

ELEKTROTEHNIČKI FAKULTET
UNIVERZITET U BEOGRADU

Aćimović Žarko 2012/3061

Kontroler (sa potencimetrima) za električno vozilo sa ZigBee komunikacijom

32-bitni mikrokontroleri i primena

mentor:

red. prof. dr Dragan M. Vasiljević

Beograd, jun 2013.

Sažetak

U ovom izveštaju objašnjen je postupak realizacije kontrolera za električno vozilo, gde se bežična komunikacija između vozila i kontrolera vrši pomoću dva *XBeePro* modula. Upravljanje brzinom i pravcem se obavlja preko dva potenciometra. Cilj projekta je upoznavanje sa arhitekturom *ARM Cortex M3* procesora i IAR razvojnim okruženjem. Ovaj projekat je rađen iz predmeta 32-bitni mikrokontroleri i primena, na master studijama odseka za elektroniku.

Ključne reči: *STM32F100RBT6B*, potenciometri, tajmeri, *ADC*, *DMA*, *UART*, *ZigBee*, *XBee Pro*.

Sadržaj

1. Uvod	5
1.1 Projektni zadatak	5
2. Hardver	6
2.1 <i>STM32F100RBT6B</i>	6
2.2 XBeePro	8
2.3 USB-to-UART komunikacija	10
2.4 Šema i način povezivanja hardvera	11
3. Softver	13
3.1 Protokl komunikacije	13
3.2 Algoritam programa	14
4. Zaključak	17
5. Literatura	18

Slike

Slika 1.1 Blok dijagram	5
Slika 2.1 Evalucionna pločica <i>STM32VLDISCOVERY</i> sa označenim korišćenim pinovima.....	7
Slika 2.2 <i>XBeePro S1</i> modul.....	9
Slika 2.3 Uparivanje <i>XBeePro</i> modula u <i>X-CTU</i> softveru	9
Slika 2.4 <i>HTerm</i> terminal.....	11
Slika 2.5 Šematski prikaz povezivanja hardvera	12
Slika 3.1 Algoritam glavnog programa.....	14
Slika 3.2 Algoritam pekidne rutine tastera	15
Slika 3.3 Algoritam <i>TIM2</i> prekidne rutine.....	16

Tabele

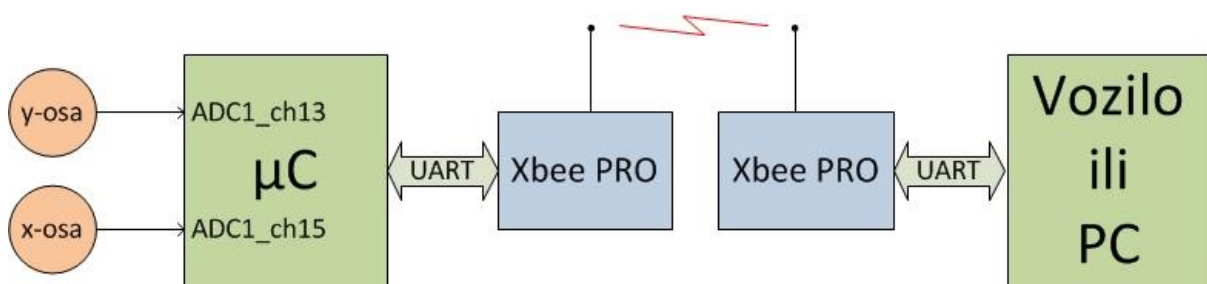
Tabela 2.1 Raspored <i>XBeePro S1</i> pinova	10
Tabela 3.1 Poruka o svetlima	14
Tabela 3.2 Izgled cele poruke.....	14

1. Uvod

Ideja za izradu ovog projekta jeste bežični daljinski kontroler za upravljanje električnim vozilom (sa step motorima). Brzinu i pravac vozila određuju dva potenciometra, a komunikacija sa vozilom se vrši preko *XBeePro* modula. Ovaj izveštaj sadrži opis realizacije projekta, odabir elemenata, objašnjenje softvera i dobijene rezultate.

1.1 Projektni zadatak

Projektni zadatak je da se realizuje bežična daljinska kontrola električnog vozila, koristeći razvojnu pločicu *STM32VLDISCOVERY* i *ZigBee* komunikaciju između vozila i kontrolera. Brzinu i pravac vozila zadaju dva potenciometra koji su povezani na dva kanala *AD* konvertora. Podaci dobijeni sa *AD* konvertora se obrađuju u mikrokontroleru, generiše se komanda, po već unapred određenom protokolu, koji se šalje na *UART* mikrokontrolera. *UART* komandu prosleđuje *XBeePro* modulu, koji je dalje prosleđuje vozilu (ili personalnom računaru). Komanda sadrži informaciju o trenutnom položaju potenciometara. Na slici 1.1 dat je blok dijagram ovog sistema.



Slika 1.1 Blok dijagram

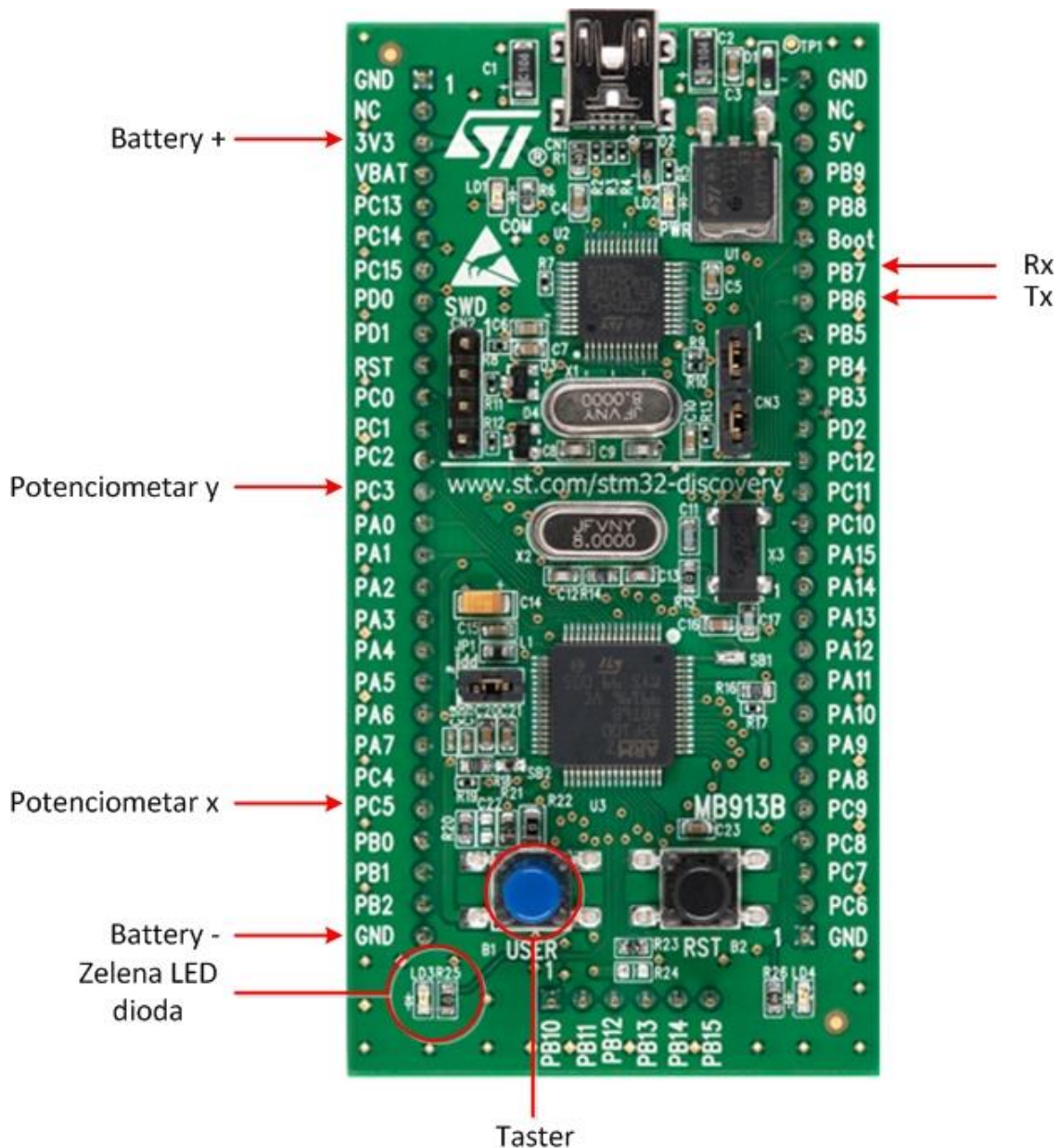
2. Hardver

U ovom poglavlju opisan je hardver korišćen za realizaciju ovog projekta. Takođe biće prikazan i šematski prikaz povezivanja hardvera.

2.1 STM32F100RBT6B

Upravljačka jedinica koja je korišćena za realizaciju ovog projekta je 32-bitni mikrokontroler *STM32F100RBT6B* proizvođača *ST Microelectronics*, koji u svojoj osnovi ima *ARM CortexTM M3* procesor. Tačnije korišćena je razvojna ploča *STM32VLDISCOVERY* istog proizvođača. Ovaj procesor je razvijen za *embedded* aplikacije koje zahtevaju rad u realnom vremenu uz postizanje visokih performansi, male potrošnje i niske cene. Ova familija koristi *Thumb-2* instrukcijski set koji pruža bolje performanse i veću gustinu koda. Sama evalucionna pločica sadrži sve što je neophodno i početnicima i iskusnim korisnicima da bi što pre otpočeli sa radom. Pored samog mikrokontrolera na pločici se nalazi i hardverski dodatak *ST-Link debugger/programmer* koji služi da olakša programiranje i debugovanje. Programiranje i debugovanje se vrši preko *USB* kabla. Na pločici se nalaze dve diode i dva tastera, od kojih je jedan taster za *reset* a drugi je korisnički. Sam mikrokontroler je u *SMD* tehnologiji, ali na pločici su izvedeni *through-hole* priključci za skoro sve pinove mikrokontrolera i time je znatno olakšano priključivanje dodatnog hardvera na mikrokontroler.

Na slici 2.1 data je razvojna pločica *STM32VLDISCOVERY* sa označenim korišćenim pinovima za pomenutu realizaciju.



Slika 2.1 Evalucionna pločica *STM32VLDISCOVERY* sa označenim korišćenim pinovima

Između pinova *GND* i *3V3* se vezuje baterija za napajanje. Taster *USER* služi da dozvoli/zabrani slanje preko *UART*-a. Zelena *LED* dioda služi za indicaciju da li je dozvoljeno ili zabranjeno slanje preko *UART*-a. Na pinove *PC3* i *PC5* vezuju se izlazi dva potencimetra koji služe za zadavanje pozicije i brzine. Pinovi *PB7* i *PB6* se koriste za *Rx* i *Tx* *UART*-a i vezuju se za pinove *ZigBee* modula.

2.2 XBeePro

ZigBee protokol je razvijen od strane *ZigBee Alliance* (udruženje većeg broja kompanija iz oblasti elektronike), i nadovezuje se na *IEEE 802.15.4* bežični protokol. Ovaj standard služi za kratkodometnu bežičnu komunikaciju, i ističe se niskom odašiljačkom snagom i malom potrošnjom. Standard određuje nominalnu izlaznu snagu od $-3dBm$ ($0,5mW$). Domet uređaja može da varira od 10 do 100m, u zavisnosti od antene, prostora, okoline... Maksimalna brzina prenosa podataka u *ZigBee* mrežama je $250 kb/s$ (na $2,4GHz$), pa se koristi za slanje i prijem kratkih komandi i prikupljanje podataka iz senzora. Sve ove karakteristike čine ovu tehnologiju idealnu za potrebe jeftinih i energetski efikasnih uređaja. Standard je našao svoju primenu u aplikacijama za industrijsku kontrolu, pametne kuće, u medicini, pametno merenje i u ostalim aplikacijama u kojima protok podataka nije toliko veliki ali je bitna mala potrošnja energije.

U ovom projektu koristi se *ZigBee* modul proizvođača *Digi International* koji se zove *XBeePro S1*, i prikazan je na slici 2.2.

Za podešavanje i konfiguraciju *ZigBee* modula koristi se *X-CTU* softver, firme *Digi International*. Konfiguracija se vrši na PC računaru. Za povezivanje *ZigBee* modula na PC računar koristi se *USB-to-UART* pločica opisana u sledećem poglavlju. U ovom slučaju se koristi *Peer-To-Peer* mreža. Podešavanje modula se svodi na upis izvorišne adrese jednog modula u registar odredišne adrese drugog modula i obrnuto. Ovim postupkom obavljeno je uparivanje modula, slika 2.3. Oba modula su podešena na isti način za serijsku komunikaciju i to:

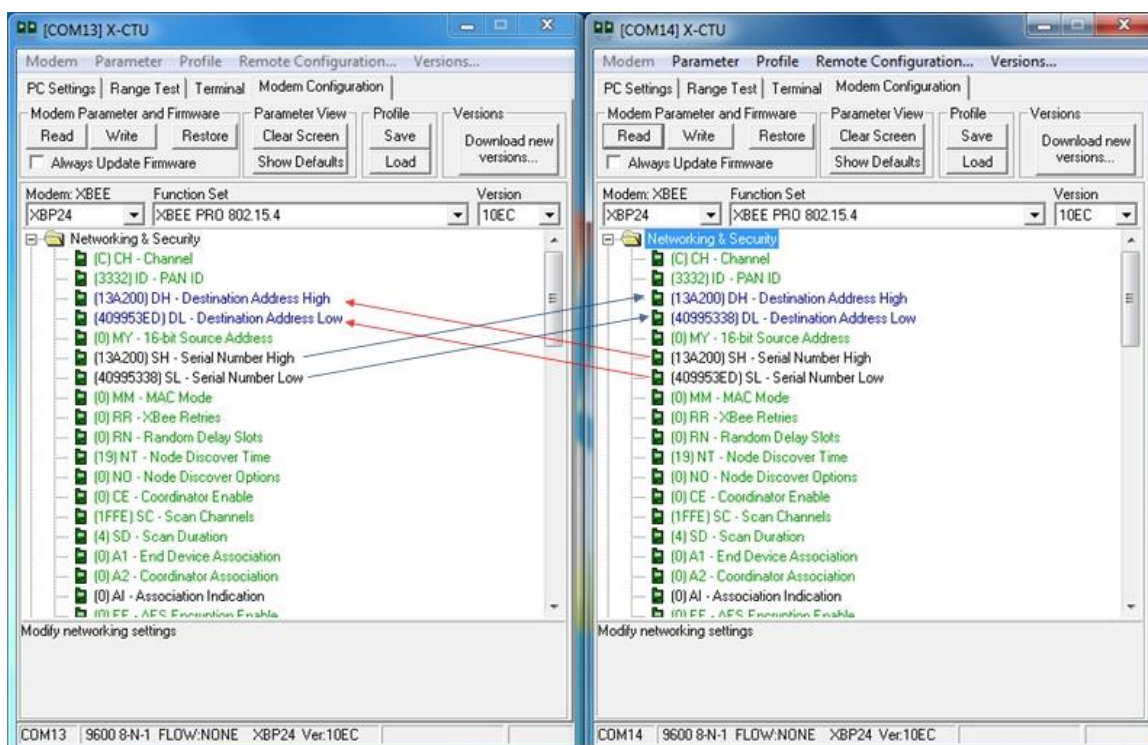
1. *Baud rate*: 9600 baud/sec
2. *Flow Control*: off
3. *Data bits*: 8
4. *Parity bit*: none
5. *Stop bit*: 1

Naravno ova podešavanja moraju biti podešena i na *UART*-u koji služi za komunikaciju sa *ZigBee* modulom.

Pinovi potrebni za povezivanje *XBeePro S1* modula dati su tabelom 2.1. Potrebno je napomenuti da je za konfiguraciju modula neophodno povezati sledeće pinove: *VCC*, *GND*, *DIN*, *DOUT*, *RESET*, *RTS* i *DTR*, dok su za komunikaciju dovoljni samo *VCC*, *GND*, *RESET*, *DIN* i *DOUT*.



Slika 2.2 *XBeePro S1* modul



Slika 2.3 Uparivanje *XBeePro* modula u X-CTU softveru

<i>Pin Number</i>	<i>Function</i>
1.	VCC
2.	DOUT
3.	DIN
5.	\overline{RESET}
9.	DTR
10.	GND
16.	RTS

Tabela 2.1 Raspored *XBeePro S1* pinova

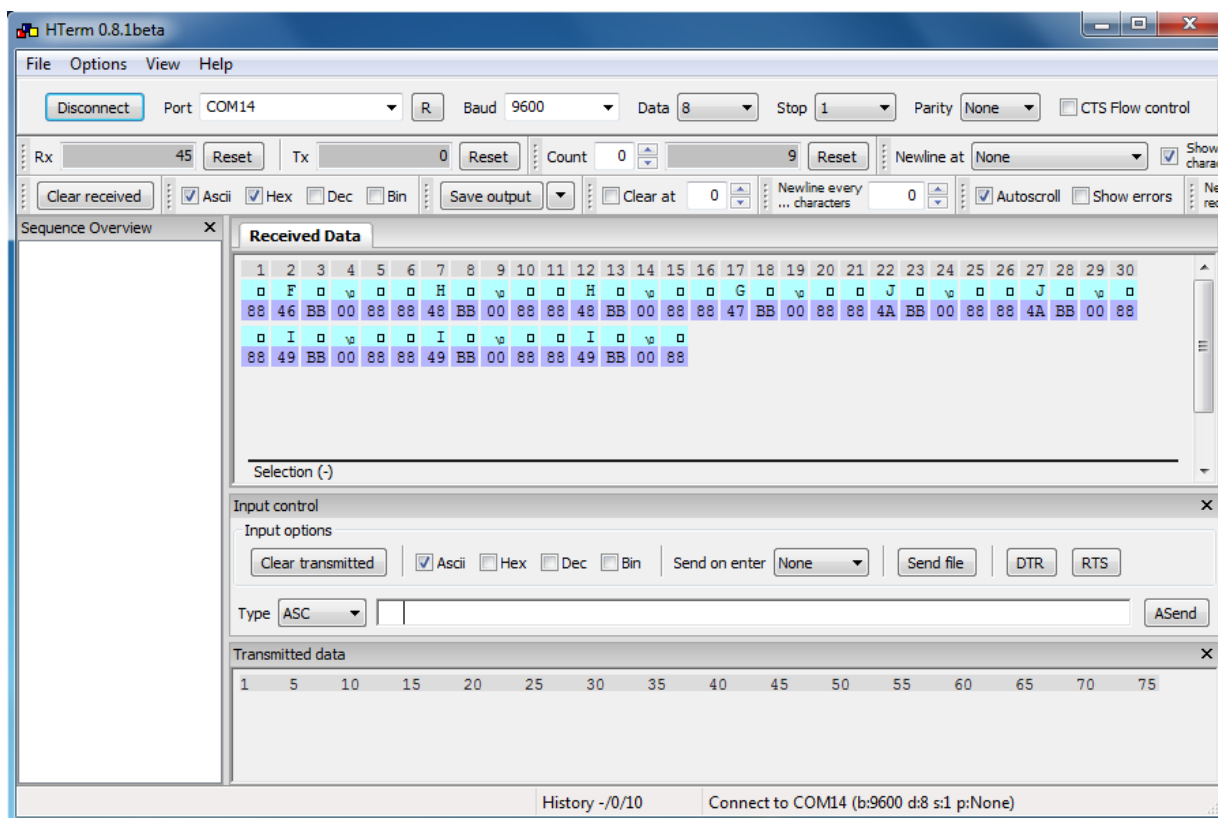
2.3 USB-to-UART komunikacija

U svrhe testiranja projekta, bilo je potrebno slati izgenerisanu poruku na računar. Takođe, bilo je potrebno konfigurisati i upariti *XBeePro* module. Za ove potrebe bilo je potrebno omogućiti vezu *ZigBee* modula i PC računara. Kako *ZigBee* modul komunicira preko *UART*-a, bilo je neophodno da se omogući *UART* komunikacija sa računarom. PC računari ne poseduju *UART* komunikaciju, ali svi poseduju *USB* serijsku komunikaciju. Jedno od mogućih rešenja jeste konverzija *USB* na *UART* i obrnuto.

Ovu konverziju odlično obavlja integrisani čip *FT232RL* proizvođača *Future Technology Devices International Ltd. FTDI*. Pored standardnih pinova ovaj čip poseduje još pinova sa dodatnim specijalnim funkcijama. Takođe, ovaj čip ima već prilagođene naponske nivoe za komunikaciju, nisu mu potrebni invertori, niti pomerači nivoa. Ima mogućnost biranja naponskog nivoa od 3.3V ili 5V. Džamperom se bira odgovarajući željeni naponski nivo.

Kao što je već napomenuto za konfiguraciju *ZigBee* modula, potrebno je povezati sve pinove iz tabele 2.1.

Terminal-program koji je korišćen za prikaz podataka poslatih putem *ZigBee* mreže, zove se *HTerm*, i njegov izgled dat je na slici 2.4.



Slika 2.4 *HTerm* terminal

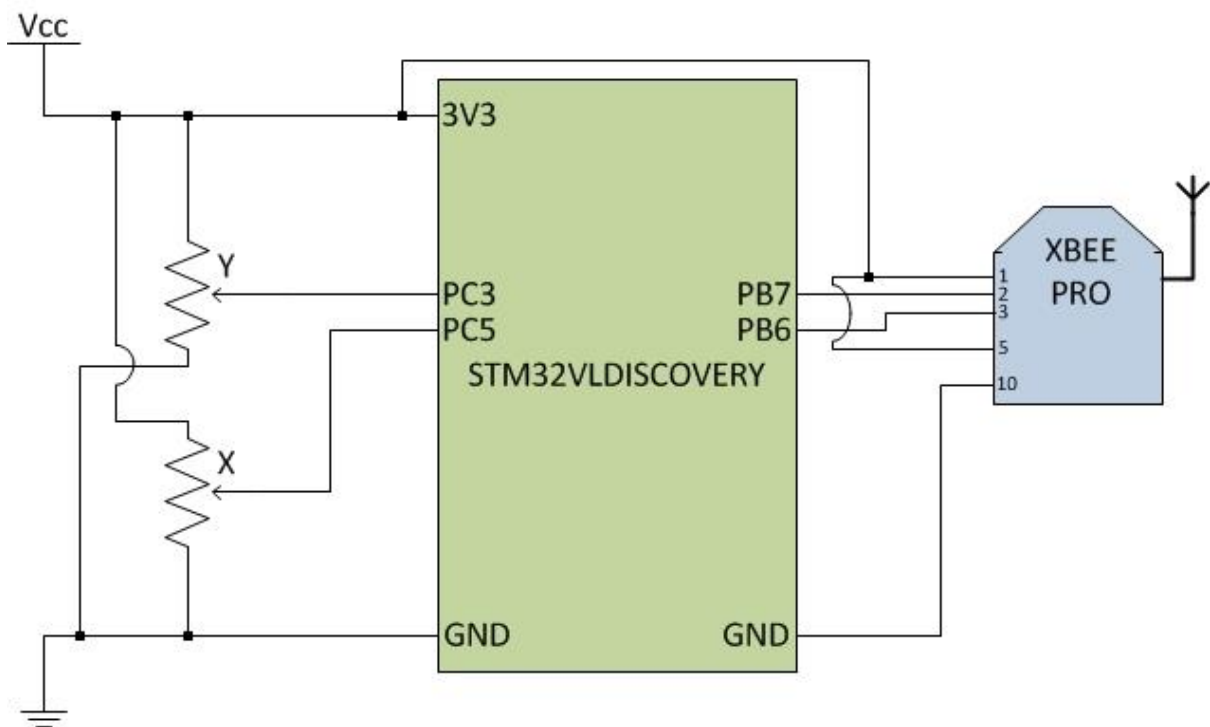
2.4 Šema i način povezivanja hardvera

Kao što je već bilo reči, poziciju električnom vozilu zadaju dva potenciometra. Korišćena su dva potenciometra od 50 kΩ. Potenciometri se napajaju sa baterije, a njihovi izlazi su povezani za portove mikrokontrolera (*PC3* i *PC5*) na kojima se nalaze kanali *AD* konvertora.

XBeePro modul je povezan preko portova *PB6* i *PB7* (*Tx* i *Rx UART-a*). Značenje pinova *XBeePro* modula dato je tabelom 2.1. Pin 5. *XBeePro* modula je *RESET* i aktivan je na logičkoj nuli. Da bi *XBeePro* modul ispravno radio potrebno je njegov pin 5. povezati na *Vcc*.

Bežični daljinski kontroler se napaja sa baterije od 3,7V 1300mAh. S obzirom da je maksimalna potrošnja ovog sistema vrlo mala, maksimalno 100mA, ova baterija je više nego dovoljna da napaja ovaj sistem nekoliko sati, garantovano 6 sati.

Način na koji je hardver u ovom sistemu povezan dat je slikom 2.5.



Slika 2.5 Šematski prikaz povezivanja hardvera

3. Softver

U ovom poglavlju biće objašnjen algoritam glavnog programa kao i prekidnih rutina. Pored toga biće opisan i protokol kojim se podaci šalju sa upravljača na vozilo.

3.1 Protokl komunikacije

U ovom poglavlju će biti opisan oblik poruke koju bežični daljinski kontroler treba da generiše da bi uspešno mogao da zada komandu električnom vozilu. Iz razloga što *UART* može da prihvati samo jedan *byte*, koji nije dovoljan za smeštanje celog podatka potrebnog za upravljanje, poruka se sastoji od više *byte*-ova, tačnije 5 *byte*-ova čini poruku. Takođe da bi mikrokontroler, koji se nalazi na vozilu, mogao da razlikuje početak i kraj poruke bilo je potrebno uvesti *byte* *MARKER*, koji označava početak i kraj poruke.

Pored markera, poruka mora da sadrži i informaciju o položaju potencimetara *Y* i *X*. Potencijometar *Y* će imati funkciju zadavanja brzine električnom vozilu, dok potencijometar *X* zadaje pravac električnog vozila. Način tumačenja ovih veličina i realizacije kretanja električnog vozila je objašnjen u izveštaju za električno vozilo. Za svaki od ova dva parametra potreban je po *byte*. Vrednost ova dva parametra može biti od -100 do 100 (*0x9C* do *0x64* heksdecimalno).

Poslednji deo poruke je informacija o svetlima. Značenja bita u *byte*-u svetlo dato je tabelom 3.1. Njegova maksimalna vrednost može biti *0x55*. Vrednost *byte*-a *MARKER* je postavljena na *0x88*, jer se ta vrednost nikada neće pojaviti na mestu *byte*-a *Y*, *X* i *byte*-a svetlo.

Dakle prvi *byte* poruke predstavljaće *MARKER*, drugi *byte* je *Y* podatak, treći je *X* podatak, četvrti je podatak o svetlima, i na kraju opet sledi *byte* *MARKER*. Format poruke dat je tabelom 3.2.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Lights	0	Hi/Low	0	MigL	0	MigR

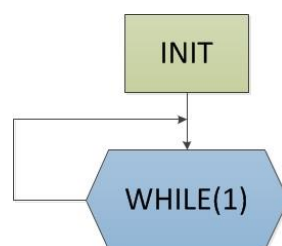
Tabela 3.1 Poruka o svetlima

Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
MARKER(0x88)	Y	X	Svetla	MARKER(0x88)

Tabela 3.2 Izgled cele poruke

3.2 Algoritam programa

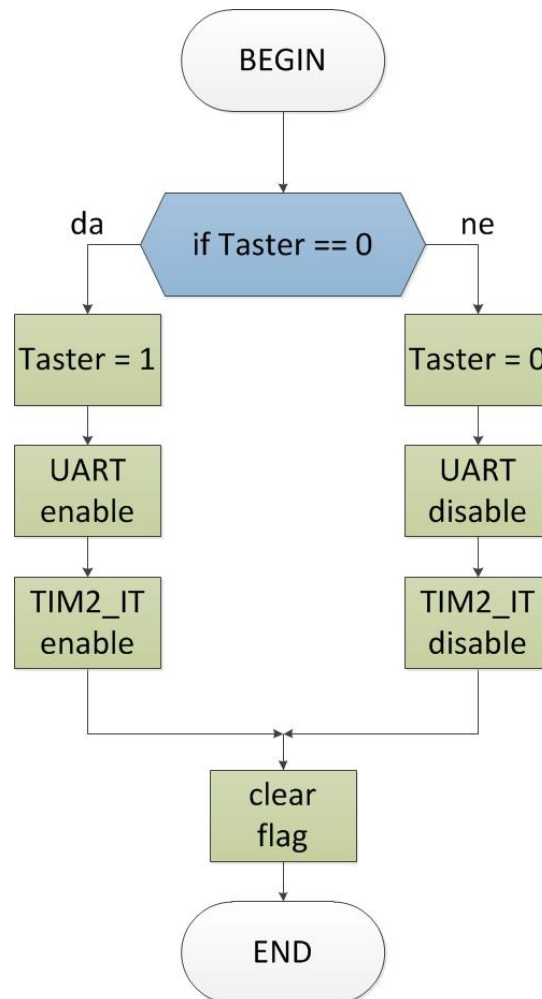
U ovom poglavlju objašnjena su algoritamska rešenja glavnog programa i svih prekidnih rutina. U glavnom programu izvršena je neophoda inicijalizacija korišćenih periferija i portova, dovođenje klock signala periferijama i dozvola generisanja prkida određenim periferijama. Korišćene periferije su *UART*, *DMA*, *ADC* i *TIM2*. Nakon toga glavni program ulazi u jednu *while* petlju u kojoj čeka na prekid. Algoritam glavnog programa dat je a slici 3.1.



Slika 3.1 Algoritam glavnog programa

Prekidna rutina koja obavlja očitavanje tastra je *EXTIO_IRQHandler*. Prekidna rutina *EXTIO_IRQHandler* služi da se pritiskom na taster *USER* na pločici omogući, odnosno onemogući prekid tajmera *TIM2* i slanje *UART*-om. Zelena *LED* dioda služi za indikaciju dozvole i zabrane prekida i slanja. Na kraju

izvršavanja prekidne rutine potrebno je resetovati fleg prekida. Algoritam ove prekidne rutine dat je slikom 3.2.

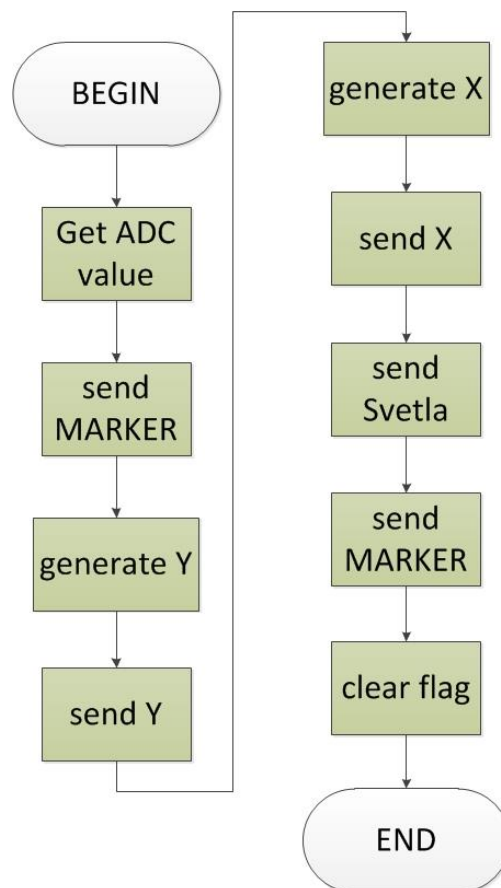


Slika 3.2 Algoritam prekidne rutine tastera

Celokupna obrada, generisanje poruke i njeno slanje obavlja se unutar prekidne rutine tajmera *TIM2*. Naziv prekidne rutine je *TIM2_IRQHandler*. Ova prekidna rutina se poziva periodično sa prekoračenjem tajmera i njegovm resetovanjem. Prekid se dešava sa učestanošću od 20Hz-a, jer se smatra da je 20Hz-a dovoljno da isprati svaki pokret prosečnog čoveka, a i zbog ograničenja motora u električnom vozilu. Algoritam ove prekidne rutine dat je na slici 3.3.

Po ulasku u prekidnu rutinu startuje se *AD* konverzija. Skeniraju se dva kanala *AD* konvertora, i uz pomoć *DMA* periferije, dobijene vrednosti se smeštaju u memoriju. Iz razloga što se cela poruka sastoji iz pet delova potrebno je voditi računa koji se deo poruke trenutno šalje. Prvo se šalje

MARKER, koji označava početak poruke. Zatim se, na osnovu vrednosti sa *AD* konvertora, generiše *Y* podatak i vrši se njegovo slanje. Nakon toga sledi generisanje i slanje *X* podatka. Podatak *Svetla* se šalje posle *X* podatka. I na kraju potrebno je opet poslati *MARKER* koji će označiti kraj poruke. Posle svakog slanja potrebno je proveriti kada će *UART* postati slobodan, da ne bi doslo do greške prilikom slanja. Potrebno je sačekati da fleg *USART_FLAG_TXE* postane jedinica, u suprotnom može doći do preskakanja nekog od delova poruke. Ukoliko prekidna rutina *EXTIO_IRQHandler* onemogući slanje *UART*-om, prekidna rutina *TIM2_IRQHandler* šalje poruku koja zaustavlja vozilo. Na kraju izvršavanja prekidne rutine potrebno je resetovati fleg prekida, kako bi se omogućio ponovni ulaz u ovu prekidnu rutinu po isteku tajmera *TIM2*.



Slika 3.3 Algoritam *TIM2* prekidne rutine

4. Zaključak

Projekat iz predmeta *32-bitni mikrokontroleri i primena* obuhvata dizajn hardvera i softvera, i negov cilj je upoznavanje sa arhitekturom *ARM Cortex M3* procesora i IAR razvojnim okruženjem. Zadatak projekta je dizajn bežičnog daljinskog upavljača za električno vozilo.

Testirana je ispravnost i funkcionalnost uređaja slanjem na PC računar kao i slanjem na električno vozilo. Rezultati testiranja su bili očekivani. Moguće su manje korekcije softvera, koje bi povećale sigurnost i stabilnost sistema.

S obzirom da je upotrebljeni mikrokontroler nedovoljno iskorišćen u ovom projektu, jer su njegove performanse veoma visoke, prilikom izrade projekta uzeta je u obzir mogućnost nadogradnje sistema, npr. dodavanjem akcelerometra kojim bi se omogućilo upravljanje vozilom samim pokretom bežičnog daljinskog kontrolera.

5. Literatura

- [1] Predavanja i vežbe iz predmeta 32-bitni mikrokontroleri I primena, Elektrotehnički fakultet, Beograd
- [2] STMicroelectronics: "STM32F10xxx Cortex-M3 Datasheet"
- [3] STMicroelectronics: "STM32F10xxx Cortex-M3 Programming manual"
- [4] STMicroelectronics: "STM32F10xxx Cortex-M3 Reference manual"
- [5] STMicroelectronics: "STM32VLDISCOVERY User manual"
- [6] Digi International: "XBee & XBeePro Product manual"
- [7] ZigBee Alliance, www.zigbee.org