



Středoškolská technika 2011

Setkání a prezentace prací středoškolských studentů na ČVUT

Robot RC-DC

Jaroslav Růžička

SPŠ a VOŠ
Karla Čapka 402, Písek

ANOTACE

Ovládání motorů ze stěrače automobilu síťovým směrovačem Edimax BR-6104KP, komunikaci zajišťuje wi-fi směrovač Edimax.

Klíčová slova: autonomní robot, Edimax BR- 6104KP, Edimax- 6204WG

ANNOTATION

Control engines from wiper of car router Edimax BR-6104KP, communication of Wi-Fi router Edimax.

Key words: autonomous robot, Edimax BR- 6104KP, Edimax- 6204WG

Obsah

1. Teorie.....	5
2. Řešení	9
2. 1. Bezdrátový router Edimax BR- 6204WG	9
2. 2. Router Edimax BR- 6104KP	11
2. 3. Převodník AVR	14
2. 4. Můstek Sabertooth 2X10	15
2. 5. Motory	17
2. 6. Display.....	18
2. 7. I2C sběrnice, UART sběrnice.....	19
2. 8. Ovládání robota	19
2. 9. Konstrukce podvozku	25
3. Závěr	27
4. Odkazy na použitou odbornou literaturu.....	29

1. Teorie

Robot je stroj pracující s určitou mírou samostatnosti, vykonávající určené úkoly, a to předepsaným způsobem a při různých mírách potřeby interakce s okolním světem a se zadavatelem. Robot je schopen své okolí vnímat pomocí senzorů, zasahovat do něj, případně si o něm vytvářet vlastní představu. Roboti se dají rozdělit do několika skupin. Podle jejich schopnosti přemísťovat se je dělíme na stacionární- nemohou se pohybovat z místa na místo a na mobilní, kteří se mohou pohybovat z místa na místo. Dále se roboti dělí podle možnosti pohybu, autonomie, účelu (boj, výroba, průzkum, přeprava), způsobu programování. Autonomie robota závisí na tom, jak moc je samotný robot dynamický a interaktivní s okolním prostředím. Čím více je robot dynamičtější, tím přesnější potřebuje informace o svém okolí.

Podle toho, kolik potřebuje lidského zásahu do svého chodu, rozdělujeme autonomní roboty do skupin. První skupina řízený stroj, přímé vedení člověka, bez rozhodovacích schopností, kromě člověka nepotřebuje interakci s okolím. Druhá skupina ovládaný stroj, vykonává svojí činnost podle zadaného úkolu, logická rozhodovací schopnost, konečný automat (např. výtah zastaví v požadovaném patře, inteligence s pamětí 1. bit). Třetí skupina regulovaný stroj, dosahuje cíle předem určeným způsobem, dosahuje cíle za různých podmínek různými cestami, analogové rozlišení míry intenzity jevu. Čtvrtou skupinou je autonomní stroj, dosahuje cíle způsobem, který si zvolí. Sice se stále může držet nejpřímější předpokládané cesty, ale nijak jí nepředpokládá, vždy si jí znovu ověřuje, a v případě překážek i sám hledá cestu k dosažení cíle, bez limitu vzdálenosti od původního přímého směru. Poslední skupinu je inteligentní stroj, který si sám volí cíle, člověka nepotřebuje. Toto je prozatím utopie a takový stroj se blíží již k umělé inteligenci.

Podvozky mobilních robotů můžeme rozlišit podle počtu poháněných kol, typu, tvaru. Základní skupiny: diferenciální podvozek-dvě hnaná kola, stabilita je udržována opěrnými body, nebo koly, synchronní podvozek- často 3 kola a každé se 2 stupni volnosti, podvozky se směrovými koly, pásové podvozky, kráčející podvozky.

Pro pohon mobilních robotů se využívají elektromotory. Nejčastěji se používají stejnosměrné, jelikož jsou jednoduché a je snadné napájení například komutátorové, bezkomutátorové, krokové, lze používat i střídavé elektromotory a to synchronní i asynchronní, nebo servomotory, které mají v sobě integrovanou převodovku a inkrementální enkodér.

Zdroje mobilních robotů lze rozdělit do dvou skupin. První skupinu tvoří bateriové články, které mají tu nevýhodu, že po vybití již je nelze znovu nabít a použít. Druhou skupinu tvoří akumulátory, které lze dobíjet a opětovně používat.

Autonomního robota může řídit jednoduchý mikrokontroler neboli jednočip. Jedničky a nuly, které zpracovává program, se převádí jako různé napěťové úrovně. Mikrokontrolery nejčastěji využívají TTL logiku. Kdy napětí 5V prezentuje logickou 1 a napětí 0V logickou nulu. Díky komunikačním pinům může mikrokontroler komunikovat se svým okolím. Komunikace probíhá předáním hodnoty napětí na vstupech tak, že vše pod 2,5 V je logická nula a vše nad tuto hodnotu je logická jednička.

Další důležitou proměnou je čas. Jelikož kdyby robot vykonával svoje úkony v reálném čase tak by se vůbec v prostředí neorientoval. Proto je důležité, aby jednotlivé úkony zpracovával několikrát za vteřinu. Například pokud je frekvence 10 MHz, tak jednočip sleduje okolní svět každých 1/10 000 vteřiny.

Jednočip stačí na ovládání jednodušších robotů. Pokud je robot složitější často jednočip zastává roli zprostředkovatele dat o reálném prostředí pro výkonnější počítače.

Komunikace robota s obsluhou může probíhat pomocí drátové technologie například pomocí sériové linky, USB. Dále může posílat hodnoty na display, který je umístěn na robotovi. Komunikaci lze také zajistit pomocí bezdrátových technologií jako je Bluetooth, kterou jsou vybaveny, jak počítače či mobilní telefony, nebo pomocí Wi-fi technologie.

Způsoby programování a učení robota rozdělujeme do několika skupin. První skupina je přímé programování, podle toho zda robota vedeme, tj. udáváme směr, nebo zadávání povelů z řídicího panelu. Druhá skupina je nepřímé programování, kdy zadáváme křivku dráhy známou z výkresu např. CNC stroje. Třetí skupina je přímé programování robot se

programuje podobně, jako v předchozí skupině, ale je ovlivňován daty, která získává z měření pomocí čidel.

Pro orientaci v prostoru slouží robotovi senzory. Obecně senzor je speciální zdroj informací sloužící pro řídicí systém. Senzory lze definovat jako zařízení, které měří určitou fyzikální veličinu a převádí ji na signál, který dálkově odešle do řídicího systému, nebo do čipu robota. Hlavní parametry senzorů jsou citlivost, práh citlivosti, dynamický rozsah, reprodukovatelnost a chyby senzoru. Výstupní hodnota senzoru je většinou dána jako rozdíl napětí, nebo proudu od nuly, která se většinou zadává.

V robotice se nejčastěji využívají senzory dotykové, tj. pružinová tykadla s mikrospínači pro detekci jejich ohnutí, distanční pro měření vzdálenosti např. sonarovou echolokaci, laserové dálkoměry, vizuální jako jsou kamery, stereo vidění, dále senzory pro radionavigaci či pomocí GPS senzorů a dalších různých senzorů.

Router Edimax BR-6104 KP je určen pro domácí použití, nejen pro sdílení internetu, ale také díky 2 USB portům lze využít, jako tiskový server. Specifikace výrobku: počet WAN portů: 1, počet switche/hub portů: 4, počet USB portů: 2, přenosové rychlosti portů: 100 Mbps, správa routeru: web-manager, vestavěný firewall, filtrace MAC adres, klonování MAC adresy. Router BR-6104 KP má integrovaný procesor AMDtek 5120P s základním taktom 170 MHz. Na tomto routeru běží operační systém Linux kernel verze 2.4.18 s úpravami pro procesor v tomto routeru. Díky tomu, že na routeru je nainstalované jádro Linuxu označil výrobce router, jako open-source zařízení. Do routeru lze pomocí přehrání firmwaru nahrát různé distribuce Linuxu a díky nim lze využít router nejen v domácnosti pro sdílení internetu či tiskáren, ale též jako řídicí počítač například pro autonomního robota.

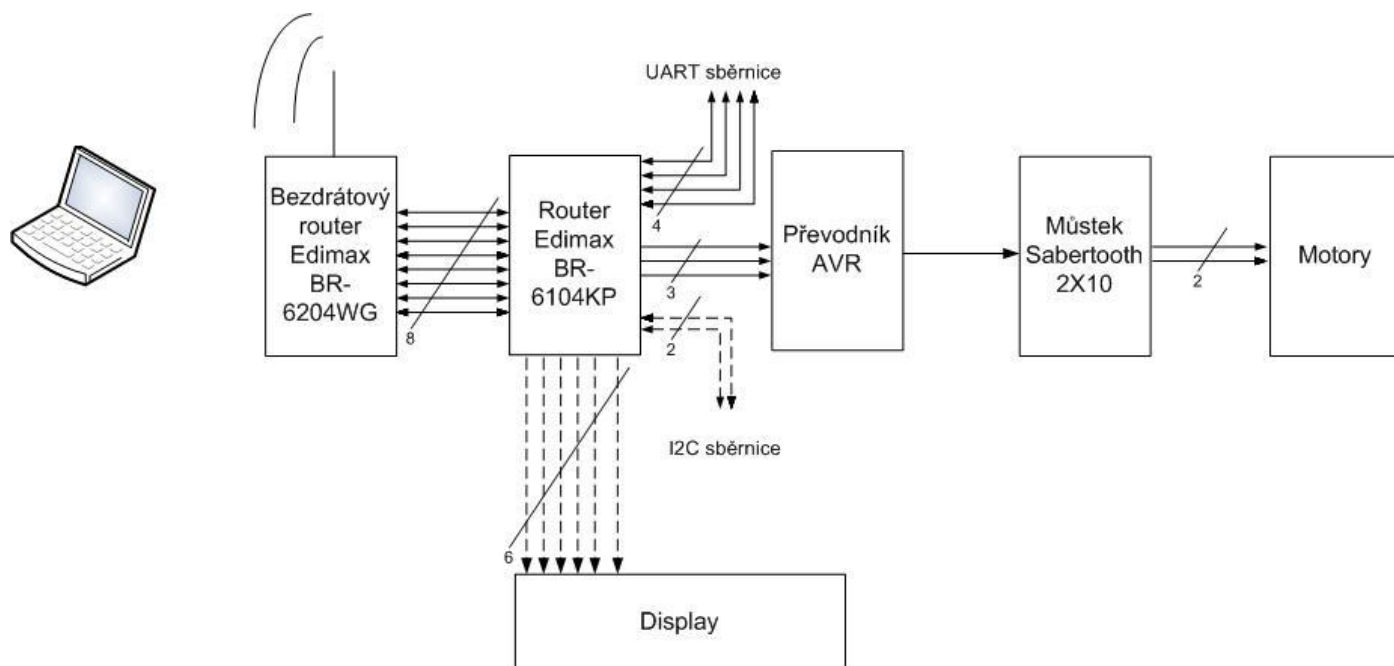
Do routeru Edimax BR-6104 KP lze nahrát 3 distribuce Linuxu kromě oficiálního firmwaru, které na něm stabilně fungují. První distribuce Amilda, která je zaměřená především na původní určení routeru a zdokonaluje jeho chod. Druhá distribuce se jmenuje Midge, která je určená především pro využití hardwarového potenciálu routeru pro jiné než síťové aplikace. Třetí distribuce se jmenuje Sweex, tato distribuce je schopná se načítat z USB flash disku a částečně je postavená na distribuci Midge. Jelikož router Edimax BR-6104 KP má paměť flash 2 Mb, proto je vhodné, využít file systém na flash disku, kdy máme větší prostor a můžeme nahrávat vlastní programy a těmi ovládat například robota. Jelikož distribuce Sweex

je shell distribuce tak pro komunikaci je výhodné používat na operačním systému Windows program PuTTY. Další výhodou této distribuce je, že má FTP server a funguje na něm i skriptovací programovací jazyk PHP verze 5. Dále tato distribuce podporuje programy napsané v jazyku C, kód napsaný v jazyku C se musí následně zkompileovat přímo v příkazové řádce například pomocí programu PuTTY.

Předlohou pro stavbu robota RC-DC byl robot Ju- Ju píseckého radioklubu KUFR, který s robotem účastnili soutěže ROBOTOUR 2009. Poté co jsem se zúčastnil jejich přednášky mě robot Ju-Ju nadchl a rozhodl jsem se, že se pokusím sestavit něco podobného. Řízení robota bude stejný pomocí routeru Edimaxe. I když byl robot Ju-Ju předlohou pro stavbu robota RC-DC nejsou to stejní roboti prozatím ani co se týče autonomie.

2. Řešení

Cílem dlouhodobé maturitní práce je sestavit autonomního robota ze skupiny číslo 1 tj. řízený stroj. Blokové schéma je rozděleno, na šest částí viz obrázky níže, které byly použity pro



realizaci robota.

(Obr. 1 blokové schéma robota RC-DC)

2. 1. Bezdrátový router Edimax BR- 6204WG

Síťový směrovač je určen především pro domácnosti a menší podniky. Směrovač je kombinován s přepínačem (switchem) a bezdrátovým přístupovým bodem pro sdílení připojení pomocí technologie Wi-Fi. Dle specifikace výrobce lze na připojení být až 253 uživatelů, směrovač dále poskytuje vyšší bezpečnost připojení díky vestavěnému firewall, podporuje šifrované spojení a další pokročilé funkce jako jsou NAT, DHCP, DDNS. Směrovač podporuje jednoduchou správu pomocí webového klienta, monitoring sítě, řízení přístupu například pomocí povolení MAC adres počítačů. Rozhraní směrovače: 1x RJ-45 WAN Ethernet 10/100 Mbps, 4x RJ-45 LAN Ethernet 10/100 Mbps, 1x SMA konektor pro

připojení antény na Wi-Fi komunikaci. Podporuje směrovací protokoly RIP v1, RIP v2 a statické směrování. Cena směrovače je 599 Kč. Tento směrovač je první část blokového schéma. Jeho úkol je zajišťovat komunikaci mezi uživatelem a síťovým směrovačem Edimax BR- 6104KP.

Komunikace probíhá obousměrně pomocí paketů ve dvou fázích. První fáze je komunikace mezi uživatelem, který se připojí na směrovač Edimax BR- 6204WB pomocí bezdrátové technologie Wi-Fi, přes kterou odesílá svoje příkazy pro směrovač Edimax BR- 6104KP. Druhá fáze je odeslání příkazu, pomocí paketu do Edimaxu BR- 6104KP, to zajišťuje stálé spojení s druhým směrovačem pomocí UTP kabelu konektoru RJ-45 Ethernet 10/100 Mbps.

Bezdrátový router Edimax BR- 6204WG umožňuje několik režimů pro bezdrátové připojení je to: přístupový bod, který je použit při nastavení, dále jsou to režimy stanice AD HOC hodí se pro připojení více uživatelů na jedno zařízení například pro sdílení souborů či hraní her v síti, most pro spojení dvou částí sítě, nebo v režimu WDS, který slouží pro komunikaci dvou bezdrátových směrovačů mezi sebou bezdrátově.

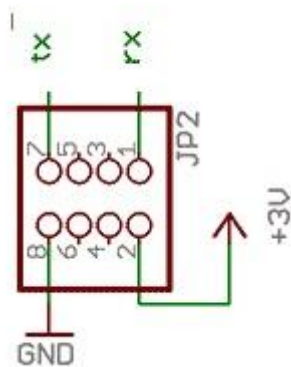
Dále se nastavuje skupina Wi-Fi karet, které budou podporovány pro komunikaci, nastaven do režimu 802.11 b+g, tak aby se na něj mohli připojit počítače jak se standardem 802.11b tak i 802.11g v pásmu 2,4 GHz. Dále se nastavuje ESSID což je vlastní jméno sítě tak, tak jak ho budou vidět uživatelé, kteří se na něj budou chtít připojit. Číslo kanálu je nastaveno na 9, je to libovolné číslo v rozmezí 1- 13. Je vhodné volit takový kanál, který není již v okolí zabrán jiným bezdrátovým síťovým zařízení, pokud to umožňuje technologie tak si ji směrovač vybere sám, ale pokud to neumí, musí uživatel vybrat kanál sám a s tím mu může pomoc například program Network Stumbler.

Poté je vhodné nastavit zabezpečení, aby se do sítě nepřipojit někdo cizí, je použito zabezpečení WEP, s 64 bitovým šifrovaným klíčem, toto je nejjednodušší zabezpečení, které zvládají i starší Wi-Fi karty. Síťový směrovač Edimax BR- 6204WG vyžaduje minimální konfiguraci uživatelského PC s Wi-Fi kartou, internetový prohlížeč Internet Explorer 4.0 a vyšší, nebo Netscape Navigator 4.7 a vyšším, doporučuji operační systém Windows XP a vyšší s nainstalovanými aktualizacemi, jak operačního systému, tak ovladačů bezdrátové síťové karty. Lze se připojit i pomocí mobilního zařízení, které disponuje technologií Wi-Fi.

Připojení je snadné v několika krocích. Po zapnutí a načtení směrovače (při zapínání se rozsvítí všechny ledky, které následně zhasnou asi na 10 sekund a poté se znovu rozsvítí používané ledky a směrovač naběhne), uživatel vyhledá síť v okolí, síť se jmenuje **RC-DC-AP**, po připojení bude vyžadovat heslo, které je: **RC-DC**. Pokud se bude uživatel chtít podívat do konfigurace síťového směrovače, musí do internetového prohlížeče zadat adresu **192.168.2.1**, poté se zobrazí přihlašovací tabulka, kde uživatelské jméno je: **admin** a heslo: **1234**.

2. 2. Router Edimax BR- 6104KP

Směrovač Edimax BR- 6104KP je použit, jako ovládací prvek svého autonomního robota prvního stupně. Aby se mohl ovládat směrovač, musel být do něj nahrána jiná Linuxová distribuce. Pro tuto operaci jsou na desce směrovače u procesoru připraveny dvě sady padů pro připojení dvou pinových lišt s označením JP1 a JP2. Směrovač, který škola zakoupila, měl už piny na JP2 vyvedeny, a proto se nemusely dodatečně osazovat.



(Obr. 2 zapojení pinů na výstupu JP2- zdroj: (ŠERÝCH, 2008))

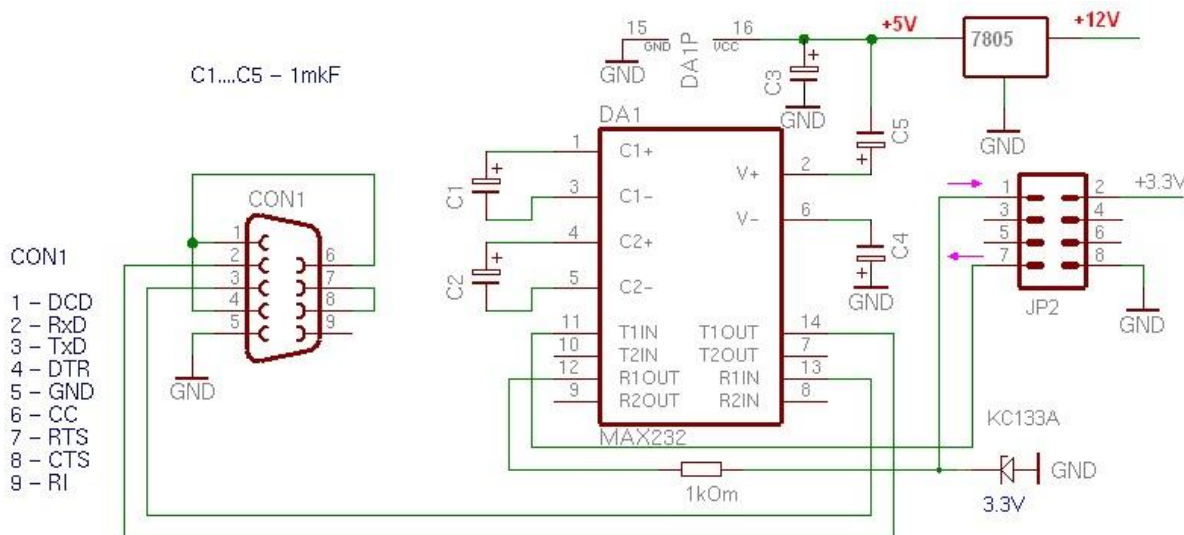
Piny tx a rx pracují v TTL technice, a proto je nutné pro komunikaci s počítačem na rozhraní COM převodem na úroveň RS232. Lze na to využít převodník MAX232, nebo jiný kompatibilní. Převodník je sestaven podle schématu, který byl na stránkách, kde se autor přehrávání firmwaru směrovačů společnosti Edimax s procesory AMD5120. Tento převodník byl použit pouze jednou, proto byl sestaven do kontaktního nepájivého pole. Po sestavení je možno přistoupit k nahrání Linuxové distribuce do směrovače.

Poté se připojil převodník k počítači a k směrovači Edimax BR- 6104KP, nahrávání Linuxové distribuce do směrovače probíhá pomocí hyper terminálu, pro správné připojení k směrovači je nutné zadat parametry připojení následovně:

Rychlost:	115 200 kbps
Databity:	8
Stopbity:	1
Parita:	N
Řízení:	Žádné

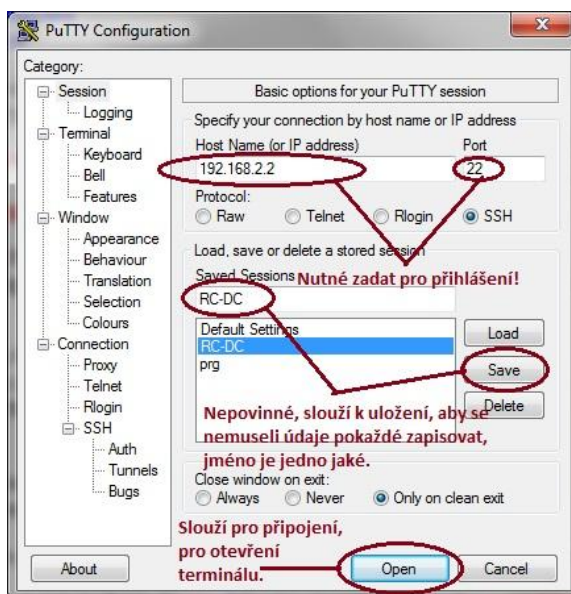
(Tab. 1, nastavení terminálu pro komunikaci se směrovačem)

(Obr. 3, schéma zapojení převodníku MAX232- UART, zdroj: (MOSKOVETS, 2007))



Poté pokud je vše v pořádku zapojené tak se po krátké chvíli zobrazí na monitoru AMD5120 Boot: zde je nutné 3x za sebou stisknout klávesu mezerník, aby se nám zobrazilo menu s nabídkou možností pro nahrávání firmwaru do směrovače buď do flash paměti nebo do sdrám (pouze pro otestování), vybereme tedy volbu první, pro nahrání firmwaru, poté vybereme firmware, který chceme nahrát. Postupně se firmware nahrává a na konci se zase objeví základní nabídka bootloaderu, kde vybereme konec a směrovač manuálně vyresetujeme a může využívat již jinou distribuci Linuxu, zde bych chtěl upozornit, že je nutné restarty provádět odpojením a zapojením napájení, i když je na směrovači tlačítko Reset (vedle napájení) nejsem si jistý, že stále funguje, nebo plní svoji funkci, a proto není vhodné ho testovat!

Zvolenou distribucí je Sweex, se kterou už měli zkušenosti členové píseckého radioklubu, kteří jej též použili na svém robotu Ju-Ju, který byl jakousi předlohou mého robota RC-DC. Jako úložiště dat pro distribuci Sweex je využita 512 Mb flash disk Kingston, která nabízí dostatečný prostor, jak pro samotnou distribuci, tak pro programy, které ovládají pohyb robota. Minimální kapacita flash disku, která by měla dostačovat je 128 Mb. Flash disky, které se běžně prodávají, jsou určeny pro použití v systému Windows, proto jejich filesystem většinou bývá FAT32, proto před tím, než na flash disk můžeme nahrát jakoukoliv distribuci Linuxu, musíme nejprve filesystem přepsat na Linuxový. Můžeme použít libovolnou distribuci Linuxu, kterou máme nainstalovanou v počítači, nebo využít speciální program, který dokáže na discích vytvořit různé filesystemy například program Partition Magic. Filesystem musí být formátu ext3, aby se na flash dala nahrát distribuce Sweex. Poté jsem se seznámil s ovládáním distribuce Sweex, pomocí programu PuTTY. PuTTY slouží, jako terminál pro připojování na různé servery. Pro připojení je nutné zadat do programu IP



adresu, kam se připojujeme: **192.168.2.2**, číslo portu: **22**, což je komunikace pomocí **SSH**, poté co se údaje vyplní a stiskne se tlačítko Open (viz obrázek), se objeví na obrazovce terminál, který slouží pro komunikaci s operačním systémem Sweex.

(Obr. 4, screen ovládacího programu PuTTY)

Pro práci s operačním systémem je nutné zadat přihlašovací jméno: **root**, a heslo: **midge**. Psaní příkazů je velmi podobné, jako v jiných Linuxových distribucích, které pracují v terminálu. Terminál nám umožňuje práci se systémem, jako vytváření uživatelů, změny hesel, nahlížení do souborů, kompilování souborů, ovládání robota, display. Proto, abych byl

nějak schopný ovládat například motory, nebo display, bylo nutné najít nějaké výstupy, které bych byl schopný ovládat.

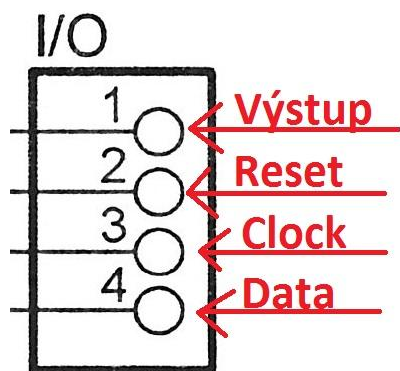
Na výběr bylo ze tří možností. První možnost je ovládání pomocí LED diod, které jsou původně určeny pro indikaci komunikace jednotlivých ethernetových portů. Další možností je ovládání pomocí sběrnice buď I2C, UART JP1 a JP2. Poslední možností by byla komunikace pomocí ethernetového portu přes UTP kabel. Po nastudování dostupných materiálů bylo rozhodnuto, že nejrychlejší a nejsnadnější bude ovládání pomocí LED diod. Na jednotlivé LED diody byly napájeny kabely, které poslouží, jako vstupy do dalších zařízení. Další výhodou je, že tvůrci distribuce Sweex, už napsali program pro ovládání led diod, a proto je velmi snadné je ovládat přes terminál. Do konzole na terminálu se napíše příkaz: **echo "led switch 1" > dev/gpiox** – kde x je číslo led diody, poté, co se takto dioda inicializuje, zadávají se již příkazy: **echo "led off" > dev/gpiox** pro vypnutí, **echo "led on" > dev/gpiox** pro zapnutí a **echo "led blink 1000" > dev/gpiox** slouží pro blikání diody, kdy číslo určuje, kolikrát blikne za 1 sekundu.

2. 3. Převodník AVR

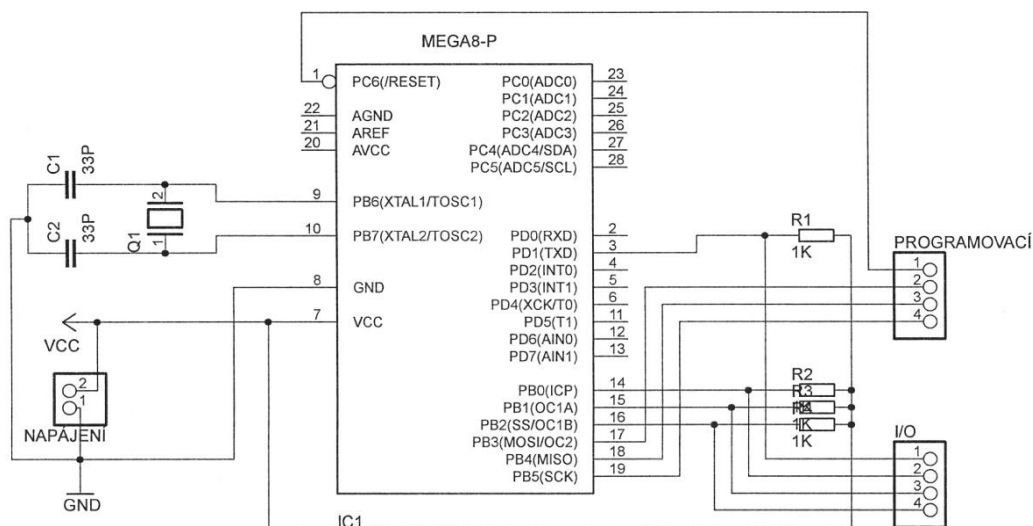
Pro správnou funkci můstku Sabertooth 2X10 v režimu zjednodušený sériový přenos je důležité použít převodník, který převede vstupní signál, nebo signály na signál, kterému rozumí můstek. Další důvod, proč musím použít převodník, je že ovládám motorčky pomocí led diod na řídicím směrovači. Převodník AVR je v mém případě tří vstupný, vstupy jsou označené reset, data, clock a má jeden výstup, který je vyveden do můstku Sabertooth 2X10. Celá elektronika je napájena 5 volty, které potřebuje i můstek. Proto je napájení vyřešeno připojením USB kabelu do volného USB portu na řídicím směrovači. Toto bylo elegantní řešení, jak vyvést ze směrovače 5 voltů, aniž by muselo dojít k hardwarové úpravě směrovače. (Obr. 9, zapojení převodníku AVR)

Schéma zapojení se skládá z mikroprocesoru MEGA8-P, dále 4 rezistory každý s odporem 1 kilo Ohm, 2 kondenzátory s kapacitou 470 pF a krystal 12 MHz, celkem 10 pinů, kdy 2 piny jsou určené napájení, 4 slouží pro vstupy led diod řídicího směrovače a jeden výstup do můstku pro řízení motorů a 4 pro případné programování čipu přímo na desce plošného spoje.

Převodník funguje tak, že má nadefinovanou a deklarovanou proměnnou, která slouží, jako buffer do které postupně přicházejí buď logická nula, nebo jednička. Poté co se do bufferu zapíše 8 znaků, které tvoří decimální číslo od 1 do 255 v binárním kódu, odešle se hodnota bufferu na výstup do můstku, který podle složení logických nul a jedniček pozná, který motor má spustit a jestli dopředu nebo dozadu a jakou rychlostí. Pro správné fungování je důležité správně připojit vstupy a výstup na piny určené pro vstupy/výstup. Na pin číslo jedna jde



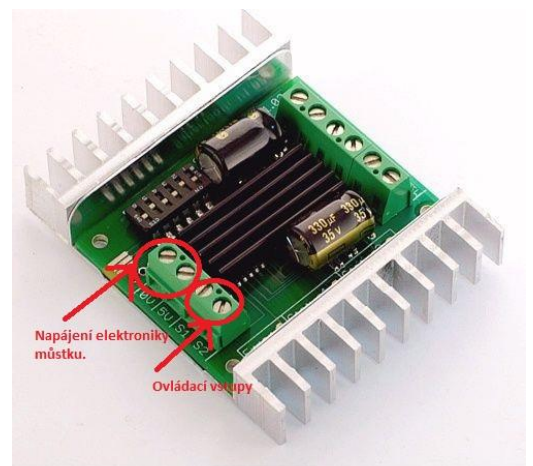
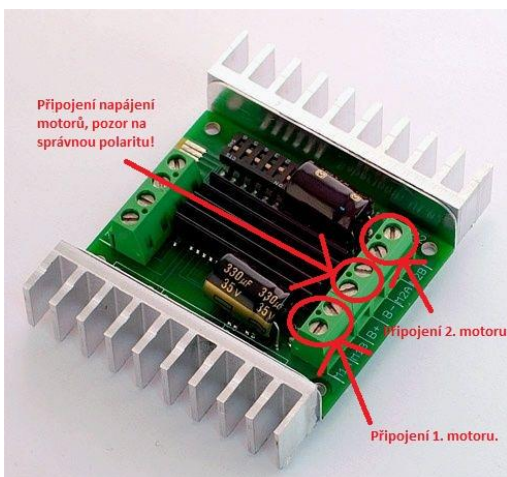
výstup do můstku, na pin číslo dva jde vstup označen jako reset (led dioda číslo 1), na třetím pinu je vstup clock (led dioda číslo 11) a na čtvrtém pinu je vstup data (led dioda číslo 8) (Obr. 10, zapojení vstupů a výstupu na převodníku)



2. 4. Můstek Sabertooth 2X10

Můstek Sabertooth 2x10 od výrobce Dimension Engineering umožňuje najednou ovládat 2 motory o napětí 2x24 V, při proudu 10A. Napětí motorů může být od 6 do 24 V pro motory, výstupní proud je 10A trvale a až 15A krátkodobě. Výrobce vyrábí ještě dva podobné můstky, kdy jeden má výstupní napětí 5A a druhý 25A. Jelikož motory, které používám pro pohon robota, dosahují maximálního proudu cca 12A, které se mi povedlo naměřit s co maximální zátěží motoru. Byl vybrán můstek právě s proudem 10A, jelikož se mi zdálo, že 5A by nemusel vydržet zátěž a 25A je trochu přebytný a hlavně cena je asi dvojnásobná. Můstek dále disponuje proudovou a tepelnou ochranou, dále chrání lithiové akumulátory před nadměrným vybitím.

Můstek má z jedné strany 4 vstupy, kdy dva jsou určeny pro napětí +5V a 0V, které řídí elektroniku můstku a dva vstupy pro signály označené S1 a S2. Na druhé straně můstku je celkem 6 vstupů, kdy prostřední jsou určeny pro připojení napájení označené B- a B+, je nutné dbát na správné připojení napájení, jelikož otočením napájení by mohlo dojít k nenávratnému poškození můstku. Po stranách jsou vždy dvojice vstupů pro přívody motorů označeny, jako M1A, M1B a M2A, M2B (viz obrázky níže).



(Obr. 5, pohledy na můstek SaberTooth 2X10, zdroj: (Dimension Engineering LLC, 1999))

Můstek má celkem čtyři možnosti ovládní motorů. Volba se provádí nastavením jednotlivých přepínačů na přepínači, podle manuálu. První možnost je řízení pomocí analogového signálu. Je to nejjednodušší ovládní motorů v nastavování napětí v rozmezí od 0V do 5V. Lze nastavovat pomocí potenciometru, PWM nebo mikrokontroléru, který musí mít RC filtr.

Druhá možnost je ovládání R/C vstupu, kdy R/C využívá dva vstupy jeden pro směr a druhý pro rychlost motorů. Lze nastavit i časovou prodlevu. Pokud je nastavena umožňuje kontrolovat motory, aby pokud ztratí signál tak dále nepokračovali. Pokud není prodleva nastavena, motory pokračují dále, i když ztratí signál. Tento režim se používá pro zařízení na bázi Basic Stamp, nebo nízko rychlostní mikrokontroléry.

Třetí režim je zjednodušený sériový. Využívá TTL úroveň RS-232, kdy jsou data posílána přímo do motorů. Motory mohou být ovládány přímo z počítače přes sériovou linku, kdy se musí nastavit přenosová rychlost, nebo ze vstupu ze sériového převodníku. Převodník ovládá motory posíláním 8 bytových znaků, kdy každý motor má vyhrazeno 7 bitů pro ovládání. První motor 1 zpětný chod, 64 motor stojí, 127 motor jede dopředu. Druhý motor 128 zpětný chod, 192 motor stojí, 255 motor jede dopředu.

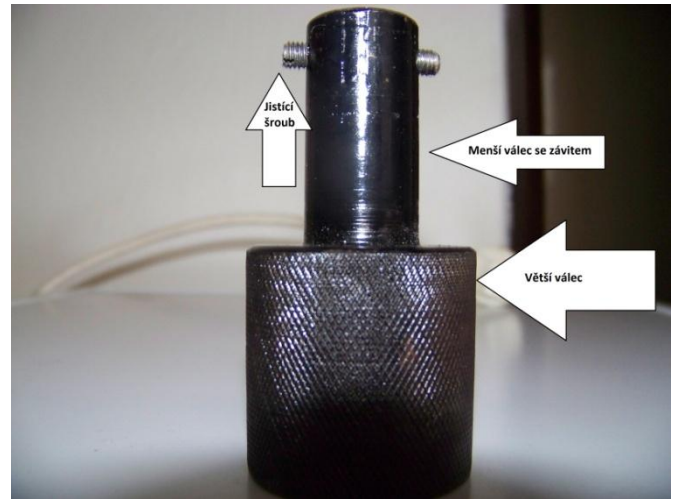
Poslední režim je paketová sériová komunikace. Tento mód se používá pouze, pokud máme propojeno více můstků, které ovládá jediný počítač. Příkaz se zabalí do speciálního paketu, který se poté rozesílá do můstků a podle záhlaví, se příkaz vykoná na můstku pro, který je určen. Tohoto zapojení by se dalo využít, kdybychom měli motory pro pohyb robota a další motor nebo motory, kterými budeme ovládat například pohyblivé rameno.

Jelikož ovládám motory pomocí rozsvěcování a zhasínání led diod rozhodl, jsem se po testování toho můstku, že využiji mód třetí zjednodušený sériový přenos. Motory lze ovládat pomocí příkazů, které lze zadávat v příkazové řádce programu PuTTY, nebo ovládat motory pomocí webového prohlížeče, kdy se pomocí skriptu ovládají led diody.

2. 5. Motory

Pohon robota je tvořen dvěma motory, které byly vyndány ze starého automobilu Škoda Favorit. S výběrem vhodných motorů mi pomohl můj otec. Motorek ze stěrače je tvořen dvěma ze dvou základních částí. Část elektrická a část mechanická, kdy elektrická část je převodovaná na mechanickou, která poté otáčí stěračem automobilu zpravidla doprava a doleva. Tyto motorky jsou tři rychlostní, proto mají tři vodiče, kdy jejich kombinací se ovládá rychlost otáčení. Motory byly vybrány, protože jsou napájeny 12 volty, a proto jsou vhodné pro mé řešení, kdy používám 12 voltové dobíjecí akumulátory.

Směr otáčení se mění záměnou plusu a mínusu na napájení. Aby motorky mohli sloužit pro pohyb robota, musel být zhotoveny odlitky, které přenášejí otáčky motorku na kolo. Odlitek je tvořen dvěma válci, kdy na menším je udělán závit, který se natočí na mechanický výstup motorku a je zajištěn dvěma zajišťovacími šrouby, kteří jsou proti sobě. Větší váleček slouží pro přenos na kolo, proto jsou na něm vyfrézované rýhy, které zajistí otočení kola, kdyby nebyl povrch vyfrézovaný, nedocházelo by k přenosu síly.



(Obr. 6 a 7, náhled na motůrek a odlitek)

Při prvním stavbě robota, bylo naplánováno, využití všech tří rychlostí pro pohyb robota. Jelikož byly ovládané, pomocí pole stykačů, bohužel se nepovedlo zprovoznit pole stykačů, tak aby řídily motory. Musel být nalezen jiný způsob řízení, náhradou mi byl můstek Sabertooth 2X10, který umožňuje i regulaci rychlosti, a proto používám na motorkách pouze jednu tu nejvyšší rychlost.

2. 6. Display

Operační systém Sweex umožňuje jako jeden z výstupů display. Display musí obsahovat čip, který vychází z architektury čipu HD44780. Je vhodné použít display, který má dva řádky po šestnácti znacích, kdy pokud se znak nevejde na první řádku, tak se zalomí text a napíše se na řádku druhou. Pro připojení k směrovači slouží sběrnice, která slouží pro zobrazování stavů sítě pomocí led diod. Sběrnice display se připájí na záporný pól led diod podle tabulky níže.

LCD pin	Function	Data sheet LED number	Circuit board LED number
1	Vss earth		
2	Vdd +5 volts		
3	5Kohm contrast potentiometer centre connection (ends go to 0 and +5 volts)		
4	RS	6	11
5	R/W - connect to earth		
6	E - enable	12	12
7	0 - data line	NC	
8	1 - data line	NC	
9	2 - data line	NC	
10	3 - data line	NC	
11	4 - data line	8	4
12	5 - data line	9	6
13	6 - data line	10	8
14	7 - data line	11	10
15	LED +5 volts (use an external 5 volt supply)		
16	LED Earth (the Sweex wall power supply is only 500mA)		

(Obr. 8, tabulka zapojení display, zdroj: (Sunspot, 2006))

Při zapojení display nastává jedna komplikace, jelikož potřebuje napájení +5 voltů, ale led diody jsou napájeny pouze +3.3 voltu, nelze je použít, jako zdroj a musí se pro něj přivést napájení z jiného místa například USB port. Jelikož já používám na ovládání motorů led diody 1, 8 a 11 nemohu již zapojit display. Pro zapojení by se muselo zjistit, jak přepsat ovladač display, poté by se snad dal display zapojit na jiné piny a pokusit se ho zpustit.

2. 7. I2C sběrnice, UART sběrnice

Tyto dvě sběrnice by se daly využít pro připojení čidel, které by robotovy sloužily pro získávání informací o okolním světě. Sběrnici UART byla již využita, při flashování firmware směrovače Edimaxu. Pracuje na podobném principu jako sběrnice RS-232. Abych mohl sběrnici UART využít pro ovládání jakéhokoliv čidla, musel bych nejdříve napsat ovladač pro operační systém Sweex, který by dokázal přijímat a vysílat data do čidla. Jelikož nemám se psaním ovladačů žádnou zkušenost a moje znalosti o operačních systémech na bázi Linuxu jsou malé, nepouštěl jsem se do tvorby čidla, které by pracovalo na této bázi.

Další sběrnice, která se dá použít pro připojení čidel je sběrnice I2C. Tato sběrnice není standardně vyvedena ze směrovače, a proto se musí dodatečně dodělat. Výhodou oproti sběrnici UART je, že už jsou na sběrnici I2C napsány ovladače a jsou implementovány přímo v operačním systému Sweex. Dále sběrnice I2C se hodně používá, a proto by asi nebyl problém, sehnat čidla pomocí kterých by robot získával informace o okolním světě. Podklady pro dodělení sběrnice I2C lze najít na webových stránkách (Sunspot, 2006).

2. 8. Ovládání robota

Úkol byl vytvořit autonomního robota první skupiny tj. řízeného uživatelem, tak pohyb robota závisí plně na rozhodnutí uživatele, který ho zrovna ovládá. Ovládání je dvojí buď pomocí odeslání příkazu, nebo pomocí grafického rozhraní. Odesílání příkazu funguje, jak v příkazové řádce programu PuTTY, nebo zadáním přes webový prohlížeč. Ovládání pomocí grafického rozhraní funguje pouze ve webovém prohlížeči. Pro správnou funkci grafického rozhraní doporučuji používat prohlížeč Mozilla Firefox, nebo Google Chrome. V žádném případě nedoporučuji používat Internet Explorer, ve kterém grafické rozhraní nebude fungovat.

Pro ovládání pomocí příkazové řádky programu PuTTY je přímo na flash disku uložen program „rizeni“, který slouží k ovládání. Aby program sloužil, jako ovládací musí se nejprve zkompileovat přímo ve směrovači, nebo pomocí tool-chain, který je na Linuxovém operačním systému, více informací lze získat přímo ze stránek (Sunspot, 2006). Poté co program zkompilejeme lze už ho využívat pro ovládání robota. Po připojení pomocí programu PuTTY se zadá příkaz „**cd c_dev/ovladani**“, který nás nasměruje do složky, kde je uložen program pro ovládání robota. Poté se zadává příkaz kam má robot jet „**./rizeni forward/backward/left/right**“. Řídící program se skládá z několika částí. V první části je výpis knihoven, které se volají při běhu programu, dále se definují, které led diody se budou využívat, definují se názvy motorů a ihned se i deklarují.

```
#include <unistd.h>
```

```
#include <fcntl.h>
```

```
#include <sys/ioctl.h>
```

```
#include <string.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define DATA_LED "/dev/gpio8"
```

```
#define CLOCK_LED "/dev/gpio11"
```

```
#define RESET_LED "/dev/gpio1"
```

```
#define CRR_A 0
```

```
#define CRR_B 0
```

(výpis kódu programu číslo 1)

Poté následuje část programu ve kterém se určuje, databit, clockbit a resetbit. Tyto proměné jsou logického typu a nabývají hodnot logické jedničky nebo nuly, podle toho zda je příslušná led dioda aktivovaná nebo ne, zde se na to používá jednoduchá podmínka na otestování stavu led diody. Aby se mohla led dioda mohla rozsvítit musí mít proměná f parametr O_RDWR, který nám určuje, že proměnná bude přístupná ke čtení a zápisu, bez tohoto parametru by nešlo ovládat led diody.

```
int DataBit(bool bitt){
    int f = open(DATA_LED, O_RDWR);
    if(bitt)
        write(f, "led switch 1", 12);
    else write(f, "led switch 0", 12);
    close(f);
}
```

```
int ClockBit(bool bitt){
    int f = open(CLOCK_LED, O_RDWR);
    if(bitt)
        write(f, "led switch 1", 12);
    else write(f, "led switch 0", 12);
    close(f);
}
```

```
int ResetBit(bool bitt){
    int f = open(RESET_LED, O_RDWR);
    if(bitt)
        write(f, "led switch 1", 12);
    else write(f, "led switch 0", 12);
    close(f);
}
```

(výpis kódu programu číslo 2)

Další část programu je sériový přenos, který slouží k přenosu informace do převodníku AVR. Tato část kódu se skládá z několika částí. V první části se definuje pole, které je logického typu a jeho velikost je osm bitů, protože potřebujeme právě osm bitů na převod dekadického čísla do binárního. Pro převod slouží jednoduchý switch, který postupně dělí číslo, které mu pošle hlavní část programu. Poté co číslo vydělí dochází k výpisu na obrazovku, slouží to pro

kontrolu, zda se tam opravdu napsalo to číslo, které mělo. Poté se testuje zda byl poslán příkaz reset tak se už nepokračuje, pokud ne tak se pokračuje dál k odeslání příkazu.

```
int SerialCommand(int i_byte){
    bool data[8];
    //Convert int to binary data
    int z = i_byte;
    int d;
    for(int i = 0; i < 8; i++){
        switch(i){
            case 0: d = 128; break;
            case 1: d = 64; break;
            case 2: d = 32; break;
            case 3: d = 16; break;
            case 4: d = 8; break;
            case 5: d = 4; break;
            case 6: d = 2; break;
            case 7: d = 1; break;
        }
        //printf("[%d/%d]", z, d);
        if(z / d > 0){
            z = i_byte % d;
            data[i] = true;
            printf("1,", z);
        } else {
            data[i] = false;
            printf("0,", z);
        }
    }
    printf("\n");

    //Reset
    ResetBit(true);
    ResetBit(false);
    usleep(1000);

    //Send
    for(int i = 0; i < 8; i++){
        if(data[i] == false)
            DataBit(true);
        else
```

```

        DataBit(false);
        ClockBit(true);
        ClockBit(false);
        usleep(1000);
    }
}

```

(výpis programu číslo 3)

Po té následuje poslední část programu. Je to samotný program, který se skládá ze 6 podmínek. Každá podmínka má dva argumenty, kdy druhý argument je slovo, které se napíše do řádky kam chceme, aby robot jel. Jelikož by byl problém s českými znaky bylo rozhodnuto pro zadávání anglických slov forward, backward, left, right. Podmínky čekají, jaké slovo se objeví v argumentu a podle toho vyšlou příkaz do předcházející části kódu, který se stará o přenos do převodníku AVR. Jelikož můžeme díky můstku Sabertooth 2X10 ovládat i rychlost motorů nejsou v jednotlivých podmínkách nastaveny maximální hodnoty, ale vždy takové hodnoty, aby se oba motory otáčeli stejnou rychlostí. Poslední podmínka slouží jako nápověď, pokud by se zadal špatný příkaz, aby vypsala do příkazové řádky příkazy, které jsou zpracovatelné.

```

int main(int argc, char *argv[]){
    //Send
    if(strcmp(argv[1], "stop") == 0){
        printf("Stopping...\n");
        SerialCommand(0);
    }
    else if(strcmp(argv[1], "forward") == 0){
        printf("Going forward...\n");
        SerialCommand(9 + CRR_A);
        SerialCommand(128 + CRR_B);
    }
    else if(strcmp(argv[1], "backward") == 0){
        printf("Going backward...\n");
        SerialCommand(127 + CRR_A);
        SerialCommand(255 + CRR_B);
    }
    else if(strcmp(argv[1], "left") == 0){
        printf("Turning left...\n");
        SerialCommand(1 + CRR_A);
    }
}

```

```

        SerialCommand(192 + CRR_B);
    }
    else if(strcmp(argv[1], "right") == 0){
        printf("Turning right...\n");
        SerialCommand(64 + CRR_A);
        SerialCommand(128 + CRR_B);
    } else printf("Usage: ./rizeni [forward|stop|backward|left|right]\n");
}

```

(výpis programu číslo 4)

Ovládání pomocí webového prohlížeče je rozděleno celkem na tři části. První část je webová stránka, která slouží pro zadávání směru a rychlosti robota. Webová aplikace je uložena přímo na flash disku, na kterém je uložen celý operační systém. Pro zobrazení stránky, musí být uživatel připojen k robotovi. Poté zadá do prohlížeče adresu: <http://192.168.2.2/rc-dc/console/>, kde uživatel klikne na odkaz Controller, který pomocí javascriptu načte formulář, který umožňuje ovládání robota. Pohyb robota lze ovládat pomocí počítačové myši, která je na ovládání přesnější, nebo pomocí kláves. Klávesa *w* slouží pro pohyb dopředu, klávesa *s* pro pohyb dozadu, klávesa *a* pro pohyb doleva, klávesa *d* pro pohyb doprava a klávesa *x* pro okamžité zastavení pohybu robota. Jelikož se stisknuté znaky porovnávají s ASCII tabulkou je nutné písmena zadávat malá, nikoliv velká. Javascript je napsán tak, že sleduje aktuální polohu kurzoru myši, podle které potom posílá hodnoty pro pohybové motory robota. Je tam ošetřeno, aby se kurzorem myši nemohlo posunout, mimo rozhas pro ovládání robota. Rychlost motorů a síla otáčení se vypočítává za využití vzorce pro výpočet úhlu v pravoúhlém trojúhelníku. Výběr motorů závisí na aktuální X souřadnici kurzoru myši, kdy je to rozděleno na dva intervaly, pokud je menší jak 0 a větší jak 0. Na tom závisí výkon motorů, pokud je to první interval tak levý motor má sílu podle úhlu od bodu 0,0 a pravý motor plný výkon, pokud je to druhý interval tak je to opačně, pokud se souřadnice X rovná 0 pak mají oba motory plný výkon. Zda se budou motory otáčet dopředu, nebo dozadu závisí na souřadnici Y kurzoru myši, kdy pokud je hodnota větší, jak 0 tak se motory otáčejí dozadu. Bod 0,0 souřadnice je uprostřed kruhu, ve kterém se pohybuje kurzor myši. Údaje o aktuální rychlosti motorů se posílají do souboru `ajax_controller.php`, který má za úkol předávat hodnoty na led diody směrovače, pro pohyb robota. Data se posílají 200 milisekund, což způsobuje pomalejší reakce, a proto robot nemůže jet úplně rovně, jelikož zde nemůže dojít ke korekci odesílaných hodnot, jako tomu je při ovládání v příkazové řádce. Každý motor má pro svoje ovládání rozsah 128 bitů, proto se do souboru posílají hodnoty v rozmezí od -63 do +63, tak aby se

vešli do rozsahu, kdy -63 je plná rychlost dozadu a +63 plná rychlost dopředu, dále je ošetřeno, aby se nedaly posílat hodnoty, které by nebyly v tomto rozsahu. Pro motor A se hodnoty dopočítávají porovnání od 64, kdy motor stojí, pro motor B od 192, kdy motor stojí. Poté se výsledek pošle zpět do formuláře pro kontrolu uživatele, že se příkaz vykonal pomocí souboru `rcdc_controller.js`. Pro odeslání dat na ledky se při provádění skriptu `ajax_controller.php` spouští soubor `rcdc_controller.php`, který obsahuje kód podobný kódu, který je napsán pro příkazovou řádku a slouží pro aktivaci motorů.

2. 9. Konstrukce podvozku

Podvozek robota je tvořen dvěma základními částmi. První část je kovový rám, který původně sloužil, jako součást kočárku. Kovový rám má tvar obdélníka s rozměry 503 mm na 412 mm a v každém rohu je přivařen kovový obdélník s dvěma otvory pro šrouby. Dále jsou na rámu připevněné dvě kola, která se otáčejí kolem své osy. Kola jsou původní z kočárku o průměru 300 mm.

Druhou část podvozku tvoří PUR panel o rozměrech výška X šířka X délka: 44 x 412 x 503 mm. Na spodní části panelu jsou umístěná dvě otáčecí kolečka. Kolečka jsou podobná těm, které se používají například u nákupních košů v obchodech. Vrchní část panelu můžeme obrazně rozdělit na dvě části. V první části se nacházejí motůrky ze stěračů, které slouží jako pohon. K panelu jsou tyto motůrky připevněné objímkou, která se používá na uchycení trubek. Aby motůrky měli lepší přilnavost ke kolům, jsou ještě připevněny kovovým páskem, který je ohnut do úhlu 90°. Dále v této přední části se nachází místo, které je určeno pro napájecí akumulátory, je to obdélník o rozměrech 165 x 154 mm, který je utvořen z hliníkového plechu, který je zahnut do úhlu 90°. Podél motůrků jsou vedeny dvě elektrikařské lišty, ve kterých jsou schovány kabely od motůrků.

V druhé části se nachází elektronika. Kdy nalevo je směrovač Edimax BR- 6204WG, který slouží pro komunikace s uživatelem. Napravo se nachází Edimax BR- 6104KP, který je ovládací prvek celého robota. Mezi nimi je umístěná plastová elektrikařská krabička, ve které je umístěn převodník AVR a bezpečnostní STOP tlačítko, které slouží k rychlému zastavení pohybu robota v případě nebezpečí. Hned za krabičkou je umístěná druhá, ve které je umístěn převodník Sabertooth 2X10.



(Obr. 9 a 10, rozložení částí robota na vrchní straně PUR panelu)

3. Závěr

Práce vytvořená do soutěže SOČ bude po skončení soutěže umístěna v dílnách SPŠ a VOŠ Písek, kde bude sloužit, jako názorná pomůcka ve vyučování. Žáci se budou moci seznámit s ovládáním robota v operačním systému Linux.

Do budoucna by bylo dobré robota rozšířit o další čidla, která by robotovi umožňovala se více osamostatnit. Jelikož má Edimax BR-6104KP celkem rychlý procesor je možné, do budoucna uvažovat o dodělení kamery, která by sloužila, jako čidlo.

Cíl práce bylo sestavit autonomního robota, který by řídil směrovač Edimax BR-6104KP s Linuxovou distribucí, který by dokázal projet bludištěm. Robot dokáže jezdit uvnitř budovy, což se hodí při praktické výuce, která probíhá v učebnách. Dále se může pohybovat i ve venkovním prostředí, i když jeho pohyb je složitější a poradí si jen s menšími nerovnostmi povrchu. Proto, aby si poradil i s většími nerovnostmi, bylo by nutné vymyslet jiný podvozek například větší nepoháněné kola, nebo přidání zátěže do části, kde je elektronika a je v tom místě robot lehčí.

Jako první pokus o ovládání robota bylo pole stykačů ovládané tranzistory. Bohužel toto ovládání se ukázalo, jako neproveditelné, a proto jsem se rozhodl pro řízení pomocí můstku. Můstek je řízen mikroprocesorem, který ovládají ledky. O pohon se starají dva motory ze stěračů. Sestavit robota není nejlevnější záležitost. Stavba tohoto robota si vyžádala zhruba 6380 Kč. Níže je detailnější rozpis jednotlivých komponent robota.

Router Edimax BR-6104KP	cena: 1500 Kč
Router Edimax BR-6204WG	cena: 1000 Kč
Můstek Sabertooth 2X10	cena: 1900 Kč
baterie 12V, 7.2 A	cena: 380 Kč
baterie 12V, 12 A	cena: 500 Kč
total stop tlačítko+ krabička	cena: 300 Kč
převodník	cena: 100 Kč
kolečka	cena: 500 Kč
ostatní materiál	cena: 200 Kč



(Obr. 11, autonomní robot RC-DC)

4. Odkazy na použitou odbornou literaturu

Dimension Engineering LLC. 1999. Dimension Engineering - R/C, Power Electronics, Sensors. *Sabertooth 2X10 regenerative dual motor driver*. [Online] Listopad 4, 1999. [Cited: Říjen 12, 2009.] <http://www.dimensionengineering.com/Sabertooth2X10.htm>.

DLOUHÝ Martin, WINKLER Zbyněk. 2004. Co je to robot? *Robotika.cz*. [Online] 1. 12 2004. [Citace: 20. 12 2010.] <http://robotika.cz/guide/robot/cs>.

Lukáš, Jiří. 2008. Radioklub OK1KPI. *Radioklub OK1KPI*. [Online] 15. Červen 2008. [Citace: 15. Říjen 2010.] <http://www.kufr.cz>.

MOSKOVETS, Vladislav. 2007. console_cable. *midge - adm5120 linux mini distribution*. [Online] 19. Srpen 2007. [Citace: 12. Říjen 2010.] http://midge.vlad.org.ua/wiki/console_cable.

Sunspot. 2006. Sunspot Homepage. *Investigating the Linux based Sweex Broadband Router*. [Online] 29. Srpen 2006. [Citace: 30. Září 2009.] <http://www.sunspot.co.uk/Projects/sweexproject.htm>.

ŠERÝCH, Jakub. 2008. Jakub Šerých. *Základní hardwarová úprava routeru*. [Online] 27. Duben 2008. [Citace: 29. Září 2010.] <http://jakub.serych.cz/zakladni-hardwarova-uprava-routeru>.