



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## **INVALIDNÍ VOZÍK**

WHEELCHAIR

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PAVEL KUTNAR**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JAROSLAV ŠEMBERA**

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Pavel Kutnar

**ID:** 106586

**Ročník:** 3

**Akademický rok:** 2009/2010

**NÁZEV TÉMATU:**

**Invalidní vozík**

**POKYNY PRO VYPRACOVÁNÍ:**

Navrhněte a zrealizujte vylepšení invalidního vozíku, který by měl být schopen zvládat některé pohyby autonomně. Invalidní vozík je v současné době vybaven řízením pomocí mikropočítače FOX Board LX832. Je osazen webovou kamerou, bezdrátovým modulem wifi, joystickem a ultrazvukovými senzory pro zabránění kolizím. Vozík je potřeba doplnit o modul odměřování otáček motorů, jednoduchou odometrii a řízení výkonové části.

**DOPORUČENÁ LITERATURA:**

Prof. Ing. František Šolc, CSc. - Modely kolových robotů

**Termín zadání:** 8.2.2010

**Termín odevzdání:** 31.5.2010

**Vedoucí práce:** Ing. Jaroslav Šembera

**prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

**UPOZORNĚNÍ:**

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

## **ABSTRAKT**

Práce popisuje návrh, vývoj a testování nové řídicí jednotky invalidního vozíku. Odometrie, kompletní ovládání vozíku a jednoduchý komunikační protokol nabízí nadřazenému systému možnost autonomního řízení. Dokument obsahuje popis modifikací vozíku, teorii odometrie včetně její aplikace a popis samotné jednotky. Základem elektronické části je mikrokontrolér ATmega16. Testy odhalují problematiku odometrie a ukazují možné zdokonalení.

## **KLÍČOVÁ SLOVA**

odometrie, robotika, invalidní vozík, autonomní řízení, navigace, Sabertooth2x25, FOX Board LX832, ATmega16

## **ABSTRACT**

Thesis describes the design, development and testing of new control unit of the wheelchair. Odometry, complete control of the wheelchair and a simple communication protocol provides to superior system an opportunity of autonomous control. The document contains a description of the wheelchair modifications, odometry theory including its application and a description of the unit itself. The basis of the electronic part is an ATmega16 microcontroller. Tests reveal problems of odometry and suggest possible improvements.

## **KEYWORDS**

odometry, robotics, wheelchair, autonomous control, navigation, Sabertooth2x25, FOX Board LX832, ATmega16

## **BIBLIOGRAFICKÁ CITACE**

KUTNAR, P. *Invalidní vozík*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 62 s. Vedoucí bakalářské práce Ing. Jaroslav Šembera.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Invalidní vozík jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **31. května 2010**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Jaroslavu Šemberovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **1. června 2009**

.....  
podpis autora

## OBSAH

|  |           |
|--|-----------|
| <b>1. ÚVOD .....</b>                       | <b>7</b>  |
| <b>2. INVALIDNÍ VOZÍK.....</b>             | <b>8</b>  |
| 2.1 Použitý vozík .....                    | 8         |
| 2.2 Upravený vozík .....                   | 9         |
| 2.2.1 Sabertooth 2X25 .....                | 10        |
| 2.2.2 FOX Board LX832 .....                | 11        |
| 2.3 Přidání modulu odometrie.....          | 12        |
| <b>3. ODOMETRIE.....</b>                   | <b>13</b> |
| 3.1 Chyby odometrie a jejich řešení .....  | 14        |
| 3.2 Aplikace odometrie .....               | 15        |
| 3.3 Odometrie difereciálního podvozku..... | 16        |
| <b>4. NÁVRH .....</b>                      | <b>18</b> |
| 4.1 Stabilizátor .....                     | 19        |
| 4.2 Výkonové výstupy .....                 | 20        |
| 4.3 Řídící logika.....                     | 21        |
| 4.4 Snímače.....                           | 23        |
| <b>5. REALIZACE .....</b>                  | <b>24</b> |
| 5.1 Změny zapojení.....                    | 25        |
| <b>6. PROGRAM.....</b>                     | <b>26</b> |
| 6.1 Konfigurace procesoru .....            | 26        |
| 6.2 Funkce main.....                       | 29        |
| 6.3 Přerušení od snímačů .....             | 31        |
| 6.4 Stop přerušení .....                   | 32        |
| 6.5 Přerušení čítače/časovače 0.....       | 33        |
| 6.6 Přerušení čítače/časovače 1 .....      | 34        |
| 6.7 Přerušení přijímače sériové linky..... | 37        |
| <b>7. KOMUNIKACE .....</b>                 | <b>40</b> |
| 7.1 Příchozí data.....                     | 40        |
| 7.1.1 Tabulka instrukcí .....              | 41        |

|   |           |
|---|-----------|
| 7.1.2 Příklady instrukcí.....                 | 43        |
| 7.2 Odchozí data .....                        | 44        |
| 7.2.1 Tabulka formátu odchozích dat .....     | 45        |
| 7.2.2 Příklad žádosti o polohová data.....    | 46        |
| 7.3 Příklady v C# .....                       | 47        |
| <b>8. ZAPOJENÍ.....</b>                       | <b>48</b> |
| 8.1 Zapojení výkonové části .....             | 48        |
| 8.2 Zapojení řídicí části.....                | 49        |
| <b>9. TESTY, MĚŘENÍ, LADĚNÍ.....</b>          | <b>51</b> |
| 9.1 Jeden průměr kol.....                     | 51        |
| 9.2 Modifikace pro dva rozdílné průměry ..... | 52        |
| 9.3 Parametry vozíku .....                    | 54        |
| <b>10. OVLÁDACÍ PROGRAM .....</b>             | <b>55</b> |
| <b>11. ZÁVĚR .....</b>                        | <b>57</b> |
| <b>12. POUŽITÁ LITERATURA A ZDROJE .....</b>  | <b>58</b> |
| <b>13. SEZNAM PŘÍLOH .....</b>                | <b>61</b> |
| <b>14. SEZNAM ZKRATEK, SYMBOLŮ .....</b>      | <b>62</b> |

## 1. ÚVOD

Tato práce má za úkol pomoci tělesně postiženým spoluobčanům. Robotika v dnešní době proniká do mnoha oblastí našeho života, zdravotnická technika není výjimkou.

Zapojil jsem se do rozsáhlého projektu *System pro automatické nakládání vozíku pro tělesně postižené*. Klade si za cíl vytvoření invalidního vozíku schopného vykonávat autonomně, případně s co nejmenší nutnou obsluhou, činnosti obtížně proveditelné pro pohybově handicapované. Například asistované, či plně autonomní zajíždění vozíku do zavazadlového prostoru automobilu.

Podmínkou pro bezchybné autonomní řízení je znalost pozice robota-vozíku. Stávající systém počítá se zpracováním obrazu z kamery umístěné na vozíku. Pro zvýšení spolehlivosti a umožnění dalších funkcí je nutná redundance dat. Mým úkolem je, vytvoření dalšího systému k měření pozice, odometrie. Zároveň umožnit nadřazenému systému ovládání celého vozíku.

Práce popisuje jednotlivé komponenty invalidního vozíku, čím se liší použitý vozík od standardního modelu. Dále je popsána teorie odometrie a její implementace do programu. Zabývám se návrhem a realizací hardwaru pro účel nahrazení původní řídicí jednotky. Jednotlivé části programu jsou popsány vývojovými diagramy. Nezbytné testy poskytují podklady pro další vývoj a potlačení nepřesností výpočtu odometrie.



## 2. INVALIDNÍ VOZÍK

Invalidní vozík je zařízení umožňující pohyb osob s postiženými dolními končetinami. Možných konstrukcí vozíku je mnoho, existují závodní speciály s tříkolovou konstrukcí, jak je zřejmé z anglického označení, wheelchair, nejčastější konstrukcí je sedadlo-židle opatřené podvozkem-koly.

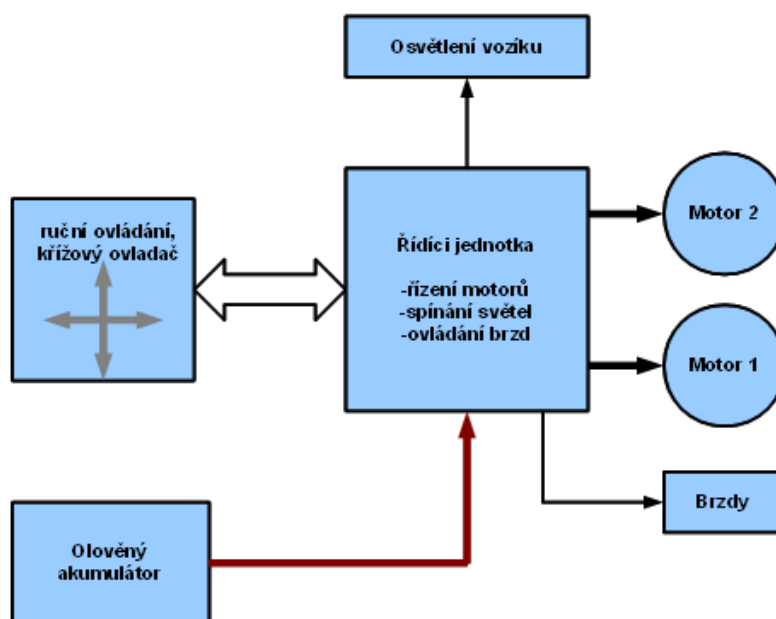
Pohon vozíku je zpravidla manuální nebo elektrický. Manuální koncepce pohonu umožňuje tlačení vozíku jinou osobou, případně pohyb obstarává sám uživatel. Mezi výhody tohoto řešení patří nízká hmotnost, spolehlivost, skladnost a cena. Nevýhoda je zřejmá, veškerou energii dodává člověk. Tento problém řeší vozíky s vlastním pohonem, v současné době elektrickým. Elektrifikace invalidního vozíku přinesla řadu možných vylepšení jako např. osvětlení, elektricky výškově polohovatelné sedadlo.

### 2.1 POUŽITÝ VOZÍK

Pro účely vývoje a testování systému je použitý invalidní vozík VASSILLI SPACE 1 [1] vybavený řídicí jednotkou Penny+Giles a motory SRG 01 [2] firmy AMT Schmid GmbH & Co. KG. Napájení zajišťuje dvojice 12V olověných akumulátorů.



Obr. 2.1 invalidní vozík SPACE 1

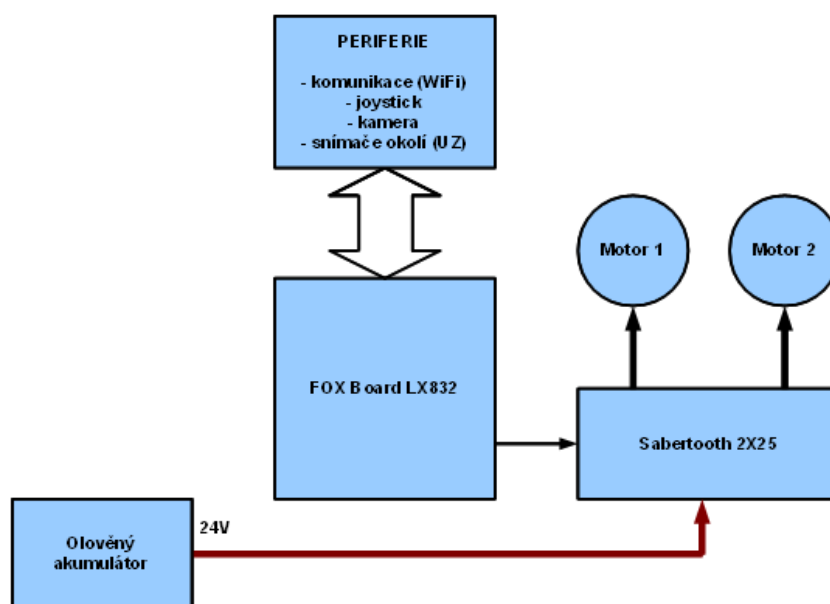


Obr. 2.2 blokové schéma elektroniky vozíku SPACE 1

## 2.2 UPRAVENÝ VOZÍK

Pro účely autonomního řízení vozíku je nutné mít přístup k ovládání jednotlivých součástí vozíku, především k motorům. Řídicí jednotka, spolu s křížovým ovladačem, využívá neveřejný komunikační protokol, nepodařilo se jej rozpoznat. Řídicí jednotku bylo nutné vyměnit. V úvahu přichází mnoho profesionálních modulů, použitelných k tomuto účelu, např. MAXON MOTOR ADS-50-10, PiKRON MARS2. Žádná z těchto jednotek však nedokáže kompletně nahradit původní, neposledním parametrem je cena takových zařízení. Nakonec byl zvolen samotný budič motorů Sabertooth 2X25 [3]. Řídicí, logickou, jednotku zastupuje Single Board Computer FOX Board LX832 [4].

Toto řešení umožňuje implementaci řídicího programu a řízení motorů za velice příznivou cenu. Vysoký výkon FOX Boardu umožňuje zpracování obrazu, vyhodnocování signálů snímačů a další komunikaci s vnějším prostředím. Neumožňuje však spínání světel a brzd vozíku. Stejně tak chybí zpětná vazba z motorů.



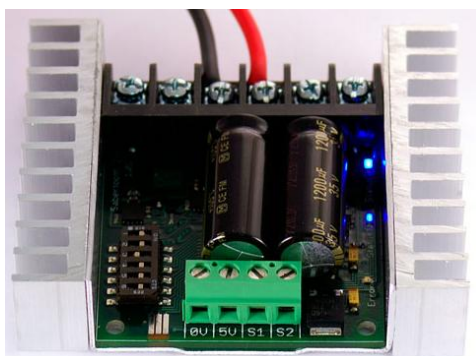
**Obr. 2.3** blokové schéma elektroniky upraveného vozíku

### 2.2.1 Sabertooth 2X25

Sabertooth 2x25 [3] je výrobkem firmy Dimension Engineering, jedná se o dvoukanálový budič stejnosměrných elektromotorů s trvalým výstupním proudem až 25A při napětí až 30V. Obsahuje nadproudovou i tepelnou ochranu, lineárně stabilizovaný 5V výstup a funkci rekuperace.

Nabízí široké spektrum možného ovládní, pomocí analogového vstupu 0-5V, výstupem standardního RC přijímače, jednoduché sériové rozhraní a paketizované sériové rozhraní.

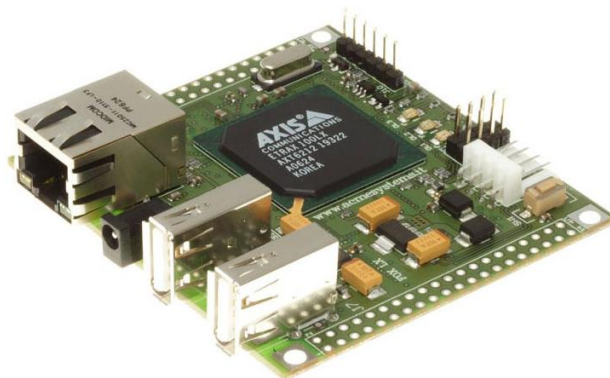
Motory jsou buzeny PWM signálem.



**Obr. 2.4** Sabertooth 2x25 [3]

### 2.2.2 FOX Board LX832

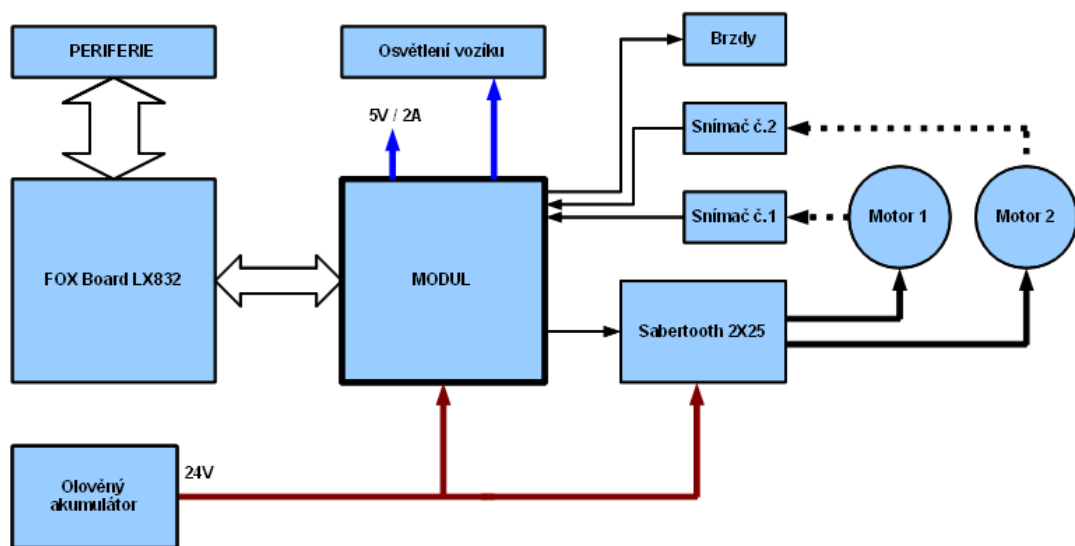
FOX Board LX832 [4] nabízí italská firma ACME SYSTEMS s. r. l.. Jedná se o embedded systém s operačním systémem Linux. Srdcem je 100MIPS RISC CPU ETRAX 100LX firmy Axis Communications. Při rozměrech 66 x 72mm a váze pouze 37g umožňuje připojení přes Ethernet, USB (Host) a další rozhraní vyvedená na vstupně výstupní porty (SPI, I2C, sériové rozhraní). Možnosti lze rozšířit připojením rozšiřující desky [5] vybavené čtečkou MMC/SD karet, rozhraním RS485, obvodem hodin reálného času a přípojným místem pro bezdrátový modul Maxstream XBee standardu ZigBee.



Obr. 2.5 FOX Board LX832 [4]

### 2.3 PŘIDÁNÍ MODULU ODOMETRIE

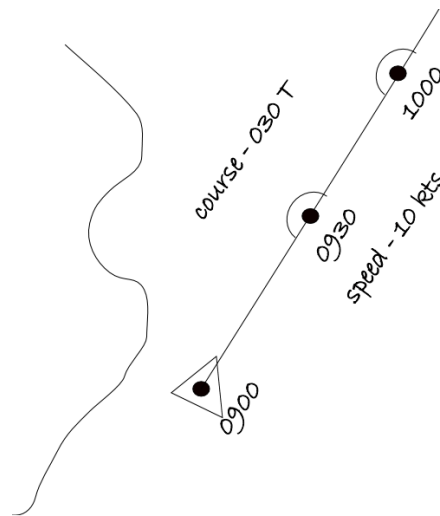
Pro přidání dalších funkcí, především odometrie se ukázalo jako nezbytné vytvoření samostatného modulu spojujícího nadřazený systém (FOX Board) se zbytkem vozíku. Tím je umožněn real-time výpočet odometrie zcela nezávisle na ostatních systémech. V návrhu je počítáno s možností spínat všechny výkonové části vozíku, jako jsou např. světlá a brzdy. Snímače natočení také zajišťují zpětnou vazbu pro regulátory otáček motorů.



Obr. 2.6 blokové schéma elektroniky vozíku s modulem odometrie

### 3. ODOMETRIE

Kořeny odometrie sahají až do 15. století A.D. Tehdejší mořeplavci využívali jako hlavní systém řízení plavby tzv. dead reckoning [6]. Výpočet polohy vycházel ze znalosti startovní pozice, azimutu (slunce, hvězdy, kompas) a rychlosti plavby (plovák / log), z těchto údajů je možné vypočítat přírůstky polohy a ty postupně integrovat. Výsledkem je odhad aktuální polohy.



**Obr. 3.1 princip dead reckoning [6]**

Odometrie spadá do kategorie dead reckoning systémů, nazývá se jí výpočet polohy především u pozemních vozidel. Také se zde využívá znalosti startovní polohy a znalosti změny polohy v čase. Pro výpočet změny polohy je nutná znalost matematického modelu vozidla a vhodně umístěné snímače.

Výhodou odometrie je, proti jiným způsobům navigace, přesnost v malých měřítcích, kdy nedochází k významnému integrování chyby. Vzhledem k principu měření umožňuje okamžitou odezvu a tím rychlou aktualizaci polohy. V neposlední řadě, přidání odometrie znamená, ve většině případů, minimální zásah do konstrukce vozidla, z toho plynou nízké náklady, což je v mnoha případech rozhodující faktor.

### 3.1 CHYBY ODOMETRIE A JEJICH ŘEŠENÍ

Integrace jednotlivých přírůstků dráhy sebou přináší podstatný nedostatek, každá, i drobná, chyba ovlivní každý další výpočet [7]. Odchylka vypočtené a skutečné pozice tak s časem může exponenciálně růst.

Chyby mohou být systematické, např.:

- rozdílný průměr kol
- vychýlení kol (kola nemají společnou osu)
- konečná frekvence vzorkování a výpočtu
- nepřesnost výpočtu
- kola se dotýkají na nenulové ploše

Nebo nahodilé:

- prokluz kol
- vnější vlivy

Znalost těchto chyb umožňuje jejich potlačení. Použijeme-li pro výpočet polohy pouze odometrii s požadavkem vysoké dlouhodobé přesnosti, je nutné k tomu přihlížet již při návrhu systému.

Geometrické nepřesnosti podvozku můžeme potlačit precizní výrobou, případně jejich změřením a započtením do matematického modelu. Dotyk kol na nenulové ploše působí náhodné drobné změny geometrických parametrů podvozku, především bod dotyku kola. Při nenulovém rozdílu rychlostí kol, při zatáčení, se imaginární bod dotyku kola náhodně pohybuje po celé ploše dotyku, to zanáší další chybu, která se samozřejmě integruje. Prokluz kol podává falešné informace o ujeté dráze, lze jej potlačit omezením akcelerace vozidla, tím však zhoršujeme parametry podvozku. Elegantním řešením prokluzu a dotyku kola je oddělení pohonných kol od měřicích. Pohonná kola se mohou dotýkat na dostatečné ploše umožňující potřebnou akceleraci, zatímco měřicí kola nebudou přenášet žádný výkon, nebude hrozit prokluz, zároveň mohou být velice tenká, pak je plocha dotyku téměř nulová.

Přesnost ovlivňují, mimo již zmíněných chyb, také rozměry vozidla, především vzdálenost (rozchod) kol (u diferenciálně řízeného podvozku), malý rozchod kol násobí chyby konstrukce, nejvíce je zasažen výpočet změny úhlu.

Pro zvýšení dlouhodobé přesnosti, odstranění integrující se chyby, je odometrie doplňována o další způsoby navigace [7], dle určení vozidla a požadovaných vlastností navigačního aparátu. Vhodné jsou především absolutní snímače polohy (triangulace pomocí radiomajáků, GPS, rozpoznávání pozemních značek), snímače založené na měření vzdálenosti (optické / akustické měření vzdálenosti od překážek), zpracování obrazu (z jednoho, případně více zdrojů – stereoskopické měření), případně, pro velmi malé vzdálenosti, taktilní senzory.

Spojením více snímačů, ideálně s rozdílným způsobem měření, získáme výhodu v možné filtraci hodnot ze snímačů podle jejich vhodnosti v daném okamžiku. Např. při spojení odometrie s GPS lze podle známé aktuální přesnosti GPS určovat jaký důraz bude kladen na hodnoty poskytnuté GPS a jaký důraz na hodnoty poskytnuté odometrií. Tento systém je v praxi často využíván, při výpadku signálu GPS je aktuální pozice počítána pomocí odometrie, či inerciální navigace.

Některé systematické chyby odometrie (typicky rozdílný průměr kol, parametry podvozku) lze potlačit průběžnou kalibrací pomocí dat z jiného snímače.

### 3.2 APLIKACE ODOMETRIE

Návrh výpočtu odometrie začíná u modelu podvozku. Využívá se kinematického nebo dynamického modelu.

Kinematický model podvozku vychází z ideálních podmínek, nepočítá s fyzikálními vlastnostmi podvozku. Kola se musí valit bez prokluzu i bez smyku. Musí platit podmínka společného středu otáčení. Výhodou kinematického modelu je jednoduchost. Umožňuje snadnou implementaci na neznámý podvozek.

Dynamický model počítá s fyzikálními parametry podvozku [8]. Je vhodný pro použití v prostředí, kde není možné zanedbat fyzikální působení. Složitě matematické modely umožňují kompenzovat především jevy způsobené nenulovou hmotností podvozku. Nevýhoda je zřejmá, s rostoucím počtem kompenzovaných

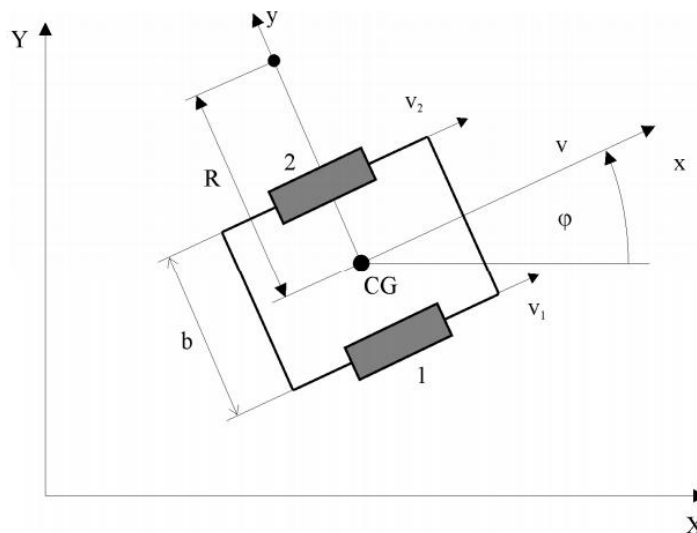


vlastností roste výpočetní náročnost. Proto je nutné zvolit model dle použití podvozku.

V našem případě je použitý kinematický model diferenciálního podvozku.

### 3.3 ODOMETRIE DIFERENCIÁLNÍHO PODVOZKU

Diferenciální podvozek [9] je nejjednodušší model pozemního vozidla schopného pohybu ve dvou osách. V ideálním případě je složený z dvojice kol o známém průměru v jedné ose se známým rozchodem. Snímače měří otočení obou kol, resp. dráhu ujetou jednotlivými koly.



**Obr. 3.2 model diferenciálního podvozku [8]**

Výpočet odometrie je zde, pro ideální kinematický model, velice jednoduchý. Známe-li parametry podvozku (průměr kol –  $d$ , rozchod kol –  $b$ ), předchozí polohu  $L_t$  (skládající se ze souřadnic  $X_t$ ,  $Y_t$  a úhlu/azimutu  $\varphi_t$ ) a rychlosti jednotlivých kol  $v_1$ ,  $v_2$ , resp. přímo dráhu ujetou jednotlivými koly  $s_1$ ,  $s_2$ , je možný výpočet odometrie.

V našem případě počítá polohu v reálném čase mikrokontrolér, je vhodný diskrétní výpočet. Počítá s vysokou frekvencí velmi malé přírůstky dráhy, díky tomu je možná aproximace dráhy přímku při zachování vysoké přesnosti.

Ze snímačů pravidelně získáme změnu natočení, kterou přepočteme na vzdálenosti ujeté jednotlivými koly –  $s_1$ ,  $s_2$ . Nejprve spočteme přírůstek vzdálenosti  $|\Delta L|$ , resp.  $|L_{t+1} - L_t|$  (3.1), kterou urazil střed osy kol, jeho hodnota je průměrem ujetých vzdáleností obou kol:

$$|\Delta L| = \frac{(s_1 + s_2)}{2} \quad (3.1)$$

Následně zjistíme změnu úhlu podvozku a nový úhel  $\varphi_{t+1}$ :

$$\Delta\varphi = \frac{(s_1 + s_2)}{b} \quad \varphi_{t+1} = \varphi_t + \Delta\varphi \quad (3.2)$$

Přírůstek v jednotlivých osách souřadného systému vypočteme goniometrickými funkcemi:

$$\Delta X = |\Delta L| \cdot \cos(\varphi_{t+1}) \quad \Delta Y = |\Delta L| \cdot \sin(\varphi_{t+1}) \quad (3.3)$$

Současnou polohu získáme přičtením přírůstků (3.3) k posledním známým souřadnicím:

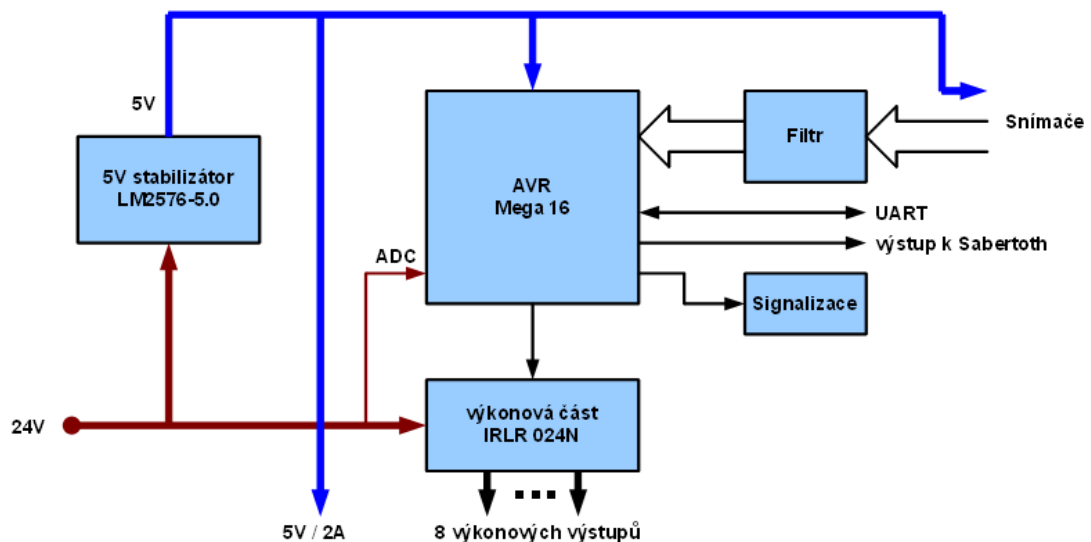
$$X_{T+1} = X_T + \Delta X \quad Y_{T+1} = Y_T + \Delta Y \quad (3.4)$$

## 4. NÁVRH

Základní myšlenkou je navrhnout zařízení, s následujícími funkcemi:

- zprostředkování komunikace s budičem motorů
- spínání výkonových částí
- odometrie
- napájení řídicích částí

Navrhl jsem zapojení vyhovující těmto podmínkám. Navíc obsahuje možnost regulace motorů na konstantní otáčky, resp. konstantní obvodovou rychlost kol. Další funkcí je např. měření palubního napětí vozíku.



Obr. 4.1 blokové schéma modulu odometrie

#### 4.1 STABILIZÁTOR

Stabilizátor je nutné dimenzovat k napájení FOX Boardu i periferií.

Požadavky na napájecí část jsou:

- výstupní napětí 5V
- výstupní proud až 2A
- vstupní napětí minimálně 30V

Vzhledem k modelářské povaze budiče motorů [3] stojí za úvahu použít BEC (battery elimination circuit) přímo v budiči, což je 5V stabilizovaný výstup. Pro použití lineárního stabilizátoru je však tato možnost nereálná. Při napájení 24V vychází potřebný úbytek napětí 19V, což při výstupním proudu 2A znamená ztrátový výkon 38W. Vzhledem k velikosti chladiče, je tato hodnota nereálná.

Byl zvolen samostatný spínaný stabilizátor napětí. Jako ideální se jeví řada stabilizátorů „simple switcher“, s minimem externích součástek dosahuje vyhovujících parametrů. Použitý je LM2576-5.0 od National Semiconductor [9].

Souhrn parametrů:

- výstupní napětí  $U_{OUT} = 5V$
- výstupní proud  $I_{OUT} = 3A$
- vstupní napětí až 40V
- spínací frekvence 52kHz

Zapojení odpovídá datasheetu [10]. Tlumivka je, kvůli dostupnosti, toroidní s průměrem 26mm o indukčnosti 330uH a maximálním proudem 2,5A, proto je výstupní proud omezen na 2A. Kapacita vstupního kondenzátoru je vypočtená ze vztahů uvedených v datasheetu, použit je LowESR hliníkový elektrolytický kondenzátor, vhodný pro podobné aplikace, s maximálním napětím 35V, což je hodnota dostačující i pro připojení nabíječky 24V Pb akumulátorů. Výstupní kondenzátor je, dle poznámky v datasheetu, jako desetinásobek minimální hodnoty. Je použita dvojice elektrolytických kondenzátorů, tantalový s kapacitou 100uF připojený co nejbližší k tlumivce a paralelně připojený hliníkový LowESR elektrolytický kondenzátor 1000μF pro minimální zvlnění výstupního napětí. Na

vstup je přidána dioda v propustném směru pro filtrování poklesu napětí vstupu, v případě krátkodobého poklesu napětí zabraňuje vybití kondenzátorů stabilizátoru některou z výkonových částí vozíku.

Schéma zapojení obsahuje příloha č. 1.

## 4.2 VÝKONOVÉ VÝSTUPY

Návrh výkonové spínací části závisí především na výběru vhodných výkonových tranzistorů. Pro tuto aplikaci jsou rozhodujícími parametry dostatečné maximální napětí  $U_{D-S}$  ( $>24V$ ), trvalý proud  $I_{D-S}$  ( $>2A$ ), co nejmenší odpor v sepnutém stavu a logic-level gate (pro plné otevření tranzistoru stačí napětí 3,5V). Dynamické parametry tranzistoru v této aplikaci nejsou podstatné.

Vhodným kandidátem je IRLR024N od International Rectifier [11].

Souhrn parametrů:

- Logic-Level Gate Drive
- maximální napětí  $U_{D-S} = 55V$
- odpor v sepnutém stavu  $R_{D-S ON} = 0.065\Omega$
- trvalý proud  $I_{D-S} = 17A$

Jedná se o N-FET tranzistor, je napojený Source elektrodou na zem. Mezi vstup a Gate je připojený ochranný rezistor s hodnotou 100  $\Omega$ , proti zemi vybíjecí rezistor s hodnotou 4k7 (tento rezistor má za účel vybíjet nahromaděný náboj na Gate při odpojení vstupu). Hodnoty rezistorů jsou vhodně zvoleny pro danou aplikaci. Proti proražení přechodu G-S přepětím je mezi zemí a Gate zenerova dioda. Napěťové špičky, vzniklé odpojením indukční zátěže, zachycuje ochranná dioda.

Schéma zapojení obsahuje příloha č. 1.

### 4.3 ŘÍDÍCÍ LOGIKA

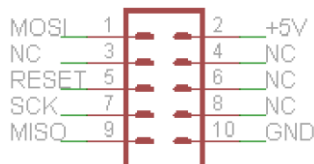
Řízení a výpočty obstarává mikrokontrolér ATmega16 firmy Atmel [12]. ATmega16 vyhovuje počtem vstupů/výstupů a dostatečnou připraveností z hlediska hardwarových periférií.

Souhrn parametrů:

- AVR® 8bit mikrokontrolér
- RISC architektura
  - 131 instrukcí – převážná část zpracovaná v jednom taktu
  - 32 x 8b universálních registrů
  - až 16 MIPS při kmitočtu 16 MHz
  - integrovaná 2 taktová násobička
- Paměti
  - 16 kB vnitřní programová FLASH
  - 512 B EEPROM
  - 1 kB SRAM
  - možnost uzamknutí paměti Lock bity
- Periferie
  - dva 8 bitové čítače/časovače s oddělenými děličkami
  - 16bitový čítač/časovač s oddělenou děličkou
  - Real Time čítač/časovač s možností externího oscilátoru
  - 4 PWM výstupy
  - 8kanálový, 10bit ADC
  - Two-wire Serial Interface
  - programovatelný USART
  - Master/Slave SPI sériové rozhraní
  - programovatelný Watchdog Timer s vlastním odděleným oscilátorem
  - integrovaný analogový komparátor

Zapojení samotného procesoru vychází z doporučení datasheetu [12].

Programování probíhá přes vestavěné SPI rozhraní, využívá se pinů MOSI, MISO, SCK a RESET vyvedených na 10pinový programovací konektor.



**Obr. 4.2** zapojení programovacího konektoru

Vstupy / výstupy procesoru jsou obsazeny následovně:

Port A:

Na pinu 1 a 2 jsou zapojeny signalizační LED, obě žluté barvy. Pin 4 (ADC4) slouží jako vstup AD převodníku. Měří se výstupní napětí děliče napojeného na napájecí napětí.

Port B:

Pin 2 (INT2) je vstupem externího přerušení, je použit pro STOP tlačítko. Piny 5 (MOSI), 6 (MISO), 7 (SCK) patří rozhraní SPI, jsou použity pro programování.

Port C:

Celý port C je využit pro ovládání výkonových výstupů.

Port D:

Piny 0 (RXD) a 1 (TXD) patří vestavěné sériovému rozhraní, slouží pro komunikaci s nadřazeným systémem. Piny 2 (INT0) a 3 (INT1) jsou vstupy externího přerušení, slouží k připojení snímačů (STEP signál). Piny 4 (OC1A) a 5 (OC1B) jsou výstupy čítače/časovače 1, slouží k připojení budiče motorů – generování RC signálu. Piny 6 a 7 slouží k připojení snímačů (DIR signál). Snímače jsou připojeny přes dolní propust pro potlačení vlivu rušení.

Schéma zapojení obsahuje příloha č. 5.

#### 4.4 SNÍMAČE

Základem snímačů je brzdový kotouček na hřídeli motoru opatřený osmi reflexními značkami. Pohyb značek snímá infračervený reflexní snímač GP2S60 od Sharp Microelectronics [13]. Autorem těchto snímačů je Ivo Amler, jsou obsahem jeho bakalářské práce [14].

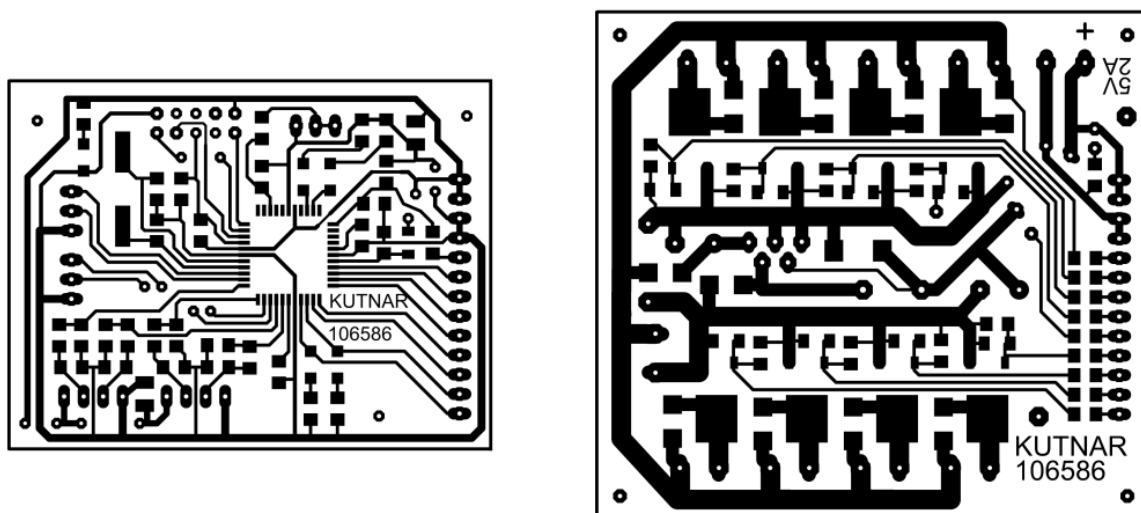


## 5. REALIZACE

Pro snadnou montáž je modul stejně velký jako FOX Board [4], je tedy možná montáž nad, případně pod desku FOX Boardu, při zachování jedné sady montážních otvorů. Vzhledem k množství součástek a jejich velikosti (výkonové tranzistory v pouzdře DPAK, diskrétní součástky velikosti 1206) je nutné zvolit řešení spočívající v rozdělní zapojení na výkonovou a řídicí část. Jednotlivé desky plošných spojů jsou umístěny nad sebou, spojeny konektorovou lištou. Spodní deska obsahuje výkonovou část, vstup napájení, stabilizátor, výstup 5V pro napájení ostatních periférií a 8 výkonových výstupů. Horní deska obsahuje procesor ATmega16, vstupy pro připojení snímačů, PWM výstup pro budič motorů, UART konektor, programovací konektor, signalizační LED a konektor pro připojení STOP tlačítka. Obě desky jsou rozebíratelně spojeny.

Výroba proběhla fotocestou. Jednotlivé předlohy desek byly vytištěny na průsvitný film a osvětleny na fotocitlivou vrstvu UVA svítidlem. Lept proběhl ve  $Fe_3Cl$ .

Vzory desek plošných spojů a osazovací plány jsou obsaženy v přílohách č. 2, 3, 4, 6, 7, 8. Soupiska použitých součástek je k dispozici v příloze č. 9 a 10.



Obr. 5.1 ukázka desek plošných spojů

## 5.1 ZMĚNY ZAPOJENÍ

Použití snímačů, jejichž výstup je tvořen otevřeným kolektorem, si vyžádalo drobnou změnu zapojení. Rezistory R5 až R8 byly nahrazeny hodnotou 2k2 a přepojeny jako pull-up (proti napájecímu napětí). Tato změna není zohledněna v dokumentaci, počítá se s použitím snímačů s aktivním výstupem.

Kapacitu kondenzátorů C11 až C14 je nutné volit s ohledem na maximální frekvenci výstupního signálu snímačů. Pro testovací účely jejich zapojení není nutné.

## 6. PROGRAM

### 6.1 KONFIGURACE PROCESORU

Nejhrubší konfigurace procesoru se provádí pomocí tzv. Fuse bitů, jejich podrobný popis je dostupný v datasheetu [12]. Pro tuto aplikaci je nastavení následující:

#### Lock Bits:

- No memory lock enabled
- No application protection
- No boot-loader protection

#### H Fuse:

- On Chip Debug disabled
- JTAG disabled
- SPI enabled
- Preserve EEPROM through chip erase enabled
- Boot Flash size 1024
- Boot Reset Vector disabled
- CKOPT=1 (disabled)

#### L Fuse:

- Brown-out level 2.7V
- Brown-out detector disabled
- High Freq Crystal/Resonator, 16CK+4ms

Ostatní konfigurace je programově přístupná pomocí registrů. Nastavení jednotlivých vstupně/výstupních portů je patrné z tabulky 6.1.

**Tab. 7.1 nastavení vstupně/výstupních portů**

| b | Port A |       | Port B |       | Port C |       | Port D |       |
|---|--------|-------|--------|-------|--------|-------|--------|-------|
|   | func   | state | func   | state | func   | state | func   | state |
| 0 | in     | T     | in     | T     | out    | 0     | in     | T     |
| 1 | out    | 0     | in     | T     | out    | 0     | in     | T     |
| 2 | out    | 0     | in     | P     | out    | 0     | in     | T     |
| 3 | in     | T     | in     | T     | out    | 0     | in     | T     |
| 4 | in     | T     | in     | T     | out    | 0     | out    | 0     |
| 5 | in     | T     | in     | T     | out    | 0     | out    | 0     |
| 6 | in     | T     | in     | T     | out    | 0     | in     | T     |
| 7 | in     | T     | in     | T     | out    | 0     | in     | T     |

Význam zkratk v tabulce 6.1:

- in – pin je konfigurován jako vstup
- out – pin je konfigurován jako výstup
- T – vstup, vypnutý pull-up rezistor
- P – vstup, zapnutý pull-up rezistor
- 0 – výstup, L úroveň
- 1 – výstup, H úroveň

Konfigurace čítačů odpovídá jejich funkci.

Čítač/časovač 0 tiká na frekvenci 62,5 kHz, přičemž vytváří 244 přerušení za sekundu, je určený ke kontrole komunikace.

Čítač/časovač 1 tiká na 2 000 kHz, má dvě hlavní funkce. Generuje PWM signál simulující výstup RC přijímače pro budič motorů Sabertooth. Zároveň vytváří 50 přerušení za sekundu, kde se provádějí všechny operace vyžadující konstantní periodu spouštění.

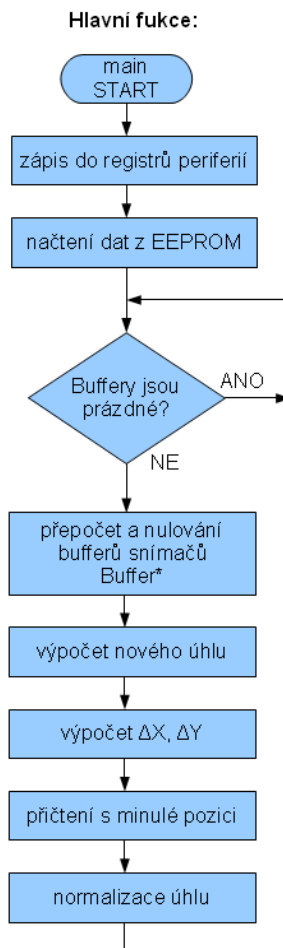
Vnější přerušení jsou použita jako vstupy STEP signálu snímačů (INT0, INT1) reagují na vzestupnou hranu. Vnější přerušení INT2 reaguje na sestupnou hranu, hlídá stisk STOP tlačítka.

Sériová linka UART pracuje s rychlostí 9600Bd s formátem 8 datových bitů, 1 stop bit, žádná parita. Je povolen vysílač i přijímač. Přijímač navíc vyvolá přerušení při příchodu bytu.

Analogově-digitální převodník s hodinovým kmitočtem 500 kHz je referenčním vstupem připojen na AVCC pin [12].

Ostatní periferie jsou v defaultním nastavení.

## 6.2 FUNKCE MAIN



**Obr. 6.1** vývojový diagram funkce main

Struktura hlavní funkce programu main je znázorněna na obrázku 6.1. Ihned po připojení napájení dojde k nastavení periférií kontroléru zápisem do registrů. Načtení parametrů vozíku uložených v EEPROM a jejich přepočet na přírůstek dráhy na jeden impuls, pro minimalizaci výpočtů v hlavním cyklu. Následuje nekonečná smyčka obsahující samotný výpočet odometrie. Toto řešení má výhodu v plném využití výkonu mikrokontroléru, kdykoliv není vykonávána jiná funkce, s maximální možnou frekvencí se provádí výpočet odometrie. Získáme tak maximální přesnost. Průměrná frekvence výpočtů je 2000 Hz.

Ukázka výpočtu odometrie:

```
Draha1 = Buffer1 * Prirustek_drahy1;           //vypocet jednotlivych drah
Draha2 = Buffer2 * Prirustek_drahy2;

Buffer1 = 0;                                  //nulovani bufferu pro dalsi plneni
Buffer2 = 0;

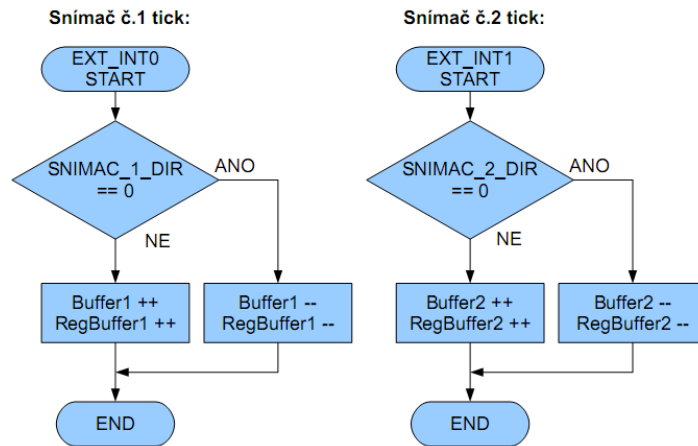
Souhlas = (Draha1 + Draha2) / 2.0;           //vypocet prumeru drah
Uhel = Uhel + (Draha2 - Draha1) / Vzdelenost_kol; //vypocet noveho uhlu

DeltaX = cos(Uhel) * Souhlas;                //delta X
DeltaY = sin(Uhel) * Souhlas;                //delta Y

X = X + DeltaX;                              //pricteni ke stavajicimu X a Y
Y = Y + DeltaY;
```

V každém cyklu se počítá dráha ujetá jednotlivými koly, jedná se o převod počtu impulsů na dráhu v mm. Nová hodnota úhlu je rovna původnímu úhlu s přičtením současné změny úhlu. Pomocí souhlasné složky obou ujetých drah (jejich průměr) a goniometrických funkcí získáme přírůstky souřadnic X a Y. Přičtením přírůstků k posledním známým souřadnicím získáme současnou pozici. Hodnota úhlu se normalizuje do rozmezí 0 až  $2\pi$ , je tedy vždy kladná. Výpočty probíhají, pro co nejlepší přesnost, s plovoucí desetinnou čárkou.

### 6.3 PŘERUŠENÍ OD SNÍMAČŮ



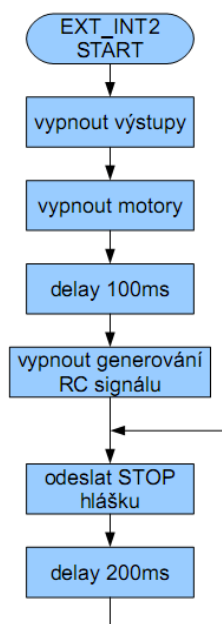
**Obr. 6.2** vývojový diagram přerušení od snímačů

Z důvodu použití inkrementálních snímačů, je nutné čítat každý jejich impuls. K tomu je využita funkce externího přerušení. Výstupem snímačů je dvojice fázově posunutých signálů, jsou programově dekodovány a podle zjištěného směru otočení dojde k přičtení, nebo odečtení k hodnotě bufferů. Buffer\* slouží k výpočtu odometrie, je čten s frekvencí přibližně 2000 Hz, RegBuffer\* je určený pro výpočet aktuální rychlosti, čte se s pevnou frekvencí 10 Hz.



## 6.4 STOP PŘERUŠENÍ

Stisk STOP tlačítka:



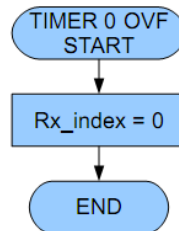
**Obr. 6.3** vývojový diagram STOP přerušení

Toto přerušení je vyvoláno po stisku STOP tlačítka, slouží k okamžitému vypnutí celého vozíku. Způsobí vypnutí všech výkonových výstupů a zastavení motorů. Následně odesílá v nekonečné smyčce chybovou hlášku.

Vzhledem k povaze STOP tlačítka, tedy poslední záchrany, nemůže nadřazený systém tento stav zrušit. Funkce se obnoví až po resetu mikrokontroléru, po odpojení a znovu připojení napájecího napětí. Tím je zajištěno, že ani nadřazený systém nemůže jít proti vůli uživatele zastavit.

## 6.5 PŘERUŠENÍ ČÍTAČE/ČASOVAČE 0

Přetečení timeru 0 ~4ms

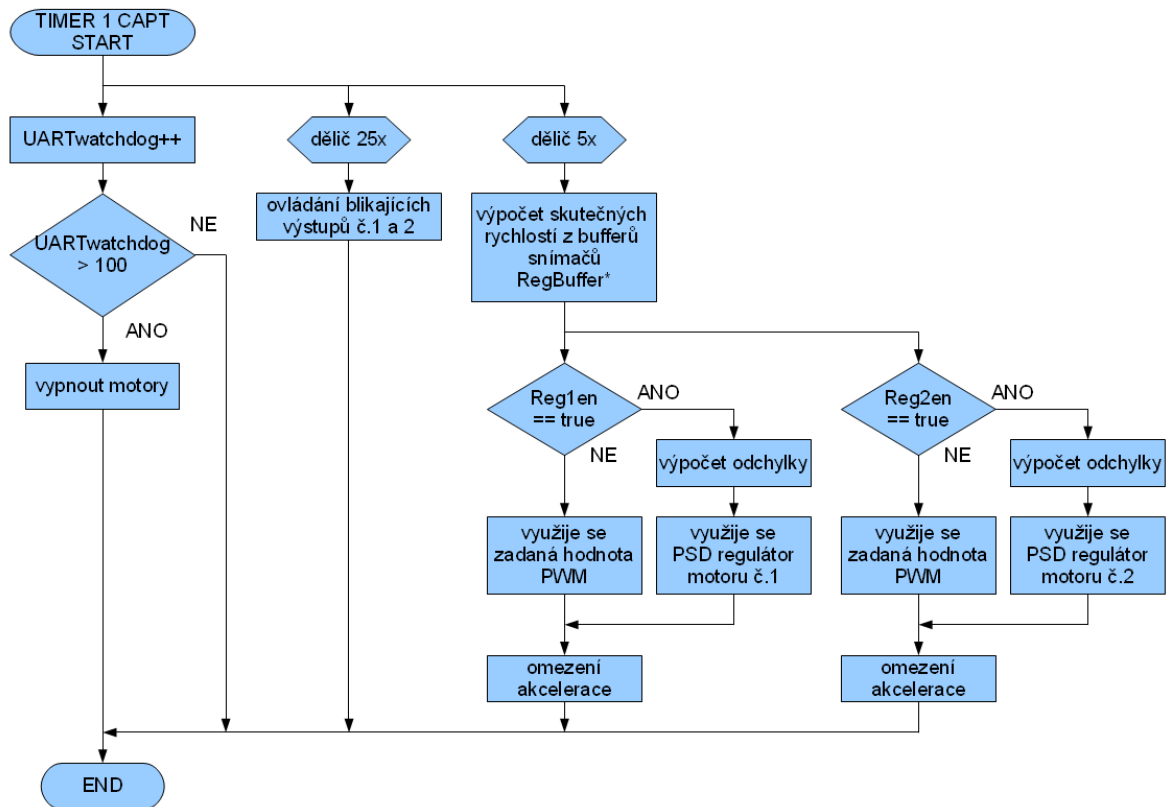


**Obr. 6.4** vývojový diagram přerušení čítače/časovače 0

Čítač/časovač 0 má jedinou funkci, hlídá časové prodlevy mezi příchozími bajty. Pokud je mezi bajty prodleva větší, než 4ms následující byte bude považován za první byte nového paketu.

## 6.6 PŘERUŠENÍ ČÍTAČE/ČASOVAČE 1

Přerušení timeru 1 20ms



Obr. 6.5 vývojový diagram přerušení čítače/časovače 1

Timer 1 má dvě hlavní funkce. První spočívá v generování RC signálu pro budič motorů (generuje se obdélníkový impuls s periodou 20ms a šířkou 1-2ms). Druhou funkcí je vykonávání všech činností vyžadující konstantní periodu spouštění.

Toho využívá UARTwatchdog, výstupy blinkrů a výpočty řízení/regulace motorů.

UARTwatchdog slouží k ošetření výpadku komunikace, pokud po nastavený čas nepřijde žádný platný příkaz, z bezpečnostních důvodů dojde k zastavení motorů.

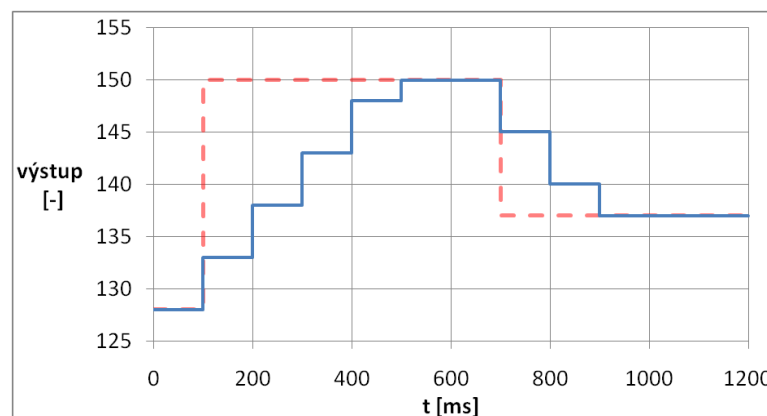
Výstupy č. 1 a 2 jsou určeny pro připojení směrovek – blinkrů, při zapnutí výstup střídá H a L úroveň s periodou 1s (500ms H, 500ms L).

S periodou 100ms je počítána aktuální rychlost obou kol. Pokud je povolený regulátor příslušného motoru, pak je motor regulován na konstantní rychlost PSD regulátorem. Pro rychlost výpočtu je zvolen celočíselný výpočet ve tvaru:

$$u(t) = \frac{KP \cdot e(t) + KI \cdot \text{sum}(e) + KD * (e(t) - e(t - 1))}{10} \quad (6.1)$$

Přesnost výpočtu na jedno desetinné místo je zajišťuje akční zásah dělený deseti, z tohoto důvodu je nutné zadat konstanty regulátoru KP, KI, KD desetkrát větší, to opět umožní zadání konstant v celočíselném tvaru s rozlišením na jedno desetinné místo, při rychlosti celočíselného výpočtu.

Se stejnou periodou se také spouští omezení akcelerace. V tomto případě není akcelerace přímo derivací rychlosti, ale derivací, resp. diferencí, změny střídy PWM modulace výstupu budiče motorů, převedené do rozsahu 0 až 255, přičemž hodnota 128 znamená stojící motor.



**Obr. 6.6 omezení akcelerace**

Obrázek 6.6 názorně ukazuje omezení akcelerace. Červená přerušovaná čára naznačuje požadovanou hodnotu, modrá pak výstup modulu k Sabertooth. V čase  $t=0$  je hodnota výstupu rovna 128, motor stojí, v čase  $t=100\text{ms}$  přijde požadavek na zvýšení rychlosti na hodnotu 150. Testy ukázaly, že takový skok spolehlivě aktivuje nadproudovou ochranu Sabertooth. Modul tedy nepošle ihned k budiči motorů plný zásah, ale postupně, s nastavenou rychlostí, integruje hodnotu, až se dostane k požadované. Jak je z grafu patrné, v tomto případě je omezení akcelerace nastaveno na 5, rozdíl současného a minulého zásahu je omezen na 5. V čase  $t=700\text{ms}$  přichází požadavek na skokové snížení rychlosti, modul opět postupně snižuje rychlost. Díky tomu je zajištěn plynulý rozjezd i brzdění. Strmost omezení akcelerace je samozřejmě možné změnit.

Ukázka PSD regulátoru:

```
RegSkutecna1 = RegBuffer1 * Prirustek_drahy * 10.0;      //vypocet rychlosti
RegBuffer1 = 0;
RegOdchylka1 = (int) RegRychlost1 - RegSkutecna1;      //vypocet odchylky

RegIntegral1 = RegIntegral1 + RegOdchylka1;           //integral odchylky

if (RegIntegral1 > 100) RegIntegral1 = 100;            //meze integralu
if (RegIntegral1 < -100) RegIntegral1 = -100;

RegZasah1 = (int) Kp*RegOdchylka1 + Ki*RegIntegral1 + Kd*(RegOdchylka1-
RegOdchylka1Minula);                                  // vypocet akcniho zasahu

RegOdchylka1Minula = RegOdchylka1;                   // uchovani minule odchylky

RegZasah1 = (int) RegZasah1/10;                        // zasah/10 kvuli presnosti int
RegZasah1 = RegZasah1 +127;                            // offset 127
```

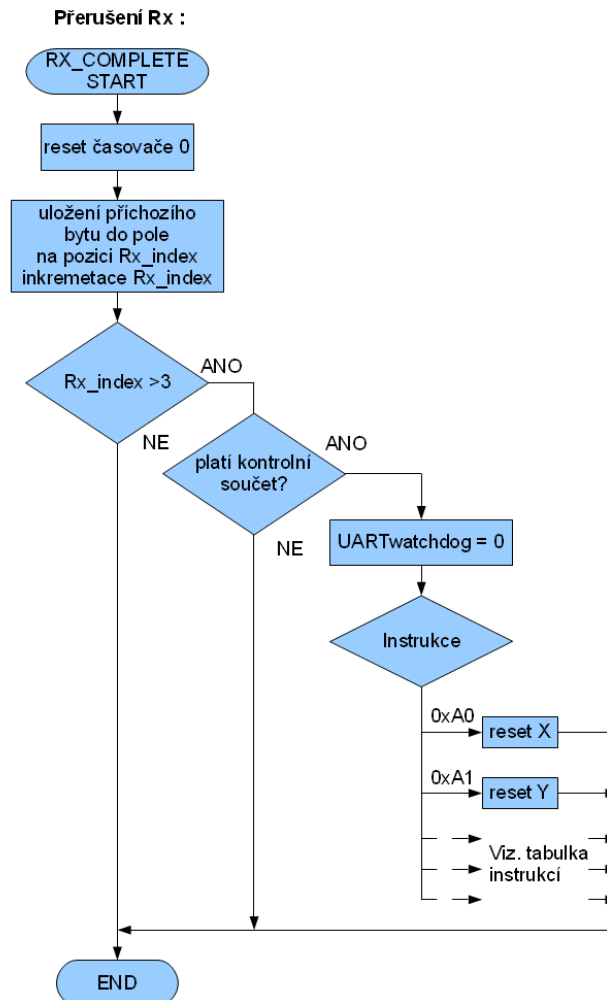
Ukázka omezení akcelerace:

```
if (RegZasah1 > (Rychlost1 + RegAcc)) RegZasah1 = Rychlost1 + RegAcc;
if (RegZasah1 < (Rychlost1 - RegAcc)) RegZasah1 = Rychlost1 - RegAcc;
// omezeni akcelerace pokud je zmena vetsi/mensi nez meze

if (RegZasah1 > 255) RegZasah1 = 255;                 // meze zasahu
if (RegZasah1 < 0) RegZasah1 = 0;

Rychlost1 = RegZasah1;
OCR1A=(unsigned int) Rychlost1 * 7.8 + 2000; //prepocteni do registru časovače
```

## 6.7 PŘERUŠENÍ PŘIJÍMAČE SÉRIOVÉ LINKY



**Obr. 6.7** vývojový diagram přerušení sériové linky

V tomto přerušení se obstarává veškerá komunikace s nadřazeným systémem, každý příchozí byte vyvolá toto přerušení. Jako první se resetuje timer 0 obstarávající nulování indexu pole příchozích dat, tím je zajištěno, že další příchozí byte (v intervalu kratším, než 4ms) nebude uložen opět na index 0. Následně dojde k uložení příchozího byte a inkrementování indexu pole příchozích dat.

S příchodem čtvrtého byte se ověří platnost kontrolního součtu. Pokud kontrolní součet odpovídá, vynuluje se UARTwatchdog (komunikace funguje) a provede se příslušná instrukce. Podrobný popis jednotlivých instrukcí je obsažen v kapitole Komunikace.

Ukázka ukládání, kontroly součtu a první instrukce:

```
TCNT0=0; // nulovat citac 0, prisel byte

Data_In[Rx_index]=UDR; // precist a ulozit data
Rx_index++; //zvednout index dat
if (Rx_index>3){ // kdyz se prijel cely ramec
    Soucet_In=Data_In[0]+Data_In[1]+Data_In[2];
    if(Soucet_In==Data_In[3]){ // kdyz sedi kontrolni soucet
        LED_IN=!LED_IN;
        // negovat LED Rx, signalizace komunikace
    }
    UARTwatchdog=0; //nulovani UARTw.

    switch (Data_In[0]){ // switch jednotlivych akci
        case 0xA0: // nastaveni souradnice X
            Pomoc=(Data_In[1]<8) + Data_In[2] - HALF_INT;
            X=Pomoc;
            XX=Pomoc;
            break;
        ...
    }
```

Ukázka instrukce 0xF0, odeslání polohových dat:

```
Pomoc=(unsigned int) X + HALF_INT;          //X souradnice, offset 7FFF  
Data_Out[0]= Pomoc>>8;  
Data_Out[1]= Pomoc;
```

```
Pomoc=(unsigned int) Y + HALF_INT;          //Y souradnice, offset 7FFF  
Data_Out[2]= Pomoc>>8;  
Data_Out[3]= Pomoc;
```

```
Pomoc=(unsigned int) Uhel;                  //Uhel v milirad  
Data_Out[4]= Pomoc>>8;  
Data_Out[5]= Pomoc;
```

```
Pomoc=(unsigned int) Napeti;               //Napeti v mV  
Data_Out[6]=Pomoc>>8;  
Data_Out[7]=Pomoc;
```

```
Data_Out[8]=OUTPUT;                        //stav vystupu  
Data_Out[9]=Rychlost1;                     // nastavena rychlost 1. motoru  
Data_Out[10]=Rychlost2;                   // nastavena rychlost 2. motoru
```

```
Data_Out[11]=Data_Out[0]+Data_Out[1]+Data_Out[2]+Data_Out[3]+Data_Out  
[4]+Data_Out[5]+Data_Out[6]+Data_Out[7]+Data_Out[8]+Data_Out[9]+Data_  
Out[10];                                   // kontrolni soucet
```

```
for(i=0;i<12;i++){                          // odeslani uartem  
UDR=Data_Out[i];  
while ( !( UCSRA & (1<<UDRE) ) );  
}
```



## 7. KOMUNIKACE

Komunikace s modulem probíhá sériovou linkou (UART). Konfigurace je 9600Bd, 8 datových bitů, žádná parita, 1 stop bit. TTL napěťové úrovně.

### 7.1 PŘÍCHOZÍ DATA

Protokol je navržen s důrazem na jednoduchost a snadnou implementaci do dalších zařízení. Počítá se s programovým ovládáním, nikoliv přes terminál, z tohoto důvodu není použito ASCII kódování.

Každá příchozí instrukce má délku 4B. V případě příchodu více bajtů se využijí první 4, jelikož se pravděpodobně jedná o chybu nadřazeného systému, nebude platný kontrolní součet a instrukce se nevykoná. V případě příchodu méně, než 4 bajtů se instrukce opět nevykoná.

Jednotlivé pakety jsou rozlišeny časovým úsekem o délce 4ms. Pokud tedy přijde neúplný paket, nemůže se stát, že by každý další paket byl, kvůli posunutí, neplatný. Po uplynutí 4ms od posledního bajtu je další příchozí byte uložen jako první byte nové instrukce. Z toho tedy vyplývá, že jednotlivé instrukce lze posílat s minimální periodou ~8ms (4x byte ~1ms + 4ms pauza). Instrukce se vykoná ihned po příchodu 4. bajtu, pokud bude platný kontrolní součet.

Formát instrukce je následující:

|               |                        |                        |                  |
|---------------|------------------------|------------------------|------------------|
| kód instrukce | vyšší datový byte - HB | nižší datový byte - LB | kontrolní součet |
|---------------|------------------------|------------------------|------------------|

Kód instrukce musí přijít jako první. Kontrolní součet je přetékačící součet ((kód instrukce) + HB + LB) výsledek je v mezích 0 až 255.

### 7.1.1 Tabulka instrukcí

Tabulka 7.1 tabulka instrukcí

| kategorie   | název   | kód            | funkce  | vstupní data                             |  |
|---|---|----------------|---|--|--|
| <b>Souřadnice</b>   | set X   | 0xA0           | nastaví X souřadnici  | HB<<8 + LB + 0x7FFF<br>[0x0000 - 0xFFFF] |  |
|   | set Y   | 0xA1           | nastaví Y souřadnici  | HB<<8 + LB + 0x7FFF<br>[0x0000 - 0xFFFF] |  |
|   | set úhel  | 0xA2           | nastaví úhel  | HB<<8 + LB<br>[0x0000 - 0xFFFF]          |  |
|   | reset všeho   | 0xA3           | reset všech souřadnic   | libovolná                                |  |
| <p><i>pozn.: souřadnice X a Y je nutné odeslat s offsetem 0x7FFF,<br/>např. pro X=5 odeslat jako 5+0x7FFF = 0x8004, úhel offset nemá, je vždy kladný</i></p>                      |   |                |   |  |  |
| <b>Spínání<br/>výstupů</b>  | všechny výstupy   | 0xB0           | sepne všechny výstupy dle<br>vstupních dat                    | LB [0x00 - 0xFF]                         |  |
|   | <p><i>pozn.: výstup č. 1 = LSB z LB ... výstup č.8 = MSB z LB</i></p>                       |                |   |  |  |
|   | výstup č. 1   | 0xB1           | sepne výstup dle hodnoty<br>vstupních dat                     | LB [==0 / !=0]                           |  |
|   | výstup č. 2   | 0xB2           |   | LB [==0 / !=0]                           |  |
|   | výstup č. 3   | 0xB3           |   | LB [==0 / !=0]                           |  |
|   | výstup č. 4   | 0xB4           |   | LB [==0 / !=0]                           |  |
|   | výstup č. 5   | 0xB5           |   | LB [==0 / !=0]                           |  |
|   | výstup č. 6   | 0xB6           |   | LB [==0 / !=0]                           |  |
|   | výstup č. 7   | 0xB7           |   | LB [==0 / !=0]                           |  |
| výstup č. 8   | 0xB8  | LB [==0 / !=0] |   |  |  |
| <p><i>pozn.: u jednotlivých výstupů platí 0x00 = vypnuto, cokoliv jiného = zapnuto<br/>pozn.: výstupy č. 1 a 2 blikají s periodou 1s (500ms H, 500ms L)</i></p>                   |   |                |   |  |  |
| <b>Řízení</b>   | motor č. 1 -<br>moment  | 0xC0           | hodnotu vstupních dat pošle<br>přímo na budič motorů          | LB [0x00 - 0xFF]                         |  |
|   | motor č. 2 -<br>moment  | 0xC1           |   | LB [0x00 - 0xFF]                         |  |
|   | <p><i>pozn.: 0x00 = plně vzad, 0x7F = stop, 0xFF plně vpřed, hodnoty jsou relativní</i></p> |                |   |  |  |
|   | motor č. 1 -<br>rychlost  | 0xC2           | hodnotu vstupních dat uloží<br>jako žádanou hodnotu rychlosti | HB<<8 + LB [0x0000 - 0xFFFF]             |  |
|   | motor č. 2 -<br>rychlost  | 0xC3           |   | HB<<8 + LB [0x0000 - 0xFFFF]             |  |
| <p><i>pozn.: offset 0x7FFF, tedy hodnoty &gt;offset = vpřed a opačně, hodnoty jsou v mm/s<br/>pozn.: řízení moment/rychlost se přepíná automaticky, platí poslední příkaz</i></p> |   |                |   |  |  |

| kategorie   | název   | kód   | funkce   | vstupní data                 |  |
|---|---|---|--|------------------------------|--|
| Nastavení   | průměr kola 1   | 0xE0  | nastaví hodnotu vstupních dat jako průměr kola 1           | HB<<8 + LB [0x0000 - 0xFFFF] |  |
|   | průměr kol 2  | 0xE1  | nastaví hodnotu vstupních dat jako průměr kola 2           | HB<<8 + LB [0x0000 - 0xFFFF] |  |
|   | rozchod kol   | 0xE2  | nastaví hodnotu vstupních dat jako rozchod kol             | HB<<8 + LB [0x0000 - 0xFFFF] |  |
|   | počet imp./ot.  | 0xE3  | nastaví hodnotu vstupních dat jako počet impulsů na otáčku | HB<<8 + LB [0x0000 - 0xFFFF] |  |
|   | přepočítat  | 0xE4  | přepočítá průměr, rozchod, počet imp.                      | libovolná                    |  |
|   | <i>pozn.: parametry je třeba přepočítat po každé změně parametrů vozíku</i>           |   |  |                              |  |
|   | akcelerace  | 0xE5  | nastaví hodnotu vstupních dat jako maximální akceleraci    | LB [0x00 - 0xFF]             |  |
|   | <i>pozn.: jedná se o relativní přírůstek momentu, využitelné jsou hodnoty 1 až 10</i> |   |  |                              |  |
|   | KP  | 0xE6  | nastaví hodnotu vstupních dat jako KP parametr regulátoru  | LB [0x00 - 0xFF]             |  |
|   | KD  | 0xE7  | nastaví hodnotu vstupních dat jako KD parametr regulátoru  | LB [0x00 - 0xFF]             |  |
| KI  | 0xE8  | nastaví hodnotu vstupních dat jako KI parametr regulátoru | LB [0x00 - 0xFF]   |                              |  |
| <i>pozn: parametry PSD regulátoru jsou 10x větší</i>                                      |   |   |  |                              |  |
| <i>pozn.: tedy <math>Y = (KP/10)*e + (KD/10)*derivace(e) + (KI/10)*integral(e)</math></i> |   |   |  |                              |  |
| Odeslání dat  | poloha  | 0xF0  | požadavek na odeslání polohových dat                       | libovolná                    |  |
|   | konfigurace   | 0xF1  | požadavek na odeslání konfiguračních dat                   | libovolná                    |  |
|   | regulátor   | 0xF2  | požadavek na odeslání dat regulátoru                       | libovolná                    |  |
| <i>pozn.: data se odesílají ihned po přijetí požadavku</i>                                |   |   |  |                              |  |

Tabulka 7.1 reprezentuje tabulku instrukcí modulu. Sloupec „kategorie“ označuje společné určení balíku instrukcí. Sloupec „název“ obsahuje název každé instrukce. Ve sloupci „kód“ je uveden kód instrukce, hodnota prvního byte v případě použití instrukce. Sloupec „funkce“ slovně popisuje význam každé instrukce. Sloupec „vstupní data“ zobrazuje význam použitých dat.

Mezní hodnoty jsou uvedeny v hranatých závorkách za formátem vstupních dat. Program nemá žádné vnitřní omezení, hodnota může být v mezích 0 až 0xFF pro instrukce využívající pouze dolní byte a 0 až 0xFFFF pro instrukce využívající oba byty. Je na obsluze zařízení aby zadala správná data, bez platných, realitě odpovídajících parametrů, by neměl výpočet odometrie žádný smysl.

### Vysvětlivky:

*vstupní data – libovolná* - znamená, že HB i LB mohou mít libovolnou hodnotu, instrukce nepotřebuje žádnou hodnotu, kontrolní součet však musí obsahovat HB i LB.

*offset* - nula je posunuta na střed pásma, tedy 0x7F v případě použití pouze LB nebo 0x7FFF v případě použití obou datových bytů, odesílá se vždy kladná hodnota, pro odeslání např. -0xAB je nutné odeslat (-0xAB + 0x7F), tedy 0xD4, modul vzápětí odečte 0x7F a získá chtěných -0xAB.

$HB \ll 8 + LB$  - znamená, že odesílaná hodnota má velikost 16b, horních 8b se odešle v HB, dolních 8b v LB, modul data spojí

$HB \ll 8 + LB + 0x7FFF$  – hodnota může dosahovat záporných hodnot, proto, před odesláním, je nutné připočíst offset o velikosti 0x7FFF

$LB [ = 0 / \neq 0 ]$  - v tomto případě modul posuzuje pouze hodnotu, je-li nulová, nebo není-li nulová.

### 7.1.2 Příklady instrukcí

Nastavit souřadnici X na hodnotu -50mm, po přičtení offsetu 0x7FFF získáme hodnotu 0x7FAF, rozdělíme na HB = 0x7F a LB = 0xAF.

příkaz: 0xA0 0x7F 0xAF 0xCE

Zapnout současně výstup č. 1, 5, 6, ostatní vypnout, tedy LB = 0b00110001.

příkaz: 0xB0 0x00 0x31 0xE1

Zapnout pouze výstup č. 3, ostatní neovlivnit,

LB != 0 (LB se nebude rovnat nule), např. LB = 0x12.

příkaz: *0xB3 0x00 0x12 0xC5*

Nastavit poloviční výkon vzad motoru č. 1, tedy  $LB = -0x40 + \text{offset } 0x7F = 0x3F$ .

příkaz: *0xC0 0x00 0x3F 0xFF*

Nastavit rychlost motoru č. 2 na 50mm/s,  $data = 50 + \text{offset } 0x7FFF = 0x8031$ ,  
rozdělím na  $HB = 0x80$  a  $LB = 0x31$ .

příkaz: *0xC3 0x80 0x31 0x74*

Nastavit průměr kol na 360mm, tedy  $data = 360$  (instrukce nemá offset) =  $0x0168$   
rozdělím na  $HB = 0x01$  a  $LB = 0x68$ .

příkaz: *0xE0 0x01 0x68 0x49*

## 7.2 ODCHOZÍ DATA

Odchozí data mají pevnou délku 12B. Stejně jako v případě příchozích dat, důraz je kladen na jednoduchost, paket nemá žádnou hlavičku, obsahuje pouze užitečná data. Identifikace paketu spočívá v odeslání správného požadavku na odeslání dat. Na odeslání žádosti o polohová data, modul zareaguje okamžitým odesláním polohových dat. Spojení více veličin do jednoho paketu je zvoleno z důvodu jejich podobnosti, zpravidla jsou potřeba všechny (nebo aspoň většina z nich) najednou.

Ukázka postupu přijmutí dat:

- vymažu příchozí buffer Rx nadřazeného systému
- odešlu instrukci (požadavek) pro příslušná data
- přijmu data (12 bajtů)
- ověřím platnost kontrolního součtu
- jednotlivé bajty spojím do proměnných

Modul odesílá 3 různé pakety dat.

Polohová data – obsahují nejpotřebnější veličiny pro nadřazený systém, polohu, hodnotu napájecího napětí, stav výstup a stav motorů.

Konfigurační data – slouží pro kontrolu nastavení parametrů vozíku.

Data regulátoru – pro kontrolu nastavení a funkce regulátoru.

### 7.2.1 Tabulka formátu odchozích dat

Tab. 8.2 tabulka formátu odchozích dat

| <b>Polohová data</b>           |                         |   |
|--------------------------------|-------------------------|---|
| požadavek: 0xF0 0x00 0x00 0xF0 |                         |   |
| byte                           | význam                  | využití   |
| 1.                             | horní byte souřadnice X | aktuální souřadnice X = (horní byte <<8) + dolní byte - 0x7FFF [mm]   |
| 2.                             | dolní byte souřadnice X |   |
| 3.                             | horní byte souřadnice Y | aktuální souřadnice Y = (horní byte <<8) + dolní byte - 0x7FFF [mm]   |
| 4.                             | dolní byte souřadnice Y |   |
| 5.                             | horní byte úhlu         | aktuální úhel = (horní byte <<8) + dolní byte [mili radian]   |
| 6.                             | dolní byte úhlu         |   |
| 7.                             | horní byte napětí       | aktuální napětí = (horní byte <<8) + dolní byte [mV]  |
| 8.                             | dolní byte napětí       |   |
| 9.                             | stav výstupů            | aktuální stav všech výstupů, LSB = výstup č. 1, MSB = výstup č. 8   |
| 10.                            | moment motoru č. 1      | aktuální hodnota momentu (PWM) příslušného motoru, offset 0x7F, tedy 0x00 = plně vzad, 0x7F = stop, 0xFF = plně vpřed |
| 11.                            | moment motoru č. 2      |   |
| 12.                            | kontrolní součet        | pro ověření platnosti dat   |

| <b>Konfigurační data</b>       |                            |   |
|--------------------------------|----------------------------|---|
| požadavek: 0xF1 0x00 0x00 0xF1 |                            |   |
| byte                           | význam                     | využití   |
| 1.                             | horní byte průměru kol     | průměr kola č. 1 (levé) = (horní byte <<8) + dolní byte [mm]                                  |
| 2.                             | dolní byte průměru kol     |   |
| 3.                             | horní byte průměru kol     | průměr kola č. 2 (pravé) = (horní byte <<8) + dolní byte [mm]                                 |
| 4.                             | dolní byte průměru kol     |   |
| 5.                             | horní byte rozchodu kol    | rozchod kol = (horní byte <<8) + dolní byte [mm]  |
| 6.                             | dolní byte rozchodu kol    |   |
| 7.                             | horní byte počtu imp./ot.  | počet impulsů na otáčku = (horní byte <<8) + dolní byte                                       |
| 8.                             | dolní byte počtu imp./ot.  |   |
| 9.                             | horní byte přírůstku dráhy | vypočtený přírůstek dráhy na imp. pro levé kolo = (horní byte <<8) + dolní byte [desetiny mm] |
| 10.                            | dolní byte přírůstku dráhy |   |
| 11.                            | nic                        | -   |
| 12.                            | kontrolní součet           | pro ověření platnosti dat   |

| Konfigurační data              |                           |  |
|--------------------------------|---------------------------|--|
| požadavek: 0xF1 0x00 0x00 0xF1 |                           |  |
| byte                           | význam                    | využití  |
| 1.                             | nastavená akcelerace      | kontrola nastavené akcelerace  |
| 2.                             | KP                        | kontrola nastavení PSD regulátoru (hodnoty jsou násobené 10x)  |
| 3.                             | KD                        |  |
| 4.                             | KI                        |  |
| 5.                             | horní byte rychlosti č. 1 | aktuální rychlost č. 1 = (horní byte <<8) + dolní byte - 0x7FFF [mm/s]   |
| 6.                             | dolní byte rychlosti č. 1 |  |
| 7.                             | horní byte rychlosti č. 2 | aktuální rychlost č. 2 = (horní byte <<8) + dolní byte - 0x7FFF [mm/s]   |
| 8.                             | dolní byte rychlosti č. 2 |  |
| 9.                             | nic                       | -  |
| 10.                            | nic                       | -  |
| 11.                            | regulace                  | stav regulátorů („0“ značí momentové řízení, „1“ regulaci, LSB pro motor č. 1 (levý) LSB+1 pro motor č. 2 (pravý)) |
| 12.                            | kontrolní součet          | pro ověření platnosti dat  |

Proměnné, které mohou dosahovat záporných hodnot, mají posunutou nulu (offset), 0x7F pro jednobajtové proměnné a 0x7FFF pro dvoubajtové, jak je patrné z tabulky.

Kontrolní součet je, stejně jako u příchozích dat, přetékačím součtem všech bajtů, výsledek je v mezích 0 až 255.

### 7.2.2 Příklad žádosti o polohová data

Přijmout aktuální souřadnici Y:

- vymažu příchozí buffer
  - odešlu instrukci pro polohová data, tedy 0xF0 0x00 0x00 0xF0
  - přijmu data (12 bajtů)
  - ověřím platnost kontrolního součtu
  - získám souřadnici Y složením horního a dolního bajtu souřadnice Y (3. a 4. byte) a odečtením offsetu
- $$Y = (3. B \ll 8) + 4.B - 0x7FFF \text{ [mm]}$$

### 7.3 PŘÍKLADY V C#

Nastavit poloviční výkon vzad motoru č. 1, tedy  $LB = -0x40 + \text{offset } 0x7F = 0x3F$ .

příkaz: *0xC0 0x00 0x3F 0xFF*

Realizace v C#:

```
motor1 = -0x40; //promenna obsahujici pozadovany vykon

vystup_byt[0] = 0xC0; //instrukce pro nastaveni vykonu motoru 1
vystup_byt[1] = 0x00; //LB,instrukce tento byte nevzuziva
vystup_byt[2] = (byte)(motor1 + 0x7F); //HB,pricteni offsetu
vystup_byt[3] = vystup_byt[0] + vystup_byt[1] + vystup_byt[2];
//kontrolni soucet

serialPort.Write(vystup_byt,0,4); //odeslani prikazu
```

Přijmout aktuální souřadnici Y.

Realizace v C#:

```
vystup_byt[0] = 0xF0; //instrukce pro odeslani polohy
vystup_byt[1] = 0x00; //LB, libovolny
vystup_byt[2] = 0x00; //HB, libovolny
vystup_byt[3] = vystup_byt[0] + vystup_byt[1] + vystup_byt[2];
//kontrolni soucet

serialPort.Write(vystup_byt,0,4); //odeslani prikazu

System.Threading.Thread.Sleep(20); //pockat 20ms na prijmuti
//mozno resit jakkoliv jinak

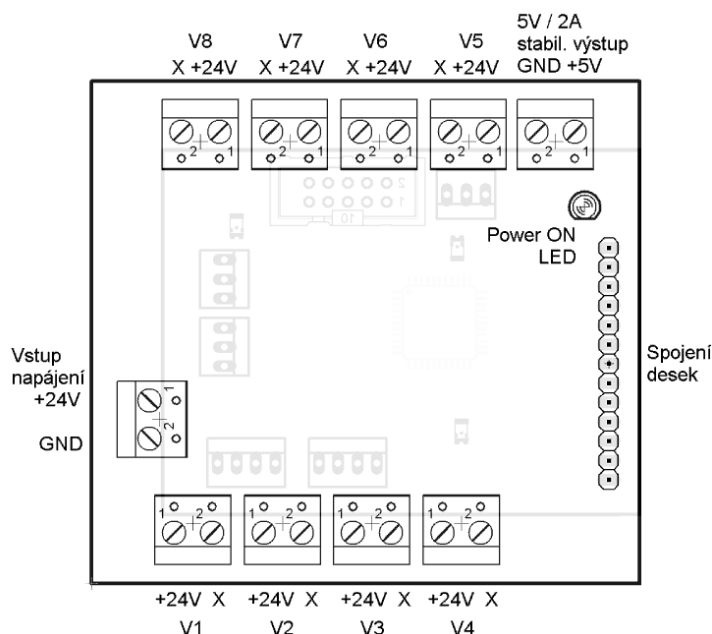
if (serialPort.BytesToRead == 12){ //pokud prislo vseh 12B
    serialPort.Read(vstup_byt, 0, 12); //nacist data
    kontrolni_soucet=(byte)(vstup_byt[0]+...+vstup_byt[10]);

    if (kontrolni_soucet==vstup_byt[11]){ //pokud plati kontrolni s.
        souradnice_Y=(vstup_byt[2]<<8)+vstup_byt[3];
        //vypocet souradnice (index pole zacina 0, proto posunuti o -1)
    }
}
```



## 8. ZAPOJENÍ

### 8.1 ZAPOJENÍ VÝKONOVÉ ČÁSTI



**Obr. 8.1** zapojení výkonové části

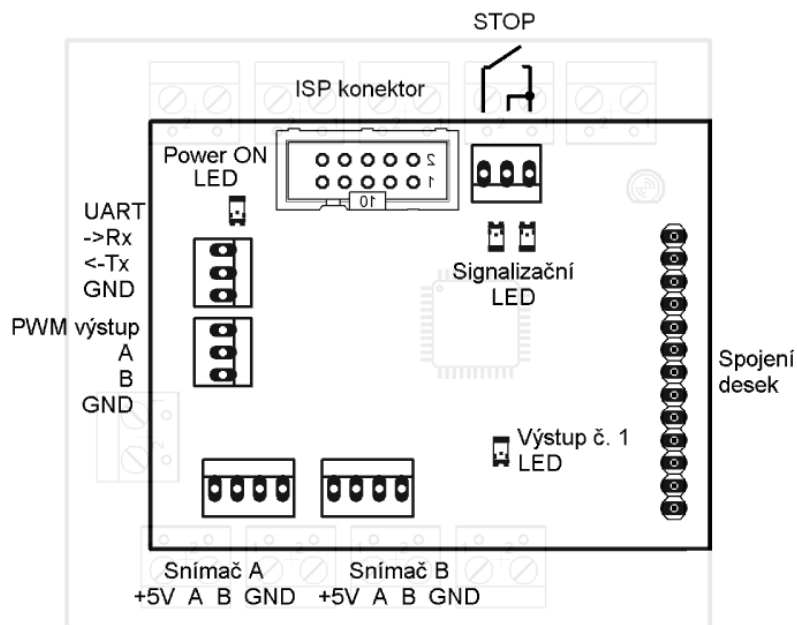
Zapojení modulu do systému je patrné z obrázku 8.1. Nezbytné, pro správnou funkci, je připojení napájecího napětí, svorkovnice, určená k tomuto účelu, je označena jako „Vstup napájení“. Napájecí napětí by mělo být v mezích 8 až 30V. Připojení napětí signalizuje svým svitem „Power ON LED“.

Pro napájení ostatních periférií vozíku je možné využít stabilizovaný výstup 5V označený jako „5V / 2A stabil. výstup“, maximální možné zatížení je 2A.

Výkonové části vozíku se připojují na spínané výstupy, označené jako „V1“ až „V8“. Výkonové tranzistory spínají svorku označenou jako „X“ k zemi (GND), svorka označená jako „+24V“ je pevně připojená ke kladnému pólu napájecího napětí. Maximální spínaný proud činí 17A, je však nutné přihlídnout k oteplení tranzistoru. Ochranné diody umožňují připojení indukční zátěže. Výstupy 1 a 2 slouží k připojení blinkrů.

Spojení s řídicí deskou zajišťuje 13pinová konektorová lišta označená jako „Spojení desek“.

## 8.2 ZAPOJENÍ ŘÍDICÍ ČÁSTI

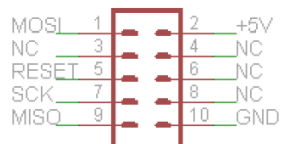


**Obr. 8.2** zapojení řídicí části

Na rozdíl od výkonové části, zde nejsou použity svorkovnice. Vhodný protikus k „ISP konektor“, je samořezný konektor na plochý kabel, dostupný jako PFL10. Jako protikusy ostatních konektorů poslouží prakticky libovolné konektory s roztečí 2,54mm.

Přítomnost napájecího napětí signalizuje „Power ON LED“. „Signalizační LED“ svým blikáním signalizují probíhající výpočet odometrie a platná příchozí data.

K programování mikrokontroléru slouží „ISP konektor“.



**Obr. 8.3** zapojení programovacího konektoru

Snímače se připojují ke konektorům označeným jako „Snímač A“ a „Snímač B“. Vnější piny konektoru poskytují napájecí napětí a GND, vnitřní piny jsou vstupy dvojice fázově posunutých obdélníkových signálů inkrementálního snímače.

Výstup modulu k řízení budiče motorů je vyveden na konektor „PWM výstup“. Zvolená je možnost řízení budiče motorů Sabertooth2x25 pomocí RC signálu [3]. Mikrokontrolér generuje dvojici RC signálů (piny označené jako „A“ a „B“), jedná se o dvojici obdélníkových signálů s periodou 20ms, šířka obdélníku se, dle žádané hodnoty, mění od 1ms do 2ms. Volba RC signálu umožňuje modul využít v širokém spektru aplikací, stačí si vybrat z nabídky regulátorů motorů, používanými modeláři.

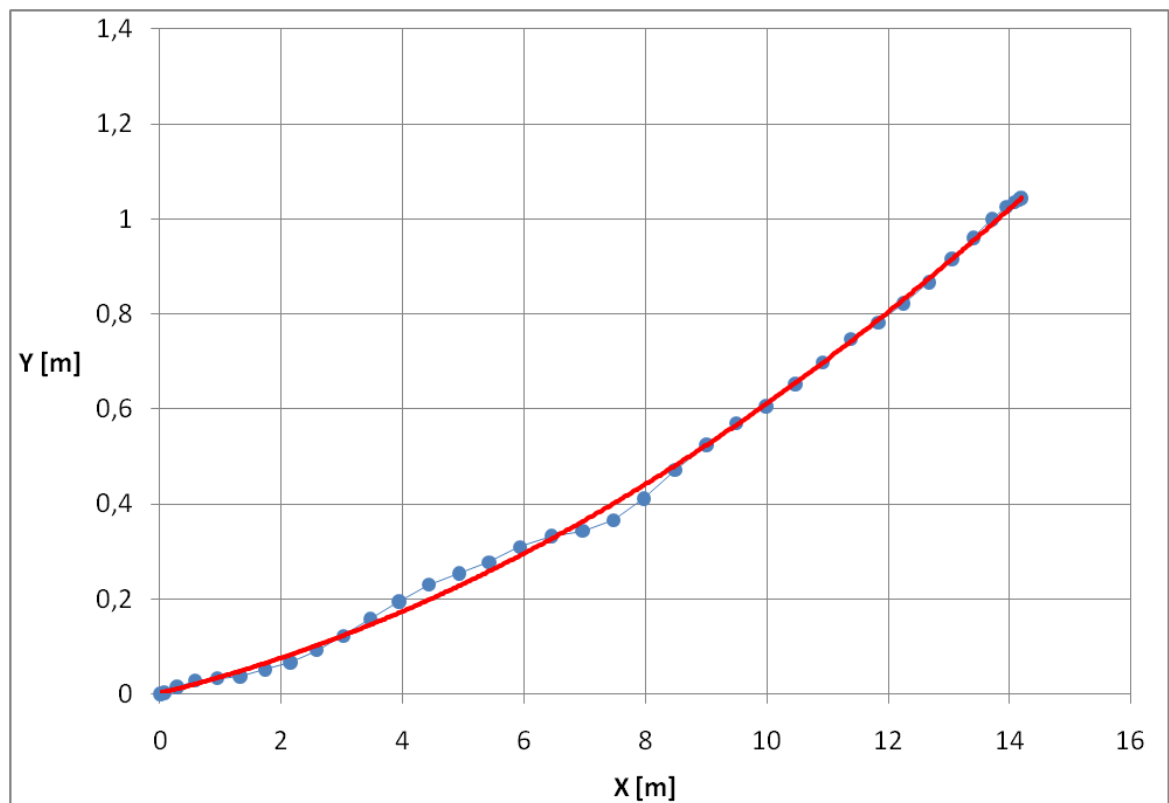
Spojení s nadřazeným systémem zajišťuje konektor „UART“, zde je vyvedeno komunikační rozhraní. Využívá se asynchronní přenos rychlostí 9600Bd s konfigurací 8 datových bitů, žádná parita, 1 stop bit.

Tlačítko sloužící k nouzovému zastavení se připojuje ke konektoru „STOP“, zapojení je patrné z obrázku 8.2. Po stisku dojde k okamžitému vypnutí všech výstupů modulu a zastavení motorů.

## 9. TESTY, MĚŘENÍ, LADĚNÍ

### 9.1 JEDEN PRŮMĚR KOL

První verze programu počítala odometrii s jedním průměrem pro obě kola. Velice rychle se ukázal tento postup jako nešťastný, vzduchem plněná kola vykazují při zatížení deformaci. Deformace způsobuje odvalování kola po menším průměru, než je změřený, tyto průměry jsou pro každé kolo jiné.



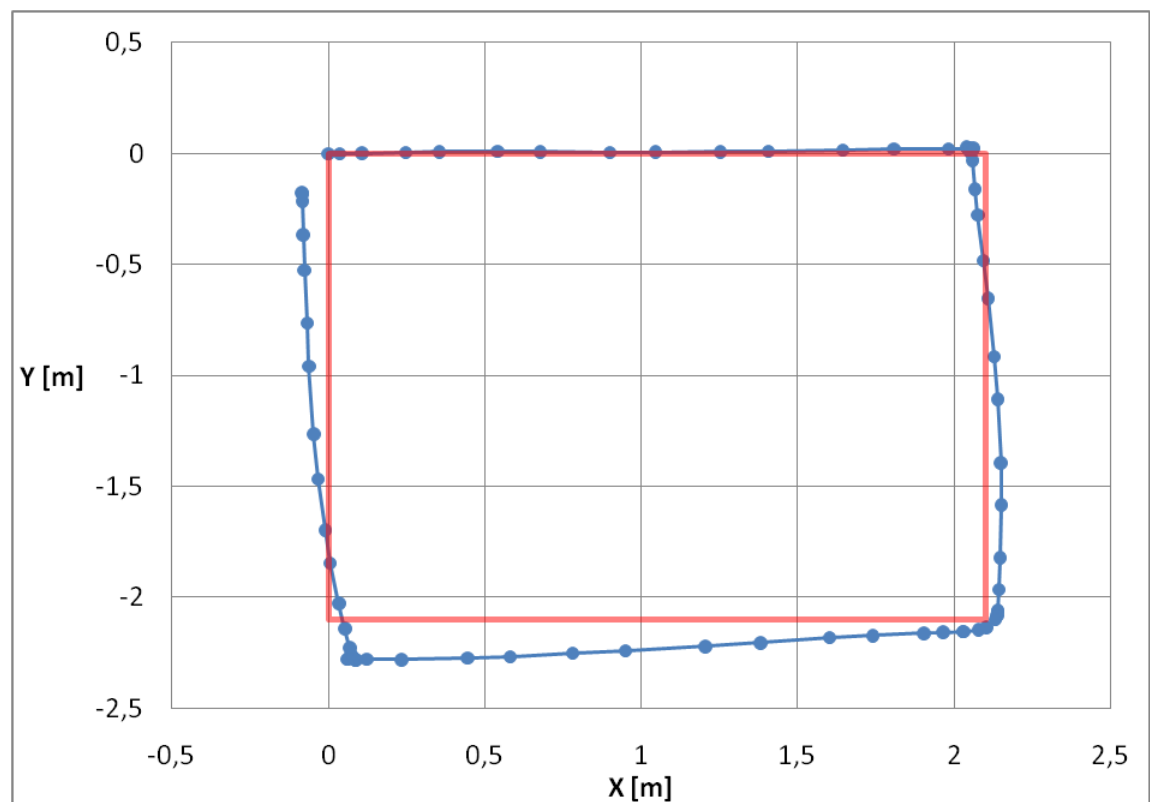
**Obr. 9.1 odometrie při použití jednoho průměru kol**

Pro první testy sloužila rovná dráha o délce 15m, výstupem odometrie by, v ideálním případě, měla být úsečka na ose X znázorňující délku 15m. Obrázek 8.1 znázorňuje jeden z výsledku měření. Je patrný vliv rozdílu průměru jednotlivých kol.

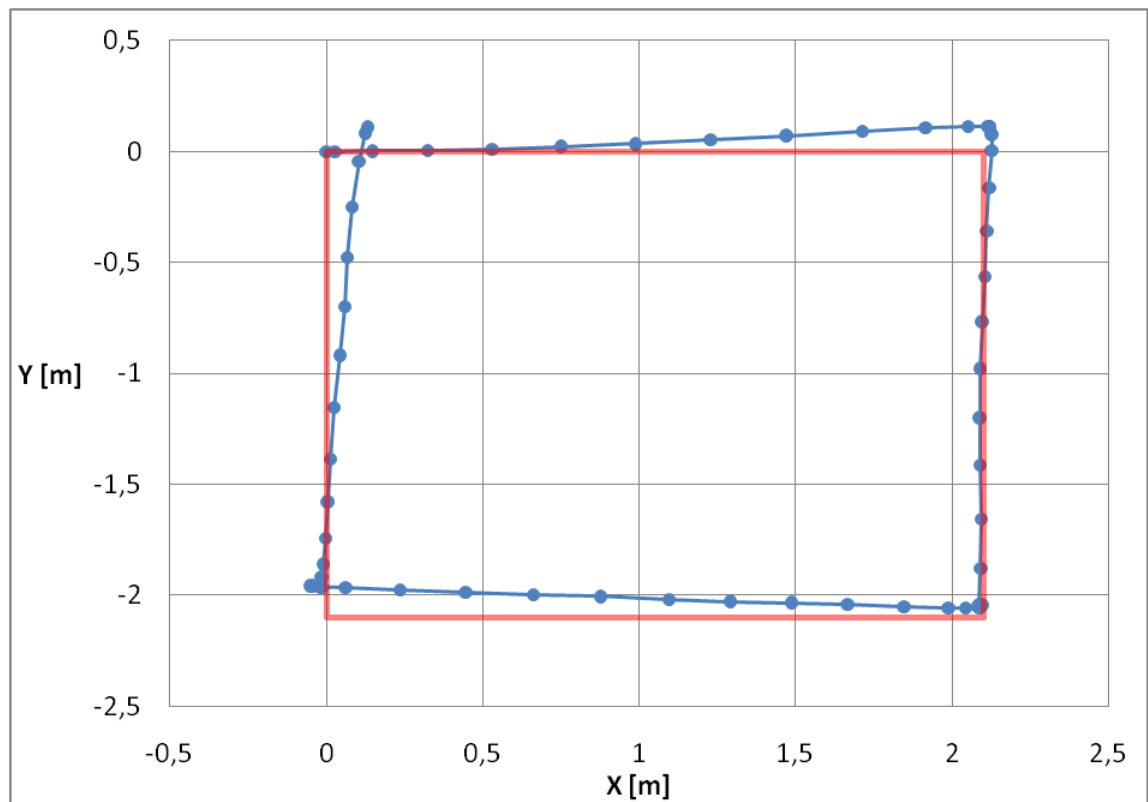
## 9.2 MODIFIKACE PRO DVA ROZDÍLNÉ PRŮMĚRY

Základní a hlavní úpravou je změna algoritmu tak, aby bylo možné zadat dva rozdílné průměry kol. Další úprava spočívá v přidání podmínky na nenulovou hodnotu bufferů snímačů. Jako zdroj integrující se chyby se ukázal výpočet goniometrických funkcí. Původní program počítal neustále i v případě, že nepřicházely žádné impulsy ze snímačů, každá nepřesnost výpočtu se tak integrovala i při stojícím vozíku. Současný program počítá pouze v případě příchodu impulsů.

Pro přesnější testování, především pro test přesnosti výpočtu úhlu, soužila dráha tvaru čtverce o straně 210 cm.

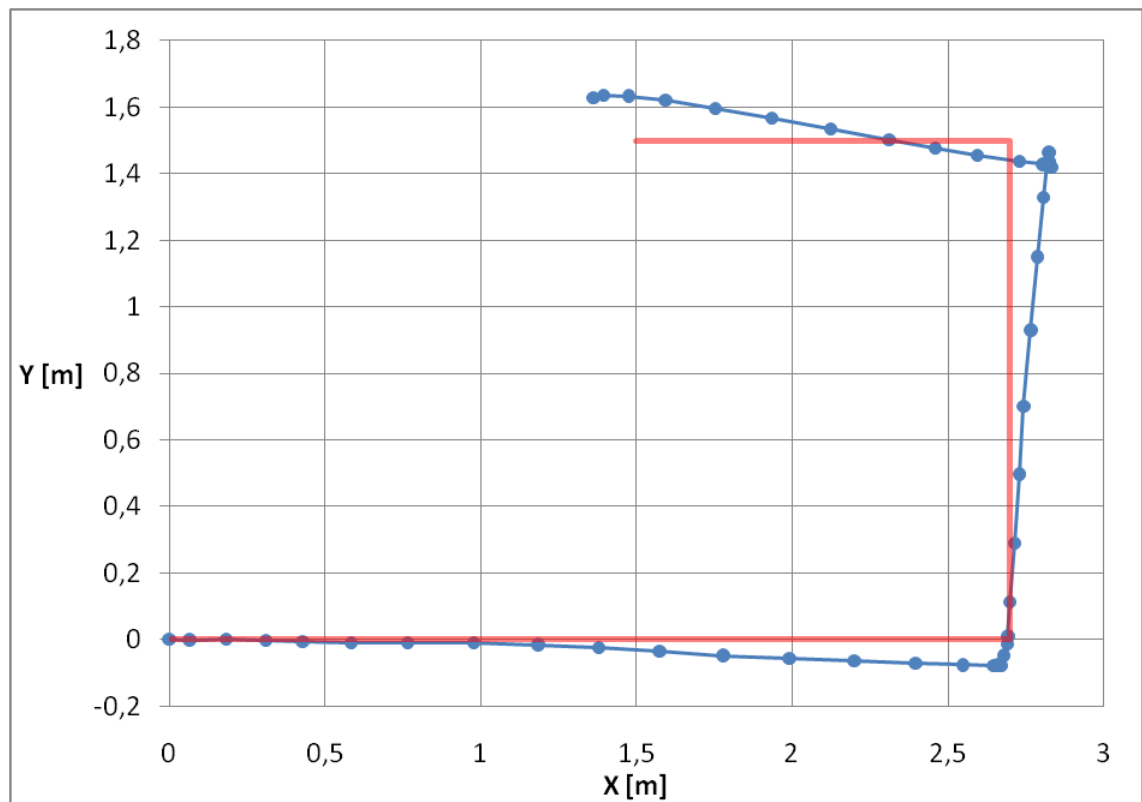


Obr. 9.2 výsledek upraveného výpočtu odometrie



**Obr. 9.3** výsledek upraveného výpočtu odometrie, další test

Červenou barvou je v obrázcích 9.2 a 9.3 znázorněna testovací dráha, modře pak výstup odometrie. Je patrné, že proti první verzi se podařilo systém odladit k nesrovnatelně lepším výsledkům. V této chvíli je největší chyba způsobená nepřesným tlačáním vozíku, zatáčky komplikují přední vlečná kola, která je nutné ručně rovnat.



**Obr. 9.4 simulace jízdy vozíku do zavazadlového prostoru**

Obrázek 9.3 zobrazuje simulaci dráhy, kterou musí vozík projet do zavazadlového prostoru automobilu.

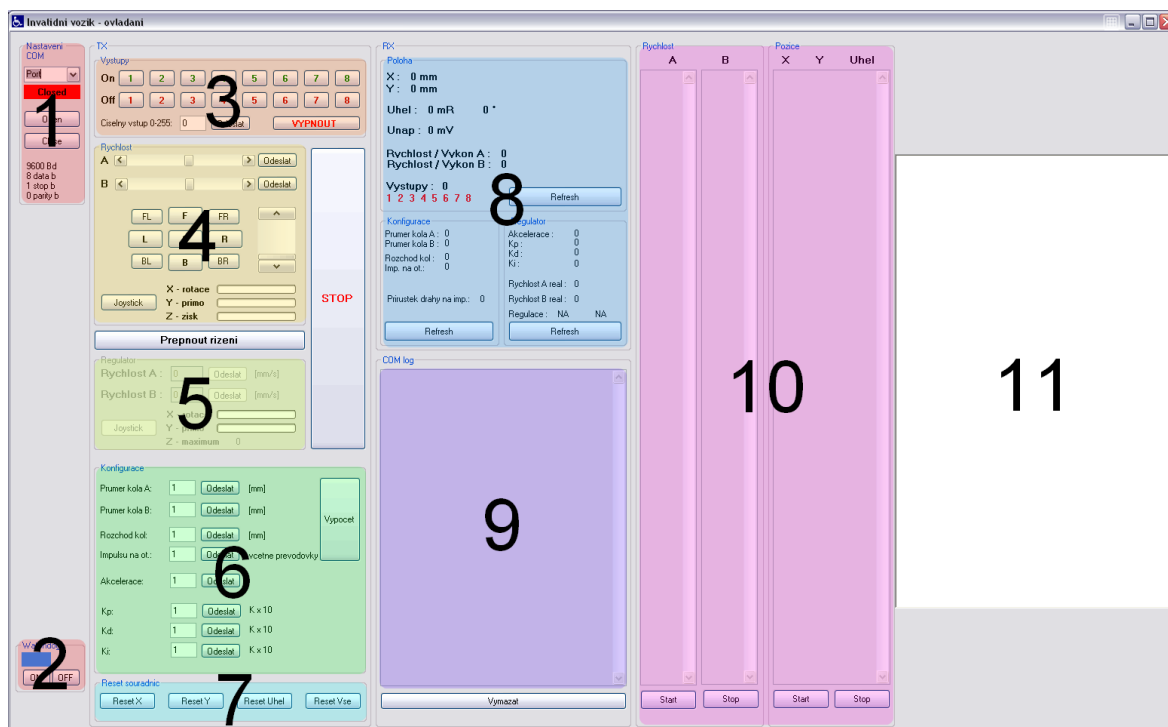
### 9.3 PARAMETRY VOZÍKU

Měření ukázala, že nejlepších výsledků je dosaženo při nastavení:

- průměr levého kola 358mm
- průměr pravého kola 362mm
- rozchod kol 595mm

Tyto hodnoty vzešly z jízdy po testovací dráze, zároveň se však prokázal vliv síly člověka tlačícího vozík na deformace pneumatik, případně prokluz. Nepodařilo se odometrii odladit tak, aby vykazovala identické výsledky při pravotočivých i levotočivých zatáčkách. Zmíněné parametry se mohou s časem měnit.

## 10. OVLÁDACÍ PROGRAM



Obr. 10.1 testovací program

Pro snadné ovládání modulu je vytvořen testovací program. Umožňuje kompletní ovládání modulu, celého vozíku. Dále obsahuje možnost záznamu aktuálních rychlostí kol pro ladění PSD regulátorů a záznam aktuální pozice pro nastavení parametrů odometrie. Aktuální pozice je v reálném čase vykreslována.

Popis jednotlivých částí uživatelského rozhraní:

- 1 – Nastavení komunikace – umožňuje uživateli vybrat port, který je připojený k modulu. Otevření se provede stiskem „Open“, uzavření stiskem „Close“.
- 2 – Watchdog – neustále odesílá data do modulu. Při výpadku dat, modul z bezpečnostních důvodů vypne motory. Funkce se automaticky aktivuje při otevření komunikačního portu, je možné dodatečné vypnutí či zanutí.



- 3 – Ovládání výstupů – umožňuje dálkově spínat výkonové výstupy modulu. Je možné ovládat každý výstup samostatně, případně všechny najednou zadáním číselné hodnoty.
- 4 – Momentové řízení – ovládá motory přímo změnou střídý PWM. Je možné řídit každý motor samostatně posuvníkem nebo osmisměrným tlačítky, kde svislý posuvník určuje zesílení. Další možností je připojení USB joysticku, aktivuje se stiskem „Joystick“, pro snadné a přesné řízení, osa Z určuje zesílení os X a Y.
- 5 – Rychlostní řízení – odešle nastavenou hodnotu do regulátorů motorů. Stejně jako v předchozím případě je možné připojit joystick, osa Z určuje měřítko os X a Y.
- 6 – Nastavení – slouží k nastavení parametrů vozíku pro výpočet odometrie a parametrů regulátorů. Při změně parametrů vozíku je nutné odeslat příkaz „Prepocitat“, tím dojde k přepočtu přírůstků dráhy na impuls.
- 7 – Reset souřadnic – stiskem příslušného tlačítka vynuluje souřadnice.
- 8 – Čtení hodnot – stiskem „Refresh“ stáhne z modulu příslušná data. Dojde k jejich okamžitému zobrazení.
- 9 – Log – uchovává obsah příchozí i odchozí komunikace, zároveň obsahuje důležitá hlášení.
- 10 – Nástroje k ladění – umožňuje sledování a uchování dat k další analýze. První dva sloupce, po stisku „Start“ vypisují aktuální rychlosti kol pro sledování chování regulátorů, hodnoty jsou v mm/s.  
Třetí sloupec, po stisku „Start“, vypisuje současnou vypočtenou pozici ve formátu X [mm] Y [mm]  $\varphi$  [mili radian]. Dráha je současně vykreslována v poli 11.
- 11 – Pole pro vykreslení ujeté dráhy. Aktivuje se současně se záznamem pozice.

## 11. ZÁVĚR

Cílem bakalářské práce bylo vytvoření modulu zprostředkovávajícího spojení nadřazeného systému se zbytkem vozíku. Na základě výzkumu a konzultací vznikl funkční celek, jenž plně zastupuje původní řídicí jednotku.

Výsledky odometrie plně odpovídají předpokladům. Projevuje se integrující charakter chyby omezující využití odometrie pro měření delších vzdáleností. Je nutná spolupráce s ostatními senzory, s čímž se také počítá. Důkladným testováním se podařilo odometrii odladit do takové fáze, kdy, pro malé vzdálenosti, zanáší největší chybu člověk nepřesným vedením po testovací dráze. Současné silové působení na vozík, má za následek deformaci pneumatik, případně drobný prokluz. Další vývoj může spočívat v nastavení parametrů PSD regulátorů pro odpovídající odezvu pohonu při zatížení vozíku všemi systémy autonomního řízení.

Modul není nijak vázaný na invalidní vozík, jistě najde široké uplatnění, lze jej použít na libovolném diferenciálním podvozku. RC signál, využitý k řízení motorů, umožňuje připojení prakticky libovolného regulátoru používaného modeláři po celém světě, výběr je obrovský. Inkrementální snímače jsou dnes běžné, jednoduchý ovládací protokol nabízí snadnou integraci do systému.

S výsledkem jsem spokojen. Práce pro mě byla přínosná v mnoha ohledech, zdokonalil jsem se v práci s mikrokontroléry AVR Mega, vývoji programu v jazyce C# a především v implementaci teoretických poznatků do praxe, kdy se naplno ukázala nezbytnost vývoje a ladění na základě získaných praktických zkušeností a výsledků testů.

Vytvořený modul poskytuje oporu pro další vývoj autonomního invalidního vozíku. Umožňuje pokračování výzkumu a dosažení vytyčeného cíle. Doufám v úspěšné dokončení projektu *Systém pro automatické nakládání vozíku pro tělesně postižené*, jehož je má práce součástí.

## 12. POUŽITÁ LITERATURA A ZDROJE

- [1] VASSILLI s. r. l. *VASSILLI s. r. l. – výrobce invalidních vozíků* [online]. [cit. 2010-04-28]. Dostupné z: <<http://www.vassilli.it/>>.
- [2] AMT Schmid GmbH & Co. KG. *Gearmotor SRG 01* [online]. [cit. 2010-04-28]. Dostupné z: <[http://www.amt-schmid.com/en/products/gearmotor\\_drives/srg\\_01/srg\\_01.php](http://www.amt-schmid.com/en/products/gearmotor_drives/srg_01/srg_01.php)>.
- [3] Dimension Engineering. *Sabertooth dual 25A motor driver* [online]. [cit. 2010-04-28]. Dostupné z: <<http://www.dimensionengineering.com/Sabertooth2X25.htm>>.
- [4] ACME SYSTEMS s. r. l. *FOX Board LX832* [online]. [cit. 2010-04-28]. Dostupné z: <<http://foxlx.acmesystems.it/>>.
- [5] ACME SYSTEMS s. r. l. *FOXZB XBee/RS485 add-on board* [online]. [cit. 2010-04-28]. Dostupné z: <<http://foxlx.acmesystems.it/?id=42>>.
- [6] *Dead reckoning* [online]. Poslední revize 24.04.2010 [cit. 2010-04-28]. Dostupné z: <[http://en.wikipedia.org/wiki/Dead\\_reckoning](http://en.wikipedia.org/wiki/Dead_reckoning)>.
- [7] DIXON, Jonathan, HENLICH, Oliver. *Mobile Robot Navigation*. Imperial College, London [online] 10.6.1997 [cit. 2010-04-28]. Dostupné z: <[http://www.atpjournal.sk/atpplus/archiv/2007\\_1/PDF/plus203\\_207.pdf](http://www.atpjournal.sk/atpplus/archiv/2007_1/PDF/plus203_207.pdf)>
- [8] ŠEMBERA, J., ŠOLC, F. *Modelování a řízení mobilního robotu s diferenciálním podvozkem*. AT&P journal [online] 2007 [cit. 2010-04-28]. Dostupné z: <[http://www.atpjournal.sk/atpplus/archiv/2007\\_1/PDF/plus203\\_207.pdf](http://www.atpjournal.sk/atpplus/archiv/2007_1/PDF/plus203_207.pdf)>

- [9] ŠOLC, František. *Modely kolových robotů*. FEKT Vysokého učení technického v Brně 4.4.2007 [cit. 2010-04-28]
- [10] National Semiconductor. *LM2576/LM2576HV Series* [online] Poslední revize 8.2004 [cit. 2010-04-28]. Dostupné z:  
<<http://www.national.com/ds/LM/LM2576.pdf>>
- [11] International Rectifier. *IRLR024N/IRLU024N* [online] Poslední revize 10.2.2000 [cit. 2010-04-28]. Dostupné z:  
<<http://www.irf.com/product-info/datasheets/data/irlr024n.pdf>>
- [12] Atmel Corporation. *ATmega16 /ATmega16L 8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash* [online] Poslední revize 5.2009 [cit. 2010-04-28]. Dostupné z:  
<[http://www.atmel.com/dyn/resources/prod\\_documents/doc2466.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf)>  
>
- [13] Sharp Microelectronics. *GP2S60* [online] Poslední revize 3.10.2009 [cit. 2010-04-28]. Dostupné z:  
<[http://document.sharpsma.com/files/GP2S60\\_DS.pdf](http://document.sharpsma.com/files/GP2S60_DS.pdf)>
- [14] AMLER, I. *Robot pro tělesně postižené*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 50 s. Vedoucí bakalářské práce prof. Ing. František Šolc, CSc.
- [15] VÁŇA, Vladimír. *Atmel AVR programování v jazyce C*. 1. vyd. BEN - technická literatura 2003, 216s. ISBN 80-7300-102-0.

- [16] BOLDIŠ, Petr. *Bibliografické citace dokumentů podle ČSN ISO 690 a ČSN ISO 690-2: Část 2 – Modely a příklady citací jednotlivých typů dokumentů*. Verze 3.0 (2004). © 1999-2004, poslední aktualizace 11.11.2004. Dostupné z:  
<<http://www.boldis.cz/citace/citace2.ps>>.  
<<http://www.boldis.cz/citace/citace2.pdf>>.
- [17] Microsoft. *Microsoft Developer Network* [online]. [cit. 2010-04-28]. Dostupné z: <<http://msdn.microsoft.com/cs-cz/default.aspx>>.

### 13. SEZNAM PŘÍLOH

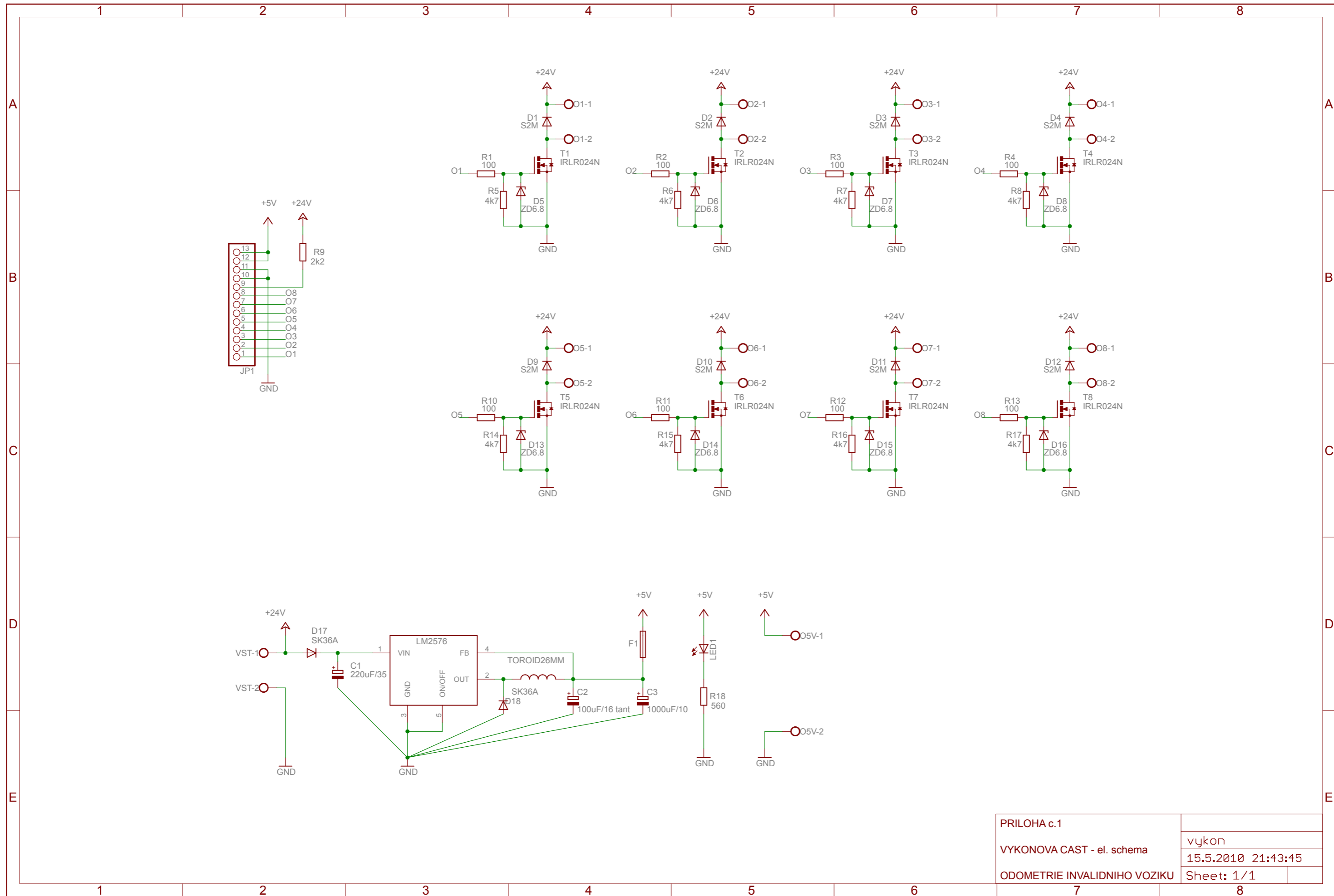
- Příloha č. 1 – schéma výkonové části modulu
- Příloha č. 2 – DPS výkonové části modulu, zobrazení spojů
- Příloha č. 3 – DPS výkonové části modulu, osazovací plán, strana spojů
- Příloha č. 4 – DPS výkonové části modulu, osazovací plán, strana součástek
- Příloha č. 5 – schéma řídicí části modulu
- Příloha č. 6 – DPS řídicí části modulu, zobrazení spojů
- Příloha č. 7 – DPS řídicí části modulu, osazovací plán, strana spojů
- Příloha č. 8 – DPS řídicí části modulu, osazovací plán, strana součástek
- Příloha č. 9 – soupiska součástek výkonové části modulu
- Příloha č. 10 – soupiska součástek řídicí části modulu

Elektronické přílohy:

- složka dokumentace – elektronická verze práce
- složka zdrojový kód – soubory zdrojového kódu pro mikrokontrolér
- složka DPS – soubory desek plošných spojů formátu Eagle5.7
- složka program – projektové soubory ovládacího programu

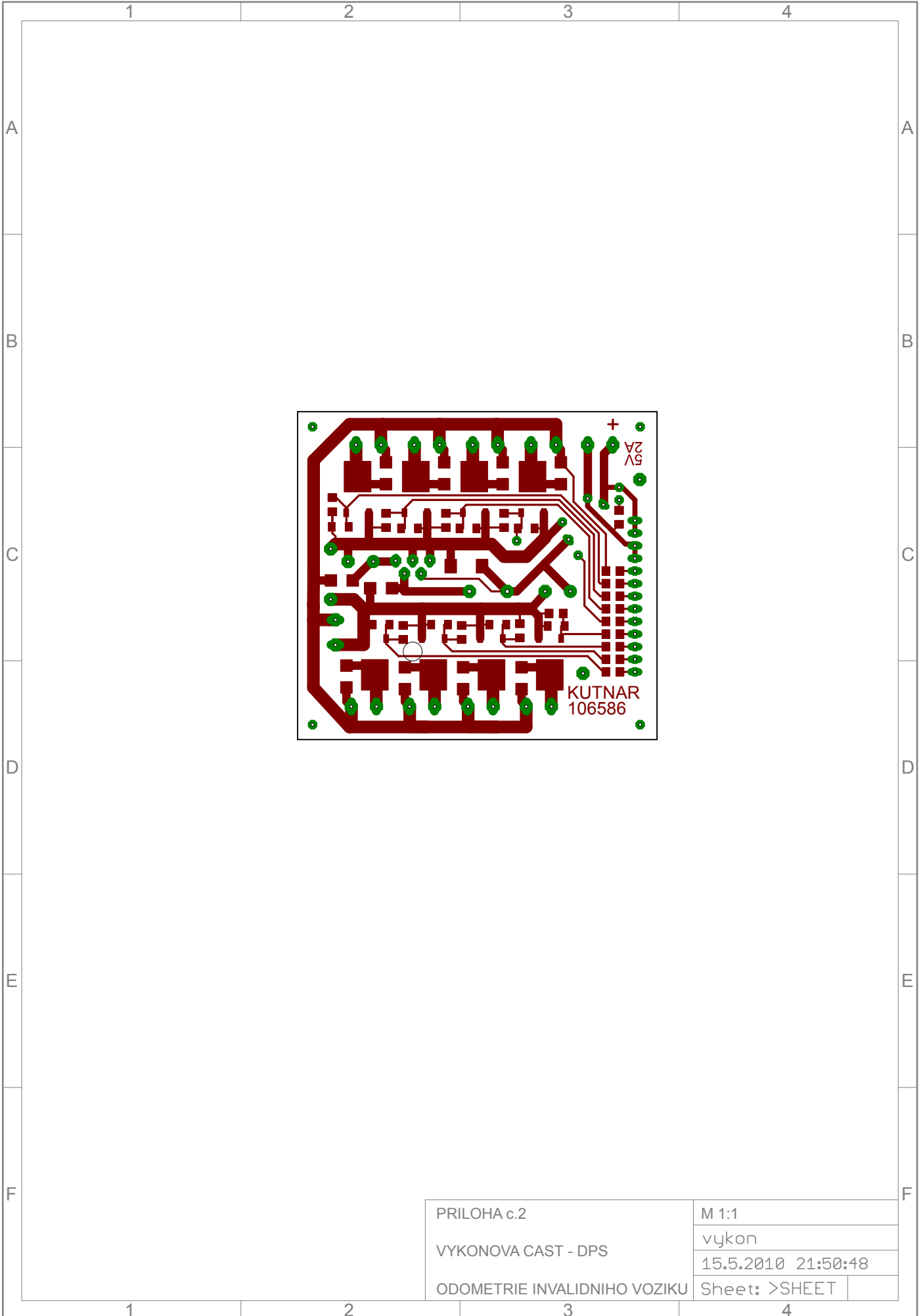
## 14. SEZNAM ZKRATEK, SYMBOLŮ

|                    |   |
|--------------------|---|
| A, mA              | Ampér - jednotka elektrického proudu                    |
| A.D.               | Anno Domini - po Kristu                                 |
| ADC                | analogově-digitální převodník                           |
| ASCII              | standard kódování znaků                                 |
| AVCC               | reference ADC viz [12]                                  |
| b                  | bit   |
| B                  | byte, bajt, osm bitů                                    |
| Bd                 | Baud - jednotka modulační rychlosti                     |
| C#                 | programovací jazyk                                      |
| CPU                | Central Processing Unit, centrální procesorová jednotka |
| DPAK               | pouzdro povrchové montáže diskretních součástek         |
| EEPROM             | elektricky programovatelná i mazatelná paměť            |
| ETHERNET           | technologie lokálních počítačových sítí                 |
| F, $\mu$ F         | Farad - jednotka elektrické kapacity                    |
| FeCl <sub>3</sub>  | chlorid železitý  |
| FLASH              | elektricky programovatelná paměť                        |
| GND                | zem, označení nulového potenciálu                       |
| GPS                | globální navigační systém                               |
| H, $\mu$ H         | Henry - jednotka indukčnosti                            |
| HB, LB             | vyšší / nižší datový byte                               |
| Hz, kHz, MHz       | Hertz - jednotka frekvence                              |
| I2C                | sériová sběrnice  |
| INT0, INT1, INT2   | vnější přerušení, viz [12]                              |
| KI, KP, KD         | parametry PSD regulátoru                                |
| LED                | světlo vyzařující dioda                                 |
| LowESR             | označení kondenzátoru se sníženým sériovým odporem      |
| LSB, MSB           | nejméně / nejvíce významný bit                          |
| m, mm, cm          | metr - jednotka délky                                   |
| MIPS               | milion provedených instrukcí za sekundu                 |
| MMC/SD             | typ paměťových karet                                    |
| MOSI, MISO, SCK    | vodiče rozhraní SPI                                     |
| ohm, $\Omega$      | Ohm - jednotka elektrického odporu                      |
| PSD                | diskretní varianta PID regulátoru                       |
| PWM                | pulsně šířková modulace                                 |
| RC                 | rádiem řízený   |
| RISC               | procesor s redukovanou instrukční sadou                 |
| S, D, G            | označení elektrod unipolárního tranzistoru              |
| s, ms              | sekunda - základní jednotka času                        |
| SPI                | sériové periferní rozhraní                              |
| SRAM               | statická paměť  |
| TTL                | tranzistorově-tranzistorová logika                      |
| TXD (Tx), RXD (Rx) | signálové vodiče sériového rozhraní                     |
| USART, UART        | synchronní a asynchronní sériové rozhraní               |
| USB                | univerzální sériová sběrnice                            |
| UVA                | dlouhovlnné ultrafialové záření                         |
| v, mV              | Volt - jednotka elektrického napětí                     |



|                              |                    |
|------------------------------|--------------------|
| PRILOHA c.1                  |                    |
| VYKONOVA CAST - el. schema   |                    |
| ODOMETRIE INVALIDNIHO VOZIKU |                    |
| vykon                        | 15.5.2010 21:43:45 |
| Sheet: 1/1                   |                    |





PRILOHA c.2

VYKONOVA CAST - DPS

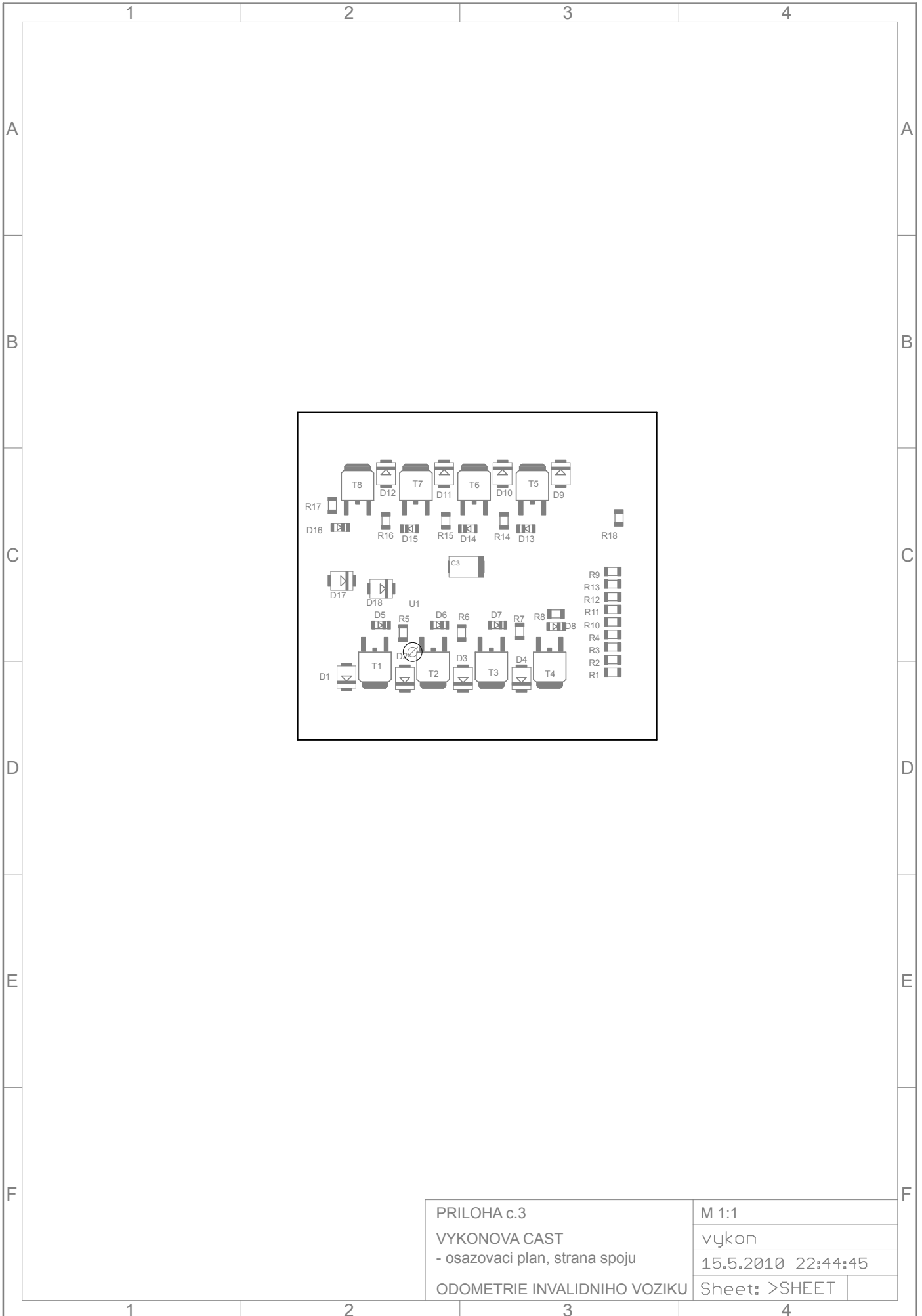
ODOMETRIE INVALIDNIHO VOZIKU

M 1:1

vykon

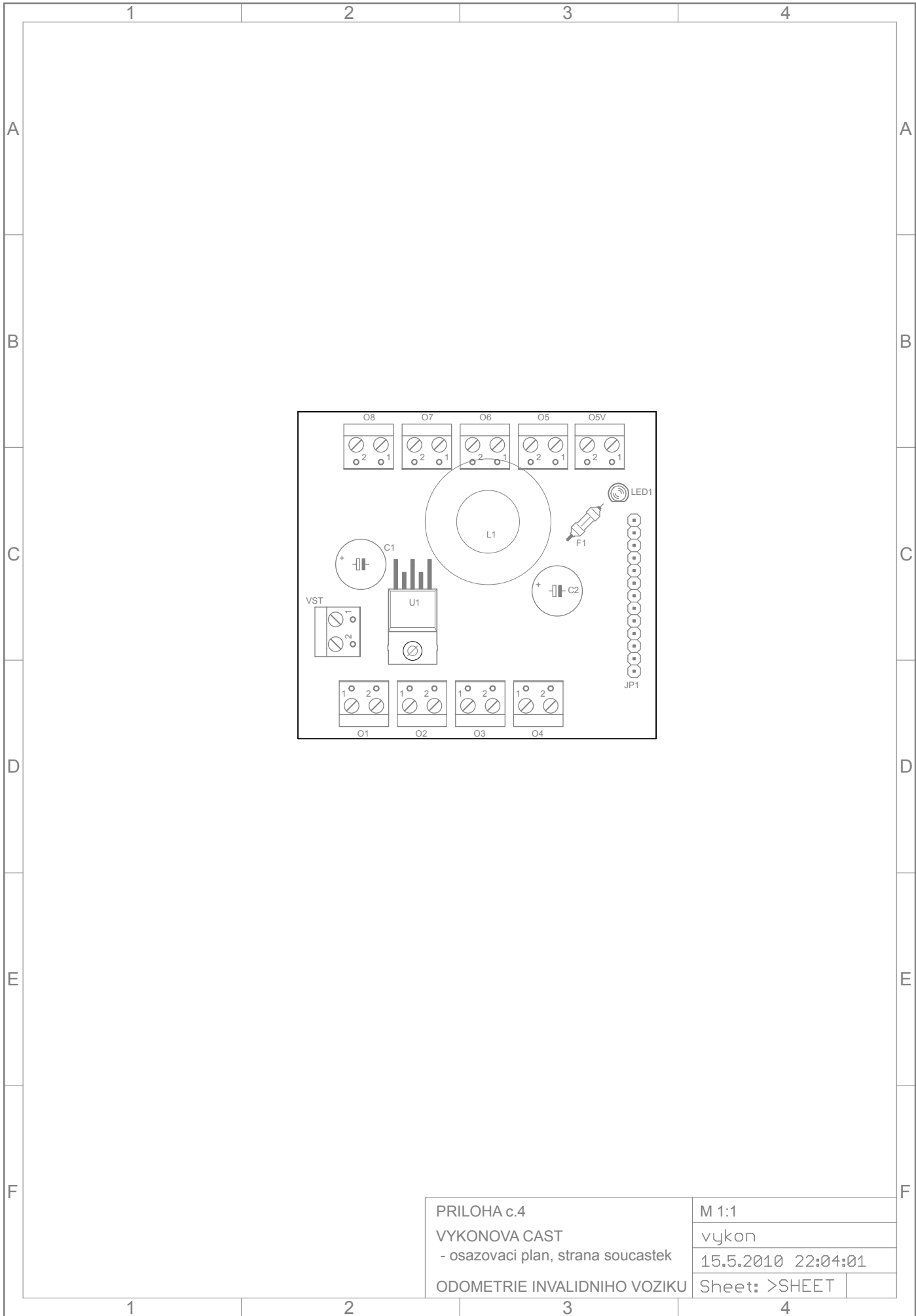
15.5.2010 21:50:48

Sheet: >SHEET



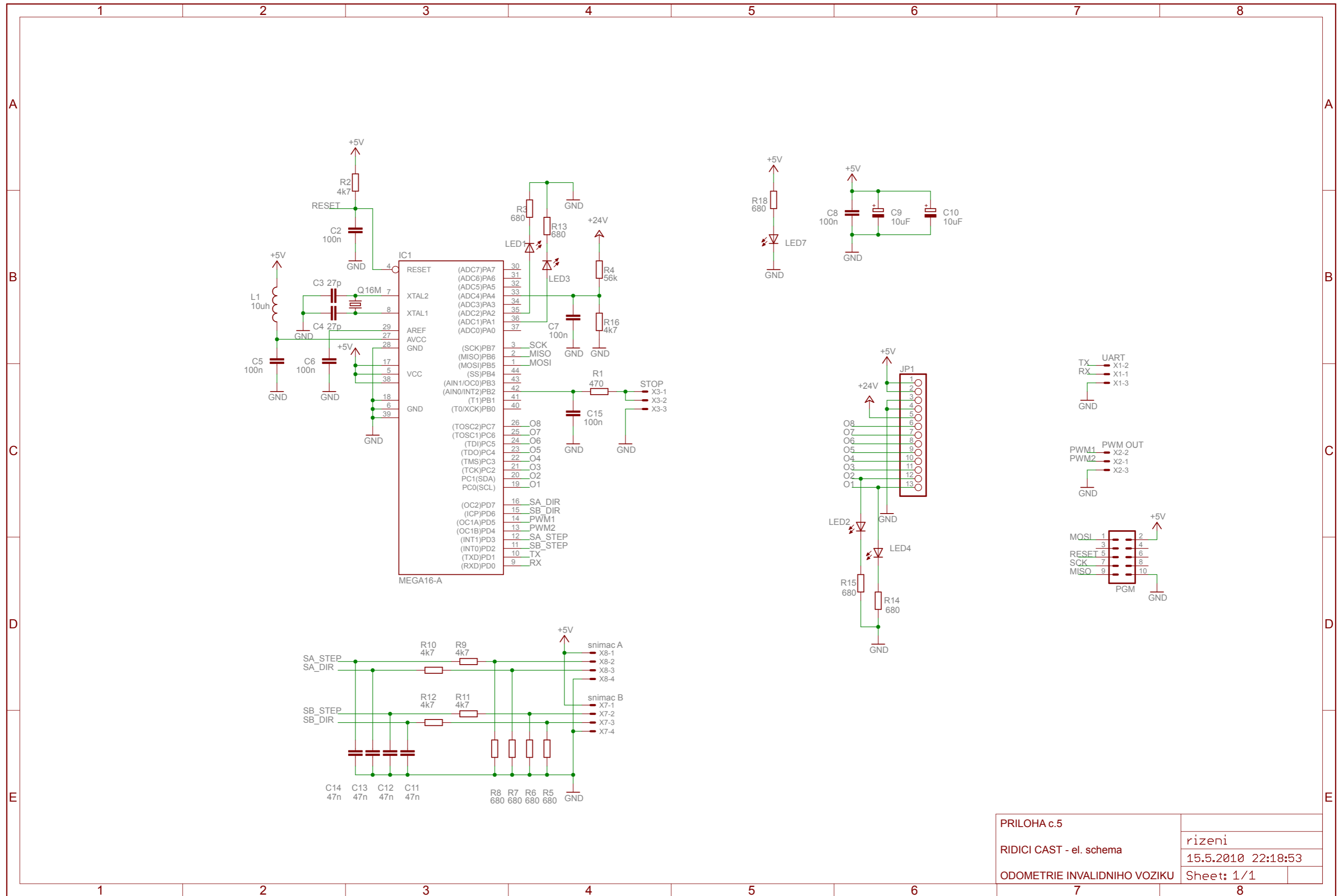
PRILOHA c.3  
 VYKONOVA CAST  
 - osazovací plan, strana spoju  
 ODOMETRIE INVALIDNIHO VOZIKU

|                    |
|--------------------|
| M 1:1              |
| vykon              |
| 15.5.2010 22:44:45 |
| Sheet: >SHEET      |

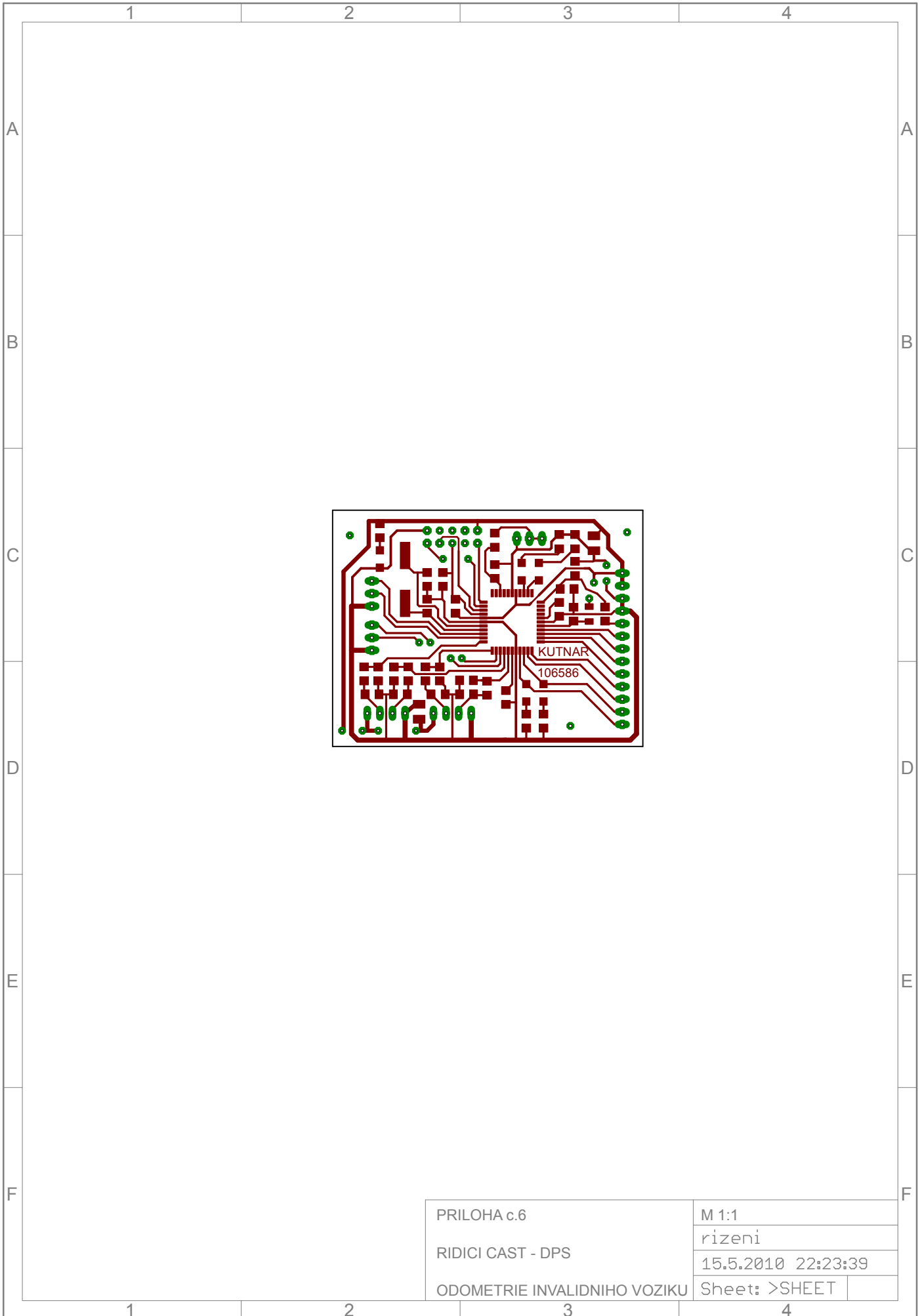


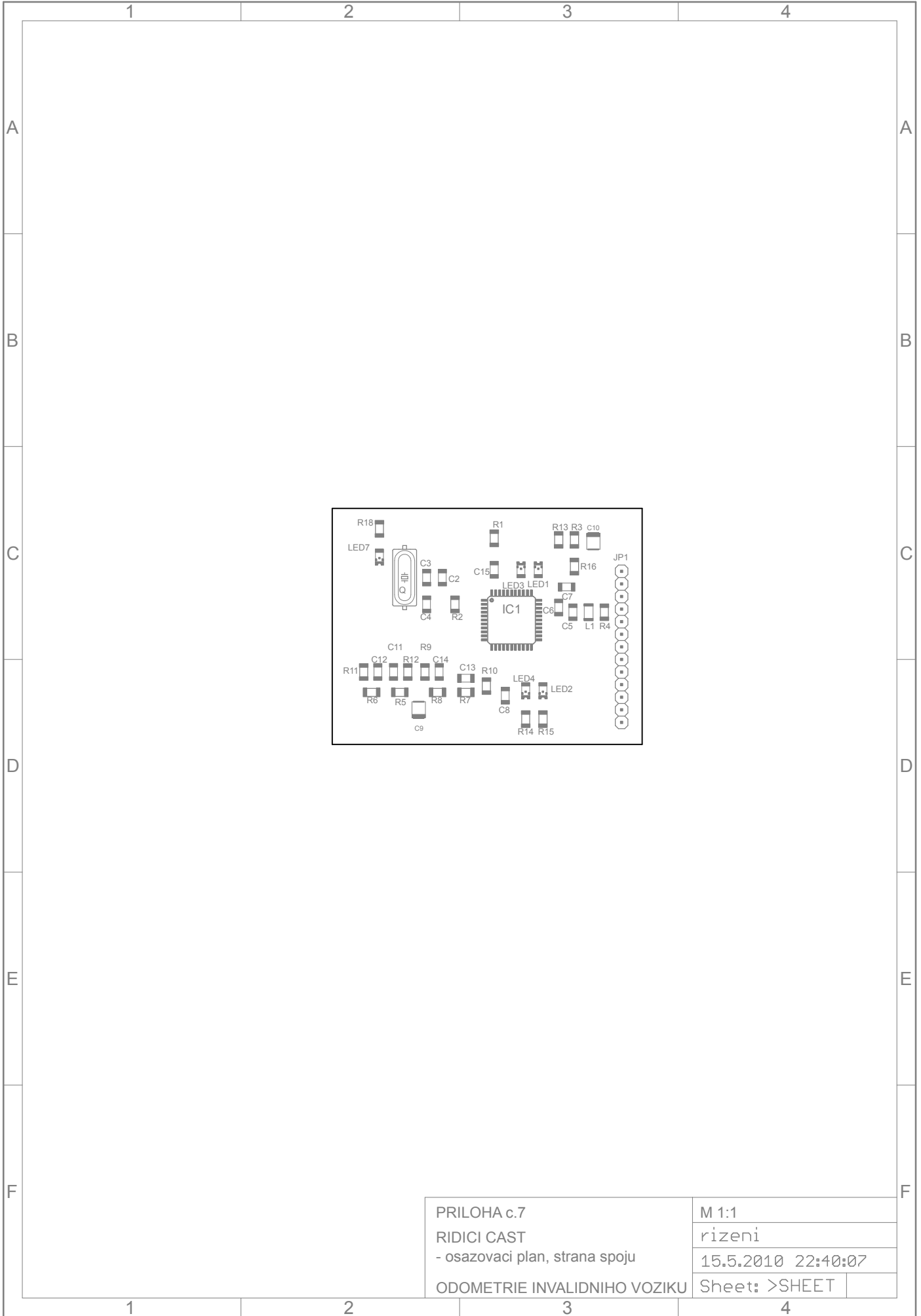
PRILOHA c.4  
 VYKONOVA CAST  
 - osazovací plan, strana soucastek  
 ODOMETRIE INVALIDNIHO VOZIKU

|                    |
|--------------------|
| M 1:1              |
| vykon              |
| 15.5.2010 22:04:01 |
| Sheet: >SHEET      |



|                              |                    |
|------------------------------|--------------------|
| PRILOHA c.5                  |                    |
| RIDICI CAST - el. schema     | rizeni             |
| ODOMETRIE INVALIDNIHO VOZIKU | 15.5.2010 22:18:53 |
|                              | Sheet: 1/1         |

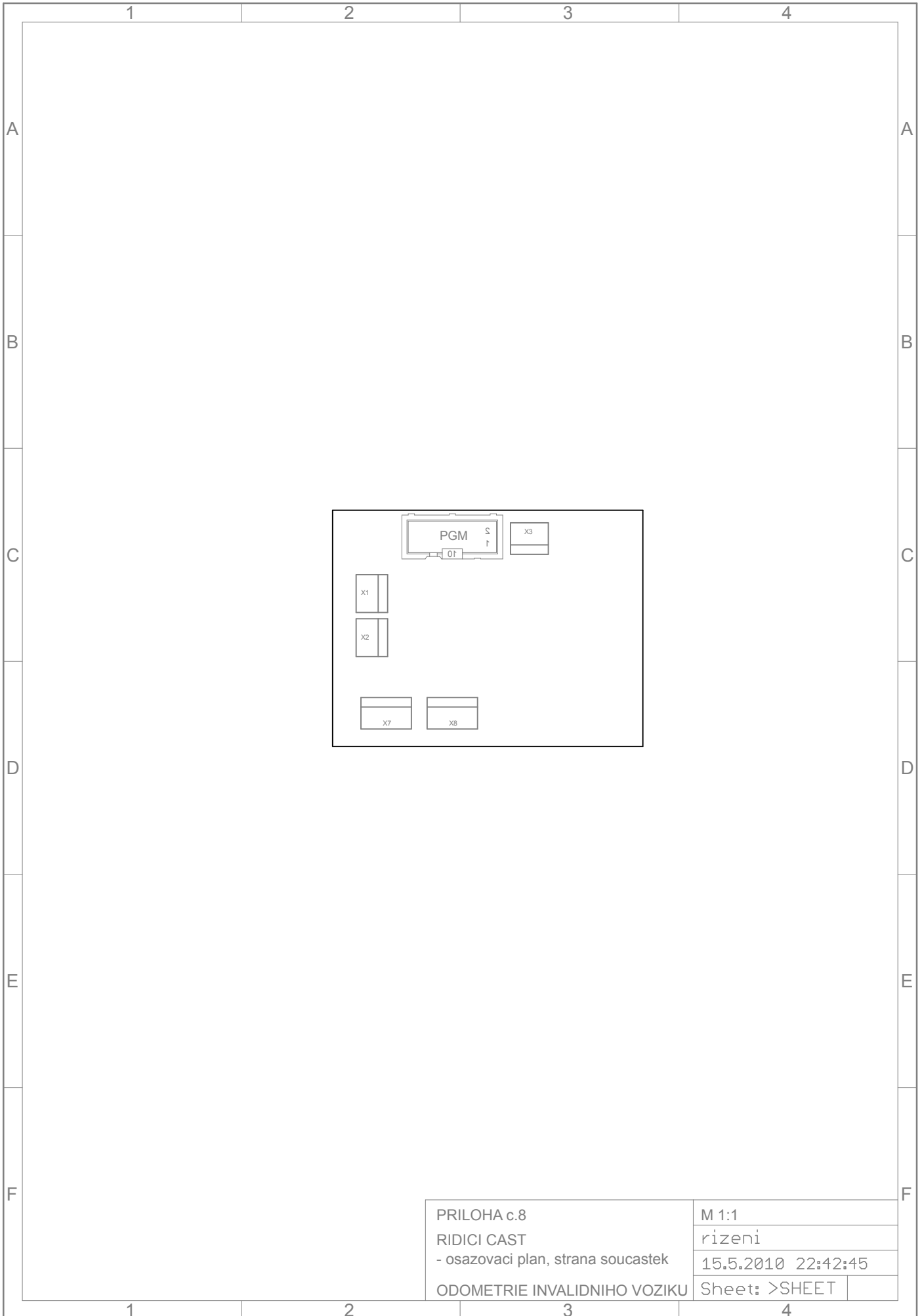




PRILOHA c.7  
 RIDICI CAST  
 - osazovací plan, strana spoju

ODOMETRIE INVALIDNIHO VOZIKU

|                    |
|--------------------|
| M 1:1              |
| rizeni             |
| 15.5.2010 22:40:07 |
| Sheet: >SHEET      |



PRILOHA c.8  
 RIDICI CAST  
 - osazovací plan, strana součástek  
 ODOMETRIE INVALIDNIHO VOZIKU

|                    |
|--------------------|
| M 1:1              |
| rizeni             |
| 15.5.2010 22:42:45 |
| Sheet: >SHEET      |

## PŘÍLOHA Č. 9

### SOUPIS SOUČÁSTEK VÝKONOVÉ ČÁSTI

| Označení                                    | Hodnota/typ  | Pouzdro | Popis  |
|---|--------------|---------|--|
| C1  | 220uF / 35V  | E5-10,5 | Low-ESR hliníkový elektrolytický kondenzátor |
| C2  | 100uF / 16V  | E5-10,5 | Low-ESR hliníkový elektrolytický kondenzátor |
| C3  | 1000uF / 10V | SMC_D   | Tantalový elektrolytický kondenzátor         |
| D1, D2, D3, D4, D9,<br>D10, D11, D12        | S2M          | SMB     | Usměrňovací dioda 2A                         |
| D5, D6, D7, D8, D13,<br>D14, D15, D16       | 6,8V / 0,5W  | SOD80C  | Zenerova dioda                               |
| D17, D18                                    | SK36A        | SMB     | Schottky dioda 60V 3A                        |
| JP1   |              |         | Lámací lišta jednořadá, rozteč 2,54mm        |
| L1  | 330uH/2,5A   |         | Toroidní tlumivka                            |
| LED1  |              | LED3MM  | Zelená LED, průměr 3mm                       |
| O1, O2, O3, O4, O5,<br>O5V, O6, O7, O8, VST |              |         | Svorkovnice, 2 kontakty, rozteč 5mm          |
| R1, R2, R3, R4, R10,<br>R11, R12, R13       | 100R / 0,25W | 1206    | Rezistor SMD                                 |
| R5, R6, R7, R8, R14,<br>R15, R16, R17       | 4k7 / 0,25W  | 1206    | Rezistor SMD                                 |
| R9  | 2k2 / 0,25W  | 1206    | Rezistor SMD                                 |
| R18   | 560R / 0,25W | 1206    | Rezistor SMD                                 |
| T1, T2, T3, T4, T5, T6,<br>T7, T8           | IRLR024N     | TO252AA | N-Channel HEXFET Power MOSFET                |
| U1  | LM2576-5.0   | TO220-5 | Spínaný stabilizátor 3A 5V                   |



## PŘÍLOHA Č. 10

### SOUPIS SOUČÁSTEK ŘÍDÍCÍ ČÁSTI

| Označení                               | Hodnota/typ  | Pouzdro | Popis  |
|--|--------------|---------|--|
| C2, C5, C6, C7, C8, C15                | 100nF / 50V  | 1206    | Keramický kondenzátor                                    |
| C3, C4                                 | 27pF / 50V   | 1206    | Keramický kondenzátor                                    |
| C9, C10                                | 10uF / 16V   | SMC_B   | Tantalový elektrolytický kondenzátor                     |
| C11, C12, C13, C14                     |              | 1206    | Keramický kondenzátor (kapacita dle požadované filtrace) |
| IC1                                    | ATMEGA16-AU  | TQFP44  | ATMEL AVR MEGA 16  |
| JP1                                    |              |         | Lámací lišta jednořadá, rozteč 2,54mm                    |
| L1                                     | 10uH / 100mA | L3216C  | Tlumivka   |
| LED1, LED2, LED3, LED4                 | SMD LED      | 1206    | Žlutá SMD LED  |
| PGM                                    | MLW10G       | ML10    | Programovací konektor, 10 pinů                           |
| Q                                      | 16MHz        | HC49UP  | SMD krystal  |
| R1                                     | 470R / 0,25W | 1206    | Rezistor SMD   |
| R2, R9, R10, R11, R12, R16             | 4k7 / 0,25W  | 1206    | Rezistor SMD   |
| R3, R5, R6, R7, R8, R13, R14, R15, R18 | 680R / 0,25W | 1206    | Rezistor SMD   |
| R4                                     | 56k / 0,25W  | 1206    | Rezistor SMD   |
| X1, X2, X3                             | PFH02-03P    |         | Konektor se zámkem, 3 piny, rozteč 2,54mm                |
| X7, X8                                 | PFH02-04P    |         | Konektor se zámkem, 4 piny, rozteč 2,54mm                |