UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Zařízení pro vyhodnocení stylu jízdy řidiče automobilu

Michal Absolon

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2012/2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Michal Absolon**
Osobní číslo:	**I09455**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Řízení procesů**
Název tématu:	**Zařízení pro vyhodnocení stylu jízdy řidiče automobilu**
Zadávající katedra:	**Katedra řízení procesů**

Zásady pro vypracování:

Cílem práce je návrh a konstrukce jednotky snímající trojosé změny rychlosti vozidla pomocí akcelerometru. Jednotka (založená na architektuře ATMega) bude osazena pamětí pro dlouhodobé uložení snímaných dat (data budou dále vyhodnocována zejména offline na osobním počítači) a diodami pro okamžitou indikaci změn.
Práce bude obsahovat teoretickou rešerši možností využití informací z trojosého akcelerometru pro posouzení stylu jízdy řidiče.
Praktická část práce bude sestávat z návrhu konkrétní jednotky a také ze série testování stylu jízdy skupiny řidičů. Vyhodnocení stylu jízdy pomocí jednotky bude porovnáno s průměrnou spotřebou zjištěnou na palubním počítači automobilu použitého pro testování.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

RIPKA, P.; TIPEK, A. Master Book on Sensors. Praha : BEN, 2003. 1020 s. ISBN 80-7300-129-2.
NEUBAUER, J.; SEDLAČÍK, M.; KŘÍŽ, O. Základy statistiky. Praha : Grada, 2012. 240 s. ISBN 978-80-247-4273-1.

Vedoucí bakalářské práce: **Ing. Petr Doležel, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **21. prosince 2012**
Termín odevzdání bakalářské práce: **10. května 2013**

Univerzita Pardubice
Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 08. 2013 Michal Absolon

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce, Ing. Petru Doleželovi, který mi poskytl cenné rady a doporučení při zpracování mé bakalářské práce. Děkuji též svoji rodině a přátelům za podporu po dobu mého studia.

Anotace

Tato bakalářská práce se zabývá návrhem jednotky pro vyhodnocování stylu jízdy řidiče pomocí akcelerometru. Teoretická část je věnována akcelerometrům a využití informace z akcelerometru pro posouzení stylu jízdy řidiče. Praktická část obsahuje návrh a vlastní realizaci zařízení a také sérii testování s vyhodnocením stylu jízdy.

Klíčová slova

Akcelerometr, mikrokontrolér

Title

Device for Driving Style Optimality Level Evaluation

Annotation

This bachelor thesis describes the design of the unit for evaluating the driving style using the accelerometer. The theoretical part describes accelerometers and the possibility of using the information from the accelerometer to evaluate driving style. The practical part contains the design and realization of the equipment and also a series of tests with evaluation of driving style.

Keywords

Accelerometer, microcontroller

Obsah

Seznam zkratek ... **8**

Seznam obrázků ... **9**

Seznam tabulek ... **10**

Úvod ... **11**

1 Teoretická část ... **12**

1.1 Akcelerometry ... 12

1.1.1 Typy akcelerometrů ... 12

1.2 Styl jízdy a jeho vliv na spotřebu ... 14

1.3 Teoretické hodnoty zrychlení ... 14

1.4 Sběrnice I^2C ... 17

1.5 Jednotka USART ... 18

2 Praktická část ... **20**

2.1 Použité komponenty ... 20

2.1.1 Akcelerometr MMA7455L ... 20

2.1.2 Mikroprocesor ATmega32 ... 23

2.1.3 Paměť EEPROM ... 24

2.1.4 Displej ... 24

2.2 Návrh schéma zapojení ... 25

2.3 Návrh plošného spoje ... 27

2.4 Programová část ... 29

2.4.1 Hlavní modul ... 29

2.4.2 Modul práce s obvodem akcelerometru ... 33

2.4.3 Modul komunikace I^2C ... 35

2.4.4 Modul sériové komunikace ... 36

2.4.5 Modul ovládání paměti EEPROM ... 37

2.4.6 Modul ovládání displeje LCD ... 39

2.5 Testovací jízdy ... 41

3 Závěr ... **46**

Literatura ... **47**

Příloha A – Tabulka registrů akcelerometru MMA7455L

Příloha B – Schéma zařízení

Příloha C – Zařízení pro vyhodnocování stylu jízdy

Seznam zkratek a symbolů

I^2C	Inter integrated circuit
SCL	Serial clock
SDA	Serial data
SPI	Serial Peripheral Interface
USART	Universal Synchronous Asynchronous Receiver Transmitter
a	zrychlení
t	čas

Seznam obrázků

Obrázek 1 – Princip piezoelektrického akcelerometru 13
Obrázek 2 – Princip kapacitního akcelerometru 13
Obrázek 3 – Zatáčka pro určení bočního zrychlení 16
Obrázek 4 – Připojování zařízení na I2C sběrnici 17
Obrázek 5 – START a STOP stavy 18
Obrázek 6 – Modul MMA7455 20
Obrázek 7 – Akcelerometr MMA7455L 21
Obrázek 8 – Rozložení vývodů pouzdra mikrokontroléru ATmega32 24
Obrázek 9 – Plošný spoj 28
Obrázek 10 – Měření zrychlení plynulé jízdy v ose X 42
Obrázek 11 – Měření zrychlení plynulé jízdy v ose Y 42
Obrázek 12 – Měření zrychlení normální jízdy v ose X 43
Obrázek 13 – Měření zrychlení normální jízdy v ose Y 43
Obrázek 14 – Měření zrychlení agresivní jízdy v ose X 44
Obrázek 15 – Měření zrychlení agresivní jízdy v ose Y 44

Seznam tabulek

Tabulka 1 – Průjezd zatáčky 16
Tabulka 2 – Styly jízdy 17
Tabulka 3 – Datový registr UDR 18
Tabulka 4 – Registr USCRA 19
Tabulka 5 – Registr UCSRB 19
Tabulka 6 – Registr UCSRC 19
Tabulka 7 – Registr UBRRH, UBRRL 20
Tabulka 8 – Popis pinů modulu MMA7455 21
Tabulka 9 – Stavový registr na adrese $09 21
Tabulka 10 – Registr I2CAD na adrese $0D 22
Tabulka 11 – Registry XOUT8, YOUT8, ZOUT8 na adresách $06, $07 a $08 22
Tabulka 12 – Registr MCTL na adrese $16 22
Tabulka 13 – Nastavení rozsahu zrychlení 22
Tabulka 14 – Nastavení typu měření 23
Tabulka 15 – Offsetové registry na adresách $10 až $15 23
Tabulka 16 – Vývody displeje 25
Tabulka 17 – Seznam globálních proměnných a jejich význam 30
Tabulka 18 – Vyhodnocení jízdy 41

Úvod

Cílem této práce je navrhnout a sestavit zařízení, které bude vyhodnocovat styl jízdy řidiče. Vyhodnocování stylu jízdy řidiče bude probíhat na základě hodnot zjištěných z akcelerometru. Výsledná data bude zpracovávat a vyhodnocovat mikroprocesor ATmega32. Indikace stylu jízdy řidiče bude zajištěna pomocí LED diod. Zařízení bude dále obsahovat paměť EEPROM, pro možnost uložení dat získaných z akcelerometru. Tato uložená data je možné následně přenést do počítače pomocí jednotky USART. V případě potřeby bude zařízení možné rozšířit i o dvouřádkový LCD displej, pomocí kterého je možné zobrazovat různé podrobnější údaje.

První kapitola obsahuje teoretickou část a zabývá se zejména akcelerometry, kde je vysvětlen princip činnosti akcelerometru a popsány různé typy akcelerometrů. Dále zde je popsán způsob využití dat získaných z akcelerometru pro posouzení stylu jízdy řidiče. V další části následují popisy komunikace pomocí I^2C sběrnice práce použité jednak pro komunikaci akcelerometru s mikroprocesorem, ale také pro zápis do paměti EEPROM a dále popis jednotky USART použité pro komunikaci s počítačem.

Druhá kapitola obsahuje praktickou část, ve které jsou nejprve popsány základní komponenty použité pro konstrukci zařízení. Je zde tedy popsán použitý akcelerometr včetně jeho registrů a možnosti nastavení a dále jsou zde popsány základní vlastnosti mikroprocesoru ATmega32. V této kapitole jsou rovněž uvedeny praktické výsledky z testování hotového zařízení.

1 Teoretická část

1.1 Akcelerometry

Akcelerometr se řadí mezi senzory, které se využívají především pro měření zrychlení. Zrychlení působící na akcelerometr můžeme rozdělit na statické a dynamické. Statické zrychlení označuje sílu způsobenou zemskou gravitací a dynamické zrychlení sílu vzniklou změnou rychlosti pohybujícího se předmětu. Statické zrychlení je již ze svého principu neustále přítomné a je tedy ho nutné při měření dynamického zrychlení ve výsledcích odstranit filtrací [1].

Akcelerometry jsou vhodné nejen pro měření zrychlení a v dnešní době mají velké možnosti využití. Používají se pro měření vibrací, otřesů, pádu, natočení nebo pro určování polohy tělesa a jeho náklonu. Využití najdou i v automobilovém průmyslu v bezpečnostních systémech, kde dávají povel k aktivaci airbagů. Důležitou roli hrají v dnešní době v telefonech, např. pro překlápění obrazu na displeji při natočení telefonu nebo u her, kdy můžeme ovládat telefon pomocí švihů či naklonění.

1.1.1 Typy akcelerometrů

Mezi nejpoužívanější akcelerometry patří piezorezistivní akcelerometry, piezoelektrické akcelerometry, tepelné akcelerometry a akcelerometry s proměnnou kapacitou.

- **Piezorezistivní akcelerometry**

  Základ těchto akcelerometrů tvoří piezorezistivní materiál. Využívá se toho, že při působení síly na tento materiál dochází ke změně jeho odporu. Sílu působící na materiál způsobuje seismická hmota při zrychlení. Změna zrychlení se tedy projeví změnou odporu. Měření odporu piezorezistivního materiálu probíhá prostřednictvím základního nebo polovičního Wheatstonova můstku. Piezorezistivní akcelerometry mohou měřit stálou akceleraci, tedy frekvenci změn od 0 Hz [2].

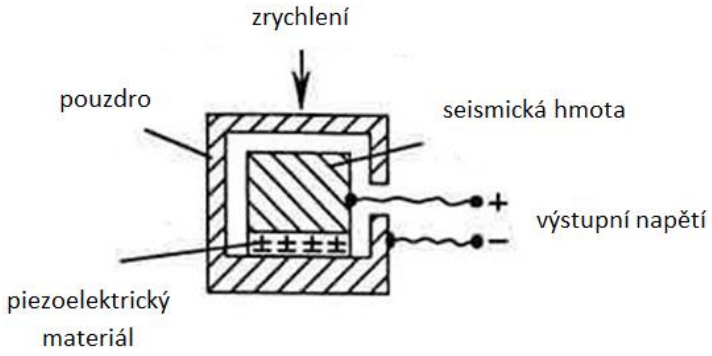
- **Piezoelektrické akcelerometry**

  Využívají piezoelektrický krystal generující náboj, který se konvertuje na napěťový výstup s nízkou impedancí. Náboj je úměrný síle působící na piezoelektrický krystal, která je vyvinuta při akceleraci. Základní strukturu těchto akcelerometrů tvoří dvě hlavní části. První část tvoří piezoelektrický materiál, který představuje snímač pro měření zrychlení. Druhou část tvoří seismická hmota, která převádí hodnotu zrychlení na sílu podle Newtonova zákona:

$$F = m \cdot a \tag{1.1}$$

  kde F – síla, N
  a – zrychlení, $m \cdot s^{-2}$
  m – hmotnost seismické hmoty, kg

Nevýhodou těchto senzorů je, že nemohou být použity pro měření frekvencí nižších než 0,1 Hz, tedy měření konstantního zrychlení [3].



Obrázek 1 – Princip piezoelektrického akcelerometru

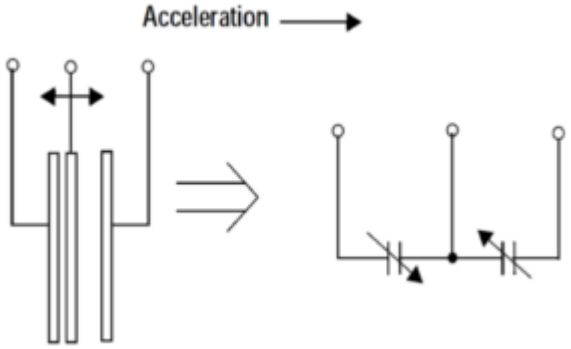
- **Kapacitní akcelerometry**

U kapacitních akcelerometrů se využívá nelineární závislosti kapacity na vzdálenosti elektrod kondenzátoru podle vzorce:

$$C = \varepsilon_0 \cdot \varepsilon_r \cdot \frac{S}{d} \tag{1.2}$$

kde C – kapacita, F
d – vzdálenost elektrod, m
S – plocha elektrod, m^2
ε_0 – permitivita vakua $8{,}854 \cdot 10^{-12}$, $F \cdot m^{-1}$
ε_r – permitivita prostředí, nemá rozměr

Na obrázku 2 je uveden příklad kapacitního akcelerometru s tříelektrodovým vzduchovým kondenzátorem s jednou pohyblivou elektrodou umístěnou mezi dvěma pevnými elektrodami. Při působícím zrychlení se prostřední elektroda pohybuje a tím se mění kapacita [4].



Obrázek 2 – Princip kapacitního akcelerometru

- **Tepelné akcelerometry**

  Tyto akcelerometry využívají změny přenosu tepla klidného a proudícího vzduchu působícím zrychlením. Zabudované topné tělísko zahřívá okolní vzduch ve vzduchové komoře na konstantní teplotu. Rozložení teploty v závislosti na vzdálenosti od topného tělíska je měřené teplotními snímači. Ty jsou realizovány soustavou termočlánků umístěných v pravidelných rozestupech. Nevýhoda těchto akcelerometrů je veliká citlivost na okolní změnu teploty a možnost měření pouze ve 2D, jelikož 3D měření je velmi špatně realizovatelné [5].

1.2 Styl jízdy a jeho vliv na spotřebu

Spotřeba paliva automobilu hodně závisí na stylu jízdy. Je velký rozdíl, pokud se jezdí stylem „brzda plyn“, nebo zda se předvídá situace na silnici, pozoruje se situace daleko před vozidlem a včas se na tuto situaci reaguje. Nebrzdí se tedy intenzivně na poslední chvíli a následně se ihned prudce akceleruje. Ideální je jezdit takzvaně s lehkou nohou, nešlapat prudce a intenzivně na plynový pedál, ale spíš pozvolna zrychlovat. Při zpomalování podřazovat namísto brzdění a nechat spíše brzdit motor. Dále udržovat motor v optimálních otáčkách. Samozřejmě v rámci udržení nízké spotřeby se nesmí jezdit přehnaně rychle.

Při úsporné plynulé jízdě můžeme oproti rychlé jízdě, kdy prudce zrychlujeme a zpomalujeme ušetřit kolem třetiny paliva a přitom nemusí výrazně klesnout průměrná rychlost vozidla.

Pokud je vozidlo vybavené palubním počítačem, který ukazuje okamžitou spotřebu, je možné se orientovat podle tohoto ukazatele. Sledování rychle se měnících údajů je však náročné a může odvádět pozornost od vlastního řízení vozidla a působit tak vlastně proti optimální spotřebě.

S jednoduchým systémem přišla na trh automobilka Honda, která do svých vozidel implementovala jednoduchou indikaci ekonomické jízdy a to pomocí podsvícení palubní desky. Palubní deska je podsvícená třemi různými barvami podle ekonomiky jízdy. Modrá barva znamená, že ekonomika jízdy je standardní, zelená barva znamená, že jízda je ekologická a ekonomická a naopak červená barva znamená neekonomický styl jízdy. Automobilka uvádí, že zavedením této jednoduché indikace dojde až k 12% úspory paliva[6].

Cílem práce byl vývoj podobného zařízení, které by jednoduchým způsobem indikovalo řidiči, jakým stylem jede. Protože styl jízdy je dán zejména způsobem zrychlování nebo zpomalování, či rychlými změny směru jízdy, byla pro účely zařízení využita data získaná z akcelerometru, který je umístěn ve vozidle tak, že v jednom směru snímá akceleraci a brzdění a v druhém směru, kolmém na směr jízdy pak prudké změny směru této jízdy.

1.3 Teoretické hodnoty zrychlení

Při zjištění teoretických hodnot zrychlení při rozjezdu vozidla se bude vycházet z údajů o zrychlení z 0 na 100 $km \cdot hod^{-1}$, které výrobci u vozidel udávají. Maximální zrychlení standardního vozidla se pohybuje kolem zrychlení z 0 do 100 $km \cdot hod^{-1}$ za 10 vteřin. Pro výpočet zrychlení platí vzorec:

$$a = \frac{v - v_0}{t} = \frac{(100/3{,}6) - 0}{10} = 2{,}78 \text{ m·s}^{-2} \tag{1.3}$$

kde a – zrychlení, m·s^{-2}
v – dosažená rychlost, m·s^{-1}
v_0 – počáteční rychlost, m·s^{-1}
t – čas, s

Pro teoretické zjištění hodnoty zpomalení při bržděnísе bude vycházet z údajů o brzdných drahách při intenzivním brzdění. Na suché vozovce zabrzdí osobní automobil jedoucí rychlostí 60 km·hod^{-1} na přibližně 20 metrech. Čas brzdění při rovnoměrně zpomaleném pohybu se spočítá podle vzorce:

$$t = \frac{2 \cdot s}{v_0 + v} = \frac{2 \cdot 20}{16{,}67} = 2{,}4 \text{ s} \tag{1.4}$$

kde t – čas, s
v – rychlost, m·s^{-1}
v_0 – počáteční rychlost, m·s^{-1}
s – dráha, m

Výsledná hodnota zpomalení vypočítaná podle vzorce (1.3) je 6,94 m·s^{-2}.

Vzhledem k tomu, že zpomalení vlivem brzdění má přibližně dvojnásobné hodnoty zrychlení, budou se hodnoty směru brzdění dělit dvěma.

Pro určení bočního zrychlení byl proveden jednoduchý experiment průjezdem zatáčky při různých rychlostech. Byla vybrána zatáčka mezi Vysokým Mýtem a Dvořiskem zobrazená na obrázku 3. Pomocí měření na mapě byla změřena délka tětivy a výška oblouku od tětivy. Délka tětivy byla naměřena 150 metrů a výška 12 metrů. Poloměr kružnice byl vypočítán podle vzorce:

$$r = \frac{d^2 + \left(\frac{l}{2}\right)^2}{2 \cdot d} = \frac{12^2 + 75^2}{2 \cdot 12} = 240 \text{ m} \tag{1.5}$$

kde r – poloměr oblouku, m
d – výška tětivy oblouku, m
l – délka tětivy oblouku, m

Pro výpočet dostředivého zrychlení použijeme vzorec:

$$a_{\text{d}} = \frac{v^2}{r} \tag{1.6}$$

kde a_d – dostředivé zrychlení, $m \cdot s^{-1}$
v – okamžitá rychlost jízdy, $m \cdot s^{-1}$
r – poloměr křivosti zatáčky, m

Obrázek 3 – Zatáčka pro určení bočního zrychlení

Tato zatáčka byla projeta třemi různými styly jízdy s následujícími výsledky uvedenými v tabulce 1.

Tabulka 1 – Průjezd zatáčky

Styl jízdy	Rychlost jízdy	Dostředivé zrychlení
Klidný průjezd	60 $km \cdot h^{-1}$	1,16 $m \cdot s^{-2}$
Standartní průjezd	90 $km \cdot h^{-1}$	2,6 $m \cdot s^{-2}$
Ostrý průjezd	130 $km \cdot h^{-1}$	5,43 $m \cdot s^{-2}$

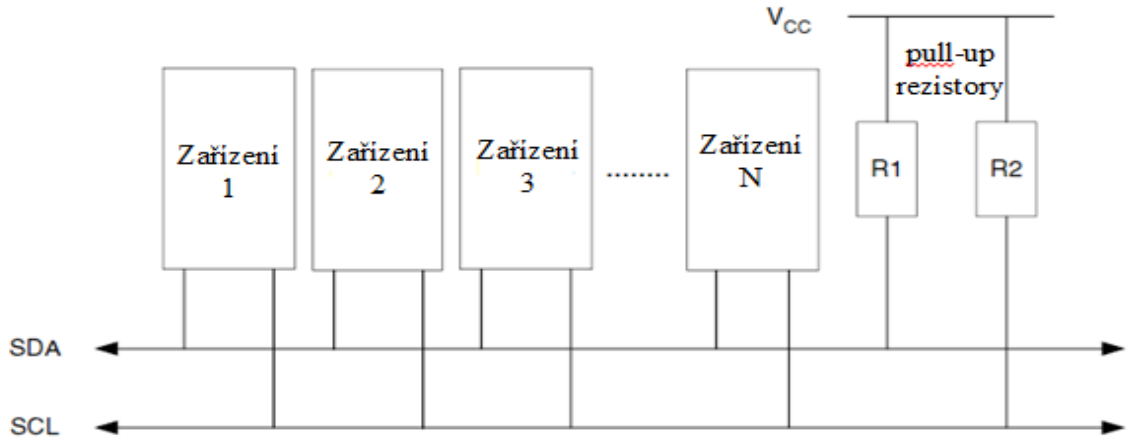
Pro vyhodnocování byly zvoleny tři úrovně stylu jízdy, které jsou uvedeny v tabulce 2. K jednotlivým stylům jízdy byl na základě výše uvedené teorie a experimentu přiřazen rozsah hodnot zrychlení a zpomalení.

Tabulka 2 – Styly jízdy

Styl jízdy	Rozsah zpomalení	Rozsah zrychlení	Rozsah bočního zrychlení
Plynulý	0 – 0,3 g	0 – 0,15 g	0 – 0,15 g
Normální	0,3 – 0,6 g	0,15 – 0,3 g	0,15 – 0,3 g
Agresivní	> 0,6 g	> 0,3 g	> 0,3 g

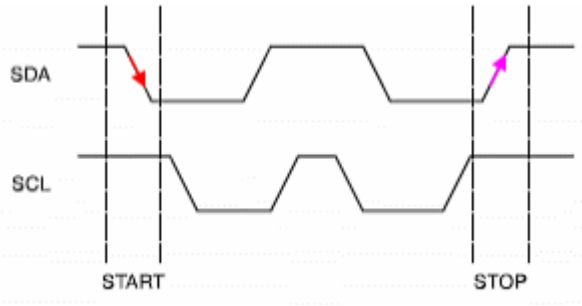
1.4 Sběrnice I²C

I²C označuje sériovou sběrnici umožňující komunikaci mezi mikrokontrolérem a libovolným zařízením. Komunikace probíhá za pomoci dvou linek sběrnice. První linka je určena pro hodinové pulsy (SCL) a druhá linka pro data (SDA). Obě linky jsou připojeny na napájecí napětí přes pull-up rezistory, které linky „vytáhnou“ směrem k logické 1 v případě nečinnosti na sběrnici. Každé zařízení připojené na sběrnici má individuální adresu. Adresa zařízení je 7bitová, což umožnuje připojit na sběrnici až 128 různých zařízení. Mikrokontrolér může pracovat jako master nebo slave. Master představuje zařízení, které začíná a ukončuje přenos a také generuje hodiny na lince SCL. Slave představuje zařízení adresované masterem. Rozhraní dovoluje konfigurovat přenosovou rychlost až na 400 kHz.



Obrázek 4 – Připojování zařízení na I2C sběrnici

Každý datový bit přenášený po sběrnici je spojen s jedním hodinovým pulsem. Logická úroveň na datové lince musí být ustálena v okamžiku, kdy SLC = 1. Přenos dat je zahájen, když master vloží na sběrnici tzv. START stav. START stav odpovídá sestupné hraně SDA v okamžiku, kdy je SCL =1. Pro ukončení přenosu dat master vloží na sběrnici tzv. STOP stav, který odpovídá náběžné hraně SDA v okamžiku kdy SCL = 1.



Obrázek 5 – START a STOP stavy

Vysílání dat po I^2C sběrnici začíná START stavem, za kterým následuje adresní paket. Poté jsou vysílány datové pakety až po ukončení přenosu. Přenos dat se ukončí STOP stavem. Adresní paket je vysílán v délce 9 bitů a skládá se z 7bitové adresy zařízení, řídícího bitu R/W a potvrzovacího bitu ACK. Řídící bit určuje, zda se jedná o čtení nebo zápis. Když slave rozpozná, že je adresován, měl by stáhnout SDA linku v okamžiku potvrzovacího bitu ACK. Je-li adresovaný slave zaneprázdněn nebo když žádné zařízení neodpovídá, je linka SDA ponechána v logické 1. Datový paket se skládá rovněž z 9 bitů, kde prvních 8 bitů obsahuje data, za kterými je opět potvrzovací bit ACK, kdy je každý úspěšně přijatý bajt ze strany přijímače potvrzen stažením linky SDA v 9. cyklu hodin [7].

1.5 Jednotka USART

USART označuje zařízení pro sériovou komunikaci. Jedná se o univerzální synchronní a asynchronní sériový přijímač a vysílač, který je plně duplexní a může tedy současně vysílat i přijímat. Skládá se ze tří hlavních částí. Z generátoru hodin sloužící k synchronizaci vysílací a přijímací části. Dále z vysílače a přijímače obsahují bufery pro vysílání a příjem bajtů. Podporuje synchronní nebo asynchronní režim. Při synchronním režimu může pracovat jako Master nebo Slave. Podporuje rámce délky 5 až 9 bitů a 1 až 2 stop-bity. Umožnuje také detekci chyby znaku nebo rámce [7].

Nejdůležitější registry:

Tabulka 3 – Datový registr UDR

D7	D6	D5	D4	D3	D2	D1	D0	Bit
RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	funkce
TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0	funkce

Datový bufer přijímače a vysílače sdílí stejnou adresu označenou jako registr UDR. Do registru UDR lze zapisovat pouze tehdy, je-li nastaven bit UDRE v registru UCSRA.

Tabulka 4 – Registr USCRA

D7	D6	D5	D4	D3	D2	D1	D0	Bit
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	funkce

Registr USCRA obsahuje příznaky komunikace a chyby příjmu. Bit UDRE indikuje, že vysílací bufer (UDR) je připraven přijmout nová data. Pokud je tedy bit UDRE nastaven, je bufer prázdný a připravený pro zápis. Bit RXC indikuje, že příjem dat je dokončen a nastaví se, pokud jsou k dispozici nepřečtená data v přijímacím buferu (UDR). Naopak bit TXC je příznak odvysílání byte a nastaví se tedy v případě, když je aktuální rámec vyslán ven a vysílací bufer zůstane prázdný. Bit FE indikuje chybu rámce a nastaví se v případě, pokud nedošlo k příjmu platného stop-bitu.

Tabulka 5 – Registr UCSRB

D7	D6	D5	D4	D3	D2	D1	D0	Bit
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXT8	TXB8	funkce

V registru UCSRB jsou nejdůležitější bity RXEN pro povolení přijímače a TXEN pro povolení vysílače. Bity RXCIE, TXCIE a UDRIE slouží k povolení přerušení. RXCIE povoluje přerušení při příjmu, TXCIE při dokončení vysílání a UDRIE při vyprázdnění UDR.

Tabulka 6 – Registr UCSRC

D7	D6	D5	D4	D3	D2	D1	D0	Bit
URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	funkce

Registr UCSRC obsahuje bity pro volbu režimu jednotky USART. Pro zápis do registru musí být nastaven bit URSEL. Bit UMSEL rozhoduje o tom, zda bude zvolen asynchronní nebo synchronní režim. Nastavením tohoto bitu je zvolen synchronní režim. Dvojice bitů UPM1 a UPM0 nastavuje režim parity. USBS volí počet stop-bitů nastavených vysílačem. Bity UCSZ1 a USCZ0 nastavují počet datových bitů rámce. Poslední bit UCPOL má význam pouze pro synchronní režim. Pro asynchronní režim musí být tento bit vynulován.

Tabulka 7 – Registr UBRRH, UBRRL

D7	D6	D5	D4	D3	D2	D1	D0	Bit
URSEL	–	–	–	UBRR11	UBRR10	UBRR9	UBRR8	funkce
UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0	funkce

Pár registrů UBRRH a UBRRL slouží k nastavení přenosové rychlosti. Pro zápis do registru UBRRH musí být bit URSEL vynulován. Výsledná přenosová rychlost je dána vzorcem:

$$f_{BR} = \frac{f_0}{UBRR + 1} \tag{1.7}$$

kde f_{BR} – výsledný kmitočet
f_0 – hodinový signál mikrokontroléru
UBRR – 16bitový registr (UBRRL, UBRRH)

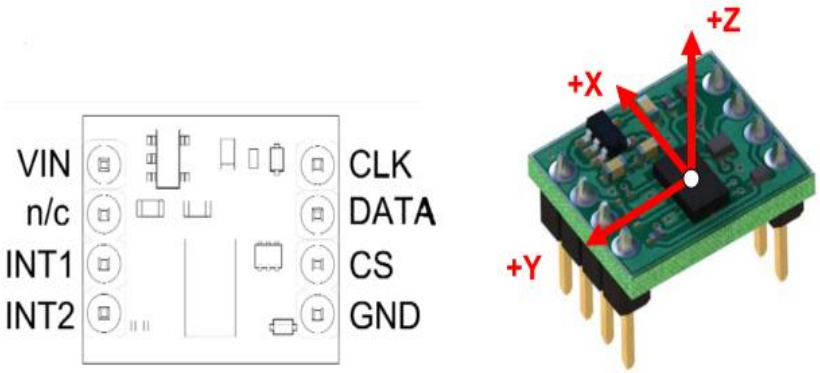
Výsledná přenosová rychlost je dále dělena 2, 8 nebo 16 podle zvoleného režimu. Pro normální asynchronní režim je výsledná rychlost dělena šestnácti.

2 Praktická část

2.1 Použité komponenty

Základní komponenty, ze kterých se skládá zařízení pro sledování stylu řidiče, jsou jednotlivě popsány v níže uvedených podkapitolách.

2.1.1 Akcelerometr MMA7455L



Obrázek 6 – Modul MMA7455

Pro tuto práci byl vybrán akcelerometr MMA7455L od firmy Freescale Semiconductor. Jedná se o kapacitní akcelerometr vyrobený technologií MEMS. Byl vybrán modul od firmy Parallax MMA7455 zobrazený na obrázku 6, který obsahuje již zabudovaný akcelerometr MMA7455L.

Tabulka 8 – Popis pinů modulu MMA7455

Číslo pinu	Pin	Definice
1	VIN	Napájení
2	n/c	Nepřipojen – rezervovaný pro budoucí použití
3	INT1	Přerušení 1
4	INT2	Přerušení 2
5	GND	Uzemnění
6	CS	Výběr komunikace – SPI (0), I^2C (1)
7	DATA	I^2C data (SDA) nebo SPI data (SDI/SDO)
8	CLK	I^2C hodiny (SCL) nebo SPI hodiny (SPC)

Základní vlastnosti akcelerometru MMA7455L jsou:

- Komunikace pomocí SPI nebo I^2C sběrnice
- Volba nastavení citlivosti 2g, 4g, 8g v 8bitovém režimu
- Možnost nastavení citlivosti 8g v 10bitovém režimu
- Napájecí napětí 2,4 – 3,6 V

Obrázek 7 – Akcelerometr MMA7455L

Přehled nejdůležitějších registrů:

Tabulka 9 – Stavový registr na adrese $09

D7	D6	D5	D4	D3	D2	D1	D0	bit
---	---	---	---	---	PERR	DOVR	DRDY	funkce

Ve stavovém registru hraje nejdůležitější roli bit DRDY, který nám oznamuje, zdali jsou naměřená data platná.

Tabulka 10 – Registr I2CAD na adrese $0D

D7	D6	D5	D4	D3	D2	D1	D0	bit
I2CDIS	DVAD[6]	DVAD[5]	DVAD[4]	DVAD[3]	DVAD[2]	DVAD[1]	DVAD[0]	funkce

V registru I2CAD je nejdůležitější bit I2CDIS, jehož nastavení na log. 1 zakáže I^2C sběrnici. Pro povolení I^2C a SPI jednotky je třeba tento bit vynulovat. Zbývajících 7 bitů představuje adresu zařízení při I^2C komunikaci.

Tabulka 11 – Registry XOUT8, YOUT8, ZOUT8 na adresách $06, $07 a $08

D7	D6	D5	D4	D3	D2	D1	D0	bit
XOUT[7]	XOUT[6]	XOUT[5]	XOUT[4]	XOUT[3]	XOUT[2]	XOUT[1]	XOUT[0]	funkce
YOUT[7]	YOUT[6]	YOUT[5]	YOUT[4]	YOUT[3]	YOUT[2]	YOUT[1]	YOUT[0]	funkce
ZOUT[7]	ZOUT[6]	ZOUT[5]	ZOUT[4]	ZOUT[3]	ZOUT[2]	ZOUT[1]	ZOUT[0]	funkce

Jednotlivé registry XOUT8, YOUT8 a ZOUT8 obsahují 8bitové hodnoty zrychlení měřené ve všech třech osách.

Tabulka 12 – Registr MCTL na adrese $16

D7	D6	D5	D4	D3	D2	D1	D0	bit
---	DRPD	SPI3W	STON	GLVL[1]	GLVL[0]	MODE[1]	MODE[0]	funkce

Nastavení rozsahu měření se provádí v registru MCTL (Mode Control Register) pomocí bitů GLVL[1] a GLVL[0]. Jednotlivé kombinace nastavení rozsahu jsou uvedeny v tabulce 13.

Tabulka 13 – Nastavení rozsahu zrychlení

GLVL[1]	GLVL[0]	rozsah
0	1	2g
1	0	4g
0	0	8g

Nastavení typu měření se provádí v registru MCTL (Mode Control Register) pomocí bitů MODE[1] a MODE[0]. Jednotlivé kombinace nastavení typu měření jsou uvedeny v tabulce 14.

Tabulka 14 – Nastavení typu měření

MODE[1]	MODE[0]	Typ měření
0	0	Standby mode
0	1	Measurement mode
1	0	Level Detection mode
1	1	Pulse Detection mode

Pro kalibraci akcelerometru jsou zde dvojice registrů pro každou osu, kam se zapisují 10bitové hodnoty. Tyto registry jsou uvedeny v tabule 15.

Tabulka 15 – Offsetové registry na adresách $10 až $15

Název registru	D7	D6	D5	D4	D3	D2	D1	D0
XOFFL	XOFF[7]	XOFF[6]	XOFF[5]	XOFF[4]	XOFF[3]	XOFF[2]	XOFF[1]	XOFF[0]
XOFFH	---	---	---	---	---	XOFF[10]	XOFF[9]	XOFF[8]
YOFFL	YOFF[7]	YOFF[6]	YOFF[5]	YOFF[4]	YOFF[3]	YOFF[2]	YOFF[1]	YOFF[0]
YOFFH	---	---	---	---	---	YOFF[10]	YOFF[9]	YOFF[8]
ZOFFL	ZOFF[7]	ZOFF[6]	ZOFF[5]	ZOFF[4]	ZOFF[3]	ZOFF[2]	ZOFF[1]	ZOFF[0]
ZOFFH	---	---	---	---	---	ZOFF[10]	ZOFF[9]	ZOFF[8]

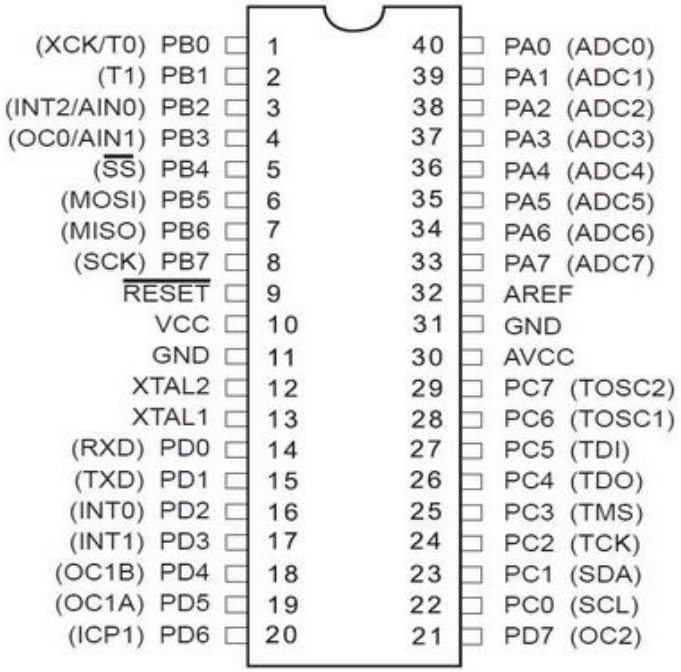
2.1.2 Mikroprocesor ATmega32

Jedná se o nezbytnou součást zařízení a jako mikroprocesor byl zvolen Atmega32. Tento mikroprocesor byl zvolen zejména díky praktickým zkušenostem, které jsem získal za dobu studia. ATmega32 je vybaven SPI a I^2C rozhraním, pomocí kterých je možné komunikovat s akcelerometrem a s pamětí. Dále obsahuje zabudovaný synchronní a asynchronní sériový kanál USART potřebný pro zasílání dat do počítače. Mikroprocesor je také vybaven 16bitovým čítačem/časovačem, potřebným pro vyvolání přerušení.

Základní vlastnosti mikrokontroléru ATmega32:

- Instrukční soubor obsahuje 131 instrukcí
- 32 registrů délky 8 bitů
- Čtyři 8bitové vstupně/výstupní porty
- Hodinový kmitočet 0 až 16 MHz
- Paměť Flash pro program s kapacitou 32 KB, počet přeprogramování je 10 000 cyklů
- Datová paměť SRAM kapacity 2 KB
- Datová paměť EEPROM kapacity 1024 B, počet přeprogramování je 100 000 cyklů

- Čtyři PWM kanály
- Jednotky USART, SPI, TWI (I2C)
- Dva 8bitové čítače/časovače, jeden 16bitový čítač/časovač



Obrázek 8 – Rozložení vývodů pouzdra mikrokontroléru ATmega32

2.1.3 Paměť EEPROM

Pro uložení dat z akcelerometru použijeme paměť typu EEPROM, která nepotřebuje pro udržení dat v paměti žádné napájení, což nám umožní přenést zařízení z vozidla k počítači a provést přenos dat uložených v paměti do počítače k dalšímu zpracování. Protože již využíváme sběrnici I^2C pro akcelometr, použijeme pro jednoduchost zapojení paměť EEPROM vybavenou toto sběrnicí.

2.1.4 Displej

K zobrazení hodnot získaných z akcelerometru a dalších informací byl použit dvouřádkový LCD displej řízený obvodem HD44780. Displej může komunikovat s mikroprocesorem pomocí čtyř nebo osmi bitů. V případě 4bitové komunikace se data zasílají ve dvou krocích, kdy se nejprve zašle horní a poté dolní polovina bajtu. Pro navrhované zařízení byla s ohledem na rychlost komunikace zvolena 8bitová komunikace.

V tabulce 16 jsou popsány vývody displeje. Pro komunikaci je třeba nejprve nastavit signály RS a R/W. Signál RS určuje, zda se budou zasílat data nebo příkazy. Signál R/W rozhoduje o tom, jestli se budou data či příkazy na displej zapisovat nebo budou z paměti displeje čtena. Jelikož budou na displej data vždy zasílána, bude tento bit trvale v logické 0. Pokud jsou požadovaná data na sběrnici DB0 až DB7 stabilní, zapíší se kladným impulsem na E.

Tabulka 16 – Vývody displeje

Číslo vývodu	Název	Funkce
1	GND	Zem
2	U_{CC}	Napájecí napětí
3	U_o	Nastavení kontrastu displeje
4	RS	Příkaz (0), data (1)
5	$R / \overline{W}$	Čtení (1), zápis (0)
6	E	Vstup povolení
7	DB0	Data nebo příkaz
8	DB1	Data nebo příkaz
9	DB2	Data nebo příkaz
10	DB3	Data nebo příkaz
11	DB4	Data nebo příkaz
12	DB5	Data nebo příkaz
13	DB6	Data nebo příkaz
14	DB7	Data nebo příkaz
15	A	Anoda podsvětlovací LED
16	K	Katoda podsvětlovací LED

2.2 Návrh schéma zapojení

Schéma zařízení se skládá z několika základních částí:

- Zdroj stabilizovaného napětí 5V
- Zapojení mikroprocesoru
- Zapojení akcelerometru
- Paměti EEPROM
- Tlačítka
- Signalizační LED diody
- Rozhraní pro připojení LDC displeje

Zdroj

Pro návrh zdroje stabilizovaného napětí 5V je použit monolitický integrovaný stabilizátor 7805, který je zapojen dle standardního schématu. Stabilizátor má maximální vstupní napětí 35V, což je pro případ napájení z 12 nebo 24 V dostatečné. Na vstupu i na výstupu stabilizátoru jsou použity blokovací kondenzátory 100 nF, které jsou umístěné co nejblíže integrovanému obvodu, aby zabránily jeho případnému kmitání. Na vstupu i na výstupu jsou rovněž filtrační

elektrolytické kondenzátory větší kapacity na dostatečně velké napětí, alespoň 2x větší než je předpokládané maximální napětí na kondenzátoru. Na vstupu zdroje je umístěna sériová dioda v propustném směru, která zabraňuje případnému otočení polarity vstupního napětí, které by jinak mělo za následek zničení zařízení. Na výstupu zdroje je signalizační dioda LED, která indikuje napětí na výstupu stabilizátoru.

Procesor

Vlastní procesor nepotřebuje mnoho vnějších součástek. Procesor je zapojen na napájecí napětí, které je opět blokováno proti rušení kondenzátorem 100nF, umístěným co nejblíže procesoru. Dále je k procesoru zapojen externí oscilátor dle doporučení v katalogovém listu procesoru. Je použit krystal maximální možné frekvence 16 MHz, aby byla dosažena dostatečná rychlost procesoru s kondenzátory hodnoty 15pF, dle tabulky v katalogovém listu. Dále je zapojen obvod resetu procesoru, který se skládá z odporu zapojeného na napájecí napětí a tlačítka. Datová sběrnice displeje je připojena na port A a řídící signály na port B, bity 6 a 7. Na portu B jsou připojené též signalizační LED diody. Komunikační signály sériové linky a I^2C sběrnice jsou zapojené na k tomuto účelu vyhrazené signály procesoru na portu C0, C1, D0 a D1.

Akcelerometr

Akcelerometr je připojen na napájecí napětí 5 V a dále na sběrnici I^2C na standardní vývody procesoru SDA a SCL. Akcelerometr je již sestaven na plošném spoji opatřeném dvěma řadami vývodů. Na plošném spoji jsou tedy osazeny dvě řady dutinek s odpovídající roztečí.

Paměť EEPROM

Pro jednoduchost zapojení je použita paměť EEPROM s rozhraním I^2C. Aby byla dosažena požadovaná kapacita, jsou použity celkem čtyři paměti 24AT1024. Tento obvod byl v konečné fázi nahrazen ekvivalentím obvodem ATMLU132. Tyto paměti neadresujeme pomocí adresovacích vstupů tak aby každá měla jinou adresu, čímž dosáhneme rozložení po celém adresovém prostoru 512 kB. Rozhraní I^2C zapojíme na standardní vývody procesoru SDA a SCL.

Tlačítka

Tlačítka jsou skenována průběžně v hlavní smyčce programu a není potřeba tedy využívat přerušení procesoru při stisku tlačítka. Tlačítka jsou zapojena přímo na port D procesoru s tím, že budou uzemňovat daný pin portu. Port je pro jistotu posílen externím odporem z napájecího napětí.

Signalizační LED

Signalizační led je zapojena rovněž přímo na port procesoru a to z napájecího napětí tak, aby proud diodou tekl při uzemnění pinu portu (v tomto

případě má port největší proudové zatížení). Proud diodou je volen tak, aby měly přibližně stejnou svítivost.

Rozhraní LCD Displeje

LDC displej je zapojen ve standardním 8vodičovém připojení. To znamená, že 8 bitů dat je zapojených přímo na port procesoru a řídící signály R/S a E jsou zapojené na volné piny libovolného portu procesoru. Signál R/W není pro jednoduchost využíván. Samozřejmě displej včetně podsvícení je také připojen na napájecí napětí. Protože připojení displeje není pro provoz zařízení nutný, je na plošném spoji osazen pouze konektor, kterým je možné displej připojit.

2.3 Návrh plošného spoje

Návrh plošného spoje spočívá ve volbě použité technologie výroby, tvorbě pouzder jednotlivých použitých součástek, definici obrysu desky, optimálnímu rozmístění součástek a návrhu vedení jednotlivých spojů. V každé výše uvedené fázi návrhu desky plošného spoje je potřeba dodržovat základní tři hlediska:

- **1. Vyrobitelnost**

  Deska musí být navržená tak, aby byla vyrobitelná. Musí se určit počet vrstev plošného spoje, respektovat třídy přesnosti a mít na zřeteli technologické možnosti výroby. Dále se musí vzít v úvahu ekonomičnost výroby s ohledem na vyráběné množství a použitou technologii.

- **2. Osazování**

  Způsob osazování navrhované desky ovlivňuje požadavky na definici pouzder součástek a jejich rozmístění na desce plošného spoje. Jsou dva základní způsoby osazování nebo jejich kombinace. Jedná se o ruční osazování, kde se používají zejména vývodové součástky nebo automatické osazování, kde se používají bezvývodové součástky SMD s různými typy vývodů. S ohledem na technologii je třeba dodržet pravidla vzdálenosti vývodů, aby nedošlo k jejich nechtěnému propojení při pájení.

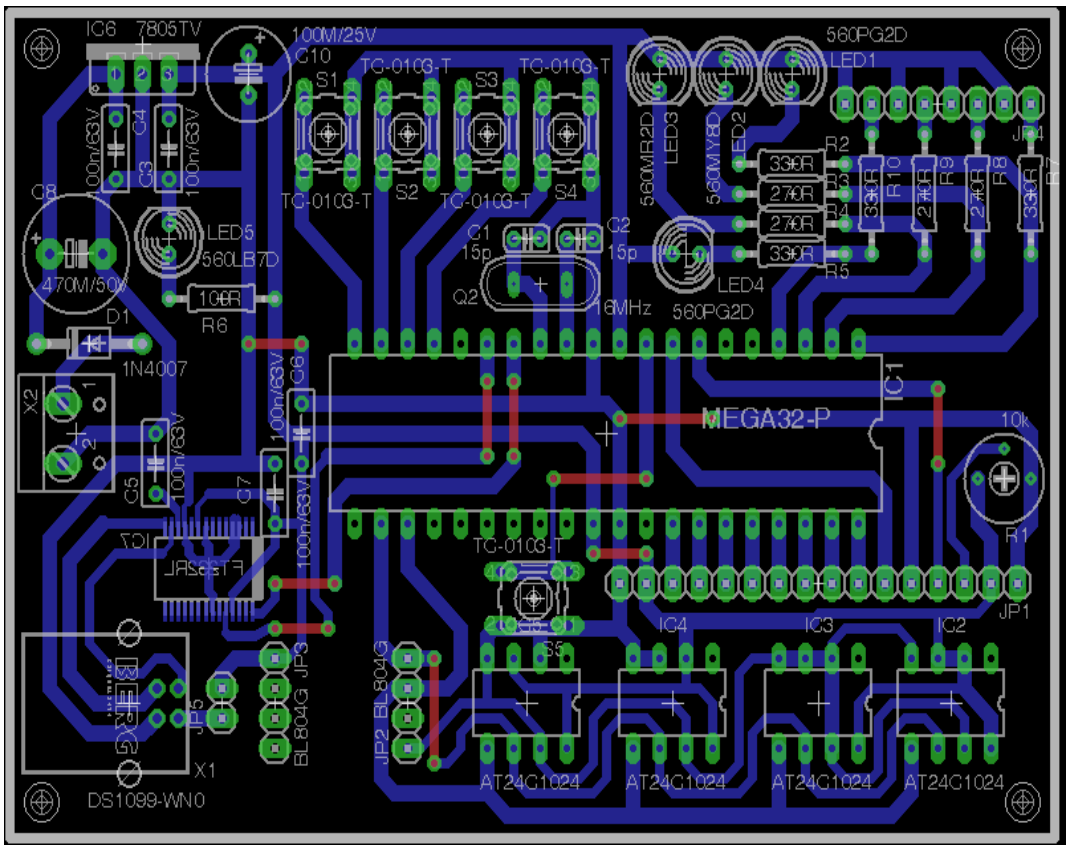
- **3. Elektrická funkce**

  Toto hledisko obsahuje mnoho důležitých parametrů. Na základě znalosti funkce obvodu, pro který je navrhován plošný spoj, musí být provedeno správné rozmístění součástek, respektována pravidla maximálního proudového a napěťového zatížení spojů, otázky přeslechu, impedancí, zpoždění při šíření signálu, způsobu zemnění, odvodu tepla, elektromagnetické kompatibility apod.

Pro účely návrhu plošného spoje daného zařízení se vychází z toho, že se jedná o kusovou výrobu v amatérských podmínkách. S ohledem na hustotu integrace je zvoleno dvouvrstvé provedení plošného spoje, které je ještě v amatérských podmínkách možno vyrobit s tím, že se použijí dostatečně silné spoje tak aby byly náhradní technologií leptání vyrobitelné. Toto povede k tomu, že i když je použita oboustranná deska, bude optimální

použít drátovou propojku a ne tenký spoj, který by byl obtížně vyrobitelný. Problém je pouze s integrovanými obvody, které se vyrábějí pouze v SMD technologii s vysokou hustotou vývodů.

Součástky jsou rozmístěny po jednotlivých obvodových celcích s dodržení zásad zejména blokování rušení na napájecím napětí a za mezení kmitání napájecího zdroje. To znamená blokovací kondenzátory co nejblíže jednotlivým integrovaným obvodů a vývodům integrovaného stabilizátoru. Další zásadou je co nejblíže u procesoru umístit krystalový oscilátor a dále vodiče, které souží pro sériové komunikace provést co nejkratší cestou. Výsledný návrh desky plošného poje je uveden na obrázku 9.



Obrázek 9 – Plošný spoj

2.4 Programová část

Pro vývoj software bylo použito prostředí Atmel Studio 6.0, které je volně k dispozici na stránkách výrobce procesorů ATMEL. Toto prostředí umožňuje programovat jak v assembleru, tak v jazyku C, pro který již obsahuje všechny potřebné knihovny.

Celý program je pro přehlednost rozdělen do několika modulů.

- hlavní modul akcelerometr.c
- modul práce s obvodem akcelerometru MMA7455L.c, MMA7455L.h
- modul komunikace i2c i2c.c, i2c.h
- modul sériové komunikace uart.c, uart.h
- modul ovládání paměti EEPROM ee_prom.c, ee_prom:h
- modul ovládání displeje LCD lcd.c, lcd.h

Níže následuje popis jednotlivých modulů.

2.4.1 Hlavní modul

Tento hlavní modul obsahuje definice základních parametrů programu tak, aby bylo možné jednoduše tyto parametry měnit. Jedná se zejména o parametry vyhodnocovacího procesu, definice související s hardware procesoru a připojenými periferiemi a definice ovlivňující překlad programu. Dále jsou zde definovány všechny globální proměnné programu.

Hlavní modul obsahuje všechny hlavní procedury včetně procedury „main“, která startuje při restartu mikroprocesoru a procedury volané při přerušení od časovače.

Seznam procedur v hlavním modulu:

- Hlavní procedura
- Funkce pro kalibraci
- Získání dat pro nastavení kalibrace
- Vypsání počtu překročení mezí na displej
- Vyhodnocovací procedura
- Odesílání dat na sériovou linku
- Konverze na mg
- Zobrazení zrychlení na LCD displeji
- Procedura přerušení od časovače

Tabulka 17 – Seznam globálních proměnných a jejich význam

Název proměnné	Datový typ	Význam
zrychleni[3]	signed char	Pole tří hodnot zrychlení zrychleni[0] – zrychlení v ose X zrychleni[1] – zrychlení v ose Y zrychleni[2] – zrychlení v ose Z
enabledWriteEeprom	unsigned char	Udává, zda je povolen zápis do EEPROM 0 – není povoleno 1 – je povoleno
count	unsigned long	počet měření
tableLevel1[x][2]	unsigned long	Tabulka překročení zrychlení první úrovně X – počet řádků tabulky daný jako podíl maximálního intervalu pro mazání a okna pro filtr zákmitů
tableLevel2[x][2]	unsigned long	Tabulka překročení zrychlení druhé úrovně X – počet řádků tabulky daný jako podíl maximálního intervalu pro mazání a okna pro filtr zákmitů
newRecordInTableLevel1	unsigned char	Ukazatel na nový záznam do tabulky překročení pro první úroveň zrychlení
newRecordInTableLevel2	unsigned char	Ukazatel na nový záznam do tabulky překročení pro druhou úroveň zrychlení

Hlavní procedura

Definice: int main(void)
Předávané hodnoty: nejsou
Návratová hodnota: není použita

Na začátku této procedury jsou nastaveny parametry portů procesoru, parametry časovače použitého pro přerušení, provedena kalibrace akcelerometru a provedeny inicializace obvodu UART a LCD displeje. Následně jsou povolena přerušení a program přechází do nekonečné smyčky, ve které jsou vyhodnocovány stisky tlačítek. Na základě těchto stisků jsou následně spouštěny některé procedury nebo nastavovány některé proměnné.

Přerušení od časovače

Definice: SIGNAL (SIG_INPUT_CAPTURE1)

Aby získaná měření mohla být jednoznačně určena v časové ose, jsou jednotlivá měření prováděna v přesných intervalech nastavených časovačem. V proceduře přerušení od časovače jsou tedy realizovány veškeré úkony spojené s měřením a vyhodnocením.

Jedná se o načtení naměřených hodnot z akcelerometru, jejich vyhodnocení a případné uložení do paměti a zobrazení na displeji. Některé události jsou podmíněné stisknutím příslušného tlačítka

Funkce pro kalibraci

Definice:	void calibrate(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

Tato funkce nastavuje do registrů akcelerometru data, která slouží pro korekci naměřených hodnot vzhledem k uložení akcelerometru (při naklopení akcelerometru od vodorovné roviny dochází k posunu naměřených hodnot vlivem zemské gravitace).

Nejdříve se pomocí funkce „Získání dat pro kalibraci“ zjistí průměrné naměřené hodnoty v jednotlivých osách. Tyto hodnoty se s opačným znaménkem uloží do korekčních registrů akcelerometru a opět se provede načtení průměrných hodnot. Tyto hodnoty již budou téměř nulové. Tyto hodnoty se přičtou k prvním hodnotám a znovu se záporným znaménkem uloží do korekčních registrů akcelerometru. Toto se provede ještě jednou. Poté se považuje nastavení korekcí akcelerometru za ukončené. Na základě pokusů bylo zjištěno, že případné další měření a korekce již tuto korekci nezpřesní.

Získání dat pro kalibraci

Definice:	void getValuesForCalibrate(long *data)
Předávané hodnoty:	nejsou
Návratová hodnota:	ukazatel na získaná data

Tato funkce získává data naměřená z akcelerometru pro potřeby kalibrace. Tato funkce provede sto měření ve všech osách a spočítá průměry pro jednotlivé osy, aby se eliminovalo nějaké rušení. Výsledné hodnoty rovnou vynásobí dvěma (korekční registry akcelerometru jsou ukládány v půlbitovém rozlišení) a otočí znaménko (korekce je opačného znaménka než změřené hodnoty).

Vypsání počtu překročení mezí na displej

Definice:	void showNumberErrorOnDisplay(unsigned char lev1, unsigned char lev2)
Předávané hodnoty:	počet překročení v jednotlivých úrovních
Návratová hodnota:	není

Při testování a ladění programu bylo potřeba zjistit, kolik překročení jednotlivých mezí je aktuálně uložené v tabulkách překročení. Pro tento účel byla vytvořena tato funkce, která na základě předaných hodnot překročení jednotlivých úrovní tyto zobrazí na displeji

Vyhodnocení

Definice:	void evaluation(unsigned long count, unsigned char max)
Předávané hodnoty:	číslo měření (odpovídá uplynulému času), maximum z naměřených hodnot
Návratová hodnota:	není

Tato funkce provádí kompletní vyhodnocení zrychlení a zobrazuje výsledek tohoto hodnocení pomocí tří LED diod. Algoritmus měření je následující. Do funkce se předává právě naměřená maximální hodnota zrychlení a číslo měření, které se využívá pro stanovení uplynulého času.

V případě, že hodnota tohoto maxima překročí jednu z nastavených úrovní zrychlení, je tato hodnota zaznamenána do tabulky zrychlení pro danou úroveň. V případě prvního záznamu se zapíše i číslo měření. Pokud další maximum přijde do nastaveného intervalu (zde byl nastaven interval 3 sekund) je porovnáno s původním zrychlením a pokud je větší, je původní zrychlení na stejné pozici tabulky přepsáno. Pokud přijde další maximum po uplynutí tohoto intervalu, je zapsáno do další pozice tabulky. Tímto způsobem se filtrují zákmity vzniklé při prudkých změnách zrychlení. Takto dochází postupně k plnění tabulek. Následně je zjišťováno, zda počet nenulových hodnot v tabulce není větší než nastavený maximální počet. Pokud ano dojde k rozsvícení příslušné Led diody. V případě překročení počtu v první tabulce se rozsvítí žlutá LED dioda, v případě druhé tabulky pak červená LED dioda. Dále jsou z tabulek hodnoty starší než nastavený interval (zde 5 minut) mazány. To znamená, že případné náhodné překročení hodnot zrychlení, je po delší době bez překročení vynulováno a v případě poklesu počtu těchto překročení pak dojde k postupnému zhasnutí červené a poté i žluté LED diody a rozsvícení zelené LED diody.

Odesílání dat na sériovou linku

Definice:	void sendData(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

Tato funkce odešle data uložená v paměti EEPROM na sériovou linku. Data jsou odesílána přímo v čitelné podobě tak aby bylo možné je přímo zachytávat do textového souboru pro další analýzu. Počet odesílaných byte je uložen na prvních čtyřech bytech v paměti EEPROM.

Konverze na mg

Definice:	signed int convertToMiliG(signed char x, signed long koef)
Předávané hodnoty:	naměřená hodnota z akcelerometru, maximální hodnota měření v mg
Návratová hodnota:	není

Tato funkce konvertuje údaj získaný v akcelerometru, který je vždy v rozsahu -127 až 128 na vlastní hodnoty zrychlení v mg, podle maximální hodnoty zrychlení nastavené v akcelerometru.

Zobrazení zrychlení na LCD displeji

Definice:	void showAllAccelerationOnDisplay(signed int x, signed int y, signed int z, unsigned long count)
Předávané hodnoty:	hodnota zrychlení v mg v ose x hodnota zrychlení v mg v ose y hodnota zrychlení v mg v ose Z číslo měření (odpovídá uplynulému času)
Návratová hodnota:	není

Tato funkce zobrazí na dvouřádkovém LCD displeji hodnoty zrychlení v mg ve všech třech osách a číslo měření. Pro zrychlení této funkce se zobrazuje při sudém číslu měření pouze první řádek a při lichém číslu měření pouze druhý řádek.

2.4.2 Modul práce s obvodem akcelerometru

V tomto modulu jsou funkce potřebné pro práci s akcelerometrem. Jedná se o inicializaci akcelerometru a funkce pro čtení a zápis dat. Jsou zde použity funkce z modulu komunikace I^2C.

Seznam procedur v modulu:

- Inicializace akcelerometru
- Zápis do registru akcelerometru pomocí I^2C
- Čtení z registru akcelerometru pomocí I^2C
- Čtení všech tří registrů zrychlení současně pomocí I^2C

Inicializace akcelerometru

Definice:	void MMA7455L_I2cInit(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

V této funkci jsou nastaveny požadované hodnoty v registrech akcelerometru a to jednak v registru Mode Control Register a jednak v registru Control 1. Hodnoty jsou nastavené takto:

Mode Control Register

GLVL [1:0] = 01; rozsah měření ± 2g
MODE [1:0] = 01; měřící mód

Control 1

XDA = 0; měření v ose x povoleno
YDA = 0; měření v ose y povoleno
ZDA = 0; měření v ose z povoleno
DFWB = 1; vzorkovací kmitočet nastaven na 125 Hz

Zápis do registru akcelerometru pomocí I^2C

Definice:	void MMA7455L_I2cWriteRegister(unsigned char address, unsigned char value)
Předávané hodnoty:	adresa registru, kam budeme zapisovat, hodnota pro zápis
Návratová hodnota:	není

Tato funkce zapisuje do registru akcelerometru pomocí I^2C sběrnice. Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Zahájení komunikace startovací sekvencí
- Odeslání na sběrnici I^2C HW adresy vlastního obvodu akcelerometru s nastavením prvního bitu na 0 (zápis)
- Odeslání na sběrnici I^2C adresy v paměti kam chceme zapisovat
- Odeslání na sběrnici I^2C vlastní hodnoty pro zápis
- Ukončení komunikace sekvencí stop

Čtení z registru akcelerometru pomocí I^2C

Definice: unsigned char MMA7455L_I2cReadRegister(unsigned char address)
Předávané hodnoty: adresa, z které budeme číst
Návratová hodnota: přečtená hodnota z registru

Tato funkce čte z hodnoty z registru akcelerometru pomocí I^2C sběrnice. Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Zahájení komunikace startovací sekvencí
- Odeslání na sběrnici I^2C HW adresy vlastního obvodu akcelerometru s nastavením prvního bitu na 0 (zápis)
- Odeslání na sběrnici I^2C adresy ze které chceme číst hodnotu
- Zopakování startovací sekvence před vlastním čtením
- Odeslání na sběrnici I^2C HW adresy vlastního obvodu akcelerometru s nastavením prvního bitu na 1 (čtení)
- Čtení hodnoty ze sběrnice I^2C
- Ukončení komunikace sekvencí stop

Čtení všech tří registrů zrychlení současně pomocí I^2C

Definice: void MMA7455L_I2cReadAccelerationRegister(unsigned char address, signed char* data)
Předávané hodnoty: adresa prvního registru ze tří po sobě jdoucích registrů, ze kterých budeme číst, ukazatel na pole přečtených hodnot
Návratová hodnota: není

Tato funkce čte současně ze tří registrů po sobě jdoucích jejich hodnoty pomocí I^2C sběrnice. Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Zahájení komunikace startovací sekvencí
- Odeslání na sběrnici I^2C HW adresy vlastního obvodu akcelerometru s nastavením prvního bitu na 0 (zápis)
- Odeslání na sběrnici I^2C adresy ze které chceme číst hodnotu
- Zopakování startovací sekvence před vlastním čtením
- Odeslání na sběrnici I^2C HW adresy vlastního obvodu akcelerometru s nastavením prvního bitu na 1 (čtení)
- Čtení hodnoty prvního registru ze sběrnice I^2C
- Čtení hodnoty druhého registru ze sběrnice I^2C
- Čtení hodnoty třetího registru ze sběrnice I^2C
- Ukončení komunikace sekvencí stop

2.4.3 Modul komunikace I^2C

Tento modul obsahuje funkce, které definují I^2C komunikaci. Kromě funkcí pro zápis a čtení pomocí I^2C jsou zde také definované START a STOP sekvence.

Seznam procedur v modulu:

- START sekvence
- STOP sekvence
- Čtení pomocí I^2C
- Zápis pomocí I^2C

START sekvence

Definice:	void I2C_start(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

V této funkci jsou nejprve nastaveny linky SDA a SCL do logické 1. Po uplynutí 10μs je linka SDA nastavena do logické 0. po dalších 10μs je do logické 0 nastavena i linka SCL. Tímto je startovací sekvence ukončena.

STOP sekvence

Definice:	void I2C_stop(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

Před voláním této funkce jsou linky SDA a SCL v logické 0. V této funkci je nejprve nastavena linka SCL do logické 1. Po uplynutí 10μs je nastavena do logické 1 i linka SDA. Tímto je sekvence ukončena.

Čtení pomocí I^2C

Definice:	unsigned char I2C_read(unsigned char ack)
Předávané hodnoty:	parametr, který určuje, zda bude čtena další hodnota v pořadí
Návratová hodnota:	přečtená hodnota ze sběrnice

Tato funkce čte ze sběrnice I^2C 8bitovou hodnotu. V cyklu jsou čteny jednotlivé bity ze sběrnice I^2C a ukládány postupně do registru návratové hodnoty funkce a to od nevyššího bitu. Před vlastním čtením bitu je nastavena linka SCL do logické 1 a čeká se 10μs. Po přečtení je linka SCL nastavena do logické 0 a opět se čeká 10μs.

Pokud má být čtena další hodnota, je nastavena linka SDA do logické 0, v opačném případě do logické 1. Následuje jeden pulz na lince SCL. Nakonec je nastavena linka SDA do logické 0.

Zápis pomocí I^2C

Definice:	unsigned char I2C_write(unsigned char b)
Předávané hodnoty:	vlastní hodnota pro zápis
Návratová hodnota:	signalizace chybného zápisu

Tato funkce zapisuje na sběrnici I^2C 8bitovou hodnotu. V cyklu jsou čteny jednotlivé bity předané hodnoty a to od nevyššího bitu. Podle toho zda se jedná o logickou 0 či 1 se nastaví linka SDA. Po nastavení hodnoty se provede pulz na lince SCL.

Obě linky SDA a SCL jsou nastaveny do logické 1 a čeká se, zda přijde potvrzení od zařízení, které přijímá na I^2C sběrnici poslanou hodnotu. Přijetí hodnoty signalizuje na lince SDA její změnou do logické 0. Pokud v daném časovém intervalu nedojde k potvrzení je návratová hodnota nastavena na 0. Nenulová návratová hodnota signalizuje, že vyslaný Byte byl přijatý.

2.4.4 Modul sériové komunikace

Tento modul obsahuje funkce, které definují sériovou komunikaci. Jedná se o inicializaci rozhraní a funkce pro zasílání dat.

Seznam procedur v modulu:

- Inicializace UART
- Vyslání znaku na UART
- Vyslání řetězce na UART
- Vyslání hodnoty typu int na UART¨
- Vyslání hodnoty typu unsigned long na UART

Inicializace UART

Definice:	void UART_init(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

V této funkci je nejprve nastavena komunikační rychlost na 115200 b/s, dále je povoleno vysílání i příjem a nastaven 8bitový formát dat.

Vyslání znaku na UART

Definice:	int UART_putchart(unsigned char c)
Předávané hodnoty:	odesílaný znak
Návratová hodnota:	není

Tato funkce odešle na UART jeden znak. Nejdříve se čeká, až bude odeslán předcházející znak (bude prázdný odesílací buffer UART). Jakmile je buffer prázdný, zapíše do odesílacího bufferu předaný znak. Po zápisu do odesílacího bufferu bude automaticky zahájeno odesílání tohoto znaku na UART.

Vyslání řetězce na UART

Definice:	void UART_putstr(char *s)
Předávané hodnoty:	ukazatel na začátek odesílaného řetězce
Návratová hodnota:	není

Tato funkce postupně v cyklu odesílá jednotlivé znaky řetězce na UART. Znaky jsou odesílány pomocí funkce UART_putchar. Protože každý řetězec je zakončen 0, jakmile v tomto cyklu bude načtena hodnota 0 (konec řetězce) je odesílání ukončeno.

Vyslání hodnoty typu int na UART

Definice:	void UART_putint(int value, unsigned char radix)
Předávané hodnoty:	hodnota pro odeslání, číselná soustava, ve které bude provedena konverze na řetězec
Návratová hodnota:	není

Tato funkce odešla předanou hodnotu int jako řetězec na UART. Nejdříve se provede konverze hodnoty na řetězec v dané číselné soustavě pomocí knihovní funkce itoa. Následně se vytvořený řetězec odešle pomocí funkce UART_putstr na UART.

Vyslání hodnoty typu unsigned long UART

Definice:	void UART_putunslong(unsigned long value, unsigned char radix)
Předávané hodnoty:	hodnota pro odeslání, číselná soustava, ve které bude provedena konverze na řetězec
Návratová hodnota:	není

Tato funkce odešle předanou hodnotu unsigned long jako řetězec na UART. Nejdříve se provede konverze hodnoty na řetězec v dané číselné soustavě pomocí knihovní funkce ultoa. Následně se vytvořený řetězec odešle pomocí funkce UART_putstr na UART.

2.4.5 Modul ovládání paměti EEPROM

Tento modul obsahuje funkce, které jsou potřebné pro zápis a čtení dat ze sériové EEPROM pomocí sběrnice I^2C. Jsou zde použity funkce z modulu komunikace I^2C.

Seznam procedur v modulu:

- Zápis adresy v paměti pomocí I^2C
- Čtení bloku dat z paměti pomocí I^2C
- Zápis bloku dat do paměti pomocí I^2C

Zápis adresy v paměti pomocí I^2C

Definice:	void I2C_EE_send_adr(unsigned long address)
Předávané hodnoty:	adresa kam budeme zapisovat nebo číst
Návratová hodnota:	není

Tato funkce odešle na sběrnici I^2C adresu ze které budeme následně číst nebo kam budeme zapisovat. Z celkového počtu 32 adresovacích bitů daných typem proměnné pro adresu je využito pouze 24 (tři byty). Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Zahájení komunikace startovací sekvencí
- Odeslání na sběrnici I^2C HW adresy spolu s nevyššími bity adresy s nastavením prvního bitu na 0 (zápis)
- Odeslání na sběrnici I^2C druhého bytu adresy
- Odeslání na sběrnici I^2C nejnižšího bytu adresy

Čtení bloku dat z paměti pomocí I^2C

Definice:	void I2C_EE_read_page(unsigned long address, unsigned char* dat, unsigned char cnt)
Předávané hodnoty:	adresa prvního bytu bloku dat, z které budeme číst, ukazatel na adresu kam se zapíší načtené hodnoty z paměti, počet čtených bytů
Návratová hodnota:	není

Tato funkce přečte blok dat požadované délky počínaje zadanou adresou. Přečtené hodnoty uloží do určeného místa v paměti. Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Nejprve se pomocí funkce I2C_EE_send_adr odešle do paměti požadovaná adresa, ze které se bude číst
- Pokračování komunikace startovací sekvencí
- Odeslání na sběrnici I^2C HW adresy, nevyšších bite adresy s nastavením prvního bitu na 1 (čtení)
- Čtení požadovaného počtu hodnot ze sběrnice I^2C v cyklu
- Ukončení komunikace sekvencí stop

Zápis bloku dat do paměti pomocí I^2C

Definice:	unsigned char I2C_EE_write_page(unsigned long address, unsigned char* dat, unsigned char cnt)
Předávané hodnoty:	adresa prvního byte bloku dat, z které budeme číst, ukazatel na adresu kam se zapíší načtené hodnoty z paměti, počet čtených byte
Návratová hodnota:	chyba zápisu

Tato funkce uloží blok dat požadované délky počínaje zadanou adresou do paměti EEPROM. Sekvence volání jednotlivých funkcí komunikace I^2C sběrnice je následující:

- Nejprve se pomocí funkce I2C_EE_send_adr odešle do paměti požadovaná adresa, kam se bude zapisovat
- Zápis požadovaného počtu hodnot na sběrnice I2C v cyklu
- Ukončení komunikace sekvencí stop
- Podle návratové hodnoty funkce I2C_write vrátí informaci, zda se zápis provedl nebo ne. Hodnota různá od 0 znamená, že se zápis nepodařil.

2.4.6 Modul ovládání displeje LCD

V tomto modulu jsou funkce potřebné pro zobrazování dat na LCD displeji a jeho ovládání.

Seznam procedur v modulu:

- Inicializace displeje
- Zápis 8bitové hodnoty na sběrnici LCD displeje
- Zápis dat do paměti LCD displej
- Zápis dat na displej LCD displeje
- Vymazání displeje
- Nastavení pozice kurzoru
- Zobrazení znaku na LCD displej
- Zobrazení řetězce na LCD displej
- Zobrazení hodnoty int na LCD displej
- Zobrazení hodnoty unsigned long na LCD displej

Inicializace displeje

Definice:	void LCD_init(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

Tato funkce inicializuje displej v 8bitovém režimu. Tato funkce se skládá ze z následující sekvence:

- Nastavení portů procesoru dle funkce jednotlivých signálů
- Odeslání posloupnosti příkazů pro displej

Zápis 8bitové hodnoty na sběrnici LCD displeje

Definice:	void LCD_clock(unsigned char data, unsigned char rs)
Předávané hodnoty:	data pro zápis, hodnota, která určuje, zda se budou zapisovat data nebo instrukce
Návratová hodnota:	není

Tato funkce zapíše na sběrnici LCD displeje data nebo instrukci. Nejprve nastaví signál RS do logické 0 nebo 1 podle předaného parametru rs, který určuje, zda jsou zapisována data nebo instrukce. Dále nastaví na sběrnici požadovaná data a vytvoří 60μs puls na signálu E. Tímto dojde k zápisu nastavených dat na displej. Po uběhnutí uvedeného pulsu se ještě počká dalších 10μs a může následovat další zápis.

Zápis instrukce do LCD displeje

Definice:	void LCD_send_0(unsigned char data)
Předávané hodnoty:	data pro zápis do paměti
Návratová hodnota:	není

Tato funkce zapíše instrukci do LCD displeje. Funkce využívá výše uvedenou funkci LCD_clock, kde do parametru zapíše požadovanou instrukci a druhý parametr nastaví na 0.

Zápis dat do LCD displeje

Definice:	void LCD_send_1(unsigned char data)
Předávané hodnoty:	data pro zápis na displej
Návratová hodnota:	není

Tato funkce zapíše data na displej LCD displeje. Funkce využívá výše uvedenou funkci LCD_clock, kde do parametru zapíše požadovaná data a druhý parametr nastaví na 1.

Vymazání displeje

Definice:	void LCD_clear(void)
Předávané hodnoty:	nejsou
Návratová hodnota:	není

Tato funkce vymaže displej pomocí příslušného příkazu a nastaví kurzor na pozici 0,0.

Nastavení pozice kurzoru

Definice:	void LCD_xy(unsigned char x, unsigned char y)
Předávané hodnoty:	souřadnice x a souřadnice y umístění kurzoru
Návratová hodnota:	není

Tato funkce podle požadovaných souřadnic a počtu řádků displeje vypočítá adresu v paměti displeje, kam je potřebu uložit další znak, čímž nastaví pozici kurzoru na x-tý sloupec a y-tý řádek.

Zobrazení znaku na LCD displeji

Definice:	int LCD_putchart(char ch)
Předávané hodnoty:	zobrazovaný znak
Návratová hodnota:	není

Tato funkce zobrazí na LCD displeji jeden znak na pozici kurzoru.

Zobrazení řetězce na LCD displeji

Definice:	void LCD_putstr(char *s)
Předávané hodnoty:	ukazatel na začátek zobrazovaného řetězce
Návratová hodnota:	není

Tato funkce postupně v cyklu vypisuje jednotlivé znaky řetězce na displej od nastavené pozice kurzoru. Znaky jsou odesílány pomocí funkce LCD_putchar. Protože každý řetězec je zakončen 0, jakmile v tomto cyklu bude načtena hodnota 0 (konec řetězce) je vypisování ukončeno.

Zobrazení hodnoty typu int na LCD displeji

Definice:	void LCD_putint(int value, unsigned char radix)
Předávané hodnoty:	hodnota pro zobrazení, číselná soustava, ve které bude provedena konverze na řetězec
Návratová hodnota:	není

Tato funkce zobrazí předanou hodnotu int jako řetězec na LCD displeji od pozice kurzoru. Nejdříve se provede konverze hodnoty na řetězec v dané číselné soustavě pomocí knihovní funkce itoa. Následně se vytvořený řetězec odešle pomocí funkce LCD_putstr na displej.

Zobrazení hodnoty typu unsigned long na LCD displeji

Definice:	void LCD_putunslong (unsigned long value, unsigned char radix)
Předávané hodnoty:	hodnota pro zobrazení, číselná soustava, ve které bude provedena konverze na řetězec
Návratová hodnota:	není

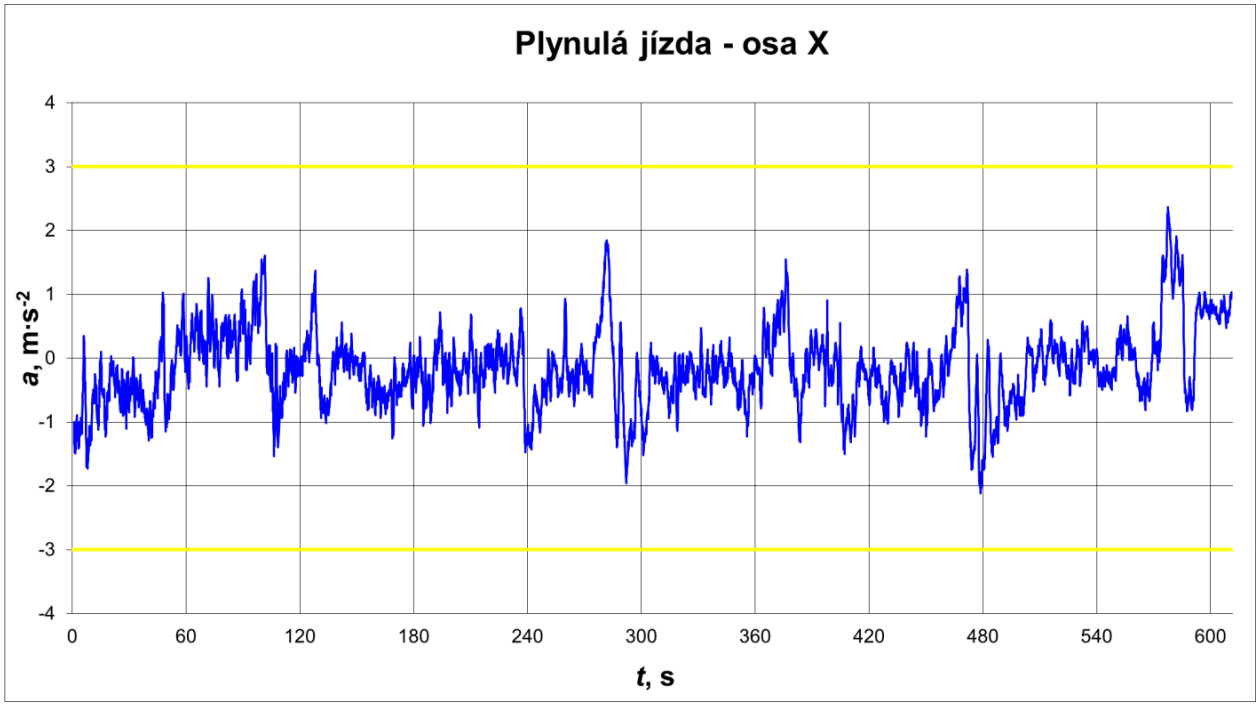
Tato funkce zobrazí předanou hodnotu unsigned long jako řetězec na LCD displeji od pozice kurzoru. Nejdříve se provede konverze hodnoty na řetězec v dané číselné soustavě pomocí knihovní funkce ultoa. Následně se vytvořený řetězec odešle pomocí funkce LCD_putstr na displej.

2.5 Testovací jízdy

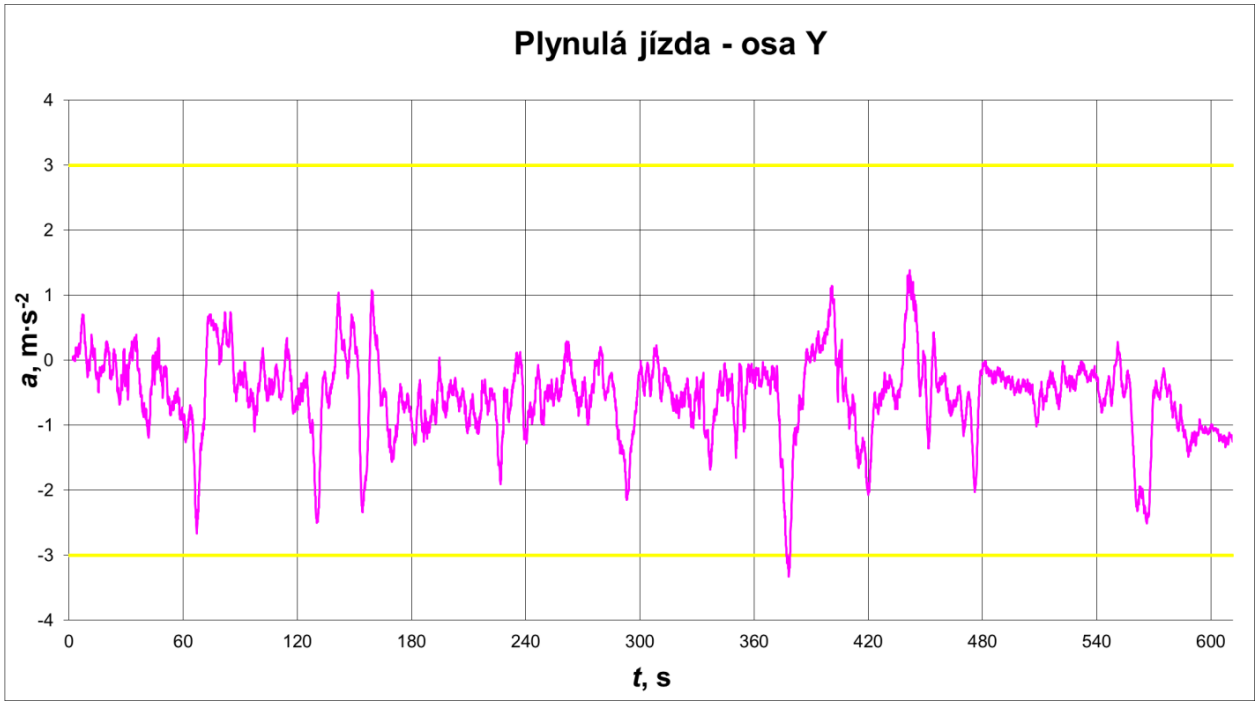
Při testovacích jízdách byla projížděna trasa o celkové délce 9200 metrů. Spotřeba při jednotlivých stylech jízdy je uvedena v litrech na 100 kilometrů v tabulce 18 spolu s dobou jízdy v minutách a průměrnou rychlostí v kilometrech za hodinu. Dále jsou zde uvedeny grafy zrychlení osy X a osy Y pro jednotlivé styly jízdy.

Tabulka 18 – Vyhodnocení jízdy

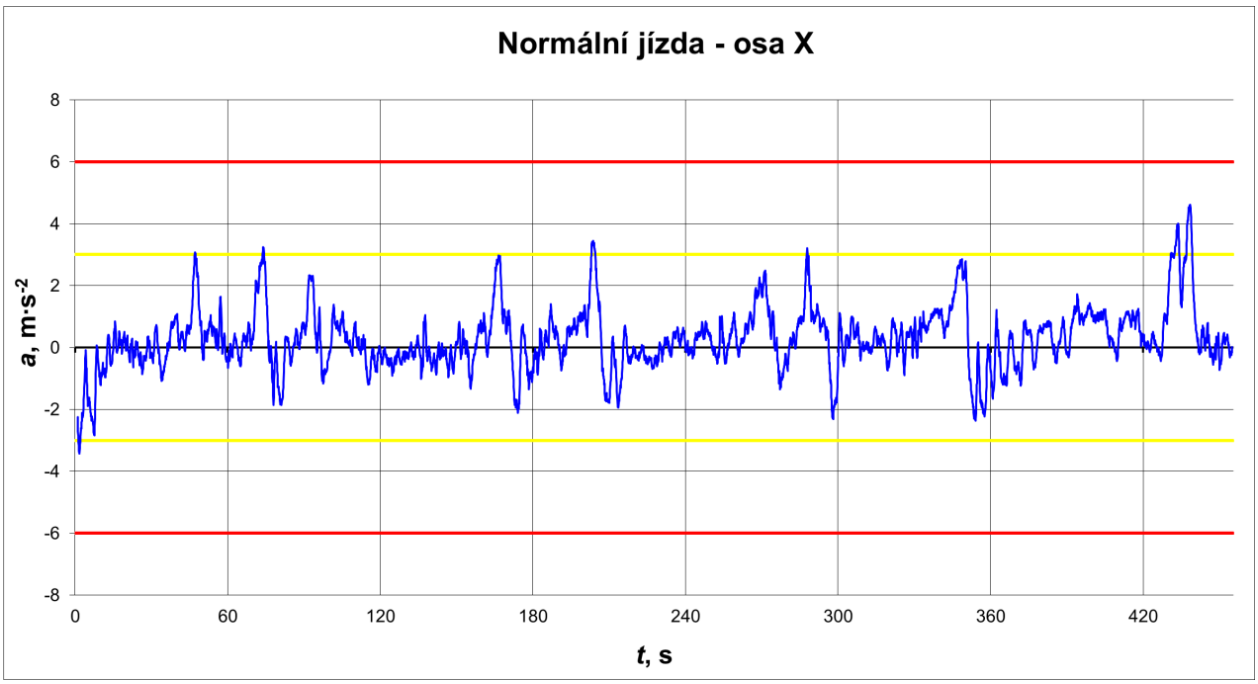
Typ jízdy	Spotřeba	Doba jízdy	Průměrná rychlost
Plynulá	5,4	10,2	54
Normální	6,9	7,6	72,6
Agresivní	9,2	6,8	80,4



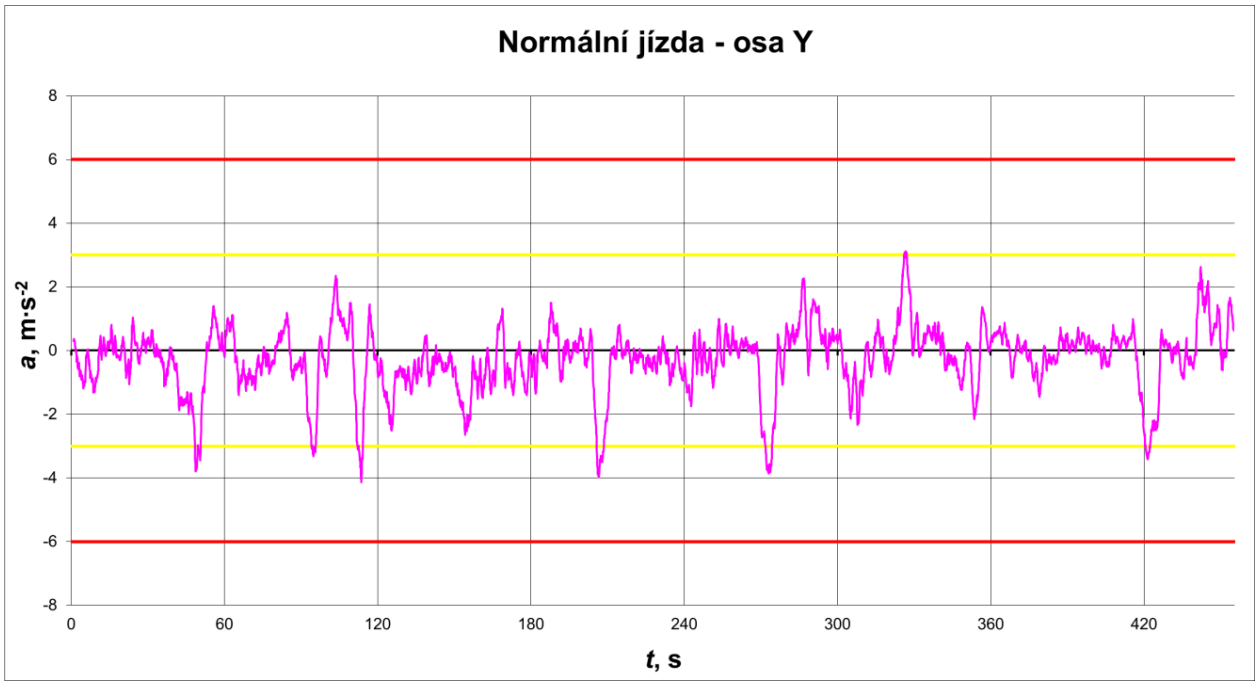
Obrázek 10 – Měření zrychlení plynulé jízdy v ose X



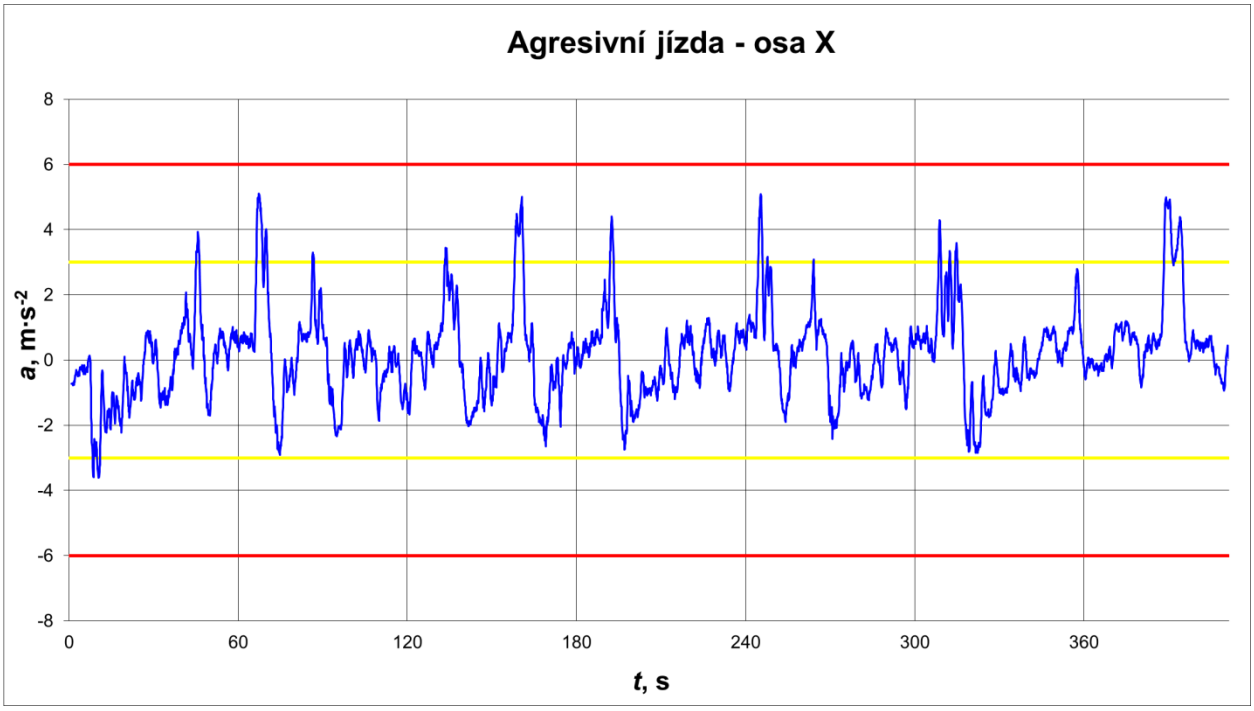
Obrázek 11 – Měření zrychlení plynulé jízdy v ose Y



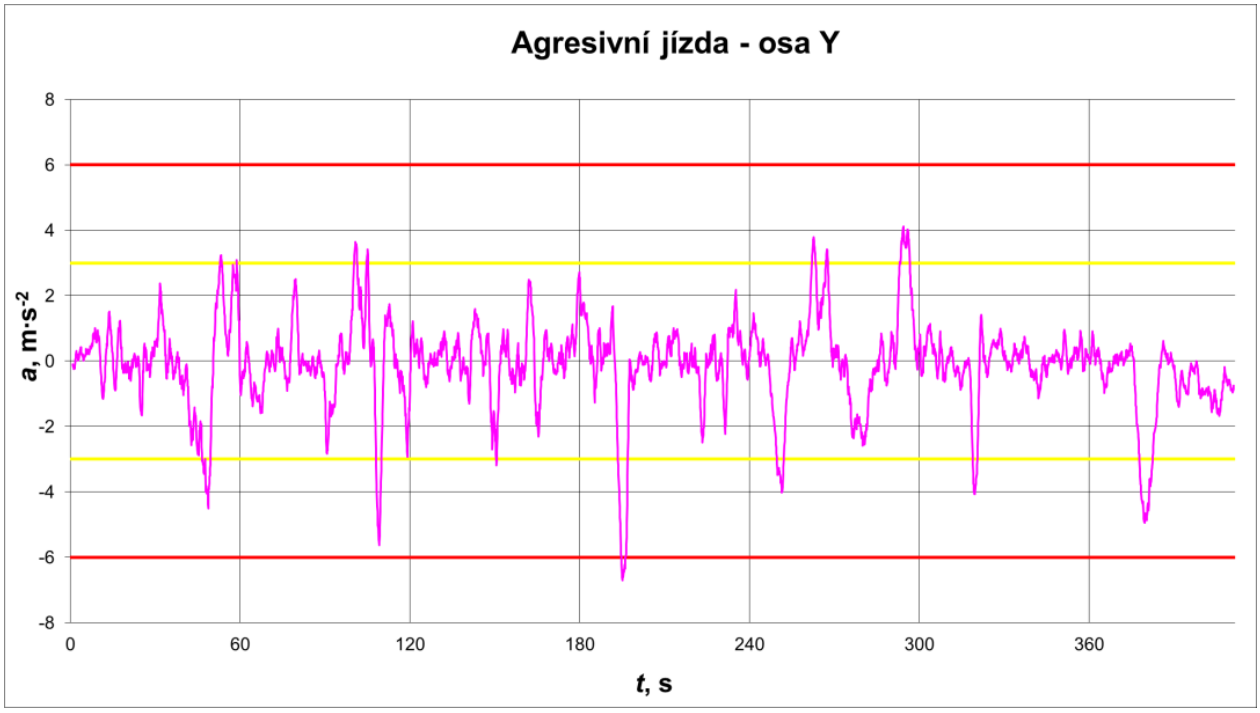
Obrázek 12 – Měření zrychlení normální jízdy v ose X



Obrázek 13 – Měření zrychlení normální jízdy v ose Y



Obrázek 14 – Měření zrychlení agresivní jízdy v ose X



Obrázek 15 – Měření zrychlení agresivní jízdy v ose Y

Z výše uvedených grafů je patrné, že teoretický předpoklad zvýšených hodnot zrychlení při agresivnější jízdě se potvrdil a je tedy možné z velikosti zrychlení provést vyhodnocení stylu jízdy řidiče.

Z výsledků je rovněž patrné, že teoreticky nastavené hranice pro agresivní styl jízdy jsou poměrně vysoké a při testovacích jízdách se je dařilo překračovat pouze výjimečně. Toto je zřejmě též zapříčiněno tím, že řidič vozidla nebyl dostatečně zkušený nebo odvážný aby dokázal svoji jízdou tyto hranice překročit a dále tím, že testovací jízdy byly prováděny v běžném provozu.

Vyhodnocení stylu pomocí jednoduchého zobrazení třemi barevnými LED diodami se jeví být dostatečné. Při plynulé jízdě většinu času svítila zelená LED dioda signalizující úspornou jízdu a jenom výjimečně docházelo ke krátkému rozsvícení žluté LED diody ve složitých úsecích trasy. Při normální jízdě většinu času svítila žlutá LED dioda signalizující střední styl jízdy a v klidných pasážích trasy na krátkou dobu zasvítila zelená LED dioda. Nikdy nedošlo k rozsvícení červené LED diody signalizující agresivní jízdu. Při agresivní jízdě již nikdy nedošlo k rozsvícení zelené LED diody (pouze na začátku testovací jízdy) a téměř celou trasu svítila žlutá LED dioda. V některých chvílích pak došlo ke krátkému rozsvícení červené LED diody signalizující agresivní styl jízdy.

Jak již bylo výše uvedeno, byly hranice pro agresivní styl jízdy nastavené poměrně vysoko, a tedy červená LED dioda se rozsvěcela jenom výjimečně.

3 Závěr

V této práci se mi podařilo navrhnout a sestavit jednotku vyhodnocující styl jízdy řidiče na základě měření zrychlení vozidla a to jak v předozadním tak bočním směru. Pro měření zrychlení byl vybrán akcelerometr MMA7455L, který komunikuje s mikroprocesorem ATMEGA32 pomocí I2C rozhraní. Na jednotce může být umístěn také LCD displej zobrazující například aktuální hodnoty zrychlení v osách X, Y a Z a také číslo měření. Měření probíhá při přerušení vyvolané mikroprocesorem, toto přerušení je nastaveno tak aby měření probíhalo desetkrát za vteřinu. Naměřená data se ukládají pomocí rozraní I2C do paměti EEPROM o celkové velikosti 4Mbit realizované na plošném spoji čtyřmi pamětmi AT24C1024. Při zapisování čtyř hodnot o velikosti jeden byte desetkrát za vteřinu umožní zapisovat hodnoty po dobu zhruba 220 minut. Data je dále možno následně zaslat přes sériové rozhraní UART do počítače, kde jsou pomocí programu terminálu uložena do textového souboru a dále vyhodnocena v programu Excel.

Zobrazení vyhodnocení stylu jízdy pro řidiče probíhá pomocí tří LED diod, které se rozsvěcí při opakovaném překročení nastavených hranic dle definovaného algoritmu. Hranice byly nastaveny na hodnoty 0,3g a 0,6g. Tyto hodnoty byly určeny na základě teoretických předpokladů. Celkový rozsah akcelerometru byl nastaven na ±2g, což je pro tuto úlohu dostačující.

Při zapnutí jednotky dojde automaticky ke kalibraci akcelerometru a je tedy možné umístit akcelerometr kamkoliv bez ohledu na přesné nastavení vodorovné polohy, pouze je nutné dodržet přibližný směr osy X ve směru jízdy vozidla.

Pro obsluhu obsahuje jednotka čtyři tlačítka. Tlačítko pro start měření, kdy jsou data zobrazována na LCD displeji a ukládána do paměti. Dále tlačítka pro pauzu a zastavení měření a tlačítko pro zaslání naměřených dat po sériové lince UART do počítače.

Pomocí jednotky byl v testovacích jízdách porovnáván styl jízdy řidiče s průměrnou spotřebou vozidla. Při překročení nejvyšší nastavené hranice je spotřeba zhruba dvakrát vyšší než při klidné jízdě. To znamená, že pokud řidič dodržuje klidný styl jízdy, dochází kromě jiného též k úspoře pohonných hmot.

Vyhodnocení stylu jízdy řidiče by bylo možné vylepšit například doplněním měření rychlosti s kontrolou překračování maximálních hodnot případně ovlivňování nastavených hranic v závislosti na rychlosti.

Literatura

[1] VOJÁČEK, Antonín. Freescale akcelerometru pro malá zrychlení (nízká g). *Automatizace.hw.cz* [online]. [cit. 2013-04-20]. Dostupné z WWW: http://automatizace.hw.cz/clanek/2005070301.

[2] VOJÁČEK, Antonín. Principy akcelerometrů - 2. díl Piezorezistivní. *Automatizace.hw.cz* [online]. [cit. 2013-04-20]. Dostupné z WWW: http://automatizace.hw.cz/clanek/2007012601

[3] VOJÁČEK, Antonín. Principy akcelerometrů - 1. díl - Piezoelektrické. *Automatizace.hw.cz* [online]. [cit. 2013-04-20]. Dostupné z WWW: http://automatizace.hw.cz/clanek/2007011401.

[4] ROZSÍVAL, Pavel. Akcelerometry. [online]. [cit. 2013-04-22]. Dostupné z WWW: http://vyuka.pavelrozsival.cz/isame/Akcelerometry.pdf

[5] VOJÁČEK, Antonín. Principy akcelerometrů - 3. díl - Tepelné akcelerometry MEMSIC. *Automatizace.hw.cz* [online]. [cit. 2013-04-21]. Dostupné z WWW: http://automatizace.hw.cz/clanek/2007040901.

[6] KOLOŠ, Petr. Zelené tlačítko v Hondě umí snížit spotřebu. *Motožurnál* [online]. [cit. 2013-07-25]. Dostupné z WWW: http://www.rozhlas.cz/motozurnal/automoto/_zprava/zelene-tlacitko-v-honde-umi-snizit-spotrebu--1037115

[7] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. 1. vyd. Praha: BEN - technická literatura, 2006. 319 s. ISBN 80-730-0174-8.

[8] Datasheet MMA7455. Parallax. [online]. [cit. 2013-04-25]. Dostupné z WWW: http://www.parallax.com/Portals/0/Downloads/docs/prod/sens/28526-MMA7455-3axisAccel-v1.1.pdf.

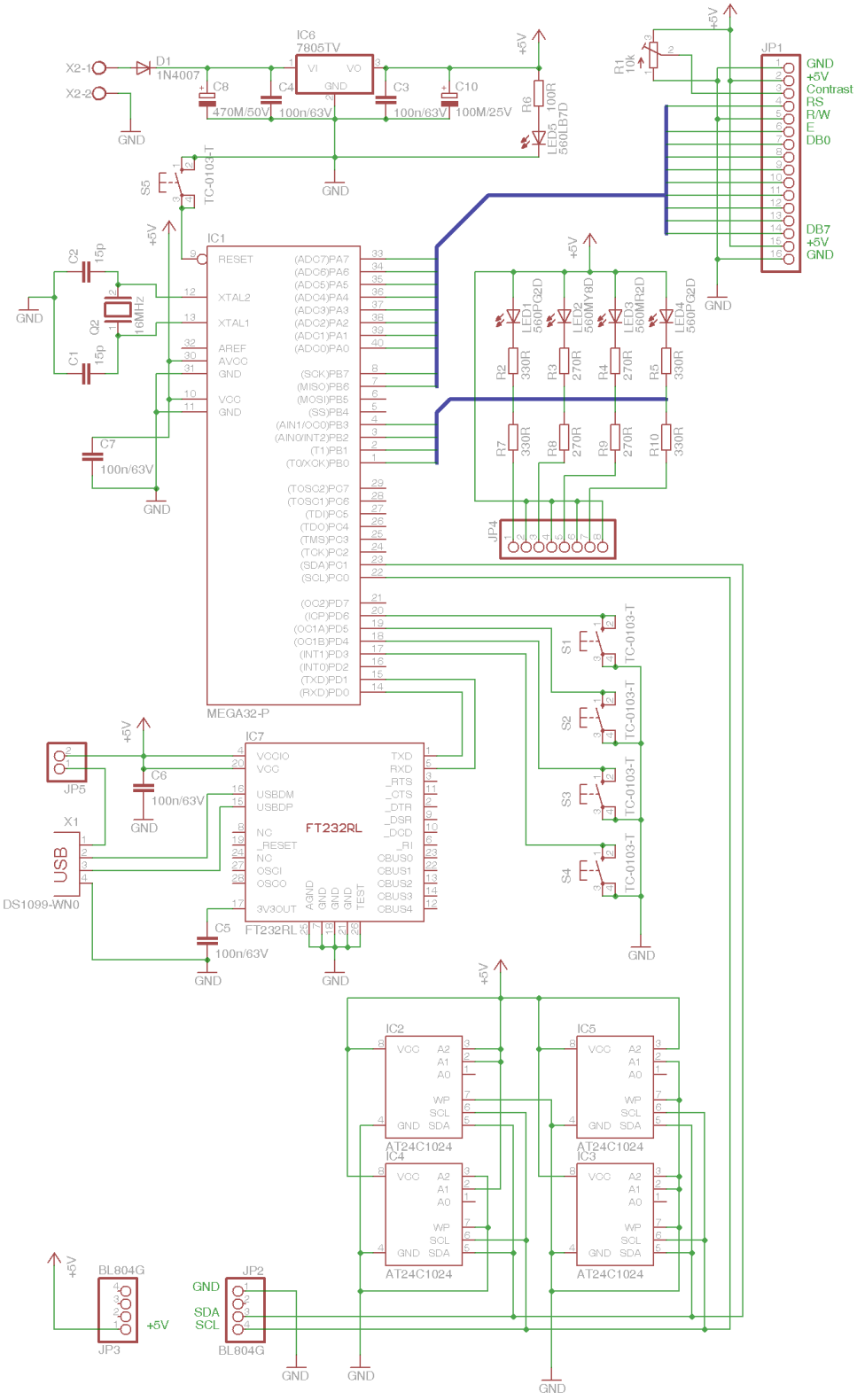
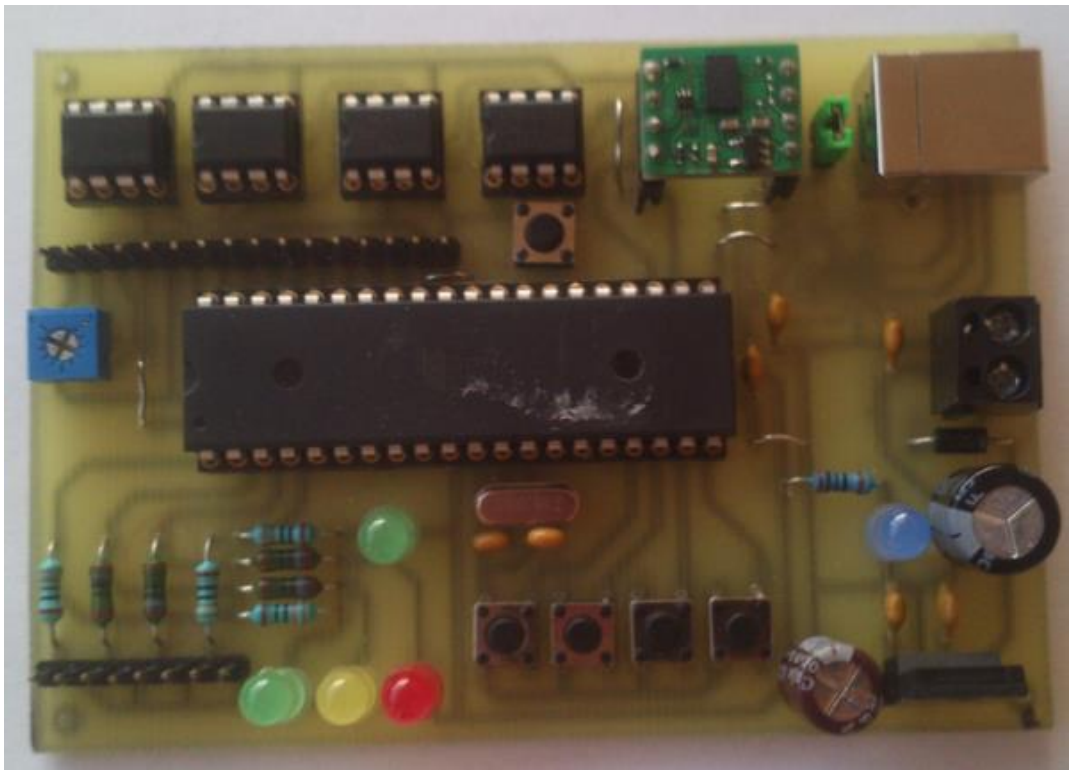
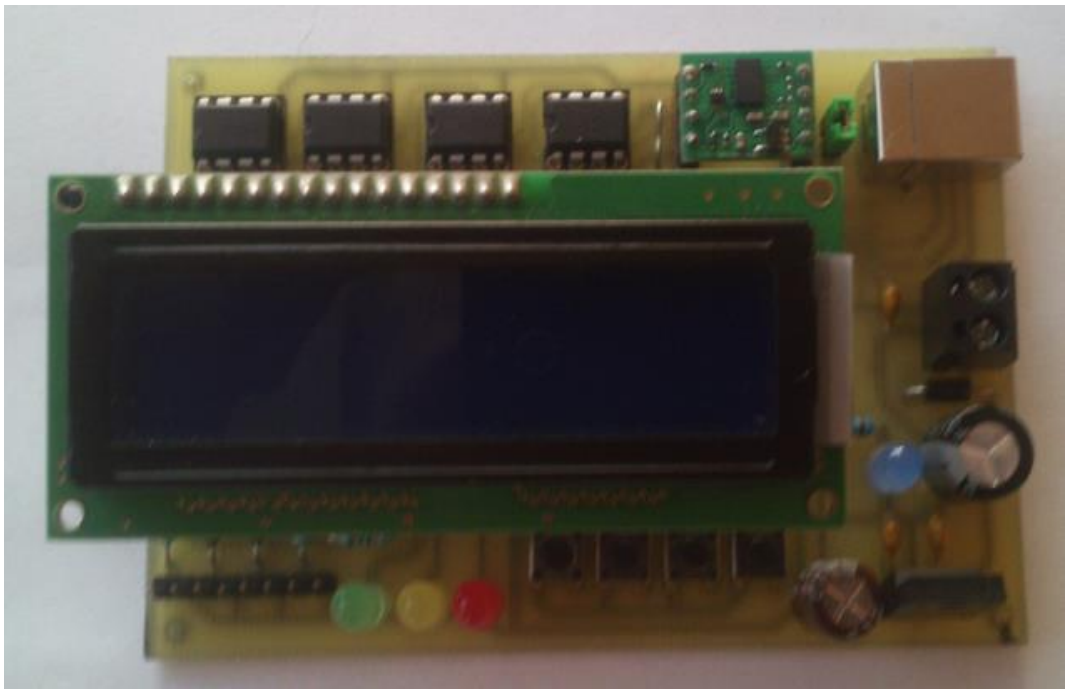
[9] Datasheet MMA7455L. Freescale. [online]. [cit. 2013-04-25]. Dostupné z WWW: http://www.freescale.com/files/sensors/doc/data_sheet/MMA7455L.pdf

[10] Datasheet ATmega32(L) Complete. Atmel. [online]. [cit. 2013-04-23]. Dostupné z WWW: http://www.atmel.com/Images/doc2503.pdf

Příloha A – Tabulka registrů akcelerometru MMA7455L

Adresa	Název	Definice	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$00	XOUTL	10 bits output value X LSB	XOUT[7]	XOUT[6]	XOUT[5]	XOUT[4]	XOUT[3]	XOUT[2]	XOUT[1]	XOUT[0]
$01	XOUTH	10 bits output value X MSB	--	--	--	--	--	--	XOUT[9]	XOUT[8]
$02	YOUTL	10 bits output value Y LSB	YOUT[7]	YOUT[6]	YOUT[5]	YOUT[4]	YOUT[3]	YOUT[2]	YOUT[1]	YOUT[0]
$03	YOUTH	10 bits output value Y MSB	--	--	--	--	--	--	YOUT[9]	YOUT[8]
$04	ZOUTL	10 bits output value Z LSB	ZOUT[7]	ZOUT[6]	ZOUT[5]	ZOUT[4]	ZOUT[3]	ZOUT[2]	ZOUT[1]	ZOUT[0]
$05	ZOUTH	10 bits output value Z MSB	--	--	--	--	--	--	ZOUT[9]	ZOUT[8]
$06	XOUT8	8 bits output value X	XOUT[7]	XOUT[6]	XOUT[5]	XOUT[4]	XOUT[3]	XOUT[2]	XOUT[1]	XOUT[0]
$07	YOUT8	8 bits output value Y	YOUT[7]	YOUT[6]	YOUT[5]	YOUT[4]	YOUT[3]	YOUT[2]	YOUT[1]	YOUT[0]
$08	ZOUT8	8 bits output value Z	ZOUT[7]	ZOUT[6]	ZOUT[5]	ZOUT[4]	ZOUT[3]	ZOUT[2]	ZOUT[1]	ZOUT[0]
$09	STATUS	Status registers	--	--	--	--	--	PERR	DOVR	DRDY
$0A	DETSRC	Detection source registers	LDX	LDY	LDZ	PDX	PDY	PDZ	INT2	INT1
$0B	TOUT	“Temperature output value“ (Optional)	TMP[7]	TMP[6]	TMP[5]	TMP[4]	TMP[3]	TMP[2]	TMP[1]	TMP[0]
$0C		(Reserved)	--	--	--	--	--	--	--	--
$0D	I2CAD	I^2C device address	I2CDIS	DAD[6]	DAD[5]	DAD[4]	DAD[3]	DAD[2]	DAD[1]	DAD[0]
$0E	USRINF	User information (Optional)	UI[7]	UI[6]	UI[5]	UI[4]	UI[3]	UI[2]	UI[1]	UI[0]
$0F	WHOAMI	“Who am I“ value (Optional)	ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]
$10	XOFFL	Offset drift X value (LSB)	XOFF[7]	XOFF[6]	XOFF[5]	XOFF[4]	XOFF[3]	XOFF[2]	XOFF[1]	XOFF[0]
$11	XOFFH	Offset drift X value (MSB)	--	--	--	--	--	XOFF[10]	XOFF[9]	XOFF[8]
$12	YOFFL	Offset drift Y value (LSB)	YOFF[7]	YOFF[6]	YOFF[5]	YOFF[4]	YOFF[3]	YOFF[2]	YOFF[1]	YOFF[0]
$13	YOFFH	Offset drift Y value (MSB)	--	--	--	--	--	YOFF[10]	YOFF[9]	YOFF[8]
$14	ZOFFL	Offset drift Z value (LSB)	ZOFF[7]	ZOFF[6]	ZOFF[5]	ZOFF[4]	ZOFF[3]	ZOFF[2]	ZOFF[1]	ZOFF[0]
$15	ZOFFH	Offset drift Z value (MSB)	--	--	--	--	--	ZOFF[10]	ZOFF[9]	ZOFF[8]
$16	MCTL	Mode control	--	DRPD	SPI3W	STON	GLVL[1]	GLVL[0]	MOD[1]	MOD[0]
$17	INTRST	Interrupt latch reset	--	--	--	--	--	--	CLRINT2	CLRINT1
$18	CTL1	Control 1	DFBW	THOPT	ZDA	YDA	XDA	INTRG[1]	INTRG[0]	INTPIN
$19	CTL2	Control 2	--	--	--	--	--	DRVO	PDPL	LDPL
$1A	LDTH	Level detection threshold limit value	LDTH[7]	LDTH[6]	LDTH[5]	LDTH[4]	LDTH[3]	LDTH[2]	LDTH[1]	LDTH[0]
$1B	PDTH	Pulse detection threshold limit value	PDTH[7]	PDTH[6]	PDTH[5]	PDTH[4]	PDTH[3]	PDTH[2]	PDTH[1]	PDTH[0]
$1C	PW	Pulse duration value	PD[7]	PD[6]	PD[5]	PD[4]	PD[3]	PD[2]	PD[1]	PD[0]
$1D	LT	Latency time value	LT[7]	LT[6]	LT[5]	LT[4]	LT[3]	LT[2]	LT[1]	LT[0]
$1E	TW	Time window for 2^{nd} pulse value	TW[7]	TW[6]	TW[5]	TW[4]	TW[3]	TW[2]	TW[1]	TW[0]
$1F		(Reserved)	--	--	--	--	--	--	--	--

Příloha B – Schéma zařízení



Příloha C – Zařízení pro vyhodnocování stylu jízdy