická literatura, 2003.

[2] KREIDL, M. Měření teploty - senzory a měřicí obvody, Praha: BEN - technická literatura, 2005.

Termín zadání: 6.2.2012

Termín odevzdání: 18.5.2012

Vedoucí práce: prof. Ing. Aleš Prokeš, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá návrhem monitorovací jednotky pro sledování provozních veličin ve vozidle. Tato jednotka je vybavena mikrokontrolérem, grafickým displejem, GPS modulem a SD kartou. Pro měření otáček podporuje vícero variant připojení na vozidlo. Teplota je měřena několika senzory DS18B20, snímající teplotu motoru a vnitřní a vnější teplotu. GPS modul poskytuje rychlost a nadmořskou výšku. Naměřené údaje je možné ukládat na SD kartu a prohlížet na PC.

KLÍČOVÁ SLOVA

Mikrokontrolér, STM32, grafický displej, SD karta, DS18B20, GPS.

ABSTRACT

This thesis is about construction of a monitoring unit for motor vehicles. The unit is equipped with microcontroller, graphical LCD for displaying all measured values, GPS module and SD card slot. For measurement of motor RPM, various methods of connecting to the motor are supported. Temperature is measured with few DS18B20 sensors, picking up engine temp, outsider and inside temp. GPS module is for measuring speed and height above MSL. All measured values can be recorded on SD card for future analysis on PC.

KEYWORDS

Microcontroller, STM32, graphic LCD, SD card, DS18B20, GPS.

POSPÍŠIL, M. *Elektronický monitorovací systém pro motorová vozidla*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2012. 33 s. Diplomová práce. Vedoucí práce: prof. Ing. Aleš Prokeš, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Elektronický monitorovací systém pro motorová vozidla jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce prof. Ing. Aleši Prokešovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

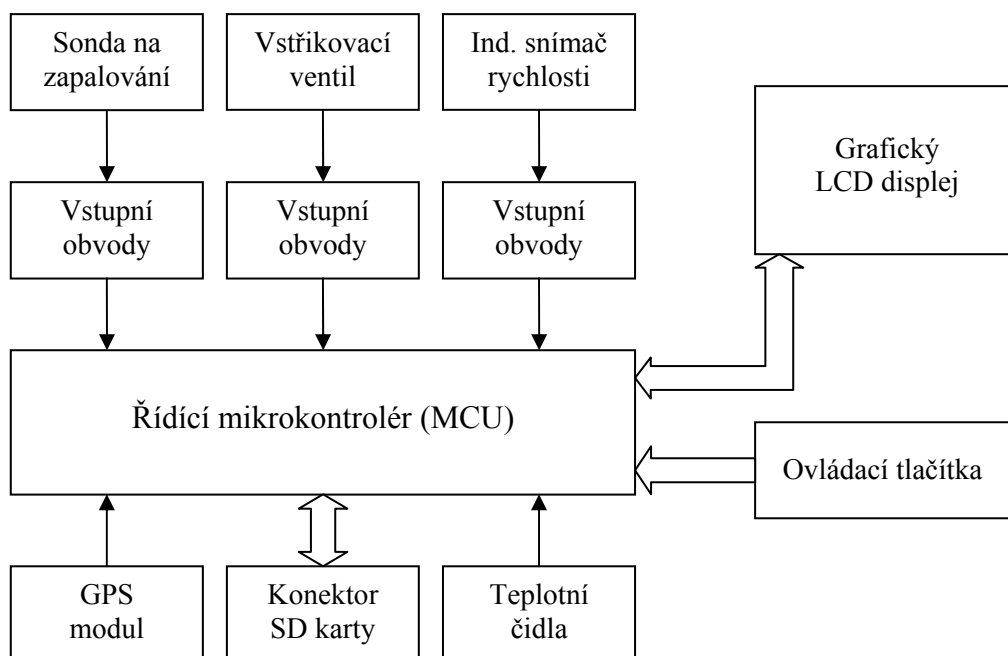
Úvod	1
1 Návrh architektury	2
1.1 Základ systému	2
1.2 Měření otáček motoru	5
1.2.1 Snímání otáček na zapalovací soustavě	5
1.2.2 Snímání otáček na vstřikovacím ventilu	6
1.2.3 Snímání otáček motoru s využitím alternátoru	6
1.2.4 Snímání otáček z čidla polohy klikového hřídele	7
1.3 Měření rychlosti vozu	7
1.4 Snímače teploty	8
1.5 GPS modul	9
1.6 Rozhraní pro ukládání dat	10
2 Hardware	11
2.1 Popis zapojení	11
2.2 Tvarovač signálu ze zapalování	14
2.3 Mechanická konstrukce jednotky	16
3 Software	19
3.1 Modul lcd.c	19
3.2 Modul gps.c	21
3.3 Modul DS18B20.c	22
3.4 Obsluha tlačítek	25
3.5 Hlavní program (main)	26
3.6 Záznam na SD kartu	28
Závěr	32
Použité zdroje	33

ÚVOD

Tato práce se zabývá vytvořením zobrazovací a monitorovací jednotky pro automobily. Je určena pro vozidla, která nejsou vybavena palubním displejem, sloužit může však v libovolném vozidle pro záznam provozních údajů. Základem je poměrně přizpůsobitelný hardware, jehož možnosti rozšíří konfigurovatelný firmware. Návrh zařízení je podřízen nízké ceně a dostupnosti jednotlivých komponent. Účelem zařízení je přehledně zobrazovat všechny důležité provozní hodnoty vozidla během jízdy, jako jsou otáčky, rychlost, teplota chladící kapaliny a zaznamenávat je pro další zpracování. Součástí práce je vytvoření hardwaru a firmwaru.

1 NÁVRH ARCHITEKTURY

Tato část se zabývá teoretickým návrhem hardware, připojením snímačů k jednotce a ovládáním periférií. Blokové schéma systému je na Obr. 1.



Obr.1: Blokové schéma systému

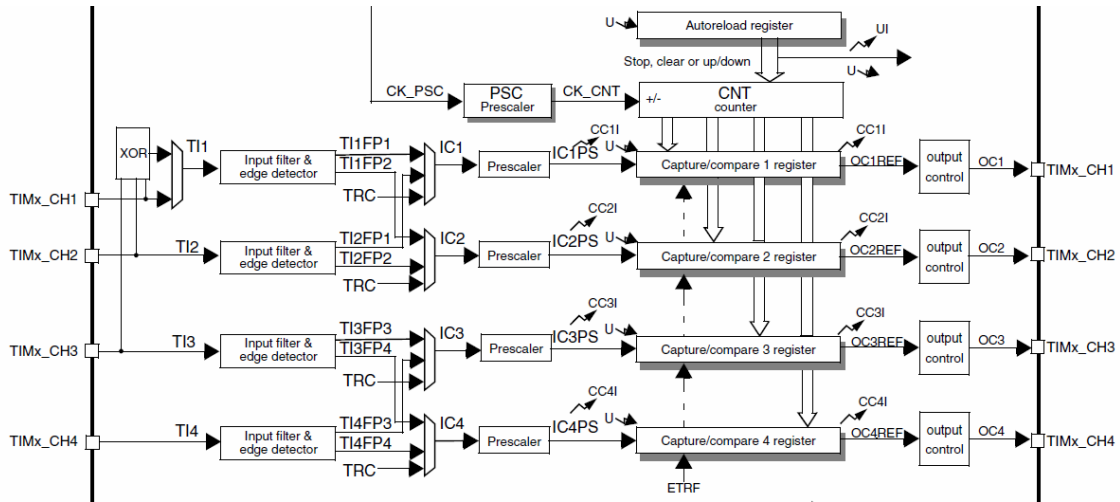
1.1 Základ systému

Jako řídicí MCU byl od začátku uvažován 8-bitový mikrokontrolér. Zvolen byl Atmel AVR, z důvodu dostupnosti, obsahuje potřebné periferie a má jednoduchou instrukční sadu, konkrétně řada ATMEGA. Procesor má dostatek uživatelských pinů pro připojení displeje, tlačítek a snímačů. Už během vývoje knihovny pro LCD displej se však ukázalo, že programovat 8-bit procesory AVR v jazyce C není vhodné pro větší projekty. Velikost kódu (a s tím souvisí i rychlost provádění) narůstala velmi rychle. Jelikož procesory AVR s větší programovou pamětí (např. ATMEGA64) jsou už poměrně drahé – cca 170Kč, byl výběr procesoru přehodnocen.

Skoro všechny funkce jednotky je možné poměrně snadno implementovat do 8-bit prostředí, až na jednu. Problematickou se jeví obsluha souborového systému FAT32, kde by bylo výhodnější použít více než osmibitový procesor. Po prozkoumání situace na trhu byl zvolen procesor STM32F100 z řady value line. Dostupnost v Česku je sice horší ale cena kolem 105Kč za kus je podstatná výhoda.

Pro snímání otáček a doby otevření vstřikovacího ventilu bude využit některý z 16-ti bitových časovačů. STM32F100 poskytuje celkem 7 časovačů, což je výhodné i z hlediska plánování pinů na desce plošných spojů. Časovače TIM1 až TIM4 poskytují pokročilé funkce, ostatní časovače jsou jednodušší. Hodinový signál prochází přes konfigurovatelnou předděličku, kde je možné nastavit dělení v rozsahu 1-65535. Časovače umožňují záchytné funkce, PWM generaci, změnu směru čítání, auto-reload, a spolupráci s DMA řadičem.

Blokové schéma části input capture časovačů je na Obr.2.



Obr.2 Blokové schéma input capture jednotky, převzato z [3].

Vstupní signál je přiveden na některý z vstupních pinů, např. TI2, který je poté zaveden do vstupu TIMx_CH2. Vstupní detektor hran je nastaven aby náběžná hrana spouštěla zachycení do CCR2 a sestupná do CCR1. Dále je využit slave-controller (není na obrázku) který na událost zachycení CCR2 provede vynulování čítače. Toto uspořádání umožňuje měřit současně jak periodu tak střidu, a to jedním vstupem (pinem). V registru CCR1 bude délka pulzu, v registru CCR2 pak celková perioda. Doba mezery se dá dopočítat jako CCR2-CCR1. Tato hodnota je doba otevření vstřikovacího ventilu, a může být později využita k indikaci okamžité spotřeby.

Výběr hodnoty do předděličky PSC je kompromis. Omezující faktor shora je přesnost měření při vysokých otáčkách, zdola zase přetečení čítače při nízkých otáčkách. Byla zvolena hodnota 240, která při vstupním hodinovém kmitočtu 24MHz vytvoří takt 100kHz pro čítač. Rozlišení čítače je tedy 10us. Maximální měřitelná perioda (přetečení čítače) je tedy 655.35ms což definuje minimální vstupní kmitočet 1.53Hz. Tomu odpovídá 46 ot/min pro čtyřtaktí čtyřválec (dva impulsy na otáčku). Maximální měřitelný počet otáček je mnohem více než 20000ot/min, takže pro použití na běžných motorech bez problémů. Směrem k vyšším otáčkám rozlišovací schopnost otáček klesá, což je dáno principem metody měření periody.

Hodnota 10us na takt vyhoví i z hlediska měření doby otevření vstřikovacího ventilu, v praxi je tato hodnota kolem 2 až 30ms, takže rozlišení by mělo být dostatečné pro odhad okamžité spotřeby.

Uživatelské rozhraní tvoří grafický LCD displej a několik tlačítek pro pohyb v menu. Jako displej bude použit modul s rozlišením 128x64, s řadičem KS0108 (např. MG12864 a kompatibilní). Tento displej je poměrně dostupný a za přijatelnou cenu. Nevýhoda je, že tento řadič je čistě grafický, tj. přímé ovládání jednotlivých bodů. Pro vykreslení textu je tedy nutné vytvořit znakovou sadu, znamená to také vyšší zátěž procesoru. Vlastní znaky také budou zabírat část programové paměti jako konstanty. Pro jízdu v noci je uvažováno o automatické regulaci intenzity podsvícení displeje.

Ovládání displeje velmi připomíná často používané znakové displeje s kontrolérem HD44780. Jelikož je plocha displeje rozdělena na dvě poloviny 64x64, jsou vyvedeny dva signály CS1 a CS2, které vybírají se kterou polovinou se pracuje. Vodič RW řídí směr komunikace. Principiálně není nutné z displeje číst, řadič však umožňuje jednak vyčítat stavový registr, tak také data z paměti pixelů. Datová sběrnice je 8-mi bitová, navíc s řídicím signálem RS, který určuje zda se přenáší instrukce (RS=0) a nebo data k zobrazení (RS=1). Vlastní přepis dat proběhne na sestupnou hranu signálu E.

Displej je organizován jako 64 sloupců x 8 stránek po 8 bitech. Adresu sloupce určuje registr Y. Při zápisu dat se automaticky inkrementuje a tak zjednodušuje obsluhu. Aktuální stránku obsahuje registr X. Jedná se vlastně o sdružení osmi řádků, do kterých se zapisuje datový bajt svisle. Registr X se neinkrementuje a je nutné jej obsloužit manuálně, viz Tab. 1.

Tab. 1: Instrukční sada řadiče KS0108, převzato z [1]

Instrukce	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Funkce
Načít data	1	1	Načtená data								Zpětně načte data z RAM displeje
Zapsat data	1	0	Zapisovaná data								Zapíše data do RAM displeje, inkrementuje Y registr.
čtení stavu	0	1	Busy	0	ON/OFF	Re-set	0	0	0	0	Načte aktuální stav Busy 0: připraven 1: probíhá operace ON/OFF 0: displej zapnut 1: displej vypnut Reset 0: normal 1: reset
Nastavit adresu Y registru	0	0	0	1	Y adresa (0~63)						Nastaví adresu sloupce
Nastavit adresu prvního řádku	0	0	1	1	Číslo prvního řádku (0~63)						Nastaví který řádek bude považován za první odshora.
Nastavit adresu X registru	0	0	1	0	1	1	1	Stránka (0~7)			Vybere osmici řádků (stránku)
Zapnout / vypnout displej	0	0	0	0	1	1	1	1	1	ON/OFF	0: vypnuto 1: zapnuto

1.2 Měření otáček motoru

Patří spolu s měřením rychlosti k základním sledovaným veličinám. Je nutné zajistit aby odezva otáčkoměru byla subjektivně dostatečně rychlá, proto se v praxi nepoužívá měření čítačem impulzů, ale měřením periody s následným přepočtem.

Signál o otáčkách je možné získat na více místech ve vozidle. Patří mezi ně zapalovací soustava, vstřikovací ventil paliva, alternátor, snímač polohy klikového a snímač polohy vačkového hřídele. Jednotlivé metody budou nyní podrobněji rozebrány.

1.2.1 Snímání otáček na zapalovací soustavě

Tento způsob je nejčastěji používaným způsobem připojení dodatečného otáčkoměru na benzinový motor. Principiálně je možné připojení jak na primární tak na sekundární stranu zapalovací cívky. Vzniklé signály se liší jen převodovým poměrem zapalovací cívky. Jelikož bývají zapalovací kabely snadno přístupné, je dobrým řešením snímání kapacitní vazbou. Na zapalovací kabel se umístí svorka nebo se jen omotá několik závitů drátu kolem kabelu. Malá kapacita mezi snímačem a zapalovacím kabelem v kombinaci se zatěžovacím rezistorem vytvoří potřebný dělicí poměr z vysokonapěťové části obvodu. Užitečný signál je úzká napěťová špička, za kterou následuje přechodový děj hoření výboje a dokmitávání systému. Touto špičkou je možné spouštět měření periody, je ovšem nutné potlačit možné falešné impulzy během dokmitávacího děje.

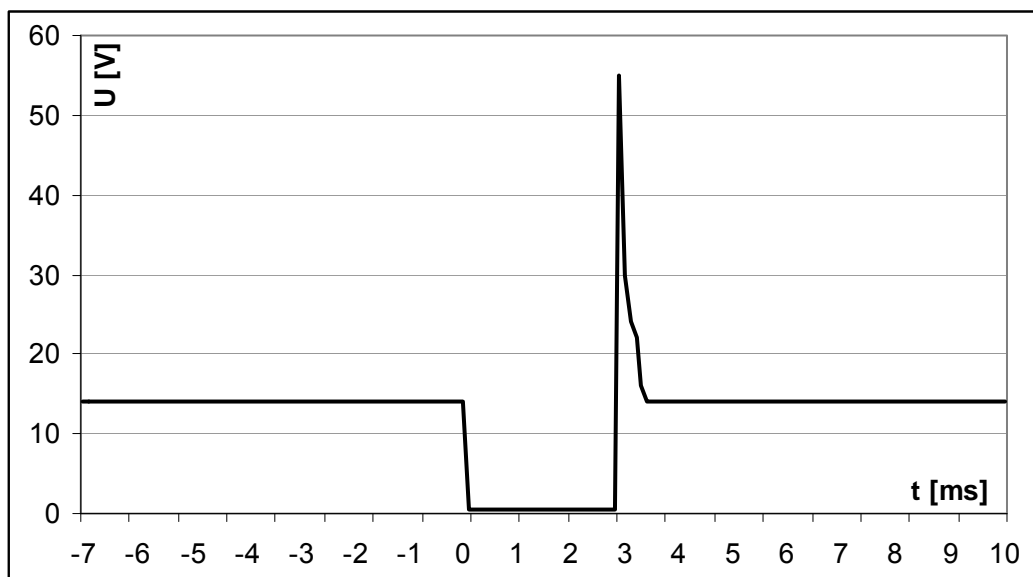
U některých vozů není vysokonapěťová část vůbec přístupná. Zapalovací cívky jsou ve tvaru lišty, přímo nasazené na zapalovacích svíčkách. Je možné připojení přímo na výstupy řídicí jednotky motoru pro zapalování, kde je obdélníkový signál určený k buzení zesilovače integrovaného v zapalovací cívce.

Ve firmwaru musí být možnost nastavení korekce otáček, podle počtu válců motoru, a podle druhu zapalování (dvoutakt nebo cívka s dvojitým výstupem). Někdy se totiž používají zapalovací cívky se dvěma vývody, pak není použit rozdělovač. Jedna cívka je připojena na dva válce, pálí jak v kompresním tak ve výfukovém zdvih. Jedna z jisker je vždy činná, druhá je nevyužitá. Pro otáčkoměr to znamená dvojnásobek impulzů než obvykle.

1.2.2 Snímání otáček na vstřikovacím ventilu

U motorů vybavených vstřikováním se jedná o jednoduchou a spolehlivou metodu měření otáček. Signál se odebírá z výstupu řídicí jednotky motoru pro solenoid. Přibližný průběh napětí na tomto vývodu je na obr. 2.

Měření periody je vhodnější vztáhnout k strmé sestupné hraně (otevření ventilu), jelikož uzavírání ventilu je děj pomalejší a navíc je doprovázen velkou napěťovou špičkou vznikající na indukčnosti ventilu. Tuto je nutno ve vstupních obvodech spolehlivě potlačit, neboť má rychlý nástup.



Obr. 2 Přibližný tvar signálu na vstřikovacím ventilu

1.2.3 Snímání otáček motoru s využitím alternátoru

Tento způsob je jednoduchý k realizaci, ovšem použití je nepraktické. Prvně, alternátor musí mít vyvedenou jednu fázi (svorka se označuje W). Tento vývod již většina dnešních alternátorů nemá, jelikož pro něj není využití. Tímto způsobem je řešen otáčkoměr u starých dieselových motorů bez el. vstřikovacích ventilů. U některých alternátorů výrobci svorku W ponechají jako kontrolní, v tom případě má smysl o této metodě uvažovat. V podstatě není problém si tuto svorku na alternátor přidat, ovšem demontáž a rozebrání alternátoru za to nestojí. Další, již menší problém je, že se alternátor bohužel otáčí jinou rychlostí než motor (vyšší), proto je nutné pro zjištění otáček buď znát převodový poměr řemenic, nebo otáčkoměr zkalibrovat.

Princip metody spočívá ve měření kmitočtu výstupního napětí na cívce alternátoru. Napětí je sinusového průběhu, s amplitudou přibližně 16V. Další zpracování spočívá v limitaci a oříznutí záporné půlvlny, pak je k dispozici přibližně obdélníkový signál. Pomocí Schmittova klopného obvodu se zvýší strmost hran. Nyní je možné tento signál připojit k mikrokontroléru a měřit jeho periodu. Po zkorigování převodového poměru jsou k dispozici otáčky motoru.

1.2.4 Snímání otáček z čidla polohy klikového hřídele

Dalším využitelným snímačem je čidlo polohy klikového hřídele. Používají ho všechny motory řízené elektronickou řídicí jednotkou. Může být buďto indukční nebo hallův. Signál má poměrně vysoký počet impulzů na otáčku (používá se kolem 60 zubů na hřídeli), a přesný počet zubů u daného vozu je nutno nejprve zjistit. Vyhodnocení signálu pro zjištění otáček komplikuje fakt, že se používají referenční značky (nejčastěji vynechání zubu). Nejčastěji to bývá jedna až dvě na otáčku. Toto je nutné ošetřit ve firmwaru, nejlépe s možností autodetekce během nastavování. Principiálně je možné měřit jak periodu mezi jednotlivými zuby, tak periodu mezi referenčními značkami.

1.3 Měření rychlosti vozu

Jednotka bude rychlost vozu získávat primárně z GPS modulu. Zásadní nevýhoda je že rychlost z GPS modulu není dostupná po celou dobu (např. v tunelu nebo hustém lese). Bylo by vhodné zajistit překlenutí z jiného čidla. Rychlost se dá získat z indukčního snímače na převodovce, je-li vůz vybaven elektronickým tachometrem. Signál z indukčního snímače má tvar sinusovky, s amplitudou několika voltů, která roste s otáčkami.

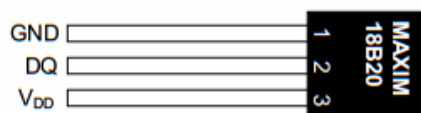
Snímač se chová jako měkký zdroj napětí, takže připojený obvod nesmí snímač příliš zatížit, aby nedošlo k vyřazení původního tachometru z činnosti. Také je neznámý počet zubů na jednotku rychlosti, toto je nutné zjistit experimentálně pro každý typ vozu.

U vozů s mechanickým tachometrem nastává problém, že není kde vzít signál odpovídající rychlosti vozu. Na těchto vozech může být ABS, který má senzory na kolech, ovšem zasahovat do tohoto systému nese bezpečnostní riziko. Někdy řídicí jednotka ABS posílá do řídicí jednotky motoru signál o rychlosti vozu. Může to být prostý analogový signál 0-10V, nebo je použit nějaký sběrníkový digitální systém (nejčastěji CAN). Z důvodu nedostatečné dokumentace byl tento způsob zavržen. Možnost přidání dalšího snímače na tachometr tu sice je (optická závora nebo indukční), tuto možnost ale komplikuje fakt že upravovat tachometr z vozu bývá náročné.

1.4 Snímače teploty

Pro měření teploty je možné využít klasických termistorů NTC, výhodné z hlediska nízké ceny, nevýhody však převažují a to nelinearita převodní charakteristiky a nutnost vedení analogového signálu motorovým prostorem, z toho vyplývá nízká odolnost proti rušení. Pro dosažení přesnosti alespoň $\pm 1^\circ\text{C}$ je nutné kalibrovat nejméně v jednom bodě, což je poměrně nepraktické.

Teplotní čidla s digitálním výstupem poskytují naproti tomu výhodu, že jsou zkalibrována z výroby (až na pár výjimek), nevýhoda je vyšší cena. Mezi často používaná digitální teplotní čidla patří Dallas DS18B20 a SMT160. S ohledem na cenu budou použita čidla 18B20. Poskytují teplotu jako 9 až 12 bitovou hodnotu se znaménkem, přímo ve stupních celsia což jednodušší zobrazení.



Obr.3 Teplotní čidlo 18B20 v pouzdře TO-92 (převzato z [4])

Zapojení vývodů je následující:

- 1 – **GND** – zem
- 2 – **DQ** – připojení ke sběrnici 1-Wire
- 3 – **VDD** – externí napájení 3.0-5.5V

Pro použití v parazitním režimu jsou propojeny vývody 1. a 3., a připojeny na zem. Sběrnice se připojí k vývodu č.2.

Teplotní čidla jsou kalibrována z výroby a mají přesnost $\pm 0,5^\circ\text{C}$ v rozsahu -10 až $+85^\circ\text{C}$. Maximální rozsah měření je -55°C to $+125^\circ\text{C}$, což je dostatečné i pro měření teploty chladicí kapaliny motoru. Pro připojení teplotních čidel je použita sériová sběrnice 1-Wire. Tato sběrnice byla navržena pro komunikaci mezi součástkami firmou Dallas Semiconductor (dnes Maxim). Výhodou této sběrnice je především možnost použití pouze jednoho vodiče (kromě zemního) pro napájení i komunikaci. K jejich odlišení se používá 64-bitové unikátní číslo, nastavené ve výrobě. Pro prvotní zjištění identifikátorů všech zařízení na 1-Wire sběrnici slouží speciální procedura zvaná „Search ROM“. Kromě přímého napájení čidel vyhrazeným napájecím pinem umožňuje sběrnice i „parazitní“ napájení. Spočívá v tom, že zařízení odebírají energii kdykoliv je sběrnice v log.1. Pro běžnou komunikaci je toto dostatečné. Každé zařízení na sběrnici 1-Wire má výstup s otevřeným kolektorem. Sběrnice je v klidovém stavu díky pull-up rezistoru v log. 1. Komunikaci zahajuje vždy Master zařízení resetovacím pulzem (log. 0) délky alespoň $480\ \mu\text{s}$. Pokud je na sběrnici přítomno Slave zařízení, po ukončení resetovacího pulzu ohlásí svou prezenci stažením sběrnice na log. 0 po definovanou dobu. Následná komunikace probíhá v tzv. timeslotech dlouhých typicky $60\ \mu\text{s}$. Komunikace je podrobněji popsána např. v katalogovém listu čidla [4].

U teplotních čidel je ale problém s napájením během probíhajícího měření teploty. Výrobce uvádí že jedno čidlo může během měření odebírat až 1,5mA, na což napájení z pull-upu 4k7 nestačí. Proto se používá technika připojování „silnějšího“ pull-upu během teplotního převodu. Prakticky se toto provádí tak, že mikrokontrolér odešle příkaz pro zahájení měření a ihned poté sepne tranzistor MOSFET, připojující napájecí napětí přímo ke sběrnici (přepnutí portu do výstupního režimu a zapsání log.1). Tato technika umožňuje vystačit s dvou vodičovým připojením vícero čidel paralelně (sběrnice a zem) a proto bude v konstrukci použita. V případě že se využije kostra vozu, je nutné natáhnout pouze jeden drát navíc. Z důvodu odolnosti vůči rušení, a také odstranění problémů s úbytky napětí na kostře bude pravděpodobně použit stíněný kabel, který bude ukostřen na jednotce.

1.5 GPS modul

K jednotce bude připojen GPS modul se sériovým výstupem. Využívaná data budou aktuální rychlost, poloha, přesný čas, nadmořská výška. Většina GPS přijímačů používá protokol NMEA 0183. Je to jednoduchý přenos textových rámců (vět) rychlostí 4800 Baudů. Věta začíná \$GP, následuje třímístný kód určující typ věty. Jednotlivé hodnoty jsou odděleny čárkami a mají pevné pořadí. Věta končí kontrolním součtem a odřádkováním. Přenosovou rychlostí 4800B se dá přenést nejvýše 480 znaků za sekundu, a proto se posílají jen vybrané věty, aby bylo možné dodržet periodu obnovy 1 sekunda. Konkrétní sada vět, kterou posílá modul Garmin GPS18LVC z výroby je GPRMC,GPGGA,GPGSA,GPGSV,GPGLL,GPVTG.

První využívanou větou bude GPGGA. Obsahuje přesný čas UTC, polohu, stav měření, počet aktuálně používaných satelitů, odhad přesnosti a nadmořskou výšku. Příklad věty:

```
$GPGGA,124746.000,4912.4895,N,01660.9255,E,1,05,2.1,125.0,M,47.1,M,,0000*4F
```

Prvním parametrem je světový čas UTC (12:47:46), následuje zeměpisná šířka (4912.4895), na které polokouli (N-severní), zeměpisná délka (01660.9255), na které polokouli (E-východní), stav měření a počet satelitů použitých k měření. Devátý parametr (125.0) udává nadmořskou výšku, následuje jednotka (M-metry). Zeměpisná šířka a délka je uváděna ve stupních jako dekadické desetinné číslo, vynásobené 100.

Stav měření udává validitu dat posílaných přijímačem. Možné hodnoty jsou:

0 : Neplatná data.

1 : Data platná, běžné měření

2 : Data platná, měření zpřesněno DGPS (systém WAAS nebo EGNOS).

Aktuální rychlost je možné získat z věty GPRMC.

Příklad věty:

```
$GPRMC,124746.663,A,4912.4895,N,01660.9255,E,0.27,358.86,200804,,*1A
```

První parametry opět obsahují čas a zeměpisnou polohu. Aktuální rychlost je sedmý parametr, jednotkou je námořní míle (knot). Osmý parametr je směr vektoru rychlosti vzhledem k severu (ve stupních).

1.6 Rozhraní pro ukládání dat

Od jednotky se požaduje ukládání všech naměřených dat během jízdy. Z důvodu velkého objemu dat je nezbytné použít nějakou externí flash paměť. Ideální by bylo, kdyby jednotka obsahovala USB-host, umožňující připojení běžného zařízení USB Mass Storage. Vzhledem ke komplexnosti tohoto řešení bylo přihlédnuto k jinému způsobu, a to využití paměťové karty. Jednotka bude obsahovat slot pro SD kartu. V současné době jsou SD karty cenově výhodné (přibližně 150Kč za 2GB). Z hlediska komfortnosti obsluhy bude nutné aby jednotka podporovala souborový systém FAT32, používaný na výměnných médiích.

Jelikož má mikrokontrolér dvě periferie UART, je možné realizovat posílání naměřených dat přes sériový port, pro přímé zobrazení či záznam na počítači. Účel tohoto rozhraní je spíše pro debugging. Zvážena však bude možnost využít těchto dat k přímému kreslení grafů a zobrazení všech veličin na velké obrazovce v doprovodném PC programu.

2 HARDWARE

2.1 Popis zapojení

Obvodové schéma jednotky je na obr.4. Napájecí napětí z palubní sítě automobilu je přivedeno na pin 5 hlavního konektoru. Jednotka je navržena pro použití ve vozech s palubní sítí 12V. Připojení napájení se předpokládá za spínací skříňkou, proto na jednotce není žádný spínač napájení. Napájecí napětí je nejdříve lineárním stabilizátorem 7805 (IC1) sníženo na 5V, které se používá pro napájení LCD displeje a GPS modulu. Proudový odběr LCD řadiče je asi 1mA, LCD podsvícení asi 10mA a GPS modul odebírá 60mA. Mikrokontrolér a SD/MMC karta vyžaduje napájení 3.3V. K tomuto účelu slouží low-drop stabilizátor LF33 (IC2). Celkový odběr z 3.3V větve nepřesahuje 30mA. Napájení mikrokontroléru je blokováno kondenzátory C1 - C4, napájení analogové části je filtrováno feritovou perličkou L1. Výrobce stabilizátoru uvádí, že pro plnou stabilitu regulátoru postačuje výstupní kondenzátor 2.2uF [7]. Nicméně specifikace SD/MMC karet uvádějí špičku proudu při zasunutí karty až 200mA, způsobenou nabíjením interního kondenzátoru na kartě. Tato špička by způsobila pokles napětí 3.3V větve dostatečný pro nespolehlivou funkci. (Nebezpečí spuštění brown-out detektoru a resetu mikrokontroléru.) Pro bezproblémové zasouvání karet za běhu zařízení jsou potřebné relativně velké kondenzátory, 220 – 470uF v běžném provedení nebo 47uF tantal. V této konstrukci byl použit kondenzátor 100uF v provedení low-ESR.

Odhad ztrátového výkonu stabilizátoru 5V:

Předpokládá se maximální napájecí napětí 15V, a odběr celého zařízení max. 100mA.

$$P = U \cdot I = (15 - 5)V \cdot 100mA = 1W$$

Ztrátový výkon 1W je typicky možné odvést z pouzdra TO-220 i bez chladiče. Tepelný odpor přechod-prostředí se pohybuje v rozmezí 23-65°C/W, záleží na výrobci i na cirkulaci vzduchu. Regulátor ale bude přišroubován k plošnému spoji což zlepší odvod tepla. V nejhorším případě bude oteplení přechodu 65°C, a při maximální teplotě přechodu 150°C vychází teplota okolí 85°C, což bude jistě splněno. Přesto je doporučeno použít alespoň malý chladič.

Mikrokontrolér je taktován kmitočtem 24MHz, který je získáván z hlavního oscilátoru 8MHz násobičkou PLL. Použit je krystal v pouzdře HC-49/S (Q1). Zatěžovací kondenzátory oscilátoru C7, C8 jsou podle výrobce 20pF. Na desce je počítáno i se sekundárním nízkofrekvenčním oscilátorem 32.768kHz, tvořeným Q2, a kondenzátory C5, C6. V této konstrukci není využit, a tím pádem ani osazen. Pro standardní start programu z flash paměti je nutné uvést vývody BOOT0 (pin 60) a BOOT1 (pin 28) na log. 0. Toto zajišťují odpory R3 a R4. Pro připojení programátoru/debuggeru je vyvedena pinová lišta SV2

Konektor SD/MMC karty je v provedení SMD. Jsou využity pouze piny potřebné k SPI komunikaci s kartou. Pro detekci karty je využit kontakt na konci slotu, který při vložení karty sepne na zem. Vývod je označen CD# (card detect). Kontakt snímání stavu přepínače zámku zápisu (write-protect) není pro jednoduchost zapojen.

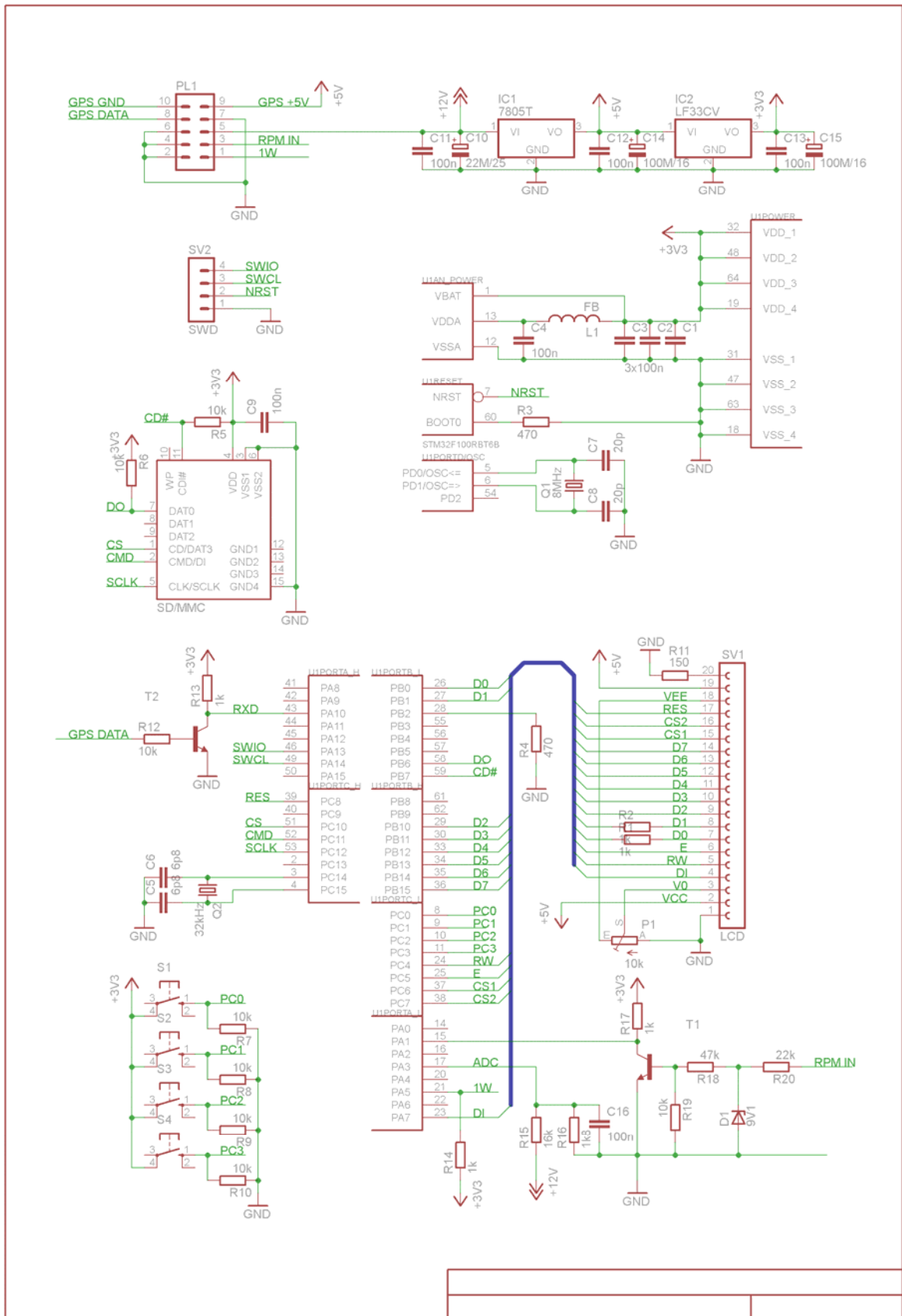
Přepokládá se že karta, která se vkládá do jednotky je určená k zápisu. Pro správnou inicializaci karty je nutný pull-up rezistor R6. Jedná se o signál ve směru karta – mikrokontrolér. Opačný směr má definovány inicializační úrovně v programu. Ovládací tlačítka jsou zapojena jako aktivní v log.1. Pull-down rezistory zajišťují vyšší odolnost proti rušení, protože vnitřní pull-down v mikrokontroléru je poměrně slabý. (Ekvivalentní odpor udává katalogový list až 50kΩ. [2], str. 56).

LCD displej je připojen 20-ti pinovou řadovou pájecí lištou (SV1). Napájení podsvícení je z 5V větve, přes odpor R11. Záporné napájení nutné pro funkci vyrábí měnič integrovaný na LCD, toto je vyvedeno na pin 18. Má hodnotu přibližně -9V a je přivedeno na potenciometr P1, kde je sníženo a připojeno ke svorce V0 (pin 3). Tímto se nastavuje kontrast displeje. V datových linkách D0 a D1 jsou zařazeny oddělovací odpory R1, R2 které slouží k převodu napěťové úrovně z 5 na 3.3V v případě že je čteno z LCD. Důvodem je že piny PB0 a PB1 mikrokontroléru nejsou 5V tolerantní, jak je uvedeno v katalogovém listu [2].

Vstup pro snímání otáček je na pinu 3 hlavního konektoru. Tento vstup je určen pro připojení na solenoid ventil vstřikování. Pokud je použit jiný způsob snímání otáček, připojí se na tento vstup obdélníkový signál z externího tvarovače/sondy. Z praktických důvodů není vhodné, aby jednotka obsahovala všechny vstupní obvody. Plošný spoj by vycházel příliš velký. Ze signálu vstřikovače jsou nejdříve pomocí R21 a zenerovy diody D1 oříznuty kladné části impulzu, včetně špiček vznikající na indukčnosti ventilu, na napětí 9.1V. Dělič R19, R20 způsobuje posun překlápěcí úrovně dané tranzistorem T1 z 0.65V na 3.7V (přibližně). Dosáhne se tak vyšší odolnosti proti rušení, a také se tím zajistí spolehlivější snímání v případě, že by úbytek napětí na zemnicím vedení a spínacím tranzistoru ECU přesáhl 0.6V. Signál po úpravě je zaveden do vstupu TIM2-CH2. Tento vstup byl opět vybrán s ohledem na návrh plošného spoje.

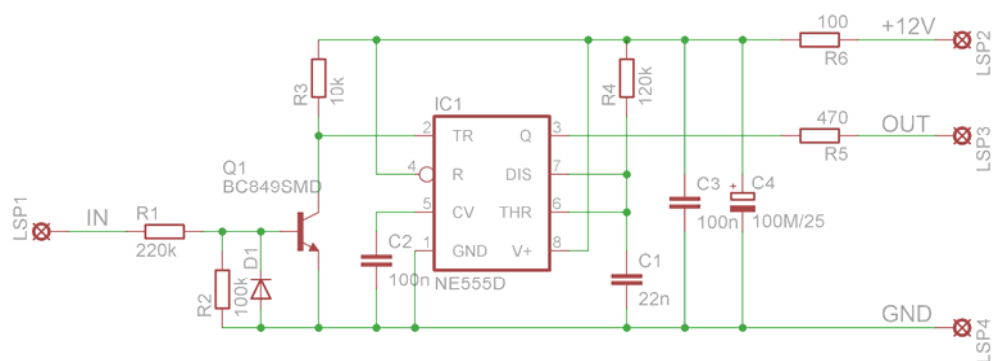
Sběrnice 1-Wire pro připojení teplotních čidel je vyvedena na pin 1 hlavního konektoru. Předpokládá se vedení tvořené krouceným párem nebo stíněným kabelem. Pro lepší ochranu mikrokontroléru by bylo lepší použít nějaký specializovaný budič této sběrnice, z důvodu špatné dostupnosti speciálních integrovaných obvodů a z toho vyplývající ceny bylo zvoleno toto nejjednodušší řešení. Není možné použít ani ochranný oddělovací odpor, protože úbytek napětí na něm během fáze měření teploty by byl neúnosný. (Minimální napájecí napětí čidel 18B20 dle výrobce je 3.0V [4], str. 14, a napájení mikrokontroléru je 3.3V). Pull-up rezistor sběrnice (v režimu komunikace) byl snížen z doporučených 4.7k na 1k. Důvodem je rychlejší návrat sběrnice do log.1 během komunikace a tím i lepší zajištění parazitního napájení. Čidla mají v úrovni log.0 daný maximální proud 4 mA (sink-current), který ještě nebude překročen.

GPS modul je připojen k jednotce na piny 10 (GND), 9 (+5V), 8 (data) hlavního konektoru. Protože použitý modul Garmin GPS 18 LVC poskytuje data invertovaná [6], byl doplněn jednoduchý invertor s tranzistorem T2. Pro rychlosti sériového přenosu 4800 Baudů plně dostačuje a zabírá relativně málo místa na plošném spoji než jiné řešení formou integrovaného obvodu. Zároveň dochází k převodu napěťové úrovně GPS modulu (5V) na úroveň napájení procesoru (3.3V). Pro snazší návrh plošného spoje byl využit druhý asynchronní přímáč USART1, s odpovídajícím vstupem RXD1.



Obr.4: Obvodové schéma jednotky

2.2 Tvarovač signálu ze zapalování



Obr.5: Schéma zapojení tvarovače



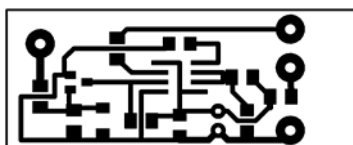
Obr.6: Detail možného provedení sondy na zapalovacím kabelu

Na obr. 5 je schéma tvarovače signálu ze zapalovací soustavy. Snímač je tvořen kouskem plechu nebo drátu, obtočeného kolem jednoho ze zapalovacích kabelů. Jedno z možných řešení je na obr. 6. Kapacitní vazbou se bezkontaktně přenáší impulzy na vstup tvarovače. Odporů R1, R2 se nastavuje citlivost. Hodnoty součástek jsou silně závislé na provedení sondy a na parametrech zapalovací soustavy daného vozidla. Je nutné je určit zkusmo. Dioda D1 ořezává záporné špičky signálu. Kladná špička způsobí sepnutí tranzistoru T1 a spuštění časovače 555, zapojeného jako monostabilní klopný obvod (MKO). Během přechodového děje výboje na svíčke dochází k rezonanci v primárním obvodu zapalovací cívky. Projevuje se to tlumenými kmity po skončení vlastního výboje. Tyto kmity mohou znovu otevřít tranzistor T1. Účelem MKO je překrýt celou dobu trvání přechodového děje, aby na výstupu byl skutečně jen impuls odpovídající jednomu výboji. Přechodový děj trvá typicky 1-1.5ms, proto byla s ohledem na tolerance součástek zvolena časová konstanta delší.

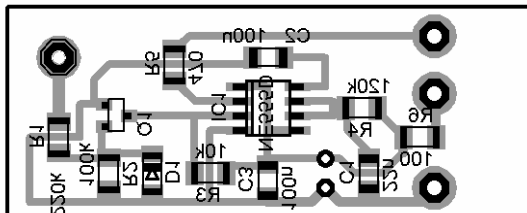
Časová konstanta se vypočítá přibližně jako

$$t = 1.1 \cdot R \cdot C = 1.1 \cdot 120k\Omega \cdot 22nF = 2.9ms$$

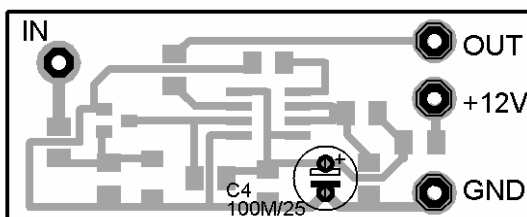
Časovou konstantu je možné v případě potřeby snížit, např. snížením odporu R4 na 82k. Maximální opakovací kmitočet je tedy přibližně 300Hz, což stačí na měření otáček do 9000. Napájecí napětí je filtrováno prvky R6,C4,C3. Maximální napájecí napětí obvodu 555 se liší podle výrobce, od +16 do +18V, což umožňuje přímé použití ve vozidle. (maximální napětí palubní sítě je 14.6V). Na odporu R6 navíc vzniká úbytek přibližně 1V. Protože bude tento tvarovač umístěn v motorovém prostoru, doporučuje se použít na místě časovače verzi SE555D s rozšířeným teplotním rozsahem -55 až 125°C, pokud je k dispozici. Napájení tvarovače se předpokládá z přívodu k zapalovací cívce, takže je zajištěno automatické zapínání a vypínání. Výstup tvarovače se připojí k jednotce na vstup pro měření otáček (pin 3). Motiv plošného spoje je na obr. 7, Osazovací výkresy na obr. 8 a obr. 9. Tvarovač je možné zabudovat do miniaturní krabičky, nebo jen zatavit do jedné či dvou vrstev silnostěnné teplem smrštitelné bužírky.



Obr. 7 Motiv plošného spoje tvarovače (měřítko 1:1)



Obr. 8 Osazovací plánec, strana spojů (měřítko 1.5:1)



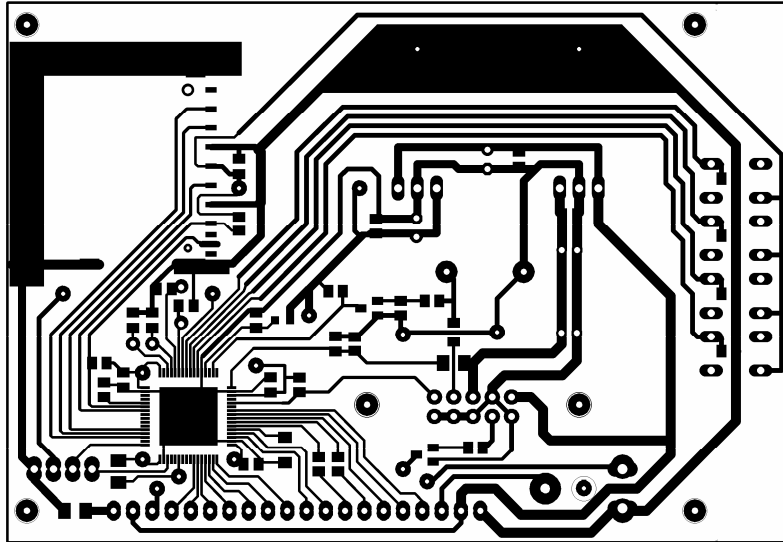
Obr. 9 Osazovací plánec, strana součástek (měřítko 1.5:1)

2.3 Mechanická konstrukce jednotky

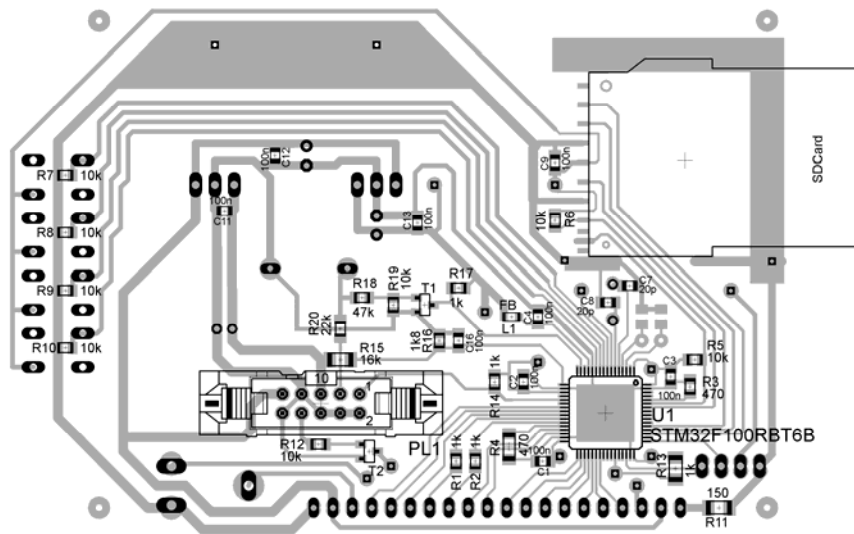
Rozměry jednotky jsou podřízeny použitému LCD displeji (93x70mm). Mechanickou konstrukci přiblíží náčrt na Obr. 10. Plošný spoj má rozměry odpovídající LCD, takže pasuje přesně pod displej. Mezi oběma deskami jsou distanční sloupky M3x8mm, udávající jejich rozestup. Celková potřebná hloubka krabičky je minimálně 18mm (bez tloušťky stěn), může však být o něco větší. Aby bylo možno použít takto malý rozestup, jsou nutné elektrolytické kondenzátory s výškou pouze 6mm. Potenciometr kontrastu je nízkoprofilový, protože přímo nad ním je prostor snižený o plechové upevnění displeje zahnutím. Pro displej se počítá s vyřiznutím obdélníkového okna, takže bude vystupovat nad krabičku. Při pohledu zepředu na displej strana na které jsou umístěna tlačítka vyčnívá mimo displej zprava. Z levé strany vyčnívá paměťová karta. Celkové rozměry plošného spoje jsou 104x72mm. Motiv plošného spoje je na obr. 11, osazovací plánky na obr. 12 a 13. Jednotka byla navržena na jednostranný plošný spoj, z důvodu nižší ceny a možnosti amatérské výroby. Rozložení napájecích pinů mikrokontroléru je největší problém při návrhu jednostranné desky. Napájení je rozvedeno pomocí drátových propojek. Celkem je na desce 8 drátových propojek. Hlavní konektor je dvouřadý typu MLW10G nebo lépe PSL10 se zámkem. Je připojen nestandardně ze strany spojů, a upevněn dvěma samořeznými vruty ze strany součástek. Připojovací kabel je desetižilový plochý kabel. Umožňuje to případné naklápění jednotky na kloubu. Naklápění je výhodné pro zajištění maximálního kontrastu při pohledu kolmo, protože má displej malé pozorovací úhly.



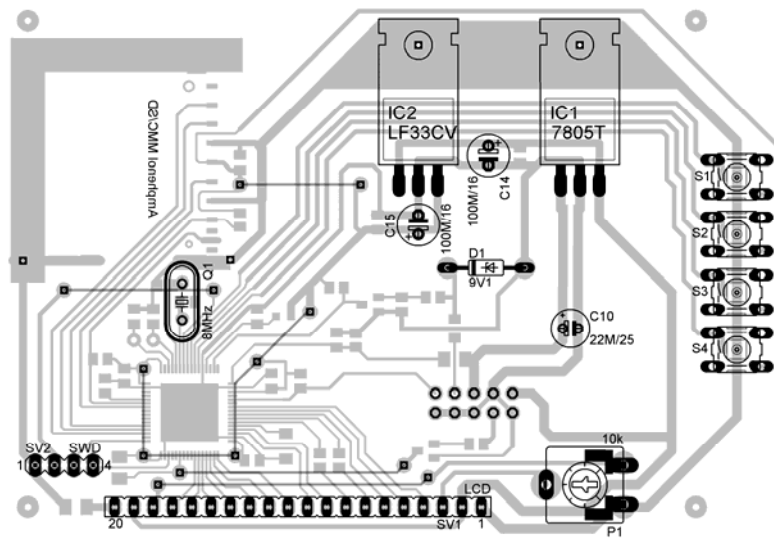
Obr. 10: Náčrtek mechanické konstrukce (rozměry neodpovídají skutečnosti)



Obr.11: Motiv plošného spoje (měřítko 1:1, pohled ze strany součástek)



Obr.12: Osazovací plánec, strana spojů (měřítko 1:1)



Obr.13: Osazovací pláněk, strana součástek (měřítko 1:1)

3 SOFTWARE

3.1 Modul lcd.c

Knihovna funkcí lcd.c obsluhuje LCD displej s řadičem KS0108. Jako základ byla použita knihovna od autora Fabiana Maximiliana Thiele, která je napsána pro 8-bit procesory AVR. Pro použití na STM32 bylo nutné nízkoúrovňové funkce pracující s displejem upravit. Změna hodnot na pinech byla přepsána z funkcí na makra, z důvodu, že funkce má dlouhý prolog kde dochází k zálohování registrů na stack. Tím došlo ke zrychlení kódu.

Pro práci s grafickým displejem je nutné mít k dispozici dostatek paměti na uložení celého obsahu displeje. V tomto případě je to 128x64 pixelů, což je 8192 bitů neboli 1kB. Použitý mikrokontrolér má dostatek paměti RAM pro tento účel, přesto knihovna využívá k práci s displejem paměť integrovanou v řadiči KS0108. Proto je sběrnice LCD displeje využívána obousměrně. Při změně obsahu displeje dochází k editaci přímo jen měněných částí (pixelů), což je obecně rychlejší než přesouvat celou kopii displeje z RAM mikrokontroléru do řadiče. Protože práce s displejem je časově nejnáročnější z hlediska množství vykonávaného kódu, byly funkce optimalizovány pro rychlost.

Pro přepínání směru na sběrnici se používá inicializační struktura definovaná globálně v souboru main.c, a je tímto modulem převzata jako extern.

```
extern GPIO_InitTypeDef GPIO_InitStructure;
```

Poté již ve funkcích pro změnu směru portu není nutné při každém volání znovu plnit inicializační strukturu, ale jen se změní bity odpovídající směru a zapíše se do registrů:

```
#define LCDportin()
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING; \
GPIO_Init(GPIOB, &GPIO_InitStructure);
// makro pro přepnutí sběrnice na čtení z displeje

#define LCDportout()
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; \
GPIO_Init(GPIOB, &GPIO_InitStructure);
// makro pro přepnutí sběrnice na zápis do displeje
```

Propojení displeje a MCU komplikuje fakt, že v použitém pouzdře LQFP64 není k dispozici souvislých 8 pinů jednoho portu. Nejvýhodnější řešení je pravděpodobně využít 6 sousedících pinů PB15 až PB10, a k nim doplnit PB1 a PB0. Takto je datová sběrnice alespoň na jednom portu. Skládání dat je tímto značně komplikováno. Pro převod z 8bit proměnné na tyto piny datové sběrnice je použito makro

```
#define LCDout(data)      spodnia=((data&0xFC)<<8)|(data&3); \
                          hornia=((~spodnia)&0xFC03)<<16; \
                          GPIOB->BSRR=hornia|spodnia;
```

Ačkoliv takto zapsané je na první pohled hrozné, bylo ověřeno v debuggeru, že toto makro je tvořeno pouze 7 instrukcemi mikrokontroléru.

Pro čtení z LCD je použita inverzní převodní funkce, která není ve formě makra, protože se vyskytuje pouze jednou ve funkci LCDDoReadData.

```
temp=GPIOB->IDR; //nacist data
data=((temp>>8)&0xFC)|(temp&3);
// Složení hodnoty z portu do 8bit proměnné
```

Zbývající řídicí signály jsou zapojeny tak, aby byl snadný návrh plošného spoje. (viz obvodové schéma na obr. 4). Kromě datové sběrnice, která je napevno zvolena, je možné přiřazení řídicích signálů změnit v souboru lcd.h

Knihovna exportuje následující funkce:

```
void LCDSetReset(uint8_t value);
void LCDInit(void);
void LCDClearScreen(void);
```

První funkce slouží k řízení resetu LCD, používá se při inicializaci, ale také je možné takto řadič uvést do režimu s nízkým odběrem (v resetu). Aktivní úroveň resetu je 0, v běžném provozu je tedy nutná na resetu log.1. Druhá funkce provede inicializaci řídicích registrů řadiče a je nutné ji zavolat po každém uvolnění resetu. Funkce LCDClearScreen smaže displej.

Funkce pro práci s textem:

```
void LCD_Setpos (uint8_t u8Column,uint8_t u8Line);
void LCD_Printf (uint8_t *au8Text, FONT_DEF *fonttype);
void LCD_Putchar (uint8_t u8Char, FONT_DEF *fonttype);
```

S textem se pracuje z důvodu zjednodušení a zrychlení v osmi řádcích. Nejdříve se vybere místo pro vypsání textu pomocí funkce LCD_Setpos. Tato nastaví aktivní pozici „kurzor“ na daný sloupec (pixel zleva doprava, tj 0-127) a řádek (0-7). Poté je možné vypsát jeden znak pomocí LCD_Putchar , nebo řetězec standardně ukončený ‘0’ pomocí LCD_Printf . Prvním parametrem jsou data k vypsání, druhým parametrem je pointer na konkrétní font v tabulce fontů.

Grafické funkce:

```
void bargraf(uint8_t xline,uint8_t start, uint8_t stop,
uint8_t value);
void LCDSetDot(uint8_t x, uint8_t y, uint8_t color);
```

Funkce bargraf vykreslí vodorovný pruh, orámovaný 1px rámečkem. Parametr xline udává řádek (0-7), start a stop jsou souřadnice začátku a konce bargrafu (rozsah 0-127). Parametr value určuje jaká část bargrafu bude vyplněna zleva. Rozsah je 0-255, funkce zajistí přepočítání podle délky bargrafu. Funkce LCDSetDot nataví jeden konkrétní pixel, zadaný pomocí x (rozsah 0-63) a y (rozsah 0-127). Parametr color může být 0 nebo 255, pro aktivní a neaktivní pixel. Standardní LCD má aktivní pixel černý a neaktivní barvy podsvícení (nejčastěji zelená). V této konstrukci je však použit inverzní displej s bílým podsvícením, proto má aktivní pixel barvu pozadí (bílou) a neaktivní je tmavší (modrá).

3.2 Modul gps.c

Tato knihovna slouží pro dekodování dat z protokolu NMEA, kterým komunikuje GPS modul. Byla využita knihovna TinyGPS, která byla přeportována z jazyka C++ do C. Zdrojový kód je k dispozici pod licencí GNU. Původní TinyGPS využívá pro získání rychlosti větu GPRMC, kde je rychlost k dispozici v knot (námořních mílich). Pro získání rychlosti tedy bylo doprogramováno dekodování věty GPVTG, kde je rychlost k dispozici přímo v km/h. V programu tak odpadne nutnost převodu - dělení desetinným číslem. Počet aktivně používaných satelitů a nadmořská výška se získávají z věty GPGGA, pozice, datum a čas z věty GPRMC.

Pro vstup dat slouží funkce `parse(char c)`, která přijme jeden znak. Tyto znaky skládá do jednotlivých vět, začínajících na \$ a končících na `\r\n` (CRLF). Tato funkce se volá z přerušení sériové linky, tak je zajištěno dekodování dat „na pozadí“ :

```
void USART1_IRQHandler(void)
{
    parse(USART_ReceiveData(USART1));
}
```

Po složení úplné věty je zavolána funkce `term_complete()`, která provede vlastní dekodování termů z věty.

Protože data ve větách NMEA jsou udávána s přesností na setiny, vrací výstupní funkce hodnoty v setinách dané veličiny. Pro získání jednotek stačí vydělit 100, což by neměl být problém z hlediska zátěže procesoru (celočíslné dělení). Použité funkce jsou:

```
unsigned long speed(void);
```

Vrací rychlost v setinách km/h.

```
long altitude(void);
```

Vrací nadmořskou výšku v centimetrech.

```
byte numsats(void);
```

Vrací počet aktuálně používaných satelitů.

```
void get_position(long *latitude, long *longitude);
```

Vrací aktuální pozici v tisícinách stupně nadmořské výšky, šířky.

```
void get_datetime(unsigned long *date, unsigned long *time);
```

Vrací aktuální datum a čas (GMT).

3.3 Modul DS18B20.c

Tato knihovna slouží k práci s čidly teploty Dallas (Maxim) DS18B20. Její součástí je i implementace protokolu 1-wire v souboru onewire.c s příslušnými hlavičkovými soubory. Je použita knihovna od Martina Thomase, opět původně pro 8-bit AVR a upravena pro STM32. Přepsána byla práce s porty a čekací rutiny. Knihovna je napevno upravena pro parazitní napájení. Upravené funkce/makra pro obsluhu portů jsou:

```
#define OW_OUT_LOW()      GPIO_ResetBits(OW_PORT,OW_PIN);
#define OW_OUT_HIGH()    GPIO_SetBits(OW_PORT,OW_PIN);
```

Tyto funkce mění stav portu v režimu výstupu, a využívají k tomu funkce poskytované ovladačem GPIO, který je dodán výrobcem mikrokontroléru (stm32f10x_gpio.c). Prakticky se pracuje přímo s registry BRR a BSRR. Příslušný port a pin na kterém se bude sběrnice 1-wire nacházet určují parametry OW_PORT,OW_PIN, které jsou definovány v souboru onewire.h. V této konstrukci je sběrnice na pinu PA5:

```
#define OW_PORT    GPIOA
#define OW_PIN    GPIO_Pin_5
```

Další funkcí je čtení hodnoty na portu, tato opět využívá dodaného ovladače:

```
uint8_t OW_GET_IN(void)
{
    return GPIO_ReadInputDataBit(OW_PORT,OW_PIN);
}
```

K obsluze portu jsou ještě potřebné funkce pro změnu směru portu:

```
void OW_DIR_OUT(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = OW_PIN;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(OW_PORT, &GPIO_InitStructure);
}

void OW_DIR_IN(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = OW_PIN;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(OW_PORT, &GPIO_InitStructure);
}
```

Práce s ovladačem GPIO v případě změny směru portu je poměrně těžkopádná a tyto funkce se provádění poměrně dlouho. Bylo by možné tyto funkce přepsat přímo až na změnu hodnot příslušných registrů, ale tím by se ztratila možnost rychlé změny pinu, a není to ani doporučeno z důvodu přenositelnosti kódu na podobný mikrokontrolér.

Pro komunikaci na sběrnici je nutné dodržet poměrně přesné časování (timesloty). Byla vytvořena funkce zpoždění 5us, jejímž parametrem je násobek tohoto zpoždění. Funkce byla odladěna v simulátoru mikrokontroléru, a její přesnost vyhovuje pro účely protokolu 1-wire.

```
void _delay_5us(uint32_t pocet)
{
    uint32_t temp;
    temp=24*pocet-3;
    for(; temp != 0; temp--);
}
```

Všechny zpoždění protokolu jsou poté určeny jako násobky pěti mikrosekund.

Nejdříve je nutné detekovat všechna čidla připojená na sběrnici pomocí procedury SearchROM. Tato je implementována ve funkci ow_rom_search() v souboru onewire.c. Je volána inicializační funkcí uint8_t DS_18B20_Init(). Funkce vrací počet čidel na sběrnici. Během toho dojde k naplnění interního pole Tcidla unikátními identifikátory nalezených senzorů. Pole Tcidla je definováno takto:

```
unsigned char Tcidla[MAXSENSORS][DS18X20_SP_SIZE];
```

Počet čidel je omezen parametrem MAXSENSORS, který je definován v hlavičkovém souboru DS18B20.h na hodnotu 3. V parazitním režimu napájení z úrovně 3.3V logiky již není spolehlivé připojovat více než 3 čidla.

Vlastní měření proběhne voláním funkce DS18X20_start_meas(). Tato funkce odešle příkaz SKIP ROM, pro vybrání všech čidel a následně CONVERT T spustí vlastní měření teploty. Ihned po posledním příkazu dojde k přepnutí portu do výstupního režimu, čímž jsou napájeny čidla během převodu. V parazitním režimu napájení není možné číst busy status čidel, proto je nutné počkat pevnou dobu danou výrobcem (max 750ms pro plné rozlišení 12-bit.)

Naměřené teploty se z čidel stáhnou pomocí funkce:

```
void get_tploty(int16_t *teploty);
```

Tato funkce má jako parametr ukazatel na pole kde budou teploty uloženy. Formát teplot je znaménkové číslo v desetínách stupně celsia.

Pro zobrazení na displeji je nutné teplotu převést na ASCII znaky. K tomu slouží funkce

```
void temptostr(int16_t in, uint8_t *out){
    uint8_t neg;
    if (in<0)
    {
        in=-in;
        neg=1;
        out[0]='-';
    } else{
        neg=0;
    }
    out[0+neg]=(in/100);
    out[1+neg]=((in-100*out[0+neg])/10);
    out[2+neg]=(in-10*out[1+neg]-100*out[0+neg]);
    out[3+neg]=0;
    out[0+neg]+=0x30;
    out[1+neg]+=0x30;
    out[2+neg]+=0x30;

    if(out[1+neg] == '0' && out[0+neg] == '0') out[1+neg] = ' ';
    if(out[0+neg] == '0') out[0+neg] = ' ';

}
```

První parametr funkce je 16-bit znaménkový integer obsahující teplotu ve stupních celsia a druhý parametr je ukazatel na pole znaků (řetězec). Funkce zajišťuje potlačení nevýznamných nul (zleva) a před záporné čísla dodá znak mínus. Detekce záporného čísla probíhá v první části. Jestliže je teplota záporná, nastaví se pomocná proměnná `neg=1`, a na místo prvního znaku se zapíše mínus. V druhé části probíhá dekadické dělení pro získání počtu stovek, desítek a jednotek stupňů. K těmto počtům se připočte `0x30`, což provede posun na odpovídající znaky v ASCII tabulce. Řetězec znaků je ukončen nulovým znakem. Poslední dva řádky testují nevýznamné nuly. Nejprve v případě nuly na místě stovek a desítek současně jsou stovky nahrazeny mezerou. Poté pokud je nula na místě desítek dojde k jejímu nahrazení mezerou.

3.4 Obsluha tlačítek

Ovládací tlačítka jsou čtyři, a jsou umístěny napravo vedle displeje. Jsou připojená k portům PC0-PC3. Ovládání využívá oblíbený způsob tzv. soft-buttons, tj. funkce tlačítek se mění podle menu ve kterém se uživatel nachází. Aktuální funkce tlačítka je vykreslena na pravém okraji displeje, vedle příslušného tlačítka.

O softwarovou obsluhu tlačítek se stará modul buttons.c . Jedná se o implementaci softwarového odstranění záskmitů tlačítek. Části zdrojového kódu byly převzaty z [8]. Modul exportuje čtyři funkce, pro každé tlačítko jedna, které vykonávají stejný algoritmus. Princip odstranění záskmitů spočívá v opakovaném čtení stavu tlačítka s malým zpožděním, a v případě že více než N načtení je stejných, je to považováno za ustálený stav tlačítka. Počet N načtení se nastavuje parametrem UNBOUNCE_CNT v souboru buttons.h. Zvolena byla hodnota 5, která se ukázala jako dostatečná.

```
int B1Pressed (void) {
    static char KeyCount1 = 0, KeyPressed1 = 1;

    if (KeyPressed1) {
        if (!(GPIOC->IDR & B1) == 0 ) { // Check if S2 is not pressed
            if (KeyCount1 < UNBOUNCE_CNT) KeyCount1++;
            else {
                KeyPressed1 = 0;
                KeyCount1 = 0;
            }
        }
    }
    else {
        if ((GPIOC->IDR & B1) == 0 ) { // Check if S2 is pressed
            if (KeyCount1 < UNBOUNCE_CNT) KeyCount1++;
            else {
                KeyPressed1 = 1;
                KeyCount1 = 0;
                return (1);
            }
        }
    }
    return (0);
}
```

Tlačítka jsou zapojena obráceně, spínají na log.1, proto je přednastaveno KeyPressed1 = 1. Funkce jsou volány neustále ve smyčce, a protože jsou lokální proměnné typu static, zůstává počet opakování uložen i pro další iteraci. V případě že je tlačítko stisknuto, vrátí funkce hodnotu 1.

3.5 Hlavní program (main)

Hlavní program má na začátku inicializační část a poté se provádí nekonečná smyčka. Inicializace jednotky začíná se povolením hodinových signálů pro periferie na sběrnicih APB1 a APB2.

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB |  
RCC_APB2Periph_GPIOC | RCC_APB2Periph_GPIOD | RCC_APB2Periph_GPIOE |  
RCC_APB2Periph_USART1 | RCC_APB2Periph_AFIO, ENABLE);
```

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
```

Dále následuje poměrně rozsáhlá část inicializačního kódu GPIO portů. Provede se nastavení jako vstupní (GPIO_Mode_IN_FLOATING) nebo výstupní (GPIO_Mode_Out_PP). K tomuto slouží funkce GPIO_Init(), kterou poskytuje ovladač stm32f10x_gpio.c dodaný výrobcem.

Pak se provede jedno zavolání funkcí pro čtení stavu tlačítek, aby se provedla alokace paměti pro proměnné typu static, včetně jejich výchozí hodnoty. Bližší informace jsou popsány v předchozí kapitole 2.4.

```
B1Pressed();  
B2Pressed();  
B3Pressed();  
B4Pressed();
```

Dalším krokem inicializace je nastavení přerušovacího systému mikrokontroléru (NVIC). Toto nastavení provádí procedura NVIC_Configuration(). Provede se umístění tabulky vektorů přerušení do flash paměti:

```
NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
```

Povolí se přerušení od USART1 s prioritou 0 (nejvyšší):

```
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;  
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

Povolí se signál z portu C na linku EXTI3, nastaví se detekce na náběžnou hranu:

```
GPIO_EXTILineConfig(GPIO_PortSourceGPIOC, GPIO_PinSource3);  
EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;  
EXTI_InitStructure.EXTI_LineCmd = ENABLE;
```

Povolí se externí přerušení z linky EXTI3 s prioritou 1 (druhá nejvyšší):

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI3_IRQn;  
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

Následuje konfigurace sériového přijímače USART 1. Parametry jsou: rychlost 4800Baud, 8N1, bez řízení toku. Po jeho aktivaci je ještě přiřazeno přerušení na příznak RXNE, což znamená že přijímací registr není prázdný, tudíž že je potřeba odebrat znak.

```
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
```


Dále se nastaví časovač TIM2 pro měření periody a délky impulzu ze vstupu otáček.

```
TIM2->CCMR1= 0x0102;  
TIM2->CCER= 0x0002;  
TIM2->SMCR= 0x0064;  
TIM2->CCER|=0x0011;  
TIM2->PSC=240;  
TIM2->CR1=1;
```

Náběžná hrana spouští zachycení do CCR2 a sestupná do CCR1. Slave-controller provádí vynulování registru časovače na událost CCR2. Časovač běží volně a trvale, předdělička je 240.

Proběhne reset a inicializace LCD displeje, inicializace měření teploty, a vykreslení statických částí základní obrazovky (funkce je nazvána `pozadiLCD()`). Poslední částí spouštěcí procedury je inicializace knihovny SD karty. Následuje vstup do nekonečné smyčky hlavního programu.

V hlavní smyčce se provádí inkrementace proměnné `dly`. Po dosažení poloviny hodnoty konstanty `INTERVAL` se provede získání všech měřených hodnot, jejich zobrazení na displeji a uložení do záznamového bufferu (dlouhý řádek). Po překročení konstanty `INTERVAL` se provede pouze získání otáček, jejich zobrazení na displeji a jejich uložení do záznamového bufferu (krátký řádek). Blíže se záznamu věnuje kapitola 3.6. Pak je proměnná `dly` vynulována a smyčka začíná znovu. Konstanta `INTERVAL` je určena tak, aby jeden průchod smyčkou trval co nejbližší 1 sekundě. Během provádění hlavní smyčky je na displeji základní obrazovka, viz obr. 14. Zobrazeny jsou otáčky formou bargrafu, rychlost, teploty, nadmořská výška a počet přijímaných GPS satelitů.



Obr. 14 Základní obrazovka jednotky

Po stisknutí tlačítka `MENU` se provede odskok do obsluhy přerušování `EXTI3_IRQHandler()`. Dojde k překreslení displeje na obrazovku záznamu (obr. 15). V obsluze přerušování je nekonečná smyčka, ve které se neustále testují tlačítka. Ze smyčky a tím i z přerušování je možno vyskočit po stisknutí tlačítka `ZPET`, kdy se provede příkaz `break`:

```
if (B1Pressed()) {  
    LCDclearScreen();  
    pozadiLCD();  
    exitmenu=1;  
    EXTI_ClearITPendingBit(EXTI_Line3);  
    break;  
}
```

3.6 Záznam na SD kartu

Pro implementaci záznamu na SD kartu byla použita knihovna FatFs [9]. Skládá se ze dvou modulů. Prvním z nich je nízkoúrovňový ovladač SD/MMC karty mmcb.c, který poskytuje několik funkcí modulu ff.c. Tento druhý modul implementuje souborové systémy FAT32, FAT16 a FAT12.

Karta je provozována v režimu SPI. Připojení karty k mikrokontroléru je z hlediska návrhu plošného spoje umístěno na generické GPIO piny. Nevyužívá se možnost SPI kontrolérů, které mikrokontrolér nabízí a místo toho se sběrnice SPI vytváří softwarově. Na začátku modulu mmcb.c je sekce kde se doplní práce s porty. Obsluha portů byla naprogramována parametricky, což umožňuje v jiné konstrukci rychle změnit umístění SD karty na portech:

```
#define SD_CSport  GPIOC
#define SD_CSpin   10
#define SD_CLKport GPIOC
#define SD_CLKpin  12
#define SD_DIport  GPIOC
#define SD_DIpin   11
#define SD_DOport  GPIOB
#define SD_DOpin   6
#define CS_H()     SD_CSport->BSRR = 1<<SD_CSpin
#define CS_L()     SD_CSport->BRR  = 1<<SD_CSpin
#define CK_H()     SD_CLKport->BSRR = 1<<SD_CLKpin
#define CK_L()     SD_CLKport->BRR  = 1<<SD_CLKpin
#define DI_H()     SD_DIport->BSRR = 1<<SD_DIpin
#define DI_L()     SD_DIport->BRR  = 1<<SD_DIpin
#define DO         (SD_DOport->IDR) & (1<<SD_DOpin)
```

Z knihovny byly použity tyto funkce:

```
f_mount(0, &Fatfs);
```

Zajišťuje inicializaci knihovny, a zaregistrování karty jako jednotku 0.

```
f_open(&Fil, filename, mode flags);
```

Tato funkce otevírá soubor, a předává na něj handler Fil. Použití v této konstrukci je jednak pro detekci zda existuje nějaký soubor (mode flag je FA_OPEN_EXISTING), tak k vytváření souborů k zápisu (flagy jsou FA_WRITE | FA_CREATE_ALWAYS).

```
f_lseek(&Fil, fileoffset);
```

Funkce seek provádí posun po otevřeném souboru. Hodnotu posunu od začátku udává parametr fileoffset.

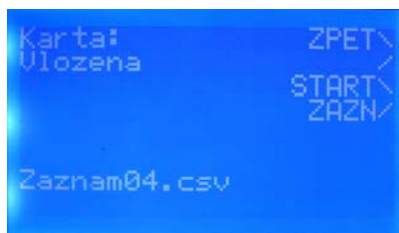
```
f_write(&Fil, WBuff, délka, &bw);
```

Vlastní funkce pro zápis do souboru. V parametrech se předává handler k souboru, buffer obsahující data, délku dat co mají být zapsána a poslední parametr vrací kolik bajtů bylo skutečně zapsáno (využití je při detekci zaplnění karty).

```
f_close(&Fil);
```

Tato funkce provede uzavření souboru (zápis do FAT tabulky). Použití je nutné jinak by došlo k chybám v souborovém systému. Pokud nejsou otevřeny žádné soubory, je možné kartu vyjmout/vložit za běhu aplikace.

V obsluze přerušeni se na displeji zobrazuje nabídka záznamu:



Obr. 15 Obrazovka před spuštěním záznamu

Vypisuje se zda je karta vložena nebo není, podle stavu kontaktu CD# (Card Detect). Pokud není karta vložena, není spuštění záznamu povoleno.

```
if (!(GPIOB->IDR & (1<<7)) == 0 ) //detekce vlozene karty
{
    zaznam=0; //karta neni vložena, nespustit zaznam
}
```

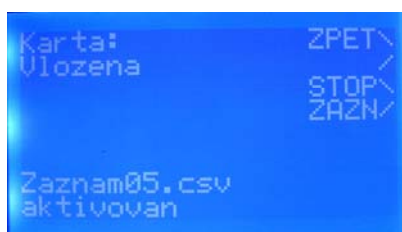
Dále je vypsán název souboru se kterým se pracovalo naposledy. Záznam se spouští tlačítkem START ZAZN. Dojde k nastavení globální proměnné zaznam = 1. Následně se prohledá karta, zda existují soubory předchozích záznamů a zjistí se následující volný název souboru. Název souboru má formát ZaznamXX.csv, kde XX je dvojčíslí od 00 do 99. Volný soubor se poté vytvoří a otevře pro zápis.

```
for(i=0;i<100;i++)
{
    temptostr(i,temp);
    if(i<10) filename[6]='0'; else filename[6]=temp[1];
    filename[7]=temp[2];
    rc = f_open(&Fil, filename, FA_OPEN_EXISTING);
    if(rc) //nalezen dalsi volny nazev
    {
        rc = f_open(&Fil, filename, FA_WRITE | FA_CREATE_ALWAYS);
```

V případě že by došlo k vyčerpání všech 99 možných souborů, program vypíše chybu a zamrzne.

```
if(i==99)
{
    LCDClearScreen();
    LCD_Setpos(0,0);
    LCD_Printf("NO FREE FILENAME",&Font_System5x8);
    for (;;) ;
}
```

Pokud otevření proběhlo v pořádku, vypíše se na displeji, že záznam byl aktivován (obr. 16) , a je možno odejít zpět na hlavní obrazovku. Záznam vlastních dat nezačne probíhat dříve, než se program vrátí z přerušeni do hlavní smyčky.



Obr. 16 Obrazovka po spuštění záznamu

Ukončení záznamu probíhá tak, že se stiskne tlačítko STOP ZAZN na obrazovce záznamu. Dojde k nastavení proměnné zaznam zpět na nulu a k zavolání funkce

```
f_close(&Fil);
```

Tato provede uzavření souboru. Na displeji se objeví znovu obrazovka před spuštěním záznamu (obr. 15).

V hlavní smyčce programu se vždy během zobrazování zároveň plní i buffer Wbuff, který obsahuje hodnoty určené k záznamu. Délka tohoto bufferu je 740 Bajtů. Obsahuje vždy 10 dlouhých a 10 krátkých vět. Celkem je to 10 sekund záznamu. Příklad záznamu:

```
290412;203732;4845.461N;01651.159E; 856; 0; 167; 22; 22
;;; 868;;;
290412;203733;4845.461N;01651.159E; 872; 0; 167; 22; 22
;;; 859;;;
```

Záznam je tvořen hodnotami oddělenými středníkem. GPS modul poskytuje data s opakovacím kmitočtem 1Hz. Otáčky se mění rychleji, proto jsou zaznamenávány 2x za sekundu. Všechny hodnoty mají pevný počet znaků, aby bylo jednodušší skládání dat do bufferu na pevné pozice. Nevyužité znaky jsou vyplněny mezerami. K adresaci bufferu se používá proměnná offset.

Formát zaznamenávaných dat je následující:

datum;čas;zem. šířka;zem. délka;otáčky;rychlost;nad. výška;teplota1;teplota2

Datum je ve formátu DDMMYY.

Čas je ve formátu HHMMSS.

Formáty zeměpisné polohy jsou stejné jak je poskytuje GPS (NMEA), tedy

Zem. šířka je ve formátu SSMM.MMMP (S – stupně, M – minuty, P - polokoule)

Zem. výška je ve formátu SSSMM.MMMP (S – stupně, M – minuty, P – polokoule)

Po naplnění bufferu se testuje, zda je spuštěný záznam. V případě že ano, provede se zápis dalšího bloku 740 B dat na kartu, a inkrementace proměnné fileoffset. Tato proměnná udává aktuální pozici v souboru, která je vždy posunuta pomocí funkce f_seek.

```
if(offset==740)
{
offset=0;
//zapis dalsiho bloku na kartu
if(zaznam)
{
rc = f_write(&Fil, WBuff, 740, &bw);
if (rc) die(rc);
fileoffset=fileoffset+740;
rc = f_lseek(&Fil, fileoffset);
if (rc) die(rc);
}
}
```

V případě, že by došlo k nějaké chybě zápisu (chyba komunikace s kartou, plná karta apod.) je zavolána funkce die(rc). Uživateli se na displeji zobrazí chybové hlášení a program zamrzne.

```
void die (FRESULT rc)
{ //chyba v pristupu na kartu
LCDClearScreen();
LCD_Setpos(0,0);
LCD_Printf("CARD ERROR",&Font_System5x8);

    for (;;) ;
}
```

Pro snížení náročnosti jsou v obsluze souborového systému přijaty určitá zjednodušení. Prvním z nich je to, že datum a čas vytvoření souboru, který si každý soubor nese v attributech je konstantní a neodpovídá skutečnosti. Jelikož je datum a čas přítomný v samotném záznamu, není to na závadu. Druhým zjednodušením je to, že není kontrolováno volné místo na kartě. Důvodem je to, že to v praktickém použití není zapotřebí, protože záznam je velmi malý. Záznam zabírá 740B za 10 sekund, tedy 4.3kB/minutu. Například pětihodinový záznam jízdy má cca 1301kB, což je směšná velikost v porovnání s dnešními kapacitami karet.

ZÁVĚR

Cílem této práce bylo vytvoření jednotky pro monitorování motorového vozidla. Byly vytvořeny softwarové moduly pro obsluhu periférií. Dále byl realizován návrh desky plošných spojů a mechanická konstrukce. Jednotka byla vyrobena a odzkoušena ve vozidle. Data z průběhu jízdy jsou zaznamenávána na SD kartu do formátu csv. Vyhodnocení dat na počítači s pomocí speciální aplikace se nepodařilo, z časových důvodů.

Na jednotce je vstup pro připojení na vstříkovací ventil. Jiné způsoby snímání otáček je možné řešit externími převodníky (tvarovači) na obdélníkový signál s amplitudou 12V. Jeden z nich je popsán, a to snímání kapacitní ze zapalovacího kabelu. Měření teploty je realizováno senzory Dallas 18B20, na jedné sběrnici 1-Wire. Firmware umožňuje volitelný počet senzorů, počítá se však se dvěma. Je měřena vnější teplota okolí a teplota motoru. GPS modul se využívá pro získání polohy, rychlosti vozu, nadmořské výšky, data a času. Jako zobrazovač slouží grafický LCD displej 128x64, s řadičem KS0108. Pro ukládání dat slouží SD nebo MMC karta, jednotka podporuje zjednodušenou implementaci souborového systému FAT32. Pro ovládání záznamu slouží menu, ovládané tlačítky.

POUŽITÉ ZDROJE

- [1] *KS0108 Datasheet*, Samsung 1997. 18 stran. Dostupné na WWW:
< <http://www.techtoys.com.hk/Displays/JHD12864J/ks0108.pdf>> [cit. duben 2011].
- [2] *STM32F100 Datasheet*, STMicroelectronics 2011. 87 stran. Dostupné na WWW:
<http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00251732.pdf> [cit. říjen 2011].
- [3] *STM32F100xx Reference manual*, STMicroelectronics 2011. 675 stran. Dostupné na WWW:
<http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/CD00246267.pdf> [cit. říjen 2011].
- [4] *DS18B20 Datasheet*, Maxim (Dallas) 2008. 22 stran. Dostupné na WWW:
< <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>> [cit. květen 2011].
- [5] *NMEA data*, popis starší verze protokolu.
< <http://www.gpsinformation.org/dale/nmea.htm>> [cit. květen 2011].
- [6] *GPS18LVC Technical specification*, Garmin 2005. 40 stran. Dostupné na WWW:
< http://static.garmincdn.com/pumac/425_TechnicalSpecification.pdf> [cit. květen 2011].
- [7] *LF33 Datasheet*, , STMicroelectronics 2012. 50 stran. Dostupné na WWW:
<http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000546.pdf> [cit. březen 2012].
- [8] *STM32 General-Purpose I/O Example*, Keil 2009. Dostupné na WWW:
< <http://www.keil.com/download/docs/350.asp>> [cit. březen 2012].
- [9] *FatFs Generic FAT File System Module*, ChaN 2011. Dostupné na WWW:
< http://elm-chan.org/fsw/ff/00index_e.html> [cit. duben 2012].