

SBot v2.0 – učebná pomôcka na výuku robotiky

Úvod

SBot je jednoduchý robotický systém, určený pre ľudí so záujmom o robotiku a programovanie mikroprocesorov. Je vhodný pre začiatočníkov na jednoduché aplikácie, ako aj pre pokročilých na skúšanie umelej inteligencie. Robot komunikuje s počítačom, alebo iným nadriadeným systémom pomocou Bluetooth rozhrania. Kvalitný podvozok s množstvom montážnych otvorov umožňuje pridávanie vlastných mechanických komponentov a univerzálna doska umožňuje pridávanie vlastných elektronických zapojení. Vďaka tomu je vhodný pre rôzne vlastné aplikácie.

Hlavné parametre

Priemer robota s osadeným nárazníkom	150mm
Rozmery robota bez nárazníku	130mm x 105mm
Napájanie	4x 1,2V NiMH/NiCD akumulátory, spotreba pri zapnutých motoroch a Bluetooth spojení: 330mA
Hmotnosť bez zdroja	276g
Hmotnosť so zdrojom	396g
Pohon	2x modelárske servomotory, max. pripojiteľných serv: 6
Rýchlosť	0,156m/s = 0,5625km/h
Hlavný procesor	ATMega128
Programovanie procesora	ISP, bootloader
Dosah Bluetooth pri priamej viditeľnosti	30m
I/O dostupné na univerzálnej DPS	8x digitálne I/O 7x analógových vstupov 1x SPI 1x I2C 1x sériová linka 2x prerušenie
Rozhrania pre pripojenie vyššieho systému	I2C, Sériová linka
Senzory v základnej výbave	2x Senzor na čiaru, nárazník na zistenie so štyrmi možnými smermi nárazu
Ďalšie pripojiteľné senzory bez akejkoľvek úpravy robota	GP2Dxx senzory na vzdialenosť, viac senzorov na čiaru
Iné užívateľské zariadenia	4x LED dióda, 3x tlačítko
Programovací jazyk	C, assembler

Príklady použitia


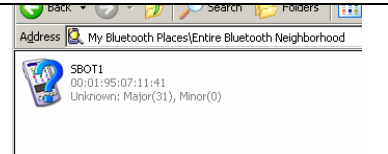
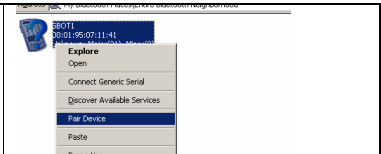
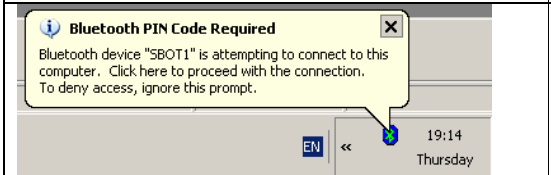

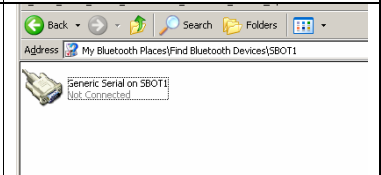
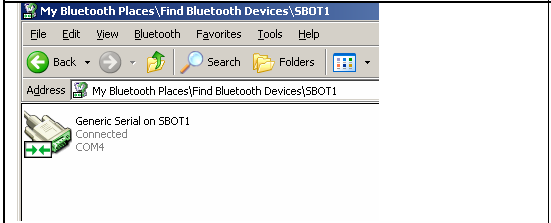
- **Podvozok pre vyšší systém** – na robota je možné namontovať vyšší systém, napr. malú dosku s ARM9 procesorom a Linuxom a kamerou aby rozpoznávala obraz. Takto vybavený robot môže napr. autonómne nasledovať loptičku, hľadať zdroj svetla alebo sledovať čiaru, obchádzať prekážky a mapovať prostredie.
- **Robot na sledovanie čiar** – senzory na podvozku robota mu umožňujú sa zúčastniť tradičnej disciplíny na robotických súťažiach – sledovanie čiar.
- **Dial'kovo ovládaný robot** – robota možno ovládať cez Bluetooth so základným firmwarom, a napojiť ho na počítač. Bluetooth vytvára sériový port, na ktorý sa možno pripojiť cez akýkoľvek terminál, alebo vlastnou aplikáciou
- **Len tak sa hrať** – s robotom a mikroprocesormi je vždy veľa zábavy, stačí len použiť fantáziu

Oživovanie Sбота

Robota najprv treba oživiť. Postup na oživovanie:

1. Najprv nastavte hlavný vypínač na vypnutú polohu.
2. Teraz vložte nabitú NiMH batérie do držiaku na batérie zospodu. Na držiaku je naznačené ako treba batérie orientovať. Robot má ochranu voči prepólovaniu, ale je lepšie ich tam dať správne.
3. Keď sú batérie na mieste, môžeme prepnúť hlavný vypínač do zapnutej polohy.
4. Robot sa začne spúšťať – je to vidno podľa toho, že 4 bočné LED diódy sa postupne rozsvetia a nakoniec zhasnú. Predné diódy potom začnú blikať.
5. Robot je teraz pripravený na prepojenie s počítačom. Má funkčný základný firmware od výroby.
6. Je dôležité aby neboli pri zapínaní napájania stlačené všetky tri tlačítka – vtedy robot resetuje Bluetooth modul na základné nastavenia. Takisto by nemal byť skratovaný jumper na spúšťanie bootloadera.

Teraz máte funkčného robota. Pripojenie sa naňho cez Bluetooth je jednoduché – stačí dať vo vašom systéme vyhľadať všetky Bluetooth zariadenia v okolí a pripojiť sa na robota s názvom „SBOT1“. Niektoré počítače sú nastavené tak, aby na takéto spojenie požadovali zabezpečené spojenie, no na robota sa dá pripojiť aj bez PIN kódu. Keby od vás počítač pri spájaní PIN kód nežiadal, je to v poriadku, ak áno, tak PIN kód je 1234 . Postup na toto je zobrazený v tabuľke:

 <p>Dáme vyhľadať všetky Bluetooth zariadenia v okolí</p>	 <p>Keď počítač nájde zariadenie „SBOT1“ klikneme naň pravým tlačítkom myši</p>	 <p>Zobrazí sa nám menu, z ktorého zvolíme „Pair device“</p>
 <p>Robot bude žiadať PIN kód. Klikneme na bublinu ktorá sa nám zobrazí.</p>	 <p>Otvorí sa nám nové okno do ktorého zadáme PIN kód - 1234</p>	 <p>Teraz sa na robota môžeme pripájať ako na sériový port. Dvojklikom na ikonu sériového portu sa pripojíme.</p>
 <p>Sme pripojení. V tomto prípade sa môžeme na robota pripojiť cez ser. port COM4.</p>		

Po tomto sa na robota môžete pripojiť programom SbotManager.exe, ktorý je možné nájsť na priloženom CD, alebo niektorým terminálovým programom.

Možné stavy SБota po zapnutí

SBot pri použití základného firmwaru má viacero stavov do ktorých môže nabehnúť.

Stav	Popis
Hlavné LED sú vypnuté a predné občas bliknú	Normálny stav – robot očakáva príkazy cez BlueTooth, prípadne ich vykonáva. Stav LED diód možno nastaviť cez špeciálny príkaz
Hlavné LED diódy blikajú veľmi rýchlo a predné občas bliknú	Slabá batéria. Robota je vtedy vhodné vypnúť a batérie dať do nabíjačky. Robot samotný funguje od približne 3,8V, no prílišné vybitie batérií by ich mohlo poškodiť.
Hlavné LED diódy sú zasvietené v poradí zapnutá, zapnutá, vypnutá, zapnutá, a predné svietia	Robot je v režime bootloadera.

Hardware – čo a ako

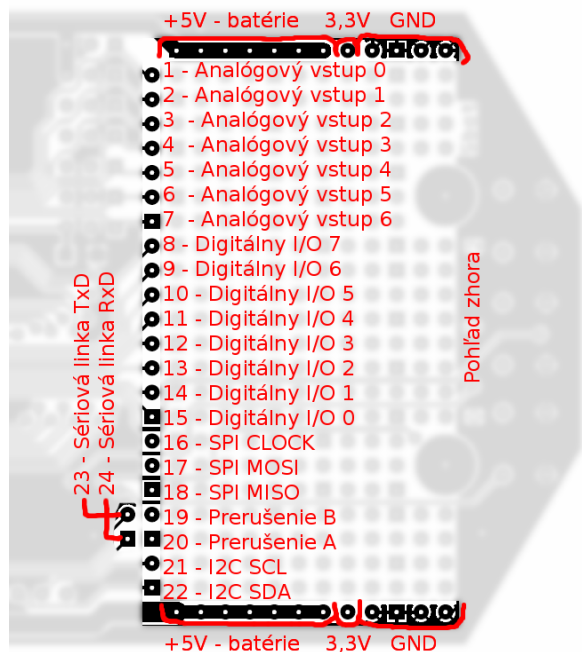
Bloková schéma robota

Všetka elektronika robota je sústredená na hlavnej riadiacej doske. Na ňu sa pripájajú senzory, užívateľské zariadenia, alebo BlueToothom sa možno na ňu pripojiť. Okrem toho je na robotovi elektronika v servomotoroch, v senzoroch na čiaru a v nárazníkových senzoroch. Možno ju nájsť v prílohách.

Hlavná riadiaca doska

Jadrom celého robota je hlavná riadiaca doska – je na nej osadený procesor ATmega128, BlueTooth modul, na ňu sa pripájajú senzory a napájanie. Jej schéma je v **prílohe 1**, a tento text pojednáva o elektronike robota, jej jednotlivých častiach a možnostiach. Väčšina elektroniky robota je sústredená priamo na tejto doske.

Celá doska je rozdelená na dve hlavné časti: elektronika a univerzálna doska. Na univerzálnu dosku si užívateľ môže naletovať vlastné zapojenia, ako napr. špeciálne senzory alebo LCD display. Zhora a zdola sú k nej privedené napájacie napätia, z boku sú na ňu privedené signály z procesora. V jej strede je miesto pre vaše zapojenie. Rozloženie pinov je ukázané na obrázku.



Univerzálna doska

Napájacie napätia privedené na dosku sú +5V z batérii, ktoré môže kolísať v závislosti od stavu a zaťaženia batérií, 3,3V brané zo stabilizátora, ktoré je stabilné, a ZEM, tj. 0V. Z 3,3V je vhodné brať najvyššiu 0,4A, z 5V je možné brať prúdy do 1A.

Analógové vstupy 0 – 6 sú pripojené aj na konektory na senzory, aj vyvedené sem na univerzálnu dosku. Preto keď je použitý analógový vstup procesora na jednu vec – napr. pripojený na senzor čiary, nie je možné ho pripojiť na niečo ďalšie v našom zapojení. Na tieto piny možno pripájať napäťové hodnoty od 0 až po $V_{ref} A/D$ prevodníka, čo je alebo napájacie napätie, tj. napätie batérií, alebo jeho vnútorný ref. zdroj, čo je 2,56V.

Digitálne I/O piny 0-7 sú vlastne celý PORTC procesora. Každý z týchto pinov môže byť vstup alebo výstup nezávisle na ostatných. Každý je schopný dodať alebo uzemniť až 20mA, keď sa má správať ako výstup. Keď sú nastavené ako vstupy, tak je možné zapnúť aj pull-up odpor o hodnote približne 20kOhm.

SPI zbernica je vyvedená pre komunikáciu

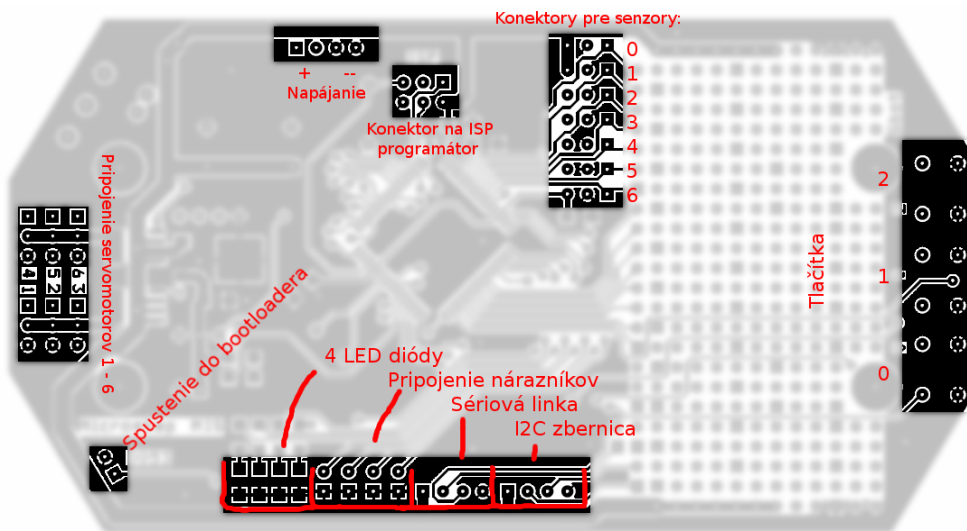
s pokročilejšími zariadeniami ako sú napr. sériové pamäte, D/A prevodníky atď. **Pin SPI CLOCK je zdieľaný s programovacím rozhraním, čo znamená, že počas programovania na ňom bude nedefinovaná hodnota napätia, alebo môže prechádzať z jednotky do nuly.**

Piny prerušení sú vyvedené na **Prerušenie A** (pripojené na pin INT6) a **Prerušenie B** (pripojené na pin INT7). Keď nastane podmienka na jednom z týchto pinov, ktorú si mi určíme – nábežná alebo dobežná hrana signálu, signál na nule, alebo zmena logickej hodnoty, program skočí okamžite na miesto programu ktoré si určíme.

I2C zbernica je tiež určená pre pokročilejšie zariadenia, navyše je možné ich pripojiť viacero – až 128 zariadení na jednu zbernici. Na oboch týchto pinoch sú 5,6kOhm odpory, vhodná rýchlosť je 400kbps.

Sériová linka je vyvedená tiež, RxD sú dáta, ktoré majú smerovať zo zapojenia, do procesora, a TxD sú dáta smerujúce od procesora do zapojenia. **Tieto piny sa využívajú pri programovaní cez ISP rozhranie! Počas programovania na nich nemusí byť stabilná hodnota! Takisto na nich počas programovania nemá byť nič zapojené.**

Všetky digitálne I/O pracujú s TTL (0/5V) úrovňami, jedine I2C zbernica má výstup riešený ako otvorený kolektor. Ak by sme chceli, nie je nutné používať jednotlivé piny ako napr. SPI piny na SPI zbernici, ale je možné ich používať aj ako obyčajné I/O piny, keď nám nestačia digitálne I/O piny.



Hlavná doska – hlavná elektronika

Schému zbytku hlavnej dosky je možné nájsť v **prílohe 1**. Na obrázku vedľa môžeme vidieť ako sú porozmiestňované rôzne zariadenia. Štvorcová ploška vždy značí PIN1 daného konektora.

Konektor na napájanie je treba spojiť

priamo s baterkami. Polarita je dôležitá – robot je chránený voči prepólovaniu, no je lepšie to nerobiť. Polarita prípojky je určená jednoducho – červený drôt je +, čierny je – . Odtiaľto je napájanie vedené ďalej, na hlavný vypínač, cez vratnú poistku na 3A – ak ju vyhodíte nie je to problém – obnoví sa za pár minút. Potom je už ďalej hlavný rozvod napájania – na procesor, na univerzálnu dosku, na motory, senzory atp. Napájanie možno merať pomocou A/D prevodníka, na kanáli č. 7. kde tam pripojené priamo napájanie cez odporový delič. K tomu aby sa zmeralo, je nutné prepnúť referenčné napätie A/D prevodníka z AVCC pinu na vnútornú, 2,56V referenciu. Viac o tomto v časti o použití A/D prevodníku. Rozloženie pinov: 1,2: +5V , 3,4: Zem.

Prípojky servomotorov: Na dosku možno pripojiť až 6 nezávislých servomotorov. Prípojky na ne sú rozdelené na dve podskupiny, každá s vlastným odpájaním napájania. Odpájanie napájania je realizované MOSFET tranzistorami neďaleko týchto prípojok. GATE týchto tranzistorov pre sadu motorov 1 - 3 sú pripojené na pin procesora PORTB, 4, sada 4 – 6 je pripojený na PORTE, 2. Ak sú GATE na nule, prúd môže tiecť do motorov. Ak je tam log. 1, tak sú motory odrezané od napájania. Vo frameworku je možné motory zapínať makrami **servo_123_on(); servo_456_on();** a vypínať makrami **servo_123_off(); servo_456_off();** . Riadiaci signál na servomotory je generovaný časovačom procesora, takže nie je nutné sa o to starať inde v softwari. Nastavenie smeru serva sa dá spraviť makrom **set_servo(CISLO_SERVA, SMER);** pričom CISLO_SERVA je priamo číslo motora ktorého smer chceme nastaviť a SMER je jedna z definícií SERVO_FW – servo forward – servo vpred, SERVO_BW – servo backward – servo vzad. Viac o tomto v časti o frameworku.

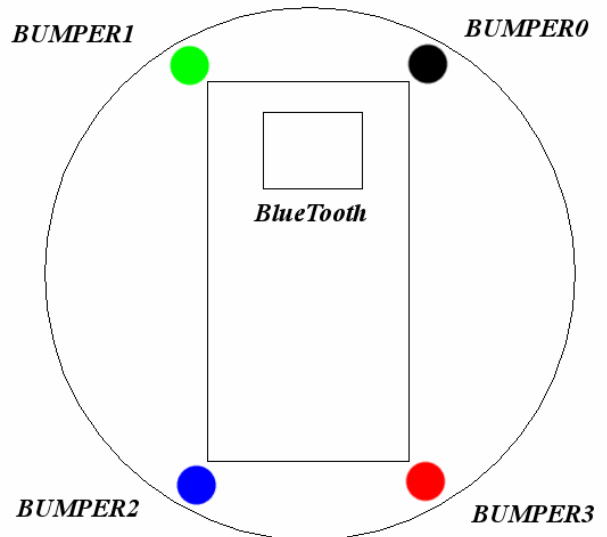
Motor	Pin
1	PORTB 5 – OC1A
2	PORTB 6 – OC1B
3	PORTB 7 – OC1C
4	PORTE 3 – OC3A
5	PORTE 4 – OC3B
6	PORTE 5 – OC3C

Na boku je tabuľka pripojenia jednotlivých signálov zo serv. Štvorčeková plôška je signál – na túto stranu je nutné pripojiť signálový kábel serva – väčšinou žltý. Čierny kábel je väčšinou zem serva – ten treba pripojiť oproti signálovému. Rozloženie pinov: 1: Signál 2: +5V – napájanie, 3: Zem

Prepojku na Spustenie bootloadera je možné skratovať jumperom a potom resetovať robota – vtedy robot prejde do bootloaderového režimu, kde je ho možné programovať cez BlueTooth spojenie. Viac o tomto v časti o programovaní robota.

4 LED diódy sú pripojené na horné 4 bity PORTA. Je možné ich riadiť cez makro **set_leds(STAV_LED)**, pričom STAV_LED je vytvorená súčtom stavu LED. Napr. zapnutie LED2 a LED3 by sme docielili kódom **set_leds(0x02 | 0x04)**.

Prípojky na nárazníky sú vstupné piny procesora, pripojené na dolné 4 bity PORTA. Pull-up odpor je v procesore. Stav nárazníku je možné zistiť cez makro **bumper_stat(KTORY_NARAZNIK)**, pričom KTORY_NARAZNIK určuje ktorý nárazník sa má zistiť. Viac o tomto v časti o makrách. Nárazníky sú rozložené tak, ako je ukázané na obrázku vedľa, pri pohľade zhora. Bumper 0 je pripojený na PORTA, nultý bit, 1 je na prvom bite, a tak ďalej. Farebne sú rozlíšené vodiče, ktorými sa spínače pripájajú. Pri zopnutí nárazníka bude logická hodnota na pine 0.



Tlačítka sú pripojené na piny: PORTD, 5 – tlačítko 0, PORTD, 6 – tlačítko 1, PORTD 7 – tlačítko 2. Je možné zistiť stav tlačítka makrom

button_pressed(KTORE_TLACITKO), pričom KTORE_TLACITKO je tlačítko ktoré chceme prečítať. Môže mať hodnotu **GP_BUTTON_0**, **GP_BUTTON_1** a **GP_BUTTON_2**. Viac o tomto v časti o makrách. Nemajú vlastné pull-up odpory, preto je nutné zapínať tie v procesore.

Konektory na senzory sú pripojené prvým pinom na jeden kanál A/D prevodníka. Nultý je pripojený na kanál 0, prvý na prvý atď. Možno sem pripojiť akýkoľvek senzor, s napäťovým analógovým výstupom. Prípojka poskytuje napájanie pre senzor z bateriek, čo je okolo 5V. Rozloženie pinov je: 1: Signál do A/D prevodníka procesora 2: Zem 3: +5V pre senzor. Senzory by mali mať výstupnú impedanciu menej alebo rovnú 10kOhm, inak môže dôjsť ku skresleniu výsledkov. AD prevodník beží neustále a updatuje výsledky prevodu, ktoré sú uložené v globálnej premennej **adc_vals[]**. **adc_vals[]** je pole integerov, ktoré majú hodnoty od 0 do 1023, pričom 0 znamená že bolo namerané 0V na danom kanále a 1023 znamená, že bolo namerané napájacie napätie. Výpočet na hodnotu **adc_vals** je $adc_vals = (U/UV_{cc}) * 1023$, kde U je merané napätie, a UV_{cc} je napájacie napätie, alebo iné referenčné napätie. Je možné prepnúť AD prevodník na stabilný referenčný zdroj, cez makro **adc_select_ref(REF)**, kde REF je alebo ADC_REF_VCC – ako referenčné napätie je použité napájacie napätie, alebo ADC_REF_256V, kde je použitý vnútorný zdroj napätia : 2,56V. Stav batérie je na **adc_vals[7]**, a pri meraní je AD prevodník prepnutý na ADC_REF_256V referenčné napätie. Hodnotu napätia batérii možno vypočítať pomocou vzorca: **Ubat = ((adc_vals[7]/1023)*2,56V)* 3,127**

Konektor sériovej linky je určený alebo pre vyšší systém, alebo pre komunikáciu „po drôtoch“ s robotom. Je pripojený na sériovú linku USART 0. Pri sériovom ISP programovaní môže byť na RxD a TxD pinoch tohto konektora nedefinovaný signál. Rozloženie pinov: 1: +5V napájanie, môže kolísaf v závislosti od stavu batérií, 2: RxD, dáta do procesora, vstup, 3: TxD, dáta z procesora, výstup, 4: Zem.

Konektor pre pripojenie I2C zbernice je určený pre pripojenie jedného alebo viacej I2C kompatibilných zariadení, ako sú napr. teplomery, rôzne senzory, D/A prevodníky alebo iné inteligentné zariadenia. Konektor poskytuje napájanie pre zariadenia, a Pull-up odpory pre zbernicu o hodnote 5,6kOhm. Rozloženie pinov: 1: +5V napájanie, 2: SDA – sériové dáta, 3: SCL – hodiny pre zbernicu, 4: Zem.

Konektor pre ISP programátor je určený na programovanie procesora v zapojení. Vhodné prípravky, ktoré možno použiť na programovanie procesora v tomto robotovi sú uvedené v [6], kde je aj návod na zostrojenie takého prípravku. Je možné si kúpiť tiež hotový programátor s viacerými funkciami od firmy Atmel [4], napr. AVR Dragon, alebo JTAG ICE MkII. Rozloženie pinov na konektore je zobrazené na obrázku vedľa. Metóda ako programovať bude bližšie popísaná neskôr, v časti o programovaní robota.

MISO	1	2	VCC
SCK	3	4	MOSI
RESET	5	6	GND

Iné časti hlavnej dosky

Hlavná doska má na sebe ešte Bluetooth modul, tlačítko RESET, hlavný vypínač, dve dekoratívne diódy zospodu dosky, LED diódu indikujúcu Bluetooth spojenie a na pripojenie 4 montážne diery.

RESET tlačítko je pripojené na RESET pin procesora. Ak je stlačené, procesor sa resetuje, tj. skočí na začiatok svojho programu, vynuluje si registre atď. Ak je vyskratovaný Jumper na Spustenie Bootloadera, tak naskočí do režimu bootloadera, odkiaľ ho možno programovať bez použitia programátora. Viac o tejto možnosti v časti o programovaní robota.

Hlavný vypínač odpája napájanie od všetkej elektroniky na hlavnej doske. Jedná sa o páčkový prepínač umiestnený na boku hlavnej dosky.

Dve **dekoratívne diódy zospodu dosky** sú napojené na pin procesora PORTG, 1, pričom svietia keď je na ňom log. 0. Je možné ich zapínať pomocou makra **set_decoleds()**, **clear_decoleds()** a **toggle_decoleds()**. Makro **set_decoleds()** zapne LED diódy, **clear_decoleds()** ich vypne, a **toggle_decoleds()** zneguje súčasný stav – ak svietia, vypne ich, ak nie, tak ich zapne.

Bluetooth modul je typu ESD200 so zabudovanou anténou, s dosahom približne 30m. Je napájaný zo zdroja 3,3V a rovnakej úrovne má aj vstupy a výstupy, preto bolo nutné použiť odporový delič na jeho vstup. Jedná sa o transparentný modul, tj. to čo pošlete na počítači do na emulovanú sériovú linku, to dostanete ako výstup z modulu. Takto sa správa, keď je spojený s počítačom. Keď sa nadväzuje spojenie, pošle procesoru informácie o tom kto sa spája. Aby bolo možné jednoducho tieto dáta alebo ignorovať, alebo rozlíšiť od príkazov čo dostáva robot od počítača, je na pin procesora privedená informácia o stave pripojenia na pin PORTB, 0. Ak je tu log. 0 tak robot je napojený na vyšší systém. Ak tu je log. 1, tak robot nie je napojený na nič. Toto je možné zisťovať pomocou makra **bluetooth_connection_established()**, ktoré vracia nenulovú hodnotu ak je spojenie, a **0** hodnotu, ak spojenie nie je nadviazané. Viac o tomto v časti o makrách a firmwari. Bluetooth modul je osadený v päťici, je možné ho vyberať, **no je to NEDOPORUČOVANÉ**. Bluetooth modul komunikuje s procesorom pomocou sériovej linky USART1, pinmi PORTE 0 a PORTE 1. Z Bluetooth modulu je okrem RxD a TxD vyvedený aj jeho RESET. Keď je na ňom log. 1, nie je aktívny, keď je na ňom log. 0, tak sa nastaví na pôvodné továrenské nastavenia. **Používať len v prípade, že viete čo robíte**. Bluetooth modul sa správa ako transparentná sériová linka – to čo ide na jednej strane dnu, ide na druhej strane von. Okrem znakov ktoré pošle počítač robotovi na linke nie sú žiadne iné informácie.

LED dióda indikujúca spojenie je napojená na pin Bluetooth modulu. Svieti keď je pripojené nejaké zariadenie na Bluetooth – napr. počítač, a nesvieti keď nie je nič pripojené na robota. Nie je možné ju ovládať programom.

Montážne diery: Doska je upevnená na kostre robota pomocou 4 skrutiek, cez 4 diery priemeru 3,5mm. Miesto skrutiek boli použité dištančné stĺpiky aby bolo možné nad dosku pripojiť nejaké ďalšie dosky, napr. už spomínaný vyšší systém s Linuxom. Vzdialené sú od seba 96,5mm na šírku a na výšku sú vzdialené 28mm.

Senzor na čiaru

Na sledovanie čiary alebo povrchu pod robotom bol zvolený senzor CNY70. Je vhodný na snímanie predmetov vo vzdialenosti od 0 do 5mm. Senzor je celá jednoduchá doska, ktorú je veľmi jednoduché namontovať – stačí na to jedna skrutka. Táto doska je napájaná 5V a poskytuje napäťový analógový výstup, ktorý je možné pripojiť na A/D prevodník procesora.



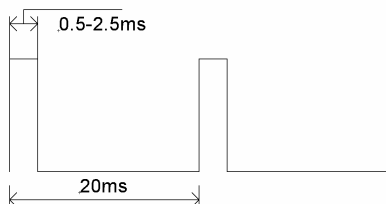
Jej funkcia je jednoduchá – zospodu je fototranzistor a infračervená LED dióda. LED dióda svieti na povrch. Čím viac svetla sa odrazí na fototranzistor, tým viac prúdu potečie cez jeho emitor. Toto spôsobí pokles napätia na fototranzistore, ktorý je možné vyhodnotiť na A/D prevodníku procesora. Tento signál sa pohybuje od približne 1V pri bielom povrchu, keď sa na fototranzistor odráža najviac svetla, až po takmer 4,5V, keď je na veľmi čiernom povrchu.

Zo senzoru sú vyvedené káble s koncovkou, ktorá sa dá pripojiť na hlavnú dosku, na konektory pre senzory, červeným drôtom na pin 1.

Rozloženie signálov na výstupe: 1: Signál von, zelený vodič, 2: Zem, čierny vodič, 3: +5V – napájanie, červený vodič.

Iná elektronika

Okrem hlavnej dosky a senzorov je na robotovi ešte elektronika v servomotore, sú tu ešte batérie a nárazníky.



Servomotor riadi jednoduchú elektroniku, ktorej úlohou je riadiť motor na základe signálu ktorý dostáva. Signál je digitálny, obdĺžnikový opakujúci sa pulz, pričom sa mení dĺžka kladnej hodnoty. Na základe šírky tohto impulzu motor určí ktorým smerom sa má točiť. Stredná hodnota, keď stojí je približne 1,5ms. Z motora vedú tri vodiče, ktoré sú pripojené na koncovku. Vodiče sú farebne označené, no rôzni výrobcovia používajú rôzne farby, no väčšinou je čierna zem. Tá je na okraji. V strede je kladný pól napájacieho napätia, a na boku je signál. Takže rozloženie pinov: 1: Signál, 2: Napájanie +5V, 3: Zem.

Nárazník je mechanicky spojený so štyrmi vhodne umiestnenými mech. spínačmi tak, aby bolo možné zistiť z ktorej strany do robota niekto/niečo narazilo. Jedná sa o obyčajné spínače, pričom normálne sú rozopnuté, a keď na niečo nárazník narazí zopnú sa.

Batérie sú NiMH nabíjateľné, kapacity 2500mAh. Dokopy dávajú napätie od 4,8V do 6V. Ak je napätie na jednom článku okolo 1,2V je vhodné ho dať nabíjať.

Mechanická stránka veci

Robot je konštruovaný ako klasický systém s tromi opornými bodmi – jedným pasívnym, a dvoma kolieskami, ktoré ním hýbu. Kolesá sú primontované na hriadele motorov. Motory sú primontované na hlavnú kostru, rovnako ako aj všetko ostatné.

Kolieska sú napojené na hriadeľ motora. Na ich povrchu je guma, čo zabezpečuje dobrý styk s povrchom.

Ovládanie robota cez Bluetooth

Na riadenie robota možno použiť alebo priložený program SbotManager.exe, alebo pomocou jednoduchého protokolu cez akýkoľvek terminálový program – napr. Hyperterminal, alebo BrayTerminal. Pri ručnom zadávaní príkazov do robota je nutné komunikovať s robotom pomocou protokolu, ktorý je popísaný v časti Popis protokolu.

Popis protokolu

Robotovi možno dávať príkazy pomocou jednoduchého protokolu. Všetky príkazy sú vo forme paketov tvaru:

!0pxxxxCC~

SBot2.0 – dokumentácia

Kde význam jednotlivých častí je:

!	Výkričník indikuje začiatok paketu – keď je zadany, robot vynuluje zásobník na paket, a začne prijímať nový paket. Výkričník musí byť pred každým paketom																												
0	Číslo robota – ak je pripojených viac robotov, tak aby nenastal konflikt má každý svoje meno. Sboty sa potom volajú napr. SBOT0, SBOT1,..., SBOTA, SBOTZ. Číslo môže byť číslo, alebo písmeno. Dá sa zmeniť príkazom popísaným nižšie. Po zakúpení má SBOT číslo 1, takže ho uvidíte ako SBOT1 keď dáte vyhľadávať BlueTooth zariadenia.																												
P	Príkaz je písmeno, ktoré hovorí robotovi, čo má robiť. Môže to byť jedno z týchto písmen <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: center;">F</td><td>Vpred – robot pôjde vpred. Príklad paketu: !1F000056↵</td></tr> <tr> <td style="text-align: center;">B</td><td>Vzad – robot pôjde vzad. Príklad paketu: !1B000052↵</td></tr> <tr> <td style="text-align: center;">L</td><td>Vľavo – robot sa bude točiť vľavo na mieste. Príklad paketu: !1L00005C↵</td></tr> <tr> <td style="text-align: center;">R</td><td>Vpravo – robot sa bude točiť vpravo na mieste. Príklad paketu: !1R000042↵</td></tr> <tr> <td style="text-align: center;">f</td><td>Vpred určitú dobu – robot pôjde vpred. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1f01f425↵ . Tento príkaz spôsobí, že robot pôjde 5 sekúnd vpred.</td></tr> <tr> <td style="text-align: center;">b</td><td>Vzad určitú dobu – robot pôjde vzad. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1b00FA75↵ . Tento príkaz spôsobí, že robot pôjde 2.5 sekundy vzad.</td></tr> <tr> <td style="text-align: center;">l</td><td>Točenie vľavo určitú dobu – robot sa bude na mieste točiť vľavo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1l03E802↵ . Tento príkaz spôsobí, že robot sa bude točiť 10 sekúnd vľavo.</td></tr> <tr> <td style="text-align: center;">r</td><td>Točenie vpravo určitú dobu – robot sa bude na mieste točiť vpravo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1r003263↵ . Tento príkaz spôsobí, že robot sa bude točiť 0.5 sekundy vpravo.</td></tr> <tr> <td style="text-align: center;">S</td><td>Stop – robot sa zastaví okamžite. Príklad paketu: !1S000043↵</td></tr> <tr> <td style="text-align: center;">n</td><td>Robot sa premenuje na premenuje. Jeho nové meno (od A do Z, alebo od a do z, alebo od 0 do 9) musí byť zadane v parametri funkcie – musí byť zadaná ASCII hodnota daného znaku. Robot sa po tomto odpojí z BlueToothu, resetuje modul, a celý sa resetuje. Je nutné znova sa s ním spárovať. Príklad paketu: !1n004278↵ . Tento príkaz spôsobí, že sa robot SBOT1 premenuje na robota SBOTB.</td></tr> <tr> <td style="text-align: center;">D</td><td>Nastaví 4 užívateľské LED diódy na požadovanú hodnotu. Hodnota musí byť v parametri príkazu, a môže mať hodnotu od 0 do 15. Podľa binárnej hodnoty tohto čísla budú zasvietené LED diódy. Príklad paketu: !1D000F22↵ . Tento paket spôsobí, že budú zasvietené všetky LED diódy. Odpoveď je !1O00005F↵ - len potvrdenie o prijatí alebo spracovaní.</td></tr> <tr> <td style="text-align: center;">A</td><td>Is Alive otázka – robot nevykoná nič, len odpovie, že je, a v parametri odpovede bude verzia jeho firmwaru. Príklad paketu - !1A000051↵ . Tento paket spôsobí, že robot odpovie na paket, no nič nespraví – niečo ako ping. Odpoveď bude napríklad !1A000150↵ – znamená to, že robot má verziu firmwaru 1.</td></tr> <tr> <td style="text-align: center;">N</td><td>Number query – otázka na číslo. Rovnako ako ping, no je vhodné ho adresovať všetkým robotom, ak máme pripojených viacej. Potom sa všetky ozvú, no počkajú istú dobu. Odpoveď je rovnaká, ako na ping – príkaz A. Príklad paketu: !*N000045↵ . Tento paket spôsobí, že všetky pripojené roboty sa ohlásia svojím číslom a verzou firmwaru v parametri.</td></tr> <tr> <td style="text-align: center;">j</td><td>V odpovedi vráti stav nárazníkov. Pokiaľ nie je žiadna kolízia na nárazníku, vracia 1, ak je, vracia 0. V parametri odpovede je v hex. tvare hodnota od 0 do 15, pričom jej hodnotu je možné vyrátať ako hodnota = (1 * stav A nárazníka) + (2 * stav B nárazníka) + (4 * stav C nárazníka) + (8 * stav D nárazníka)). Rozloženie nárazníku je možné nájsť v časti o nárazníku. Príklad paketu: !1j00007a↵.</td></tr> </table>	F	Vpred – robot pôjde vpred. Príklad paketu: !1F000056 ↵	B	Vzad – robot pôjde vzad. Príklad paketu: !1B000052 ↵	L	Vľavo – robot sa bude točiť vľavo na mieste. Príklad paketu: !1L00005C ↵	R	Vpravo – robot sa bude točiť vpravo na mieste. Príklad paketu: !1R000042 ↵	f	Vpred určitú dobu – robot pôjde vpred. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1f01f425 ↵ . Tento príkaz spôsobí, že robot pôjde 5 sekúnd vpred.	b	Vzad určitú dobu – robot pôjde vzad. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1b00FA75 ↵ . Tento príkaz spôsobí, že robot pôjde 2.5 sekundy vzad.	l	Točenie vľavo určitú dobu – robot sa bude na mieste točiť vľavo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1l03E802 ↵ . Tento príkaz spôsobí, že robot sa bude točiť 10 sekúnd vľavo.	r	Točenie vpravo určitú dobu – robot sa bude na mieste točiť vpravo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1r003263 ↵ . Tento príkaz spôsobí, že robot sa bude točiť 0.5 sekundy vpravo.	S	Stop – robot sa zastaví okamžite. Príklad paketu: !1S000043 ↵	n	Robot sa premenuje na premenuje. Jeho nové meno (od A do Z, alebo od a do z, alebo od 0 do 9) musí byť zadane v parametri funkcie – musí byť zadaná ASCII hodnota daného znaku. Robot sa po tomto odpojí z BlueToothu, resetuje modul, a celý sa resetuje. Je nutné znova sa s ním spárovať. Príklad paketu: !1n004278 ↵ . Tento príkaz spôsobí, že sa robot SBOT1 premenuje na robota SBOTB.	D	Nastaví 4 užívateľské LED diódy na požadovanú hodnotu. Hodnota musí byť v parametri príkazu, a môže mať hodnotu od 0 do 15. Podľa binárnej hodnoty tohto čísla budú zasvietené LED diódy. Príklad paketu: !1D000F22 ↵ . Tento paket spôsobí, že budú zasvietené všetky LED diódy. Odpoveď je !1O00005F ↵ - len potvrdenie o prijatí alebo spracovaní.	A	Is Alive otázka – robot nevykoná nič, len odpovie, že je, a v parametri odpovede bude verzia jeho firmwaru. Príklad paketu - !1A000051 ↵ . Tento paket spôsobí, že robot odpovie na paket, no nič nespraví – niečo ako ping. Odpoveď bude napríklad !1A000150 ↵ – znamená to, že robot má verziu firmwaru 1.	N	Number query – otázka na číslo. Rovnako ako ping, no je vhodné ho adresovať všetkým robotom, ak máme pripojených viacej. Potom sa všetky ozvú, no počkajú istú dobu. Odpoveď je rovnaká, ako na ping – príkaz A . Príklad paketu: !*N000045 ↵ . Tento paket spôsobí, že všetky pripojené roboty sa ohlásia svojím číslom a verzou firmwaru v parametri.	j	V odpovedi vráti stav nárazníkov. Pokiaľ nie je žiadna kolízia na nárazníku, vracia 1, ak je, vracia 0. V parametri odpovede je v hex. tvare hodnota od 0 do 15, pričom jej hodnotu je možné vyrátať ako hodnota = (1 * stav A nárazníka) + (2 * stav B nárazníka) + (4 * stav C nárazníka) + (8 * stav D nárazníka)). Rozloženie nárazníku je možné nájsť v časti o nárazníku. Príklad paketu: !1j00007a ↵.
F	Vpred – robot pôjde vpred. Príklad paketu: !1F000056 ↵																												
B	Vzad – robot pôjde vzad. Príklad paketu: !1B000052 ↵																												
L	Vľavo – robot sa bude točiť vľavo na mieste. Príklad paketu: !1L00005C ↵																												
R	Vpravo – robot sa bude točiť vpravo na mieste. Príklad paketu: !1R000042 ↵																												
f	Vpred určitú dobu – robot pôjde vpred. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1f01f425 ↵ . Tento príkaz spôsobí, že robot pôjde 5 sekúnd vpred.																												
b	Vzad určitú dobu – robot pôjde vzad. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1b00FA75 ↵ . Tento príkaz spôsobí, že robot pôjde 2.5 sekundy vzad.																												
l	Točenie vľavo určitú dobu – robot sa bude na mieste točiť vľavo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1l03E802 ↵ . Tento príkaz spôsobí, že robot sa bude točiť 10 sekúnd vľavo.																												
r	Točenie vpravo určitú dobu – robot sa bude na mieste točiť vpravo určitú dobu. Zastaví sa po čase určeným parametrom paketu, pričom jedna jednotka parametru je 0,01 sekundy. Príklad paketu: !1r003263 ↵ . Tento príkaz spôsobí, že robot sa bude točiť 0.5 sekundy vpravo.																												
S	Stop – robot sa zastaví okamžite. Príklad paketu: !1S000043 ↵																												
n	Robot sa premenuje na premenuje. Jeho nové meno (od A do Z, alebo od a do z, alebo od 0 do 9) musí byť zadane v parametri funkcie – musí byť zadaná ASCII hodnota daného znaku. Robot sa po tomto odpojí z BlueToothu, resetuje modul, a celý sa resetuje. Je nutné znova sa s ním spárovať. Príklad paketu: !1n004278 ↵ . Tento príkaz spôsobí, že sa robot SBOT1 premenuje na robota SBOTB.																												
D	Nastaví 4 užívateľské LED diódy na požadovanú hodnotu. Hodnota musí byť v parametri príkazu, a môže mať hodnotu od 0 do 15. Podľa binárnej hodnoty tohto čísla budú zasvietené LED diódy. Príklad paketu: !1D000F22 ↵ . Tento paket spôsobí, že budú zasvietené všetky LED diódy. Odpoveď je !1O00005F ↵ - len potvrdenie o prijatí alebo spracovaní.																												
A	Is Alive otázka – robot nevykoná nič, len odpovie, že je, a v parametri odpovede bude verzia jeho firmwaru. Príklad paketu - !1A000051 ↵ . Tento paket spôsobí, že robot odpovie na paket, no nič nespraví – niečo ako ping. Odpoveď bude napríklad !1A000150 ↵ – znamená to, že robot má verziu firmwaru 1.																												
N	Number query – otázka na číslo. Rovnako ako ping, no je vhodné ho adresovať všetkým robotom, ak máme pripojených viacej. Potom sa všetky ozvú, no počkajú istú dobu. Odpoveď je rovnaká, ako na ping – príkaz A . Príklad paketu: !*N000045 ↵ . Tento paket spôsobí, že všetky pripojené roboty sa ohlásia svojím číslom a verzou firmwaru v parametri.																												
j	V odpovedi vráti stav nárazníkov. Pokiaľ nie je žiadna kolízia na nárazníku, vracia 1, ak je, vracia 0. V parametri odpovede je v hex. tvare hodnota od 0 do 15, pričom jej hodnotu je možné vyrátať ako hodnota = (1 * stav A nárazníka) + (2 * stav B nárazníka) + (4 * stav C nárazníka) + (8 * stav D nárazníka)). Rozloženie nárazníku je možné nájsť v časti o nárazníku. Príklad paketu: !1j00007a ↵.																												

SBot2.0 – dokumentácia

		<p>Tento paket spôsobí, že robot odpovie stavom svojich nárazníkov, napr. paketom !1b000F04↵. Z tejto odpovede možno vyčítať, že ani jeden z nárazníkov nemá kolíziu.</p>
	a	<p>V odpovedi vráti stav zvoleného analógového vstupu procesora. V parametri paketu je nutné zadať ktorý analógový vstup chceme zmerať. Odpoveď bude obsahovať hex. vyjadrenie hodnoty z A/D prevodníka a číslo meraného vstupu.</p> <p>Napr. ak chceme zmerať 7 analógový vstup – stav batérie, tak pošleme paket: !1a000776↵, a dostaneme ako odpoveď paket ako napr. !1a724171↵. Tento paket určuje, že hodnota 7 analógového vstupu je 0x0241. Stav batérie je teda 4,515V. viz. Časť o A/D prevodníku, kde je aj vzorec na výpočet stavu batérie.</p>
XXXX		<p>Parameter príkazu – niektoré príkazy vyžadujú parameter. Napríklad ako dlho má robot ísť vpred, stav ktorého analógového vstupu treba poslať, alebo aké je nové meno robota. To je treba zadať v hexadecimálnej forme do parametru. Napr. ak by mal robot ísť vpred 10sekúnd, čo je 1000 jednotiek času robota, tak parameter funkcie bude: 03E8. Nuly je treba písať tiež, je nutné zachovať dĺžku 4 znaky. Písmená môžu byť veľké aj malé. Na tom nezáleží.</p> <p>Ak bol zadán nevhodný parameter, napr. na nastavenie stavu LED diód sme chceli použiť číslo 55 (0037), tak dostaneme odpoveď Invalid parameter: Napr. !1i000069↵, ktorý nám hovorí, že parameter, ktorý sme zadali nie je možné použiť pre danú funkciu.</p>
CC		<p>Checksum – aby sa skontrolovalo, či bol paket správne prijatý, robot kontroluje checksum. Checksum sa ráta takto:</p> <p>Checksum = 0x00 XOR (Prvý bajt paketu) XOR (Druhý bajt paketu XOR) ...</p> <p>Výsledok tejto operácie je nutné potom poslať na konci paketu v hexadecimálnej forme. Napr. Checksum paketu !1b000F vyrátame takto:</p> <p>Checksum = 0x00 XOR '!' XOR 'b' XOR '0' XOR '0' XOR '0' XOR 'F'</p> <p>Hodnota na konci bude 4, čoho hex. hodnota je 0x04. Preto potom pošleme paket: !1b000F04.</p> <p>Ukončenie paketu – nový riadok – nezaráčujeme do checksumu.</p> <p>Ak zadáme nesprávny checksum, robot príkaz nevykoná, no pošle späť paket, v ktorom je správny checksum. Vďaka tomuto sa pri manuálnom zadávaní príkazov nemusíme trápiť zdĺhavými výpočtami – stačí poslať paket ktorý chceme poslať s akoukoľvek checksum a potom si pozrieť, aký paket nám došiel späť. Ak bol paket tvaru, odpovede !1B00XXcc↵, tak XX je hexadecimálny tvar správneho checksumu a cc je hexadecimálny tvar checksumu odpovede. XX potom stačí skopírovať do nášho paketu.</p> <p>Príklad takejto konverzácie:</p> <p>Posielame: !1b005055↵ ;posielame paket, aby robot išiel vzad 80 časových jednotiek (0,8sekundy) vzad</p> <p>Príjem: !1B007752↵ ;prijali sme paket, ktorý hovorí o tom, že sme poslali zlý checksum, a že správny checksum je 0x77</p> <p>Posielame: !1b005077↵ ;posielame opäť paket, tento krát už so správnym checksumom</p> <p>Príjem: !1O00005F↵ ;dostávame potvrdenie o prijatí a spracovaní paketu.</p>
↵		<p>Ako potvrdenie konca paketu musíme zadať nový riadok. Tj. stlačiť ENTER. Treba poslať znak 0x0D (CR) alebo 0x0A (LF). Koniec paketu sa nezaráčava do checksumu.</p>

Robot odpovedá na príkazy odpoveďovými paketmi. Tvar majú rovnaký, ako príkazové pakety, no nastávajú len ak robot odpovedá na príkaz.

Tvar paketu:

!0pxxxxCC↵

Pričom jednotlivé časti majú význam:

!	Rovnaké ako u príkazu														
0	Názov robota, ktorý odpovedá, rovnaké ako u príkazu														
p	<p>Druh odpovede – pre zjednodušenie komunikácie robot odpovedá na rôzne požiadavky rôzne. Takisto, aj pre rôzne chyby pri komunikácii (nesprávny parameter, checksum atď.) má rozdielne odpovede. Môže to byť jedno z týchto písmen:</p> <table border="1"> <tr> <td>B</td><td>Zlý checksum – robot takto odpovedá keď sme mu poslali zlý checksum v príkazovom pakete. Tvar parametru odpovede je 00xx, kde xx je hexadecimálne vyjadrený správny checksum pre daný paket. Príklad odpovede: !1B007257↵ – tento paket hovorí o tom, že príkaz, ktorý sme posledne poslali mal zlý checksum, a správny checksum je 0x72</td></tr> <tr> <td>U</td><td>Takto odpovedá robot, keď sme mu do príkazu zadali neznámy príkaz – napr. zadali sme !1H000058↵. H nie je medzi príkazmi, tak nám robot pošle paket !1U000045↵. Tento paket hovorí o tom, že posledne zadaný príkaz nepozná.</td></tr> <tr> <td>I</td><td>Nesprávny parameter – takto odpovedá robot, keď sme zadali parameter, ktorý robot nemôže použiť pre danú funkciu. Napr. ak zadáme paket !1n00237F↵ (zmena mena na @, čo nie je znak ktorý môže byť použitý ako meno), tak dostaneme od robota odpoveď: !1I000059↵, čo znamená, že parameter je nepoužiteľný pre túto funkciu.</td></tr> <tr> <td>O</td><td>OK – robot vracia tento paket keď vykonanie príkazu prebehlo úspešne. Robot ho vracia pre príkazy: F, B, L, R, f, b, r, l, S, D, n - v prípade ich úspešného vykonania a správnych parametrov.</td></tr> <tr> <td>a</td><td>Stav analógového vstupu – po príkaze na poslanie stavu analógového vstupu robot pošle posledné meranie. Odpoveďový paket má parameter tvaru xyyy, kde x je číslo analógového vstupu, ktorého stav je vysielaný, a yyy je hexadecimálne vyjadrenie hodnoty z A/D prevodníka v rozsahu od 0x000 – 0V až do 0x3FF, čo je referenčné napätie, alebo napätie od neho vyššie. Príklad paketu: !1a202F07↵. Tento paket hovorí o tom, že na kanále 2 je napätie hodnoty 0x02F, čo je po prepočte približne 0,11V, s referenčným napätím 2,56V.</td></tr> <tr> <td>b</td><td>Odpoveď na požiadanie o stav nárazníkov. Odpoveď bude mať parameter tvaru 000x, kde x je hexadecimálna hodnota od 0 do 15 vyrátaná ako: $\text{hodnota} = (1 * \text{stav A nárazníka}) + (2 * \text{stav B nárazníka}) + (4 * \text{stav C nárazníka}) + (8 * \text{stav D nárazníka})$ Stav nárazníka je 1 ak nie je kolízia, 0 ak je kolízia. </td></tr> <tr> <td>A</td><td>Odpoveď na ping. Pokiaľ robota požiadame o príkazom A alebo N, tak dostaneme odpoveď, kde parameter je tvaru xxxx, kde xxxx je hexadecimálne vyjadrená verzia firmwaru.</td></tr> </table>	B	Zlý checksum – robot takto odpovedá keď sme mu poslali zlý checksum v príkazovom pakete. Tvar parametru odpovede je 00xx, kde xx je hexadecimálne vyjadrený správny checksum pre daný paket. Príklad odpovede: !1B007257↵ – tento paket hovorí o tom, že príkaz, ktorý sme posledne poslali mal zlý checksum, a správny checksum je 0x72	U	Takto odpovedá robot, keď sme mu do príkazu zadali neznámy príkaz – napr. zadali sme !1H000058↵ . H nie je medzi príkazmi, tak nám robot pošle paket !1U000045↵ . Tento paket hovorí o tom, že posledne zadaný príkaz nepozná.	I	Nesprávny parameter – takto odpovedá robot, keď sme zadali parameter, ktorý robot nemôže použiť pre danú funkciu. Napr. ak zadáme paket !1n00237F↵ (zmena mena na @, čo nie je znak ktorý môže byť použitý ako meno), tak dostaneme od robota odpoveď: !1I000059↵ , čo znamená, že parameter je nepoužiteľný pre túto funkciu.	O	OK – robot vracia tento paket keď vykonanie príkazu prebehlo úspešne. Robot ho vracia pre príkazy: F, B, L, R, f, b, r, l, S, D, n - v prípade ich úspešného vykonania a správnych parametrov.	a	Stav analógového vstupu – po príkaze na poslanie stavu analógového vstupu robot pošle posledné meranie. Odpoveďový paket má parameter tvaru xyyy , kde x je číslo analógového vstupu, ktorého stav je vysielaný, a yyy je hexadecimálne vyjadrenie hodnoty z A/D prevodníka v rozsahu od 0x000 – 0V až do 0x3FF, čo je referenčné napätie, alebo napätie od neho vyššie. Príklad paketu: !1a202F07↵ . Tento paket hovorí o tom, že na kanále 2 je napätie hodnoty 0x02F, čo je po prepočte približne 0,11V, s referenčným napätím 2,56V.	b	Odpoveď na požiadanie o stav nárazníkov. Odpoveď bude mať parameter tvaru 000x, kde x je hexadecimálna hodnota od 0 do 15 vyrátaná ako: $\text{hodnota} = (1 * \text{stav A nárazníka}) + (2 * \text{stav B nárazníka}) + (4 * \text{stav C nárazníka}) + (8 * \text{stav D nárazníka})$ Stav nárazníka je 1 ak nie je kolízia, 0 ak je kolízia.	A	Odpoveď na ping. Pokiaľ robota požiadame o príkazom A alebo N , tak dostaneme odpoveď, kde parameter je tvaru xxxx, kde xxxx je hexadecimálne vyjadrená verzia firmwaru.
B	Zlý checksum – robot takto odpovedá keď sme mu poslali zlý checksum v príkazovom pakete. Tvar parametru odpovede je 00xx, kde xx je hexadecimálne vyjadrený správny checksum pre daný paket. Príklad odpovede: !1B007257↵ – tento paket hovorí o tom, že príkaz, ktorý sme posledne poslali mal zlý checksum, a správny checksum je 0x72														
U	Takto odpovedá robot, keď sme mu do príkazu zadali neznámy príkaz – napr. zadali sme !1H000058↵ . H nie je medzi príkazmi, tak nám robot pošle paket !1U000045↵ . Tento paket hovorí o tom, že posledne zadaný príkaz nepozná.														
I	Nesprávny parameter – takto odpovedá robot, keď sme zadali parameter, ktorý robot nemôže použiť pre danú funkciu. Napr. ak zadáme paket !1n00237F↵ (zmena mena na @, čo nie je znak ktorý môže byť použitý ako meno), tak dostaneme od robota odpoveď: !1I000059↵ , čo znamená, že parameter je nepoužiteľný pre túto funkciu.														
O	OK – robot vracia tento paket keď vykonanie príkazu prebehlo úspešne. Robot ho vracia pre príkazy: F, B, L, R, f, b, r, l, S, D, n - v prípade ich úspešného vykonania a správnych parametrov.														
a	Stav analógového vstupu – po príkaze na poslanie stavu analógového vstupu robot pošle posledné meranie. Odpoveďový paket má parameter tvaru xyyy , kde x je číslo analógového vstupu, ktorého stav je vysielaný, a yyy je hexadecimálne vyjadrenie hodnoty z A/D prevodníka v rozsahu od 0x000 – 0V až do 0x3FF, čo je referenčné napätie, alebo napätie od neho vyššie. Príklad paketu: !1a202F07↵ . Tento paket hovorí o tom, že na kanále 2 je napätie hodnoty 0x02F, čo je po prepočte približne 0,11V, s referenčným napätím 2,56V.														
b	Odpoveď na požiadanie o stav nárazníkov. Odpoveď bude mať parameter tvaru 000x, kde x je hexadecimálna hodnota od 0 do 15 vyrátaná ako: $\text{hodnota} = (1 * \text{stav A nárazníka}) + (2 * \text{stav B nárazníka}) + (4 * \text{stav C nárazníka}) + (8 * \text{stav D nárazníka})$ Stav nárazníka je 1 ak nie je kolízia, 0 ak je kolízia.														
A	Odpoveď na ping. Pokiaľ robota požiadame o príkazom A alebo N , tak dostaneme odpoveď, kde parameter je tvaru xxxx, kde xxxx je hexadecimálne vyjadrená verzia firmwaru.														
xxxx	Parameter odpovede – robot posielá späť to, čo sme od neho chceli. Formát parametru závisí od jeho typu.														
CC	Rovnaké ako u príkazu														
↵	Rovnaké ako u príkazu														

Protokol nemá žiadne timeouty, takže nie je nutné sa ponáhľať s písaním príkazov.

Programovanie robota

Programovaním je teraz myslené nahranie vlastného programu do robota. To je možné robiť dvoma spôsobmi – použiť **bootloader** alebo použiť **ISP rozhranie**.

Pri programovaní cez **ISP rozhranie** máme viacero možností, než len nahrávať program – môžeme nastavovať poistky procesora – napr. zabrániť ďalšiemu zápisu do pamäte, zmeniť zdroj hodín a iné nastavenia. Viac o jednotlivých poistkách sa môžete dočítať v datasheetu procesora ATmega128.

ISP konektor má štandardné rozloženie pinov – viz. Časť o elektronike hlavnej dosky, kde je bližšie popísaný. Ako programátor môžeme použiť alebo jeden z oficiálnych programátorov – napr. AVR Dragon, alebo JTAG ICE Mk2 alebo AVR ONE, no ak nám nevyhovuje ich cena, je možné si vyrobiť vlastný programátor, podľa návodov v [6]. Programovanie potom už závisí od toho, ktorý programátor si zvolíme.

Poznámka: ISP rozhranie zdieľa piny so sériovou linkou 0, takže počas programovania na sériovej linke 0 nesmie byť pripojené žiadne iné zariadenie.

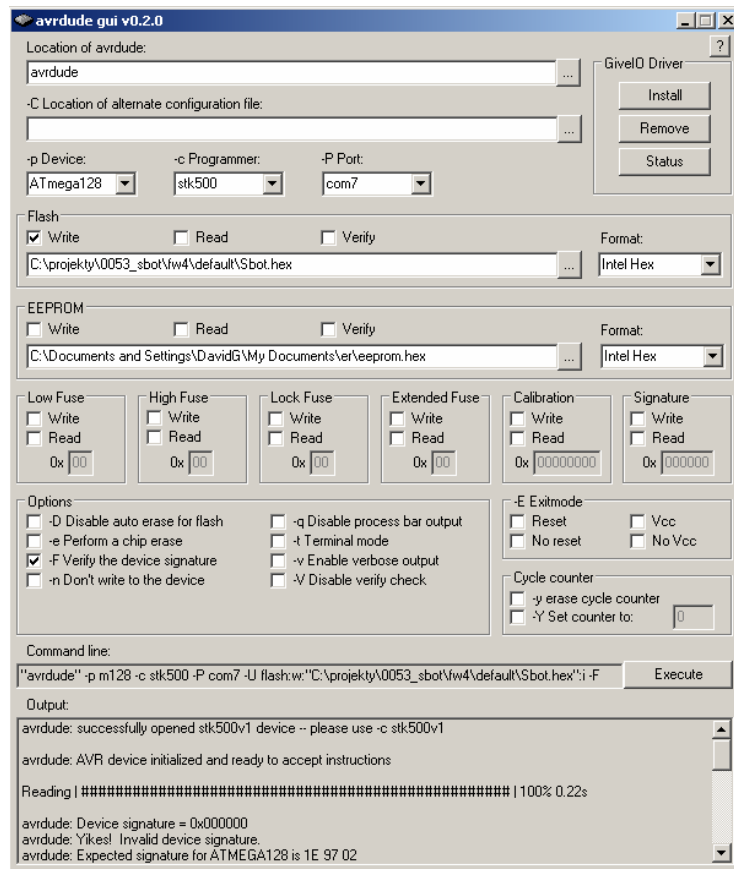
Alternatívou k použitiu externého programátora je **bootloader**. Bootloader je malá aplikácia nahraná na konci pamäte robota, ktorá sa stará o to, aby sa program dostal z počítača do pamäte robota. Jeho výhodou je, že nepotrebujeme žiaden ďalší hardware a bootloader komunikuje rovno s počítačom cez Bluetooth. Stačí sa pripojiť na robota tak, ako to bolo opísané v časti o **Oživovaní SBota**, a následne sa s priloženým programom AVRdude pripojiť na robota.

Pred samotným programovaním je nutné mať nainštalovaný balík WinAVR, ktorý obsahuje aj program AVRdude.

Postup ako robota dostať do režimu bootloaderu:

1. Vypnite robota
2. Skratujte **Prepojku na spustenie Bootloadera** – na skratovanie je na nej malý jumper – osadíte ho.
3. Zapnite robota
4. Pripojte sa na robota vez Bluetooth tak, ako je popísané v časti o oživovaní SBota.

Teraz je možné sa pripojiť na robota bootloaderom. Spustíme aplikáciu Avrdude GUI, ktoré je priložené na CD.



Na obrázku vedľa je vidno screenshot z programu. Ako zariadenie je nutné nastaviť procesor ATmega128, ako programátor STK500 a do Port nastavenia je nutné napísať meno COM portu, na ktorom máme pripojeného robota cez Bluetooth (v screenshote na obrázku vedľa je na COM7). V časti Flash si zvolíme hex súbor vytvorený kompilátorom z nášho programu, ktorý chceme vložiť do procesora. EEPROM je možné tiež programovať, no je nutné dbať na to, že na adrese 0 je uložená ASCII hodnota čísla robota. Okrem toho je nutné ešte zaškrtnúť –F, tak ako je na obrázku. Potom stačí stlačiť Execute tlačítko.

Po nahratí programu do robota môžeme program avrdude zrušiť.

Teraz je treba resetovať robota. Najprv však je treba odstrániť jumper z prepojky na spustenie bootloadera, aby po resete robot nenabehol znova do bootloaderovského režimu. Potom stačí robota vypnúť a zapnúť, alebo stlačiť RESET tlačítko na hlavnej doske.

Po tomto by sa mal spustiť váš program.

Framework firmwaru

Aby bolo vytváranie programu do robota jednoduché nie je nutné sa veľmi hrať s registrami procesora, stačí použiť sadu makier a funkcií ktoré boli vopred pripravené. Základný framework je aj pre A/D prevodník a aj senzory a ďalšie časti robota – motory apod.

Vo funkciách, kde je možné si vkladať vlastný kód (napr. prerušenie ktoré nastane každých 0,01 sekundy) je komentár, označujúci kde je možné vkladať vlastný kód. Pri prerušeniach je vhodné, aby kód bol čo najkratší.

Funkcie a makrá pre A/D prevodník – adc.c a adc.h

unsigned int adc_vals[8]; - Jedná sa o pole 8 integerov, v ktorých je uložené posledné meranie A/D prevodníka. Všetky používajú ako referenčné napätie napájacie napätie, okrem kanálu 8, ktorý meria stav batérie. Tam je použité referenčné 2,56V. Ukážka programu:

if(adc_vals[7] < 500) set_leds(0x0F);

Program skontroluje, či je napätie na kanáli 8 vyššie ako určitá hodnota, ak nie, tak zasvieti všetky LED diódy.

void start_adc(); - toto makro spustí A/D prevodník tak, aby vykonal jeden prevod a potom skočil do prerušenia. A/D prevody sú robené automaticky, takže sa ním netreba zaoberať.

adc_select_ref(unsigned char x); - vyberie referenčný zdroj pre A/D prevodník: môže byť vstup alebo **ADC_REF_256V** (vyberie referenčný zdroj 2,56V) alebo **ADC_REF_AVCC** (vyberie referenčný zdroj z bateriek)

ISR(ADC_vect) – toto prerušenie nastane vždy, keď sa ukončí jeden prevod A/D prevodníkom. Do tejto funkcie je možné vkladať si vlastný kód.

Funkcie a makrá pre časovač – timers.c a timers.h

unsigned int ticks_until_stop; - táto premenná určuje koľko jednotiek času má robot ešte ísť daným smerom. Jedna jednotka času je rovná 0,01 sekunde. Ak je rovná 0, tak vypne servomotory. Ak je rovná **TIMER_SERVO_CONTINUOUS_OPERATION** tak robot pôjde daným smerom neustále. Ak chceme povedať robotovi aby išiel niekam, treba nastaviť aj túto premennú. Príklad programu:

```
ticks_until_stop = 1000;
set_servo(1,SERVO_FW);
set_servo(4,SERVO_FW);
servo_123_on();
servo_456_on();
```

Robot sa bude po tomto kóde 10 sekúnd otáčať vpravo.

ISR(TIMER0_COMP_vect) – toto prerušenie nastáva každých 10 ms – 0,01 sekundy. Je možné si doň vložiť vlastný kód, no mal by sa čo najkratšie vykonávať. Prerušenie sa stará o blikanie predných LED diód, a o časovanie pohonu.

Funkcie a makrá pre pohon – servo.c a servo.h

void set_servo(unsigned char which, unsigned int value); - táto funkcia nastaví hodnotu riadiacej PWM pre vybraný servomotor. Servomotory (prvý parameter - **which**) môžeme voliť od 1 do 6 pre nastavenie jedného servomotora, alebo 0 pre nastavenie všetkých servomotorov na danú hodnotu. Ako druhý parameter zadávame hodnotu. Motor môže ísť vpred alebo vzad, alebo stáť. Ak má ísť vpred, je treba použiť konštantu **SERVO_FW**, ak má ísť vzad, ako parameter zadáme **SERVO_BW** a ak má stáť, je treba použiť **SERVO_STOP** konštantu. Ak má motor stáť, je vhodné nutné ho vypnúť makrom **servo_xyz_off()**, aby nespotreboval zbytočne energiu. Ľavý motor má číslo 1, pravý motor je zapojený na výstup číslo 4.

Poznámka: Robot má automatický časovač, ktorý vypína motory. Aby ich nevypol keď ich zapnete, je nutné nastaviť globálnu premennú **ticks_until_stop** na hodnotu **TIMER_SERVO_CONTINUOUS_OPERATION**. Tým vyradíte časovač a servo pôjde stále. Pokiaľ chceme aby išlo len určitý čas, napr. 10 sekúnd, nastavíme túto premennú na hodnotu, ktorú vyrátame ako **HODNOTA = (počet sekúnd, koľko majú servá ísť) x 100**. Z **ticks_until_stop** môžeme aj čítať, a tak zisťovať, kedy motor zastal.

Príklad použitia:

```
ticks_until_stop = TIMER_SERVO_CONTINUOUS_OPERATION;
servo_123_on();
servo_456_on();
set_servo(1, SERVO_BW);
```

set_servo(4, SERVO_FW);

Tento kód spôsobí, že robot pôjde vzad tak dlho, dokým nedostane iný príkaz.

servo_123_on(), servo_456_on(), servo_123_off(), servo_456_off() – tieto makrá slúžia na zapojenie a odpojenie napájania od servomotorov. Servomotory sú organizované po trojiciach – 1, 2, 3 a trojica 4, 5, 6. Makro **servo_on_xyz** pripojí napájacie napätie na servomotory v trojici xyz a **servo_off_xyz** odpojí im napájacie napätie.

Funkcie a makrá výstupu a vstupu

stdout – je štandardný výstup. Naňho je nasmerovaná funkcia printf. Tento výstup možno podľa potreby meniť. V základnom nastavení je nastavený na sériovú linku 1, ktorá je pripojená na BlueTooth. Stdout je možné nasmerovať aj na nejaké iné zariadenie, no je nutné z neho najprv urobiť „súbor“, pomocou funkcie **FDEV_SETUP_STREAM**. Príklad jej použitia :

FILE usart1 = FDEV_SETUP_STREAM(usart1_putchar, NULL, _FDEV_SETUP_WRITE);

Tento riadok kódu vytvorí pointer, ktorý môžeme použiť ako stdout. Prvým parametrom je pointer na funkciu pomocou ktorej bude zapisovať jeden byte na zariadenie, druhým je funkcia na čítanie jedného bytu zo zariadenia a posledným je RW parameter, ktorý určuje, čo sa dá všetko robiť so zariadením. Viac o tomto je možné nájsť na webstránke http://www.nongnu.org/avr-libc/user-manual/group_avr_stdio.html.

Po tomto stačí už len nasmerovať stdout na naše zariadenie, napr.:

stdout = &usart1;

Odteraz môžeme používať printf ako výstup na BlueTooth.

Definované sú „súbory“ usart0 a usart1, no je možné vytvoriť si ďalšie vlastné.

int usart0_putchar(char c, FILE *stream) – pre ser. link. 0

int usart1_putchar(char c, FILE *stream) – pre ser. link. 1

Táto funkcia pošle jeden znak – ktorý je ako prvý parameter, na sériovú linku 0 alebo 1. FILE parameter je tam potrebný z dôvodu, aby sa funkcia dala použiť na stdout, je vhodné zaňho dávať null hodnotu. Príklad použitia:

usart0_putchar('a', null);

Tento riadok vyšle sériovou linkou 0 znak a. Funkcia vracia hodnotu 0.

unsigned char usart1_getchar(void) – pre ser. link. 0

unsigned char usart0_getchar(void) – pre ser. link. 1

Táto funkcia počká na znak zo sériovej linky 0 alebo 1, a potom ho vráti ako návratovú hodnotu funkcie. Príklad použitia:

if(usart1_getchar()=='a')set_leds(0x0F);

Tento riadok kódu spôsobí, že ak znak, ktorý najbližšie príjme bude a, tak sa zapnú LED diódy.

Štruktúra zdrojových kódov

Program robota je písaný v jazyku C a je rozdelený podľa toho, o ktorú časť robota sa funkcie v súboroch starajú. Zoznam súborov a čo obsahujú:

\Adc.c, adc.h	A/D prevodník, prerušenie A/D prevodníka
\Main.c, main.h	Hlavná funkcia, inicializácia HW, spracovanie paketov od vyššieho systému
\Misc.c, misc.h	Nezaraditeľné funkcie, ako nastavenie LED diód, alebo čakanie, alebo zistenie stavu nárazníkov
\Servo.c, servo.h	Nastavenia servomotorov

SBot2.0 – dokumentácia

\Timers.c, timers.h	Prerušenie, ktoré nastane každých 10ms
\Usart0.c, usart0.h	Sériová linka 0
\Usart1.c, usart1.h	Sériová linka 1 – používaná BlueToothom
\Protokol_defs.h	Definície pre komunikačný protokol
\Makefile	Make súbor – script ktorý sa stará o to, aby sa program skompiloval, je generovaný automaticky AVRStudiom.
\Sbot.aps, sbot.aws	Súbory AVRStudia – informácie o projekte
\default\sbot.hex	Výstupný .hex súbor, ktorý treba nahrat' do procesora
\default\vsetko ostatne	Vedľajšie súbory z kompilácie

Veci na CD

K robotovi je priložené aj CD obsahujúce rôzne užitočné veci na robota:

- WinAVR balík aplikácií – verzia , AVR Dude – bootloader
- Datasheety jednotlivých súčiastok v robotovi
- Zdrojové kódy robota a bootloadera
- Skompilované hex súbory
- SbotManager.exe – program pre manuálne ovládanie SBota
- Túto dokumentáciu

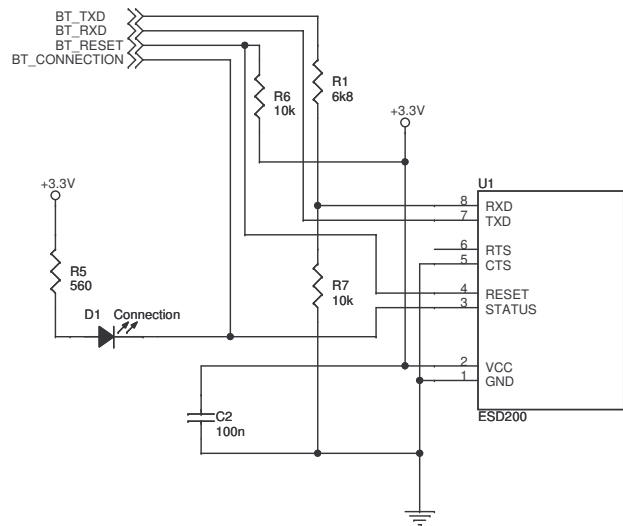
Z licenčných dôvodov nebolo možné na CD priložiť AVRStudio – vývojové prostredie pre procesory AVR, no je možné si zadarmo stiahnuť jeho najnovšiu verziu zo [4].

Prílohy

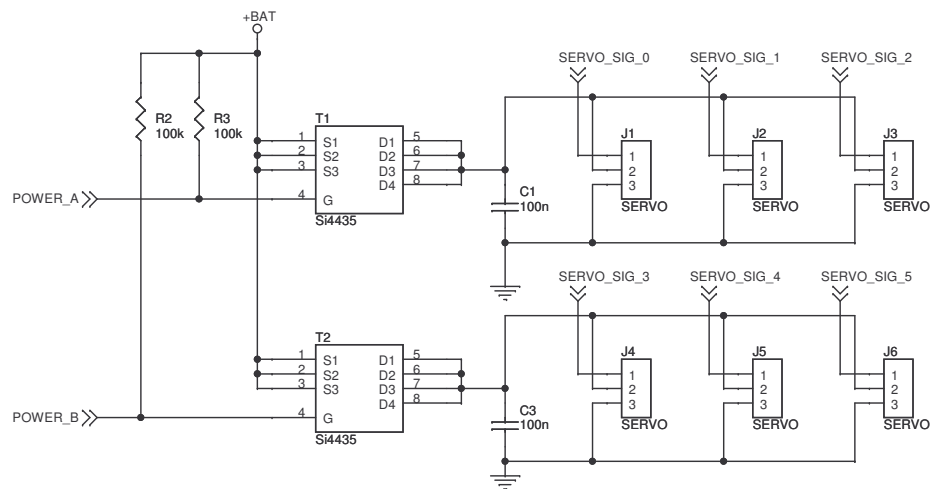
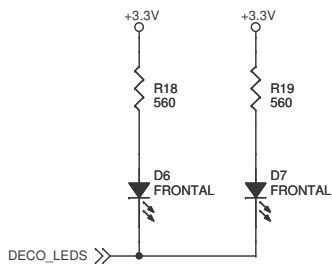
Strana	Obsah
1	Schéma robota – BlueTooth, zdroj, spínanie servomotorov a ich pripojenie
2	Procesor, LED diódy a nárazníky
3	Univerzálna doska
4	Bloková schéma robota

Linky na užitočné stránky

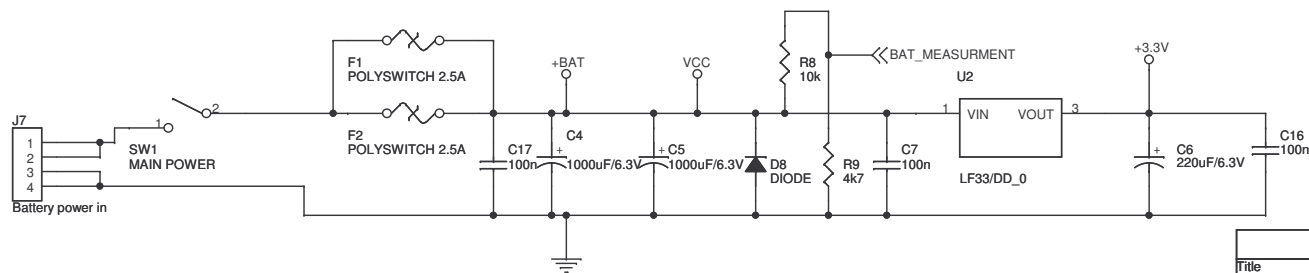
- [1] <http://www.microstep-mis.com/> - výrobca a distribútor tohto robota
- [2] <http://www.robotika.sk/> - Slovenský portál robotiky
- [3] <http://gme.sk/> - distribútor súčiastok na Slovensku
- [4] <http://atmel.com/> - výrobca AVR procesorov
- [5] <http://winavr.sourceforge.net/> - C kompilátor pre AVR procesory
- [6] <http://www.lancos.com/prog.html> - jednoduchý programátor pre procesory AVR



BLUETOOTH

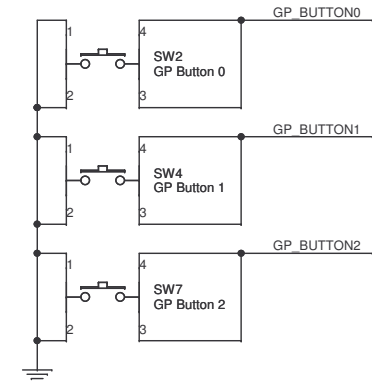


SERVA A ICH SPINANIE

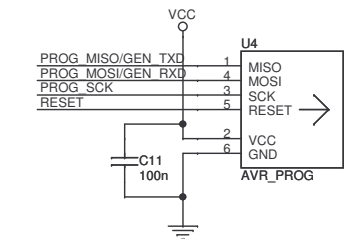


ZDROJ

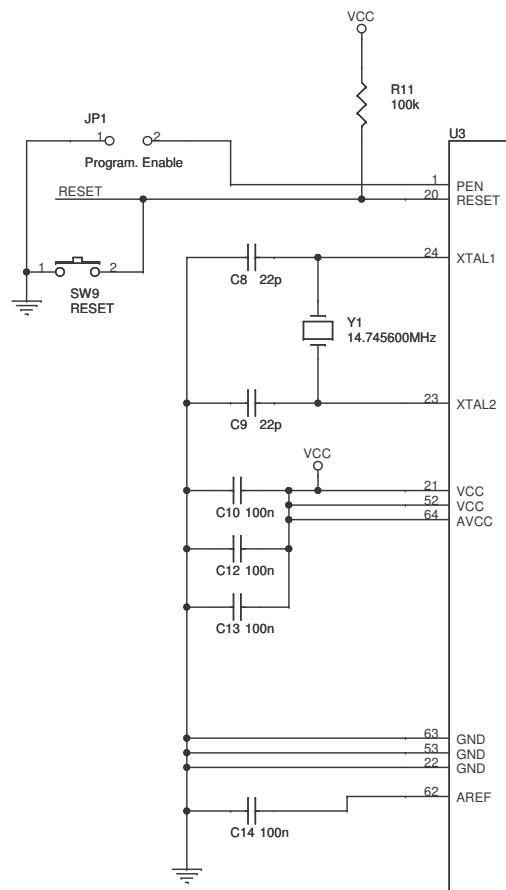
Title		
SBot zdroj a serva		
Size	Document Number	
B		
Date:	Tuesday, July 08, 2008	Sheet 1 of 4
Rev	<RevCode>	



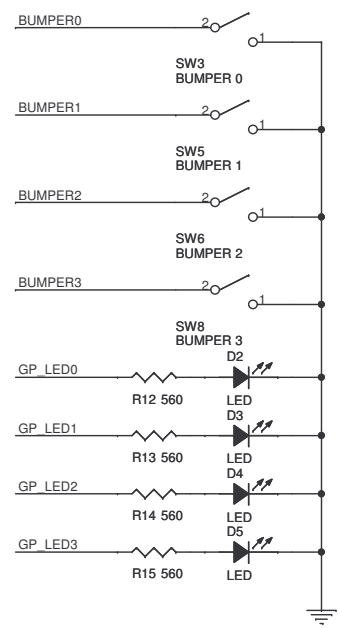
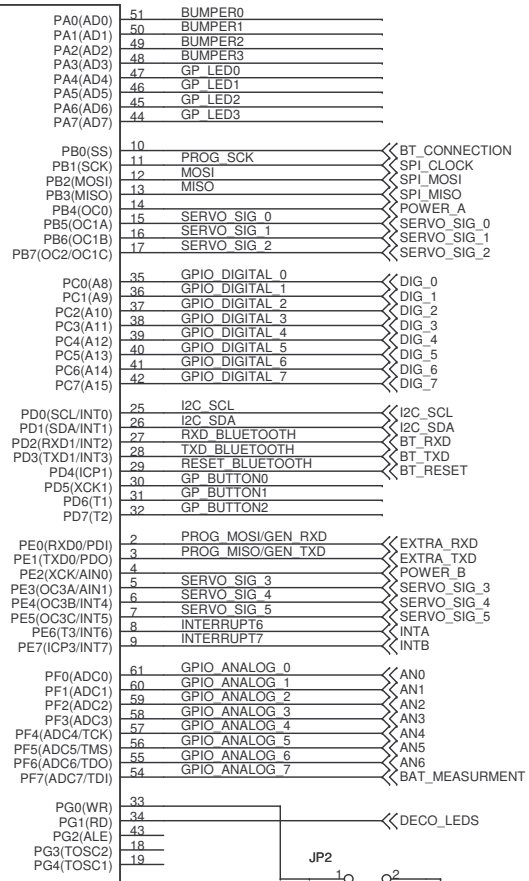
UZIVATELSKE TLACITKA



ISP PROGRAMATOR

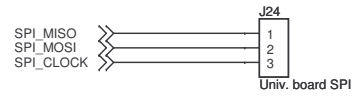
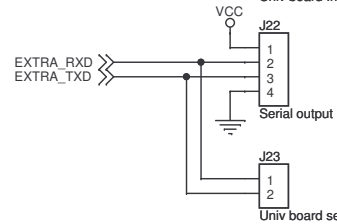
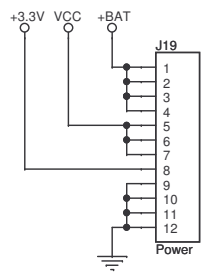
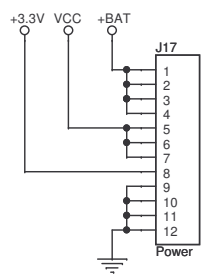
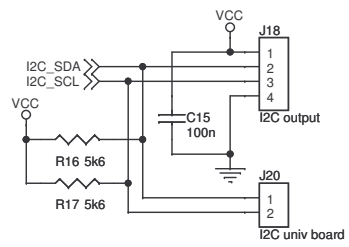
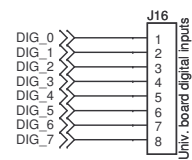
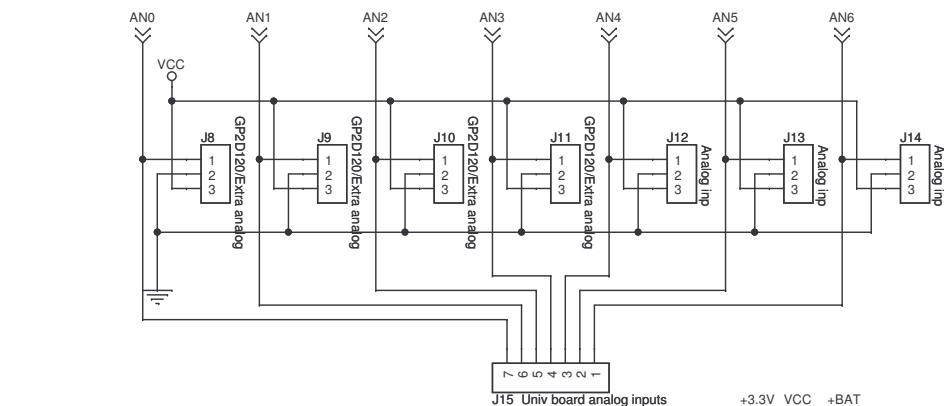


PROCESOR



LED DIODY A NARAZNIK

Title			Sbot procesor		
Size	B	Document Number	<Doc>		Rev
Date:	Tuesday, July 08, 2008		Sheet	3	of 4



UNIVERZALNA DOSKA

Title		
Sbot - univerzalna doska		
Size	Document Number	Rev
B	<Doc>	<RevCode>
Date:	Tuesday, July 08, 2008	Sheet 4 of 4

Blokova schema robota

