



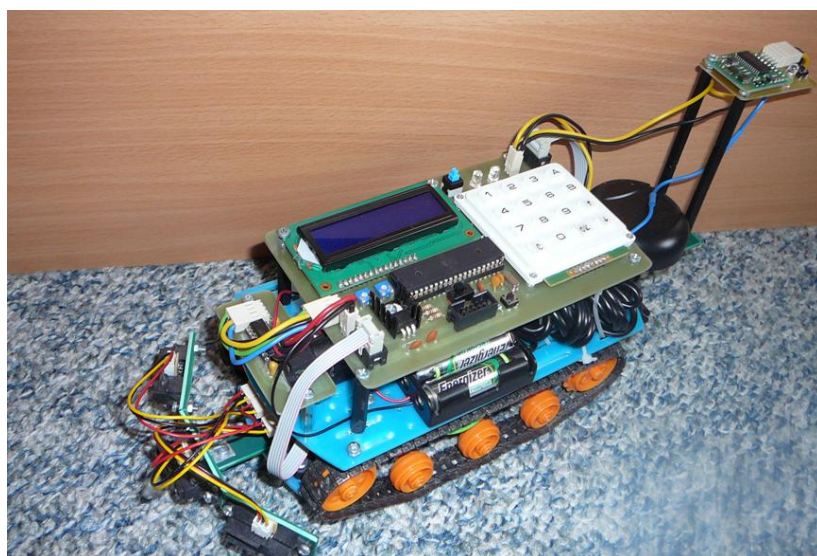
Středoškolská technika 2011

Setkání a prezentace prací středoškolských studentů na ČVUT

Robotické vozítko

Jiří Kubeš

Střední průmyslová škola elektrotechnická
a Vyšší odborná škola Pardubice, Karla IV. 13,



Obsah

1. Seznam použitých značek a symbolů	3
2. Úvod	4
2.1. Motory	4
2.2. Řadič motorů – polomost L293D	4
2.3. Optické senzory vzdálenosti GP2Y0A21YK0F	5
2.4. Maticová klávesnice.....	5
2.5. LCD displej	6
2.6. GPS modul.....	6
2.6.1. Protokol MNEA 0183	7
2.7. Převodník úrovní RS232/TTL MAX232	8
3. Popis funkce programu	8
4. Blokové schéma	13
5. Elektrické schéma	14
Hlavní deska.....	14
5.1. Deska pro senzory	14
5.2. Deska pro řízení motorů	15
5.3. Převodník napětové úrovně RS232 na TTL	15
Oživení.....	16
6. Závěr.....	17
7. Literatura, materiály	18

1. Seznam použitých značek a symbolů

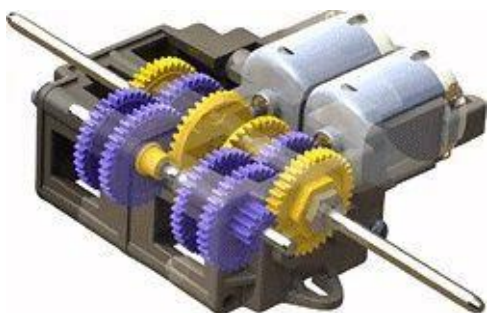
A	Ampér, základní jednotka elektrického proud
B	Byte - 8 bitů, jednotka množství dat
cm	centimetr, setina metru, jednotka vzdálenosti
I/O	Input/output, vstupně výstupní porty
MHz	Milion hertzů, Hertz - základní jednotka frekvence
kb/s	kilo bity za sekundu, přenosová rychlost
kB	1024 bytů
kW	Kilo ohm, 1000W
mA	Mili ampér, tisícina ampéru
mNm	Mili newton metr (tisícina newton metru), jednotka točivého momentu
nF	Nano farad, 10^{-9} faradu, jednotka elektrické kapacity
ss.mm.mmmm	Formát zobrazení souřadnic, $xx^{\circ}xx.xxxx'$
ss.mm.vv.vvv	Formát zobrazení souřadnic, $xx^{\circ}xx'xx.xxx''$
V	Volt, základní jednotka elektrického napětí
mF	Mikro farad, miliontina faradu, jednotka elektrické kapacity
W	Ohm, základní jednotka elektrického odporu
°	Stupně, jednotka azimutu

2. Úvod

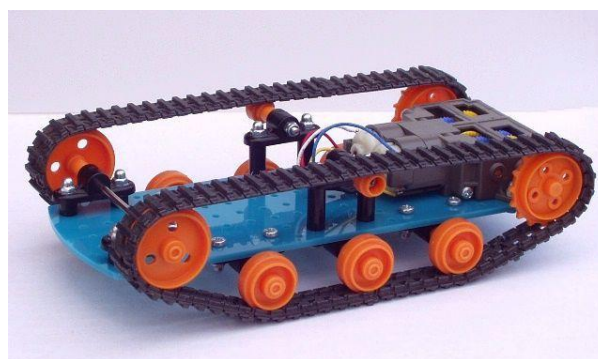
Jedním z rychle se rozvíjejících oborů je robotika a snaha o její využití v praxi. Tento projekt právě do tohoto oboru zapadá a zabývá se navrhnutím systému autonomního pohybu pro robota, jenž by mohl být do budoucna využíván i v praxi. Uplatnil by se zejména při práci v oblastech člověku nebezpečných či těžko přístupných, například při záchranných pracích, vojenských operacích, atd. Jednou by mohl být tento princip využit i k řízení vozidel například ve velkých továrnách či letištích (podobně jako v Rotterdamském přístavu, kde jsou ovšem vozidla řízena počítačem).

2.1. Motory

Pro pohon slouží dva na sobě nezávislé motory napájené 5 V. Ty jsou usazeny v malé čtyřrychlostní převodovce. Převod se nastavuje při sestavování převodovky, a tudíž ho není možné měnit za chodu. Na převodovce je zvolen poměr 38,2:1, což znamená 345 otáček za minutu a krouticí moment 27 mNm. Hřídel z převodovky vede na hnací kola, jež otáčejí s pásy.



Obr 3. Převodovka TM70168



Obr 4. Podvozek s převodovkou

2.2. Řadič motorů – polomost L293D

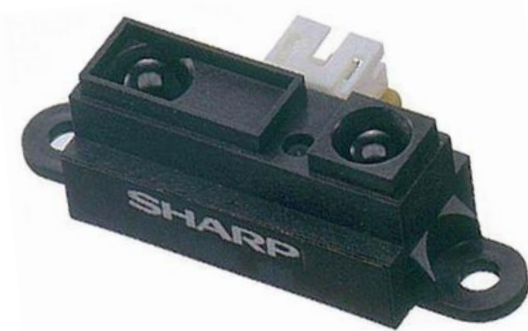
Obvod L293D je navržen tak, aby mohl při napětí 4,5-36V poskytovat motorům až 1,2 A (600 mA na jeden kanál). Řadič musí být napájen dvěma napětími – jedno slouží jako napájení pro logiku (4,5-5,5V) a druhé jako napájení pro motory (4,5-36 V). Pro ovládání motorů slouží 6 vstupů – čtyřmi z nich se ovládá směr otáčení motorů, zbylé dva slouží k regulaci výkonu pomocí PWM.



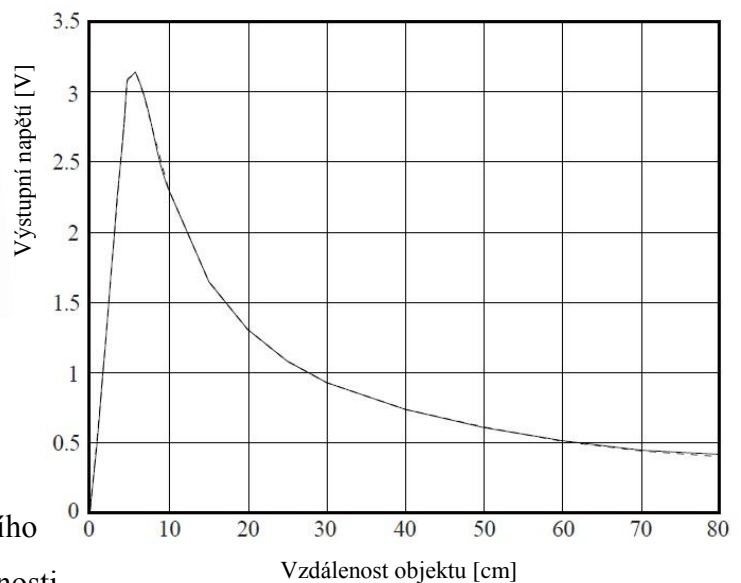
Obr 5. Integrovaný obvod L293D

2.3. Optické senzory vzdálenosti GP2Y0A21YK0F

Tento infračervený snímač vzdálenosti je schopen přesně měřit vzdálenost v rozmezí mezi 10-80 cm. Na výstupu snímače se objevuje analogové napětí (0,4-3,2 V) závislé na vzdálenosti objektů před ním. Díky řídicí jednotce je schopen téměř odbourat okolní rušivé vlivy jako je například malá reflexivnost objektů nebo jejich rozdílná teplota. Odbourávání rušivých vlivů ovšem zabírá čas a tudíž výstupní signál není spojitý – jedno měření trvá přibližně 40 ms. Na obrázku 7 je zobrazena závislost výstupního signálu na vzdálenosti.



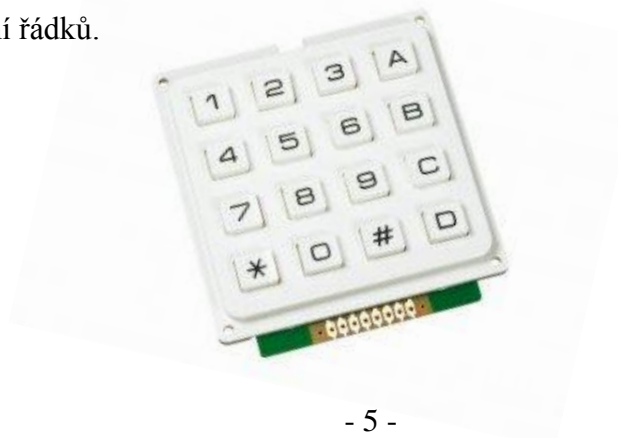
Obr 6. Infračervený snímač vzdálenosti GP2Y0A21YK0F



Obr 7. Závislost výstupního napětí senzoru na vzdálenosti

2.4. Maticová klávesnice

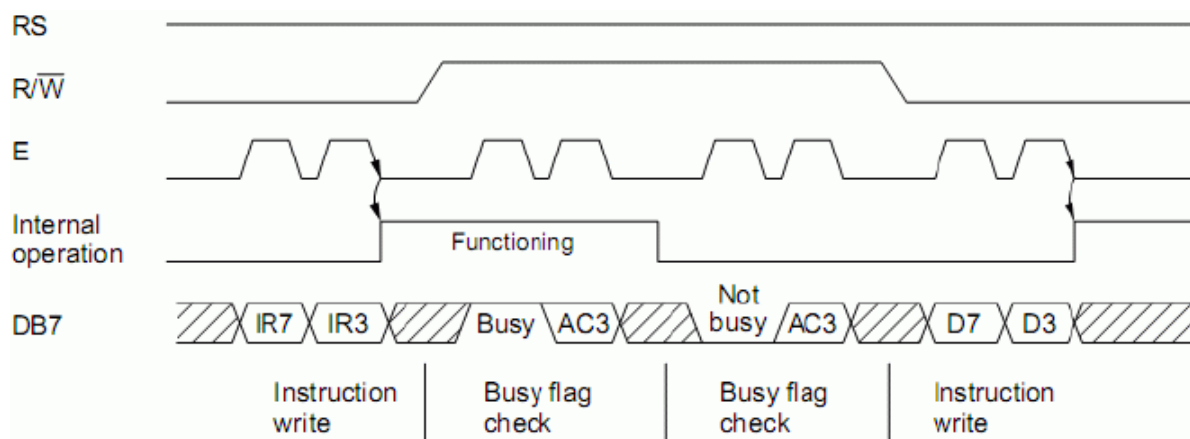
Slouží k orientaci v hlavním menu a zadávání pozice. Princip maticové klávesnice spočívá v tom, že tlačítka jsou v matici, kde souřadnice tlačítka určuje číslo řádku a sloupce. Určení souřadnic stisknutého tlačítka se provádí tak, že vhodným nastavením logických nul a jedniček se vybere jeden řádek, na kterém se testuje stisk tlačítka. Pokud je stisknuto tlačítko právě na tomto řádku, zjistí se, ve kterém sloupci, uloží se jeho kód do proměnné. Pokud ne, pokračuje se s testováním na další řádek. Tento postup funguje i obráceně, tedy vybírání sloupců a testování řádků.



2.5. LCD displej

K zobrazení všech potřebných informací je použit LCD displej ATM1602B, který má znakovou matici velkou 16x2 znaky, modré podsvícení a řadič HD44780. Právě ten je velmi důležitý, jelikož velmi usnadňuje práci s LCD – pro výpis námi požadovaného řetězce stačí pouze jedna až dvě instrukce (v jazyce C). Pokud by zde řadič nebyl obsažen, obsluha by se stala značně složitější.

K LCD displeji jsou připojeny dva trimry (500 Ω a 2,5 k Ω), kterými lze nastavit jas a kontrast displeje. Kromě 5-ti pinů zajišťujících napájení je na displeji dalších 11 pinů určených pro komunikaci s mikroprocesory. Pin RS sloužící k resetování displeje je připojen k mikroprocesoru na vývod PA0, k vývodu PA1 je připojen k pinu R/W, jenž přepíná mezi zapisováním a čtením z displeje, a k vývodu PA2 je připojen pin EN, který povoluje zápis. Zbýlých 8 pinů je datových, ovšem piny DB0-3 jsou spojeny se zemí, tudíž pro přenos informací používáme pouze piny DB4-7 připojené k vývodům mikropočítače PA3-6.



Na obrázku je vidět, jak funguje komunikace mikroprocesoru s displejem pomocí čtyř vodičů. V tomto případě nejdříve mikroprocesor displeji zašle informaci, že pracuje ve čtyřbitovém režimu, a poté nadvakrát pošle displeji data.

2.6. GPS modul

Asi nejdůležitějším komponentem je GPS modul od německé firmy Navilock – NL-404P. Tento GPS modul je schopný zachytit signál GPS i Galileo. Za ideálních podmínek dokáže určit polohu s přesností na 3 metry. Slouží zde nejen k určení současné polohy vozítka, ale i k určení azimutu, ve kterém je vozítko natočeno. K desce je připojen pomocí 6-ti pinového

konektoru MiniDIN (PS/2), kde 2 piny zajišťují napájení, dva jsou určeny k sériové komunikaci na napěťové úrovni RS232 a poslední dva nejsou zapojeny.



2.6.1. Protokol NMEA 0183

Protokol NMEA 0183 byl vyvinut a stále je vyvíjen americkou Námořní asociací pro námořní elektroniku. Kombinuje elektrickou a datovou specifikaci pro komunikaci mezi mořskými elektronickými zařízeními jako jsou například sonary, echolokátory, autopiloti nebo GPS přijímače.

Protokol využívá standardní ASCII tabulku (ovšem pouze tisknutelné znaky). Informace jsou zasílány ve větách pomocí sériové linky jedním vysílačem několika přijímačům. Existuje několik druhů vět, kde každá z nich má pevně daný formát a informace v ní obsažené. Modul NL-404P využívá vět GGA, GLL, GSA, GSV, RMC, VTG a ZDA, přičemž robot využívá pouze větu RMC (Recommended minimum specific GNSS data).

Příklad: \$GPRMC,081728.000,A,2500.0032,N,12159.9914,E, 0.00,42.00,010206,,A*76

Jméno	Příklad	Jednotka	Popis
ID zprávy	\$GPRMC		RMC – udává typ věty
UTC čas	081728.000		Čas udávaný ve formátu hhmmss.sss
Status	A		A=správná data, V=špatná data
Šířka	2500.0032		Zem. šířka ve formátu ssmm.mmmm
Indikace S/J	N		N=North (sever), S=South (jih)
Délka	12159.9914		Zem. délka ve formátu sssmm.mmmm
Indikace V/Z	E		E=East (východ), W=West (západ)

Rychlost	0.00	Uzly	
Azimut	42.00	Stupně	
Datum	010206		Datum ve formátu ddmrrr
Magnetické změny	Není	stupně	
Orientace změn	Není		
Mód	A		N=žádný, A=autonomní, D=DGPS, E=DR
Kontrolní součet	76		

2.7. Převodník úrovní RS232/TTL MAX232

Integrovaný obvod MAX232 složí k převodu napěťových úrovní RS232 ($\pm 15V$) na TTL



úrovně (5V) a opačně. Jeho největší cenou je to, že nepotřebuje napájení $\pm 15V$, ale stačí pouze napájení 5V. Napětí $\pm 15V$ zajišťuje napěťová pumpa tvořená tantalovými kondenzátory 1 μF . Obvod obsahuje dva vysílače a dva přijímače a je schopný pracovat až do 120 kb/s.

3. Popis funkce programu

Program se dělí na tři hlavní bloky – inicializační, konfigurační a mobilní. Tyto tři části následují pevně za sebou a nelze se z libovolné části vrátit k předchozí.

3.1. Inicializační část

Tato část se spustí okamžitě po zapnutí a uživatel do ní nemůže zasahovat (což by ani nebylo možné, trvá řádově několik milisekund). V tomto bloku se nastavuje vše potřebné pro chod systému, v tomto případě:

- Nastavení I/O portů
- Zapnutí časovače 0, jehož pomocí obsluhujeme klávesnici; ostatní čítače časovače jsou prozatím vypnuté
- Povolení sériové linky a její nastavení na asynchronní režim, přenosovou rychlost 38 400 b/s, 8 datových bitů, 1 stop bit a žádnou paritu.

- Nastavení LCD displeje

Po nastavení všech potřebných periférií se vyčká na zapnutí GPS modulu (potřebuje přibližně 2 s) a poté se načte jedna věta RMC a uloží do paměti.

3.2. Konfigurační část

Tato část je jediná, ve které může uživatel provádět jakoukoli interakci s robotem. Hlavním úkolem této části je, aby uživatel zadal souřadnice cíle, na který má robot dojet. Jedná se o menu, ve kterém může uživatel zjistit současnou pozici, nastavovat, měnit či mazat cílové souřadnice nebo prohlížet hodiny. V menu se program orientuje pomocí ukazatele, který ukazuje na pozici v menu. Ukazatel měníme pomocí šipek či pomocí čísel. Tlačítko OK slouží pro potvrzování, tlačítko C slouží jako zpět.

3.2.1. Současná pozice

Pokud zvolíme tuto možnost, zobrazí se současná pozice. Kvůli přehlednosti se zobrazuje zvlášť zeměpisná šířka a zeměpisná délka. Aby se zobrazila, musí se povolit přerušení od sériové linky, díky kterému začne mikroprocesor načítat data přicházející z GPS modulu. Ze všech přicházejících vět se uloží do paměti pouze věta RMC, ze které dostaneme potřebné informace o současně pozici. Vzhledem k tomu, že formát souřadnic zasílaných GPS modulem je ss.mm.mmmm (pouze stupně a minuty v desetinném tvaru), musíme je převést na formát ss.mm.vv.vvv, jenže všichni znají. GPS poloha se neaktualizuje průběžně, pouze při přepínání mezi zobrazením zeměpisné šířky a délky, kdy se opakuje stejná rutina jako popsána výše.

3.2.2. Zadávání pozice

V tomto případě máme dvě možnosti. Můžeme buďto zadat nový cíl nebo změnit cíle již zadané. Celkem můžeme zadat až pět různých cílů, které jsou řazeny za sebou. V prvním případě zadáme nový cíl, který se přidá na další volné místo. Zadávání probíhá tak, že po stisku klávesy se dané číslo zobrazí na displeji a zároveň uloží do pomocného pole.

Po dokončení zápisu se zkontroluje, zda nebyly údaje zadány špatně. Pokud byly zadány špatně, musejí být zadány znovu. V opačném případě se čísla z pomocného pole zkopírují do pole umístěného v paměti EEPROM. V paměti EEPROM jsou umístěny kvůli tomu, aby

souřadnice byly přístupné i po vypnutí. Pokud nechceme přidávat nový cíl, ale pouze změnit stávající cíle, musíme tuto možnost vybrat a poté ještě zvolit cíl, který chceme měnit. Poté následuje to samé jako při přidávání nového cíle s tím, že do EEPROM nezapisujeme na další volná místa, ale přepisujeme informace o jiných cílech.

3.2.3. Současně cíle

Tato možnost slouží ke kontrole zadaných souřadnic. Pokud tuto možnost zvolíme, musíme ještě vybrat cíl, který chceme prohlížet. Po vybrání cíle se jednoduše načtou informace o cíli z paměti EEPROM a zobrazí se na displeji.

3.2.4. Mazání pozic

Po zvolení této položky musíme, podobně jako v případě prohlížení pozic, nejdříve zvolit cíl, který chceme smazat. Po vybrání cíle se tento cíl smaže, a pokud nebyl posledním cílem, tak se na jeho místo zkopírují ještě souřadnice následujícího cíle. Nakonec se musí zmenšit počítadlo cílů, které nám udává počet cílů.

3.2.5. Čas a datum

Jelikož GPS modul dokáže podávat i informace o čase, můžeme si ho zobrazit. V menu se tato položka nachází pod číslem 5. Po jejím zvolení se na horním řádku zobrazuje čas a na spodním řádku datum. Aby se čas mohl zobrazit, musíme ze všeho nejdříve povolit přerušování od sériové linky, pomocí kterého načteme první větu RMC. V okamžiku, kdy je věta načtená celá, zakážeme přerušování a nastavíme příznak pro načtení nových dat. Po načtení věty z ní zkopírujeme informace o čase a datumu, které zobrazíme na displeji. Poté skočíme do smyčky, která trvá tak dlouho, dokud nestiskneme tlačítko C. V této smyčce čekáme jen na příchod nových dat či stisk tlačítka C. V případě, že mikrokontrolér načte novou větu RMC, vynulujeme příznak pro nová data a z načteného řetězce zkopírujeme informace o času do oddělených řetězců, které pak převedeme na čísla, abychom s nimi mohli pracovat. Poté zkontrolujeme, zda přichodí data odpovídají předpokladům (neliší se o více než jednu vteřinu), a jestliže data vyhovují, obnovíme informaci na displeji. Informace o datumu se neaktualizuje, protože se předpokládá, že robot nebude sloužit jako hodinky či kalendář.

3.2.6. Povel k odjezdu

Zvolením této možnosti ukončíme konfigurační část a dáme vozítku povel k tomu, aby odjel na námi zadané souřadnice. Předtím ale musí provést ještě několik nezbytných operací. Nejdříve musíme převést zadané souřadnice z formátu ss.mm.vv.vvv na formát ss.mm.mmmm, ve kterém posílá informace GPS modul. Děláme to proto, abychom nemuseli po každém novém zjištění informace převádět souřadnice na stejný formát jako zadávané. Poté zakážeme časovač 0, který již nebude potřeba (klávesnici již používat nebudeme), a povolíme časovač 1, který na displeji zobrazuje informace například o současných pozici či času.

3.3. Mobilní část

V této části opět uživatel nemůže zasáhnout do procesů dějících se v robotovi, jelikož v této části se robot chová zcela autonomně. Jedinou výjimku tvoří reset. V této části se program stále točí v nekonečné smyčce, ve které provádí dva bloky příkazů. V první části robot zjišťuje informace o pozici a směru jízdy a ty následně zpracovává. V druhé části robot pomocí zpracované informace a stavu terénu před sebou řídí svůj pohyb.

3.3.1. Zpracování informací

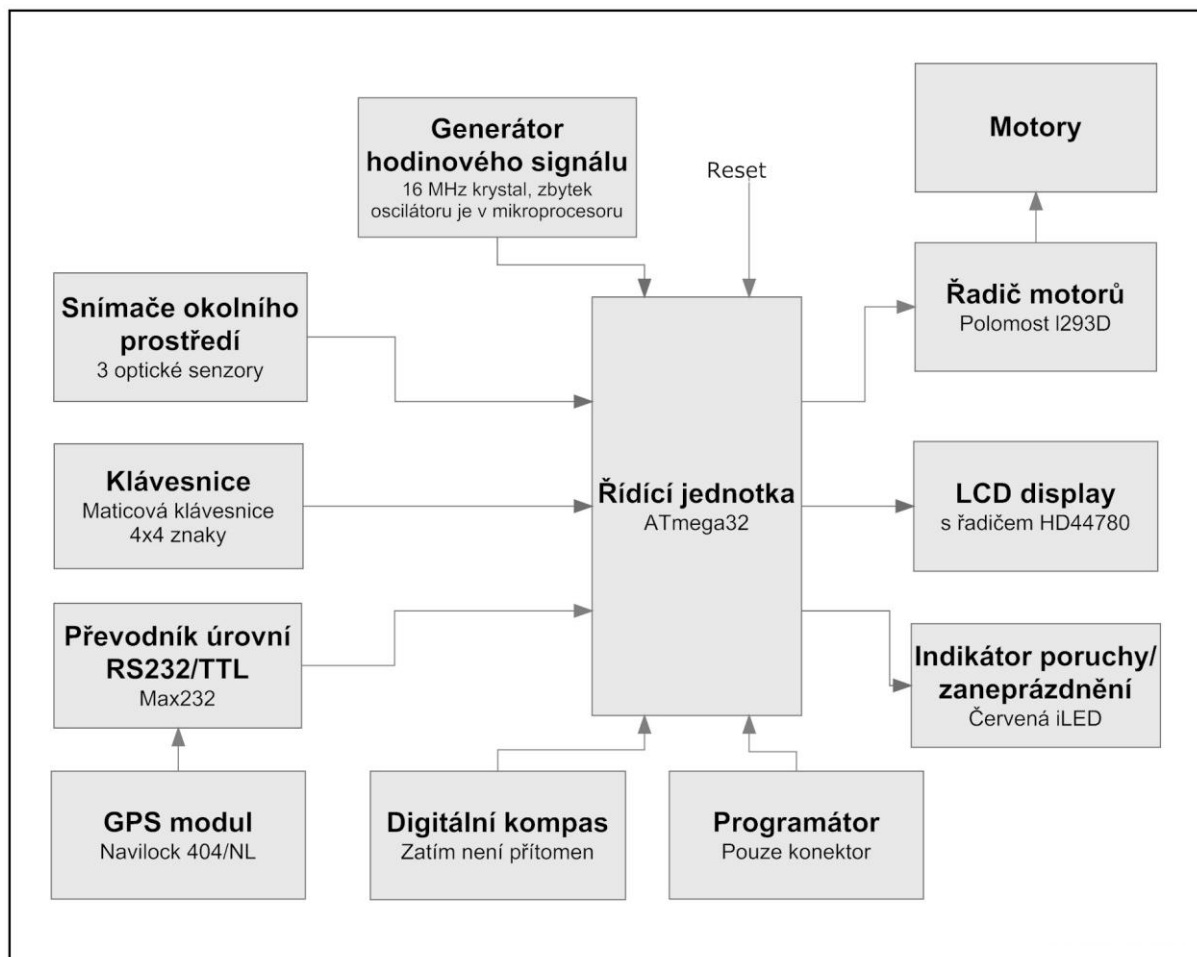
Nekonečná smyčka vždy začíná rozdělením načteného řetězce do jednotlivých částí, tedy na zeměpisnou šířku a délku, čas, datum a kontrolní informace. Pokud jsou informace validní, vypočteme z posledních 15-ti hodnot průměr a ten považujeme za správnou pozici. Průměr počítáme z toho důvodu, že GPS modul není naprosto přesný a informace o pozici se může lišit v řádu několika desítek centimetrů a jednotek metrů. V případě, že současná pozice a cíl jsou vzdálené, nemusíme tento průměr počítat, jelikož nepřesnost se příliš neprojeví. V případě, že se vozítko nachází blízko cíli, musíme již počítat v tímto průměrem, jelikož by tyto nepřesnosti mohli naprosto zničit schopnost robota orientovat se v prostoru a jet na dané souřadnice. Pro urychlení výpočtu počítáme s tím, že pokud se současná souřadnice a souřadnice cíle liší ve stupních, můžeme zanedbat vteřiny a desetiny vteřin, jelikož jejich zanedbáním se nedopouštíme chyby větší než 1,5%. Pokud se souřadnice liší maximálně o jeden stupeň, počítáme již i s vteřinami. V případě, že se souřadnice liší maximálně o jednu minutu, používáme k výpočtu také desetiny vteřin. Po vypočtení souřadnic směrového vektoru ze současných a cílových souřadnic z tohoto vektoru vypočteme jeho úhel. K tomu

využíváme funkce atan2, která ze dvou čísel typu double (v našem případě zeměpisná šířka a délka) vypočte úhel udávaný v rozmezí od $-\pi$ do π . Poté co získáme úhel v radiánech, převedeme jej na stupně. Vzhledem k tomu, že na jednotkové kružnici (z níž vycházejí všechny goniometrické funkce včetně tangens) je úhel 0° na pravé poloose vodorovné osy a „směr otáčení“ je proti směru hodinových ručiček, kdežto u azimutu značí hodnota 0° sever (tzn. horní poloosu svislé osy) a „směr otáčení“ je ve směru otáčení hodinových ručiček, musíme ještě vypočtený úhel upravit tak, abychom ho mohli porovnat s azimutem. Úprava má několik kroků, z nichž hlavními jsou odečtení vypočteného úhlu od 360 stupňů (změna „směru otáčení“) a následné přičtení 90° , což zajistí posun 0° z pravé vodorovné poloosy na horní svislou poloosu. Po menších korekcích, jimiž mohou být například azimut větší než 360° v důsledku přičítání 90° , porovnáme současný azimut s azimutem, který jsme právě vypočítali, a získáme odchylku od ideálního směru. Poté přechází běh nekonečné smyčky do své druhé půle.

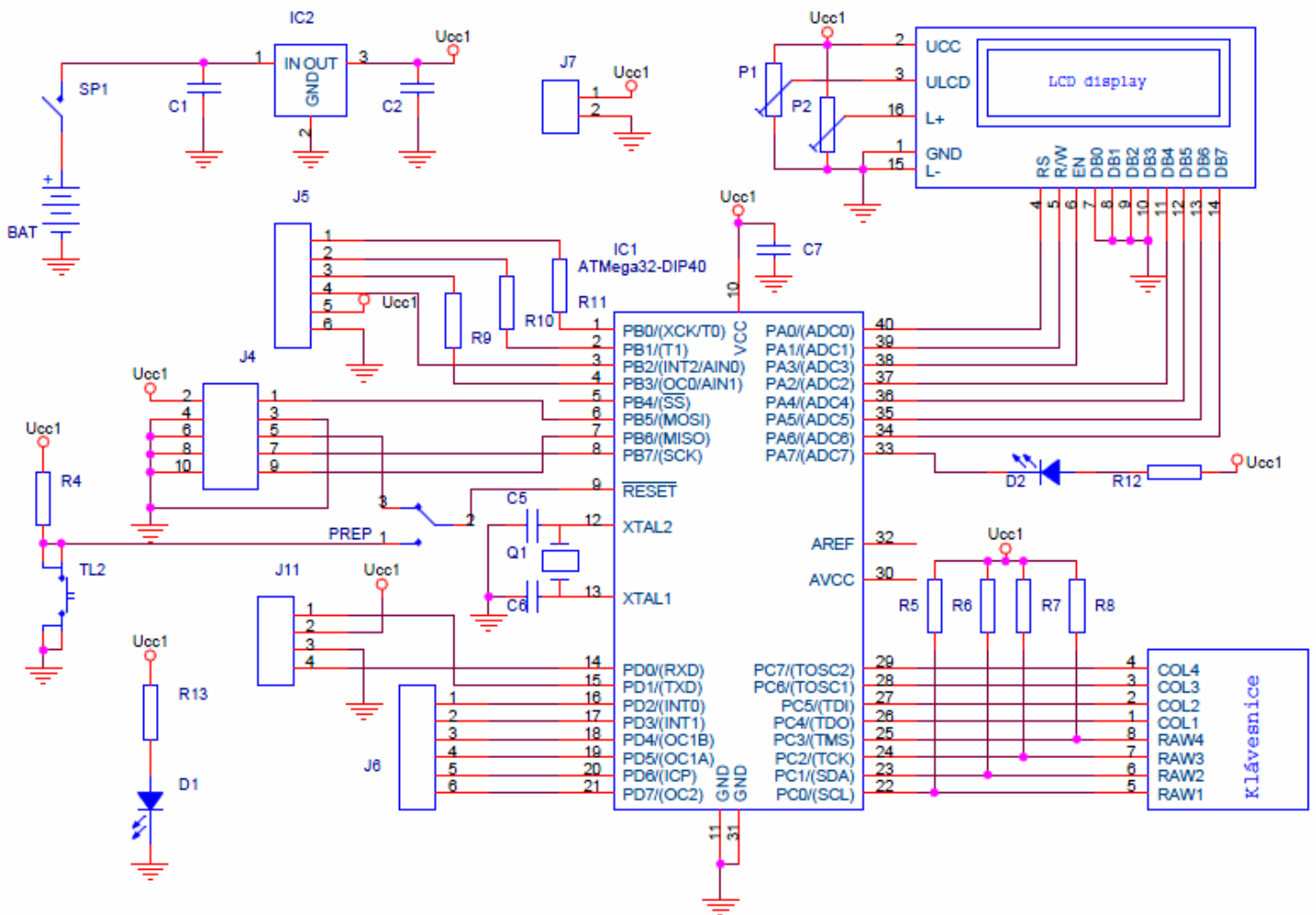
3.3.2. Řízení pohybu

Podle odchylky od ideálního směru se dále ubírá běh programu. Větví se na tři větve – odchylka je nulová (tendence robota jet rovně), odchylka je kladná (tendence robota jet vlevo) nebo je odchylka záporná (tendence robota jet vpravo). V každé větvi je téměř stejný, pouze jsou pozměněné priority. Vždy kontrolujeme, zda je volno v tom směru, kam by měl robot ideálně směřovat. Pokud je v tomto směru volno, robot se tam nasměruje. Pokud se v daném směru nachází nějaká překážka, robot testuje nejbližší možný směr, kterým se může vydat. Mezitím, co robot vyhledává vhodnou trasu, se načítají nová data o současně pozici, aby mohla co nejrychleji zpracována.

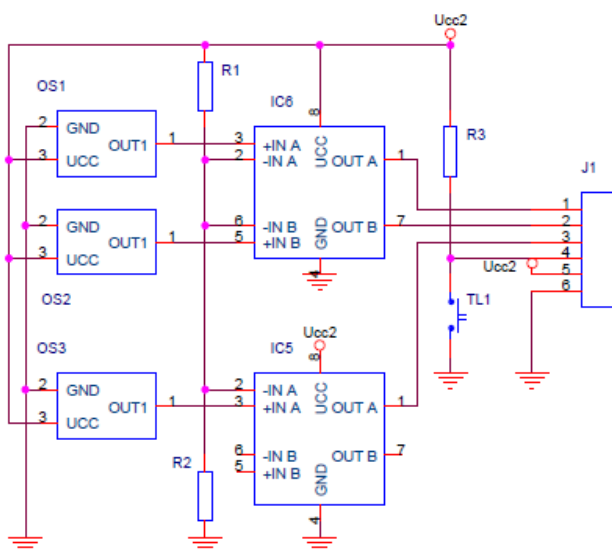
4. Blokové schéma



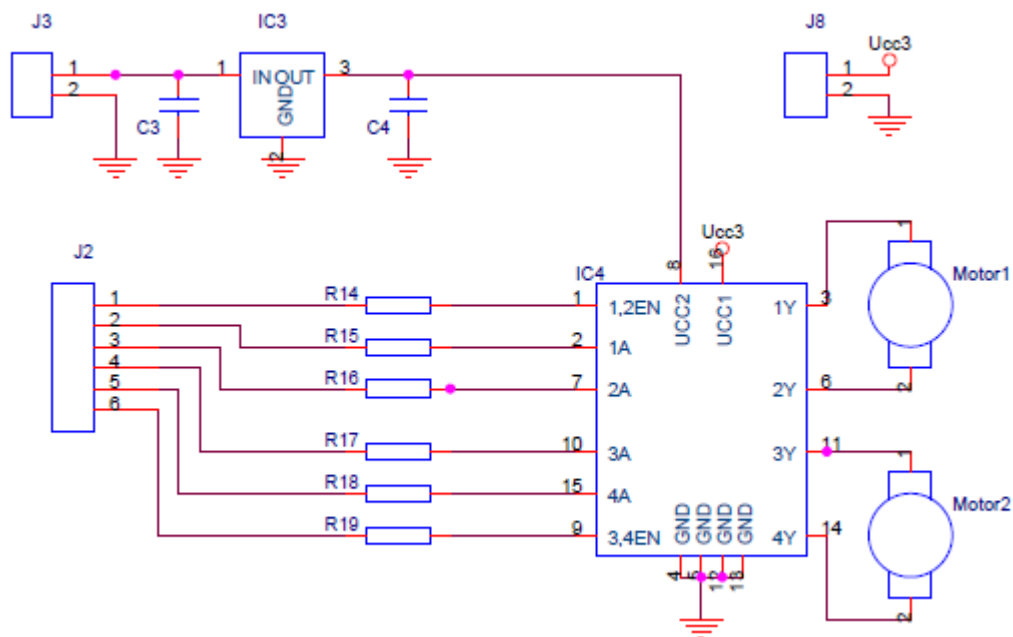
5. Elektrické schéma



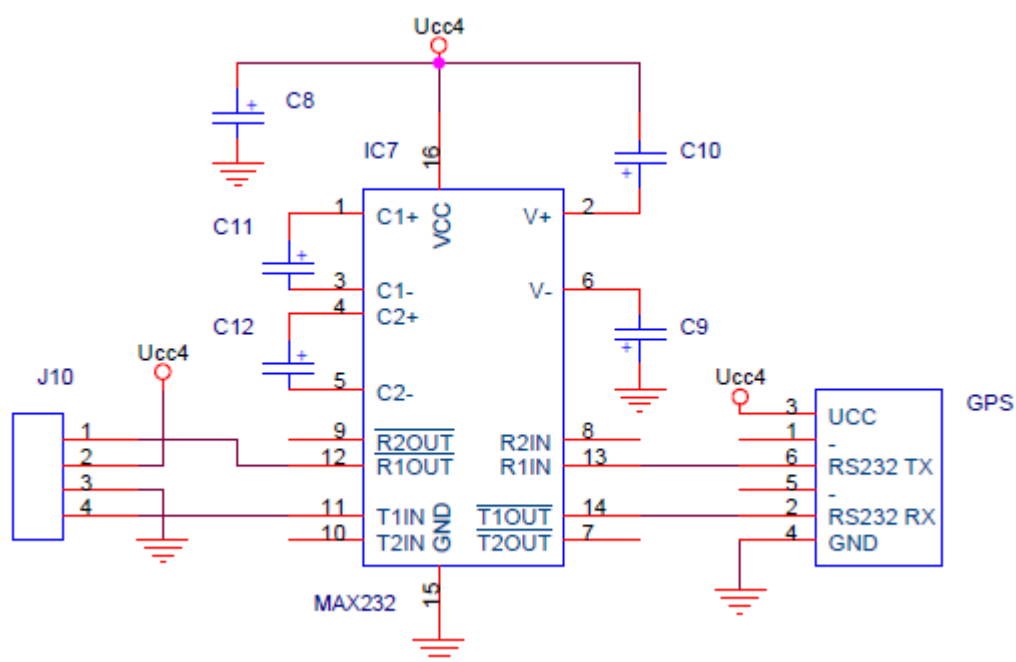
5.1. Deska pro senzory



5.2. Deska pro řízení motorů



5.3. Převodník z úrovně RS232 na TTL



Oživení

Jako první část jsem zkoušel LCD displej., u kterého se vyskytla první chyba. V datasheetu tohoto displeje byly v nákresu, kterým jsem se řídil, špatně popsané piny pro napájení podsvícení. V rozpise je sice vše v pořádku, řídil jsem se ovšem pouze nákresem. Poté jsem vyzkoušel funkčnost regulátorů jasu a kontrastu. Po zprovoznění displeje jsem kontroloval správnou funkčnost klávesnice, respektive správnost kódů jednotlivých kláves a jejich případné korekce. Správnost kódů kláves jsem zjišťoval tak, že jsem si na displej nechal vypsat proměnnou, ve které byl uložen kód právě stisknuté klávesy, a zobrazenou hodnotu jsem porovnal s předpokládanou hodnotou a případně pozměnil.

Funkčnost senzorů jsem testoval tak, že jsem v nekonečné smyčce stále kontroloval stav senzorů a ten si nechával vypisovat na LCD displej. Funkčnost senzorů byla bezchybná, ovšem později při kontrole kooperace senzorů a motorů jsem zjistil, že vzdálenost 14 cm, ve které čidla detekovala překážky, je nedostačující a tudíž je potřeba změnit poměr na děliči napětí $R1:R2$, který udává referenční napětí komparátoru. Změnil jsem pouze jeden rezistor, a to $R2$, jenž byl změněn z 18 k Ω na 10 k Ω . Nová detekční vzdálenost je tedy 24 cm.

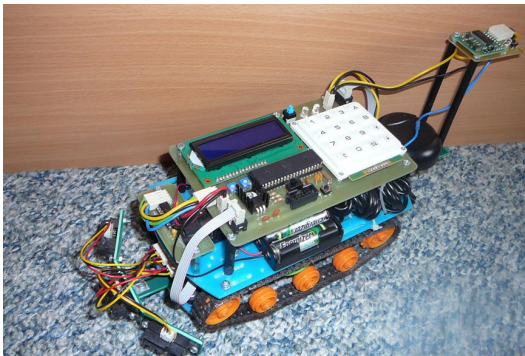
Při testování motorů se točil pouze pravý motor, levý zůstával nehybný při jakémkoliv nastavení řídicích portů mikroprocesoru. Po zkontrolování softwaru jsem byl nucen rozmontovat celé vozítko a zkontrolovat desku pro řízení motorů. Hlavní desku jsem nechal propojenou s deskou pro řízení motorů a do řídicí jednotky jsem nahrál program, který na celém portu D (port pro řízení motorů), nastaví jedničky a cyklicky nastavuje nulu na jednotlivých pinech (pokud je na pinu enable a jednom z řídicích pinů logická jednička a na druhém řídicím pinu logická nula, motor se otáčí), nikdy ne však na více současně. Měřením jsem zjistil, že ve dvou fázích se na některých pinech objevuje napětí přibližně 2,5 V, což je polovina 5V, tedy napěťové úrovně pro logickou jedničku u obvodů TTL. Vzhledem k tomu, že mikrokontrolér je k řadiči připojen přes rezistory 10 k Ω , bylo jasné, že se někde mezi cestami vytvořil nechtěný kontakt, což vedlo k vytvoření děliče napětí s poměrem 1:1. Šlo o cesty připojené k řídicím pinům řadiče a vzhledem k tomu, že oba piny byly stále ve stavu logické jedničky (napětí 2,5 V je v TTL logice ještě bráno jako logická jednička), se výstup levého motoru nacházel stále ve stavu vysoké impedance a motor se nemohl otáčet.

Oživení GPS modulu mi dalo největší práci. K tomuto modelu jsem nenašel žádný datasheet, musel jsem se tedy spoléhat na zdroje z internetu, které mohou být méně i více nekvalitní.

Bohužel jsem narazil na více nekvalitní. Po dotazu mi firma Navilock zaslala popis konektoru tohoto modelu, kde jsou piny údajně poskytující TTL data označeny jako nezapojené. Data byla poskytována pouze na úrovni RS232, což jsem si raději ověřil na osciloskopu. Pro fungující komunikaci byl tedy ještě potřeba převodník mezi těmito úrovněmi, pro nějž byla vytvořena malá destička umístěná vpředu robota.

6. Závěr

Vzhledem k tomu, že tento robot je mojí první zkušeností s robotikou a programováním



mikrokontrolerů v jazyce C, je výsledek velmi dobrý. Komunikace s GPS modulem funguje bezchybně, zjišťování azimutu je také bezproblémové, menu pro zadávání polohy sice není podle nejnovější módy s dotykovým displejem, ale i přesto plní svou úlohu plně dobře, a snímání okolí senzory je přesné. Jediný problém nastává s překážkami typu noha židle

či kmen malého stromku, které senzory nemusí zaregistrovat (tento problém by šel odstranit ultrazvukovým čidlem vzdálenosti). Robot je v tuto chvíli schopný se pohybovat po rovném terénu a vyhýbat se překážkám. Robot si dokáže udržet alespoň přibližně správný směr i v případě, že se dostane mimo signál nebo dostává špatná data.

I přes to, že robot funguje, chtěl bych se nadále věnovat jeho vylepšování. Místo stávajícího podvozku, který slouží spíše jako testovací, bych chtěl použít větší a robustnější, který bude schopný se vyrovnat i s náročným terénem. V souvislosti s podvozkem bych chtěl využít silnějších motorů, jelikož je robot relativně těžký. Dále bych chtěl vyměnit snímání okolí pomocí infračervených čidel za kamerový systém, který je kromě ceny ve všem ostatním několikanásobně lepší než infračervená čidla.

Zatím robot nemůže sloužit k žádnému praktickému užití, myslím si ale, že s dostatkem času, nápadů a peněz by jednou mohl být člověku velmi užitečný.

7. Literatura, materiály

Literatura:

[1] Váňa, V.: Mikrokontroléry ATMEL AVR – programování v jazyce C. 1. vydání Praha, BEN – technická literatura 2003. 205 s.

Internet:

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

[2] <http://www.gme.cz>

[3] <http://en.wikipedia.org>

[4] <http://cs.wikipedia.org>

[5] <http://www.navilock.de>

[6] <http://www.snainstruments.com>

[7] <http://www.linuxsoft.cz>

[8] <http://www.8bitu.cz>

