

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

# PALUBNÍ MULTIFUNKČNÍ JEDNOTKA PRO MOTOCYKLY

MULTIFUNCTION BOARD COMPUTER FOR MOTORCYCLES

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

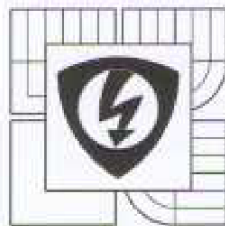
AUTOR PRÁCE  
AUTHOR

Bc. Tomáš Netáhlo

VEDOUCÍ PRÁCE  
SUPERVISOR

prof. Ing. Lubomír Brančík, CSc.

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. Tomáš Netáhlo

**Ročník:** 2

**ID:** 98501

**Akademický rok:** 2011/12

**NÁZEV TÉMATU:**

### Palubní multifunkční jednotka pro motocykly

**POKYNY PRO VYPRACOVÁNÍ:**

Prostudujte možnosti realizace palubního počítače pro motocykl . Přístroj by měl disponovat všemi běžně využívanými funkcemi jako měření otáček , rychlosti, palubního napětí, teploty okolí, teploty motoru, teplota v sání včetně příslušné signalizace přepětí , zvýšené teploty apod . Doplněte přístroj také o nadstandardní funkce jako např . přenos naměřených dat do PC , výškoměr apod . Vypracujte základní návrhovou koncepci .

Vypracujte veškeré konstrukční podklady pro návrh a realizaci zařízení . Zaměřte se při konstrukci zařízení na dobrou odolnost proti vysokofrekvenčnímu rušení . Navrhněte koncepci pro přenos dat do počítače .

Realizujte navrženou palubní jednotku . Jednotku otestujte na uvažovaném motocyklu a vypracujte stručnou zprávu o dosažených výsledcích včetně jednoduchého uživatelského manuálu .

**DOPORUČENÁ LITERATURA:**

[1] HRBÁČEK, J. Komunikace mikrokontroleru s okolím. Praha: BEN - technická literatura, 2000.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 18.5.2012

**Vedoucí práce:** prof. Ing. Lubomír Brančík, CSc.

**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**

Diplomová práce popisuje návrh a realizaci palubní multifunkční jednotky pro motocykly. Úkolem zařízení je měření rychlosti, otáček motoru, teploty okolí, teploty motoru, teploty v sání, včetně příslušné signalizace přepětí a zvýšené teploty. Přístroj je doplněn o nadstandardní funkce, jako je měření náklonu motocyklu, stoupání cesty, záznam trasy a přenos naměřených dat do počítače. Práce obsahuje kompletní řešení hardwarové i softwarové části zařízení.

## **Klíčová slova**

Mikrokontrolér, ATmega128, SPI, TWI, USART, 1-Wire, jazyčkový snímač, otáčkoměr, senzor teploty, akcelerometr, LCD, A/D převodník, RTC, GPS modul, SD karta, USB, DC/DC měnič, stabilizátor napětí, Eagle, AVR studio, jazyk C.

## **Abstract**

Master's thesis deals with a design and a realization of multifunction board computer for motorcycles. The goal of the device is measuring speed, rotation speed of engine, temperature of ambient, temperature of engine, temperature in intake including signalization of overvoltage and hightemperature. The device is completed by above standard function such as measuring titl of motorcycle, climbing of route, route recording and data transmission to the computer. The thesis includes a complete solution of hardware and software part of the device.

## **Keywords**

Microcontroller, ATmega128, SPI, TWI, USART, 1-Wire, reed sensor, engine speed indicator, temperature sensor, accelerometer, LCD, A/D converter, RTC, GPS, SD Card, USB, DC/DC converter, voltage regulator, Eagle, AVR Studio, C language.

NETÁHLO, T. *Palubní multifunkční jednotka pro motocykly*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 90 s. Vedoucí diplomové práce prof. Ing. Lubomír Brančík, CSc..

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma Palubní multifunkční jednotka pro motocykly 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 18. května 2012

.....  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce prof. Ing. Lubomíru Brančíkovi, CSc a konzultantovi Ing. Břetislavu Ševčíkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 18. května 2012

.....  
podpis autora

# Obsah

Úvod.....	11
<b>1 Návrh hardwarové části.....</b>	<b>12</b>
1.1 Mikrokontrolér ATmega128 .....	13
1.1.1 Sériový kanál SPI .....	15
1.1.2 Sériové rozhraní TWI (I <sup>2</sup> C) .....	16
1.1.3 Jednotka USART.....	17
1.1.4 Sběrnice 1-Wire.....	18
1.2 Senzory.....	20
1.2.1 Jazýčkový snímač .....	20
1.2.2 Otáčkoměr .....	21
1.2.3 Senzor teploty DS18B20 .....	22
1.2.4 Akcelerometr MMA7260QT.....	23
1.3 Displej .....	25
1.3.1 Řadič displeje .....	25
1.3.2 Displej RG16080B-YHY-X.....	25
1.3.3 Nastavení kontrastu displeje.....	27
1.4 A/D převodník .....	27
1.5 Obvod hodin reálného času .....	29
1.6 GPS modul .....	30
1.7 SD karta .....	32
1.8 Převodník USB na UART.....	34
1.9 Napájení .....	35
1.9.1 Stabilizátor napětí 7805 STM .....	35
1.9.2 Stabilizátor napětí LT1121-3,3.....	36
1.10 Návrh obvodového zapojení a DPS.....	37
1.10.1 Odolnost vůči vysokofrekvenčnímu rušení.....	37
1.10.2 Návrh obvodového zapojení a DPS .....	38
<b>2 Návrh softwarové části.....</b>	<b>39</b>
2.1 Vývojové prostředky .....	39
2.1.1 Programování mikrokontroléru .....	39
2.1.2 ISP programátor .....	39
2.1.3 Vývojové prostředí .....	40
2.2 Měření rychlosti motocyklu .....	41
2.2.1 Aktuální rychlost .....	41
2.2.2 Maximální rychlost.....	43
2.2.3 Průměrná rychlost.....	43
2.3 Měření denní a celkové vzdálenosti.....	44
2.4 Měření času jízdy .....	44
2.5 Měření otáček motoru .....	45
2.5.1 Aktuální otáčky .....	45
2.5.2 Maximální otáčky.....	46
2.5.3 Průměrné otáčky.....	46
2.6 Měření teploty .....	46
2.7 Měření palubního napětí .....	48

2.8	Měření náklonu a stoupání.....	51
2.9	Zobrazení na displeji.....	53
2.10	Komunikace s obvodem reálného času.....	55
2.11	Komunikace s GPS modulem.....	57
2.12	Ukládání dat.....	59
2.12.1	Paměť EEPROM.....	60
2.12.2	SD karta .....	61
2.13	Přenos naměřených dat do PC .....	62
2.14	Zpracování naměřených dat v PC .....	65
<b>3</b>	<b>Uživatelský manuál.....</b>	<b>67</b>
3.1	Uspořádání palubní jednotky .....	67
3.2	Úvodní nastavení palubní jednotky.....	68
3.2.1	Nastavení času.....	68
3.2.2	Nastavení data .....	68
3.2.3	Nastavení obvodu kola motocyklu .....	69
3.2.4	Nastavení signalizace .....	69
3.2.5	Nastavení typu motoru .....	70
3.2.6	Výstup z úvodního menu.....	70
3.3	Hlavní menu palubní jednotky .....	70
<b>4</b>	<b>Výsledky práce.....</b>	<b>73</b>
4.1	Realizace hardwarové části .....	73
4.2	Realizace softwarové části .....	73
4.3	Testování a technické parametry.....	73
	<b>Závěr.....</b>	<b>76</b>
	<b>Seznam literatury .....</b>	<b>77</b>
	<b>Seznam symbolů, veličin a zkratk .....</b>	<b>81</b>
	<b>Seznam příloh .....</b>	<b>83</b>

## Seznam obrázků

<b>Obr. 1:</b> Blokové schéma palubní multifunkční jednotky. ....	12
<b>Obr. 2:</b> Rozmístění vývodů mikrokontroléru ATmega128 na pouzdře TQFP.....	14
<b>Obr. 3:</b> Propojení mezi masterem a slavem na sběrnici SPI. ....	15
<b>Obr. 4:</b> Připojení zařízení na sériové rozhraní TWI. ....	16
<b>Obr. 5:</b> Platnost dat, startovací a ukončovací podmínka u TWI. ....	17
<b>Obr. 6:</b> Asynchronní rámec jednotky USART v módu 8N1, data 80.....	18
<b>Obr. 7:</b> Zahájení komunikace na sběrnici 1-Wire reset pulsem .....	19
<b>Obr. 8:</b> Vysílání dat na sběrnici 1-Wire .....	19
<b>Obr. 9:</b> Příjem dat na sběrnici 1-Wire. ....	19
<b>Obr. 10:</b> Připojení jazýčkového snímače k mikrokontroléru ATmega128. ....	20
<b>Obr. 11:</b> Spínací charakteristika jazýčkového snímače.....	21
<b>Obr. 12:</b> Připojení otáčkoměru k mikrokontroléru ATmega128.....	22
<b>Obr. 13:</b> Zapojení senzorů teploty DS18B20 na sběrnici 1-Wire. ....	23
<b>Obr. 14:</b> Rozmístění vývodů obvodu MMA7260QT na pouzdru QFN. ....	24
<b>Obr. 15:</b> Schéma modulu s akcelerometrem MMA7260QT.....	24
<b>Obr. 16:</b> Blokové schéma displeje RG16080B-YHY-X.....	26
<b>Obr. 17:</b> Kaskádní zapojení dvou spínaných napěťových měničů LMC7660. ....	27
<b>Obr. 18:</b> Vnitřní blokové schéma A/D převodníku MCP3204 .....	28
<b>Obr. 19:</b> Zapojení A/D převodníku na rozhraní SPI. ....	29
<b>Obr. 20:</b> Zapojení obvodu RTC DS1307 na sériové rozhraní TWI. ....	30
<b>Obr. 21:</b> Zapojení GPS modulu FGMMOPA6B na rozhraní USART.....	31
<b>Obr. 22:</b> Rozmístění vývodů paměťové karty SD.....	32
<b>Obr. 23:</b> Zapojení SD karty na sériový kanál SPI. ....	33
<b>Obr. 24:</b> Zapojení převodníku FT232RL při napájení ze sběrnice USB.....	34
<b>Obr. 25:</b> Celkové schéma zapojení stabilizátoru napětí 7805.....	35
<b>Obr. 26:</b> Celkové schéma zapojení stabilizátorů napětí pro GPS modul. ....	36
<b>Obr. 27:</b> Celkové schéma zapojení stabilizátorů napětí pro SD kartu. ....	36
<b>Obr. 28:</b> Možnosti provedení ohybu napájecího rozvodu .....	37
<b>Obr. 29:</b> Zapojení blokovacího kondenzátoru k mikrokontroléru.....	38
<b>Obr. 30:</b> Zapojení ISP programování mikrokontroléru ATmega128. ....	39
<b>Obr. 31:</b> Blokové schéma programátoru AVRISP mkII.....	40
<b>Obr. 32:</b> Rozmístění vývodů ISP konektoru programátoru AVRISP mkII.....	40
<b>Obr. 33:</b> Proces úspěšného ověření signatury mikrokontroléru ATmega128. ....	41
<b>Obr. 34:</b> Vývojový diagram měření periody a následného výpočtu rychlosti. ....	43
<b>Obr. 35:</b> Vývojový diagram měření ujeté vzdálenosti.....	44
<b>Obr. 36:</b> Vývojový diagram měření času jízdy.....	45
<b>Obr. 37:</b> Vývojový diagram měření teploty pomocí teploměru DS18B20.....	47
<b>Obr. 38:</b> Průběh komunikace A/D převodníku MCP3204 na rozhraní SPI. ....	49
<b>Obr. 39:</b> Vývojový diagram komunikace A/D převodníku MCP3204 na rozhraní SPI.....	50
<b>Obr. 40:</b> Závislosti výstupního napětí X, Y a Z na natočení akcelerometru. ....	51
<b>Obr. 41:</b> Závislosti procent stoupání na vzdálenosti a rozdílu výšky.....	52
<b>Obr. 42:</b> Vývojový diagram inicializace displeje v grafickém módu.....	53
<b>Obr. 43:</b> Zobrazení měřených hodnot na displeji. ....	54
<b>Obr. 44:</b> Průběh zápisu do obvodu RTC na rozhraní TWI.....	55
<b>Obr. 45:</b> Průběh čtení z obvodu RTC na rozhraní TWI.....	56



<b>Obr. 46:</b> Vývojový diagram komunikace s obvodem reálného času.....	56
<b>Obr. 47:</b> Vývojový diagram získání věty GGA z GPS modulu. ....	58
<b>Obr. 48:</b> Vývojový diagram ukládání NMEA vět na SD kartu.....	62
<b>Obr. 49:</b> Výpis naměřených dat z paměti EEPROM v programu Terminal.....	64
<b>Obr. 50:</b> Výpis NMEA věty GGA z SD karty v programu Terminal.....	64
<b>Obr. 51:</b> Mapa zaznamenané trasy vytvořená v aplikaci GPS Visualizer. ....	65
<b>Obr. 52:</b> Výškový profil zaznamenané trasy vytvořený v aplikaci GPS Visualizer.....	65
<b>Obr. 53:</b> Mapa zaznamenané trasy vytvořená v aplikaci Google Earth. ....	66
<b>Obr. 54:</b> Výškový profil zaznamenané trasy vytvořený v aplikaci Google Earth. ....	66
<b>Obr. 55:</b> Celkový vzhled palubní multifunkční jednotky.....	67
<b>Obr. 56:</b> Signalizace při překročení povolené teploty motoru. ....	72
<b>Obr. 57:</b> Palubní jednotka na motocyklu Suzuki V-Strom 650.....	74

## Seznam tabulek

<b>Tab. 1:</b> Napětí na výstupech akcelerometru při měření statického zrychlen .....	25
<b>Tab. 2:</b> Obsazení vývodů displeje RG16080B-YHY-X .....	26
<b>Tab. 3:</b> Obsazení vývodů paměťové karty SD.....	33
<b>Tab. 4:</b> Přehled použitých funkcí z knihovny DS18B20 .....	46
<b>Tab. 5:</b> Organizace paměti teploměru DS18B20.....	48
<b>Tab. 6:</b> Přehled použitých funkcí z knihovny SPI a MCP3204.....	48
<b>Tab. 7:</b> Přehled použitých funkcí z knihovny lc7981 .....	53
<b>Tab. 8:</b> Přehled použitých funkcí z knihovny I <sup>2</sup> C Master Interface .....	55
<b>Tab. 9:</b> Základní typy NMEA vět využívající GPS modul FGPMOPA6B.....	57
<b>Tab. 10:</b> Význam jednotlivých částí věty GGA.....	57
<b>Tab. 11:</b> Přehled použitých funkcí z knihovny UART.....	58
<b>Tab. 12:</b> Přehled použitých funkcí z knihovny avr/eeprom.....	60
<b>Tab. 13:</b> Přehled použitých funkcí z knihovny SD_routines.....	61
<b>Tab. 14:</b> Přehled použitých funkcí z knihovny UART .....	62
<b>Tab. 15:</b> Technické parametry palubní multifunkční jednotky. ....	75

# Úvod

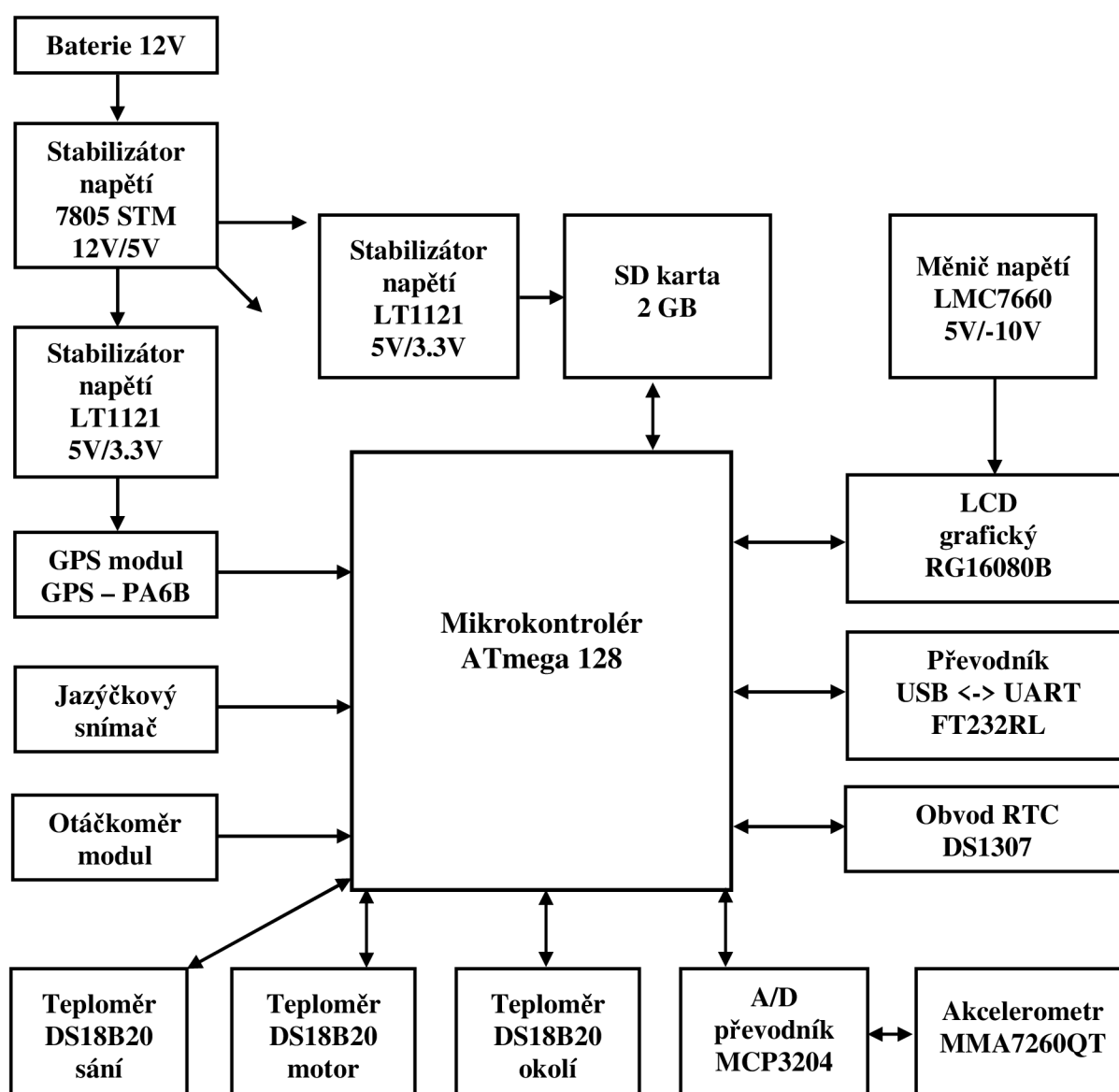
Cílem diplomové práce je návrh a realizace palubní multifunkční jednotky pro motocykly. Navržený přístroj disponuje funkcemi, jako je měření rychlosti motocyklu, otáček motoru, napětí baterie, teploty okolí, teploty motoru, teploty v sání, včetně příslušné signalizace zvýšené teploty a přepětí. Jednotka je navíc doplněna o nadstandardní funkce, jako je měření náklonu motocyklu, stoupání cesty a záznam trasy. Měřené hodnoty jsou ukládány. Následně je možné přenést uložené hodnoty do počítače, kde lze provést podrobnou analýzu naměřených dat.

Práce je rozdělena do dvou hlavních kapitol. První kapitola se věnuje návrhu hardwarové části zařízení. Na začátku je popsán výběr vhodného typu mikrokontroléru a popis použitých rozhraní. Další část pojednává o vlastnostech zvolených senzorů. Je zde uveden popis vybraného jazýčkového snímače, otáčkoměru, teplotních senzorů a akcelerometru. Následně jsou popsány parametry a zapojení všech ostatních obvodů, které provádí vyžadované funkce palubní jednotky. Závěr kapitoly pojednává o návrhu napájení a také návrhu obvodového zapojení a desky plošných spojů.

Následující kapitola je zaměřena na softwarovou část zařízení. Na začátku je uveden popis použitého programátoru a vývojového prostředí. Dále jsou popsány postupy programování jednotlivých funkcí přístroje, včetně všech použitých knihoven. V závěru práce je uveden uživatelský manuál pro ovládání palubní jednotky, který obsahuje popis základních nastavení a funkcí přístroje.

# 1 Návrh hardwarové části

Blokové schéma celého zařízení znázorňuje obr. 1. Pro snímání všech potřebných hodnot obsahuje multifunkční jednotka jazýčkový snímač, otáčkoměr, tři snímače teploty, A/D převodník, akcelerometr a GPS modul. Jazýčkový snímač slouží k měření rychlosti pomocí magnetu umístěného na předním kole motocyklu. Modul otáčkoměru snímá otáčky motoru. Sensory teploty měří teplotu okolí, teplotu motoru a teplotu v sání. Akcelerometr měří náklon motocyklu a stoupání cesty. GPS modul slouží k zaznamenávání projeté trasy. Naměřené hodnoty jsou přenášeny ze senzorů do mikrokontroléru, kde se dále zpracovávají. Tyto hodnoty se zobrazují na grafickém displeji spolu s hodnotami o aktuálním čase a datu. Změřené hodnoty jsou ukládány do paměti EEPROM a na SD kartu. Uložené hodnoty je možné přenést prostřednictvím USB rozhraní do počítače.



Obr. 1: Blokové schéma palubní multifunkční jednotky.

Palubní multifunkční jednotka obsahuje následující funkce:

- funkce rychlosti – aktuální, průměrná a maximální rychlost,
- funkce měření otáček - aktuální, průměrné a maximální otáčky motoru,
- funkce teploty – aktuální teplota okolí, motoru a v sání,
- funkce náklonu - aktuální a maximální náklon motocyklu
- funkce sklonu - aktuální a maximální sklon cesty,
- funkce napětí – aktuální palubní napětí a úroveň nabití baterie,
- funkce času – denní čas jízdy, aktuální čas a datum,
- funkce vzdálenosti – denní ujetá vzdálenost a celková ujetá vzdálenost,
- funkce polohy – aktuální zeměpisná šířka, délka a nadmořská výška,
- funkce trasy – ukládání záznamu trasy a jejího profilu z GPS modulu.

## 1.1 Mikrokontrolér ATmega128

K řízení celé multifunkční jednotky slouží mikrokontrolér. K mikrokontroléru jsou připojeny všechny ostatní obvody, je jsou jazýčkový snímač, otáčkoměr, snímače teploty, akcelerometr, LCD displej, A/D převodník, obvod hodin reálného času, GPS modul, SD karta a také převodník USB na UART. V současné době je nabídka mikrokontrolérů velmi široká. Jednotlivá provedení se od sebe mohou odlišovat šířkou registrů a sběrnice, počtem zabudovaných periférií, počtem vstupně/výstupních portů, nebo velikostí programové a datové paměti. Mezi největší výrobce mikrokontroléru patří firmy ATMEL, FREESCALE, INTEL, MICROCHIP atd. Při návrhu multifunkční jednotky byl vybrán mikrokontrolér od výrobce ATMEL z řady ATmega typ ATmega128, viz [1]. V porovnání s nižšími typy mikrokontrolérů z řady ATmega, nabízí ATmega128 větší velikost programové i datové paměti a také paměti EEPROM. Kromě velikosti paměti byl při návrhu zařízení dalším důležitým požadavkem dostatečný počet vstupně/výstupních portů pro připojení všech potřebných obvodů. Při vývoji aplikací s mikrokontroléry ATMEL je také výhodou volně dostupné vývojové prostředí AVR Studio, které je po registraci ke stažení na stránkách firmy ATMEL. Velmi užitečná je také možnost sériového programování ISP mikrokontroléru přímo v aplikaci.

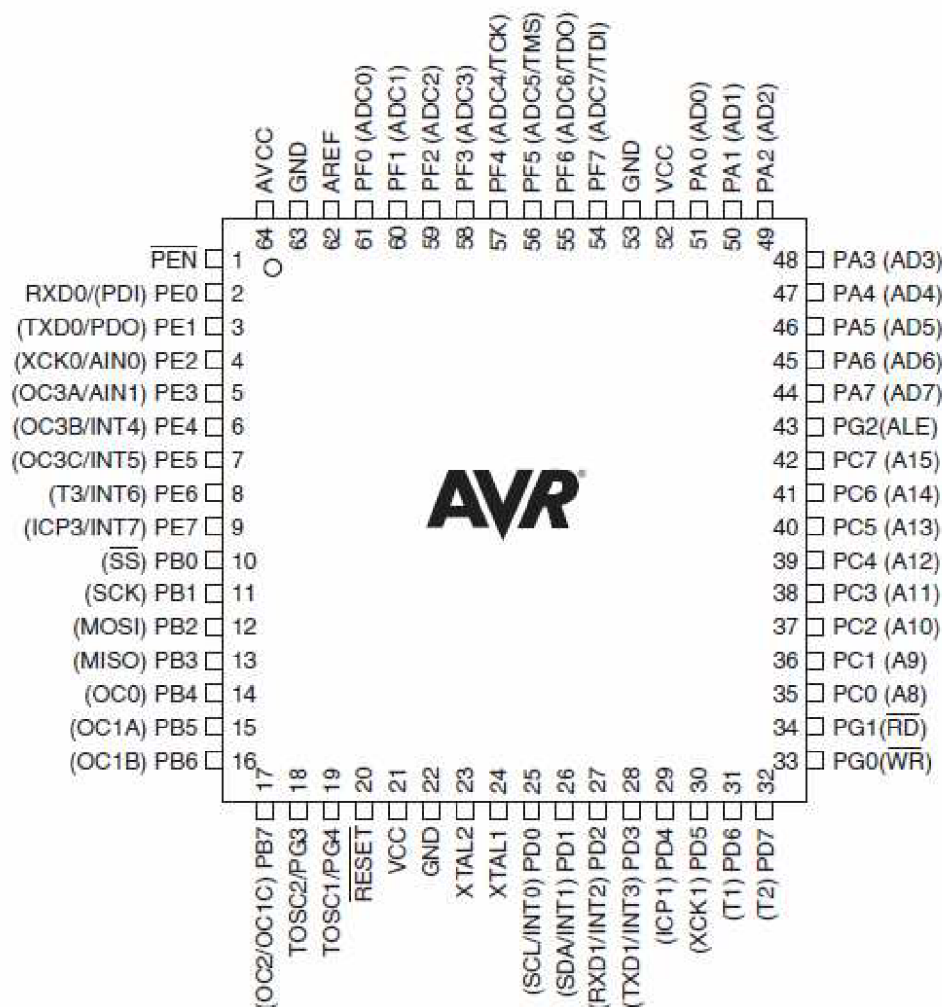
Vybraný mikrokontrolér ATmega128 je 8bitový mikrokontrolér založený na AVR architektuře. Jedná se o Harvardskou architekturu s oddělenou pamětí pro program a data. Procesor je postaven na architektuře RISC (Reduced Instruction Set Computer), má tedy redukovanou instrukční sadu. Instrukce jsou vykonávány v jediném hodinovém cyklu.

Základní vlastnosti mikrokontroléru ATmega128:

- instrukční soubor obsahuje 130 instrukcí,
- 32 registrů o délce 8 bitů,
- hodinový kmitočet 0 až 16 MHz,
- maximální výpočetní výkon 16 MIPS,
- programová paměť typu FLASH s kapacitou 128 KB, počet přeprogramování 10 000,
- datová paměť SRAM s kapacitou 4 KB,
- datová paměť EEPROM s kapacitou 4 KB, počet přeprogramování 100 000,

- paměť FLASH a EEPROM jsou programovatelné přímo v systému pomocí rozhraní SPI nebo JTAG,
- dva 8bitové čítače/časovače, dva 16bitové čítače/časovače,
- 8kanálový 10bitový A/D převodník,
- dva 8bitové PWM kanály, šest kanálů s programovatelným rozlišením 1 - 16 bitů,
- jednotky TWI (I<sup>2</sup>C), 2x USART, SPI,
- interní kalibrovaný RC oscilátor,
- napájecí napětí obvodu 4,5-5,5 V.

Mikrokontrolér obsahuje šest kompletních 8bitových vstupně/výstupních portů, které jsou označeny zkratkami PA až PF. Všechny porty mohou pracovat jako obousměrné. U mikrokontrolérů AVR se provádí zřetězené vykonávání instrukcí, neboli jednofázový pippeling. Princip spočívá v tom, že nejprve se v programové paměti načte první instrukce, současně s tím, jak se tato instrukce v následujícím hodinovém cyklu provádí, se předvybírá druhá instrukce, viz [2]. Další důležitou vlastností, která velmi výrazně zvyšuje výpočetní výkon mikrokontrolérů ATmega, je jednotná délka instrukce 16 bitů. Rozmístění jednotlivých vývodů na pouzdře TQFP uvádí obr. 2.



**Obr. 2:** Rozmístění vývodů mikrokontroléru ATmega128 na pouzdře TQFP (převzato z [1]).

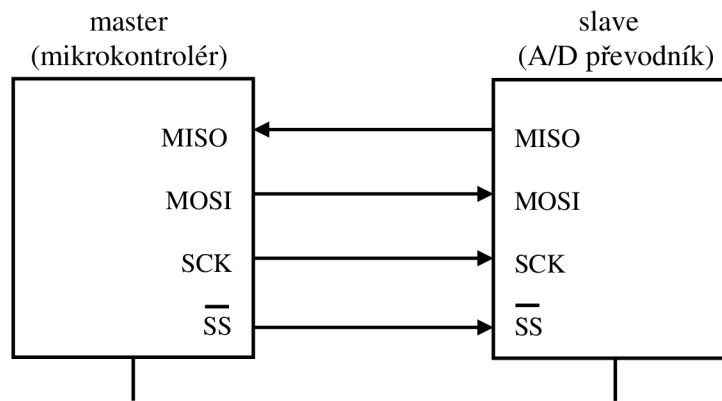
### 1.1.1 Sériový kanál SPI

Pro vysokorychlostní přenos dat mezi mikrokontrolérem a periferními zařízeními se používá sériový kanál SPI (Serial Peripheral Interface), viz [3]. Samozřejmostí je podmínka, že periferní zařízení musí být vybaveno SPI rozhraním. V zapojení multifunkční jednotky jsou tyto zařízení reprezentovány A/D převodníkem a SD kartou, viz kapitoly 1.4 a 1.7. Další funkcí sběrnice SPI je možnost sériového programování mikrokontroléru, viz kapitola 2.1.1.

Základní vlastnosti kanálu SPI mikrokontroléru ATmega128:

- plný duplex (schopnost současně přijímat a vysílat),
- třívodičový synchronní přenos dat,
- může pracovat jako master nebo slave,
- 7 programovatelných rychlostí, maximální přenosové rychlosti až 8 Mb/s,
- lze volit pořadí bitů (LSB až MSB, MSB až LSB),
- příznak konce a kolize přenosu.

Ke komunikaci mezi zařízením typu master a slave se používá trojice vodičů. Jedná se o vodiče označené zkratkami MISO, MOSI a SCK. Čtvrtý vývod SS (Slave Select) slouží k výběru zařízení, které je nastaveno jako slave. Propojení mezi zařízením master a slave udává obr. 3.



**Obr. 3:** Propojení mezi masterem a slavem na sběrnici SPI (převzato z [2]).

Přenos dat na sběrnici SPI je synchronizován pomocí hodinového signálu, který generuje master na vývodu SCK (Serial Clock). Synchronizaci lze nastavit buď na vzestupnou, nebo sestupnou hranu hodinového signálu. Zápis dat do datového registru SPI startuje hodinový signál. Vývod MOSI (Master Out Slave In) slouží k vysílání dat z masteru do slavu. Po vyslání celého bajtu dojde k zastavení hodinového signálu a nastavení příznaku konce přenosu SPIF v registru SPSR. Vývod MISO (Master In Slave Out) slouží k vyslání dat ze slavu do masteru.

K řízení sériového kanálu SPI se používají tři registry. Jedná se o řídicí registr SPCR pro nastavení rychlosti a formátu přenosu, stavový registr SPSR obsahující příznaky dokončení a kolize přenosu a datový registr SPDR pro vyslání a příjem dat, viz [2].

### 1.1.2 Sériové rozhraní TWI (I<sup>2</sup>C)

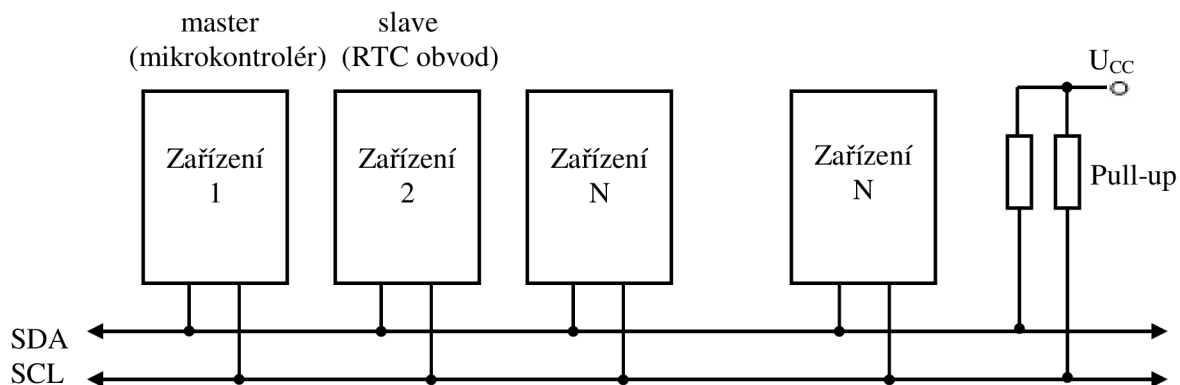
Sběrnice TWI je dalším typem sériového rozhraní. Toto rozhraní je známější spíše pod označením I<sup>2</sup>C, což je rozhraní vyvinuté firmou Philips Semiconductor. Obě rozhraní jsou však plně kompatibilní. Rozhraní TWI je použito ke komunikaci s obvodem hodin reálného času, jehož popis je uveden v kapitole 1.5.

Narozdíl od sériového kanálu SPI neumožňuje rozhraní TWI duplexní přenos. K přenosu dat slouží pouze jeden datový vodič. Každé zařízení na rozhraní TWI musí mít individuální adresu pro svou identifikaci. Přenos dat na sběrnici je definován TWI protokolem, viz [2].

Základní vlastnosti sériového rozhraní TWI mikrokontroléru ATmega128:

- podpora práce v režimu master nebo slave,
- plně nastavitelná adresa slave obvodů,
- 7 bitový adresní prostor umožňuje až 128 různých slave adres,
- zařízení může pracovat jako vysílač i přijímač,
- rychlost přenosu dat až 400 kHz,
- potlačení šumu pro obě linky sběrnice.

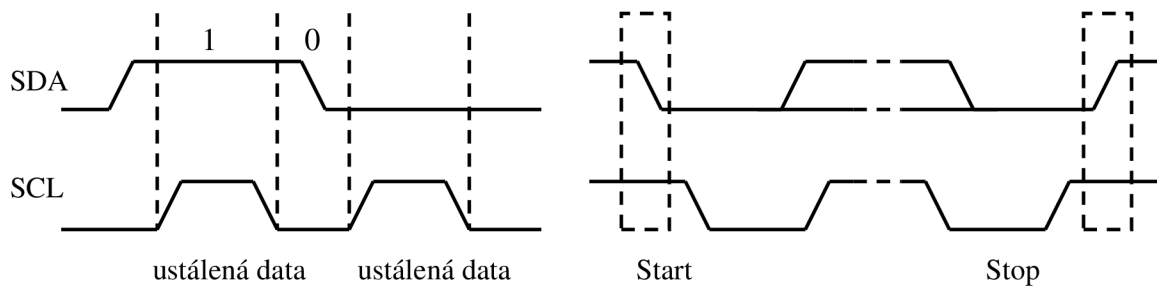
Ke komunikaci mezi zařízeními typu master a slave se používá dvojice vodičů. Jedná se o datový vodič SDA a vodič s hodinovým signálem SCL. Komunikace je tedy řízena podobně jako u SPI pomocí hodinového signálu. Připojení jednotlivých zařízení na sériové rozhraní uvádí obr. 4. Oba vodiče jsou připojeny na napájecí napětí přes pull-up rezistory, které zajišťují vysokou úroveň v klidovém stavu.



**Obr. 4:** Připojení zařízení na sériové rozhraní TWI (převzato z [2]).

Master řídí přenos dat a generuje hodinový signál na vývodu SCL. Každý přenášený bit je sjednocen s jedním pulzem hodin. Úroveň na datovém vodiči musí být konstantní v okamžiku vysoké úrovně hodinového signálu. Výjimku tvoří pouze startovací a ukončovací podmínka. Startovací podmínka zahajuje přenos dat, poté následuje adresní paket a za ním jeden nebo více datových paketů ukončených ukončovací podmínkou. Průběh startovací a ukončovací podmínky je uveden na obr. 5.





**Obr. 5:** Platnost dat, startovací a ukončovací podmínka u TWI (převzato z [2]).

Adresní pakety mají délku 9 bitů a skládají se ze 7bitové adresy, řídicího bitu R/W a potvrzovacího bit ACK. Datové pakety mají také délku 9 bitů a skládají se z 8bitových dat a potvrzovacího bit ACK. Rozhraní TWI je založeno na přerušení. K řízení se používá registr přenosové rychlosti TWBR, řídicí registr TWCR, stavový registr TWSR, datový registr TWDR a adresový registr TWAR, viz [1], [6].

### 1.1.3 Jednotka USART

Pro komunikaci mikrokontroléru s GPS modulem a s převodníkem FT232RL se používá jednotka USART, viz kapitoly 1.6 a 1.8. Jedná se o univerzální synchronní a asynchronní sériový přijímač a vysílač. Mikrokontrolér ATmega128 obsahuje dvě jednotky USART označené zkratkami USART0 a USART1. Jednotka USART0 umožňuje pouze asynchronní režim.

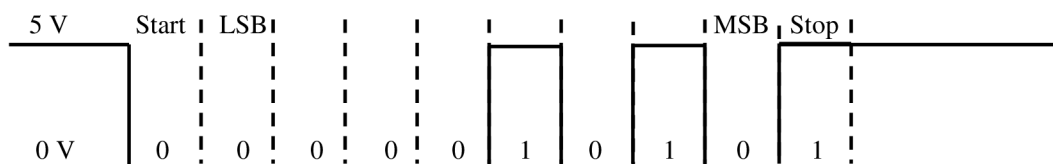
Základní vlastnosti jednotky USART mikrokontroléru ATmega128:

- plně duplexní provoz (může současně přijímat i vysílat),
- asynchronní nebo synchronní režim,
- při asynchronním režimu může pracovat jako master nebo slave,
- generátor sudé/liché parity pro vysílač a hardwarové testování parity pro přijímač,
- volitelný paritní bit (sudá Even, lichá Odd nebo žádná parita),
- detekce ztráty znaku, chyby rámce, falešného start bitu.

Jednotka USART se skládá ze tří bloků:

- Generátor hodin – používá se v synchronním režimu. Obsahuje synchronizační logiku pro vývod hodin XCK.
- Vysílač – před vysláním i příjmem dat je nutné mít nastaveny všechny řídicí registry. Před vysláním dat se nejprve provádí kontrola, jestli je výstupní buffer UDR připraven přijmout nová data. Data k odeslání poté stačí zapsat do datového registru UDR. Po odeslání dat se nastaví příznak TXC v registru UCSRA.
- Přijímač – přijaté data je možné přečíst v datovém registru UDR. Po příjmu dat se nastaví příznak RXC v registru UCSRA.

Pro zabezpečení správného příjmu dat se používá předem zvolená struktura datového rámce. Rámec pro asynchronní přenos dat se skládá z jednoho start bitu ve stavu logické nuly, z 5 až 9 datových bitů v pořadí od LSB po MSB, z volitelného paritního bitu a jednoho nebo dvou stop bitů ve stavu logické jedničky. Struktura asynchronního rámce v módu 8N1 je uvedena na obr. 6. Shodné nastavení rámce používá vybraný GPS modul. První číslo módu značí 8 datových bitů, N je žádná parita a číslo jedna odpovídá 1 stop bitu.



**Obr. 6:** Asynchronní rámec jednotky USART v módu 8N1, data 80.

Pro bezchybný přenos dat musí mít vysílač i přijímač shodné nastavení datového rámce a rovněž přenosové rychlosti. Hodnotu přenosové rychlosti je nutné určit s ohledem na frekvenci mikrokontroléru  $f_{osc}$  a uložit do registru UBRRn, viz [1]:

$$UBRRn = \frac{f_{osc}}{16 \cdot BAUD} - 1, \quad [ -, \text{ Hz, Bd } ] \quad (1)$$

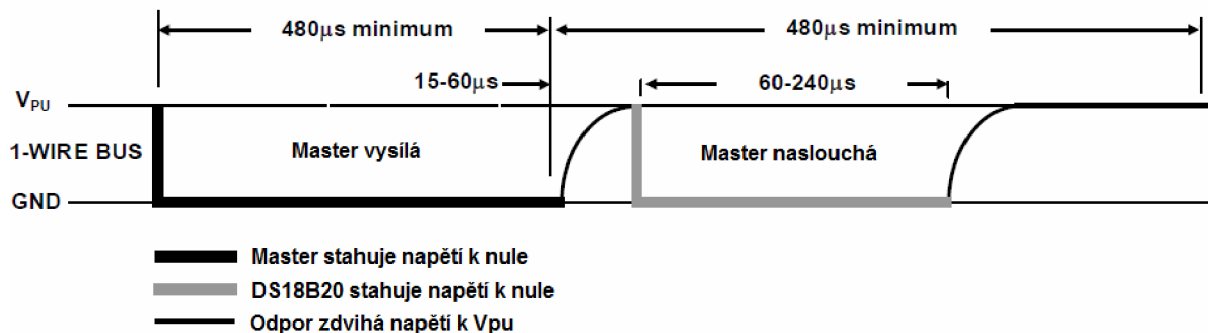
kde  $f_{osc}$  hodinový signál mikrokontroléru;  
 $BAUD$  přenosová rychlost.

#### 1.1.4 Sběrnice 1-Wire

Sběrnice 1-Wire umožňuje připojit k mikrokontroléru několik zařízení prostřednictvím pouhého jednoho vodiče, tato sběrnice byla navržena firmou Dallas Semiconductor, viz [4], [8]. Pomocí sběrnice 1-Wire jsou k mikrokontroléru připojeny tři teplotní senzory DS18B20, jejichž popis je uveden v kapitole 1.2.3.

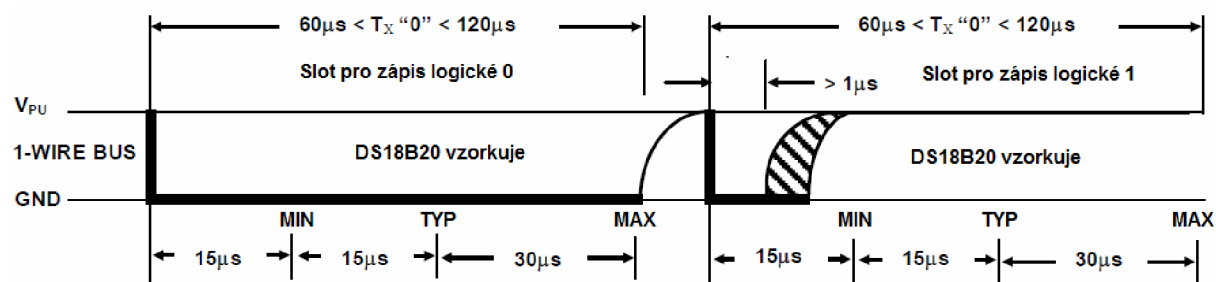
Sběrnice má vždy jeden řídicí obvod (master) a jeden nebo více podřízených obvodů (slave). Všechny obvody jsou připojeny na společnou zem a také paralelně na společný datový vodič, který bývá připojen přes rezistor 4,7 k $\Omega$  na napájecí napětí, a tím nastaví sběrnici na vysokou úroveň.

Data jsou na sběrnici vysílána v tzv. „time slotech“. Jeden slot je dlouhý 60 až 120  $\mu\text{s}$  a během jednoho slotu je vysílán nebo přijat jeden bit. Mezi jednotlivými sloty musí být mezera minimálně 1  $\mu\text{s}$ . Sloty lze rozdělit na čtyři druhy a to zápis 1, zápis 0, čtení 1 a čtení 0. Komunikace na sběrnici je realizována jako asynchronní a poloduplexní. Komunikaci zahajuje vždy master reset pulsem, průběh celé inicializace s reset pulsem je uveden na obr. 7.



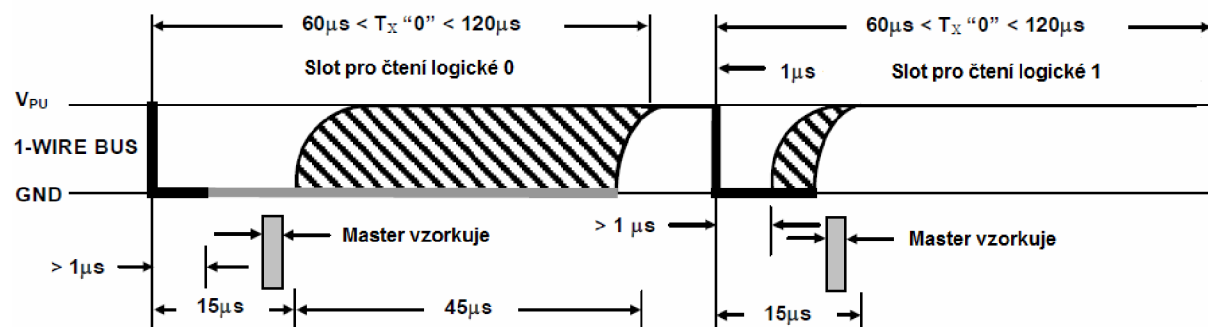
**Obr. 7:** Zahájení komunikace na sběrnici 1-Wire reset pulsem (převzato z [8]).

Master nejprve nastaví datový vodič na nízkou úroveň a čeká 480  $\mu\text{s}$ , poté sběrnici uvolní a čeká 70  $\mu\text{s}$ . Odpor vrátí sběrnici zpět na vysokou úroveň. Pokud je na sběrnici připojeno nějaké zařízení, tak detekuje vzestupnou hranu a po prodlevě 15 až 60  $\mu\text{s}$  nastaví datový vodič na 60 až 240  $\mu\text{s}$  na nízkou úroveň. Jestliže se zařízení správně ohlásí, může master začít vysílat a přijímat data. Průběh při vysílání dat je uveden na obr. 8.



**Obr. 8:** Vysílání dat na sběrnici 1-Wire (převzato z [8]).

Při zápisu logické nuly master stáhne datový vodič na nízkou úroveň a čeká 60  $\mu\text{s}$ , poté sběrnici uvolní a čeká 10  $\mu\text{s}$ . Při zápisu logické jedničky master stáhne datový vodič na nízkou úroveň a čeká 6  $\mu\text{s}$ , poté sběrnici uvolní a čeká 64  $\mu\text{s}$ . Průběh pro čtení dat je uveden na obr. 9.



**Obr. 9:** Příjem dat na sběrnici 1-Wire (převzato z [8]).

Při čtení dat master nejprve nastaví sběrnici na nízkou úroveň a čeká minimálně 1  $\mu$ s. Poté sběrnici uvolní a čeká 9  $\mu$ s. Master přečte sběrnici, její stav udává přečtený bit, buď logická nula, nebo logická jednička.

Každé zařízení na sběrnici 1-Wire má v sobě paměť ROM, která obsahuje 64bitové unikátní číslo, pomocí kterého je možné od sebe jednotlivé zařízení odlišit. Unikátní číslo se skládá z typu zařízení, sériového čísla a CRC kódu. Pokud je na jedné sběrnici připojeno více zařízení, je třeba po reset pulsu vyslat příkaz Match ROM, pak 64bitový kód zařízení a poté se posílají příkazy.

## 1.2 Senzory

### 1.2.1 Jazýčkový snímač

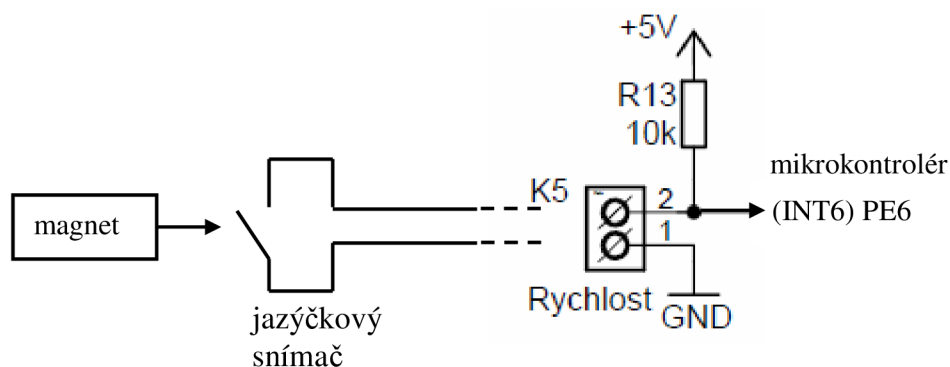
Pro měření rychlosti motocyklu je nejvhodnější použít některý ze senzorů citlivých na magnetické pole. Tyto senzory můžeme rozdělit dle využívaného fyzikálního jevu na indukční senzory, magnetorezistivní senzory, Hallovy senzory a jazýčkové snímače, viz [5]. Pro detekci impulsů a následné měření rychlosti jsou nejvhodnější Hallovy senzory a jazýčkové snímače.

Hallovy senzory jsou založeny na principu Hallova jevu. Jestliže prochází magnet v blízkosti Hallova senzoru, vzniká na něm Hallovo napětí ve formě impulsů. Tyto impulzy lze dále zpracovávat a měřit tak rychlost, počet otáček za minutu a další veličiny. Hallovy senzory se tedy využívají zejména pro měření a detekci pohybu a pro měření magnetického pole.

Jazýčkové snímače pracují na principu mechanického spínání a rozpínání kontaktu v závislosti na změně polohy magnetu, viz [6]. Výhodou jazýčkových spínačů je jejich jednoduchá konstrukce a spolehlivost. Proto se v současné době využívají ve velké míře pro měření rychlosti a otáček.

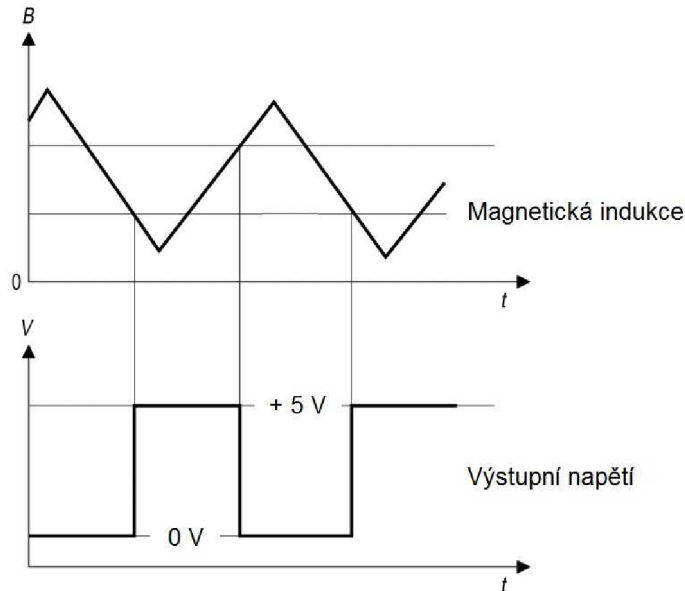
Při návrhu palubní multifunkční jednotky byl pro měření rychlosti vybrán právě jazýčkový snímač. Zvolen byl jak kvůli spolehlivosti, tak také dobré dostupnosti a parametrům, které jsou vhodné pro měření rychlosti.

Snímač je k mikrokontroléru ATmega128 připojen na pin externího přerušení PE6. Zapojení snímače k mikrokontroléru udává obr. 10.



**Obr. 10:** Připojení jazýčkového snímače k mikrokontroléru ATmega128.

Spínací charakteristika jazýčkového snímače je uvedena na obr. 11. Na výstupu snímače je obdélníkový signál. Při vysoké úrovni magnetické indukce je na výstupu nízká úroveň napětí a naopak při nízké úrovni magnetické indukce je úroveň napětí vysoká. Mikrokontrolér měří pomocí čítače/časovače dobu periody mezi dvěma impulsy a následně počítá rychlost a další funkce.



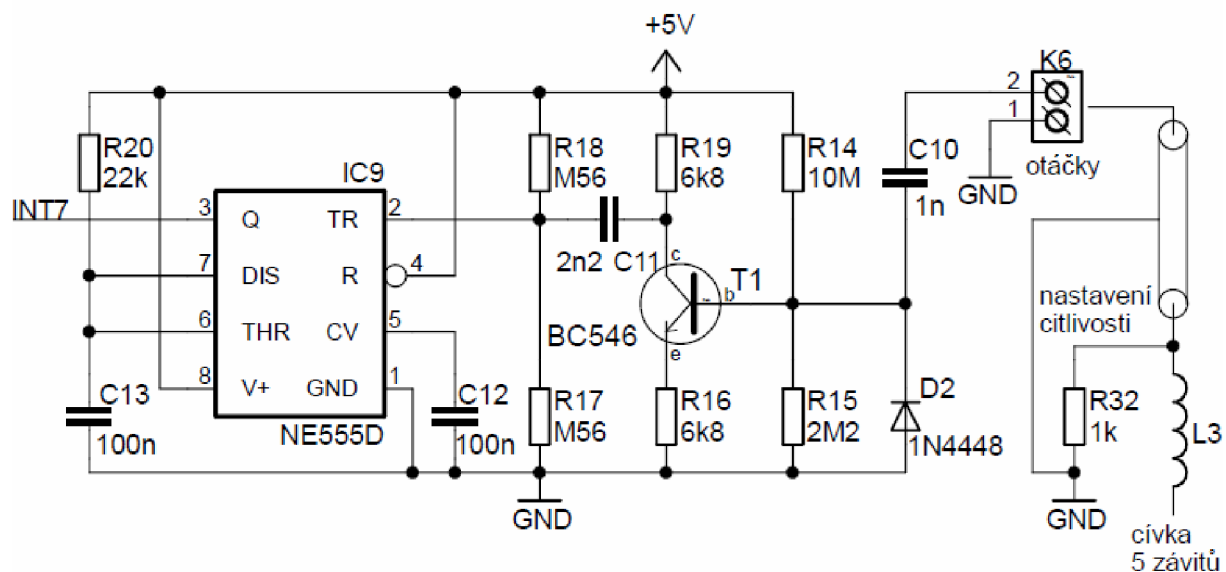
**Obr. 11:** Spínací charakteristika jazýčkového snímače.

### 1.2.2 Otáčkoměr

Pro měření otáček motoru bylo nutné najít řešení, které by umožňovalo přesné měření otáček bez toho, aniž by se muselo zasahovat do konstrukce motoru. Při použití snímačů magnetického pole, by bylo nutné umístit na hřídel motoru magnet, což by nebylo snadné a z technického hlediska i nevhodné.

Proto byl vybrán způsob měření periody mezi dvěma výboji na zapalovací cívce, viz [7]. Při tomto způsobu měření se snímají vysokonapět'ové impulsy, které procházejí vodičem ke svíčke motocyklu. Celkové schéma zapojení otáčkoměru je na obr. 12.

Základním prvkem tohoto zapojení je cívka s pěti závity, která je navinuta kolem kabelu vedoucího z cívky na svíčky motocyklu. Cívka je připojena k otáčkoměru pomocí stíněného kabelu, který je vyveden na svorky označené K6. Mezi cívku a stíněný kabel je ještě připojen rezistor o velikosti 1 k $\Omega$ , sloužící k nastavení citlivosti snímače. Pro zesílení napět'ových impulsů slouží tranzistor BC546. Zesílené impulsy jsou poté přivedeny do obvodu NE555D, který tyto impulsy vytvaruje do tvaru obdélníka a upraví na logické úrovně 0 V a 5 V. Upravené impulsy jsou přivedeny na pin mikrokontroléru PE7. Pomocí externího přerušení INT7 reagujícího na náběžnou hranu se měří délka periody mezi dvěma výboji na zapalovací cívce. Změřená perioda se přepočítává na počet otáček za minutu.



**Obr. 12:** Připojení otáčkoměru k mikrokontroléru ATmega128 (převzato z [7]).

### 1.2.3 Senzor teploty DS18B20

Pro měření teploty okolí, teploty sání a motoru byl vybrán senzor teploty typ DS18B20 od výrobce Dallas Semiconductor, viz [8]. Senzor komunikuje s mikrokontrolérem ATmega128 pomocí sběrnice 1-Wire, jejíž popis je uveden v kapitole 1.1.4. Teplotní rozsah senzoru je od  $-55\text{ }^{\circ}\text{C}$  do  $+125\text{ }^{\circ}\text{C}$ , což je pro dané použití zcela dostačující, protože teplota motoru i teplota v sání motocyklu dosahují nižších hodnot. Všechny tři senzory měří aktuální teplotu a také slouží jako signalizace zvýšené teploty.

Základní parametry obvodu DS18B20:

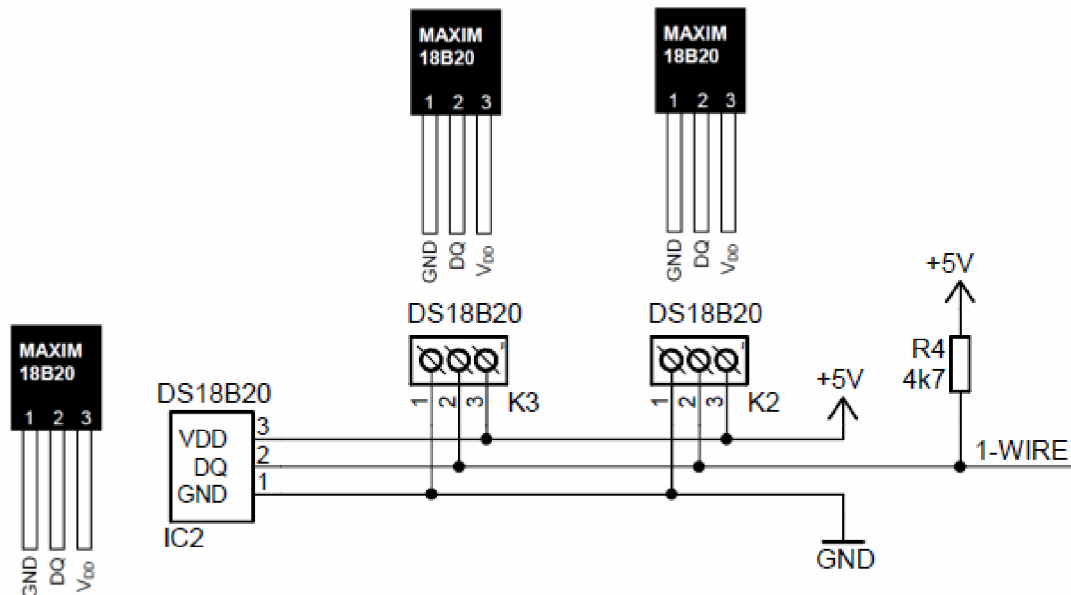
- komunikace po sběrnici 1-Wire,
- rozsah měření teploty od  $-55\text{ }^{\circ}\text{C}$  do  $+125\text{ }^{\circ}\text{C}$ ,
- přesnost  $\pm 0,5\text{ }^{\circ}\text{C}$  v rozsahu od  $-10\text{ }^{\circ}\text{C}$  do  $+85\text{ }^{\circ}\text{C}$ , v celém rozsahu  $\pm 2\text{ }^{\circ}\text{C}$ ,
- rozlišení teplotního senzoru je od 9 do 12 bitů,
- programovatelný alarm teploty,
- každé zařízení je rozlišeno unikátním 64bitovým sériovým kódem,
- napájecí napětí od 3 V do 5,5 V.

Pouzdro obvodu DS18B20 obsahuje celkem tři vývody, jedná se o vývody  $D_Q$  (datový vstup/výstup),  $V_{DD}$  (napájení) a GND (zem), Vývod  $D_Q$  je softwarově nastavitelný jako bistabilní výstup, nebo jako komunikační linka. Výstup je svým charakterem s otevřeným kolektorem.

Princip měření teploty je takový, že uvnitř senzoru jsou dva oscilátory. Jeden oscilátor má nízký teplotní koeficient a druhý oscilátor naopak vysoký. Jakmile dá mikrokontrolér příkaz k zahájení měření teploty, spustí se čítání obou oscilátorů. Tento proces trvá až 750 ms (dle teploty) a výsledkem je 12bitová hodnota odpovídající aktuální změřené teplotě. Tato hodnota je uložena ve vnitřním registru senzoru.

Pro signalizaci snížené, nebo zvýšené teploty se používají registry TL a TH. Změřená teplota se porovnává s hodnotou nastavenou v těchto registrech a pokud jsou tyto hodnoty překročeny, je nastaven příznak alarmu uvnitř senzoru.

Zapojení senzorů na sběrnici 1-Wire je velmi jednoduché. Mikrokontrolér pracuje jako master a senzor jako slave. Vývod D<sub>Q</sub> se připojí k jednomu pinu mikrokontroléru. Pro správnou funkci je nutné mezi vývod D<sub>Q</sub> a V<sub>DD</sub> připojit rezistor o velikosti 4,7 kΩ. Zapojení všech tří senzorů na sběrnici 1-Wire znázorňuje obr. 13. Senzor pro měření teploty okolí je umístěn přímo na desce multifunkční jednotky. Sensory pro měření teploty motoru a sání jsou připojeny k jednotce přes svorkovnice.



**Obr. 13:** Zapojení senzorů teploty DS18B20 na sběrnici 1-Wire.

#### 1.2.4 Akcelerometr MMA7260QT

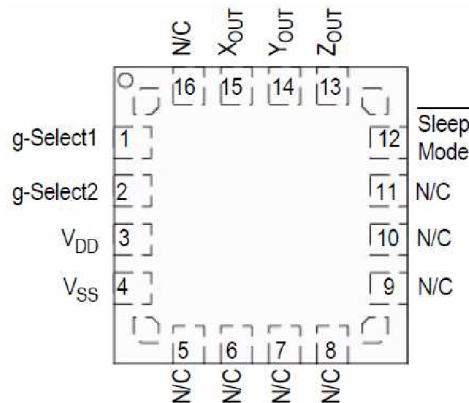
Pro měření náklonu motocyklu a stoupání cesty v procentech obsahuje multifunkční jednotka akcelerometr. Akcelerometry jsou senzory, které jsou schopné měřit statické i dynamické zrychlení a převádět jejich velikost na elektrický signál.

Jako vhodný typ byl vybrán akcelerometr MMA7260QT od výrobce Freescale, viz [9]. Jedná se o tříosý akcelerometr, tzn., že měří zrychlení ve všech třech osách X, Y i Z. Akcelerometr má analogový výstup a měnitelný měřicí rozsah. Senzor má již z výroby nastavenou nulovou hladinu a zlomový kmitočet.

## Základní parametry obvodu MMA7260QT:

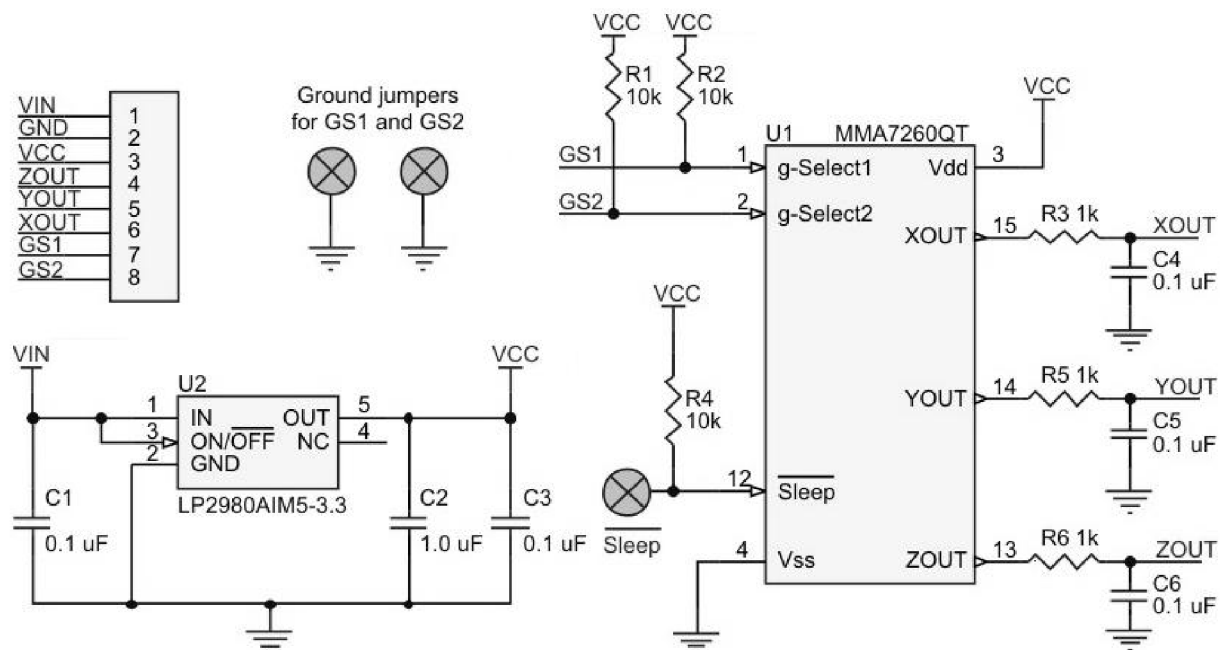
- měnitelný rozsah 1,5 g, 2 g, 4 g a 6 g, vývody g-Select,
- maximální citlivost 800 mV/g na rozsahu 1,5 g,
- nelinearita  $\pm 1 \%$  z rozsahu,
- spotřeba 500 $\mu$ A, v sleep módu jen 3  $\mu$ A,
- napájecí napětí od 2,2 V do 3,6 V.

Pouzdro obvodu MMA7260QT obsahuje celkem 16 vývodů, vzhled pouzdra s rozmístěním jednotlivých vývodů udává obr. 14.



**Obr. 14:** Rozmístění vývodů obvodu MMA7260QT na pouzdru QFN (převzato z [9]).

Akcelerometr se vyrábí v miniaturním QFN pouzdře, u kterého by byl problém s pájením obvodu, proto byl použit modul od výrobce Pololu Robotics & Electronics, viz [10]. Modul obsahuje již zmíněný akcelerometr a další doplňující obvody. Celý modul lze napájet napětím od 3,6 V do 15 V, protože obsahuje stabilizátor napětí LP2980AIM5-3.3. Na výstupech všech tří os X, Y i Z jsou zapojeny RC filtry, jak je doporučeno v datasheetu akcelerometru. Celkové schéma zapojení modulu udává obr. 15.



**Obr. 15:** Schéma modulu s akcelerometrem MMA7260QT (převzato z [10]).



Výstupy akcelerometru jsou analogové, proto se pro převod napětí ze všech tří os akcelerometru používá A/D převodník, jehož popis je uveden v kapitole 1.4. Velikost napětí na výstupech X, Y a Z akcelerometru při měření statického zrychlení na rozsahu 1,5 g udává tabulka č. 1.

**Tab. 1:** Napětí na výstupech akcelerometru při měření statického zrychlení (převzato z [10]).

	- 1 g	0 g	1 g
U <sub>x</sub> [V]	0,85	1,65	2,45
U <sub>y</sub> [V]	0,85	1,65	2,45
U <sub>z</sub> [V]	0,85	1,65	2,45

Jak již bylo zmíněno, akcelerometr může pracovat ve dvou módech, tedy měření statického a dynamického zrychlení. Pro měření náklonu motocyklu a stoupání cesty je využito pouze měření statického zrychlení. Proto bylo nutné na jednotlivé kanály zapojit dolní propust, která odfiltruje krátkodobé signály a impulsy způsobené vibracemi, tedy dynamickým zrychlením.

## 1.3 Displej

### 1.3.1 Řadič displeje

V dnešní době je většina grafických displejů řízena řadičem. Nejčastěji používanými řadiči jsou KS108 a Toshiba T6963C. Pro displeje s větším rozlišením se používají např. řadiče Sanyo LC7981, nebo Epson SED 1335, viz [11], [12].

Displej RG16080B-YHY-X použitý u palubní jednotky používá již zmíněný řadič Sanyo LC7981. Řadič obsahuje vlastní generátor znakové sady o velikosti 192 znaků. Znaková sada začíná prázdným znakem a pokračuje standardně jako ASCII sada znakem ! až po znak s kódem 0xFF. Displej také umožňuje naprogramovat si v textovém režimu vlastní znaky.

### 1.3.2 Displej RG16080B-YHY-X

K zobrazení všech měřených hodnot slouží displej RG16080B-YHY-X od výrobce Raystar, viz [13]. Jedná se o grafický LCD displej s řadičem Sanyo LC7981, který má rozlišení 160x80 bodů. Displej má žlutozelené pozadí, obsahuje LED podsvícení a je typu STN (Super Twisted Nematic), tzn., že kontrastní poměr displeje je 7:1 a pozorovací úhel 60°.

Základní parametry obvodu RG16080B-YHY-X:

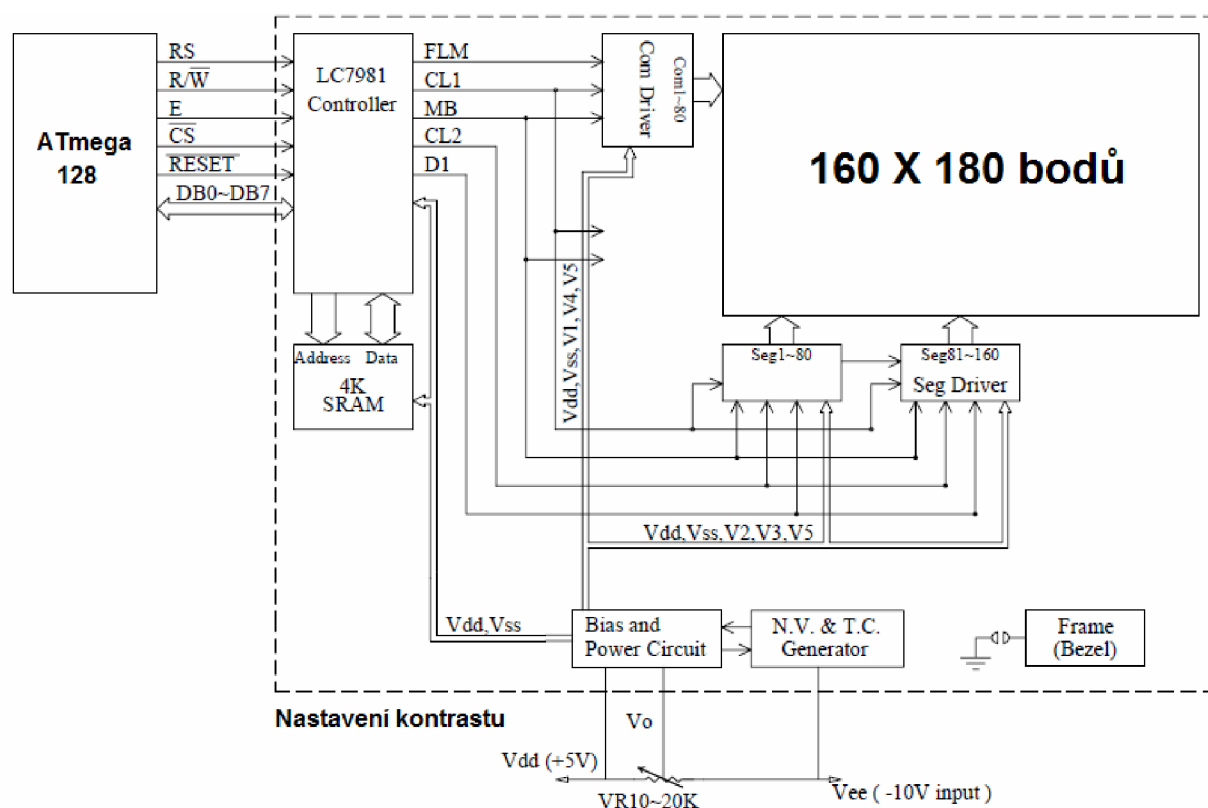
- rozlišení displeje 160x80 bodů (pixelů),
- velikost viditelné oblasti 67,17 mm x 33,57 mm,
- velikost jednoho bodu 0,48 mm x 0,48 mm,
- pracovní teplota od -20 °C do 70 °C,
- napájecí napětí od 4,75 V do 5,25 V.

Displej obsahuje celkem 20 vývodů, význam jednotlivých vývodů udává tabulka č. 2.

**Tab. 2:** Obsazení vývodů displeje RG16080B-YHY-X (převzato z [13]).

Číslo vývodu	Signál	Funkce
1	V <sub>SS</sub>	Zem (0 V)
2	V <sub>DD</sub>	Napájecí napětí (+ 5 V)
3	V <sub>O</sub>	Nastavení kontrastu
4	RS	Výběr instrukcí (L) nebo dat (H)
5	W/R	Volba zápisu (L) nebo čtení dat (H).
6	E	Povolení signálu
7	DB0-DB7	Data vstup / výstup (LSB –MSB)
15	/CS	Chip povolen aktivní při L
16	/RES	Reset řadiče aktivní při L
17	VEE	Nastavení kontrastu (-10 V)
18	NC	-
19	LED + (A)	Anoda podsvěcovací LED
20	LED - (K)	Katoda podsvěcovací LED

Displej obsahuje osm datových vývodů DB0 až DB7 a pět řídicích vývodů RS, W/R, E, /CS a /RES. Připojení displeje k mikrokontroléru ATmega128 zabere téměř dva celé porty mikrokontroléru. Displej je připojen na porty PC a PD mikrokontroléru. K portu PC jsou připojeny datové vývody DB0 až DB7 displeje. Na piny portu PD jsou připojeny řídicí vývody displeje. Blokové schéma displeje je uvedeno na obr. 16.

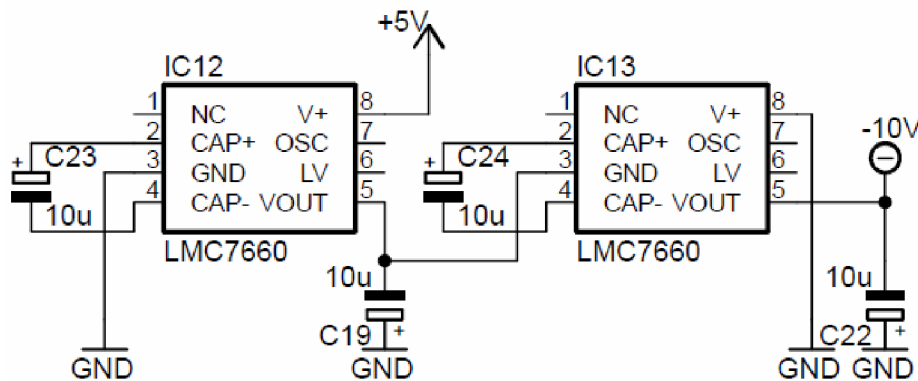


**Obr. 16:** Blokové schéma displeje RG16080B-YHY-X (převzato z [13]).

### 1.3.3 Nastavení kontrastu displeje

Vybraný grafický displej byl nejprve otestován na nepájivém kontaktním poli. Při zapojení displeje podle datasheetu výrobce nastal problém s nastavením kontrastu displeje, viz [13]. V datasheetu je uvedeno, že pro nastavení kontrastu je nutné mezi napájecí napětí 5 V, vývod Vo a zem připojit odporový trimr o velikosti 10 k $\Omega$  až 20 k $\Omega$ . Po konzultaci s technickým oddělením Taiwanského výrobce Raystar bylo zjištěno, že datasheet obsahuje několik chyb. Kontrast displeje nemá být ovládán napětím v rozsahu od 0 V do 5 V, jak bylo uvedeno původně, ale napětím v rozsahu od -10 V do 5 V.

Díky nutnosti použití záporného napětí -10 V muselo být upraveno zapojení displeje. Pro změnu napětí z hodnoty 5 V na -10 V bylo použito kaskádní zapojení dvou spínaných napěťových měničů LMC7660 od výrobce National Semiconductor, viz [14]. Tento měnič slouží k převodu kladného napětí v rozsahu 1,5 V až 10 V na odpovídající hodnoty záporného napětí -1,5 V až -10 V. Pokud bychom použili pouze jeden měnič LMC7660, tak při vstupním napětí 5 V by bylo na jeho výstupu záporné napětí pouze -5 V. Proto je použito kaskádní zapojení dvou měničů, na jehož výstupu je požadované záporné napětí -10 V. Celkové schéma kaskádního zapojení měničů LMC7660 udává obr. 17.



Obr. 17: Kaskádní zapojení dvou spínaných napěťových měničů LMC7660.

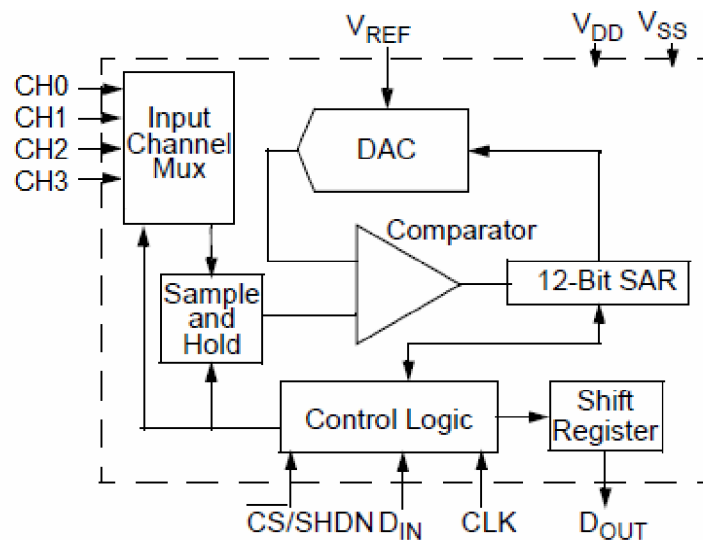
## 1.4 A/D převodník

Pro převod analogových signálů lze použít buď interní A/D převodník mikrokontroléru, nebo samostatný externí A/D převodník. Pro měření palubního napětí a také pro převod analogových signálů z akcelerometru byl zvolen převodník MCP3204 od výrobce Microchip, viz [15]. Jedná se o 4kanálový 12bitový A/D převodník. Výhodou tohoto převodníku je větší počet rozlišitelných úrovní převáděného signálu. U 10bitového převodníku mikrokontroléru je to 1024 úrovní a u 12bitového převodníku 4096 úrovní.

Základní parametry obvodu MCP3204:

- 12bitové rozlišení, integrální nelinearita  $\pm 1$  LSB,
- datová komunikace pomocí rozhraní SPI,
- čtyři analogové vstupní kanály,
- pracovní teplota -40 °C až 85 °C,
- napájecí napětí od 2,7 V do 5,4 V.

A/D převodník komunikuje s mikrokontrolérem pomocí sběrnice SPI, jejíž popis je uveden v kapitole 1.1.1. Pouzdro převodníku obsahuje celkem 14 vývodů a je typu SOIC. Vnitřní blokové schéma zapojení převodníku je uvedeno na obr. 18.



**Obr. 18:** Vnitřní blokové schéma A/D převodníku MCP3204 (převzato z [15]).

Na kanály CH0 až CH2 jsou připojeny analogové výstupy X, Y a Z akcelerometru a na kanál CH3 je připojeno palubní napětí motocyklu. Velikost palubního napětí dosahuje velikosti 12 V, proto musí být na vstup kanálu CH3 připojen dělič napětí. Ten tuto hodnotu sníží pod 3,3 V, což je nastavená velikost referenčního napětí A/D převodníku. Při výpočtu hodnot rezistorů se vychází ze vzorce pro dělič napětí. Aby nedocházelo při kolísání palubního napětí k překročení napětí 3,3 V na výstupu děliče, je uvažovaná hodnota vstupního napětí  $U_1$  dosazená do vzorce 13 V.

Hodnoty rezistorů se vypočítají ze vzorce pro dělič napětí.

$$U_2 = U_1 \cdot \frac{R_2}{R_1 + R_2}, \quad [V; V, \Omega, \Omega, \Omega] \quad (2)$$

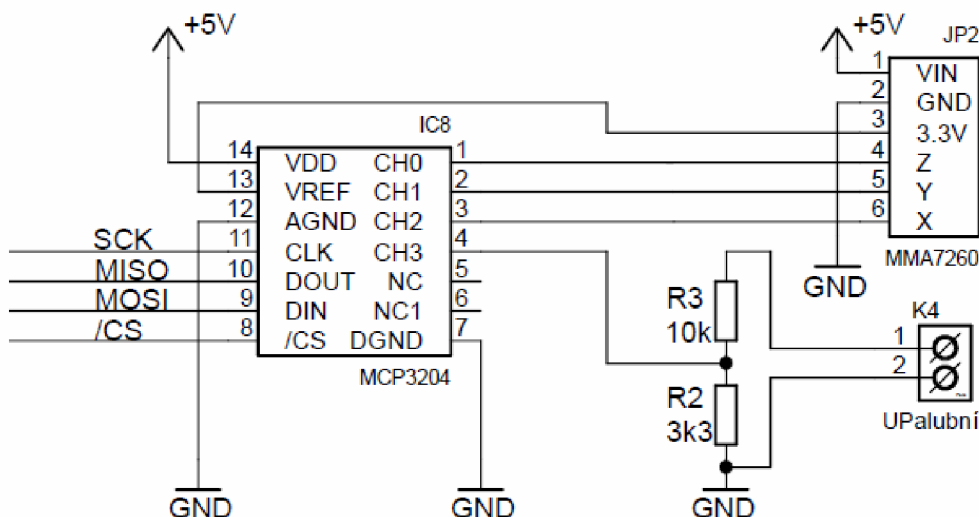
kde  $U_2$  výstupní napětí děliče;  
 $U_1$  vstupní napětí děliče;  
 $R_1$  odpor  $R_1$  děliče;  
 $R_2$  odpor  $R_2$  děliče.

Hodnota rezistoru  $R_2$  byla zvolena 3,3 k $\Omega$ ,

$$\text{z toho plyne } \Rightarrow R_1 = R_2 \cdot \frac{U_1}{U_2} - R_2 = 3300 \cdot \frac{13}{3,3} - 3300 = \underline{\underline{9,7k\Omega}}$$

Pro výslednou hodnotu rezistoru  $R_1$  byla zvolena nejbližší hodnota z řady, tedy 10 k $\Omega$ .

Celkové schéma zapojení A/D převodníku pro převod napětí z akcelerometru a měření napětí baterie je uvedeno na obr. 19.



Obr. 19: Zapojení A/D převodníku na rozhraní SPI.

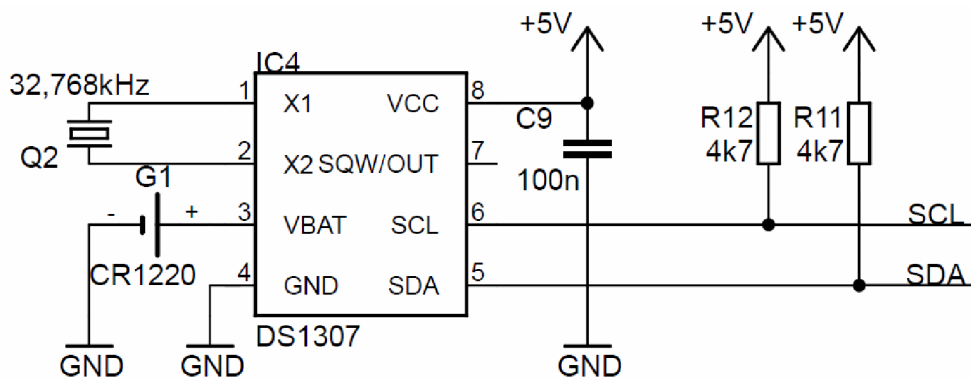
## 1.5 Obvod hodin reálného času

Pro generování aktuálního času, data a roku je nejvýhodnější použít obvod hodin reálného času, neboli obvod RTC (Real Time Clock). Další možností je využití GPS modulu, nebo čítače/časovače mikrokontroléru. U GPS modulu může při jízdě docházet k výpadkům signálu, proto je pro přesné měření času výhodnější použít obvod RTC. Čítač/časovač mikrokontroléru je vhodný spíše pro měření krátkých časových intervalů, což se využívá pro měření rychlosti motocyklu a otáček motoru. Další výhodou obvodu RTC je napájení pomocí záložní baterie, v případě výpadku napájení. Údaje o aktuálním čase a datu se zobrazují na grafickém displeji a také se současně s dalšími naměřenými hodnotami ukládají do paměti EEPROM mikrokontroléru. Jako vhodný obvod reálného času byl zvolen typ DS1307 od výrobce MAXIM-DALLAS, viz [16]. Obvod měří sekundy, minuty, hodiny, dny v týdnu, datum v měsíci, měsíce (včetně měsíců s méně než 31 dny) a roky až do roku 2100 (včetně opravy přestupného roku).

Základní parametry RTC obvodu DS1307:

- datová komunikace pomocí rozhraní TWI ( $I^2C$ ),
- $I^2C$  slave adresa obvodu 0b1101000,
- umožňuje nastavení formátu 12 nebo 24 hodin s AM/PM indikátorem,
- paměť RAM (nonvolativní) s kapacitou 56 Byte pro uložení dat,
- při napájení z baterie v záložním režimu je spotřeba menší než 500 nA
- provozní napětí v rozsahu 4,5V až 5,5 V, při napájení z baterie připojené k vývodu  $V_{BAT}$  2 V až 3,5 V.

Obvod reálného času komunikuje s mikrokontrolérem pomocí sériového rozhraní TWI. Popis sběrnice TWI je uveden v kapitole 1.1.2. Celkové zapojení obvodu na sběrnici uvádí obr. 20.



**Obr. 20:** Zapojení obvodu RTC DS1307 na sériové rozhraní TWI.

Pouzdro obvodu obsahuje celkem osm vývodů. Vývody SCL a SDA se používají ke komunikaci po sériovém rozhraní TWI. Na těchto vývodech jsou dle doporučení výrobce připojeny pull-up rezistory o velikosti 4,7 k $\Omega$ , které zajišťují vysokou úroveň v klidovém stavu. K vývodům X1 a X2 je připojen externí krystal o frekvenci 32,768 kHz. Vývod V<sub>BAT</sub> je záložní vstup podporující standardní 3V Lithiové baterie. Doba zálohování udávaná výrobcem je více než 10 let u baterií s kapacitou 48 mAh a vyšší. Pokud je vývod V<sub>CC</sub> připojen k napájecímu napětí, tak se mohou data číst i zapisovat. V případě, že napájecí napětí klesne pod hodnotu 1,25 x V<sub>BAT</sub>, obvod se přepne do záložního režimu. Dojde-li k poklesu napájecího napětí a přepnutí do záložního režimu, nelze již provádět čtení a zápis, ale funkce měření času se provádí i při nižším napětí.

Inicializace času a data se uskuteční zapsáním do příslušného registru. Obsah registru je uložen v BCD (Binary Coded Decimal) formátu.

## 1.6 GPS modul

Palubní jednotka umožňuje zobrazovat zeměpisnou polohu a nadmořskou výšku a také zaznamenávat projetou trasu. Pro přesné snímání polohy byl vybrán GPS modul FGMMOPA6B od výrobce GlobalTop, viz [17]. Tento modul je osazen chipsetem Mediatek MT3229, viz [18]. Výhodou modulu jsou jeho malé vnější rozměry a také integrovaná anténa.

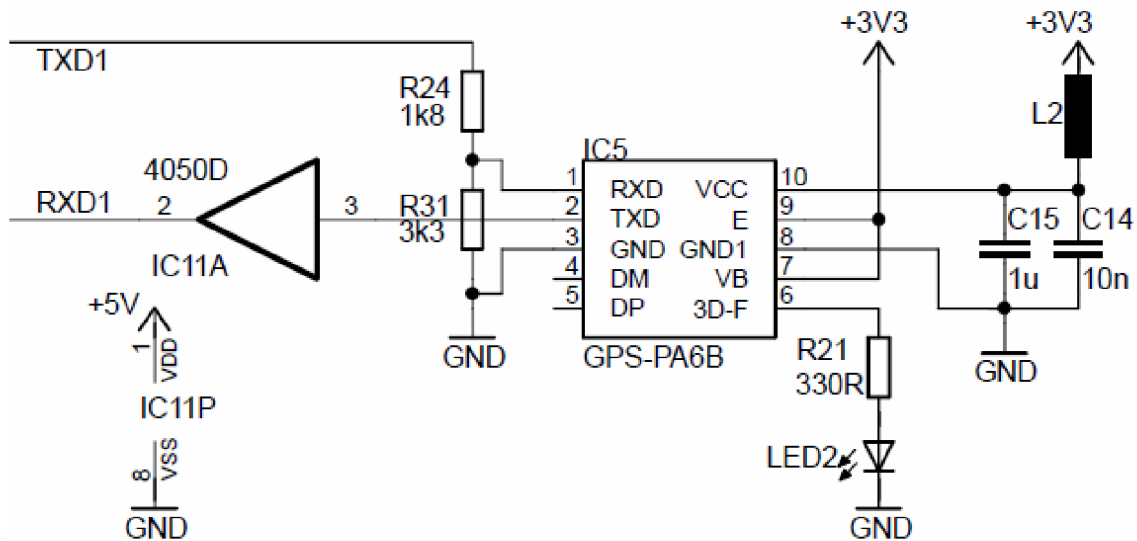
Základní parametry obvodu FGMMOPA6B:

- modul je osazen chipsetem MT3329,
- vysoká citlivost až do -165 dBm,
- nízká spotřeba energie 48 mA při získávání, 37 mA při sledování,
- možnost přímého připojení na USB rozhraní,
- podpora protokolu NMEA 0183 V3.01,
- napájecí napětí od 3,2 V do 5 V.

GPS modul je připojen k mikrokontroléru pomocí rozhraní USART, jehož popis je uveden v kapitole 1.1.3. Na obr. 18. je uvedeno celkové zapojení GPS modulu k mikrokontroléru. Obvod obsahuje celkem 10 vývodů a pro komunikaci slouží vývody TXD a RXD. Na sériovém datovém výstupu TXD jsou vysílána data ve formátu protokolu NMEA, viz [19].

NMEA 0183 Standart Interface definuje požadavky na elektrický signál, protokol pro přenos dat a času a specifikuje konkrétní formát pro sériovou sběrnici s přenosovou rychlostí 4800 baudů. Každá sběrnice může mít pouze jeden vysílač, ale jeden nebo více přijímačů. Tento standart je určen pro podporu sériového přenosu dat z jednoho vysílače k jednomu nebo více přijímačům. Data jsou v ASCII formátu a mohou obsahovat informace jako je poloha, rychlost, výška a další.

Přenosovou rychlost modulu lze nastavit v rozmezí od 4800 baudů do 115200 baudů, standardně je nastaveno 9600 baudů. Formát asynchronního rámce je 8N1, tedy 8 datových bitů, bez parity a jeden stop bit. Přes vývody DP a DM je možné přímé připojení na USB rozhraní.



**Obr. 21:** Zapojení GPS modulu FGMMOPA6B na rozhraní USART.

Požadované napájecí napětí GPS modulu je 3,3 V, proto bylo nutné použít stabilizátor napětí ke snížení hodnoty napětí z 5 V na 3,3 V, popis stabilizátoru LT1121 CZ-3,3 je uveden v kapitole 1.9.2. Pro správnou funkci GPS modulu je důležité dodržet podmínku, aby zvlnění napájecího napětí bylo menší než  $50 \text{ mV}_{pp}$ . Proto je dle doporučení výrobce na napájecí vývod Vcc připojena feritová perlička a dvojice kondenzátorů o kapacitách  $1\mu\text{F}$  a  $10 \text{ nF}$ . Feritová perlička slouží pro potlačení elektromagnetického vyzařování, neboli zkráceně EMI. V zapojení je použit typ BLM31AJ601SN1L od výrobce Murata, viz [20].

Kromě napájecího napětí bylo potřeba upravit úroveň na datových vodičích rozhraní UART. Protože logická úroveň H na vývodu TXD se dle datasheetu pohybuje v rozmezí 2,1 V až 2,8 V a u mikrokontroléru je doporučena minimální hodnota 3 V, byla tato úroveň zvýšena pomocí budiče HEF4050B na velikost 5 V, viz [21]. Jedná se o neinvertující budič, který již od hodnoty vstupního napětí 2,3 V poskytuje na svém výstupu napětí 5 V. Tím je zabezpečen bezchybný přenos dat mezi GPS modulem a mikrokontrolérem.

Naopak pro snížení úrovně napětí je na vývodu RXD zapojen dělič napětí, který snižuje úroveň napětí z 5 V na 3,3 V, při výpočtu se vychází ze vzorce č. 2, viz kapitola 1.4.

Hodnota rezistoru  $R_2$  byla zvolena  $3,3 \text{ k}\Omega$ ,

$$\text{z toho plyne } \Rightarrow R_1 = R_2 \cdot \frac{U_1}{U_2} - R_2 = 3300 \cdot \frac{5}{3,3} - 3300 = \underline{\underline{1,7 \text{ k}\Omega}}$$

Pro výslednou hodnotu rezistoru  $R_1$  byla zvolena nejbližší hodnota z řady, tedy  $1,8 \text{ k}\Omega$ .

Výrobce udává tři typické hodnoty inicializace modulu, horký start 1s, teplý start 34 s a studený start 35 s. Záznam trasy z GPS modulu se ukládá na paměťovou kartu SD. Záznam je uložen ve formátu NMEA. Tento formát je možné po přenesení dat do počítače zpracovat pomocí různých programů, např. Google Earth. A zjistit tak například trasu na mapě a její výškový profil.

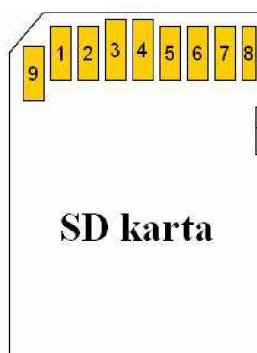
## 1.7 SD karta

Pro ukládání naměřených hodnot musela být vybrána paměť s dostatečnou kapacitou. Zvolena byla paměťová karta SD (Secure Digital) s kapacitou 2 GB od výrobce Kingston, viz [22]. Na kartu se ukládají jednotlivé NMEA věty z GPS modulu. Ostatní data jako jsou denní ujetá vzdálenost, čas jízdy, teplota okolí, napětí baterie, celková ujetá vzdálenost atd. se ukládají do paměti EEPROM mikrokontroléru.

Základní vlastnosti použité paměťové karty SD:

- kapacita paměti 2 GB,
- systém souborů FAT16,
- datová komunikace pomocí rozhraní SPI,
- pracovní teplota od  $-25\text{ }^{\circ}\text{C}$  do  $85\text{ }^{\circ}\text{C}$ ,
- rozměry pouzdra 24 mm x 32 mm x 2,1 mm,
- napájecí napětí od 2,7 V do 3,6 V.

Paměťová karta komunikuje s mikrokontrolérem prostřednictvím sériového rozhraní SPI, jehož funkce je popsána v kapitole 1.1.1. Mikrokontrolér je nastaven v režimu master a paměť v režimu slave. Pouzdro paměťové karty obsahuje celkem 9 vývodů, z nichž tři vývody jsou určeny pro napájení. Vzhled pouzdra s rozmístěním jednotlivých pinů udává obr. 22.



**Obr. 22:** Rozmístění vývodů paměťové karty SD (převzato z [22]).

Pro komunikaci paměťové karty prostřednictvím rozhraní SPI slouží čtyři vývody CMD/DI, DAT0/DO, DAT3/CS a SCK. Popis jednotlivých vývodů lze nalézt v tabulce č. 3.



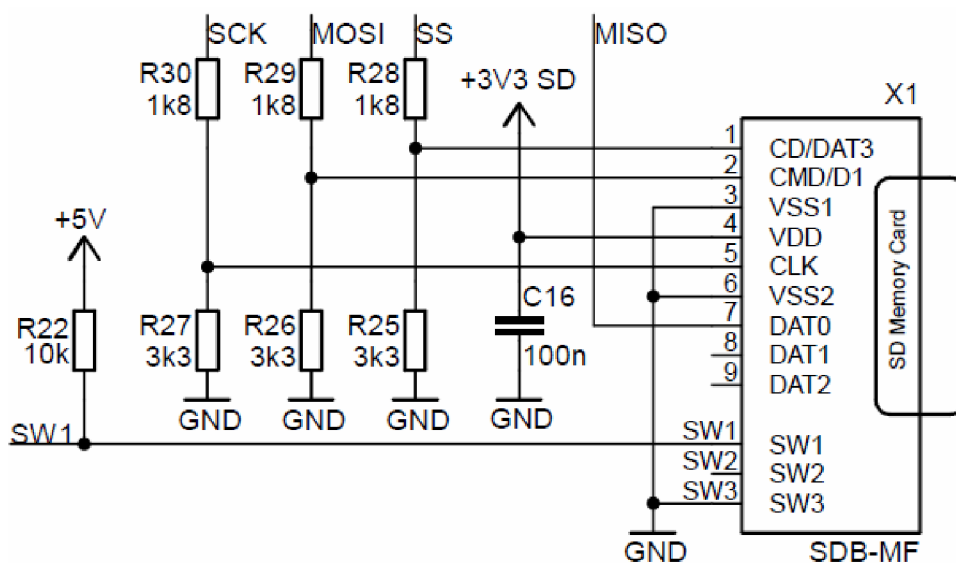
**Tab. 3:** Obsazení vývodů paměťové karty SD (převzato z [22]).

Číslo vývodu	Název	Funkce (SPI)
1	DAT3/CS	Chip select / Slave select
2	CMD/DI	MOSI Master Out Slave In
3	VSS1	Zem (0 V)
4	VDD	Napájecí napětí (3,3 V)
5	CLK	Clock
6	VSS2	Zem (0 V)
7	DAT0/DO	MISO Master In Slave Out
8	DAT1/IRQ	Nepoužívá se nebo IRQ
9	DAT2/NC	Nepoužívá se

Napájecí napětí SD karty je požadováno v rozsahu 2,7 V až 3,6 V, proto bylo nutné použít stabilizátor napětí. K tomuto účelu slouží stabilizátor LT1121 CN8-3,3, který upravuje napájecí napětí 5 V na napětí 3,3 V. Popis stabilizátoru je uveden v kapitole 1.9.2. Výhodou tohoto stabilizátoru je možnost vypnutí jeho výstupu. K tomu slouží pin SHDN stabilizátoru, který je aktivní v logické nule. Tento pin je připojen na pin PA7 mikrokontroléru. Pokud tedy dojde k situaci, že SD karta přestane reagovat, můžeme pomocí mikrokontroléru provést reset SD karty tím, že ji odpojíme od napájení.

Kromě napájecího napětí SD karty bylo třeba také upravit úroveň napětí na datových vodičích SPI rozhraní. Proto jsou na vývodech SCK, MOSI a SS zapojeny děliče napětí, které snižují napětí na hodnotu 3,3 V. Vzor výpočtu odporů děliče napětí je uveden v kapitole 1.6. Hodnota napětí 3,3 V postačuje pro přímou komunikaci s mikrokontrolérem, protože minimální hodnota úrovně H u mikrokontroléru je 3 V.

Pro snadnější detekci SD karty je použit konektor na SD karty typ SDB-MF od výrobce Mutlicomp, viz [23]. Hlavní výhodou tohoto konektoru je možnost detekce vložení karty a také detekce ochrany proti přepsání karty. V zapojení je využita kontrola vložení karty. Po zasunutí karty do konektoru dojde k propojení vývodů SW1 a SW3, což je indikováno na pinu PF2 mikrokontroléru. Celkové schéma zapojení SD karty k mikrokontroléru udává obr. 23.



**Obr. 23:** Zapojení SD karty na sériový kanál SPI.

## 1.8 Převodník USB na UART

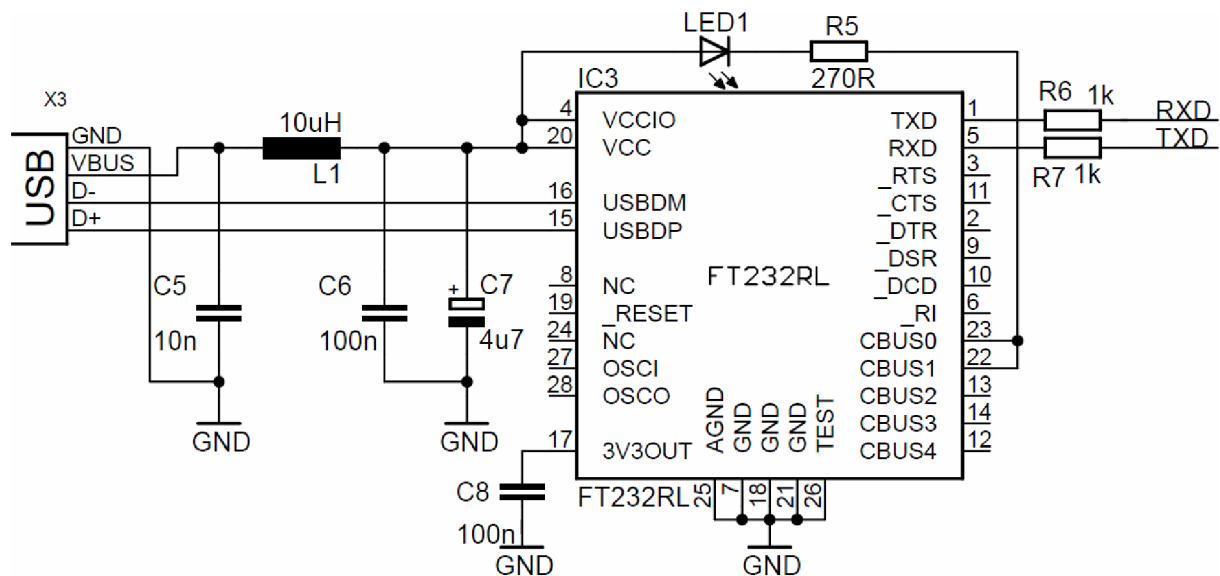
Pro převod naměřených hodnot z multifunkční jednotky do počítače bylo vybráno připojení pomocí sběrnice USB. Ke komunikaci s počítačem slouží jednočipový převodník USB na UART typ FT232RL od výrobce FTDI, viz [24]. Výhodou převodníku je spojení možností sběrnice USB spolu s jednoduchým připojením vnějších zařízení, protože komunikace sériovým asynchronním kanálem je snazší než po sběrnici USB, viz [25].

Po připojení multifunkční jednotky do PC a nainstalování volně dostupného ovladače VCP (Virtual COM Port) dojde k vytvoření virtuálního sériového portu. Tento port je poté možné využívat jako standardní sériový port COM. Mimo ovladače VCP je možné ještě použít druhou variantu, a to přímé ovladače D2XX. Tyto ovladače umožňují komunikaci aplikačního programu s převodníkem pomocí kódu, který je uložen v dynamické knihovně DLL.

Základní parametry převodníku FT232RL:

- celý USB protokol přímo na čipu,
- kompatibilita se standardy USB 1.1 a USB 2.0,
- unikátní identifikační číslo, označované jako FTDIChip-ID,
- podpora 7 nebo 8bitového přenosu, 1 nebo 2 stop bity, parita (lichá, sudá, značená, mezerová, žádná),
- přenosové rychlosti lze nastavit v širokých mezích od 300 Bd až do 3 MBd,
- VCP ovladač podporován operačními systémy Windows, Linux a MAC,
- napájení obvodu je v rozsahu od 3,3 V do 5,25 V, lze napájet přímo z USB.

Převodník FT232RL komunikuje s mikrokontrolérem prostřednictvím rozhraní USART, které využívá dvojici vodičů TXD (výstup vysílaných dat) a RXD (vstup přijímaných dat). Princip komunikace je popsán v kapitole 1.1.3. Celkové schéma zapojení převodníku FT232RL na sběrnici USB udává obr. 24.



Obr. 24: Zapojení převodníku FT232RL při napájení ze sběrnice USB.

Kvůli úspoře energie je napájení obvodu FT232RL realizováno přímo z USB konektoru. Převodník je připojen na sběrnici USB pomocí mini USB konektoru. Pro indikaci přenosu dat je na vývodech CBUS0 a CBUS1 připojena LED dioda. Vývody RXD (pin PE0) a TXD (pin PE1) se u mikrokontroléru ATmega128 používají také pro ISP programování, proto jsou na těchto vývodech připojeny rezistory 1 k $\Omega$ .

Po připojení multifunkční jednotky k počítači, je možné přenést veškeré naměřené údaje z paměti EEPROM mikrokontroléru a z SD karty. Výpis přijatých dat se provede v programu Terminal 1.9b.

## 1.9 Napájení

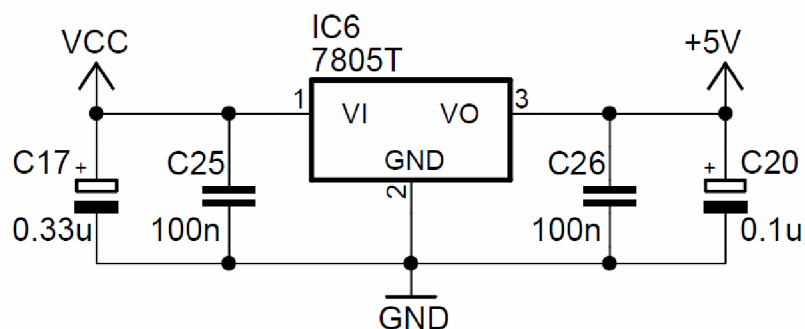
### 1.9.1 Stabilizátor napětí 7805 STM

Palubní jednotka je napájena pomocí 12 V baterie, která je umístěna na motocyklu. Mikrokontrolér, displej, senzory teploty, jazýčkový snímač, otáčkoměr, akcelerometr, A/D převodník a obvod hodin reálného času vyžadují napájecí napětí o velikosti 5 V. Převodník USB na UART není napájen z baterie, ale přímo z konektoru USB po připojení do počítače. Pouze SD karta a GPS modul musí mít napájecí napětí nižší a to v rozsahu 2,7 V až 3,6 V. Změna napájecího napětí z 12 V na 5 V je realizována stabilizátorem napětí 7805 STM od výrobce STMicroelectronics, viz [27].

Základní vlastnosti stabilizátoru napětí 7805 STM:

- stabilizátor pracuje od vstupního napětí od 7 V do 35 V,
- pevné výstupní napětí 5 V,
- výstupní proud až 1 A,
- pracovní teplota -55 °C až 150 °C.

Pouzdro stabilizátoru typu TO-220 obsahuje celkem tři vývody. Celkové schéma zapojení stabilizátoru napětí 7805 je uvedeno na obr. 26. Kromě doporučených elektrolytických kondenzátorů o kapacitách 0,33  $\mu$ F a 0,1  $\mu$ F jsou na vstupu a výstupu stabilizátoru zapojeny blokovací keramické kondenzátory o kapacitě 100 nF.



**Obr. 25:** Celkové schéma zapojení stabilizátoru napětí 7805.

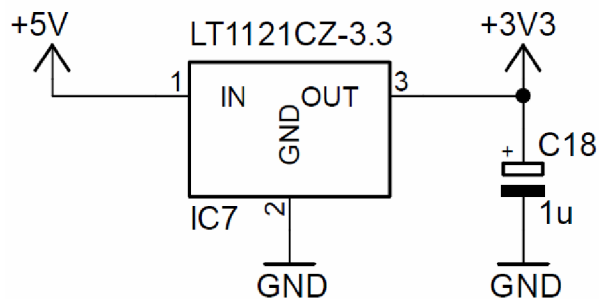
### 1.9.2 Stabilizátor napětí LT1121-3,3

SD karta a GPS modul vyžadují nižší napájecí napětí, proto bylo nutné použít další stabilizátor napětí, který upraví výstupní napětí stabilizátoru 7805 o velikosti 5 V na velikost napájecího napětí 3,3 V. Jako vhodný typ byl zvolen stabilizátor LT1121-3,3V od výrobce Linear Technology, viz [28].

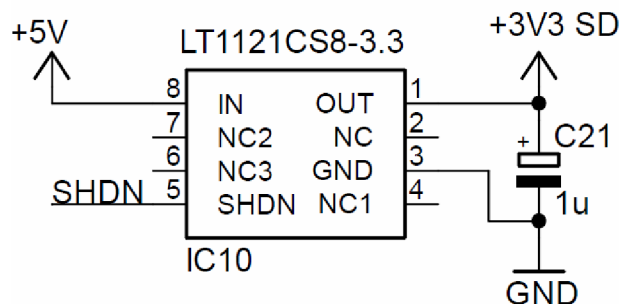
Základní vlastnosti stabilizátoru LT1121-3,3 :

- rozsah vstupního napětí je 4,3 V až 20 V,
- pevné výstupní napětí 3,3 V,
- výstupní proud 150 mA,
- nízký klidový proud 30  $\mu$ A.

Jeden stabilizátor je použit zvlášť pro SD kartu a druhý pro GPS modul. Důvodem použití dvou stabilizátorů je zejména vyšší proudový odběr obou zařízení. SD karta může mít při startu a inicializaci špičkový odběr až 150 mA, což je hodnota blízká se maximálnímu zatížení stabilizátoru. Běžná spotřeba GPS modulu je okolo 50 mA. Dalším důvodem je možnost odpojení SD karty od napájecího napětí a provedení jejího resetu, jak je popsáno v kapitole 1.7. Pro GPS modul je využit jednodušší typ stabilizátoru LT1121 CZ-3,3 v pouzdře TO-92. SD karta využívá složitější typ LT1121 CN8-3,3 v pouzdře SO-8, který má navíc řídicí vývod pro vypnutí výstupu stabilizátoru. Schéma zapojení obou stabilizátorů napětí je uvedeno na obr. 26 a 27.



**Obr. 26:** Celkové schéma zapojení stabilizátorů napětí pro GPS modul.



**Obr. 27:** Celkové schéma zapojení stabilizátorů napětí pro SD kartu.

## 1.10 Návrh obvodového zapojení a DPS

### 1.10.1 Odolnost vůči vysokofrekvenčnímu rušení

Vysokofrekvenční rušení úzce souvisí s provozem elektronických obvodů. VF rušení se projevuje jak na napájecí sběrnici, tak také přeslechy mezi vodiči. V souvislosti s VF rušením je důležitá spolupráce číslicových a analogových obvodů, viz [29].

Aby byla navrhovaná multifunkční jednotka dostatečně odolná vůči VF rušení, je nutné dodržet při návrhu desky plošného spoje určitá kritéria, viz [30]. Odolnost proti VF rušení lze zvýšit výběrem vhodných součástek. Důležité je maximální využití SMD součástek, které jsou odolnější vůči rušení oproti klasickým součástkám. Použité rezistory a keramické kondenzátory byly vybrány v SMD provedení v pouzdru 1206, pouze elektrolytické kondenzátory jsou v klasických pouzdrech. Zvolené velikosti součástek lze bez větších potíží pájet ručně, což u menších pouzder může být problematické. Dalším krokem při návrhu DPS je vhodné rozmístění součástek. Důležité je dodržet minimální vzdálenosti mezi součástkami za účelem minimalizace proudových smyček. Na minimalizaci proudových smyček má také pozitivní vliv již zmíněné použití SMD součástek.

Při návrhu napájecího obvodu je nutné dodržet dostatečné průřezy napájecích vodičů. Prudké ohyby vodičů jsou nevhodné, protože představují místo odrazu vysokofrekvenčního signálu. Vhodné je provést ohyb pod úhlem  $45^\circ$ . Nejméně vhodný ohyb je pod úhlem  $90^\circ$ , zatímco nejvhodnější je zakulacený ohyb, viz obr. č. 28.

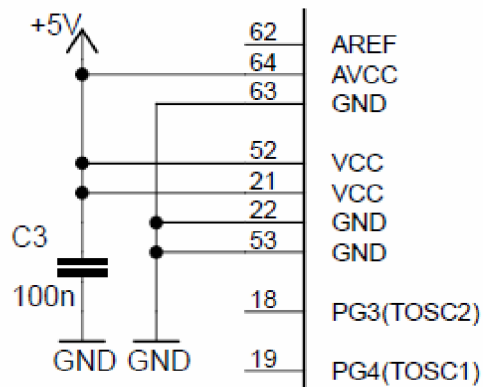


**Obr. 28:** Možnosti provedení ohybu napájecího rozvodu, ohyb  $90^\circ$ , ohyb  $45^\circ$  a zakulacený ohyb.

Při návrhu oboustranného plošného spoje je dobré se vyvarovat se přechodu napájecího rozvodu do druhé strany desky plošného spoje, protože odpor prokůvů je poměrně vysoký. Velký vliv na kvalitu napájecího rozvodu má také použitý napájecí zdroj, v našem případě stabilizátor napětí.

Dalším důležitým krokem je blokování napájení pomocí blokovacích kondenzátorů u každého integrovaného obvodu. Kondenzátory musí mít malou sériovou indukčnost a sériový odpor na vysokých kmitočtech. Vhodné jsou kvalitní keramické kondenzátory s kapacitou 100 nF. Blokovací kondenzátory se umísťují co nejbližše pouzdru integrovaného obvodu. Zapojení blokovacích kondenzátorů k mikrokontroléru udává obr. č. 29.

Odolnost vůči vysokofrekvenčnímu rušení lze zvýšit nejen vhodným výběrem součástek a správným návrhem desky plošného spoje, ale i minimalizací rychlosti přenosu. Tedy tam, kde to není nezbytně nutné, nepoužívat zbytečně rychlou datovou komunikaci.



**Obr. 29:** Zapojení blokovacího kondenzátoru k mikrokontroléru.

### 1.10.2 Návrh obvodového zapojení a DPS

Návrh obvodového zapojení celého zařízení byl vytvořen v editoru plošných spojů Eagle verze 5.6.0 od firmy Cadsoft, viz [31]. Obvodové zapojení palubní jednotky lze nalézt v příloze A.1. Zapojení panelu s tlačítky je uvedeno v příloze A.5. Pasivní součástky i integrované obvody byly z velké části použity v SMD provedení. Knihovny programu Eagle obsahovaly většinu pouzder použitých součástek, pouze pro GPS modul a LCD displej musely být knihovny modifikovány.

Návrh plošného spoje byl vytvořen také v programu Eagle. Deska plošného spoje palubní jednotky je v oboustranném provedení. Návrh desky z pohledu ze strany spojů (Bottom) i ze strany součástek (Top) lze nalézt v příloze A.2 a A.3. Na straně spojů je použita rozlévaná měď (polygon), která je připojena k nulovému potenciálu, čímž je zajištěn komplexní rozvod zemnění. Deska panelu s tlačítky je v jednostranném provedení. Pohled ze strany spojů i ze strany součástek je uveden v příloze A.6 a A.7. Rozměry použité přístrojové krabičky jsou uvedeny v příloze A.8 a A.9.

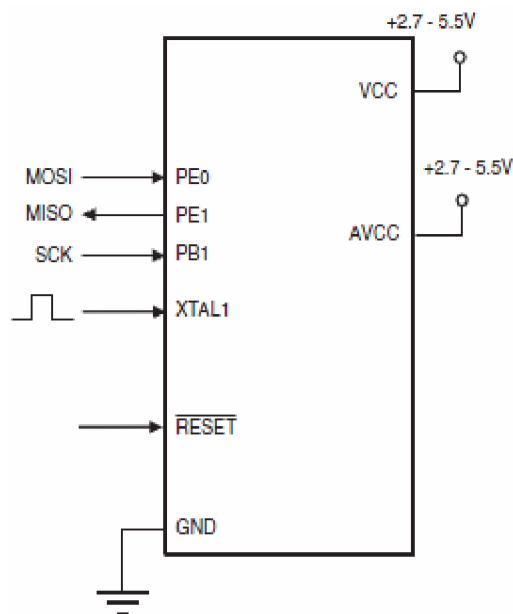
## 2 Návrh softwarové části

### 2.1 Vývojové prostředky

#### 2.1.1 Programování mikrokontroléru

Vybraný mikrokontrolér ATmega128 lze programovat několika způsoby. A to paralelně, sériově pomocí SPI rozhraní, nebo pomocí JTAG rozhraní. Pro vývoj multifunkční jednotky bylo vybráno sériové programování. Tento způsob programování bývá také označován zkratkou ISP (In System Programming). Sériové programování sice nenabízí možnost ladění programu jako při použití JTAG rozhraní, ani nedosahuje rychlostí jako paralelní programování, ale jeho hlavní výhodou je možnost programování mikrokontroléru přímo ve vyvíjené aplikaci, což velmi usnadňuje a urychluje vývoj programu, viz [2].

Při ISP programování se používá šestice vodičů. Jedná se o vodiče označené MOSI, MISO, SCK, RESET, dále napájení a zem. U mikrokontroléru ATmega128 je však oproti většině mikrokontrolérů jedna výjimka. Vývody MISO a MOSI, které jsou vyvedeny na pinech PB2 a PB3, se k programování nepoužívají. Místo nich jsou použity piny PE0 a PE1. Při programování je dále nutné zapojit krystal mezi vývody označené XTAL1 a XTAL2. Schéma zapojení ISP programování mikrokontroléru ATmega128 je uvedeno na obr. 30.



**Obr. 30:** Zapojení ISP programování mikrokontroléru ATmega128 (převzato z [1]).

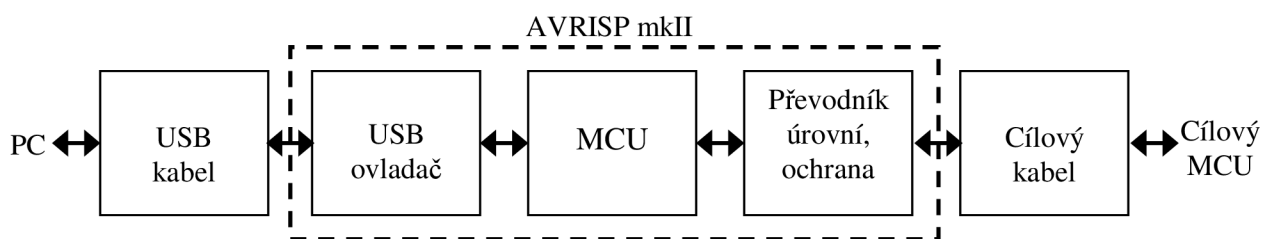
#### 2.1.2 ISP programátor

Důležitým článkem mezi vývojovým prostředím a mikrokontrolérem je programátor. Ten slouží pro převod obslužného programu do cílového mikrokontroléru. Pro tento účel byl vybrán programátor od firmy Atmel, typ AVRISP mkII, viz [32]. Jedná se o ISP programátor, který v kombinaci s vývojovým prostředím AVR Studio podporuje programování veškerých 8bitových mikrokontrolérů AVR s ISP rozhraním.

Základní parametry programátoru AVRISP mkII:

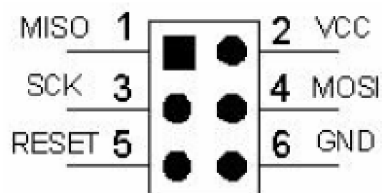
- kompatibilita s vývojovým prostředím AVR Studio verze 4.12 a novější,
- možnost programování paměti FLASH i EEPROM,
- nastavitelná rychlost programování (50 Hz až 8 MHz),
- možnost programování propojek (fuses) a zámkových bitů (lock bit),
- podpora zařízení s napájecím napětím od 1,8 V do 5,5 V.

Programátor se připojuje k počítači pomocí USB rozhraní, podporovány jsou verze USB 1.1 a 2.0. Napájení je realizováno přímo z USB konektoru. Hlavní částí programátoru je mikrokontrolér (MCU), který řídí komunikaci mezi cílovým mikrokontrolérem a vývojovým prostředím. Blokové schéma programátoru AVRISP mkII je uvedeno na obr. 31.



**Obr. 31:** Blokové schéma programátoru AVRISP mkII (převzato z [32]).

Pro indikaci aktuálního stavu je na programátoru umístěna tříbarevná LED dioda. Která informuje uživatele o tom, zda je cílový mikrokontrolér připojen na napájecí napětí a také o průběhu programování. Aby byla zachována vzájemná kompatibilita mezi odlišnými typy ISP programátorů, používá programátor standardní rozmístění vývodů ISP konektoru. Doporučené zapojení ISP konektoru uvádí obr. 32.



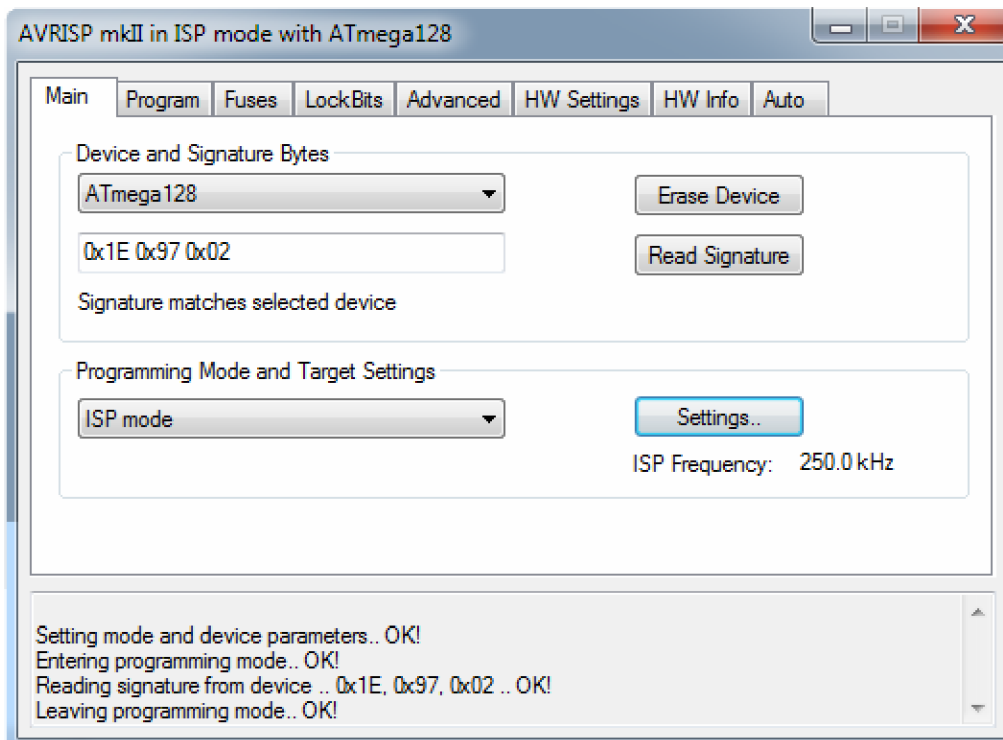
**Obr. 32:** Rozmístění vývodů ISP konektoru programátoru AVRISP mkII (převzato z [32]).

### 2.1.3 Vývojové prostředí

Pro vývoj obslužného programu palubní jednotky byl vybrán programovací jazyk C. Jeho výhodou oproti jazyku symbolických adres JSA (assembler) je lepší přehlednost kódu a také rychlejší vývoj aplikace. Určitou nevýhodou oproti assembleru může být větší velikost výsledného kódu. Při tvorbě programu v jazyce C je potřeba použít překladač (kompilátor), který zdrojový kód přeloží do assembleru a následně do strojového kódu srozumitelného pro mikrokontrolér.



Jako vhodné vývojové prostředí bylo vybráno volně dostupné AVR Studio ve verzi 4.19 od firmy Atmel, viz [33]. Výhodou tohoto prostředí je zejména neomezená délka kódu a možnost krokování aplikace. AVR Studio neobsahuje přímo kompilátor jazyka C, proto je potřeba doinstalovat překladač WinAVR. Po nainstalování vývojového prostředí a kompilátoru byl otestován mikrokontrolér ATmega128 na desce palubní jednotky tak, že byla vyčtena jeho signatura. Proces úspěšného testování mikrokontroléru je uveden na obr. 33.



**Obr. 33:** Proces úspěšného ověření signatury mikrokontroléru ATmega128.

## 2.2 Měření rychlosti motocyklu

### 2.2.1 Aktuální rychlost

Rychlost motocyklu je měřena pomocí jazýčkového snímače, jehož popis je uveden v kapitole 1.2.1. Pro měření rychlosti bylo potřeba najít způsob měření periody jednoho otočení kola motocyklu.

První možností měření periody je využití jednotky Input Capture čítače/časovače mikrokontroléru. Tato jednotka slouží k zachycení vnější události (signálu) převedené na vývod ICP mikrokontroléru. Po zachycení události je jí přiřazena časová známka, ve které událost nastala. Časovou známku poté můžeme použít pro výpočet frekvence nebo periody. Druhou alternativou je využití jednotky Output Compare čítače/časovače a externího přerušení mikrokontroléru. Celé měření se poté spouští pomocí náběžné, nebo sestupné hrany signálu převedeného na vývod externího přerušení. Následně je čítačem/časovačem měřena perioda signálu.

Jako vhodný způsob měření periody byl vybrán postup využívající jednotku Output Compare a externí přerušení. K měření byl využit 16bitový čítač/časovač 1 v módu CTC (Clear Time on Compare Match). Tento mód umožňuje nastavení rozlišení čítače, tedy hodnoty přetečení. Doba přetečení byla nastavena na 5 ms. Nastavení se provádí pomocí následujícího vztahu, viz [1].

$$f_{OCO} = \frac{f_{clk\_IO}}{2 \cdot N \cdot (1 + OCR1A)}, \quad [\text{Hz}; \text{Hz}, -, -,] \quad (3)$$

kde  $f_{clk\_IO}$  pracovní kmitočet mikrokontroléru;  
 $N$  dělicí poměr;  
 $OCR1A$  registr pro manipulaci s rozlišením čítače.

Nastavený pracovní kmitočet mikrokontroléru je 8 MHz, předdělička 8. Kmitočet generovaného signálu 200 Hz je dán dobou přetečení 5 ms.

$$\text{z toho plyne } \Rightarrow OCR1A = \frac{f_{clk\_IO}}{2 \cdot N \cdot f_{OCO}} - 1 = \frac{8000000}{2 \cdot 8 \cdot 200} - 1 \cong 2499$$

Vypočtená hodnota byla uložena v hexadecimální soustavě do registru  $OCR1A = 0x9C3$ . Nastavení čítače/časovače 1 je provedeno pomocí funkce `init_timer1()`.

```
void init_timer1() // inicializace čítače/časovače 1
{
    OCR1A = 0x9C3; // doba přetečení 5 ms
    TCCR1B = (1<<WGM12) | (1<<CS11); // režim CTC, dělička 8
    TCNT1H = 0; // obsah č/č 1
    TCNT1L = 1; // obsah č/č 1
    TIMSK = (1<<OCIE1A); // přerušení jednou za 5 ms
}
```

Měření rychlosti se spouští přiblížením magnetu k jazýčkovému snímači. Sepnutí snímače vytváří sestupnou hranu, na kterou reaguje externí přerušení INT6. Externí přerušení poté aktivuje inkrementaci počtu přetečení v obsluze čítače/časovače `Timer1_COMPA_vect`.

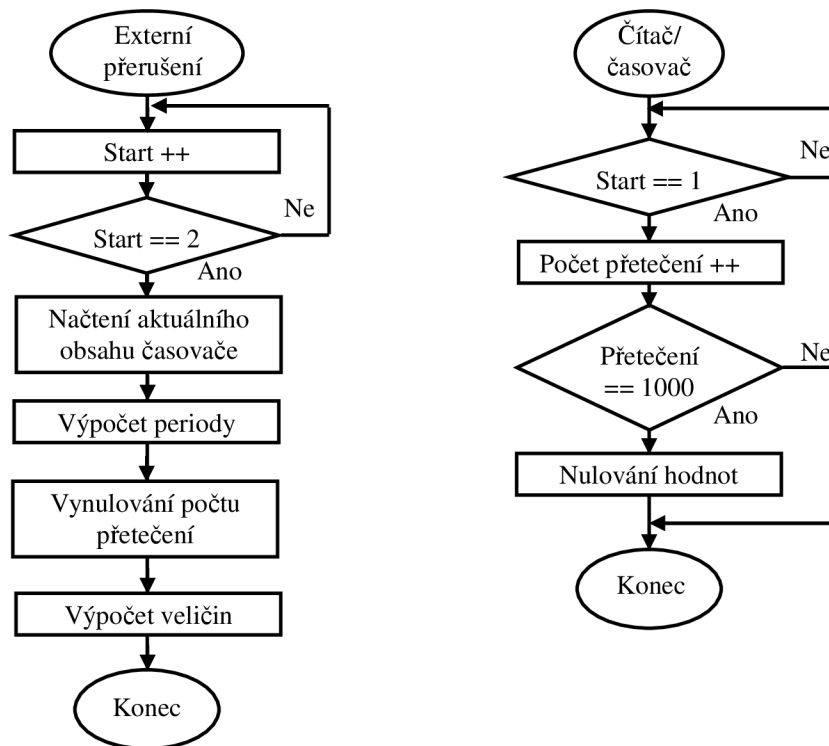
Při druhém sepnutí snímače se spustí výpočet periody v obsluze externího přerušení INT6. Velikost periody je dána součinem počtu přetečení a doby přetečení. Pro zvýšení přesnosti je k hodnotě periody přičtena aktuální hodnota čítače/časovače, která je uložena v registru `TCNT1`. Po výpočtu periody se vynuluje počet přetečení čítače/časovače a provede se výpočet aktuální rychlosti motocyklu pomocí následujícího vztahu.

$$v = \frac{o}{T} \cdot 3,6, \quad [\text{km/h}; \text{m}, \text{s}] \quad (4)$$

kde  $o$  nastavený obvod kola motocyklu;  
 $T$  změřená perioda jednoho otočení kola.

Výpočet aktuální rychlosti se provede vydělením obvodu kola motocyklu periodou jednoho otočení kola a následným vynásobením číslem 3,6 pro převod z metrů za sekundu na kilometry za hodinu. Pokud nedojde k opětovnému sepnutí jazýčkového snímače do čtyř sekund, celé měření končí.

Celý postup měření periody a následného výpočtu rychlosti je nejlépe patrný z vývojového diagramu uvedeného na obr. 34.



**Obr. 34:** Vývojový diagram měření periody a následného výpočtu rychlosti.

## 2.2.2 Maximální rychlost

Palubní jednotka umožňuje také měření maximální rychlosti motocyklu. Výpočet rychlosti se provádí v obsluze externího přerušení INT6 pomocí podmínky, která porovnává aktuální a maximální rychlost.

```

if (Speed > Speed_Max) // porovnání aktuální a maximální rychlosti
    Speed_Max = Speed; // uložení maximální rychlosti

```

## 2.2.3 Průměrná rychlost

Při výpočtu průměrné rychlosti se postupně přičítá ujetá vzdálenost daná obvodem kola motocyklu a také jednotlivé periody otočení kola. Vztah pro výpočet průměrné rychlosti je totožný jako u aktuální rychlosti, viz vztah č. 4.

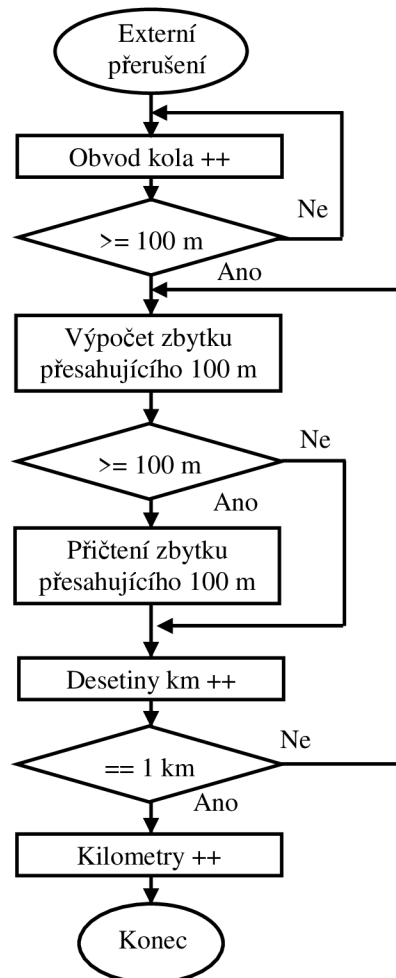
```

Track_Avg++; // inkrementace násobků obvodu kola
Time_Avg += Period_Speed; // sčítání period
Speed_Avg = (Track_Avg * Circuit_m / Time_Avg) * 3.6; // průměrná rychlost

```

## 2.3 Měření denní a celkové vzdálenosti

Další funkcí palubní jednotky je měření denní a celkové ujeté vzdálenosti. Princip měření spočívá v postupném přičítání obvodu kola motocyklu. Přičítání probíhá, dokud není splněna podmínka, že součet hodnot je roven nebo větší než 100. Poté dochází k přičtení vzdálenosti 0,1 km, což je nejmenší měřená vzdálenost. Pro dosažení větší přesnosti měření je do výpočtu zahrnuta další podmínka, která přičítá jednotlivé zbytky vzdáleností přesahující hodnotu 100 m. Stejným principem postupného přičítání se dále počítají denní a celkové ujeté kilometry. Postup měření vzdálenosti je znázorněn na vývojovém diagramu na obr. 35.

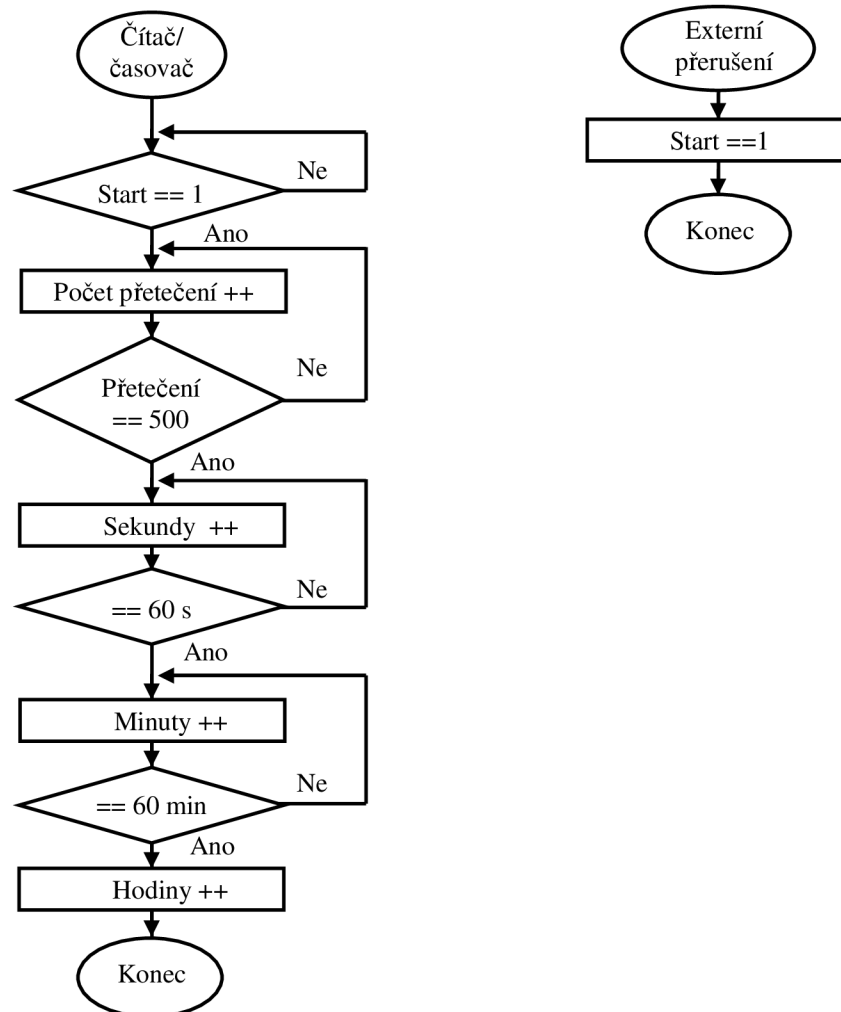


Obr. 35: Vývojový diagram měření ujeté vzdálenosti.

## 2.4 Měření času jízdy

Princip měření času jízdy je podobný jako u vzdálenosti. Pouze místo obvodu kola se přičítají počty přetečení čítače/časovače 1. Nastavená doba přetečení je 5 ms, viz vztah č. 3. Měření času se spouští při první sepnutí jazýčkové snímače. Pokud se magnet neobjeví do čtyř sekund, měření času se zastaví. Inkrementace přetečení probíhá, dokud není splněna podmínka, že počet přetečení je roven hodnotě 500. Poté dochází k přičtení jedné sekundy.

Tímto principem se dále přičítají minuty a hodiny. Postup měření času jízdy je znázorněn na vývojovém diagramu na obr. 36.



Obr. 36: Vývojový diagram měření času jízdy.

## 2.5 Měření otáček motoru

### 2.5.1 Aktuální otáčky

Měření otáček motoru se provádí pomocí měření periody mezi dvěma výboji na zapalovací cívce. Popis otáčkoměru je uveden v kapitole 1.2.2. Otáčkoměr je připojen k externímu přerušení INT7, které reaguje na sestupnou hranu. Pro měření periody je opět použit čítač/časovač 1 s dobou přerušení 5 ms v módu CTC. Celé měření je spouštěno sestupnou hranou signálu. Vývojový diagram pro výpočet rychlosti motocyklu odpovídá i výpočtu aktuálních otáček motoru, viz obr. 34.

Při druhé sestupné hraně je změřena perioda a vypočten počet otáček za minutu. Při výpočtu se hodnota 60 podělí aktuální hodnotou periody mezi dvěma výboji. Následně je počet otáček násoben konstantou  $m$ . Pokud je motor dvoudobý, je konstanta  $m$  rovna jedné. V případě čtyřdobého motoru jsou mezi dvěma výboji provedeny dvě otáčky, a proto je nutné vynásobit počet otáček dvěma.

$$f = \frac{60}{T} \cdot m, \quad [\text{ot/min; s, -}] \quad (5)$$

kde  $T$  změřená perioda mezi dvěma výboji na zapalovací cívce,  
 $m$  konstanta motoru, pro dvoudobý motor je  $m = 1$ , pro čtyřdobý je  $m = 2$ .

## 2.5.2 Maximální otáčky

Výpočet maximálních otáček motoru je shodný s výpočtem maximální rychlosti motocyklu. Pomocí podmínky if se porovnávají aktuální a maximální otáčky za minutu.

```
if (RPM>RPM_Max)           // porovnání aktuálních a maximální otáček
    RPM_Max = RPM;        // uložení maximálních otáček
```

## 2.5.3 Průměrné otáčky

Při výpočtu průměrných otáček motoru se postupně inkrementují násobky šedesáti sekund a sčítá se doba jednotlivých period. Výpočet průměrných otáček vychází ze vztahu č. 5.

```
RPM_Count++;                // inkrementace násobků šedesáti
RPM_Time_Avg+=Period_RPM;  // sčítání period
RPM_Avg=((60*RPM_Count)/RPM_Time_Avg)* Engine; // průměrné otáčky
```

## 2.6 Měření teploty

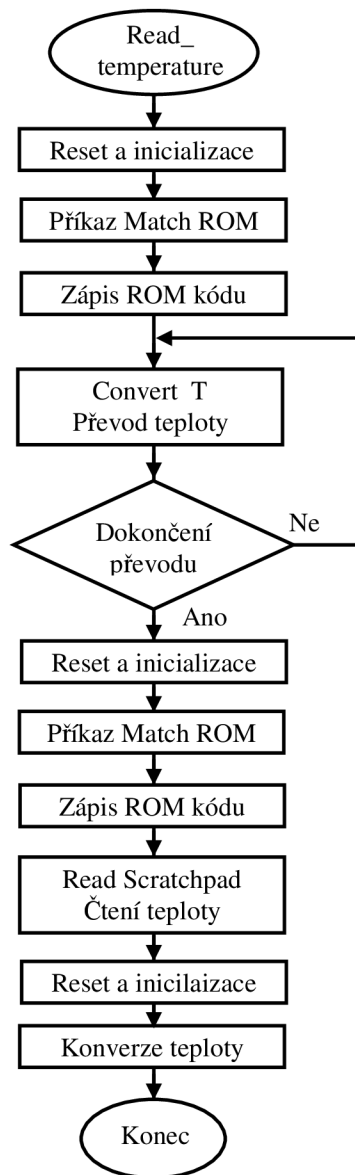
Měření teploty okolí, motoru, a v sání se provádí pomocí trojice teploměrů DS18B20, které komunikují pomocí rozhraní 1-Wire. Popis teploměrů a rozhraní je uveden v kapitolách 1.2.3 a 1.1.4. Pro komunikaci na rozhraní 1-Wire je použita knihovna s názvem DS18B20, která obsahuje soubor DS18B20.c a hlavičkový soubor DS18B20.h, viz [34]. Knihovna obsahuje základní příkazy k ovládání teploměru a popis pinů, na které je teploměr připojen, a v poslední řadě také jednotlivé funkce pro komunikaci s teploměrem. Stručný přehled použitých funkcí je lze nalézt v tabulce č. 4.

**Tab. 4:** Přehled použitých funkcí z knihovny DS18B20 (převzato z [34]).

Funkce	Popis
therm_reset(void)	Inicializace a vyresetování teploměru
therm_write_bit(uint8_t bit)	Zápis jednoho bitu do teploměru
therm_write_byte(uint8_t byte)	Zápis jednoho bajtu do teploměru
write_ROM_CODE(char *pt)	Zápis ROM kódu
therm_read_bit(void)	Čtení jednoho bitu z teploměru
therm_read_byte(void)	Čtení jednoho bajtu z teploměru

Při použití více teploměrů na rozhraní 1-Wire jsou dvě možnosti. První a jednodušší možností je zapojit každý teploměr zvlášť na jeden pin mikrokontroléru. Určitá nevýhoda je ve zbytečném plýtvání piny mikrokontroléru.

Druhou možností je zapojení všech tří teploměrů na jeden pin mikrokontroléru. Při realizaci zařízení byla vybrána právě tato alternativa. Aby bylo možné jednotlivé teploměry rozlišit, je nutné nejprve zjistit jejich ROM kódy. K tomuto účelu slouží dvě funkce. První funkcí je Search ROM, která zajišťuje identifikaci všech slave zařízení na sběrnici, viz [35]. Druhou funkcí je Read ROM. Tento příkaz lze použít pouze v případě, že je na sběrnici připojeno pouze jedno slave zařízení. Při návrhu obslužného softwaru byly nejprve zjištěny jednotlivé ROM kódy teploměrů pomocí funkce Read ROM a poté byla vytvořena funkce pro měření teploty `read_temperature`. Celý postup měření teploty je patrný z vývojového diagramu uvedeného na obr. 37.



**Obr. 37:** Vývojový diagram měření teploty pomocí teploměru DS18B20.

Na příkaz Scratchpad teploměr odešle zprávou o velikosti 9 bajtů. Uspořádání zprávy je uvedeno v tabulce č. 5. Pro určení teploty stačí přečíst první dva bajty označené bajt 0 a bajt 1. Druhý a třetí bajt je určen pro nastavení alarmu teploty. Konfigurační registr slouží k nastavení rozlišení teploměru. To lze nastavit v rozmezí od 9 do 12 bitů. Rozlišení teploměru bylo nejprve nastaveno na 12 bitů, ale doba převodu je velmi dlouhá a to 750 ms.

Proto bylo nastaveno 9bitové rozlišení, u kterého je doba převodu pouze 93,75 ms. Změna rozlišení se provádí příkazem `therm_write_byte(WRITE_SCRAT)`, po kterém teploměr očekává 3 bajty dat. V pořadí alarm horní bajt, alarm dolní bajt a nakonec bajt pro nastavení konfiguračního registru. Pro nastavení 9bitového rozlišení je nutné zaslat hodnotu `therm_write_byte(0x1F)`.

**Tab. 5:** Organizace paměti teploměru DS18B20, (převzato z [8]).

bajt 0	Teplota – dolní bajt
bajt 1	Teplota – horní bajt
bajt 2	Alarm – horní bajt
bajt 3	Alarm – dolní bajt
bajt 4	Konfigurační registr
bajt 5	Rezervováno 0xFF
bajt 6	Rezervováno
bajt 7	Rezervováno 0x10
bajt 8	CRC

Následná konverze dvou bajtů na číslo s dvěma desetinnými místy se provede pomocí následujícího zápisu.

```
digit=temperature[0]>>4; // celá část
digit|=(temperature[1]&0x7)<<4;

decimal=temperature[0]&0xf; // 2 desetinná místa
decimal*=THERM_DECIMAL_STEPS_9BIT/100;
```

Překročení teploty motoru a sání je signalizováno výpisem na displeji. Hranice zvýšené teploty lze nastavit v úvodním menu palubní jednotky.

## 2.7 Měření palubního napětí

Pro měření palubního napětí a také převod analogového napětí z akcelerometru se používá A/D převodník MCP3204, jehož popis je uveden v kapitole 1.4. Převodník komunikuje s mikrokontrolérem pomocí rozhraní SPI, viz kapitola 1.1.1.

Pro práci s převodníkem byly využity dvě knihovny. První knihovna s názvem SPI pro komunikaci po stejnojmenném rozhraní a druhá knihovna MCP3204 pro inicializaci a čtení dat z jednotlivých kanálů A/D převodníku, viz [36]. Přehled použitých funkcí je uveden v tabulce č. 6.

**Tab. 6:** Přehled použitých funkcí z knihovny SPI a MCP3204, (převzato z [36]).

Funkce	Popis
<code>SPI_Init()</code>	Inicializace rozhraní SPI
<code>SPI_Close()</code>	Zákaz interního SPI zařízení
<code>SPI_Write(uint8_t data)</code>	Zápis dat na rozhraní SPI
<code>Init_MCP3204()</code>	Inicializace A/D převodníku
<code>Read_MCP3204(uint8_t ch)</code>	Čtení dat z A/D převodníku



K inicializaci rozhraní SPI je nutné nastavit řídicí registry a také definovat piny mikrokontroléru, které rozhraní využívá. Nastavení inicializace je provedeno pomocí funkce `SPI_Init()`.

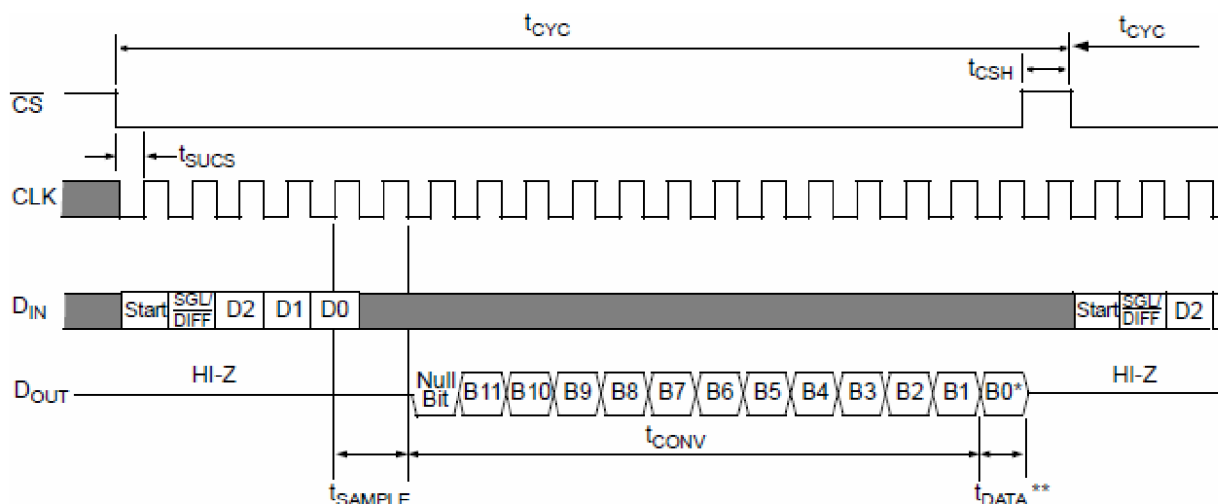
```
void SPI_Init()
{
    SPI_DDR |= ((1<<MOSI_POS) | (1<<SCK_POS) | (1<<SS_POS)); // nastavení portů
    SPI_PORT |= (1<<SS_POS); // nastavení vývodu SS
    CS_DDR |= (1<<CS_POS);
    CS_HIGH(); // nastavení CS do vysoké úrovně
    SPCR |= (1<<SPE) | (1<<MSTR) | (1<<SPR0); // aktivace SPI, master, rychlost
}
```

V řídicím registru SPCR je nastaven bit SPE pro aktivaci SPI kanálu, bit MSTR pro konfiguraci mikrokontroléru jako master a bit SPSR pro nastavení přenosové rychlosti 500 kHz.

Pro zápis dat se používá funkce `SPI_Write(uint8_t data)`. Pro vyslání dat stačí zapsat potřebná data do datového registru SPDR. Konec přenosu je indikován příznakem SPIF.

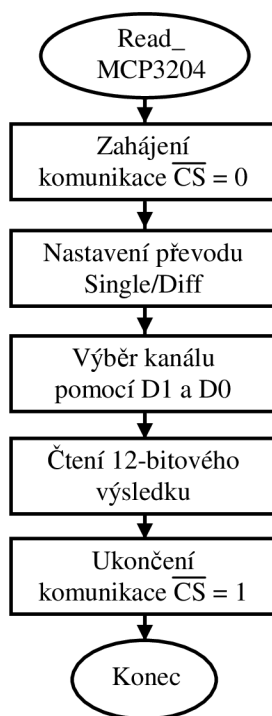
```
uint8_t SPI_Write(uint8_t data)
{
    SPDR = data; // start přenosu
    while(!(SPSR & (1<<SPIF))); // čekání na dokončení přenosu
    return (SPDR);
}
```

Ke čtení dat z A/D převodníku slouží funkce `Read_MCP3204(uint8_t ch)`. Proměnná `ch` značí číslo kanálu, ze kterého se provádí čtení. Průběh komunikace převodníku MCP3204 na sériovém kanálu SPI je uveden na obr. 38.



**Obr. 38:** Průběh komunikace A/D převodníku MCP3204 na rozhraní SPI (převzato z [15]).

Na začátku komunikace se nejprve nastaví vývod  $\overline{CS}$  na nízkou úroveň. Poté mikrokontrolér vyšle první bajt pro nastavení jednoduchého převodu pomocí bitu Single/Diff, který je nastaven na vysokou úroveň. V dalším bajtu se pomocí bitů D1 a D0 zvolí číslo kanálu, ze kterého se budou číst data. Bit D2 se používá pouze u 8kanálového převodníku. Následně A/D převodník vrací sekvenci začínající nulovacím bitem, za kterým následuje 12bitový výsledek A/D převodu. Celý postup komunikace s A/D převodníkem je patrný z vývojového diagramu na obr. 39.



**Obr. 39:** Vývojový diagram komunikace A/D převodníku MCP3204 na rozhraní SPI.

Po dokončení A/D převodu je vypočtena velikost palubního napětí. Jak je uvedeno v kapitole 1.4, je na vývodu CH3 převodníku zapojen dělič napětí. Nejprve se určí velikost napětí na kanálu CH3. Vzájemná souvislost mezi hodnotou vstupního napětí  $U_{IN}$  a její digitální reprezentaci AD je uveden ve vztahu č. 6.

$$AD = \frac{U_{IN}}{U_{REF}} \cdot 2^n, \quad [-; V, V, -] \quad (6)$$

kde  $AD$  digitální hodnota převedeného analogového napětí;  
 $U_{IN}$  velikost vstupního napětí;  
 $U_{REF}$  velikost referenčního napětí;  
 $n$  počet rozlišitelných úrovní A/D převodníku.

Velikost referenčního napětí A/D převodníku je 3,3 V a počet rozlišitelných úrovní je 4096.

$$\text{z toho plyne } \Rightarrow U_{IN} = \frac{AD \cdot U_{REF}}{2^n} = \frac{AD \cdot 3,3}{4096}, \quad [V; -, V, -]$$

Velikost palubního napětí se poté určí ze vztahu pro dělič napětí, viz vztah č. 2.

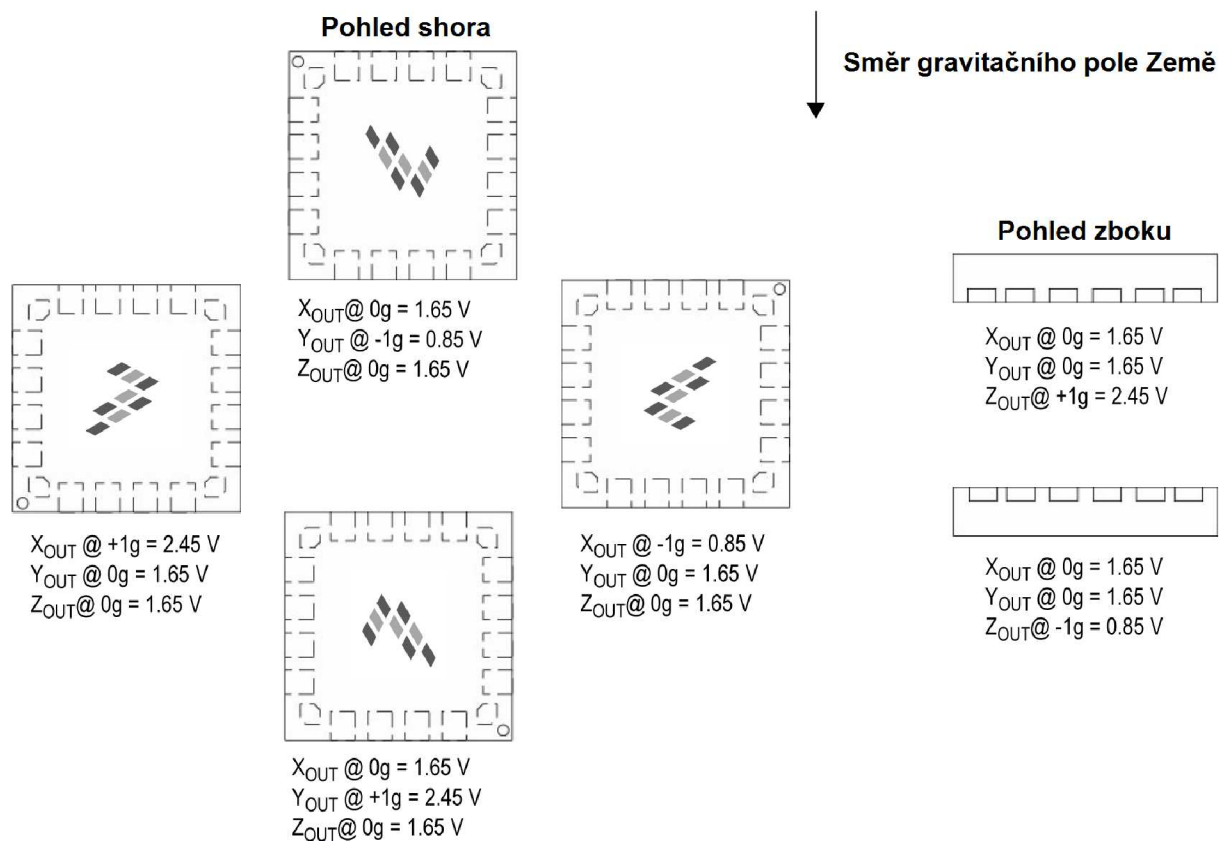
$$U_{palubni} = U_{IN} \cdot \frac{R_1 + R_2}{R_2} = U_{IN} \cdot \frac{9980 + 3290}{3290}, \quad [V; V, \Omega, \Omega]$$

Pro dosažení vyšší přesnosti jsou do vztahu dosazeny změřené hodnoty rezistorů  $R_1$  a  $R_2$ . Velikost napětí se zobrazuje na displeji s přesností na dvě desetinná místa. V případě přepětí je signalizace provedena výpisem varování na displej LCD.

## 2.8 Měření náklonu a stoupání

Pro měření náklonu motocyklu a stoupání cesty se používá akcelerometr, jehož popis je uveden v kapitole 1.2.4. Akcelerometr má tři analogové výstupy odpovídající jednotlivým osám natočení X, Y a Z. Pro převod analogových napětí je použit A/D převodník MCP3204, který současně slouží pro měření palubního napětí motocyklu, viz kapitola 2.7. Princip měření a převodu napětí je tedy totožný jako u měření palubního napětí. Jediný rozdíl spočívá ve výpočtu konečných hodnot.

Výhodou akcelerometru je možnost volby citlivosti 1,5 g, 2g, 4 g a 6 g. Pro měření byla zvolena nejvyšší citlivost a to 1,5 g. Závislosti výstupního napětí X, Y a Z na natočení akcelerometru při měření statického zrychlení udává obr. 40.



**Obr. 40:** Závislosti výstupního napětí X, Y a Z na natočení akcelerometru (převzato z [9]).

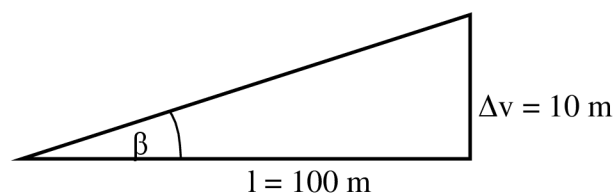
Hodnoty jednotlivých výstupních napětí akcelerometru jsou čteny z A/D převodníku pomocí funkce `Read_MCP3204()`. Následně je vypočítána velikost napětí podle vztahu č. 6. Vypočtená hodnota napětí se v dalším kroku přepočítá na statické tíhové zrychlení g. Tíhové zrychlení nabývá hodnot od -1 g do 1 g. Hodnota 0 g odpovídá vodorovné ose, neboli náklonu 0°. Hodnoty -1 g a 1 g odpovídají poloze osy kolmé k zemskému povrchu, tedy náklonu -90° a 90°. Přepočet se provede rozdílem aktuální hodnoty napětí a hodnoty odpovídající hodnotě 0 g, tedy 1,65 V. Tento rozdíl se vydělí hodnotou odpovídající rozdílu napětí mezi úrovněmi 1 g a 0 g. Tím je zajištěno, že hodnota zrychlení se bude vždy pohybovat v rozmezí -1 g až 1 g, viz [37].

Po výpočtu statického zrychlení se určuje náklonu motocyklu podle vztahu č. 7. Ve výpočtu se provádí převod z radiánů na stupně, protože funkce `asin()` z knihovny `math.h` vrací hodnotu úhlu v radiánech.

$$\alpha = \arcsin(g) \cdot \frac{180}{\pi}, \quad [^\circ; -, -] \quad (7)$$

kde  $\alpha$  úhel náklonu motocyklu;  
 $g$  statické tíhové zrychlení.

Zatímco náklon motocyklu se určuje ve stupních, stoupání nebo klesání cesty se udává v procentech. Při určení procent stoupání se vychází z výpočtu, o kolik metrů se změní výška na vzdálenosti 100 m. Rozdíl výšky např. 10 m odpovídá stoupání 10 %, viz obr. 41. Obecný vztah pro výpočet stoupání v procentech uvádí vztah č. 8.



**Obr. 41:** Závislosti procent stoupání na vzdálenosti a rozdílu výšky.

$$\% = \frac{\Delta v}{l} \cdot 100, \quad [\%; \text{m}, \text{m}] \quad (8)$$

kde  $\%$  procento stoupání;  
 $\Delta v$  rozdíl výšky;  
 $l$  vzdálenost.

Při přepočtu stoupání ve stupních na stoupání v procentech se vychází ze vztahu č. 8. Tangens úhlu  $\beta$  je dán poměrem  $v/l$ . Následným dosazením do vztahu č. 9 získáme hodnotu stoupání v procentech vypočtenou z úhlu  $\beta$ . Při zápisu v jazyce C byly místo funkce tangens využity funkce sinus a cosinus a opět musel být proveden převod z radiánů na stupně.

$$\% = \operatorname{tg}(\beta) \cdot 100 \cdot \frac{180}{\pi} = \frac{\sin(\beta)}{\cos(\beta)} \cdot 100 \cdot \frac{180}{\pi}, \quad [\%; ^\circ, ^\circ, \text{m}] \quad (9)$$

kde  $\%$  stoupání v procentech;  
 $\beta$  stoupání ve stupních.

Palubní jednotka tedy umožňuje měření náklonu motocyklu ve stupních a také stoupání cesty v procentech. Celkový výpočet stoupání se provede pomocí následujícího zápisu. Většina veličin je upravena do celočíselného tvaru, aby se předešlo použití proměnné `float`.

```
AD = Read_MCP3204(2); // čtení dat z A/D převodníku
Voltage = ((AD)*(3.3/4096)*1000); // výpočet napětí v mV
Gx = ((Voltage-1650)/8); // tíhové zrychlení 100 x větší
Anglex = (asin(Gx/100)); // úhel stoupání
Percent = ((sin(Anglex)/cos(Anglex))*180/3.14)*100; // procenta stoupání
```

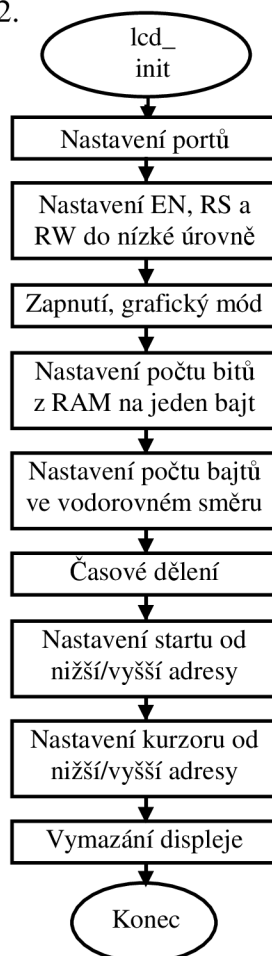
## 2.9 Zobrazení na displeji

Pro zobrazení měřených hodnot se používá grafický displej RG16080B, jehož popis je uveden v kapitole 1.3. Displej je řízen řadičem Sanyo LC7981. Pro ovládání řadiče se používá stejnojmenná knihovna lc7981, viz [38]. Knihovna obsahuje základní funkce pro práci v textovém nebo grafickém módu a také dva hlavičkové soubory s fonty písma o velikosti 8x8 a 12x16 pixelů. Při jízdě na motocyklu je vhodné použít větší velikost písma, a proto je displej využíván pouze v grafickém módu. Přehled základních použitých funkcí pro ovládání displeje je uveden v tabulce č. 7.

**Tab. 7:** Přehled použitých funkcí z knihovny lc7981, (převzato z [38]).

Funkce	Popis
lcd_init(uint8_t mode)	Inicializace displeje v textovém nebo grafickém módu
lcd_clear(void)	Vymazání displeje
lcd_clear_rect(uint8_t x, uint8_t y, uint8_t w, uint8_t h)	Vymazání předem definované plochy displeje. Vstupními hodnotami jsou souřadnice x, y a také velikost plochy w, h.
lcd_plot_text(uint8_t x, uint8_t y_off, const char *text, uint8_t fw, uint8_t fh, PGM_P font)	Zobrazení řetězce znaků na displeji. Vstupními hodnotami jsou souřadnice x, y. Zobrazovaný řetězec a také zvolený font písma.

Inicializace displeje `lcd_init()` se skládá z několika kroků. Postup inicializace je patrný z vývojového diagramu na obr. 42.



**Obr. 42:** Vývojový diagram inicializace displeje v grafickém módu.

Pro správnou funkci displeje je potřeba v hlavičkovém souboru lc7981.h nastavit porty, na kterých jsou zapojeny datové vývody DB0 až DB7, a také pět řídicích vývodů. V následujícím zdrojovém kódu je uvedeno využití knihovny lc7981. Jedná se o zobrazení aktuální rychlosti a denní vzdálenosti.

```
if (Speed>0)
lcd_clear_rect(30,0,60,16); // mazání při obnovení hodnoty
sprintf(buffer,"%3d km/h", Speed); // zápis hodnoty do bufferu
lcd_plot_text(30,0,buffer,16,16,font_12x16); // zobrazení font 12x16

lcd_plot_text(5,40,"Vzdalenost",8,8,font_8x8); // zobrazení font 8x8
sprintf(buffer,"Denni %5d,%1d km", Distance, Distance_Dec); // buffer
lcd_plot_text(5,55,buffer,8,8,font_8x8); // zobrazení font 8x8
```

V grafickém módu je potřeba při každém obnovení hodnoty provést vymazání předchozí hodnoty. K tomuto účelu slouží funkce `lcd_clear_rect()`, která vymaže předem zvolenou oblast displeje. Znaky na displeji jsou zobrazovány pomocí funkce `lcd_plot_text()`. V této funkci je třeba nastavit souřadnice x a y, zobrazovaný text a velikost fontu 8x8 nebo 12x16 pixelů. Zobrazení měření aktuální rychlosti, otáček, času a data je uvedeno na obr. 43.



**Obr. 43:** Zobrazení měřených hodnot na displeji.

K řízení jasu displeje se používá rychlý PWM režim. Vývod pro ovládání jasu displeje je připojen na vývod OC2 mikrokontroléru, na kterém je generován PWM signál. K řízení se tedy používá čítač/časovač 2. Nastavení čítače je provedeno pomocí funkce `Init_PWM()`. V registru TCCR2 je nastaven rychlý neinvertující PWM režim bez předděličky. Vývod OC2, který je na pinu PB7, musí být nastaven jako výstupní.

```
void Init_PWM() // inicializace rychlého PWM režimu
{
TCCR2|=(1<<WGM20)|(1<<WGM21)|(1<<COM21)|(1<<CS20); // neinvertující PWM
DDRB|=(1<<PB7); // nastavení vývodu OC2 jako výstupní
}
```

Řízení jasu je ovládáno pomocí tlačítka, které inkrementuje, nebo dekrementuje obsah registru OCR2. Velikost výstupního napětí na pinu OC2 se poté mění podle vztahu č. 10.

$$U_{oc2} = \frac{OCR2}{255} \cdot U_{cc}, \quad [V; -, - V] \quad (10)$$

kde  $U_{oc2}$  výstupní napětí na pinu OC2;  
 $OCR2$  registr pro nastavení rozlišení čítače;  
 $U_{cc}$  napájecí napětí.

## 2.10 Komunikace s obvodem reálného času

Obvod hodin reálného času DS1307 komunikuje s mikrokontrolérem pomocí sběrnice TWI. Popis sběrnice a obvodu je uveden v kapitolách 1.1.2 a 1.5. K ovládání obvodu slouží knihovna I<sup>2</sup>C Master Interface obsahující soubor twimaster.c a hlavičkový soubor i2cmaster.h, viz [39]. Knihovna zahrnuje základní funkce pro komunikaci obvodu na rozhraní TWI. Popis použitých funkcí z knihovny je uveden v tabulce č. 8.

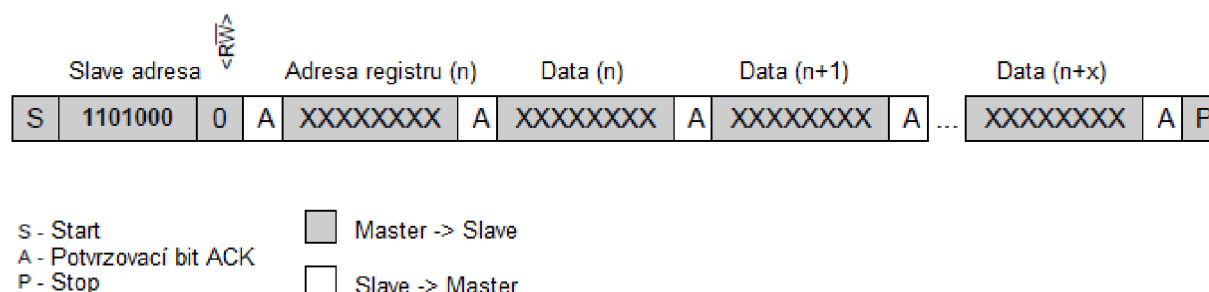
**Tab. 8:** Přehled použitých funkcí z knihovny I<sup>2</sup>C Master Interface (převzato z [39]).

Funkce	Popis
twi_init()	Inicializace rozhraní TWI
twi_stop()	Ukončení přenosu dat a uvolnění sběrnice TWI
twi_rep_start (unsigned char addr)	Opakování startovací podmínky, poslání adresy a směru přenosu
twi_start_wait(unsigned char addr)	Vytvoření startovací podmínky, poslání adresy a směru přenosu. Pokud je zařízení zaneprázdněno, tak čeká
twi_write (unsigned char data)	Poslání jednoho bajtu do zařízení TWI
twi_readNak (void)	Čtení jednoho bajtu z TWI zařízení a vytvoření ukončovací podmínky

K zápisu dat do obvodu reálného času slouží funkce `write_DS1307()`. Proměnná `reg` udává adresu registru, do kterého se bude zapisovat. Proměnná `val` představuje hodnotu, která se do registru zapíše.

```
unsigned char write_DS1307(unsigned char reg, unsigned char val)
{
    twi_start_wait(DS1307+TWI_WRITE); // adresa RTC obvodu a režim zápisu
    twi_write(reg); // zapis adresy registru
    twi_write(val); // zapis hodnoty do registru
    twi_stop(); // nastavení ukončovací podmínky
    return 0;
}
```

Průběh zápisu dat do obvodu RTC je uveden na obr. 44. Komunikace začíná startovací podmínkou, poté následuje slave adresa obvodu, adresa registru a data zapisovaná do registru. Komunikace končí ukončovací podmínkou. Mezi jednotlivými zapisovanými daty jsou vloženy potvrzovací bity ACK.



**Obr. 44:** Průběh zápisu do obvodu RTC na rozhraní TWI (převzato z [16]).

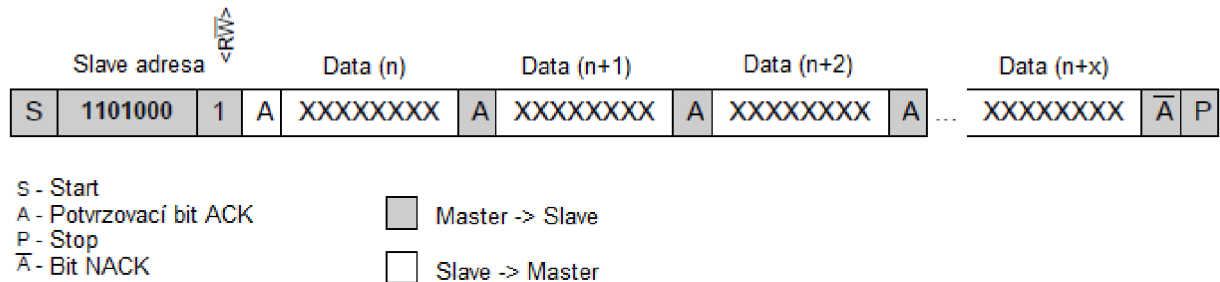
Ke čtení dat z obvodu RTC se používá funkce `read_DS1307()`. U této funkce je vstupní proměnná pouze registr, ze kterého se provádí čtení dat.

```

unsigned char read_DS1307 (unsigned char reg)
{
    unsigned char ret;
    twi_start_wait (DS1307+TWI_WRITE); // adresa RTC obvodu a režim zápisu
    twi_write (reg); // zapis adresy registru
    twi_rep_start (DS1307+TWI_READ); // adresa RTC obvodu a režim čtení
    ret = twi_readNak (); // čtení jednoho bytu z RTC
    twi_stop (); // nastavení ukončovací podmínky
    return ret;
}

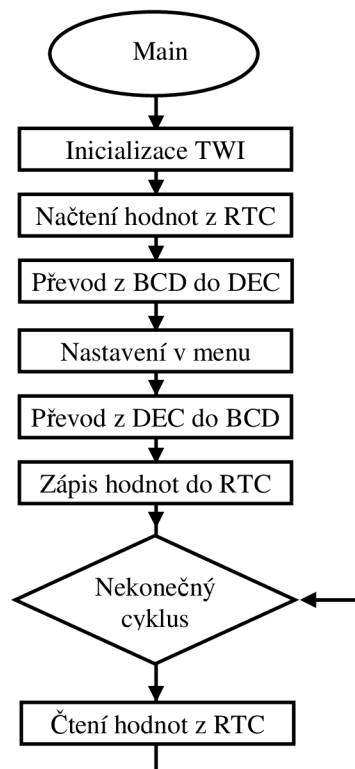
```

Průběh čtení dat z obvodu reálného času je uveden na obr. 45. Komunikace opět začíná startovací podmínkou a slave adresou obvodu. Poté následují data a ukončovací podmínka.



**Obr. 45:** Průběh čtení z obvodu RTC na rozhraní TWI (převzato z [16]).

Nastavení času a data probíhá v úvodním menu palubní jednotky. Veškeré informace jsou uloženy v registrech obvodu reálného času v BCD formátu, proto je potřeba provést převod do dekadické soustavy. Po zadání hodnot v úvodním menu se provádí zpětný převod z dekadické soustavy do BCD kódu. Celý postup komunikace s obvodem RTC je patrný z vývojového diagramu na obr. 46.



**Obr. 46:** Vývojový diagram komunikace s obvodem reálného času.



## 2.11 Komunikace s GPS modulem

K záznamu projeté trasy slouží GPS modul FGPMOPA6B, jehož popis je uveden v kapitole 1.6. GPS modul komunikuje s mikrokontrolérem pomocí sériového rozhraní USART1, viz 1.1.3. Formát vysílaných dat je 8N1, tedy 8 datových bitů, bez parity a jeden stop bit. Přestože výrobce udává implicitně nastavenou přenosovou rychlost 9600 baudů, skutečná přenosová rychlost modulu je 4800 baudů.

GPS modul používá komunikační protokol NMEA 0183 verze 3.01. Protokol NMEA se skládá z několika typů vět, které obsahují informace o čase, poloze, aktivních satelitech atd. Každá věta začíná znakem dolar, poté následuje identifikátor věty a data oddělená čárkou. Věta končí sekvencí <CR><LF>. GPS modul používá 5 základních typů vět. Jedná se o věty s názvem GGA, GSA, GSV, RMC a VTG. Popis jednotlivých vět je uveden v tabulce č. 9.

**Tab. 9:** Základní typy NMEA vět využívající GPS modul FGPMOPA6B (převzato z [17]).

Název věty	Popis
GGA	Čas, poloha, nadmořská výška a data vztahující se k GPS přijímači
GSA	Aktivní satelity použité pro určení polohy a hodnoty DOP (Dilution Of Precision)
GSV	Počet GPS satelitů včetně hodnoty ID. Elevace, azimut a hodnoty SNR
RMC	Čas, datum, poloha, kurz a rychlost. Minimální doporučená informace pro navigaci
VTG	Informace o kurzu a rychlosti vztahující se k zemi

Pro určení polohy lze použít věty GGA a RMC. Při vývoji obslužného programu byla zvolena věta GGA, protože obsahuje navíc i informace o nadmořské výšce. Význam jednotlivých informací je uveden v tabulce č. 10. Věta bývá zapsána v následujícím formátu.

```
$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65
```

**Tab. 10:** Význam jednotlivých částí věty GGA (převzato z [17]).

Název	Příklad	Jednotky	Popis
ID zprávy	\$GPGGA		Hlavička protokolu GGA
Čas UTC	064951.000		hhmmss.sss
Zeměpisná šířka	2307.1256		ddmm.mmmm
Indikátor N/S	N		N = sever, S = jih
Zeměpisná délka	12016.4438		dddmm.mmmm
Indikátor E/W	E		E = východ, W = západ
Indikátor kvality	1		0 – nelze určit pozici, 1 a 2 – pozice určena
Počet satelitů	8		Rozsah 0 až 14
HDOP	0.95		Vliv rozestavení družic na určení polohy
Výška MSL	39.9	metry	Nadmořská výška antény
Jednotky	M	metry	Jednotka pro nadmořskou výšku antény
Rozlišení geoidu	17.8	metry	Rozdíl mezi WGS-84 zemským elipsoidem a střední úrovní moře
Jednotky	M	metry	Jednotky rozlišení geoidu
Stáří aktualizace		sekundy	Nulové pole, pokud není DGPS použit
Kontrolní součet	*65		
<CR><LF>			Konec zprávy

Ke komunikaci po rozhraní USART1 se používá knihovna s názvem UART, viz [39]. Knihovna obsahuje základní funkce pro nastavení rámce a rychlosti přenosu a také funkce pro přenos jednotlivých znaků nebo řetězců. Přehled použitých funkcí je uveden v tabulce č. 12.

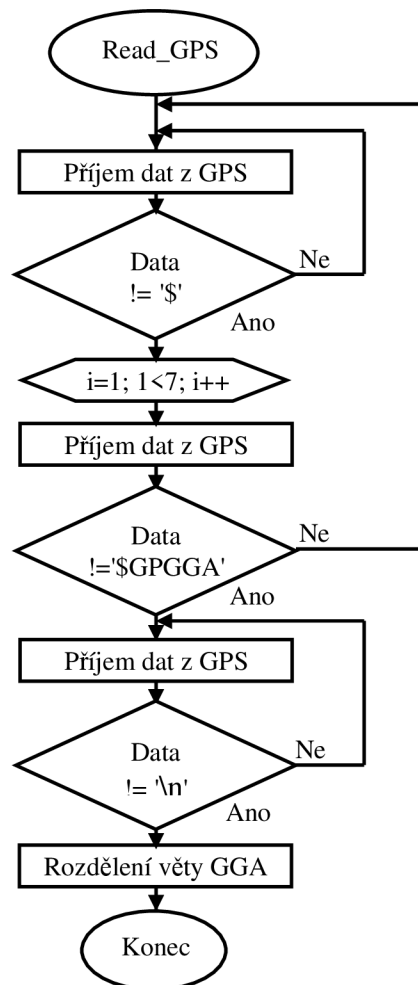
**Tab. 11:** Přehled použitých funkcí z knihovny UART (převzato z [39]).

Funkce	Popis
uart1_init()	Inicializace rozhraní USART1
uart1_getc()	Příjem jednoho bajtu

Pro příjem dat z GPS modulu přes rozhraní USART1 byla vytvořena funkce `gps_getc()`.

```
char gps_getc() { // funkce pro příjem dat z GPS
    unsigned int c = 0;
    do {
        c = uart1_getc(); // příjem jednoho bajtu
    } while((c & UART_NO_DATA) || (c & UART_FRAME_ERROR) ||
            (c & UART_OVERRUN_ERROR) || (c & UART_BUFFER_OVERFLOW));
    return((char) c);
}
```

V dalším kroku bylo potřeba napsat funkci pro příjem celé věty GGA. K tomuto účelu slouží funkce s názvem `read_GPS()`. Celý postup příjmu věty GGA je uveden ve vývojovém diagramu na obr. 47.



**Obr. 47:** Vývojový diagram získání věty GGA z GPS modulu.

Po získání celé věty GGA je potřeba provést její rozdělení na jednotlivá data oddělená čárkou. K rozdělení je využita funkce strtok z knihovny string.h. Funkce strtok se používá k rozdělení řetězce na tzv. tokeny, což jsou sekvence sousedících znaků oddělených předem definovaným znakem. V následujícím zdrojovém kódu je uveden postup rozdělení věty GGA.

```
token = strtok(nmea, ",");           // rozdělení NMEA věty na tokeny

while (token != NULL)
{
    point++;
    switch (point)
    {
        case 1:  ID=token;  break;    // ID zprávy
        case 2:  UTC=token; break;    // čas UTC
        case 3:  Lat=token; break;    // zeměpisná šířka
        case 4:  NS=token;  break;    // N/S indikátor
        ...
        case 15: Sum=token; break;    // kontrolní součet
    }
    token = strtok(NULL, ",");
}
```

Aktuální údaje o zeměpisné šířce, délce a nadmořské výšce jsou zobrazovány na grafickém displeji. Zároveň je testován počet dostupných satelitů. Jestliže je počet satelitů nulový, je na displeji zobrazen nápis “ Vyhledávám signál... “. Současně se zobrazením na displeji jsou celé GGA věty ukládány na SD kartu pro pozdější zpracování v počítači.

## 2.12 Ukládání dat

Ukládání dat je rozděleno na dvě části. První část dat se ukládá do paměti EEPROM mikrokontroléru. Jedná se o základní informace o jízdě na motocyklu. Do paměti lze uložit data za posledních 20 dnů. Uložení se provádí při ukončení měření, tedy při zastavení motocyklu. Po zaplnění paměti se data ukládají opět na začátek od prvního dne. Data jsou v paměti uspořádány do tabulky, ve které jsou obsaženy souhrnné informace za jeden den.

- datum jízdy ve formátu den.měsíc.rok,
- teplota okolí ve °C,
- napětí baterie ve voltech,
- čas jízdy ve formátu hodiny:minuty:sekundy,
- denní ujetá vzdálenost v km,
- průměrná a maximální rychlost v km/h,
- průměrné a maximální otáčky motoru v ot/min,
- celková ujetá vzdálenost v km.

Druhá část dat jsou NMEA věty GGA, které vysílá GPS modul. Tyto informace jsou vysílány každou sekundu, což zvyšuje nároky na kapacitu paměti. Proto jsou data z GPS modulu ukládány na SD kartu. Ukládání se provádí každé dvě sekundy. Na SD kartě jsou za sebou uloženy kompletní věty GGA od znaku dolar až po kontrolní součet. Ukládání celých vět GGA přináší výhodu v možnosti jejich následné analýzy v počítači.

```
$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,*,*65
```

## 2.12.1 Paměť EEPROM

K ovládání paměti EEPROM byla využita knihovna `avr/eeprom`, která je běžnou součástí vývojového prostředí AVR Studio. Knihovna obsahuje základní funkce pro zápis a čtení paměti EEPROM. Popis použitých funkcí je uveden v tabulce č. 12.

**Tab. 12:** Přehled použitých funkcí z knihovny `avr/eeprom` (převzato z [33]).

Funkce	Popis
<code>eeprom_write_byte (uint8_t *__p, uint8_t __value)</code>	Zápis jednoho bajtu na adresu v paměti
<code>eeprom_write_word (uint16_t *__p, uint16_t __value)</code>	Zápis 16-bitového slova na adresu v paměti
<code>eeprom_read_word (const uint16_t *__p)</code>	Čtení 16-bitového slova z adresy v paměti

Před zápisem dat do paměti EEPROM se provádí kontrola data posledního zápisu, aby nedošlo k přepsání dat. Nejprve se testuje den, potom měsíc a následně rok. Po otestování dojde k zápisu dat.

```
TTable = eeprom_read_byte(&Posledni);
if (TTable == 0xFF) TTable = 0;          // kontrola data posledního zápisu
else
{
    E = eeprom_read_word(&Table [TTable] [DAY]);      //kontrola dne
    if (E != MDate) TTable ++;
    else
    {
        E = eeprom_read_word(&Table[TTable] [MONTH]); //kontrola měsíce
        if (E != MMonth) TTable ++;
        else
        {
            E = eeprom_read_word(&Table [TTable] [YEAR]); //kontrola roku
            if (E != MYear) TTable ++;
        }
    }
}
```

K zápisu dat do paměti EEPROM se využívá funkce `eeprom_write_word()`. Pokud dojde k ukončení měření, data se uloží v následujícím formátu do vytvořené tabulky. Ukládání se provádí v obsluze čítače/časovače 1.

```
if (TTable >= DAYTAB) TTable = 0; // návrat na začátek po uložení 30 dne
eeprom_write_byte(&Last, TTable);
...
eeprom_write_word(&Table[TTable] [TIME_HOUR], Hour); // zápis hodin
eeprom_write_word(&Table[TTable] [TIME_MIN], Minute); // zápis minut
eeprom_write_word(&Table[TTable] [TIME_SEC], Second); // zápis sekund
```

Údaje o celkové ujeté vzdálenosti a také o nastaveném obvodu kola a typu motoru jsou uchovávány v paměti i při odpojení napájení. Po zapnutí přístroje jsou načteny z paměti pomocí funkce `eeprom_read_word()`.

```
Circuit_mm = eeprom_read_word(&Circuit); // obvod kola
Distance_Tot = eeprom_read_word(&SumKm); // celkové kilometry
Engine = eeprom_read_word(&TypeEngine); // typ motoru
```

## 2.12.2 SD karta

SD karta stejně jako A/D převodník komunikuje s mikrokontrolérem pomocí rozhraní SPI, viz 1.1.1. Pro ovládání SD karty jsou použity dvě knihovny. První knihovna s názvem `spi` slouží k řízení komunikace po rozhraní SPI. Druhá knihovna `SD_routines` slouží ke komunikaci s SD kartou. V této knihovně se nachází definice pinů  $\overline{CS}$  pro ovládání karty a také funkce pro inicializaci, čtení dat, zápis dat a mazání jednotlivých bloků karty. Popis použitých funkcí z knihovny `SD_routines` je uveden v tabulce č. 13.

**Tab. 13:** Přehled použitých funkcí z knihovny `SD_routines`.

Funkce	Popis
<code>SD_init()</code>	Inicializace SD karty
<code>SD_sendCommand(unsigned char cmd, unsigned long arg)</code>	Odeslání příkazu na SD kartu
<code>SD_readSingleBlock(unsigned long startBlock)</code>	Přečtení jedné stránky o velikosti 512 B z SD karty
<code>SD_writeSingleBlock(unsigned long startBlock)</code>	Zápis stránky o velikosti 512 B do SD karty
<code>SD_erase (unsigned long startBlock, unsigned long totalBlocks)</code>	Vymazání udaného počtu bloků SD karty

K inicializaci SPI rozhraní a přenosu dat jsou stejně jako u A/D převodníku použity funkce `SPI_Init()` a `SPI_Write()`, viz kapitola 2.7.

Data se ukládají na SD kartu po stránkách, které mají zpravidla velikost 512 bajtů. Určitý počet počátečních stránek je nepřepisovatelný. Informace o tom, která ze stránek je jako první přepisovatelná, je zapsána na nulté stránce SD karty. V knihovně `SD_routines` jsou definovány stránky označené `MyPage` a `NXTPage`. Obě stránky mají velikost 512 bajtů. Při zápisu dat se nejprve naplňuje stránka `MyPage`, která může být dle potřeby přeplněna do stránky `NXTPage`. Po zaplnění stránky `MyPage` je však nutné provést její uložení. V následujícím zdrojovém kódu je uveden celý proces uložení NMEA věty na SD kartu.

```
read_GPS(); // Funkce pro přečtení NMEA věty z GPS

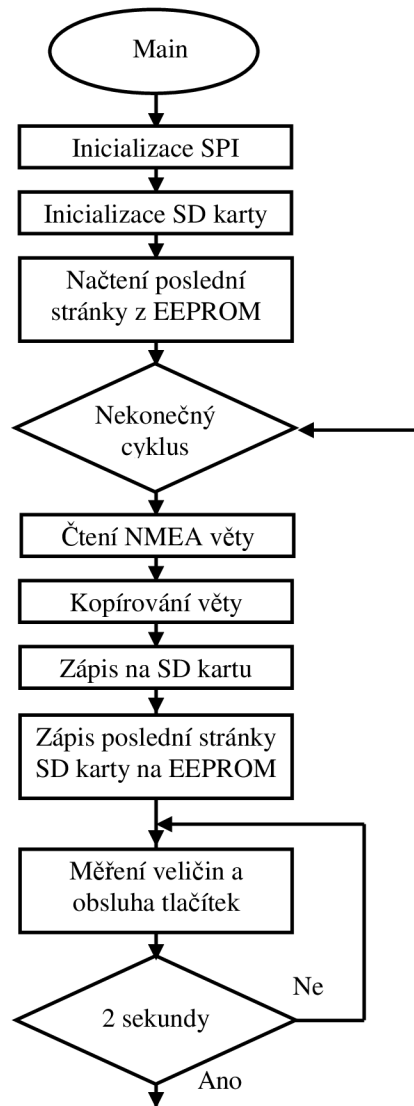
for (unsigned char a=0; GGA[a]!=0; a++) // kopírování věty do bufferu
    MyPage[PageOffset++] = GGA[a];

if (PageOffset >= MAXPAGE) // po zaplnění bufferu zápis na SD kartu
{
    SD_writeSingleBlock(LastSector++); //zápis na SD kartu
    if (LastSector >= 0x4F4C) LastSector = 300; // načtení od začátku
    eeprom_write_dword(&LastPG, LastSector); //zápis poslední stránky do EEPROM
    CLEARPAGE; // vymazání bufferu
    COPYPAGE; // zkopírování obsahu přetečeného přes buffer
    PageOffset -= MAXPAGE; // snížení adresace
}
```

Ukládání dat na SD kartu se provádí každé dvě sekundy. Doba uložení je řízena pomocí jednoduchého cyklu `do while`. K vytvoření času dvou sekund se používá obvod RTC.

```
do { // cyklus ve kterém se provádí měření hodnot
...
}while ( RTC_Second % 2 ); // po splnění podmínky dochází k uložení dat
```

Celý postup ukládání NMEA vět na SD kartu je patrný z vývojového diagramu na obr. 48.



**Obr. 48:** Vývojový diagram ukládána NMEA vět na SD kartu.

## 2.13 Přenos naměřených dat do PC

K přenosu naměřených dat z paměti EEPROM a z SD karty se používá rozhraní USART0, viz 1.1.3. Následný převod na rozhraní USB obstarává převodník FT232RL, jehož popis je uveden v kapitole 1.8. K obsluze jednotky USART0 je využita stejně jako u GPS modulu knihovna UART, viz [39]. Popis použitých funkcí z knihovny je uveden v tabulce č. 14.

**Tab. 14:** Přehled použitých funkcí z knihovny UART (převzato z [39]).

Funkce	Popis
uart_init()	Inicializace rozhraní USART0
uart_putc(unsigned char data)	Odeslání jednoho bajtu
uart_puts(const char *s )	Odeslání řetězce znaků

Před zahájením komunikace musel být nastaven formát asynchronního rámce a přenosová rychlost. Rámec byl nastaven v módu 8N1 a přenosová rychlost 4800 baudů. Struktura použitého rámce je uvedena v kapitole 1.1.3. na obr. 6. Pro inicializaci jednotky USART0 se používá funkce `uart_init()`, ve které se nastavuje přenosová rychlost v registru `UBRR0`, formát rámce v registru `UCSR0C` a povolení vysílače a přijímače pomocí bitů `TXEN0` a `RXEN0` v registru `UCSR0B`.

```
void uart_init(unsigned int baudrate)      // inicializace jednotky USART0
{
#ifdef ATMEGA_USART0
    UBRR0H = (unsigned char) (baudrate>>8); //nastavení přenosové rychlosti
    UBRR0L = (unsigned char) baudrate;

    UART0_CONTROL = _BV(RXCIE0) | (1<<RXEN0) | (1<<TXEN0); // povolení RX a TX

#ifdef URSEL0
    UCSR0C = (1<<URSEL0) | (3<<UCSZ00);      // formát asynchronního rámce 8N1
#else
    UCSR0C = (3<<UCSZ00);
#endif
}

```

Pro přenos jednotlivých znaků byla použita funkce `uart_putc()`, k přenosu řetězců po rozhraní USART0 se používá funkce `uart_puts()`.

```
void uart_puts(const char *s)             // funkce pro přenos řetězce
{
    while (*s)
        uart_putc(*s++);                 // vyslání řetězce znaků
}

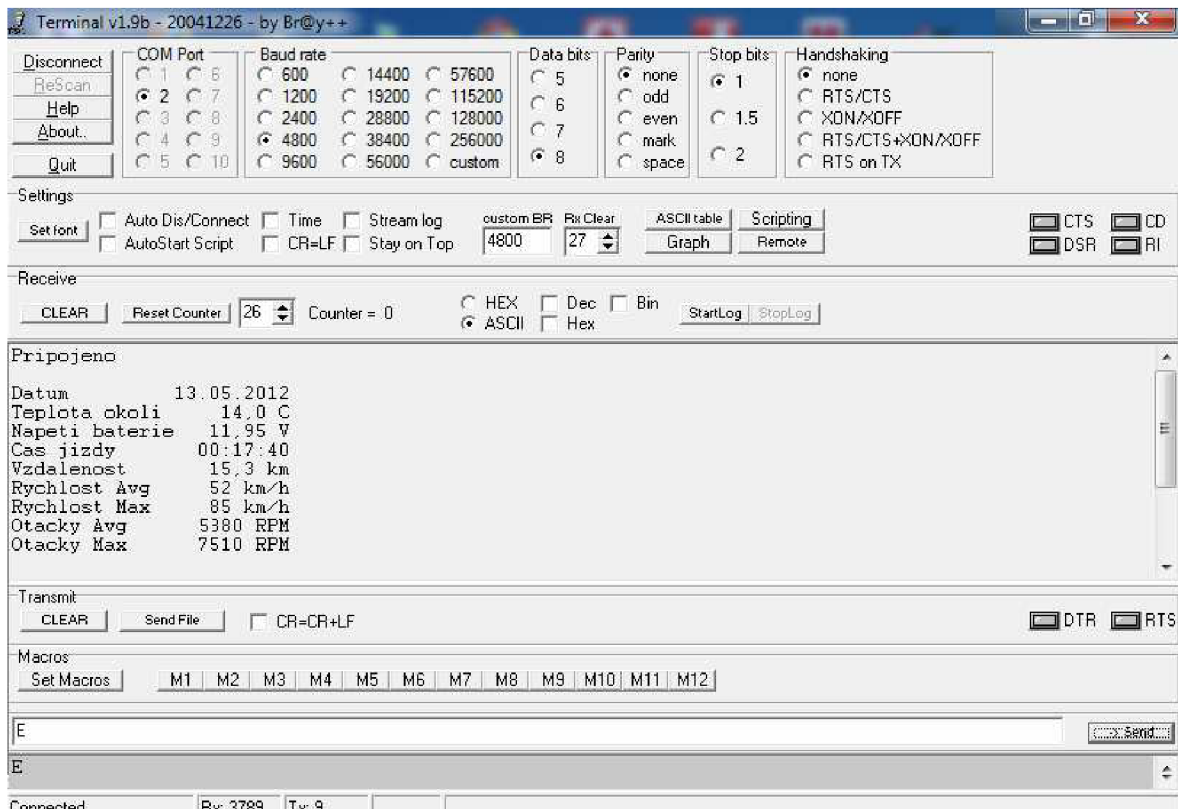
```

Palubní jednotka se připojuje k počítači pomocí mini USB konektoru. Po připojení je potřeba nainstalovat tzv. VCP ovladače pro vytvoření virtuálního sériového portu COM. Data lze poté vypsát např. v programech Terminal, nebo HyperTerminal, který je součástí Windows XP a starších. U Windows Vista a 7 jej lze však doinstalovat.

Pro výpis dat byl použit program Terminal verze 1.9b, který je volně dostupný na internetu. Velkou výhodou je jeho přehlednost a jednoduché nastavení. Program umožňuje přístup k portům COM1 až COM10 a také velmi přehledné nastavení přenosové rychlosti a formátu rámce. Výpis dat se provádí pomocí trojice kláves. Při stisknutí klávesy “P” se testuje, zda-li je jednotka správně připojena. Pokud ano, zobrazí se nápis “Připojeno”. Výpis dat z paměti EEPROM se provádí stiskem klávesy “E”. Výpis dat je uveden na obr. 49.

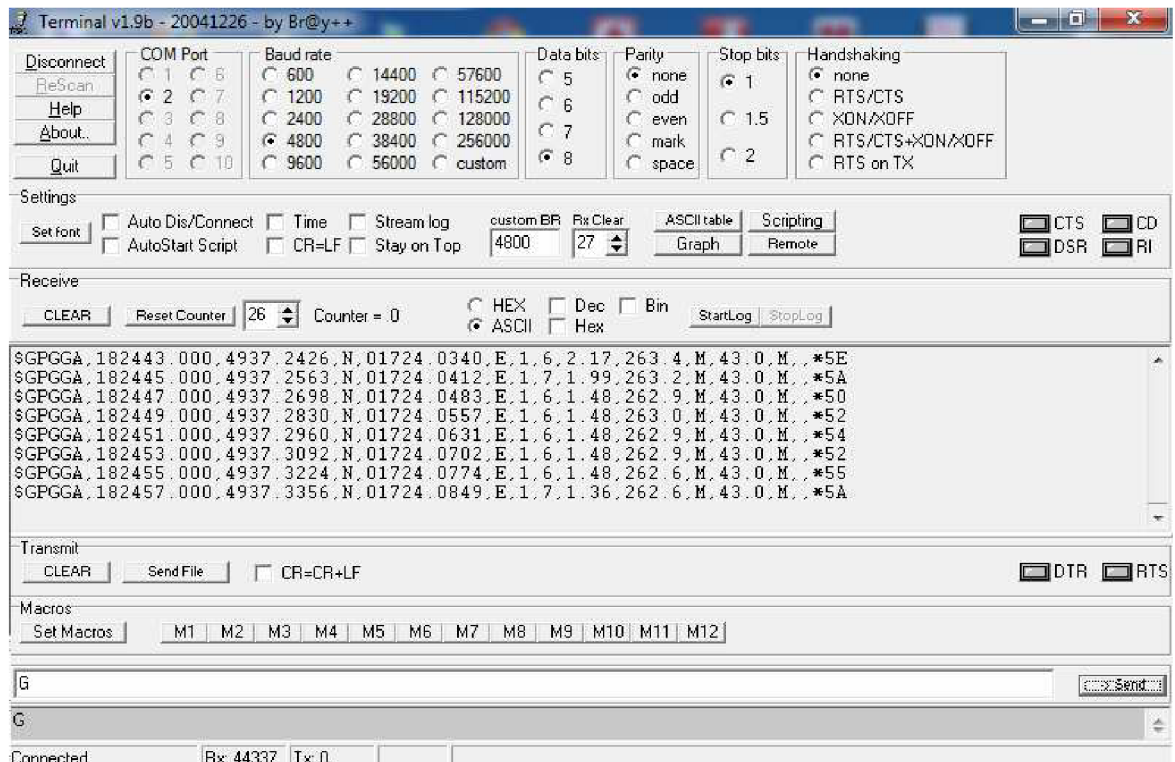
```
if (Touch == 'E') // při stisku klávesy E dojde k výpisu hodnot z EEPROM
{
    int i;
    for (i = 0; i < DAYTAB; i++) // výpis za posledních 14 dnů
    {
        ... // výpis ostatních hodnot
        sprintf(buffer, "Otacky Avg      %5d RPM\r\n",
            eeprom_read_word(&Table[i][AVG_RPM])); // načtení průměrných otáček
        uart_puts(buffer1); // vyslání přes UART
        sprintf(buffer1 "Otacky Max      %5d RPM\r\n\r\n",
            eeprom_read_word(&Table[i][MAX_RPM])); // načtení maximálních otáček
        uart_puts(buffer1); // vyslání přes UART
    }
}

```



**Obr. 49:** Výpis naměřených dat z paměti EEPROM v programu Terminal.

V případě požadavku podrobnější analýzy dat je možné provést výpis naměřených dat z GPS, která jsou uložena na SD kartě. Výpis uložených vět GGA se provádí stisknutím klávesy “G”. Zobrazení jednotlivých vět v programu Terminal je uvedeno na obr. 50.



**Obr. 50:** Výpis NMEA věty GGA z SD karty v programu Terminal.



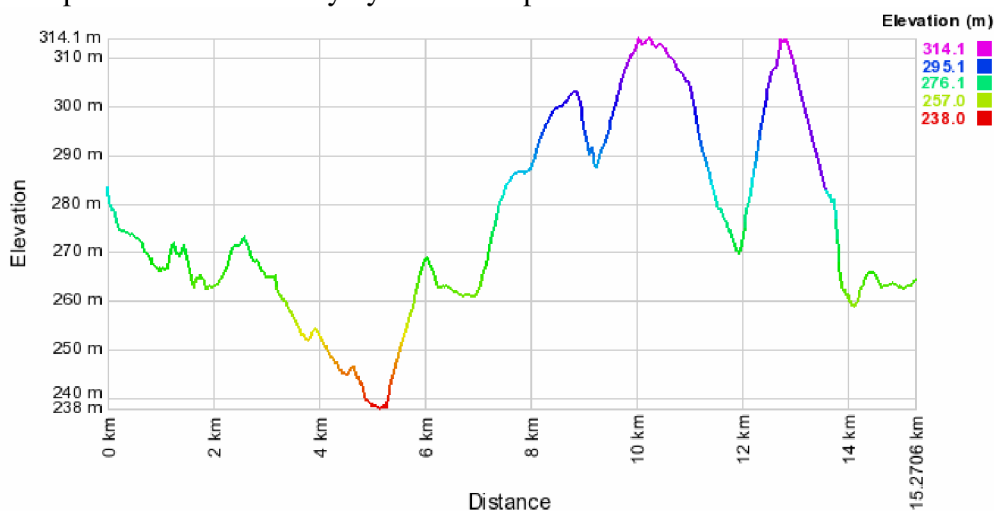
## 2.14 Zpracování naměřených dat v PC

Ukládání naměřených dat z GPS modulu na SD kartu poskytuje široké možnosti ve zpracování dat na počítači. Aby bylo možné data analyzovat, je potřeba provést jejich uložení z programu Terminal např. do textového souboru.

Jak již bylo uvedeno v kapitole 2.13, data jsou uložena ve formátu NMEA. Na internetu existuje velké množství online aplikací, které jsou schopny pracovat s daty s tímto formátem. Jednou z těchto aplikací je GPS Visualizer, viz [40]. Jedná se o online aplikaci k tvorbě map, výškových profilů a dalších funkcí. Nejjednodušší funkcí je zobrazení trasy na mapě. Po nahrání textového souboru dojde k vykreslení trasy ve formátu jpeg, png atd. Tímto způsobem lze zaznamenat i výškový profil. Nevýhodou je, že nelze provádět podrobnější analýzu jednotlivých úseků trasy. Vykreslení trasy na mapě je uvedeno na obr. 51. Výškový profil změřené trasy je uveden na obr. 52.

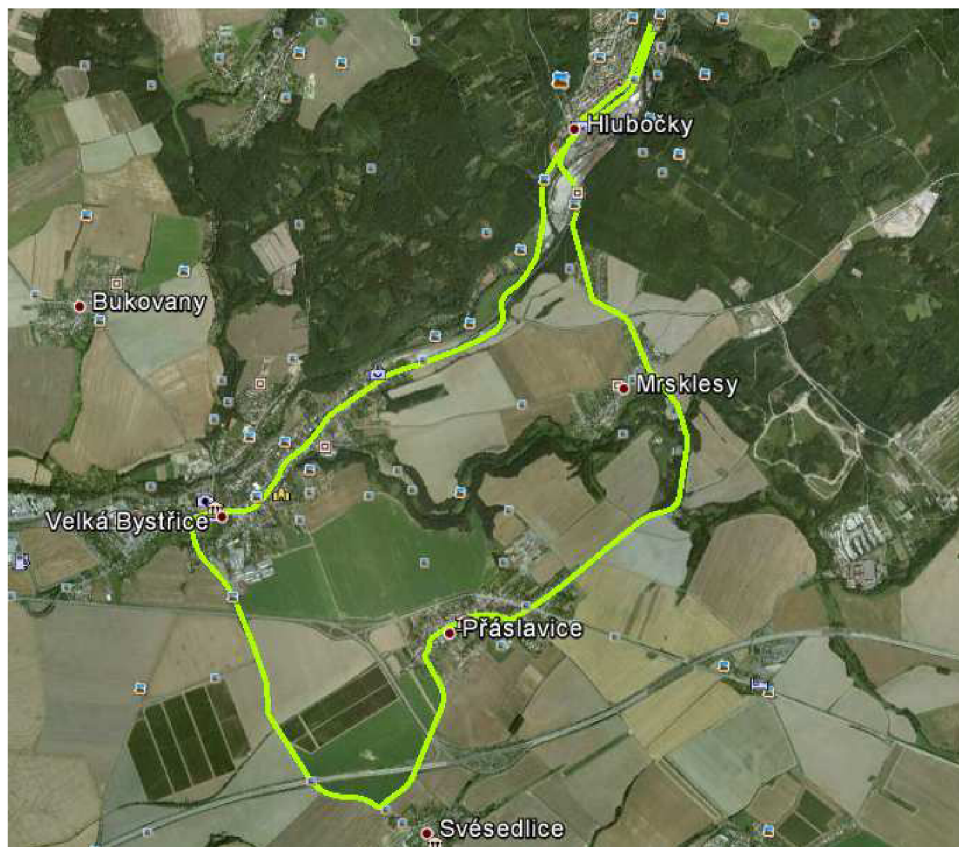


Obr. 51: Mapa zaznamenané trasy vytvořená v aplikaci GPS Visualizer.



Obr. 52: Výškový profil zaznamenané trasy vytvořený v aplikaci GPS Visualizer.

Pro rozsáhlejší analýzu lze v programu GPS Visualizer převést data ve formátu NMEA do formátu KML. Jedná se o formát, který využívá volně dostupná aplikace Google Earth, viz [41]. Po nainstalování programu Google Earth stačí vytvořený KML soubor otevřít a následně dojde k vykreslení trasy na mapě. Záznam z palubní jednotky je zobrazen na obr. 53. Jedná se o záznam, který byl vytvořen při testování jednotky.



**Obr. 53:** Mapa zaznamenané trasy vytvořená v aplikaci Google Earth.

Kromě vykreslení trasy na mapě umožňuje Google Earth také zobrazení výškového profilu trasy. V profilu jsou dále uvedeny informace o celkové ujeté vzdálenosti, celkovém stoupání i klesání cesty, maximálním stoupání i klesání a také o průměrném sklonu a nadmořské výšce. Výškový profil naměřený při záznamu trasy palubní jednotkou je uveden na obr. 54.



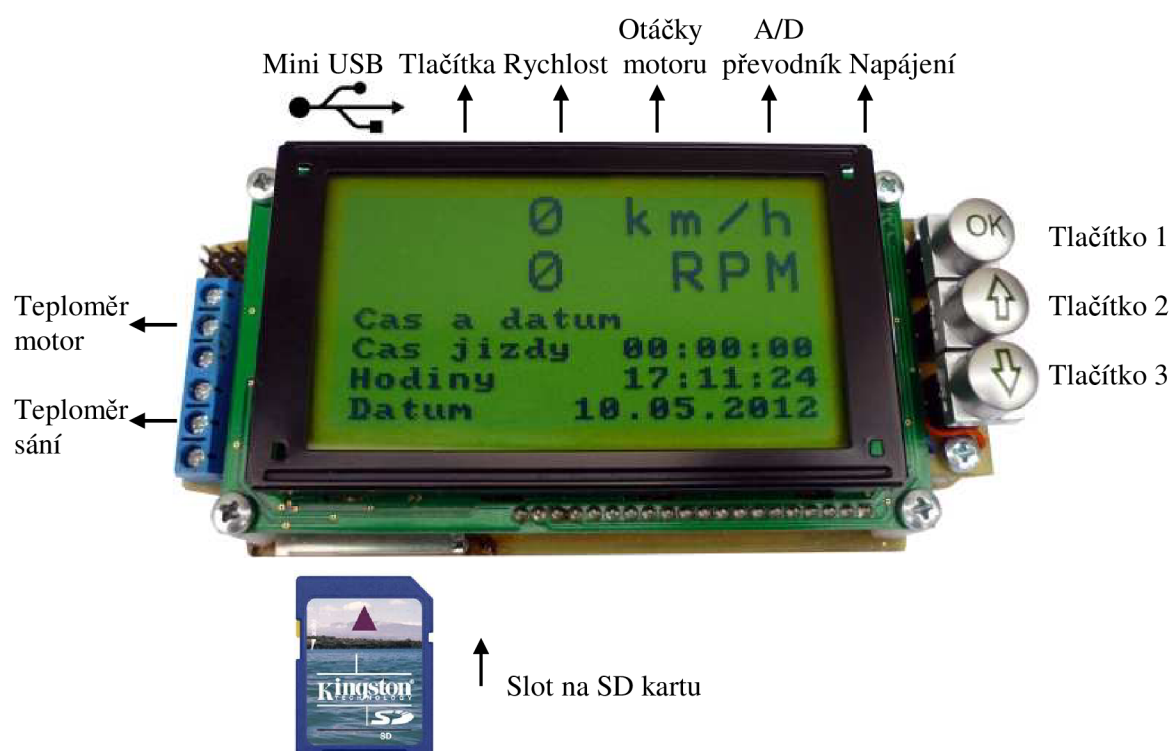
**Obr. 54:** Výškový profil zaznamenané trasy vytvořený v aplikaci Google Earth.

Pohybem kurzoru lze zkoumat jednotlivé úseky trasy a zobrazovat si tak aktuální sklon, ujetou vzdálenost a nadmořskou výšku. Pomocí programu Google Earth lze tedy provést kompletní analýzu zaznamenaných dat z palubní jednotky.




## 3 Uživatelský manuál

### 3.1 Uspořádání palubní jednotky

Celkový vzhled palubní multifunkční jednotky je uveden na obr. 55. V horní části přístroje jsou umístěny svorkovnice pro připojení k napájecímu napětí (baterie motocyklu), dále svorkovnice pro připojení k A/D převodníku, pro připojení stíněného kabelu pro měření otáček a jazýčkového snímače pro měření rychlosti. Poslední svorkovnice je pro připojení ovládacího panelu s tlačítky. Vedle svorkovnic je umístěn mini USB konektor umožňující připojit jednotku k počítači. V levé části jednotky jsou dvě svorkovnice pro připojení teploměrů pro měření teploty motoru a v sání. Ve spodní části přístroje je slot pro vložení SD karty. V pravé části jednotky je umístěn ovládací panel skládající se z trojice tlačítek.





**Obr. 55:** Celkový vzhled palubní multifunkční jednotky.

- Tlačítko 1  - úvodní menu- vstup do podmenu  
- pohyb v podmenu  
- výstup z podmenu i úvodního menu  
- hlavní menu - vymazání hodnot kromě celkové vzdálenosti  
- nastavení jasu displeje
- Tlačítko 2  - úvodní menu- pohyb v menu  
- změna hodnot v podmenu  
- hlavní menu - pohyb v menu
- Tlačítko 3  - úvodní menu- pohyb v menu  
- změna hodnot v podmenu  
- hlavní menu - pohyb v menu

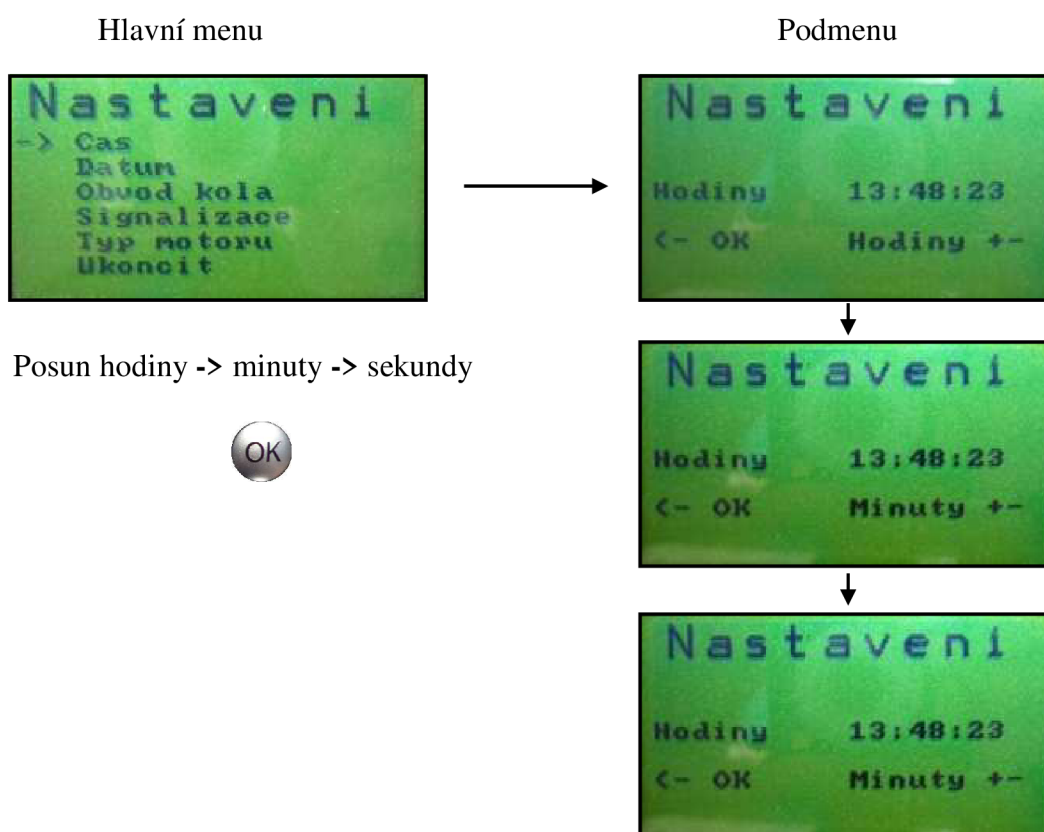
## 3.2 Úvodní nastavení palubní jednotky

Po připojení jednotky k napájecímu napětí je možná provést úvodní nastavení přístroje. V úvodním menu lze nastavit aktuální čas a datum, obvod kola motocyklu, hodnoty pro signalizaci zvýšené teploty a přepětí a typ motoru motocyklu.

- vstup do podmenu i výstup z podmenu se provádí pomocí tlačítka 
- změna hodnot v podmenu se provádí pomocí dvojice tlačítek 

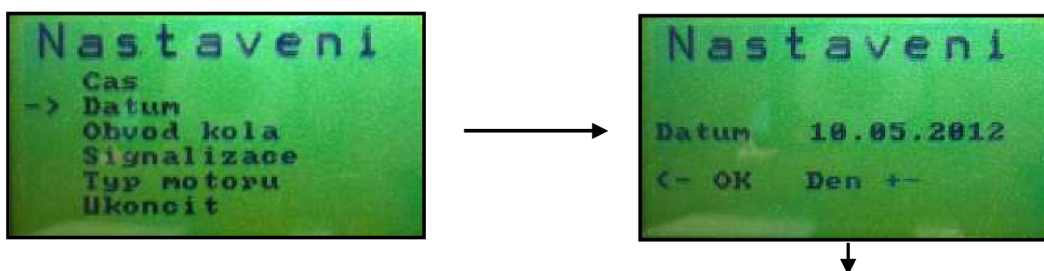
### 3.2.1 Nastavení času

Nastavení času se provádí v pořadí hodiny, minuty a sekundy.



### 3.2.2 Nastavení data

Nastavení data se provádí v pořadí den, měsíc a rok.

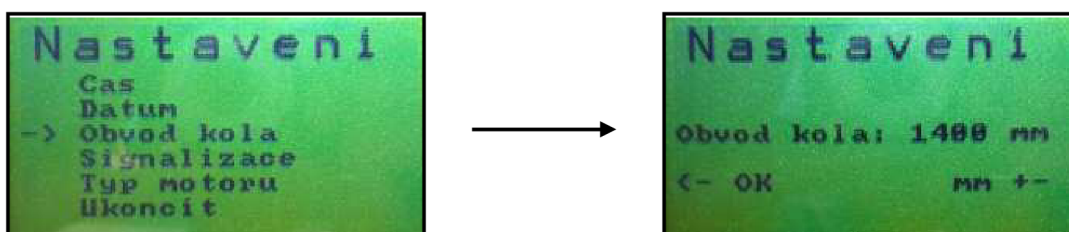


Posun dny -> měsíce -> roky



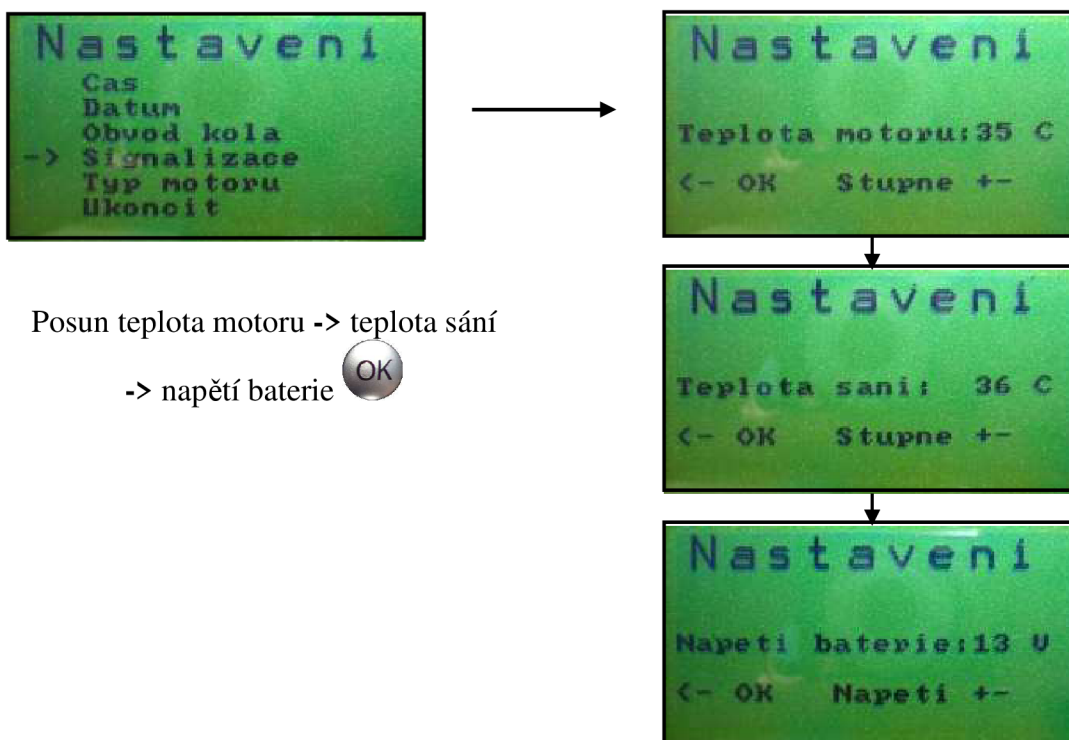
### 3.2.3 Nastavení obvodu kola motocyklu

Nastavení obvodu kola se provádí v milimetrech.



### 3.2.4 Nastavení signalizace

Nastavení signalizace při překročení nastavené teploty motoru, teploty sání a napětí baterie.



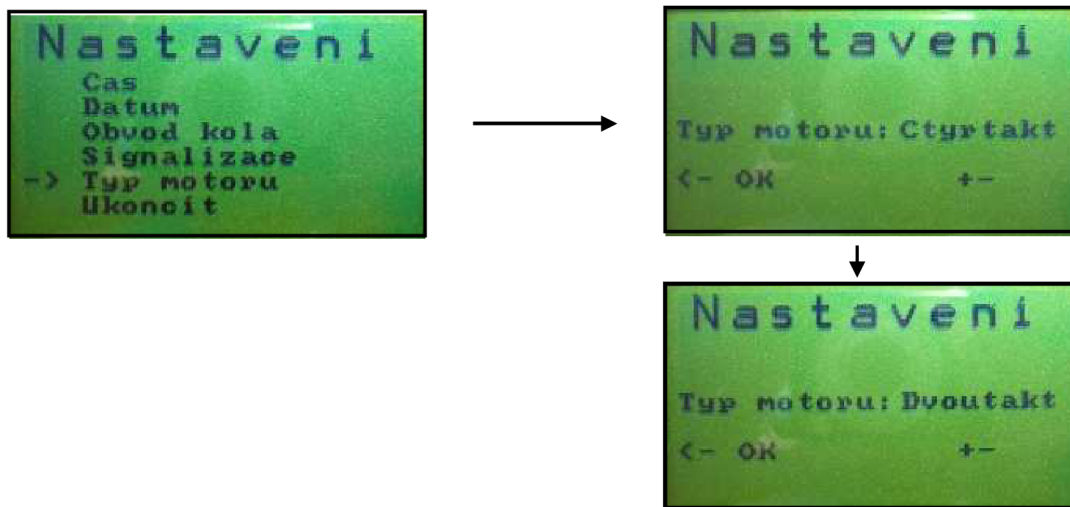
Posun teplota motoru -> teplota sání

-> napětí baterie




### 3.2.5 Nastavení typu motoru

Nastavení typu motoru, lze zvolit mezi dvoudobým a čtyřdobým motorem.






### 3.2.6 Výstup z úvodního menu

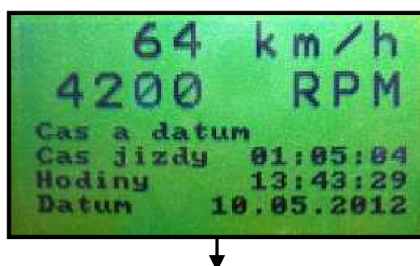
Výstup z úvodního menu se provádí pomocí tlačítka .



## 3.3 Hlavní menu palubní jednotky

Po úvodním nastavení jednotky následuje hlavní menu, ve kterém se provádí jednotlivá měření. V horní části displeje se zobrazuje měření aktuální rychlosti a otáček motoru.

- mazání měřených hodnot a nastavení jasu displeje se provádí pomocí tlačítka 
- posun mezi jednotlivými funkcemi se provádí pomocí dvojice tlačítek  



Menu 1: Funkce času a data

- Čas jízdy ve formátu hh:mm:ss
- Aktuální čas ve formátu hh:mm:ss
- Aktuální datum ve formátu dd.mm.rr



Menu 2: Funkce měření rychlosti

- Průměrná rychlost v km/h
- Maximální rychlost v km/h



Menu 3: Funkce měření vzdálenosti

- Denní ujetá vzdálenost v km
- Celková ujetá vzdálenost v km



Menu 4: Funkce měření otáček motoru

- Průměrné otáčky motoru v ot/min
- Maximální otáčky motoru v ot/min



Menu 5: Funkce měření napětí baterie

- Napětí baterie ve Voltech
- Úroveň nabití baterie



Menu 6: Funkce měření náklonu motocyklu

- Aktuální náklon motocyklu ve stupních
- Maximální náklon motocyklu ve stupních



Menu 7: Funkce měření stoupání cesty

- Aktuální stoupání cesty v procentech
- Maximální stoupání cesty v procentech



Menu 8: Funkce měření teploty

- Aktuální teplota okolí ve °C
- Aktuální teplota motoru ve °C
- Aktuální teplota v sání ve °C



#### Menu 6: Funkce měření polohy a výšky

- Zeměpisná šířka
- Zeměpisná délka
- Nadmořská výška v m.n.m

V případě překročení nastaveného limitu teploty motoru, teploty v sání, nebo napětí baterie se provádí signalizace výpisem varování na displej. Používají se tři typy varování. Výpis varování při překročení teploty motoru je zobrazen na obr. 56.

- M ! – znak pro překročení teploty motoru
- S ! – znak pro překročení teploty v sání
- B ! – znak při přepětí baterie



**Obr. 56:** Signalizace při překročení povolené teploty motoru.



## 4 Výsledky práce

### 4.1 Realizace hardwarové části

Při realizaci hardwarové části byla palubní jednotka nejprve sestavena na nepájivém kontaktním poli, aby se předešlo případným chybám při návrhu plošného spoje.

V první řadě bylo otestováno napájení jednotky. Použité stabilizátory napětí byly připojeny k baterii motocyklu a bylo změřeno zvlnění napájecího napětí. Tento údaj byl zvláště důležitý pro správnou funkci GPS modulu, u kterého výrobce požaduje zvlnění napájecího napětí menší než  $50 \text{ mV}_{pp}$ . Po zapojení feritové perličky a dvojice kondenzátorů na napájecí vývod GPS modulu bylo dosaženo zvlnění napětí  $40 \text{ mV}_{pp}$ , což vyhovuje požadavkům výrobce.

Jako nejproblématictější se ukázala komunikace s grafickým displejem, kdy po mnoha pokusech nebylo možné nastavit kontrast displeje. Problém byl vyřešen až po konzultaci s technickým oddělením Taiwanského výrobce Raystar, kdy bylo zjištěno, že technická dokumentace displeje obsahuje chyby. Kontrast displeje neměl být ovládán napětím v rozsahu od 0 do 5 V jak bylo uvedeno, nýbrž napětím v rozsahu od  $-10 \text{ V}$  do  $5 \text{ V}$ . V důsledku toho muselo být doplněno schéma zapojení jednotky o dva napěťové měniče.

Následně bylo provedeno otestování A/D převodníku, GPS modulu a použitých senzorů. Po otestování byl navržen a osazen oboustranný plošný spoj palubní jednotky a také jednostranný plošný spoj pro ovládací panel s tlačítky.

### 4.2 Realizace softwarové části

Při realizaci softwarové části bylo prvním krokem otestování správné funkce mikrokontroléru ATmega128 tak, že byla vyčtena jeho signatura. Po úspěšném ověření následovalo zprovoznění komunikace s grafickým displejem. Zde bylo nutné změnit u mikrokontroléru nastavení fuse bitu M103C. Mikrokontrolér ATmega128 je totiž standardně nastaven do módu kompatibility s ATmega103 a port C, na kterém jsou datové vývody displeje pracuje v tomto módu pouze jako výstupní. Zobrazení měřených hodnot na displeji výrazně urychlilo další vývoj obslužného programu.

Postupně byly naprogramovány funkce měření rychlosti, otáček motoru, palubního napětí, teploty okolí, teploty motoru a teploty v sání. Palubní jednotka byla také vybavena o nadstandardní funkce, jako je měření náklonu motocyklu, stoupání cesty, měření zeměpisné polohy a nadmořské výšky, včetně záznamu trasy z GPS modulu.

Posledním krokem bylo naprogramování ukládání naměřených dat na SD kartu a do paměti EEPROM mikrokontroléru. Uložené hodnoty lze přenést pomocí USB rozhraní do počítače a provést tak podrobnou analýzu všech dat.

### 4.3 Testování a technické parametry

Palubní jednotka byla otestována na motocyklu Suzuki V-Strom 650. Jedná se o motocykl se čtyřdobým motorem do V, objemem  $645 \text{ ccm}$  a maximálním výkonem  $67 \text{ k}$  při  $8800 \text{ ot/min}$ .

Po instalaci jednotky na řídítka motocyklu a zapojení jednotlivých senzorů bylo provedeno nastavení obvodu kola a typu motoru v úvodním menu. Umístění jednotky na motocyklu je zobrazeno na obr. 57.



**Obr. 57:** Palubní jednotky na motocyklu Suzuki V-Strom 650.

Po úvodním nastavení proběhlo testování palubní jednotky na zvolené trase. Změřená aktuální rychlost a otáčky motoru byly souběžně porovnávány s hodnotami, které ukazovaly rychloměr a otáčkoměr motocyklu. Obě sledované hodnoty se shodovaly a odchylky byly zanedbatelné. Po ujetí vzdálenosti 15,3 km zobrazovaly obě zařízení shodnou vzdálenost. Současně s měřením rychlosti a otáček byly sledovány další veličiny, jako napětí baterie, teplota okolí, teplota motoru, teplota v sání, náklon motocyklu a stoupání cesty. Při měření stoupání cesty bylo zjištěno, že měření je částečně ovlivňováno působením dynamického zrychlení ve směru jízdy.

Palubní jednotka po celou dobu zaznamenávala pomocí GPS modulu projetou trasu a hodnoty ukládala na SD kartu. Příjem GPS signálu byl po celou dobu velmi dobrý a jednotka byla neustále připojena k šesti až deseti družicím. Výpis zaznamenaných dat je uveden v kapitole 2.13. Vykreslení projeté trasy na mapě včetně výškového profilu je uveden v kapitole 2.14.

Po otestování jednotky na motocyklu proběhlo měření spotřeby elektrické energie. Největší spotřebu energie má grafický displej a GPS modul. U displeje se pohybuje odběr proudu okolo 45 mA při vypnutém podsvícení a 300 mA při maximálním podsvícení. U GPS modulu se odběr proudu pohybuje od 35 mA do 45 mA. Spotřeba celé multifunkční jednotky je při vypnutém podsvícení displeje 155 mA. Při zapnutém podsvícení je celkový odběr proudu 195 mA. Maximální podsvícení se z důvodu vysoké spotřeby nevyužívá. V tabulce č. 15. jsou uvedeny technické parametry palubní multifunkční jednotky.

**Tab. 15:** Technické parametry palubní multifunkční jednotky.

Rozsah měřených hodnot		
Rychlost motocyklu	1 až 255	km/h
Otáčky dvoudobého motoru	15 až 12000	ot/min
Otáčky čtyřdobého motoru	30 až 24000	ot/min
Čas jízdy	0 až 255:59:59	h:min:s
Denní vzdálenost	0,1 až 65535	km
Celková vzdálenost	0,1 až 65535	km
Napětí baterie	0,01 až 13	V
Náklon motocyklu	0 až 90	°
Stoupání cesty	0 až 100	%
Teplota okolí	-55 až +125	°C
Teplota motoru	-55 až +125	°C
Teplota v sání	-55 až +125	°C
Napájení		
Napájecí napětí	7 až 35	V
Spotřeba proudu	155 až 195	mA
Rozměry a hmotnost		
Šířka	129,2	mm
Délka	67,5	mm
Výška	28	mm
Hmotnost	245	g

## Závěr

Diplomová práce se zabývá návrhem a realizací palubní multifunkční jednotky pro motocykly. Při návrhu hardwarové části zařízení byl nejprve vybrán vhodný typ mikrokontroléru. Důležitým požadavkem při výběru byl dostatečný počet vstupně/výstupních portů a také velikosti programové a datové paměti. Tyto požadavky zcela splnil zvolený mikrokontrolér ATmega128.

K měření rychlosti motocyklu byl vybrán jazýčkový snímač. Pro měření otáček motoru bylo nutné najít řešení, které by umožňovalo měření otáček bez toho, aniž by se muselo zasahovat do konstrukce motoru. Proto byl zvolen způsob měření periody mezi dvěma výboji na zapalovací cívce. K měření teploty okolí, teploty motoru a teploty v sání byly vybrány digitální teplotní senzory DS18B20, které komunikují pomocí sběrnice 1-Wire. Tyto senzory také plní funkci indikace zvýšené teploty. Pro měření palubního napětí slouží externí A/D převodník MCP3204, který zároveň převádí analogové napětí na výstupech akcelerometru. Akcelerometr se používá pro měření náklonu motocyklu a sklonu cesty. Naměřené hodnoty se zobrazují na grafickém LCD displeji s rozlišením 160x80 bodů. K zobrazení hodnot jsou využity dvě velikosti písma. Pro větší přehlednost jsou rychlost a otáčky zobrazeny větším fontem písma než ostatní měřené hodnoty. K vytvoření aktuálního času a data slouží obvod hodin reálného času DS1307. Aktuální zeměpisná poloha a nadmořská výška se určují pomocí GPS modulu. Data z GPS modulu jsou následně ukládány na paměťovou kartu SD s kapacitou paměti 2 GB. Ostatní naměřené hodnoty se ukládají do paměti EEPROM mikrokontroléru. Uložené hodnoty je možné přenést prostřednictvím USB rozhraní do počítače a provést tak podrobnou analýzu dat. Připojení k PC obstarává převodník USB na UART typ FT232RL.

Palubní jednotka je napájena z 12V baterie motocyklu. Ke změně napájecího napětí z 12 V na 5 V slouží stabilizátor napětí 7805 STM. SD karta a GPS modul vyžadují napájecí napětí nižší. Proto byly použity další dva stabilizátory napětí, které upravují napájecí napětí na hodnotu 3,3 V.

K vývoji obslužného programu palubní jednotky byl vybrán programovací jazyk C a vývojové prostředí AVR Studio. Při realizaci softwarové části byly postupně naprogramovány funkce měření rychlosti, otáček, teploty a palubního napětí. Následně byly doprogramovány nadstandardní funkce, jako je měření náklonu motocyklu, stoupání cesty, měření zeměpisné polohy, nadmořské výšky, ukládání a přenos dat do počítače. Po vytvoření obslužného softwaru byl sepsán uživatelský manuál popisující základní funkce a nastavení přístroje.

Palubní jednotka byla otestována na motocyklu Suzuki V-Strom 650. Při testování byly porovnávány hodnoty, které ukazovaly přístroje motocyklu s hodnotami měřenými palubní jednotkou. Po přenosu dat do počítače byly tyto údaje porovnány se záznamem trasy z GPS modulu. Celý proces testování palubní jednotky je uveden v kapitole 4.3.

## Seznam literatury

- [1] ATMEL. *ATmega128, 8-bit AVR Microcontroller with 64K Bytes In-System Programmable Flash* [online]. Leden 2011 - [cit. duben 2011]. Dostupné z WWW: <[http://www.atmel.com/dyn/resources/prod\\_documents/doc2467.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf)>.
- [2] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL ATmega16, 4. díl*. Praha: BEN – technická literatura, 2006. 320 pages. ISBN 80-7300-174-8.
- [3] HRBÁČEK, J. *Komunikace mikrokontroléru s okolím 1 díl*. Praha: BEN – technická literatura, 1999, 1. vyd., 159 stran. ISBN 80-86056-42-2.
- [4] MALÝ, Martin. *Sběrnice 1-Wire*. Hw.cz [online]. Listopad 2004 - [cit. Duben 2011]. Dostupné z WWW: <<http://hw.cz/rozhrani/art1215-sbernice-1-wire.html>>.
- [5] FROHN, M.; OBERTHÜR, W.; SIEDLER, H.-J.; WIEMER, M.; ZASTROW, P. *Elektronika – polovodičové součástky a základní zapojení*. Praha: BEN – technická literatura, 2006. 480 pages. ISBN 80-7300-123-3.
- [6] NETÁHLO, T. *Cyklopočítač s měřičem frekvence*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2010. 74 s., 5 s. příloh. Bakalářská práce.
- [7] BROŽEK, Tomáš. *Otáčkoměr pro spalovací motory*. Hw.cz [online]. Únor 2004 - [cit. Listopad 2011]. Dostupné z WWW: <<http://www.hw.cz/constrc/auto/otackomer/index.html>>.
- [8] MAXIM-DALLAS. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer* [online]. Duben 2008 - [cit. duben 2011]. Dostupné z WWW: <<http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>>.
- [9] FREESCALE SEMICONDUCTOR. *±1.5g - 6g Three Axis Low-g Micromachined Accelerometer* [online]. Březen 2008 - [cit. duben 2011]. Dostupné z WWW: <[http://www.freescale.com/files/sensors/doc/data\\_sheet/MMA7260QT.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf)>.
- [10] POLOLU ROBOTICS & ELEKTRONICS. *MMA7260QT 3-Axis Akcelerometr ±1.5/2/4/ 6g* [online]. Březen 2008 - [cit. duben 2011]. Dostupné z WWW: <<http://www.pololu.com/catalog/product/766>>.
- [11] KUČERA, Jan. *Grafické displeje a jejich řízení*. Hw.cz [online]. Červen 2002 - [cit. Duben 2011]. Dostupné z WWW: <<http://hw.cz/Firemni-clanky/Elatec/ART309-Graficke-LCD-displeje-a-jejich-rizeni.html>>.
- [12] SANYO. *LC7981 FOR THE LC DOT MATRIX GRAPHIC DISPLAY*. [online]. [cit. listopad 2011]. Dostupné z WWW: <[http://www.datasheetcatalog.org/datasheet/sanyo/ds\\_pdf\\_e/LC7981.pdf](http://www.datasheetcatalog.org/datasheet/sanyo/ds_pdf_e/LC7981.pdf)>.

- [13] RAYSTAR. *RG16080B-YHY-X*. [online]. Červenec 2008 - [cit. listopad 2011]. Dostupné z WWW: < <http://www.tme.eu/dok/R/rg16080b-yhy-x.pdf>>.
- [14] NATIONAL SEMICONDUCTOR. *LMC7660 Switched Capacitor Voltage Converter*. [online]. Únor 2005 - [cit. březen 2012]. Dostupné z WWW: < <http://www.ti.com/lit/ds/symlink/lmc7660.pdf>>.
- [15] MICROCHIP. *MCP 3204/32082.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI Serial Interface*. [online]. Lisopad 2010 - [cit. prosinec 2011]. Dostupné z WWW: < <http://ww1.microchip.com/downloads/en/DeviceDoc/21298e.pdf>>.
- [16] MAXIM-DALLAS. *DS1307, 64 x 8, Serial, I2C Real-Time Clock* [online]. Říjen 2008 - [cit. listopad 2011]. Dostupné z WWW: <<http://datasheets.maxim-ic.com/en/ds/DS1307.pdf>>.
- [17] GLOBALTOP. *66-channel GPS Engine Board Antenna Module with MTK Chipset FGPMOPA6B* [online]. Listopad 2010 - [cit. duben 2011]. Dostupné z WWW: <<http://www.gtop-tech.com/ShowAttach?UNID=1faedead382979956bba9bcdf7588b20ff757e4>>.
- [18] MEDIATEK. *MT3329*. [online].[cit. duben 2011]. Dostupné z WWW: <<http://www.mediatek.com/en/product/info.php?sn=50>>.
- [19] NATIONAL MARINE ELECTRONICS ASSOCIATION. *NMEA 0183 Standart*. [online].[cit. duben 2011]. Dostupné z WWW: < [http://www.nmea.org/content/nmea\\_standards/nmea\\_083\\_v\\_400.asp](http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp)>.
- [20] MURATA MANUFACTURING Co., Ltd. *Chip Ferrite Beads BLM31A Series*. [online].[cit. březen 2012]. Dostupné z WWW: < <http://www.tme.eu/dok/B/BLM31AF700SN1L-Murata-datasheet-153800.pdf> >.
- [21] NXP SEMICONDUCTORS. *HEF4050B Hex Non-Inverting Buffers*. [online].[cit. březen 2012]. Dostupné z WWW: <[http://www.nxp.com/documents/data\\_sheet/HEF4050B.pdf](http://www.nxp.com/documents/data_sheet/HEF4050B.pdf)>.
- [22] KINGSTON. *Secure Digital Cards – Standard*. [online]. [cit. duben 2011]. Dostupné z WWW: < <http://www.kingston.com/flash/securedigital.asp?id=2>>.
- [23] MULTICOMP. *SDB-MF- SD Memory Card Push- Push W/Connector* . [online]. [cit. březen 2012]. Dostupné z WWW: < <http://www.farnell.com/datasheets/358993.pdf>>.
- [24] FTDI CHIP. *FT232R USB UART ( USB - Serial) I.C.*[online]. Prosinec 2009 - [cit. duben 2011]. Dostupné z WWW: <[http://www.ftdichip.com/Documents/DataSheets/DS\\_FT232R\\_V205.pdf](http://www.ftdichip.com/Documents/DataSheets/DS_FT232R_V205.pdf)>.
- [25] MATOUŠEK, David. *USB prakticky s obvody FDTI – 1 díl; Měření, řízení a regulace pomocí několika jednoduchých přípravků*. Praha: BEN – technická literatura, 2003. 262 pages. ISBN 80-7300-103-9.

- [26] OLMR, Vít. *Terminal – sériový terminál pro Windows*. [online]. Květen 2000 -[cit. duben 2011]. Dostupné z WWW:< <http://hw.cz/teorie-praxe/art1928-terminal-seriovy-terminal-pro-windows.html>>.
- [27] STMICROELECTRONICS. *L7800 Series Positive Voltage Regulators*. [online]. [cit. březen 2012]. Dostupné z WWW: < <http://www.gme.cz/dokumentace/330/330-149/dsh.330-149.1.pdf> >.
- [28] LINEAR TECHNOLOGY. *LT1121 - Micropower Low Dropout Regulators with Shutdown* [online]. [cit. duben 2011]. Dostupné z WWW: <<http://cde.linear.com/docs/Datasheet/1121ff.pdf>>.
- [29] MATOUŠEK, David. *Číslicová technika, základy konstruktérské praxe*. Praha: BEN – technická literatura, 2001. 208 pages. ISBN 80-7300-025-3.
- [30] ŠENDERA, Josef. *Návrh plošných spojů pro povrchovou montáž*. Praha: BEN – technická literatura, 2006. 272 pages. ISBN 80-7300-181-0.
- [31] CADSOFT. *EAGLE Layout Editor* [online]. [cit. listopad 2011]. Dostupné z WWW: <<http://www.cadsoft.de/index.htm>>.
- [32] ATMEL. *AVRISP mkII In-System Programmer* [online]. 2005 - [cit. duben 2011]. Dostupné z WWW: < [http://www.atmel.com/Images/AVRISPMkII\\_UG.pdf](http://www.atmel.com/Images/AVRISPMkII_UG.pdf) >.
- [33] ATMEL. *AVR Studio 4*. [online]. [cit. duben 2012]. Dostupné z WWW:< <http://www.atmel.com/tools/AVRSTUDIO4.aspx> >.
- [34] MARULL PARETASS, Gepard. *Using DS18B20 digital temperature sensor on AVR Microcontrollers*. [online]. [cit. duben 2012]. Dostupné z WWW:< [http://teslabs.com/openplayer/docs/docs/other/ds18b20\\_pre1.pdf](http://teslabs.com/openplayer/docs/docs/other/ds18b20_pre1.pdf)>.
- [35] HRBÁČEK, J. *Komunikace mikrokontroléru s okolím 2 díl*. Praha: BEN – technická literatura, 2000, 1. vyd., 151 stran. ISBN 80-86056-73-2.
- [36] EXTREME ELECTRONICS. *Interfacing 12 bit SPI ADC (MCP3204) with AVR Micro*. [online]. [cit. duben 2012]. Dostupné z WWW:< <http://extremeelectronics.co.in/avr-tutorials/interfacing-12-bit-spi-adc-mcp3204-with-avr-micro/>>.
- [37] GÁBRLÍK, Petr. *Test akcelerometru Freescale MMA7260QT* [online]. [cit. duben 2012]. Dostupné z WWW:< [http://pandatron.cz/?776&test\\_akcelerometru\\_freescale\\_mma7260qt](http://pandatron.cz/?776&test_akcelerometru_freescale_mma7260qt)>.
- [38] SCHUMB, Sebastian. *AVR library for lc7981*. [online]. [cit. duben 2012]. Dostupné z WWW:<<http://www.sebastians-site.de/homepage/elektronik/avr-library-for-lc7981.html>>.
- [39] FLEURY, Peter. *Peter Fleury Online*. [online]. 2006 -[cit. květen 2012]. Dostupné z WWW:< <http://jump.to/fleury> >.

- [40] GPS Visualizer, [online]. [cit. květen 2012]. Dostupné z WWW:<<http://www.gpsvisualizer.com/>>.
- [41] Google Earth, [online]. [cit. květen 2012]. Dostupné z WWW:< <http://www.google.com/intl/cs/earth/index.html>>.



## Seznam symbolů, veličin a zkratk

AD	digitální hodnota převedeného analogového napětí
BAUD	přenosová rychlost v bitech za sekundu, bps
BCD	Binary Coded Decimal, dvojkově reprezentované dekadické číslo
CPU	Central Processing Unit, procesor
CS	Chip Select, výběr čipu
EAGLE	Easily Applicable Graphical Layout Editor
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná paměť ROM
EMI	Electromagnetic interference, elektromagnetická interference
$f_{clk\_IO}$	pracovní kmitočet mikrokontroléru
$f_{OSC}$	hodinový signál mikrokontroléru
FTDI	Future Technology Devices International
$g$	statické tíhové zrychlení.
I <sup>2</sup> C	Inter-Integrated Circuit, sériová sběrnice
JSA	Jazyk symbolických adres
JTAG	Joint Test Action Group, standart pro testování plošných spojů, pamětí atd
$l$	vzdálenost.
LCD	Liquid Crystal Display, displej z tekutých krystalů
LSB	Least Significant Bit, nejméně významný bit
$m$	konstanta motoru
MCU	Microcontroller, mikrokontrolér
MIPS	Milion Instructions per Second, milion instrukcí za sekundu
MISO	Master In Slave Out, vstup do masteru, výstup ze slavu
MOSI	Master Out Slave In, výstup z masteru, vstup do slavu
MSB	Most Significant Bit, nejvíce významný bit
$n$	počet rozlišitelných úrovní A/D převodníku.
$N$	dělicí poměr
NMEA	National Marine Electronics Association
$o$	nastavený obvod kola motocyklu
OCR1A	Output Compare Register, registr pro manipulaci s rozlišením čítače
OCR2	registr pro nastavení rozlišení čítače
PWM	Pulse Width Modulation, pulzně šířková modulace
$R_1$	odpor $R_1$ děliče
$R_2$	odpor $R_2$ děliče
RICS	Reduced Instruction Set Computer, počítač s redukovanou instrukční sadou
RTC	Real-Time Clock, hodiny reálného času
SCK	Serial Clock, hodinový signál
SCL	Synchronous Clock, hodinový signál
SDA	Synchronous Data, datový kanál
SMD	Surface Mount Device, součástky pro povrchovou montáž
SPI	Serial Peripheral Interface, sériové periferní rozhraní
$\overline{SS}$	Slave Select, výběr zařízení slave
STN	Super Twisted Nematic, pasivní LCD
$T$	perioda
TWI	Two-Wire Serial Interface, sériové rozhraní

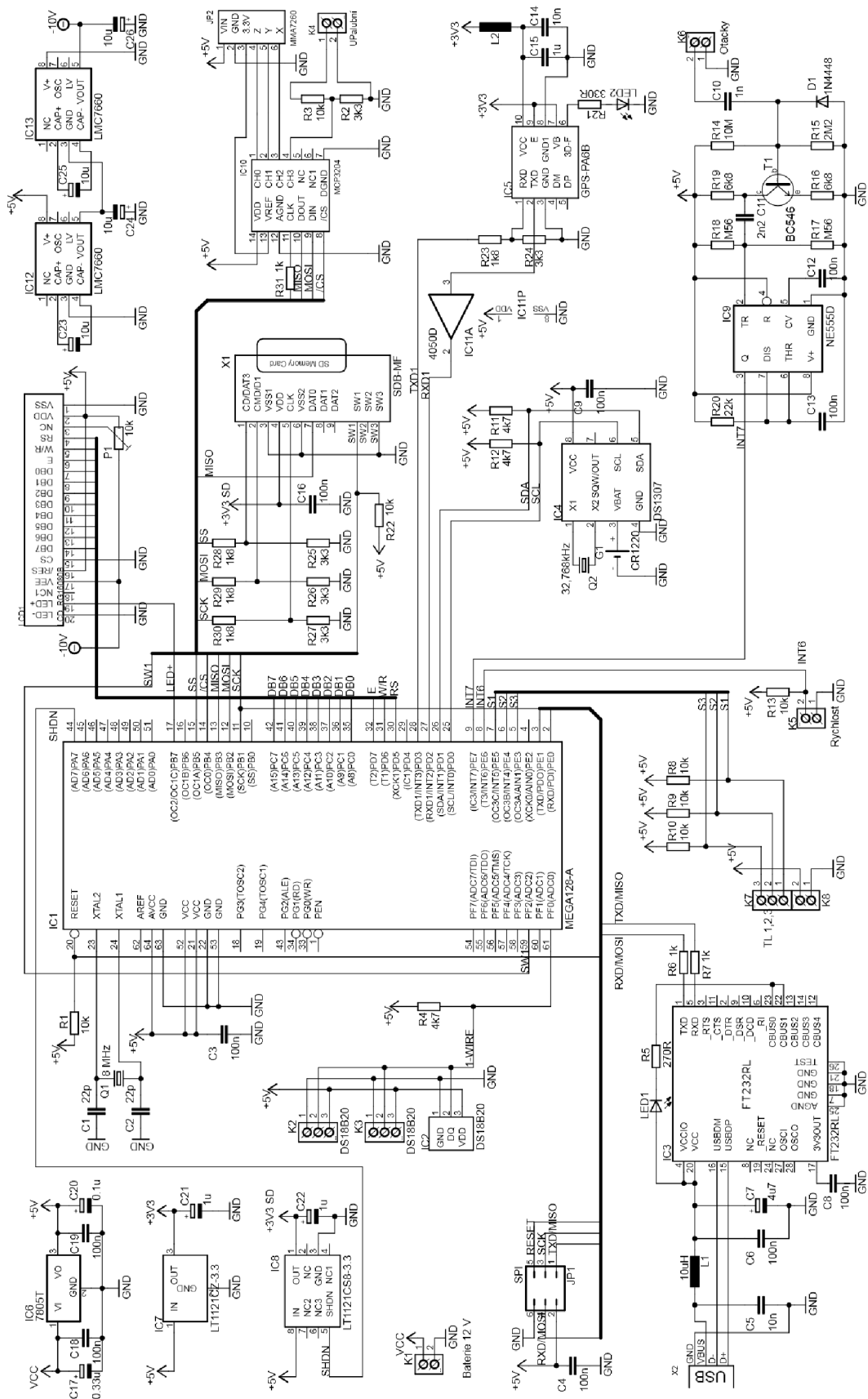
USART	Universal Synchronous Asynchronous Receiver Transmitter, univerzální synchroní a asynchroní seriové rozhraní
$U_2$	výstupní napětí děliče
$U_1$	vstupní napětí děliče
$U_{CC}$	napájecí napětí.
$U_{IN}$	velikost vstupního napětí
$U_{OC2}$	výstupní napětí na pinu OC2
$U_{REF}$	velikost referenčního napětí
$\Delta v$	rozdíl nadmořské výšky
USB	Universal Serial Bus, univerzální sériová sběrnice
VCP	Virtual COM Port, virtuální port COM
%	procento stoupání
$\alpha$	úhel náklonu motocyklu
$\beta$	stoupání ve stupních

## Seznam příloh

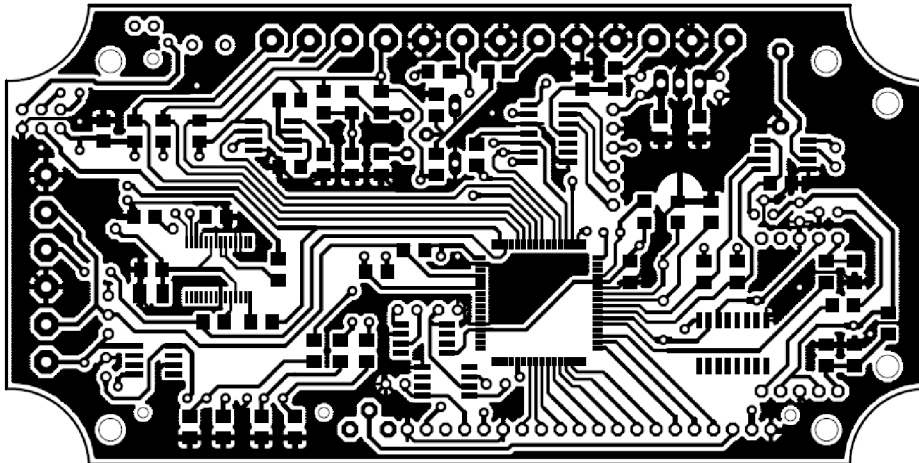
<b>A</b>	<b>Návrh zařízení .....</b>	<b>84</b>
A.1	Obvodové zapojení – palubní jednotka.....	84
A.2	Deska plošného spoje – bottom (strana spojů).....	85
A.3	Deska plošného spoje – top (strana součástek).....	85
A.4	Rozmístění součástek – top (strana součástek).....	85
A.5	Obvodové zapojení – panel s tlačítky .....	86
A.6	Deska plošného spoje – bottom (strana spojů).....	86
A.7	Deska plošného spoje – top (strana součástek).....	86
A.8	Přístrojová krabička – spodní strana .....	87
A.9	Přístrojová krabička – boční strana.....	87
<b>B</b>	<b>Seznam součástek .....</b>	<b>88</b>
B.1	Palubní jednotka.....	88
B.2	Panel s tlačítky .....	90

# A Návrh zařízení

## A.1 Obvodové zapojení – palubní jednotka

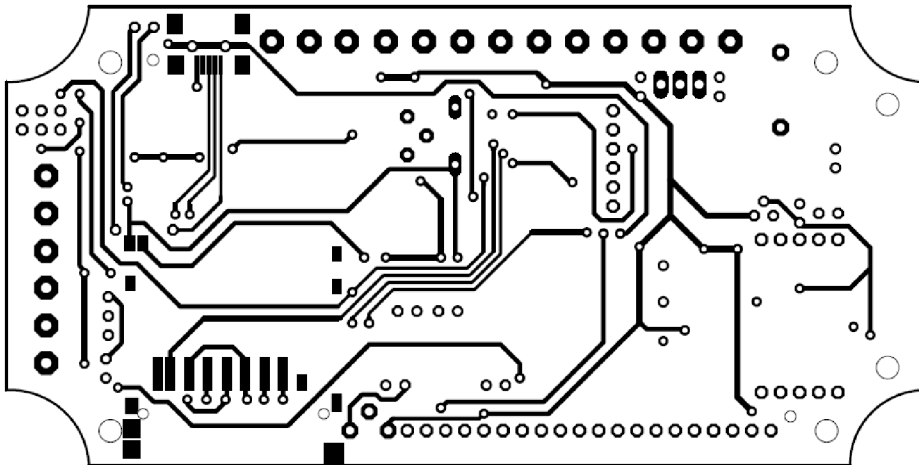


## A.2 Deska plošného spoje – bottom (strana spojů)



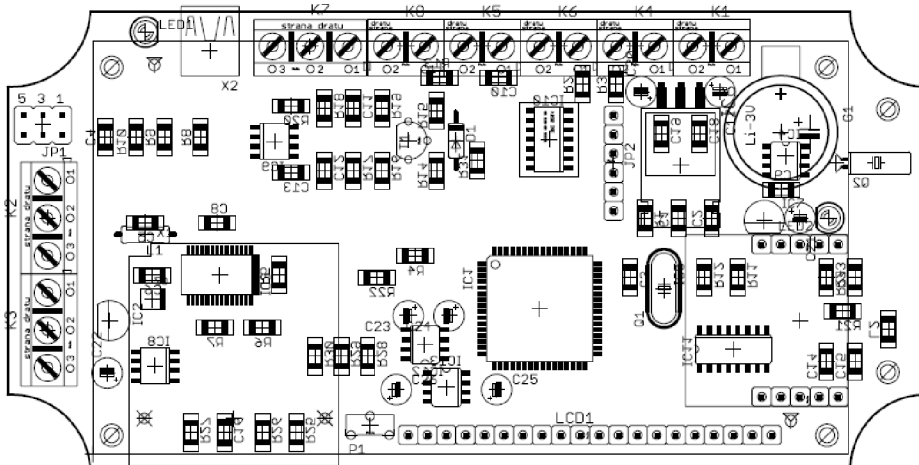
Rozměr desky 122x62 [mm], měřítko M1:1

## A.3 Deska plošného spoje – top (strana součástek)



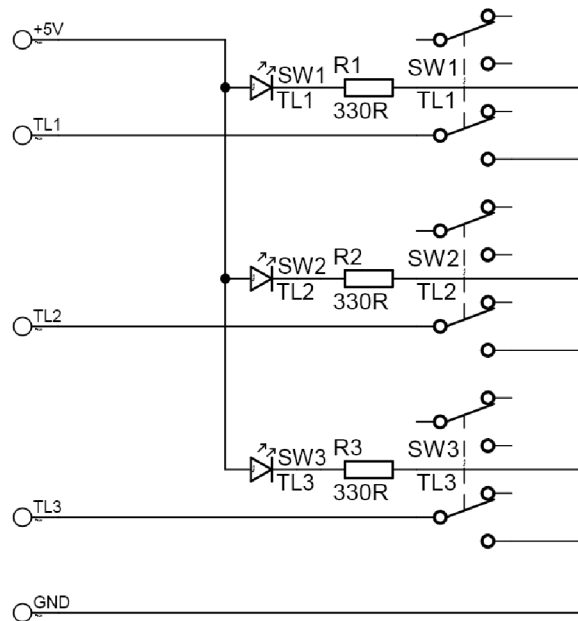
Rozměr desky 122x62 [mm], měřítko M1:1

## A.4 Rozmístění součástek – top (strana součástek)



Rozměr desky 122x62 [mm], měřítko M1:1

## A.5 Obvodové zapojení – panel s tlačítky

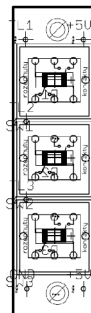


## A.6 Deska plošného spoje – bottom (strana spojů)



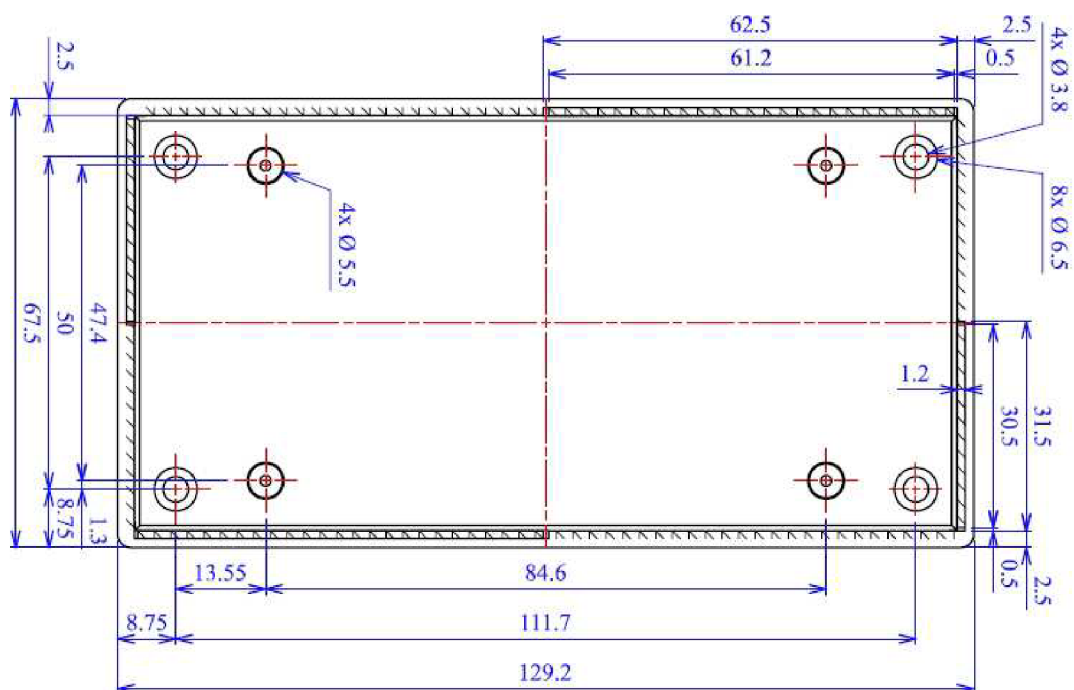
Rozměr desky 41x11 [mm], měřítko M1:1

## A.7 Deska plošného spoje – top (strana součástek)

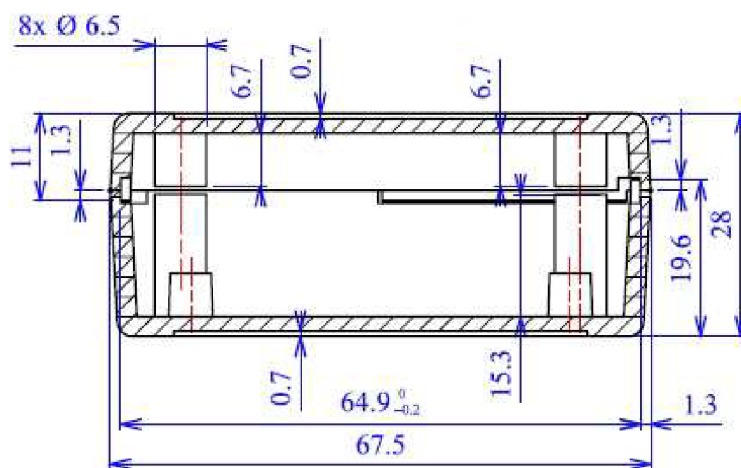


Rozměr desky 41x11 [mm], měřítko M1:1

## A.8 Přístrojová krabička – spodní strana



## A.9 Přístrojová krabička – boční strana



## B Seznam součástek

### B.1 Palubní jednotka

Označení	Typ	Pouzdro	Popis
C1	CK1206 22P/50V NPO	SMD 1206	Keramický kondenzátor 22 pF
C2	CK1206 22P/50V NPO	SMD 1206	Keramický kondenzátor 22 pF
C3	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C4	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C5	CK1206 10N/50V X7R	SMD 1206	Keramický kondenzátor 10 nF
C6	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C7	CTS 4M7/16V A 20%	SMD A	Tantalový kondenzátor 4,7 $\mu$ F
C8	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C9	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C10	CK1206 1N/50V X7R	SMD 1206	Keramický kondenzátor 1 nF
C11	CK1206 2N2/50V X7R	SMD 1206	Keramický kondenzátor 2,2 nF
C12	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C13	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátorv 100 nF
C14	CK1206 10N/50V X7R	SMD 1206	Keramický kondenzátor 10 nF
C15	CK1206 1M/50V X7R	SMD 1206	Keramický kondenzátor 1 $\mu$ F
C16	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátor 100 nF
C17	CE 0,33M/63V	4x7 mm	Elektrolytický kondenzátor 0,33 $\mu$ F
C18	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátor 100 nF
C19	CK1206 100N/50V X7R	SMD 1206	Keramický kondenzátor 100 nF
C20	CE 0,1M/63V	4x7 mm	Elektrolytický kondenzátor 0,1 $\mu$ F
C21	CE 1M/63V	4x7 mm	Elektrolytický kondenzátor 1 $\mu$ F
C22	CE 1M/63V	4x7 mm	Elektrolytický kondenzátor 1 $\mu$ F
C23	ELRS 10M/35V	5x7 mm	Elektrolytický kondenzátor 10 $\mu$ F
C24	ELRS 10M/35V	5x7 mm	Elektrolytický kondenzátor 10 $\mu$ F
C25	ELRS 10M/35V	5x7 mm	Elektrolytický kondenzátor 10 $\mu$ F
C26	ELRS 10M/35V	5x7 mm	Elektrolytický kondenzátor 10 $\mu$ F
D1	1N5817	DO15	Usměrňovací Schottkyho dioda
G1	CR1220	12,5x2 mm	Knoflíková lithiová baterie
IC1	ATmega128	TQFP64	Mikrokontrolér
IC2	DS18B20	TO-92	Digitální teploměr
IC3	FT232RL	SSOP	Převodník USB na UART
IC4	DS1307	SO	Obvod hodin reálného času
IC5	FGPMMOPA6B	2,54 mm	GPS modul
IC6	7805 STM	TO220	Stabilizátor napětí
IC7	LT1121CZ-3.3	TO-92	Stabilizátor napětí
IC8	LT1121CZ-3.3	SO-08	Stabilizátor napětí
IC9	NE555D	SO-08	Tvarovací obvod
IC10	MCP3204	SO-14	A/D převodník
IC11	HEF4050B	SO-16	Budič
IC12	LMC7660	SO-08	DC/DC měnič
IC13	LMC7660	SO-08	DC/DC měnič
JP1	S2G10	2,54 mm	Konektorový kolík lámací
JP2	S1G20	2,54 mm	Konektorový kolík lámací
K1	ARK500/2	2,54 mm	Šroubovací svorkovnice
K2	ARK500/3	2,54 mm	Šroubovací svorkovnice
K3	ARK500/3	2,54 mm	Šroubovací svorkovnice
K4	ARK500/2	2,54 mm	Šroubovací svorkovnice



K5	ARK500/2	2,54 mm	Šroubovací svorkovnice
K6	ARK500/2	2,54 mm	Šroubovací svorkovnice
K7	ARK500/2	2,54 mm	Šroubovací svorkovnice
K8	ARK500/2	2,54 mm	Šroubovací svorkovnice
L1	TLEC24-100K	3x6 mm	Tlumivka 10 $\mu$ H
L3	BLM31AJ601SN1L	SMD 1206	Feritová perlička
LCD	RG16080B	100x55 mm	LCD displej
LED1	LED 3mm 2MA/Y	3 mm	LED 3mm 2mA žlutá
LED2	LED 3mm 2MA/Y	3 mm	LED 3mm 2mA žlutá
Q1	QM 8MHz	HC49U-S	Krystal 8MHz
Q2	Q 37,768 kHz	3x8mm	Krystal hodinový
P1	PT6HK010	5x2,5 mm	Rezistorový trimr stojatý
R1	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R2	R1206 3K3 1%	SMD 1206	Rezistor 3,3 k $\Omega$
R3	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R4	R1206 4K7 1%	SMD 1206	Rezistor 4,7 k $\Omega$
R5	R1206 270R 1%	SMD 1206	Rezistor 270 $\Omega$
R6	R1206 1K 1%	SMD 1206	Rezistor 1 k $\Omega$
R7	R1206 1K 1%	SMD 1206	Rezistor 1 k $\Omega$
R8	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R9	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R10	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R11	R1206 4K7 1%	SMD 1206	Rezistor 4,7 k $\Omega$
R12	R1206 4K7 1%	SMD 1206	Rezistor 4,7 k $\Omega$
R13	R1206 10K 1%	SMD 1206	Rezistor 10 k $\Omega$
R14	R1206 10M 1%	SMD 1206	Rezistor 10 M $\Omega$
R15	R1206 2M2 1%	SMD 1206	Rezistor 2,2 M $\Omega$
R16	R1206 6k8 1%	SMD 1206	Rezistor 6,8 k $\Omega$
R18	R1206 M56 1%	SMD 1206	Rezistor 560 k $\Omega$
R19	R1206 6k8 1%	SMD 1206	Rezistor 6,8 k $\Omega$
R20	R1206 22k 1%	SMD 1206	Rezistor 22 k $\Omega$
R21	R1206 330R 1%	SMD 1206	Rezistor 330 $\Omega$
R22	R1206 10k 1%	SMD 1206	Rezistor 10 k $\Omega$
R23	R1206 1k8 1%	SMD 1206	Rezistor 1,8 k $\Omega$
R24	R1206 3k3 1%	SMD 1206	Rezistor 3,3 k $\Omega$
R25	R1206 3k3 1%	SMD 1206	Rezistor 3,3 k $\Omega$
R26	R1206 3k3 1%	SMD 1206	Rezistor 3,3 k $\Omega$
R27	R1206 3k3 1%	SMD 1206	Rezistor 3,3 k $\Omega$
R28	R1206 1k8 1%	SMD 1206	Rezistor 1,8 k $\Omega$
R29	R1206 1k8 1%	SMD 1206	Rezistor 1,8 k $\Omega$
R30	R1206 1k8 1%	SMD 1206	Rezistor 1,8 k $\Omega$
R32	R1206 1k 1%	SMD 1206	Rezistor 1 k $\Omega$
T1	BC546	TO-92	Bipolární NPN tranzistor
X1	SD karta 2 GB	SD	SD karta
X2	USB-MINI B F SMD	SMD	USB konektor

## B.2 Panel s tlačítky

Označení	Typ	Pouzdro	Popis
R1	R1206 330R 1%	SMD 1206	Rezistor 330 $\Omega$
R2	R1206 330R 1%	SMD 1206	Rezistor 330 $\Omega$
R3	R1206 330R 1%	SMD 1206	Rezistor 330 $\Omega$
TL1	PB61302BL	-	Mikrospínač bez aretace
TL2	PB61302BL	-	Mikrospínač bez aretace
TL3	PB61302BL	-	Mikrospínač bez aretace