

Ovládanie robota pomocou mobilného telefónu

Jackuliak Lukáš · Elektrotechnika, Študentské práce

05.10.2011



Hlavným cieľom našej práce, bolo zabezpečiť bluetooth komunikáciu medzi robotickým systémom a mobilným zariadením. Po zabezpečení komunikácie sme sa zamerali na riadenie robotického systému. Klientská aplikácia v mobilnom zariadení vytvorená na platforme Java MicroEditon s podporov MIDlet. Pripojením mobilného zariadenia k

bluetooth modulu v robotovi, môžem ovládať vzdialenosť, uhol natočenia a rýchlosť, ktorým smerom sa má robot pohybovať.

1. Úvod

Následne robot spätne posiela údaje či prijal údaje, ktoré mu boli odoslané z mobilného zariadenia, potvrdenie o natočení o požadovaný uhol a aktuálnu polohu. Ak robot narazí na prekážku automaticky zastane a vyšle kód mobilnému zariadeniu o snímačoch a tým sa sprístupní v aplikácii možnosť zadania novej polohy. V prípade ak by sa niečo nečakané stalo aplikácia obsahuje tlačidlo „central stop“, ktoré po stlačení vyšle reťazec aby robot okamžite zastal. Robot je postavený na diferenciálnom podvozku, ktorý je poháňaný dvomi krokovými motormi. Robota riadi mikrokontrolér ATmega32.

2. Popis aplikácie

Aplikácia pre mobilné zariadenie je stavaná pre dotykový mobil s podporov MIDlet technológie. Dotykové mobilné zariadenie od spoločnosti Samsung typ GT-S5620 podporuje Java aplikácie a MIDlet 2.1. Bluetooth komunikácia s podporov Java nástrojov Java API a knižníc java.bluetooth .

3. Odosielanie

Vo vlákne volaná funkcia „send“ zabezpečuje odosielanie dát otvorením connector.open a cez OutputStream. Dáta sú odoslané tlačidlom „Pošli“ , ktoré sa nachádza v zložke „Control“. V tejto zložke je dôležité zadať pozíciu kam chceme poslať robota a pomocou algoritmov, ktoré sú v tejto zložke sa vypočíta uhol vo funkcií „uholprepocet“ a vzdialenosť vo funkcií „prepocet“. Rýchlosť má 6 stupňov, ktoré sa dajú nastaviť. Zatlčením tlačidla „Pošli“ odosiela používateľ robotovi kód, ktorý obsahuje: Začiatkový znak pre overenie hex číslo 0xAA a vzdialenosť vypočítavanú funkciou „prepocet“ ktorá je v rozsahu 0-100. Uhol vypočítavaný vo funkcií „uholprepocet“, ktorý je v rozsahu 0 - 180.

Znamienko k uhlu aby robot vedel do ktorej strany sa má natočiť +/- a jedno miesto na rýchlosť Konečný znak pre overenie hex číslo 0x0F. Celkový zápis v hex: AA 00 00 00 0F. V zložke „Control“ sa nachádza aj tlačidlo „CentralStop“ pod názvom stop , zatlačením tlačidla vyšle robotovi automaticky kód v hex AA 53 54 50 0F. Tento kód dáva znamenie robotovi aby okamžite zastavil. Po každom odoslaní pre kontrolu 4 krát mobilné zariadenie čaká na prijatie potvrdenia o prijatí až potom môže používateľ znovu zadať novú pozíciu kam sa má robot pohnúť.

4. Prijatie dát od robota

Prijatie dát cez bluetooth do robota má za úlohu funkcia „prijmi“, ktorá má vstupnú premennú InputStream pomocou, ktorej môžem čítať dáta. Vo funkcii je vytvorené bytové pole „dataprijmi“. Do poľa sa bude vkladať všetko čo príde na vstup. InputStream obsahu funkciu `is.read(dataprijmi, 0, dataprijmi.length)`. Následne si dáta prepíšeme do stringu `str = new String(dataprijmi)`. Tieto hodnoty v stringoch porovnávam a zisťujem o akú kódovú správu ide.

Pre správne prijatie správy ako spätnú väzbu dostávame kód: AOK00000F. Stav robota o kolko sa otočil podľa toho aký uhol sme mu poslali, kde šieste písmenko buď P alebo N nám udáva P+ alebo N- uhol, uhol otočenia 90°: AUHOKP090F, AUHOKN090F. Pre potvrdenie, že robot sa dostal na žiadanú hodnotu a následné poslanie nových hodnôt prijmem kód od robota: APOOK0000F. Ak sa niečo stane s robotom pošle nám kód: AER00000F, tým sa v zložke „Graf“ zobrazí status robota ako ER.

Od robota budeme dostávať aktuálnu pozíciu cca každé 2s, správa bude zakódovaná v tvare: AC0506000F , kde C indikuje že ide o prijatie vzdialenosti, v ktorej sa aktuálne nachádza a rýchlostný stupeň.



Obr. 1. Prijímanie a odosielanie dát

5. Hlavné menu

Menu obsahuje položky:

- Connection - pripojenie mobilného zariadenia k robotovi cez RFCOMM protokol
- Control - ovládanie vzdialenosti , uhlu a rýchlosti robota, posielanie
- Simulation - simulácia dráhy robota
- Settings - nastavenia , farby , prepočet na rad/deg , vlastné zadanie vzdialenosti rýchlosti a uhlu.
- Graf - zobrazovanie , rýchlosti , uhlu , súradníc , status robota , prichádzajúce kódy.
- About - o programe, ako ovládať program.



Obr. 2. Hlavné menu zobrazené hneď po spustení

6. Connection (pripájanie)

V prípade nášho mobilného zariadenia je potrebné zapnúť bluetooth ešte pred spustením aplikácie. Dôvod je nasledovný. Mobilné zariadenie Samsung GT-S5620 má svoj vlastný operačný systém, ktorý je uzamknutý a preto sa nám nepodarilo naprogramovať prístup aplikácií k zapínaniu bluetooth, aby sa automaticky pri stlačení položky „Connection“ zapol bluetooth.

Po zapnutí bluetooth v mobilnom zariadení a stlačení položky „Connection“ začne mobilné zariadenie vyhľadávať bluetooth modul v robotovi. Na displeji je toto hľadanie zobrazené ako start Inquiry... , následne nasleduje hlásenie Searching for device značí, že hľadá zariadenie. Hlásenie Device Discovered oznamuje, že zariadenie bolo nájdené a vypíše následne jeho bluetooth adresu a meno zariadenia a tým to sa start Inquiry končí hláškou InquiryCompleted.

Mobilné zariadenia načína hľadať servis bluetooth modulu v robotovi. Ak sa nepodarí nájsť servis zariadenie na displeji sa vypíše hlásenie SERVICE_SEARCH_NO_RECORDS. Ak nastane chyba pri hľadaní servisu zobrazí sa hlásenie SERVICE_SEARCH_ERROR. V prípade ak servis nie je zistiteľný hlásenie bude v tvare SERVICE_SEARCH_DEVICE_NOT_REACHABLE, môže nastať situácie že pri hľadaní nastane chyba v mobilnom zariadení tým sa hlásenie zmení na SERVICE_SEARCH_TERMINATED. Servis sa nájde vypíše na displeji hlásenie SERVICE_SEARCH_COMPLETED.

Nasleduje otvorenie komunikácie. Otvorenie komunikácie je založené na URL adrese. Adresa obsahuje

```
btsp://adresazariadenia:,UUID,Master=false,encrypt=false,authenticate=false
```

btsp je bluetooth protokol, kde spp (serial port profile) znamená, že sa komunikáciu emuláciou sériového portu. Adresu zariadenia získam pri hľadaní zariadenie, UUID je identifikácia protokolu RFCOMM. V programe ju nastavujem `uuidSet[0] = Bluetooth.RFCOMM_UUID;`

7. Control (Riadenie)

Riadenie robota cez mobilné zariadenie funguje na základe posielanie údajov o vzdialenosti, uhle natočenia a rýchlosti. Všetky údaje sú prepočítavané v mobilnom zariadení a následne stlačením tlačidlom pošli poslané robotovi.

Súradnicový systém v mobilnom zariadení je definovaný od ľavého horného rohu, kde začína hodnotou $x, y = [0,0]$ a maximálne rozlíšenie je 400×240 bodov. Stred súradnicového systému je v bode $x, y = [200,120]$. Keďže pri zapnutí aplikácie sa aktivuje klávesnica querty, použiteľné rozlíšenie sa znižuje, preto sme boli nútení zmeniť stred súradnicového systému na $x, y = [120,120]$. Vzdialenosť vypočítavame z hodnôt x a y pri dotyku s displejom. Aby sme dostávali správne hodnoty museli sme vypočítané hodnoty upraviť o stred, o ktorý sú posunuté do hodnoty $x, y = [0,0]$.

Vyhodnocovanie súradníc x a y zabezpečuje funkcia vstavaná v canvase `pointerPressed(int x, int y)` a funkcia `pointerDragged(int x, int y)`.

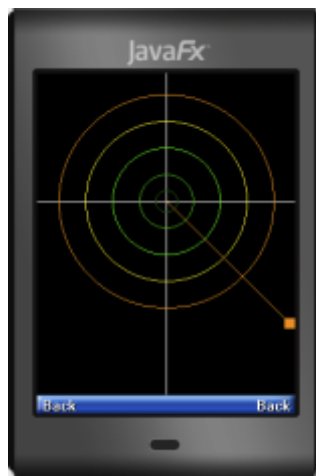
Na obrázku je zobrazená simulácia mobilného zariadenia, v ktorej beží aplikácia. Control obsahuje šípky pomocou ktorých je možné zvyšovať a znižovať rýchlosť, súradnicový systém a tlačidlo „Stop“, ktoré zabezpečuje zastavenie robota v prípade problému. V dolnej časti aplikácie sa zobrazujú údaje o súradniciach x a y , vzdialenosti, uhle otočenia a rýchlosti. Tlačidlom pošli sa posielajú dáta robotovi. Aplikácia sa následne prepne do položky Graf.



Obr. 3. znázorňuje aplikáciu v položke Control na riadenie vzdialenosti, uhla natočenia a rýchlosti robota

8. Simulation (Simulácia)

Simulácia pohybu robota. Prijaté dáta od robota, ktoré posiela každé 2 sekundy. Dáta si uložíme ako string a následne si ho rozložíme na čísla.

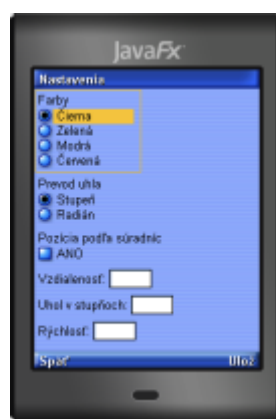


Obr. 4. Zobrazenie položky simulácia

9. Settings (Nastavenia)

V nastaveniach je možné zmeniť si farbu pozadia simulácie, prepnúť prepočítavanie uhlu v stupňoch na radiány a možnosť poslať robota na presné súradnice aké chceme. Prepínanie farieb pomocou ChoiceGroup. Pri zmene farby a stlačení tlačidla uložiť sa prepíše premenná a v simulácii sa splní podmienka podľa toho akú farbu sme si zvolili a zmení sa farba príkazom `mod.setColor(RGB);`

Prepočet na radiány sa prepočítava funkciou `atan(x,y)`. Využívajú sa pritom tabulkové hodnoty, ktoré sú v poli integer. Pomocou tejto tabuľky sa prepočítajú hodnoty na stupne a následné prepnutie v nastaveniach na radiány zmení hodnotu prevodu v podmienke na 1. Premenná „pocitadlo“ reprezentuje uhol v stupňoch. Táto hodnota sa podelí konštantou 57.3 a priradí sa do „floatpocitadlo“ následne hodnotu podelíme 0.01 a predefinujeme na integer. Táto zmena oreže desatiny miesta. Ďalšie delenie 100 a priradenie do floatu pridá dve desatiny miesta aby sme dostali hodnotu v tvare dvoch desatinných miest. Do premennej „Bluetooth.uhol“ sa priradí string, ktorý sa vykresľuje na displej.



Obr. 5. Zobrazenie nastavení v aplikácii spolu so zadávaním údajov

10. Nastavenie vlastných súradníc

Na displeji v položke Nastavenia je potrebné zaškrtnúť „Ano“ v pozícii podľa súradníc. Zadeinovať vzdialenosť v tvare 000 teda ak chceme dať vzdialenosť 8 musíme zadať 008. Zadeinovať uhol v stupňoch rozsah je od 0° - 180°. Zápis v tvare 000 v prípade, že chceme zadať uhol 45° zadáme 045. Zadeinovanie rýchlosti je v tvare 000 a rozsah

je od 0 - 6 teda tvar na definovanie je 004 pre rýchlosť 4. Celé zadávanie súradníc sa zakladá na podmienke ak je ChoiceGroup zaškrtnutý podmienka je splnená a do pola selected2 sa priradí hodnota true alebo false.

Priradenie je zabezpečované metódou Nas2.getSelectedFlags(selected2). Ak je aj nasledovná podmienka splnená teda selected2[0] = true. Priradí sa int premennej su hodnota 1. Hodnota premennej su zmení nastavenie posielanie robotovi. Do stringu pzv, pzu, pzs sa priradí string zo zadaných hodnôt vzdialenosti, uhla a rýchlosti. V cykle for, ktorý sa zopakuje 3 krát pretože hodnoty majú tvar 000 sa priradí do charového pola char v prislúchajúcom stringu. A následne prevedené do int pomocou funkcie Integer.parseInt(String).

11. Graf

Na displeji sa zobrazujú aktuálne príkazy, ktoré prichádzajú z mobilného zariadenia. Zobrazuje sa aktuálny stav robota a hodnoty vzdialenosti, uhla a rýchlosti. Stav robota zobrazený v mobilnom zariadení:

- OK - potvrdenie o prijatí súradníc, ktoré sú posielanie z mobilného zariadenia AOK000000F
- U OK - potvrdenie od robota o otočení podľa uhla, ktorý bol zadaný + uhol o koľko sa otočil AUOK09000F.
- P OK - potvrdenie o konečnom stave, robot sa dostal na požadovanú súradnicu a vyžaduje zadanie nového bodu, tým to príkazom sa sprístupní aj posielanie súradníc, ktoré je zablokované kým nepríde kód APOK00000F
- NN - prijatie od robota, stav snímačov
- RE - prijatie od robota, aktuálne súradnice vzdialenosti a rýchlosti, každé dve sekundy. Pomocou týchto súradníc sa vykresľuje simulácia aktuálnej polohy robot AC1002000F
- ER - prijme od robota chybové hlásenie AER000000F, zastavenie komunikácie

Vykresľovanie grafu rýchlosti je založené na kreslení čiary v canvse a inkrementovaní premennej o jedna. Zobrazenie sledovania senzorov prijímané od robota, ak príde kód ANN111100F všetky senzory sú v poriadku ako náhle pri prijímaní dát zobrazí kód AN011100F znamená to, že robot je pred prekážkou a treba zadať nový smer. Zobrazovanie je prevedené kreslením štyroch zelených štvorcíkov z označeniami 1 až 4 v prípade, že robot sa dostane pred prekážku zmení sa farba štvorca na červenú. ANN111100F, kódovanie je nasledovné:

- A - začiatok reťazca
- NN - indikuje, že pôjde o snímače
- 1111 - robot má 4 snímače dva vpredu a dva vzadu, prvé dve čísla 11 indikujú, že ide buď o snímače vpredu alebo vzadu, druhé dvojčíslo indikuje či ide o snímač vpravo alebo vľavo.
- 0F - ukončenie reťazca

Príklad: ANN100100F - vzadu, ľavý snímač narazil na prekážku, ANN101000F - vzadu, pravý snímač narazil na prekážku.



Obr. 6. Zobrazenie „želaných hodnôt a hodnôt získavaných z robota, zobrazenie kódov

12. Rozkódovanie prijatých kódov

V prípade prijatých kódov je rozkódovanie spracúvané do dátového typu char a následne predefinované na int. Rozdelenie prijatého stringu AC0205000F do charu pomocou cyklu for. Cyklus for sa zopakuje toľko krát koľko je dĺžka príslušného stringu a do pola char[] arr priraduje aktuálnu hodnotu. V canvase funkciou Paint(Graphics g) je pomocou premennej g vykresľované charové pole na displej mobilného zariadenia.

13. About (O programe)

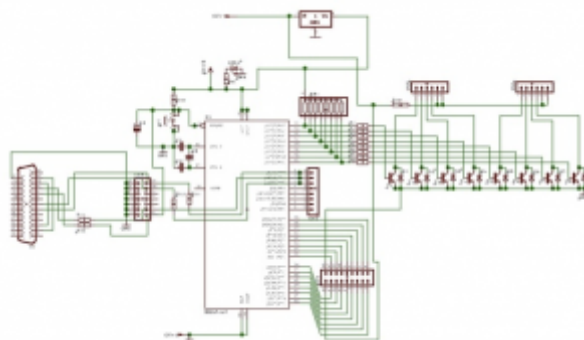
V položke „About“ je zobrazený kompletný návod ako správne ovládať program a ako sa riadi robot. Vytvorený za pomoci triedy Canvas, ktorá obsahuje grafické nástroje a funkciu paint(). Vytvorením Itemu Image imag a zadaním do imag cestu k obrázku je vykresľovaný obrázok vo funkcií paint() v triede Canvas.

14. Popis robota

Robot sa skladá z niekoľkých častí. Prvá z nich tvorí mechanická. Skladá sa z diferenciálneho podvozku s dvomi krokovými motormi Sanio CP-047. Z nich je moment prenášaný na kolieska pomocou ozubeného prevodu s pomerom 1:4. Podvozok má namiesto tretieho pasívneho kolieska statický oporný bod, ktorý sa kľže po podložke. Konštrukcia je vyrobená z plechu hrúbky 0,5mm, ktorý pochádza zo starej počítačovej skrinky. V jej vnútri je priestor pre 8 akumulátorov typu AA, ktoré sú zdrojom energie pre robota.

15. Hlavná doska

Robot je riadený mikrokontrolérom ATmega32 od spoločnosti Atmel. Ten komunikuje prostredníctvom bluetooth modulu s aplikáciou v mobilnom telefóne. Okrem neho sa na hlavnej doske nachádza stabilizátor napätia L78L05, ktorý „vyrába“ napájanie pre mikrokontrolér. Taktiež sú na tejto doske umiestnené budiče pre motory. Tieto sú tvorené zo spínacích tranzistorov BU406. Existujú aj integrované budiče, avšak z dôvodu väčšej prúdovej zaťažiteľnosti sme sa rozhodli použiť tranzistory v separátnych púzdrach. Všetkých osem tranzistorov je mechanicky spojených elektricky izolovaným chladičom.



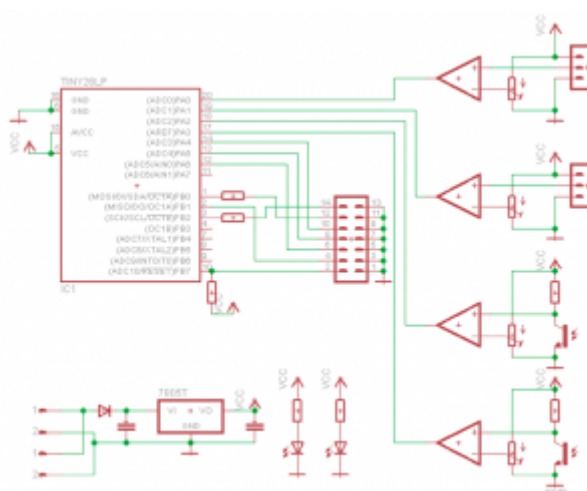
Obr. 7. Schéma zapojenia hlavnej dosky

16. Senzory

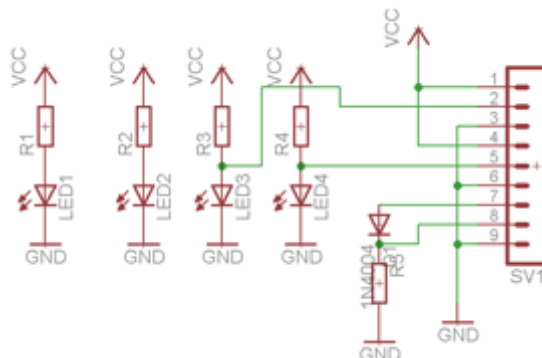
Senzory fungujú na princípe odrazu svetla od pevnej prekážky. Zdroj svetla, v našom prípade LED vysiela svetlo priamo pred seba. Ak pred snímačom nič nie je, svetlo sa rozplynie v priestore. V prípade, že sa pred snímačom nachádza prekážka, svetlo sa od nej odrazí a zachytáva ho svetlocitlivá súčiastka, v našom prípade fototranzistor. Ten sa otvára priamoúmerne tomu, koľko svetla naň dopadá. Aby sme však mohli riadiť citlivosť snímača je napätie z deliča, ktorý je tvorený rezistorom a fototranzistorom, privedené na komparátor.

Na druhý vstup komparátora je privedené napätie z potenciometra. To znamená, že potenciometrom si určujeme úroveň, kedy je snímač aktívny. Túto úroveň však treba voliť veľmi citlivo, pretože snímač môže reagovať aj na osvetlenie miestnosti, čo je v tejto chvíli nežiaduce. Veľkou nevýhodou tohto druhu snímača je, že nereaguje na predmety čiernej farby.

Signál z komparátorov je ďalej spracovávaný mikrokontrolérom ATtiny26. Pri komunikácii využívame generovanie prerušovacieho impulzu, ktorý je privedený na vstup INTO hlavného mikrokontroléra a následne hlavnému mikrokontroléru posielame kód aktívneho snímača. Týmto si šetríme strojový čas hlavného mikrokontroléra.



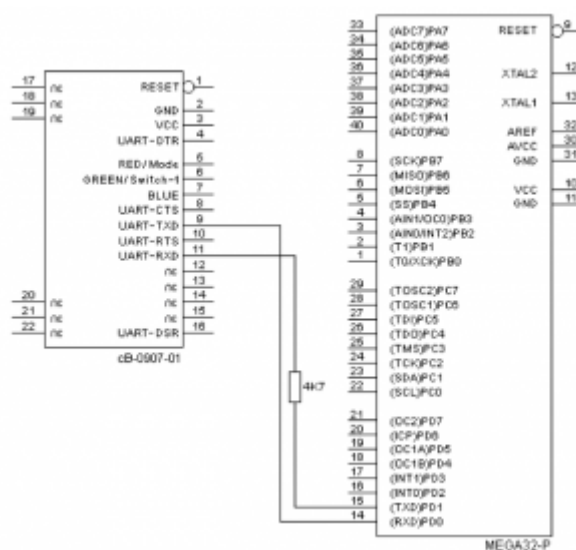
Obr. 8. Schéma zapojenia predných snímačov



Obr. 9. Schéma zapojenia zadných snímačov

17. Pripojenie bluetooth modulu

Na ovládanie robota sme sa rozhodli využiť modul od spoločnosti ConnectBlue cB-0907-01. Podrobnejší popis tohto modulu je uvedený vyššie. Mikrokontrolér ATmega32 obsahuje priamo na čipe jednotku USART, ktorá dovoľuje jednoduché prepojenie modulu bez potreby ďalších obvodov. Táto jednotka však nepodporuje hardwarové riadenie toku, takže toto riadenie nesmie obsahovať ani aplikácia v mobilnom telefóne. Rezistor s hodnotou 4K7 je do obvodu zapojený z dôvodu napätového nesúladu mikrokontroléra a bluetooth modulu. Mikrokontrolér pracuje s TTL úrovňami, zatiaľ čo modul pracuje s úrovňami 3,3V logiky. Pokiaľ modul vysiela informácie do mikrokontroléra, tento nesúlad neprekáža. Ak je to naopak, tento rezistor robí prúdové obmedzenie a chráni modul pred poškodením.



Obr. 10. Pripojenie bluetooth modulu k mikrokontroléru

18. Popis hlavného programu

Hlavný program môžeme rozdeliť na tri základné bloky. Prvý blok zabezpečuje komunikáciu prostredníctvom bluetooth modulu s aplikáciou v mobilnom telefóne. Ďalší blok zabezpečuje komunikáciu s doskou snímačov a vyhodnotenie signálu z nej. Tretí blok zabezpečuje generovanie impulzov pre budiče motorov.



Obr. 11. Vázby medzi jednotlivými blokmi programu

Prvý blok má za úlohu prijať povel z aplikácie z mobilného telefónu, vyhodnotiť, či povel nebol pri prenose poškodený a posunúť ho bloku, ktorý sa stará o pohyb robota na ďalšie „spracovanie“. Ďalším kritériom je, že vyhodnotiť správnosť rámca musí čo možno najrýchlejšie, to znamená, že sme ochotní „obetovať“ 300µs zo strojového času.

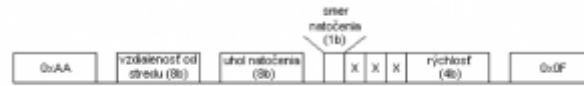
Ako sme už spomínali o prepojenie na hardwarovej úrovni sa nám stará jednotka USART. Táto jednotka má možnosť generovať tri vektory prerušení. Prvý vektor po odoslaní dát (TxC), druhý po prijatí dát (RxC) a tretí pri prázdnom dátovom registri jednotky (DRE). My v tejto chvíli využívame prvé dva z uvedených vektorov. Vieme, že z aplikácie v mobilnom telefóne nám príde príkaz vo formáte 0xAA potom 3 dátové bajty a potom 0x0F. Tento reťazec sa opakuje štyrikrát. Nám však stačí, aby bol trikrát rovnaký. Potom môžeme potvrdiť príjem reťazcom AOK00000F a poslať príkaz bloku motorov. Pokiaľ by nám vyššie uvedené požiadavky prijaté dáta nespĺňali, vysielame reťazec AER00000F, čím iniciujeme opakovanie prenosu.

Pri prijímaní sa spoliehame na vektor RxC. Pri každom prijatí bajtu je vyvolaný a my si aktuálny bajt uložíme do prijímacieho buffra. Po prijatí 17 bajtov vyhodnotíme zhodu troch päť bajtových reťazcov. V prvom rade si zistíme polohu prvého začiatku reťazca, to je prvú hodnotu 0xAA. V najhoršom prípade môže byť táto hodnota na pozícii číslo sedem. My máme však maximálnu možnú pozíciu nastavenú na číslo jedenásť. Je to z dôvodu rozsynchronizovania sa prijímacieho buffra s mobilnou aplikáciou. Z tohto dôvodu taktiež robíme zoradenie v prijímacom buffri ešte pred samotným vyhodnocovaním.

Vyberieme časť pred prvým začiatkom príkazu a všetko ostatné v buffri poposúvame na začiatok. Potom to čo sme vybrali vložíme na koniec buffra. Teraz môžeme pristúpiť k samotnému porovnávaniu, keďže vieme že dáta nám idú pekne po poradí. Porovnáваме dvojice posunuté vždy o päť miest napríklad pokiaľ je bajt na pozícii 2, porovnáme ho s bajtom na pozícii 7 a 12. To znamená, že po dvoch porovnaníach musí byť v premennej, ktorá nám počíta počet zhôd číslo 10. Ak je môžeme potvrdiť prenos. Ak nie je žiadame o opakovanie prenosu.

Odosielanie je realizované tým, že naplníme odosielací buffer a povolíme obsluhu vektora TxC. Po každom odoslaní bajtu sa nám posunie ukazovateľ na nasledujúci. Postupne odošleme všetky a po odoslaní posledného zakážeme obsluhu vektora TxC. Toto platí pre akékoľvek odosielanie z robota smerom do mobilného telefónu.

Tvar príkazu je 0xAA, vzdialenosť od stredu súradnej sústavy (8b), uhol natočenia 0 – 180° (8b), natočenie vľavo alebo vpravo (1b – najvyšší), v tom istom bajte ako smer natočenia idem aj rýchlosť (spodné 4b) a ukončovací znak 0x0F.



Obr. 12. Formát jedného príkazu

Po úspešnom prijatí príkazu, tento pokračuje do bloku, ktorý zabezpečuje pohyb robota. Robot je kalibrovaný tak, aby maximálna vzdialenosť, ktorú je schopný prejsť na jeden príkaz bola približne jeden meter. Aby sme vedeli, koľko krokov musia spraviť motory kým prejdú túto vzdialenosť, musíme vzdialenosť vynásobiť konštantou 8. Pri uhle natočenia musíme konštantu z príkazu, ktorá hovorí o uhle vynásobiť číslom 1,3.

Robot sa najprv natočí do potrebného uhla a až potom ide dopredu. Po natočení pošle správu AUHOKXXXFY, pričom za XXX je dosadená hodnota uhlu v ASCII znakoch, do ktorej sa robot natočil a za Y či to natočenie bolo vľavo alebo vpravo. Pri pohybe rovno robot posíla ubehnutú vzdialenosť vo formáte APOLKXXXF, kde za XXX dosadzuje ubehnutú vzdialenosť v ASCII znakoch. Ubehnutú vzdialenosť posíla v dvojsekundových intervaloch.

Komunikácia so snímačmi je realizovaná pomocou vonkajšieho prerušenia s číslom 0 (INT0). Prerušenie je vyvolané nábežnou hranou impulzu na pine PD2 (číslo 16). Po vyvolaní prerušenia sa na pinoch PD3 (17) a PD4 (18) objaví kód aktivovaného snímača. Informácia o aktivovanom snímači sa ďalej prenáša pomocou reťazca AXXXX0000F. Podľa toho ktorý snímač bol aktivovaný, je za jedno z X dosadená jednotka a za ostatné je dosadená 0. Vždy sa prenáša informácia len o jednom aktívnom snímači. Robot len odosiela stav snímačov, pohybom na ne nereaguje.

19. Popis programu snímačov

Program dopytovaním kontroluje stav snímačov. Pokiaľ bol niektorý z nich aktivovaný, posíla do hlavného mikrokontroléra najprv žiadosť o komunikáciu v podobe impulzu na prerušovacom pine INT0. Následne posíla kód aktivovaného snímača. Kód 00 prislúcha ľavému prednému snímaču, 01 pravému prednému, 10 je ľavý zadný snímač a 11 pravý zadný. V jednom momente môže byť aktívny len jeden snímač. Je však veľmi nepravdepodobné, aby sa v jednom momente naraz aktivovalo viac snímačov.

20. Záver

Cieľom nášho projektu bolo vytvoriť grafické rozhranie v mobilnom zariadení a prispôbiť ho ovládaniu pomocou dotykového display-a. Tiež sme museli vyvinúť komunikáciu mikrokontroléra, ktorý ovláda robota s mobilným zariadením. Popri konštruovaní nášho projektu sme urobili aj testy prenosu prostredníctvom bluetooth. Výsledky týchto testov sú 99,99% pri prenose z počítača do mobilného zariadenia. Objem prenášaných dát pri teste bol 776918B. Prenos z počítača do robota mal úspešnosť 99,96%. Objem prenášaných dát pri teste bol 1152082B. Z uvedených údajov vyplýva, že rozhranie bluetooth je dostatočne spoľahlivé na využitie v priemyselných aplikáciách a v robotike.

Odkazy na literatúru

1. Bruce Hopkins a Ranjith Antony. 2003. Bluetooth for Java: Apress, 2003. ISBN:

1590590783, 352 strán

2. Dee Bakker, Diane McMichael Gilster, Ron Gilster.2002.NewYork,NY: Bluetooth End to End, Hungry Minds Inc., 2002, ISBN:0-7645-4887-5, 309 strán
3. Bluetooth, [online], Dostupné na internete:
<http://www.bluetooth.com>
4. NetBeans Open Source, [online], Dostupné na internete:
<http://netbeans.org>
5. Oracle , [online],Dostupné na internete:
www.oracle.com
6. Sun ,[online], Dostupné na internete:
developers.sun.com
7. Katalógové údaje k modulu cB-0907-01, Dostupné na internete:
https://www.spezial.com/doc/conblu/em_ds_oemspa_310.pdf
8. Kataógové údaje k mikrokontroléru ATmega32, Dostupné na internete:
http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf
9. Kataógové údaje k mikrokontroléru ATtiny26, Dostupné na internete:
http://www.atmel.com/dyn/resources/prod_documents/doc1477.pdf

Spoluautorom článku sú Róber Spielmann a Ing. Michal Blaho , Ústav riadenia a priemyselnej informatiky, Fakulta elektrotechniky a informatiky STU

Práca bola prezentovaná na Študentskej vedeckej a odbornej činnosti (ŠVOČ 2011) v sekcii Kybernetika a robotika a získala Diplom dekana, ISBN 978-80-227-3508-7
