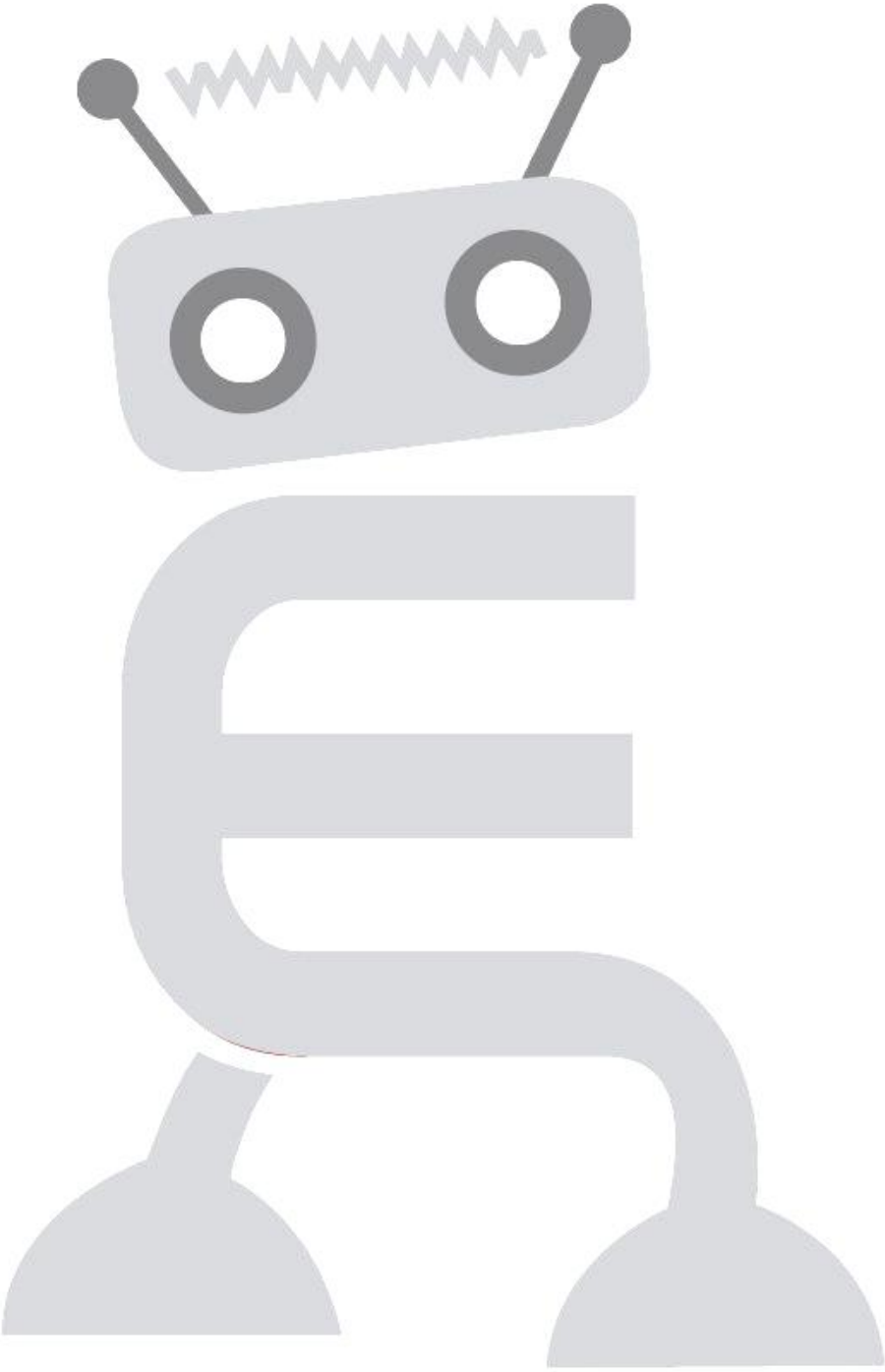


OGRANIČENO JAMSTVO PROIZVOĐAČA

Pozor!

Ovo ograničeno jamstvo ne utječe na Vaša zakonska prava u sklopu državnih zakona koji se odnose na prodaju potrošačkih proizvoda.

Inovatic ICT Vam jamči da će za trajanja jamstva besplatno i u poslovno razumnom vremenu popraviti proizvod ili, ako je potrebno, zamijeniti ga u skladu s ovim ograničenim jamstvom (osim ako drugačije ne nalaže zakon).

JAMSTVO

Trajanje jamstva počinje od trenutka prvotne kupnje proizvoda prvog korisnika. Jamstvo traje 6 mjeseci. Međutim, jamstvo za popravljene i zamijenjene dijelove vrijedi kao za originalno trajanje jamstva proizvoda ili za šezdeset (60) dana od datuma popravka ili zamjene.

7-DNEVNI POVRAT NOVCA

Ako u roku od 7 dana od primitka proizvoda zaključite da ne odgovara Vašim potrebama, možete ga vratiti uz puni povrat novca. Inovatic ICT osigurava povrat kupovne cijene proizvoda, isključujući troškove dostave i manipulativne troškove. Ovo jamstvo ne vrijedi ako je proizvod izmijenjen ili oštećen. Ako proizvod niste kupili od tvrtke Inovatic ICT, ovo se pravo regulira prema općim uvjetima distributera kod kojeg ste kupili naš proizvod.

KONTAKT SERVISA

Ako želite prijaviti slučaj pod ovim ograničenim jamstvom, molimo vratite svoj proizvod u tvrtku Inovatic ICT d.o.o.. Možete nazvati +38513864320 ili kontaktirati s nama elektroničkom poštom na info@inovatic.hr za daljnje pojedinosti o tome kako napraviti zahtjev za jamstvo.

Kada radite zahtjev pod ovim ograničenim jamstvom, potrebno je imati proizvod i originalni dokaz o kupnji koji jasno pokazuje ime i adresu prodavača, vrijeme i mjesto kupnje, tip proizvoda. Pri povratu robe, molimo priložite: svoje ime, telefonski broj, adresu isporuke i opis problema. Inovatic ICT će vratiti Vaš proizvod ili ga zamijeniti istim načinom dostave koja se koristila pri povratku.

JAMSTO NE POKRIVA

1. Ovo ograničeno jamstvo ne pokriva a) oštećenja koja su nastala grubim rukovanjem (uključujući, bez ograničenja, oštećenja nastala oštrim predmetima, savijanjem, stiskanjem, padom ili naknadnim lemljenjem itd.), i/ili b) oštećenja i štetu nastalu pogrešnim korištenjem proizvoda, uključujući korištenje koje je u suprotnosti s uputama koje je izdala tvrtka Inovatic ICT d.o.o. (npr. kao što je napisano u korisničkim uputama za proizvod) i/ili c) ostale akcije koje su izvan razumne upotrebe proizvoda.
2. Ovo ograničeno jamstvo ne pokriva oštećenja ili navodna oštećenja prouzročena činjenicom da je proizvod bio korišten s bilo kojim drugi proizvodom ili je bio spojen na njega, dodatnom opremom i/ili uslugom koji nije izradio ili dostavio Inovatic ICT d.o.o. ili je korišten na drugi način osim onako kako je zamišljeno da se rabi.
3. Ovo ograničeno jamstvo ne pokriva oštećenja prouzročena napajanjem programiranog uređaja izravno preko programatora (USB-a na računalu – PWR kratkospojnik)
4. Ovo ograničeno jamstvo nije valjano ako je proizvod bio izložen vlazi ili ekstremnim termalnim uvjetima ili uvjetima okoliša ili brzim promjenama tih uvjeta, koroziji, oksidaciji, prosipanju hrane ili prolijevanju tekućine ili utjecaju kemijskih proizvoda.

NAPOMENA

Ako proizvod nije pokriven uvjetima ograničenog jamstva Inovatic ICT d.o.o. zadržava pravo naplate troškova rukovanja. Pri popravku ili zamjeni proizvoda tvrtka Inovatic ICT d.o.o. može se koristiti proizvodima ili dijelovima koji su novi, jednaki novima ili reparirani.

OGRANIČENJE ODGOVORNSTI

Tvrtka Inovatic ICT d.o.o. nije odgovorna za bilo kakvu štetu nastalu kršenjem bilo kojeg dijela jamstva, uključujući zdravlje i život. Korisnik preuzima punu odgovornost za sve projekte, u kojima se koristi proizvodima tvrtke Inovatic ICT, bez obzira na to koliko su oni opasni za život.

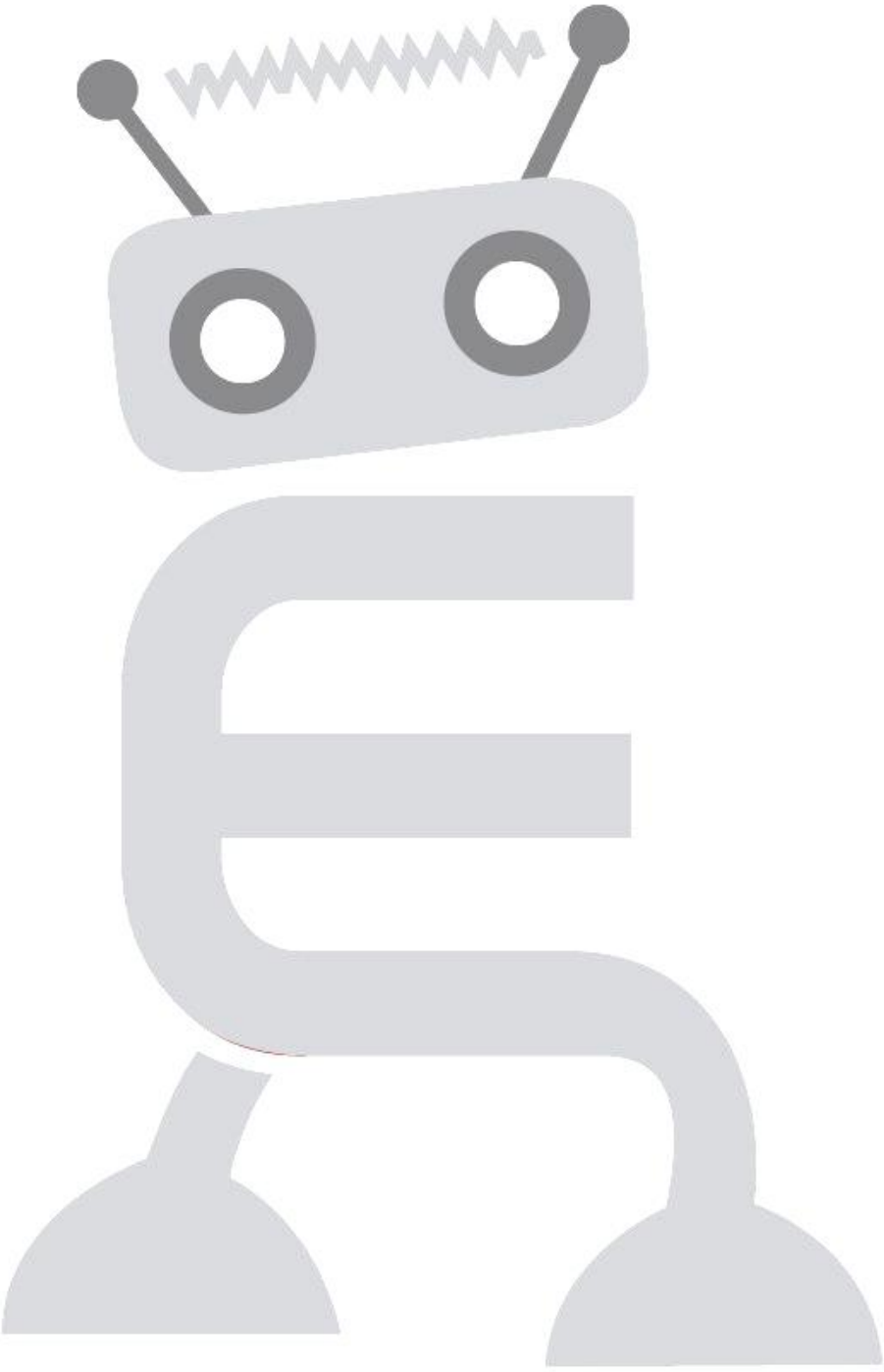
AUTORSKA PRAVA I ZAŠTITNI ZNAKOVI

Ova dokumentacija vlasništvo je tvrtke Inovatic ICT. Preuzimanjem ove dokumentacije ili softvera slažete se da se njome koristite isključivo s proizvodima tvrtke Inovatic ICT. Bilo koja druga uporaba nije dopuštena i može značiti povredu autorskih prava, što je kažnjivo prema zakonu o intelektualnom vlasništvu. Bilo kakvo umnoživanje ove dokumentacije za komercijalnu uporabu izričito je zabranjeno. Umnoživanje u edukativne svrhe je dopušteno bez dopuštenja tvrtke Inovatic ICT. Ovo pravo temelji se na sljedećim uvjetima: tekst, ili bilo koji njegov dio ne može se umnoživati u komercijalne svrhe, nego samo u obrazovne svrhe ako se koristi isključivo u kombinaciji s proizvodima Inovatic ICT. E-Bot, EMoRo, Inovatic ICT, Inovatic ICT logotip, E-Bot logotip, te EMoRo logotip zaštitni su znakovi tvrtke Inovatic ICT. Ako odlučite upotrebljavati zaštitne znakove tvrtke Inovatic ICT na web stranici ili u tiskanim materijalima, morate navesti da je "(zaštitni znak) je zaštitni znak tvrtke Inovatic ICT d.o.o.".

TISKARSKE POGREŠKE

Iako je uloženo mnogo truda kako bi se osigurala točnost naših tekstova, pogreške su moguće. Ako ste pronašli pogrešku, javite nam e-poštom na podrska@inovatic.hr. Nastojanjem za poboljšanjem svih naših obrazovnih materijala i dokumentacije zadržavamo pravo na izmjenu tekstova.

ISBN 2378456293874562398475623789562394 87



SADRŽAJ

1. UVOD 2

2. SASTAVLJANJE EMORO-A 6

2.1 Sastavljanje kotača 7

2.1.1 Kotač za pogonske servo motore 8

2.1.2 Kotač za elektromotorni prijenos 9

2.2 Sastavljanje prednjeg kotača 10

2.2.1 Svesmjerni kotač - Omniwheel 10

2.2.2 Svesmjerna kugla - BallCaster 12

2.3 Sastavljanje pogona 14

2.3.1 Servo motori 14

2.3.2 Elektromotorni pogon s prijenosom – low speed C 16

2.3.3 Elektromotorni pogon s prijenosom – high speed A 21

2.4 Sastavljanje šasije 27

2.4.1 Osnovni set 27

2.4.2 Radar set 28

2.4.3 Alkar set 31

2.5 Ožičenje 35

2.5.1 Napajanje sučelja 35

2.5.2 Spajanje servo motora na sučelje 36

2.5.3 Spajanje elektromotora na sučelje 36

2.6 Spajanje senzora 37

2.6.1 Taktilni senzori 37

2.6.2 Foto senzori 38

2.6.3 IR senzor 39

3. SUČELJA 42

3.1 USB sučelje PRO 43

3.1.1 Funkcionalnost 43

3.1.2 Modularni prikaz i portovi 43

3.1.3 Instalacija, provjera instalacije i USB komunikacije 45

3.2 CPU sučelje PRO 47

3.2.1 Funkcionalnost 47

3.2.2 Modularni prikaz i portovi 47

3.3 BT sučelje PRO 50

3.3.1 Funkcionalnost 50

3.3.2 Modularni prikaz i portovi 50

3.3.3 Instalacija upravljačkih programa i provjera instalacije 53

4. PROGRAMSKA PODRŠKA 60

4.1 FlawDraw 61

4.1.1 Instalacija 61

4.1.2 Registracija 62

4.2 FreeBasic IDE 63

4.2.1 Instalacija 63

4.3 BascomAVR (demo) 66

4.3.1 Instalacija 66

4.3.2 Kratke upute za Inovatic STK500 programatora 68

4.4 AVR Studio 4 i WinAVR 69

4.4.1 Instalacija 69

4.4.2 Kratke upute za Inovatic STK500 programatora 74

4.5 Atmel Flip 75

4.5.1 Instalacija 75

4.5.2 Instalacija Bootloader upravljačkih programa za USB sučelje 76
4.5.3 Kratke upute za korištenje 77
4.6 SinaProg 78
4.6.1 Instalacija 78
4.6.2 Kratke upute za korištenje s Inovatic STK500 programatorom 79
4.7 HyperTerminal 80
4.7.1 Instalacija 80
4.7.2 Kratke upute za korištenje 81

5. FLOWDRAW 83

6. FREEBASIC 89

7. BASCOMAVR 95
7.1 USB sučelje – AT90USB162 95
7.2 CPU sučelje – ATmega168 101
7.3 Bluetooth sučelje – ATmega32 109

8. AVR STUDIO 118

9. PROGRAMATOR 124
9.1 USB Inovatic STK500v2 ISP 124
9.1.1 Instalacija upravljačkih programa 125
9.1.2 HID i Slow SCK način rada 127

10. DODATAK 131
10.1 Princip rada preklopnih (relejnih) izlaza 131
10.2 Upravljanje servo motorima 133
10.3 Centriranje servo motora 134
10.4 Eksperimentalna pločica 138
10.5 Popis funkcija u FreeBasic-u 141
10.6 Popis funkcija u AVR Studio-u 162
10.7 Popis portova i sheme pojedinih modula 208
10.8 Uklanjanje pogrešaka (Troubleshooting) 219

11. INDEKS 224

UVOD

1

1. UVOD

Riječ „robot“ prvi je upotrijebio romanopisac Karel Čapek (Austro-Ugarska, danas Češka, 1890-1938) u svojoj predstavi R.U.R. (Rossumovi univerzalni roboti), koja je premijerno prikazana u Pragu 1921. Predstava prikazuje „umjetne ljude“ napravljane u tvornici i dizajnirane tako da uživaju u napornom radu, a bave se pitanjem imaju li takvi roboti ikakva prava. Riječ „robot“ Čapeku je sugerirao njegov brat Josef, a proizlazi iz češke riječi koja označava ropski rad.

Guiness World Records 2008

Proces izgradnje i programiranja robota je kombinacija mehanike, elektronike i logičnog razmišljanja. Edukacijski Mobilni Robot, (EMoRo) će Vas kroz projekte uputiti u „stvarni svijet“ elektronike i robotskog upravljanja, kakav možete vidjeti u praksi. Jedine razlike su veličina i sofisticiranost, ali načelo rada je gotovo isto.

Svrha ovog teksta je zaintrigirati svaku osobu koja želi napredovati u znanju o robotici i programiranju da na jednostavnim primjerima razvija kreativno razmišljanje i tako napreduje i stvara nove projekte, te traži novu primjenu i otkrije inženjera u sebi.

U ovoj knjizi pronaći ćete upute za sastavljanje robota, opis sučelja i upute za korištenje programske podrške. EMoRo je višejezično-programobilni robot, što u biti znači da se možete koristiti različitim programskim jezicima, od najjednostavnijih (dijagram tijeka) do profesionalnih programerskih alata (AVR Studio). Također ćete u knjizi doznati o kalibraciji motora, senzora i sve što je potrebno za „savršen“ rad bilo kojeg mobilnog robota.

KORISNICI

EMoRo je didaktičko sredstvo namijenjeno svim dobnim skupinama. Iako zvuči nemoguće, to je istina, jer educirati se uz pomoć EMoRo-a mogu i djeca predškolske dobi, ali i studenti i drugi napredni korisnici.

Djeca predškolske dobi mogu se koristiti EMoRo-om kao „autićem na daljinsko upravljanje“ kojim upravljaju preko tipkovnice i tako upoznaju računalo. Na taj način djeca razvijaju motoričke sposobnosti i razvijaju osjećaj za robotiku. Učenici osnovnih škola programiraju robota uz pomoć dijagrama toka i to tako da jednostavnim dodavanjem logičkih blokova „programiraju“ akcije koje robot u realnom vremenu izvodi. Napredniji osnovnoškolci i srednjoškolci mogu programirati u programskom jeziku FreeBasic tako da upisivanjem jednostavnih naredbi upravljaju robotom, a time i razvijaju pojam o stvarnoj logici upravljanja mikrokontrolerom, tj. robotom. Najnapredniji način programiranja je naravno C. Rabeći BascomAVR ili čak AVR studio najnapredniji korisnici imaju beskonačno mogućnosti i primjena, jedino ograničenje jenjihova kreativnost i fizičke mogućnosti pojedinog mikrokontrolera.

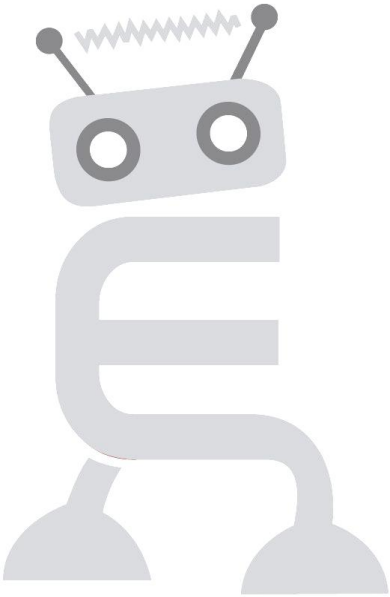
EMoRo EDUKACIJA

“EMoRo – Korisnički priručnik”, v1.0, Inovatic ICT d.o.o., 2012

Usporedno s korisničkim priručnikom možete na Inovatic youtube kanalu pratiti i učiti putem videoprimjera. Podijeljeni su na lekcije koje su organizirane tako da napredujete od jednostavnih do kompleksnih primjera i svakom prijeđenom lekcijom činite korak naprijed. Knjigu u digitalnom obliku i sve programe te primjere možete preuzeti na stranici www.inovatic.hr.

Kontinuirano dorađujemo i poboljšavamo edukacijski program, pa najnovije revizije potražite na našoj web stranici www.inovatic.hr.

Imate li prijedlog, mislite da ste pronašli pogrešku ili želite pridonijeti sljedećem izdanju knjige „EMoRo – Korisnički priručnik“, kontaktirajte s nama na podrska@inovatic.hr. Predbilježite se na naš edukacijski Youtube kanal – www.youtube.com/inovatic i ostanite u korak s lekcijama i novim primjerima.



Za sve nejasnoće ili probleme obratite se na:

podrska@inovatic.hr

ili na

01/3864-320 pon-pet od 8-16h

Poglavlje 1: UVOD

Sastavljanje EMoRo-a

2

2. SASTAVLJANJE EMoRo-a

EMoRo se sastoji od sljedećih mehaničkih dijelova:

- Šasija robota
- Dva pogonska kotača
- Svesmjerna kugla/Omniwheel[1]
- Pogonski servomotori/Elektromotorni pogon s prijenosom[2]
- Senzori (taktilni, IR, foto)
- Alkar set/Radar set/Testna pločica s komponentama[3]

Šasija robota sastoji se od dva dijela i izrađena je od crno-eloksiranog tvrdog aluminija. Konstruirana je tako da se mogu odabrati dvije vrste pogona, dva prednja svesmjerna kotača, potpuno je kompatibilna sa svim senzorima i dodacima za EMoRo.

Pogonski kotači dolaze u setu koji se sastoji od nekoliko dijelova. Kotače je potrebno sastaviti prema uputama priloženima u setu ili prema uputama koje će biti objašnjene unutar ovog korisničkog priručnika.

U setu je i prednji svesmjerni kotač koji dolazi u dvije verzije (ovisno o prethodnom izboru) – Omniwheel i BallCaster. Upute za sastavljanje i opis pojedinog kotača nalazi se u ovom priručniku.

Sastavljanje pogona EMoRo-a također se dijeli na dvije verzije, tako možemo upotrijebiti za pogon servomotore s kontinuiranom rotacijom ili elektromotorni pogon s prijenosom. Servomotore potrebno je samo spojiti sa šasijom, dok je elektromotorni pogon s prijenosom potrebno prije sastaviti. Elektromotorni pogon s prijenosom može se sastaviti tako da je prijenos podešen na omjer 58:1 ili na 203:1. Ako se prijenos sastavi u omjeru 203:1, EMoRo će se gibati manjom brzinom te će njime biti lakše rukovati. Ovakav se prijenos preporučuje za početnike. Prijenos 58:1 je gotovo četiri puta manji od 203:1, što znači da bi se robot tada gibao gotovo četiri puta brže. Ovakav prijenos rabe napredni korisnici koji se s EMoRo-om žele nadmetati u brzinskim natjecanjima. U ovom korisničkom priručniku obradit ćemo sastavljanje prijenosa 203:1, a korisnici koji žele sastaviti prijenos u omjeru 58:1 mogu se služiti uputama uzset elektromotornog prijenosa.

EMoRo ima nekoliko setova s kojima dolazi, ovisno o odabiru konfiguracije (EMoRo USB Classic, EMoRo CPU Senior, EMoRo IR Radar - Bluetooth ili EMoRo Alkar 3D - Bluetooth). Sve setove i njihov sadržaj možete pronaći na našoj stranici www.inovatic.hr.

[1] Pri narudžbi moguće je odabrati prednji kotač koji može biti BallCaster (svesmjerna kugla) ili Omniwheel

[2] Pri narudžbi moguće je odabrati vrstu pogona – elektromotori sa prijenosom ili servo motori

[3] Ovisno o odabranoj konfiguraciji (USB Classic, CPU senior, IR Radar ili 3D alkar) dolaze pojedini setovi

2.1 **Sastavljanje kotača**

Napomena!

Odvojiti plastični dio iz kompleta za matice kojim ćete se koristiti u nastavku sastavljanja EMoRo robota – slika 1.

Slika 1. Plastični dio za pritezanje matica

2.1.1 Kotač za pogonske servo motore

Slika 2. Kotač za pogonske servo motore

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Po potrebi skalpel ili neki nožić

Postupak
1. Odvojiti kotač od ostalih plastika iz vrećice
2. Odrezati preostale komadiće plastike s kotača
3. Postaviti zvjezdasti nastavak za servo (servo rog) na kotač tako da se srednje rupe poklapaju s velikim rupama na kotaču
4. Zavrnuti tri kratka špičasta vijka na kotač
5. Postaviti gume na kotače

Slika 3. Sastavljeni kotač za pogonske servo motore

2.1.2 Kotač za elektromotorni prijenos

Slika 4. Kotač za elektromotorni prijenos

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Po potrebi skalpel ili neki nožić

Postupak
1. Odvojiti kotač i prirubnicu za osovinu od ostalih plastika iz vrećice
2. Odrezati preostale komadiće plastike s kotača i prirubnice
3. Postaviti prirubnicu na sredinu kotača u odgovarajuće rupe
4. Zavrnuti tri duža vijka na kotač
5. Postaviti gume na kotače

Slika 5. Sastavljeni kotač za elektromotorni prijenos

2.2 Sastavljanje prednjeg kotača

2.2.1 Svesmjerni kotač - Omniwheel

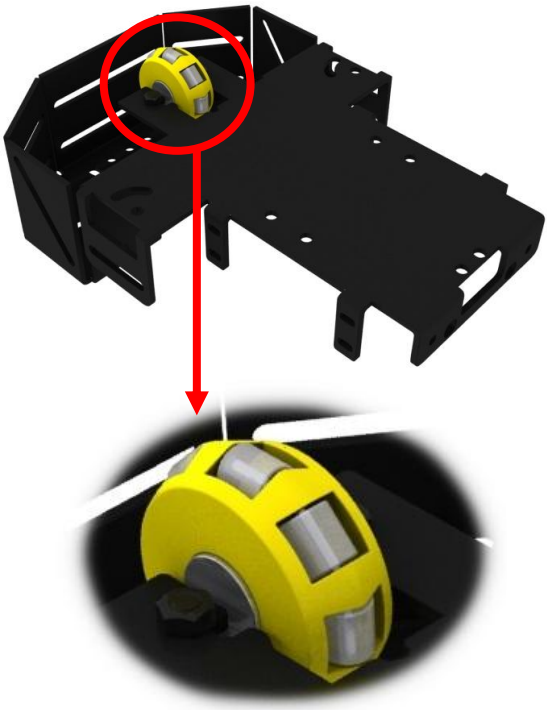
Slika 6. Svesmjerni kotač - Omniwheel

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Plastični zatezač za matice (uz kotač)

Postupak
1. Izvaditi sve dijelove iz vrećice
2. Postaviti osovinu u svesmjerni kotač
3. Postaviti dva vijka tako da prolaze kroz osovinu kao na slici ispod

Slika 7. Svesmjerni kotač - Omniwheel

4. Postaviti osovinu s kotačem i vijcima na odgovarajuće mjesto na donjem dijelu šasije robota
5. Stegnuti matice



Slika 8. Sastavljeni kotač za elektromotorni prijenos

2.2.2 Svesmjerna kugla - BallCaster

Slika 9. Svesmjerna kugla - BallCaster

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Plastični zatezač za matice (uz kotač)
- Po potrebi skalpel ili neki nožić

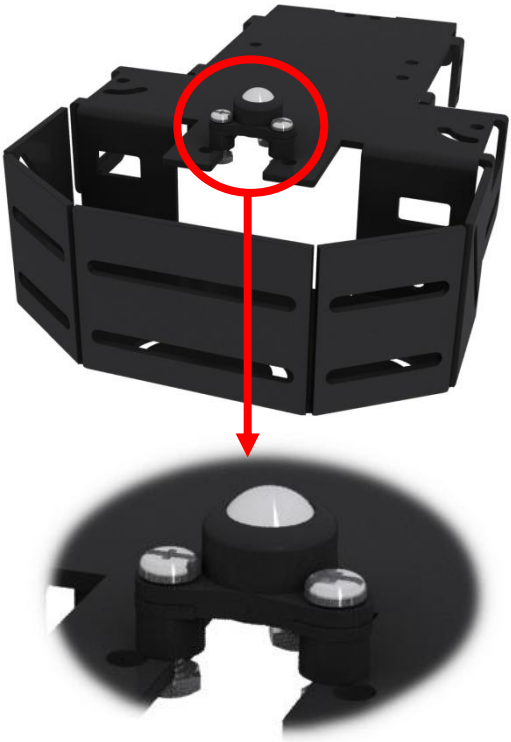
Postupak
1. Postaviti kuglu i metalne osovine na donji plastični dio kao na slici

Slika 10. Postavljanje kugle i metalnih osovina u donji plastični dio

2. Staviti odgovarajući plastični dio - slika

Slika 11. Postavljanje gornje plastike

3. Sastavljenu svesmjernu kuglu zavrnuti na donji dio šasije s tri dugačka vijka i tri matice iz kompleta



Slika 12. Sastavljena svesmjerna kugla na šasiji

2.3 Sastavljanje pogona

2.3.1 Servo motori

Slika 13. Pogonski servo motor

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Plastični zatezač za matice (uz kotač)

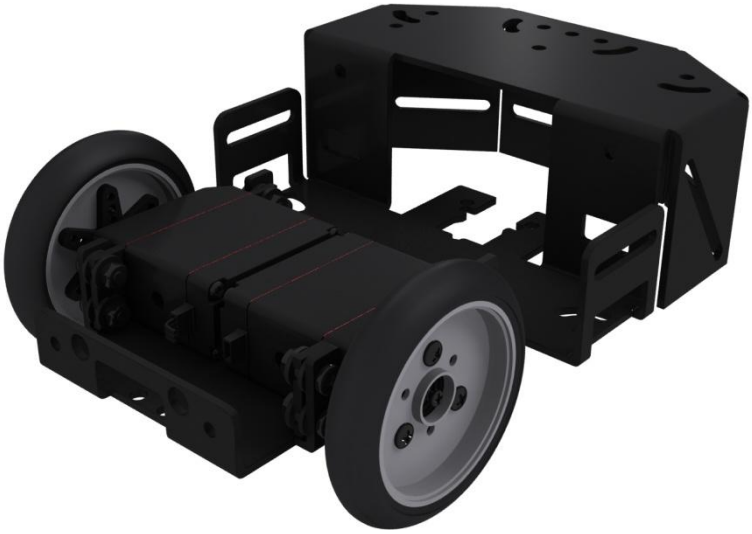
Postupak
1. Postaviti gumice u odgovarajuća mjesta na servo motoru
2. U gumice utisnuti metalne čahurice

Slika 14. Postavljanje gumica i metalnih čahurica na servo motor

3. Postaviti servo motor na šasiju kao na sljedećoj slici
4. Pritegnuti vijke i matice

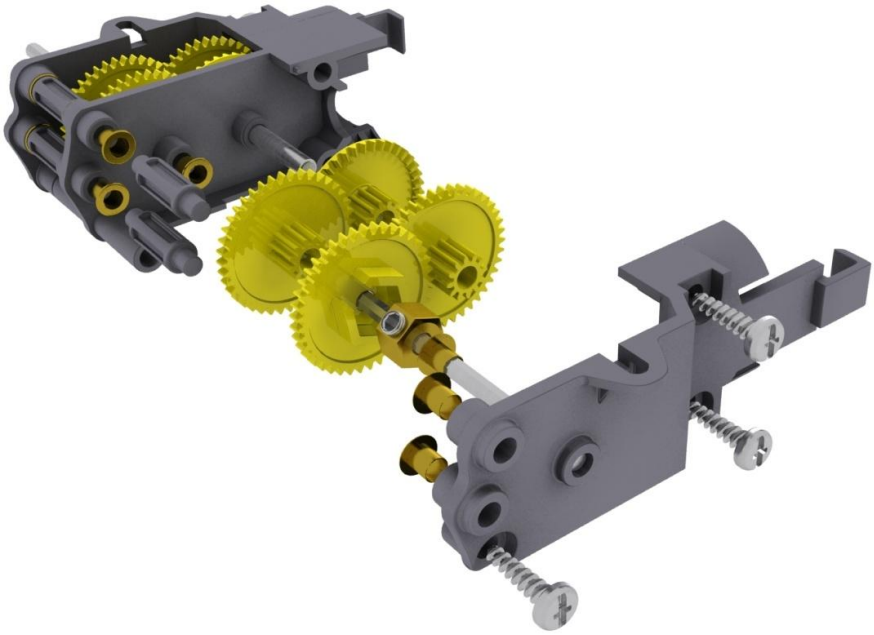
Slika 15. Postavljanje pogonskog servo motora na šasiju

5. Zategnuti srednji vijak (mali crni) na osovinu servo motora



Slika 16. Sastavljeni pogonski servo motori i kotači na šasiji

2.3.2 Elektromotorni pogon s prijenosom – low speed C



Slika 17. Prijenos za elektromotore – Twin Motor Gearbox (1. brzina)

Potreban alat:

- Križni odvijač PH1 (dolazi uz robotića)
- Mali imbus (dolazi uz EM pogon)
- Po potrebi skalpel ili nožić

Postupak

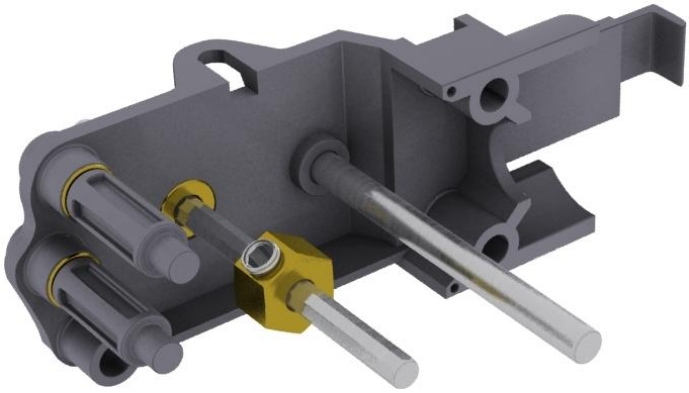
1. Postaviti male bijele plastične zupčanike na elektromotore (najlakši način je postaviti zupčanik na radnu površinu i pritisnuti motor na njega – osovina motora mora doći do ruba zupčanika)

Slika 18. Postavljanje zupčanika na elektromotor

2. Postaviti šesterokutnu mesinganu maticu na osovinu prijenosa
3. Podesiti visinu na središnjem plastičnom dijelu, u rupi s oznakom C
4. Stegnuti mali imbus vijak – **napomena**[4]
5. Ponoviti isti postupak s drugim vretenom

Slika 19. Podešavanje visine matice na osovini pogona

6. Izabrati plastični dio (veliki „desni“ dio kao na slici ispod)
7. Postaviti tri mesingane čahurice na odgovarajuća mjesta (ispod mesingane matice, te ispod plastičnih osovina, kao što je prikazano na slici)
8. Postaviti dvije plastične osovine
9. Postaviti šesterokutnu osovinu s maticom tako da duži kraj ide kroz rupu C
10. Valjkastu osovinu postaviti u rupu desno od šesterokutne osovine



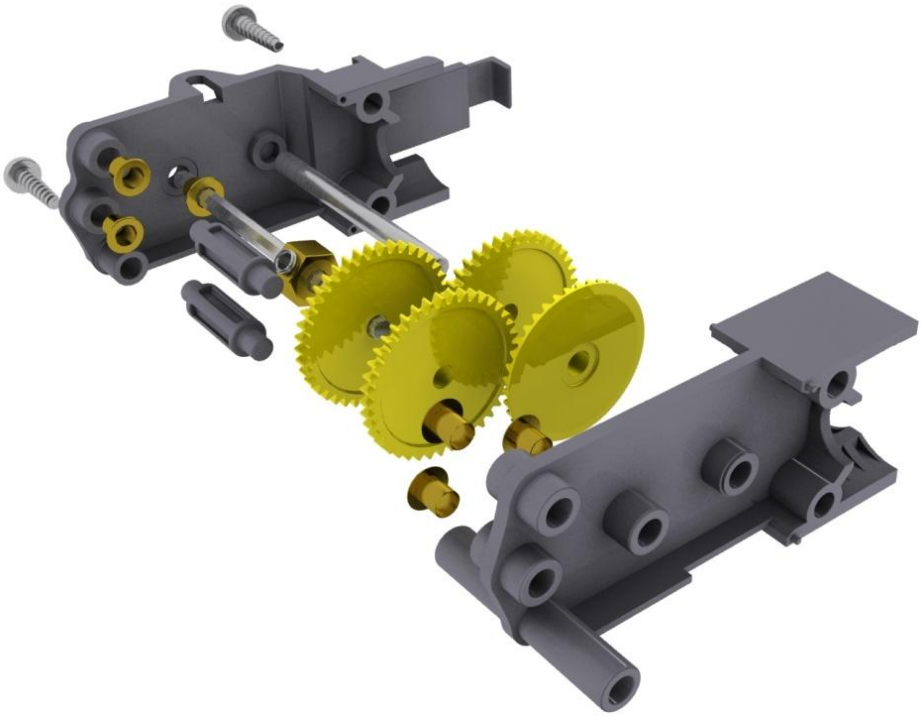
Slika 20. Sastavljanje desnog dijela prijenosa 01

[4] Obratiti pozornost na to da imbus stegnemo na stranicu osovine, a ne na brid radi boljeg držanja cijelog pogona. Ako je imbus zategnut na bridu, vrlo brzo se rasklima.

11. Postaviti zupčanik s ležištem za mesinganu maticu(imbus prolazi kroz prorez na ležištu)
12. Postaviti dva ista obična zupčanika iz vrećice
13. Postaviti zupčanik koji nema zupce na obodu
14. Postaviti mesingane čahurice kao na slici ispod

Slika 21. Sastavljanje desnog dijela prijenosa 02

15. Potrebno stegnuti tri vijka na predviđenom mjestu



Slika 22. Shematski prikaz spajanja desnog dijela prijenosa

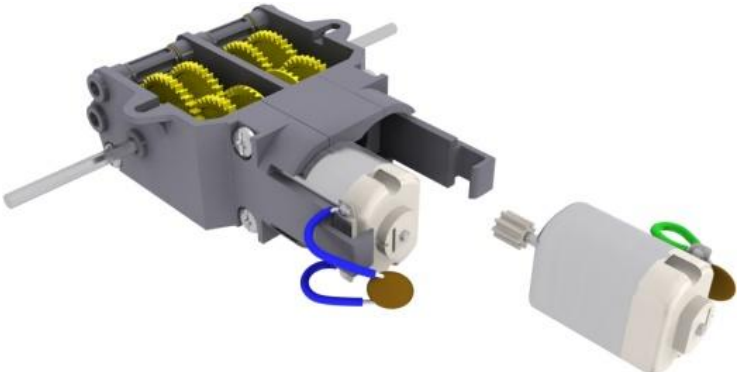
16. Ponoviti postupak za lijevu stranu prijenosa (od 14. prema 6. koraku)

Slika 23. Sastavljanje lijevog dijela prijenosa

17. Potrebno stegnuti tri vijka na predviđenom mjestu

Slika 24. Sastavljen prijenos - gearbox

18. Kod postavljanja elektromotora na prijenos obratiti pozornost na boju žice – desno zeleni, lijevo plavi



Slika 25. Postavljanje elektromotora na prijenos

19. Postaviti elektromotorni pogon s prijenosom na donji dio šasije
20. Zavrnuti vijke i matice kao na slici ispod

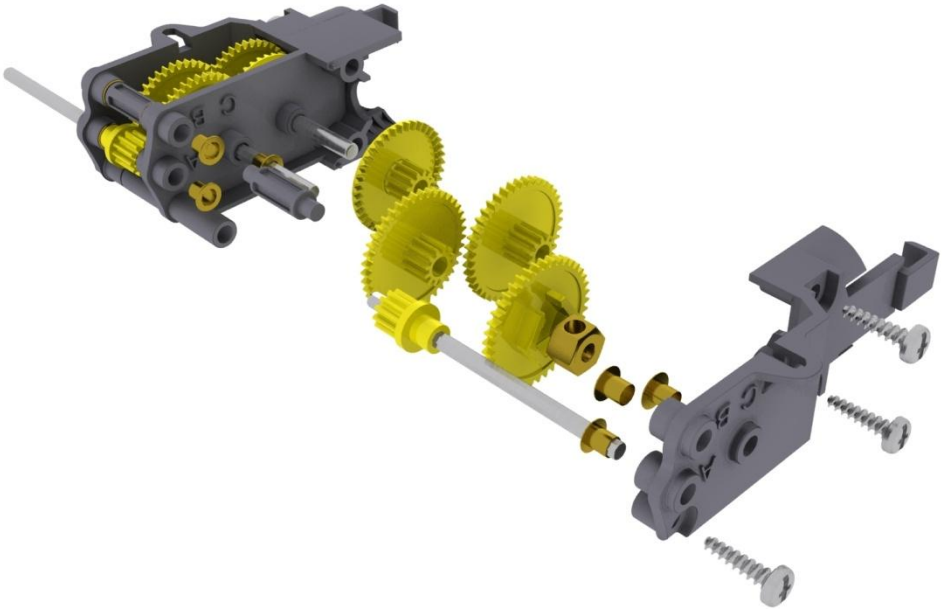
Slika 26. Postavljanje elektromotornog pogona na šasiju

21. Istovremeno nabiti lijevi i desni kotač na osovinu

Slika 27. Sastavljen elektromotorni pogon s prijenosom i kotačima na šasiji

22. Okretati polagano kotače i dobro namastiti sve zupčanike (mast je u plavoj tubi priloženoj uz prijenos)

2.3.3 Elektromotorni pogon s prijenosom – high speed A



Slika 28. Prijenos za elektromotore – Twin Motor Gearbox (1. brzina)

Potreban alat:

- Križni odvijač PH1 (priložen uz robotića)
- Mali imbus (priložen uz EM pogon)
- Po potrebi skalpel ili nožić

Postupak

1. Postaviti male bijele plastične zupčanike na elektromotore (najlakši način je postaviti zupčanik na radnu površinu i pritisnuti motor na njega – osovina motora mora doći do ruba zupčanika)

Slika 29. Postavljanje zupčanika na elektromotor

2. Postaviti šesterokutnu mesinganu maticu na jednu osovinu prijenosa
3. Podesiti visinu na središnjem plastičnom dijelu, u rupi s oznakom A
4. Stegnuti mali imbus vijak – **napomena**[5]

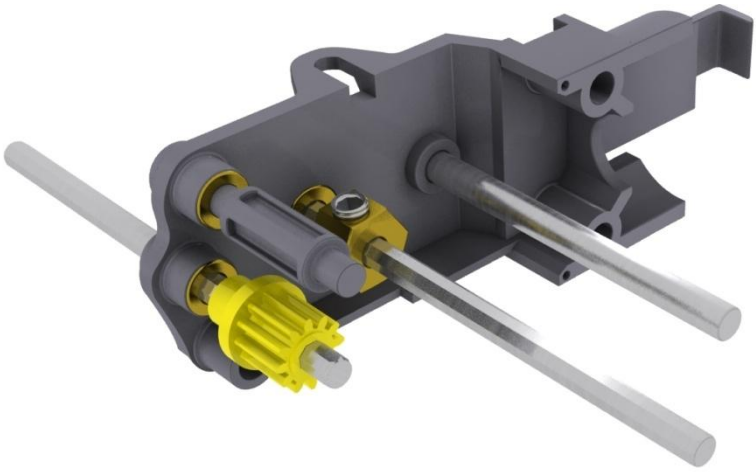
Slika 30. Podešavanje visine matice na osovini pogona

5. Postaviti plastični zupčanik na osovinu prijenosa kao na slici ispod
6. Podesiti visinu na središnjem plastičnom dijelu, u rupi s oznakom A
7. Ponoviti postupak

Slika 31. Podešavanje visine zupčanika na osovini pogona

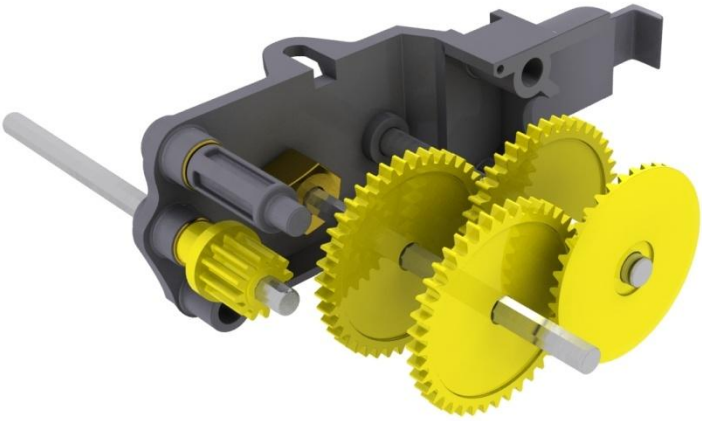
[5] Obratiti pozornost na to da imbus stegnemo na stranicu osovine, a ne na brid radi boljeg držanja cijelog pogona. Ako je imbus zategnut na bridu, vrlo brzo se rasklima.

8. Izabrati plastični dio (veliki „desni“ dio kao na slici ispod)
9. Postaviti tri mesingane čahurice na odgovarajuća mjesta (ispod mesingane matice, te ispod plastičnih osovina, kao što je prikazano na slici)
10. Postaviti dvije plastične osovine
11. Postaviti šesterokutnu osovinu s maticom tako da kraći kraj ide kroz rupu C
12. Valjkastu osovinu postaviti u rupu desno od šesterokutne osovine



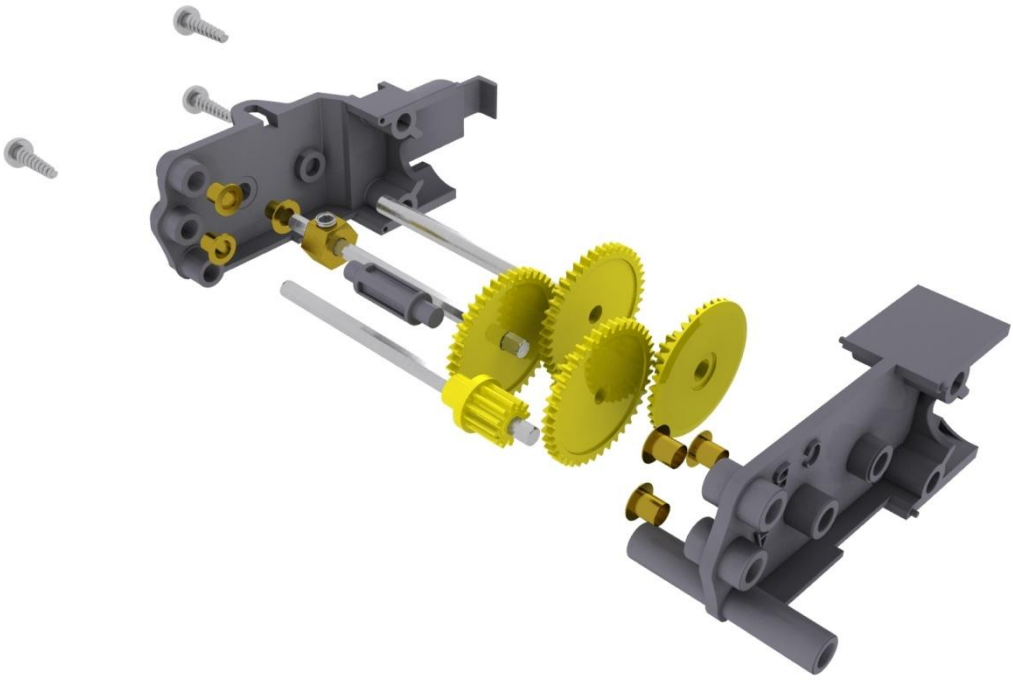
Slika 32. Sastavljanje desnog dijela prijenosa 01

13. Postaviti zupčanik s ležištem za mesinganu maticu(imbus prolazi kroz prorez na ležištu)
14. Postaviti osovinu sa zupčanikom tako da duži kraj ide kroz rupu A
15. Postaviti dva ista obična zupčanika iz vrećice
16. Postaviti zupčanik koji nema zupce na obodu
17. Postaviti mesingane čahurice kao na slici ispod



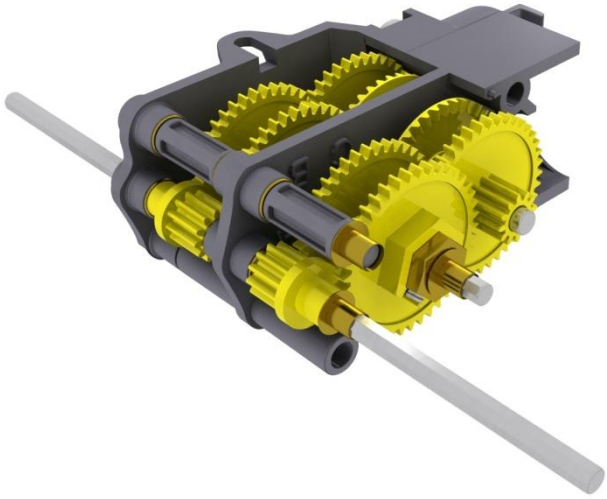
Slika 33. Sastavljanje desnog dijela prijenosa 02

18. Potrebno stegnuti tri vijka na predviđenom mjestu



Slika 34. Shematski prikaz spajanja desnog dijela prijenosa

19. Ponoviti postupak za lijevu stranu prijenosa (od 14-og prema 6-om koraku)

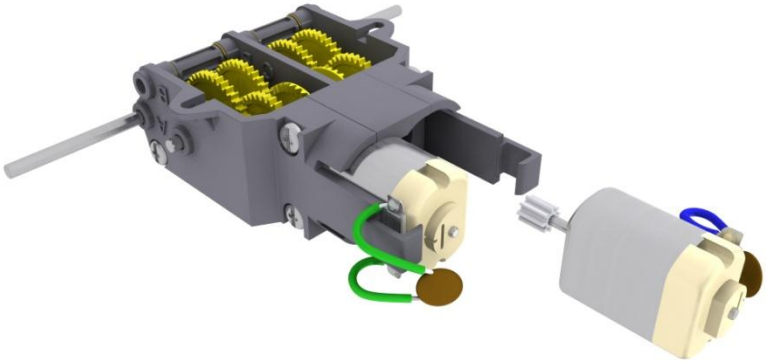


Slika 35. Sastavljanje lijevog dijela prijenosa

20. Potrebno stegnuti tri vijka na predviđenom mjestu

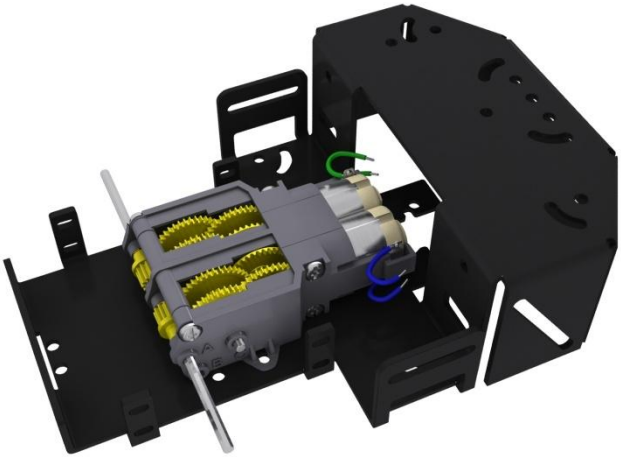
Slika 36. Sastavljen prijenos - gearbox

21. Pri postavljanju elektromotora na prijenos obratiti pozornost na boju žice – desno zeleni, lijevo plavi



Slika 37. Postavljanje elektromotora na prijenos

22. Postaviti elektromotorni pogon s prijenosom na donji dio šasije
23. Zavrnuti vijke i matice kao na slici ispod



Slika 38. Postavljanje elektromotornog pogona na šasiju

24. Istovremeno nabiti lijevi i desni kotač na osovinu

Slika 39. Sastavljen elektromotorni pogon sa prijenosom i kotačima na šasiji

25. Okretati polagano kotače i dobro namastiti sve zupčanike (mast se nalazi u plavoj tubi priloženoj uz prijenos)

2.4 Sastavljanje šasije

2.4.1 Osnovni set

Potreban alat:

- Križni odvijač PH1 (dolazi uz robotića)

Postupak

1. Postaviti sučelje na gornji metalni dio šasije kao na sljedećoj slici
2. Zategnuti četiri vijka

Slika 40. Postavljanje sučelja na gornji dio šasije

3. Postaviti četiri plastične spojke (snap-in) na odgovarajuća mjesta na šasiji, tako da se učvrste dva dijela šasije

Slika 41. Spajanje gornjeg i donjeg dijela šasije plastičnim spojkama

2.4.2 Radar set

Potreban alat:
- Križni odvijač PH1 (dolazi uz robotića)
- Plastični zatezač za matice (uz kotač)

Postupak
1. Postaviti senzor udaljenosti na odgovarajući metalni dio
2. Zategnuti vijke i matice

Slika 42. Postavljanje senzora udaljenosti na metalni dio

3. Ravni plastični dio od servo motora postaviti kao na slici
4. Stegnuti dva mala crna špičasta vijka

Slika 43. Spajanje servo roga sa senzorom udaljenosti

5. Postaviti servo motor na metalni radar nosač kao na sljedećoj slici
6. Zategnuti dva crna špičasta vijka

Slika 44. Postavljanje servo motora na metalni nosač za radar

7. Spojiti dio sastavljen u koraku 4 s osovinom servo motora - **napomena**[6]
8. Zategnuti vijak servo motora u osovinu

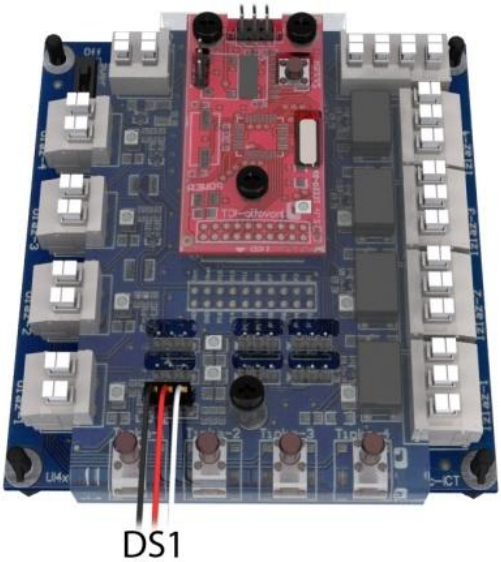
Slika 45. Kompletiranje radar seta

9. Postaviti sastavljeni radar na šasiju
10. Zategnuti vijke i matice iz kompleta kao na sljedećoj slici

Slika 46. Postavljanje radar seta na šasiju

[6] Pri spajanju potrebno obratiti pozornost na kut rotacije servo motora, znači postaviti senzor tako da bude točno na polovici maksimalnog kuta. Najlakše je to odrediti tako da stavimo tu plastiku, okrenemo servo motor na maksimalni položaj u lijevu stranu, te izvadimo plastiku i postavimo ju ponovno da senzor „gleda“ točno pod 90°, tj. lijevo. Poseban oprez pri ručnom pomicanju senzora IR radara (lagano zakretati senzor kako se ne bi oštetio precizan prijenos zakretnog servo motora

11. Spojiti senzor udaljenosti na sučelje kao na sljedećoj slici (s lijeva na desno – crna-crvena-bijela)



Slika 47. Spajanje senzora udaljenosti na sučelje

2.4.3 Alkar set

Potreban alat:
- Križni odvijač PH1 (priložen uz robotića)
- Plastični zatezač za matice (uz kotač)

Postupak
1. Postaviti gumice u odgovarajuće utore na servo motoru
2. Plastični dio za servo motor pričvrstiti na osovinu - **napomena**[7]
3. Zategnuti vijak za osovinu servo motora

Slika 48. Postavljanje gumica i servo roga za alkar set

4. Postaviti metalne podloške na gumice
5. Postaviti servo motor na metalni nosač
6. Stegnuti vijke i matice kako je prikazano na sljedećoj slici

Slika 49. Sastavljanje donjeg servo motora za alkar set

[7] Prilikom postavljanja plastike na osovinu servo motora obratiti pažnju na sljedeće – staviti plastiku, motor zakrenuti u maksimalni desni položaj, izvaditi plastiku, te ju ponovno staviti u ravnini sa servo motorom

7. Postaviti drugi metalni nosač servo motora kao na sljedećoj slici
8. Zategnuti dva mala crna špičasta vijka

Slika 50. Postavljanje metalnog nosača za drugi servo motor

9. Na drugom servo motoru postaviti gumice
10. Staviti servo motor kako je prikazano na sljedećoj slici (osovina servo motora okrenuta prema van)

Slika 51. Postavljanje drugog servo motora

11. Postaviti metalne podloške kao na prethodnoj slici
12. Zategnuti vijke i matice iz kompleta (glava vijka „naslonjena“ na metalni podložak, a matica s vanjske strane metalnog nosača)

13. Stisnuti plastični nosač koplja (dio iz kompleta za servo motor) kroz odgovarajuću rupu na koplju
14. Zategnuti dva vrlo mala crna vijka kao što je prikazano na sljedećoj slici

Slika 52. Sastavljanje koplja za set alkar

15. Spojiti koplje sa servo motorom - **napomena**[8]
16. Zategnuti vijak za osovinu servo motora

Slika 53. Kompletiranje alkar seta

[8] Pri spajanju potrebno obratiti pažnju na kut rotacije servo motora, znači postaviti koplje tako da bude točno na polovici maksimalnog kuta. Najlakše je to odrediti tako da stavimo koplje, okrenemo servo motor na maksimalni položaj u lijevu stranu, te izvadimo plastiku i postavimo ju ponovno da koplje „gleda“ točno pod 90□, tj. lijevo, tako da kopljem možemo upravljati punim kutem od 180□.

17. Postaviti sastavljeni alkar set na dvije središnje rupe kao na slici ispod
18. Stegnuti dva vijka i dvije matice

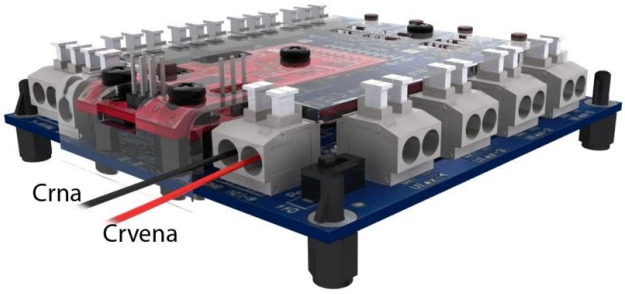
Slika 54. Postavljanje alkar seta na šasiju 02

Slika 55. Postavljanje alkar seta na šasiju 02

2.5 **Ožičenje**

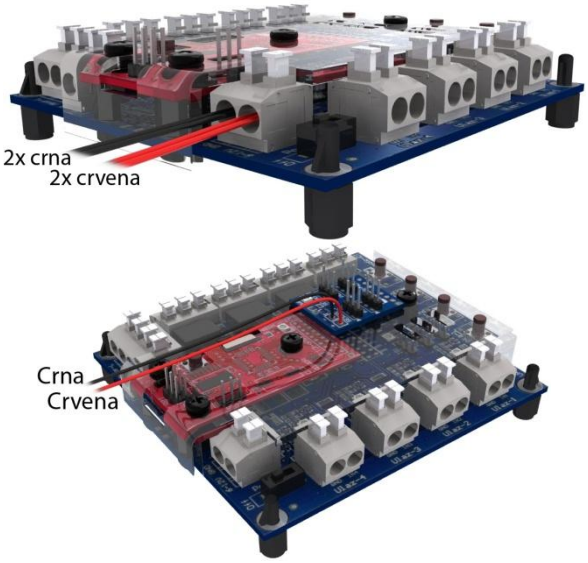
2.5.1 Napajanje sučelja

Sučelje se napaja preko 6AA baterija (preporučeno punjive baterije). Napajanje se spaja kao na sljedećoj slici (crna žica s baterija na GND, a crvena žica na 6-12V).



Slika 56. Spajanje napajanja na sučenje

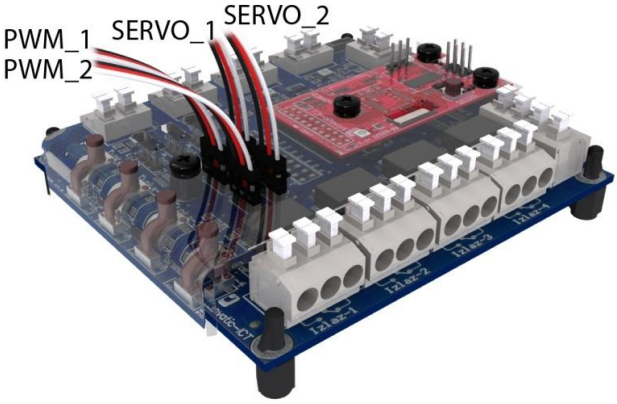
Servo motori se napajaju sa sučelja i njihov napon je 5V, iako postoji mogućnost nadogradnje robota s adapterom koji je priložen uz set alkar, s kojim se napon servo motora diže na 7,2V, tj. na napon baterija i dobivaju se bolje performanse. Način spajanja adaptera prikazan je na sljedećoj slici.



Slika 57. Spajanje adaptera za servo motore

2.5.2 Spajanje servo motora na sučelje

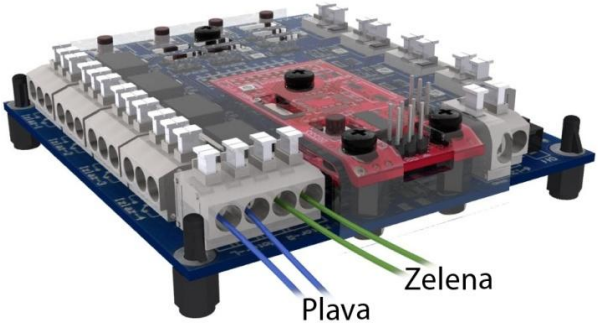
Pogonski servo motori spajaju se uvijek na konektore SERVO_1 i SERVO_2, dok se radar i alkar set spajaju na PWM_1 i PWM_2. Ako se koristi alkar set, servo motori se spajaju na adapter koji je priložen uz set, s tim da su na adapteru potpuno isti nazivi kao i na pločici pa je i načelo spajanja isto.



Slika 58. Spajanje servo motora na sučelje

2.5.3 Spajanje elektromotora na sučelje

Spajanje elektromotora na sučelje predviđeno je kao na sljedećim slikama. Za servo motore bitna je orijentacija, tj. koji je lijevi, a koji desni. Na slikama za sastavljanje elektromotornog pogona, vidljivo je da desni motor ima zelene žice, a lijevi plave, na to treba obratiti pozornost i pri spajanju na sučelje (Motor_L – lijevi, Motor_R – desni)



Slika 59. Spajanje elektromotora na sučelje

2.6 Spajanje senzora

Na EMoRo se može spojiti velik broj senzor. Mogućnost stavljanja taktilnih, foto, IR i senzora udaljenosti je velika, svi mogu biti postavljeni na gotovo bilo koje mjesto na robotu. Sama šasija zamišljena je tako da korisnik može, ovisno o svojim potrebama, staviti senzore na prednji, bočni ili stražnji dio robota. U nastavku teksta objasnit ćemo načelo stavljanja pojedinih senzora na mjesta koja će biti iskorištena u video uputama.

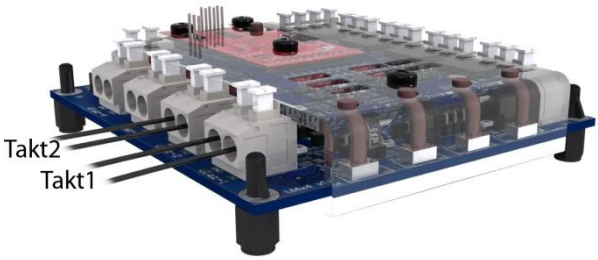
2.6.1 Taktilni senzori

Potreban alat:
- Križni odvijač PH1 (priložen uz robotića)
- Plastični zatezač za matice (uz kotač)

Postupak
1. Postaviti taktilne senzore na prednji dio šasije tako da nasjedaju na predviđene rupe (mogućnost zakretanja po potrebi)
2. S gornje strane taktilnih senzora staviti vijke, a s unutarnje strane šasije zavrnuti maticu

Slika 60. Postavljanje taktilnih senzora na šasiju

11. Spojiti senzore na sučelje kao na sljedećoj slici



Slika 61. Spajanje taktilnih senzora na sučelje

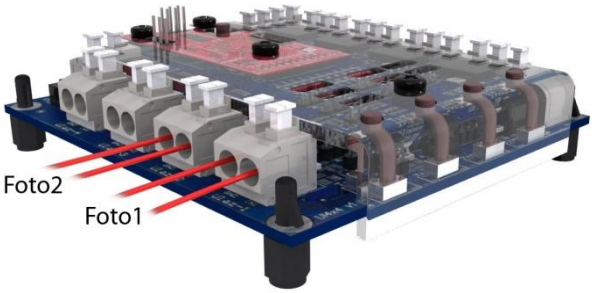
2.6.2 Foto senzori

Potreban alat:
- Križni odvijač PH1 (priložen uz robotića)

Postupak
1. Postaviti fotosenzore na unutrašnji dio šasije
2. Podesiti odgovarajuću visinu (pomaknuti nosač senzora na gornju ili donju rupu na prednjem dijelu šasije, te podesiti visinu senzora)
3. S vanjske strane šasije staviti vijak koji prolazi kroz senzor i stegnuti maticu

Slika 62. Postavljanje foto senzora na šasiju

4. Spojiti senzore na sučelje kao na sljedećoj slici



Slika 63. Spajanje foto senzora na sučelje

2.6.3 IR senzor

Potreban alat:

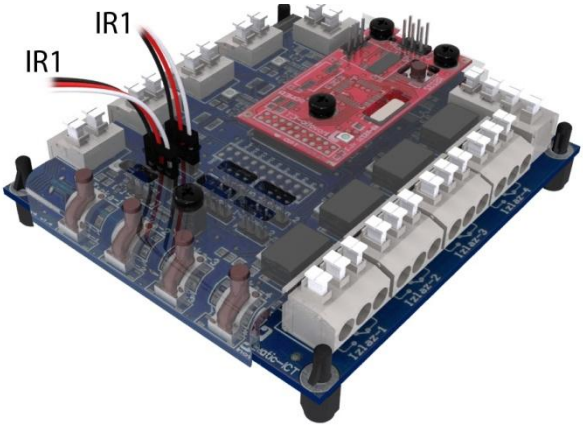
- Križni odvijač PH1 (priložen uz robotića)

Postupak

1. Postaviti IR (infra crvene) senzore na unutrašnji dio šasije
2. Podesiti odgovarajuću visinu (pomaknuti nosač senzora na gornju ili donju rupu na prednjem dijelu šasije, te podesiti visinu senzora – oko 2cm od poda)
3. S vanjske strane šasije staviti vijak koji prolazi kroz senzor i stegnuti maticu

Slika 64. Postavljanje IR senzora na šasiju

4. Spojiti senzore na sučelje kao na sljedećoj slici



Slika 65. Spajanje IR senzora na sučelje

Sučelja

3

3. SUČELJA

EMoRo sučelje ima širok spektar primjene, kao što je upravljanje robotom, upravljanje modelima svjetlosne signalizacije, servo motorima, DC motorima, mnogim senzorima kao osjetilom za interakciju s okolinom i sl. Koncept sučelja prilagođen je početnicima, ali i naprednim korisnicima. EMoRo sučelje podržava programske alate za sve dobne skupine i kao edukacijsko sredstvo može se koristiti u osnovnoj školi, srednjoj školi te na fakultetu.

EMoRo sučelja zasnivaju se na AVR porodici mikroračunala (ATMEGA32, ATMEGA168, AT90USB162), danas među najzastupljenijima na tržištu, te su najbolji izbor za ulazak u svijet programiranja, mikroračunala i elektronike. "EMoRo UI4x4 pro" sučelje podržava 3 mikrokontrolerska modula koji korisnicima omogućavaju prilagodbu sučelja vlastitim potrebama.

	USB pro	**CPU pro**	**Bluetooth pro**
Napajanje	5V USB[9]/ 6-9V ext	6-9V ext	6-9V ext
USB	Da	Ne	Ne
Bluetooth	Ne	Ne	Da
RS232	Ne	Da	Ne
Broj ulaza	11	11	11
Broj relejnih izlaza	4	4	4
Broj servo izlaza	7 max	7 max	7 max
Foto senzori	4	4	4
Taktilni senzori	4 (11)	4 (11)	4 (11)
IR senzori	11 max	11 max	11 max
Temperaturni senzor	7 max	7 max	7 max
Senzori udaljenosti	0	1 max	1 max
Izlaz DC motora	2	2	2
Autonomni rad	Da	Da	Da
Način programiranja	Bez programatora / programator	Programator	Programator
Programator	Bootloader / ISP	ISP	ISP / JTAG
JTAG Debug	Ne	Ne	Da
One-Wire Debug	Da	Da	Ne
GUI	Da	Da	Da
FreeBasic	Da	Da	Da
Bascom AVR	Da	Da	Da
AVR Studio	Da	Da	Da

Tablica 1. Usporedba EMoRo sučelja

[9] Napajanje sučelja preko USB porta na računalu moguće je uz korištenje samo ulaza i relejnih izlaza. Za servo motore potrebno je dovesti vanjsko napajanje (6-9V), jer USB na računalu ima mogućnost dati 5V i 0.5A, što nije dovoljno za servo motor.

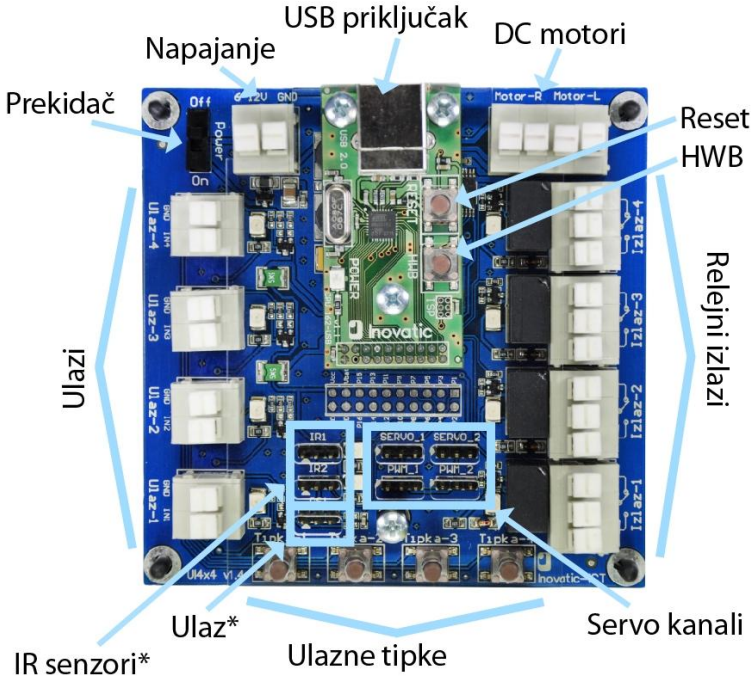
3.1 *USB sučelje PRO*

3.1.1 Funkcionalnost

Mikrokontrolerski modul CPU162USB podržava USB komunikaciju s računalom preko koje provodi naredbe slikovnog programiranja, ili programiranja u programskom jeziku Free Basic. Ovakav koncept sučelja idealan je za osnovne i srednje škole zbog jednostavnosti programiranja u alatima koji su namijenjeni toj dobnoj skupini.

Sučelje podržava korištenje naprednijih alata BASCOM, C te assembler kojima sučelje postaje potpuno autonomno i neovisno o računalu. Za programiranje mikrokontrolerskog modula nije potreban dodatni programator jer se program može upisati preko USB komunikacije.

3.1.2 Modularni prikaz i portovi



Slika 66. USB sučelje – modularni prikaz

USB sučelje sastoji se od četiri preklopna (relejna) izlaza[10] (Izlaz-1 – Izlaz4) i četiri ulaza[11] (Ulaz1 – Ulaz4) koji su povezani s četiri tipke (Tipka-1 – Tipka-4). Na sučelju postoji mogućnost spajanja najviše sedam servo motora[12] (SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1). Kako mikrokontroler AT90USB162 nema ADC (analogno-digitalni pretvarač), ulaz DS1 početno je postavljen kao klasičan ulaz, dok su ulazi IR1 i IR2 namijenjeni za spajanje dvaju IR senzora. Sučelje također ima mogućnost upravljanja s dva DC motora (Motor-R, Motor-L) s karakteristikama 5V, max 1A po motoru.

Napomena! Ako se spajaju DC motori, moguće je upotrijebiti samo četiri servo kanala (SERVO_1, IR1, IR2 i DS1).

Mogućnosti spajanja na sučelje:

	Mogućnosti spajanja/primjena
Ulaz1 – Ulaz4	Taktilni senzori Foto senzori IR senzori (napajanje[13]) Bilo koji senzor koji ima izlaz 0V i 5V
Tipka1 – Tipka4	Ulaz1 – Tipka1 Ulaz2 – Tipka4 Ulaz3 – Tipka3 Ulaz4 – Tipka4
Izlaz1 – Izlaz4	Relejni izlazi (NO, NC i CO[14]) Spajanje žaruljice Preklapanje ulaza (u kompleksnijim video uputama)
IR1* i IR2*	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Foto senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)
DS1 (Ulaz*)	TTL, Servo motor
SERVO_1, SERVO_2, PWM_1 i PWM_2	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)
Motor-L	DC motor (5V, 1A)
Motor-R	DC motor (5V, 1A)
Reset	Ponovno pokretanje programa
HWB	U kombinaciji s Reset tipkom – Atmel Flip
6-12V – GND	Napajanje 6-12V (+ i -), s tim da uz dodatak za alkara napajanje je ograničeno na 6-9V
USB priključak	Povezivanje s računalom (USB A-B kabel)
Power prekidač	ON/OFF prekidač za sučelje

Tablica 2. USB sučelje – mogućnosti

[10] U „Dodatku“ je kratak opis i upute za korištenje relejnih izlaza
[11] Mogućnost spajanja taktilnih, foto i IR senzora, te ostalih senzora s izlaznom naponskom razinom 0V i 5V
[12] U svim primjerima koristit ćemo SERVO_1 i SERVO_2 za pogonske 360 servo motore, a PWM_1 i PWM_2 za spajanje dodatnih servo motora (Alkar i Radar setovi)
[13] Crna žica na GND ulaza, a bijela na INx, crvenu potrebno spojiti na 2. pin jednog od preostalih konektora (IR1, IR2, DS1, SERVO_1, SERVO_2, PWM_1, PWM_2)
[14] NO – normalno otvoren, NC – normalno zatvoren i CO – preklopni kontakt

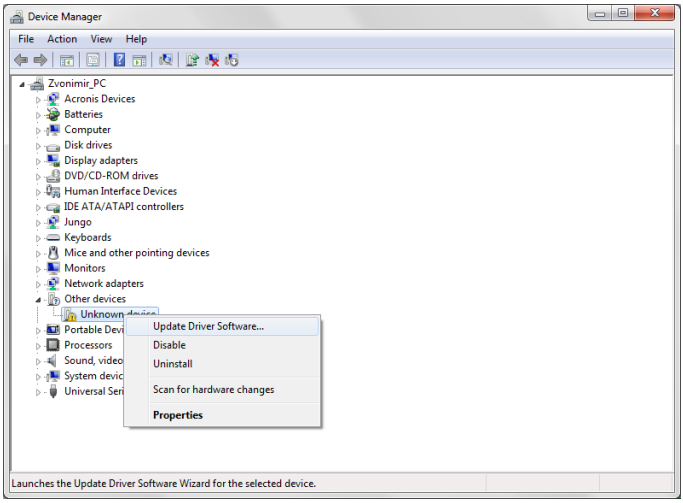
3.1.3 Instalacija, provjera instalacije i USB komunikacije

Ovisno o tome imate li x86 ili x64 Windowse, postoje dvije vrste upravljačkih programa. Provjeru obavite na sljedeći način:

- Start → All Programs → Accessories → System Tools → System Information
- Pod stavkom „System type“ ispisan je tip operacijskog sustava

Instalacija upravljačkih programa:

- Povezati USB sučelje USB kabelom s računalom
- Otvoriti Device Manager (Pritisnuti zajedno na tipkovnici znak „Windows“ (između Ctrl i Alt) i slovo „R“ → upisati „devmgmt.msc“ (bez navodnika) → OK)



Slika 67. Instalacija upravljačkih programa – USB sučelje

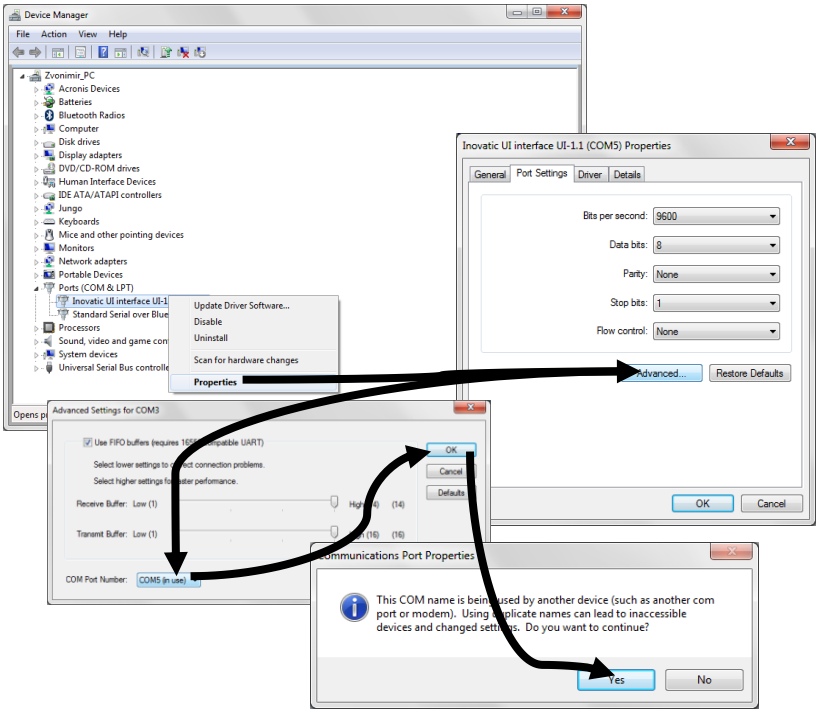
- <u>Windows Vista i 7</u> - (desni klik na Unknown device) → Update Driver Software → Browse my computer for driver software → Browse → (*DVD* → Drivers → USB sucelje → x86_XpVista7 ili x64_XpVista7) → OK → Next → Install this driver software anyway → Close

- <u>Windows XP</u> - (desni klik na Unknown device) → Update Driver → No, not this time → Next → Install from specific location (Advanced) → Next → Include this location in the search → Browse → (*DVD* → Drivers → USB sucelje → x86_XpVista7 ili x64_XpVista7) → OK → Next → Continue Anyway → Finish

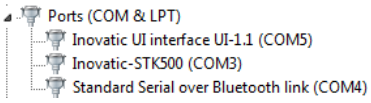
Nakon uspješne instalacije upravljačkih programa potrebno je provjeriti COM port na kojem je instaliran (AVR Studio prepoznaje COM1, COM2, COM3 i COM4), te je potrebno prebaciti na neki niži slobodni (COM3 ili COM4)

— (desni klik na Inovatic UI interface) → Properties → Port Settings → Advanced → COM Port Number promijeniti u manji od COM8 → OK → Yes → OK → (Odspojiti i spojiti ponovno sučelje s računalom)

Prikaz tijeka provjere COM portova



Nakon uspješno završene instalacije upravljačkih programa i provjere COM portova, sučelje je spremno za korištenje. Uspješno instalirano sučelje prikazano je na slici ispod.



Slika 68. Uspješna instalacija USB sučelja

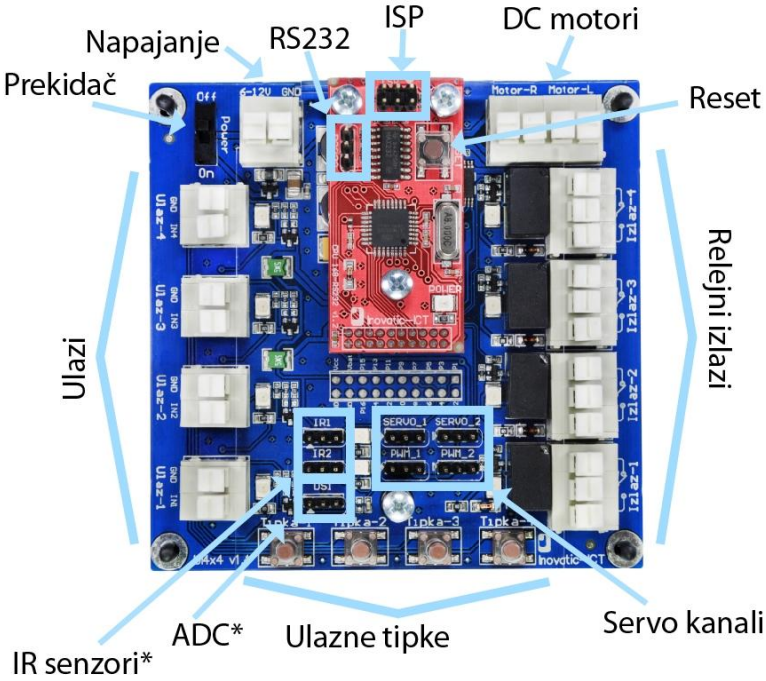
3.2 *CPU sučelje PRO*

3.2.1 Funkcionalnost

Mikrokontrolerski modul CPU168-RS232 podržava RS232 komunikaciju sa računalom preko koje provodi naredbe slikovnog programiranja, ili programiranja u programskom jeziku Free Basic. Ovo sučelje prvenstveno je namijenjeno naprednijim korisnicima, jer uz pregršt gotovih biblioteka omogućava jednostavno programiranje u AVR Studio-u.

Sučelje podržava korištenje naprednijih alata BASCOM, C te assembler kojima sučelje postaje potpuno autonomno i neovisno o računalu. Za programiranje mikrokontrolerskog modula potreban je ISP programator Inovatic STK500.

3.2.2 Modularni prikaz i portovi



Slika 69. CPU sučelje – modularni prikaz

CPU sučelje sastoji se od četiri preklopna (relejna) izlaza[15] (Izlaz-1 – Izlaz4) i četiri ulaza[16] (Ulaz1 – Ulaz4) koji su povezani s četiri tipke (Tipka-1 – Tipka-4). Na sučelju postoji mogućnost spajanja maksimalno sedam servo motora[17] (SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1). Za senzore koji za izlaz daju analognu vrijednost (npr. Senzor udaljenosti) koristimo ADC (analogno-digitalni pretvarač), tj. ulaz DS1, dok su ulazi IR1 i IR2 namijenjeni za spajanje dvaju IR senzora. Sučelje također ima mogućnost upravljanja s dva DC motora (Motor-R, Motor-L) s karakteristikama 5V, max 1A po motoru.

Napomena! Ako se spajaju DC motori, moguće je upotrijebiti samo četiri servo kanala (SERVO_1, IR1, IR2 i DS1).

Serijska komunikacija (RS232) na CPU sučelju omogućava nam spajanje sa računalom, te samim time primanje/slanje informacija/naredbi sučelju. Primanje/slanje obavljamo u Windows programu HyperTerminal. Upute za korištenje programa opširnije su opisane u poglavlju „*4.8 HyperTerminal*“.

Mogućnosti spajanja na sučelje:

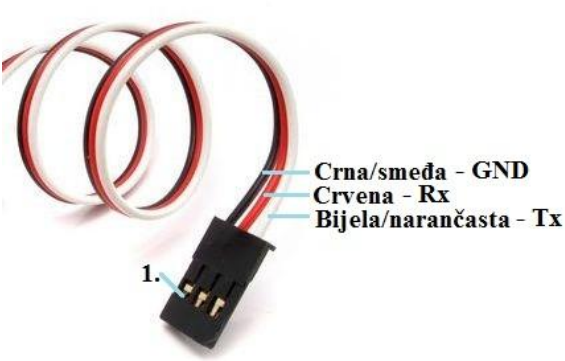
	Mogućnosti spajanja/primjena
Ulaz1 – Ulaz4	Taktilni senzori Foto senzori IR senzori (napajanje[18]) Bilo koji senzor koji ima izlaz 0V i 5V
Tipka1 – Tipka4	Ulaz1 – Tipka1 Ulaz2 – Tipka4 Ulaz3 – Tipka3 Ulaz4 – Tipka4
Izlaz1 – Izlaz4	Relejni izlazi (NO, NC i CO[19]) Spajanje žaruljice Preklapanje ulaza (u kompleksnijim video uputama)
IR1* i IR2*	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Foto senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)
DS1 (ADC*)	Senzor udaljenosti TTL Servo motor Bilo koji senzor koji kao izlaz daje analognu vrijednost između 0V i 5V
SERVO_1, SERVO_2, PWM_1 i PWM_2	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)

[15] U „Dodatku“ nalazi se kratak opis i upute za korištenje relejnih izlaza
[16] Mogućnost spajanja taktilnih, foto i IR senzora, te ostalih senzora s izlaznom naponskom razinom 0V i 5V
[17] U svim primjerima koristit ćemo SERVO_1 i SERVO_2 za pogonske 360 servo motore, a PWM_1 i PWM_2 za spajanje dodatnih servo motora (Alkar i Radar setovi)
[18] Crna žica na GND ulaza, a bijela na INx, crvenu potrebno spojiti na 2. pin jednog od preostalih konektora (IR1, IR2, DS1, SERVO_1, SERVO_2, PWM_1, PWM_2)
[19] NO – normalno otvoren, NC – normalno zatvoren i CO – preklopni kontakt

Motor-R	DC motor (5V, 1A)
Reset	Ponovno pokretanje programa
ISP	Konektor za programiranje sa Inovatic STK500 programatorom
6-12V – GND	Napajanje 6-12V (+ i -), s tim da uz dodatak za alkara napajanje je ograničeno na 6-9V
RS232 priključak	Povezivanje na serijski port računala
Power prekidač	ON/OFF prekidač za sučelje

Tablica 3. CPU sučelje - mogućnosti

RS232 – Serijska komunikacija



Slika 70. Konektor za RS232 - pinovi[20]

[20] Prvi pin za RS232 na CPU sučelju označen je s trokutićem i na njega se spaja crna žica

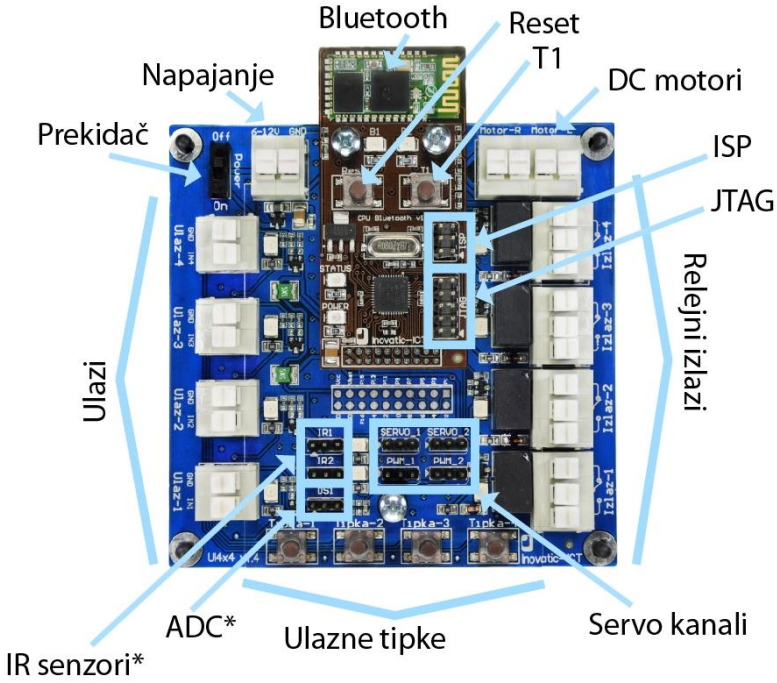
3.3 *BT sučelje PRO*

3.3.1 Funkcionalnost

Mikrokontrolerski modul CPU Bluetooth podržava Bluetooth komunikaciju sa računalom preko koje provodi naredbe slikovnog programiranja, ili programiranja u programskom jeziku Free Basic. Ovakav koncept sučelja idealan je za učenike osnovne i srednje škole zbog jednostavnosti programiranja u alatima koji su namijenjeni toj dobi. CPU Bluetooth sučelje osigurava bežičnu komunikaciju između računala, te je idealan za autonoman pogon robota i omogućuje nam primanje povratnih informacija na računalo, i slanje dodatnih instrukcija na robota.

Sučelje podržava korištenje naprednijih alata BASCOM, C te assembler kojima sučelje postaje potpuno autonomno i neovisno o računalu. Za programiranje mikrokontrolerskog modula potreban je ISP programator Inovatic STK500.

3.3.2 Modularni prikaz i portovi



Slika 71. Bluetooth sučelje – modularni prikaz

Bluetooth sučelje sastoji se od četiri preklopna (relejna) izlaza[21] (Izlaz-1 – Izlaz4) i četiri ulaza[22] (Ulaz1 – Ulaz4) koji su povezani s četiri tipke (Tipka-1 – Tipka-4). Na sučelju postoji mogućnost spajanja maksimalno sedam servo motora[23] (SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1). Za senzore koji za izlaz daju analognu vrijednost (npr. senzor udaljenosti) koristimo ADC (analogno-digitalni pretvarač), tj. ulaz DS1, dok su ulazi IR1 i IR2 namijenjeni za spajanje dvaju IR senzora. Sučelje također ima mogućnost upravljanja s dva DC motora (Motor-R, Motor-L) s karakteristikama 5V, max 1A po motoru.

Napomena! Ako se spajaju DC motori, moguće je upotrijebiti samo četiri servo kanala (SERVO_1, IR1, IR2 i DS1).

Bluetooth komunikacija na Bluetooth sučelju omogućava nam spajanje sa računalom, te samim time primanje/slanje informacija/naredbi sučelju. Primanje/slanje obavljamo u Windows programu HyperTerminal. Upute za korištenje programa dopširnije su opisane u poglavlju „*4.8 HyperTerminal*“.

Mogućnosti spajanja na sučelje:

	Mogućnosti spajanja/primjena
Ulaz1 – Ulaz4	Taktilni senzori Foto senzori IR senzori (napajanje[24]) Bilo koji senzor koji ima izlaz 0V i 5V
Tipka1 – Tipka4	Ulaz1 – Tipka1 Ulaz2 – Tipka4 Ulaz3 – Tipka3 Ulaz4 – Tipka4
Izlaz1 – Izlaz4	Relejni izlazi (NO, NC i CO[25]) Spajanje žaruljice Preklapanje ulaza (u kompleksnijim video uputama)
IR1* i IR2*	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Foto senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)
DS1 (ADC*)	Senzor udaljenosti TTL Servo motor Bilo koji senzor koji kao izlaz daje analognu vrijednost između 0V i 5V
SERVO_1, SERVO_2, PWM_1 i PWM_2	Infracrveni senzor Servo motor Taktilni senzori (1. i 2. pin) Bilo koji senzor koji ima izlaz 0V i 5V (1. i 2. pin)

[21] U „Dodatku“ je kratak opis i upute za korištenje relejnih izlaza
[22] Mogućnost spajanja taktilnih, foto i IR senzora, te ostalih senzora s izlaznom naponskom razinom 0V i 5V
[23] U svim primjerima koristit ćemo SERVO_1 i SERVO_2 za pogonske 360 servo motore, a PWM_1 i PWM_2 za spajanje dodatnih servo motora (Alkar i Radar setovi)
[24] Crna žica na GND ulaza, a bijela na INx, crvenu potrebno spojiti na 2. pin jednog od preostalih konektora (IR1, IR2, DS1, SERVO_1, SERVO_2, PWM_1, PWM_2)
[25] NO – normalno otvoren, NC – normalno zatvoren i CO – preklopni kontakt

Motor-L	DC motor (5V, 1A)
Motor-R	DC motor (5V, 1A)
Reset	Ponovno pokretanje programa
T1	Programibilna tipka na bluetooth modulu U kombinaciji s Reset tipkom vraća modul na početne postavke (passkey, ime...)
ISP	Konektor za programiranje s Inovatic STK500 programatorom
JTAG	Konektor za programiranje s AVR JTAG programatorom/debuggerom
6-12V – GND	Napajanje 6-12V (+ i -), s tim da je uz dodatak za alkara napajanje ograničeno na 6-9V
Bluetooth	Povezivanje preko bluetootha na serijski port računala
Power prekidač	ON/OFF prekidač za sučelje

Tablica 4. Bluetooth sučelje – mogućnosti

Postavljanje Bluetooth sučelja na početne postavke:

1. Pritisnuti i držati tipku Reset
2. Pritisnuti tipku T1 (dalje držati Reset)
3. Pustiti tipku Reset
4. Pustiti tipku T1

3.3.3 *Instalacija upravljačkih programa i provjera instalacije*

Odmah nakon spajanja USB Bluetooth dongla s računalom, Windowsi prepoznaju novi uređaj i automatski instaliraju sve potrebne drivere (ako se koriste Windows XP, Vista, 7). Nakon što Windowsi „odrade“ svoj dio posla, u donjem desnom kutu ekrana, pojavi se ikona za Bluetooth, kako je prikazano na slici ispod. To je znak da je Bluetooth dongle dobro instaliran i spreman za korištenje.

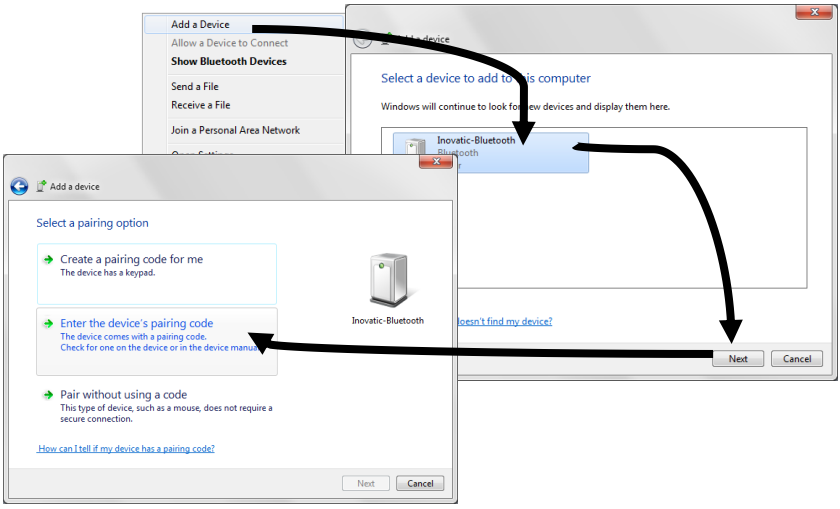
Slika 72. Pronađen USB bluetooth adapter

Nakon završene instalacije USB Bluetooth-a potrebno je spojiti Bluetooth sučelje na napajanje i upaliti na ON/OFF prekidaču. Kad smo upalili sučelje, svijetli crvena „Power“ svjetleća dioda i treperi plava B1 svjetleća dioda. Nakon uspješno završenog povezivanja svjetleća dioda B1 polako treperi i prilikom pokretanja programa u FreeBasic-u ili pri spajanju na HyperTerminal svjetleća dioda B2 konstantno svijetli.

Povezivanje Bluetooth sučelja s računalom (Vista i 7)

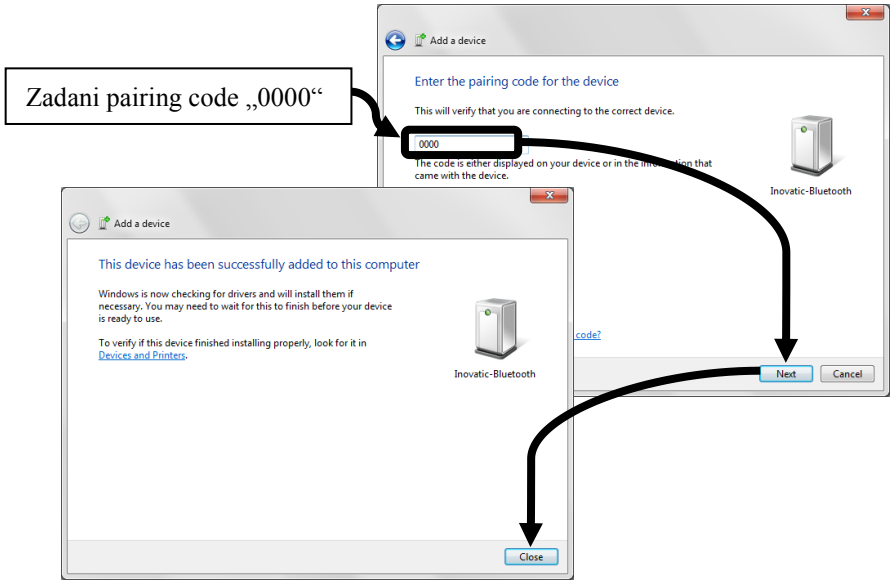
Kako bi povezali Bluetooth sučelje sa računalom, potrebno je desnom tipkom miša kliknuti na ikonu Bluetooth u donjem desnom kutu, te odabrati stavku „Add a Device“. Odabirom te stavke otvara nam se novi prozor gdje Bluetooth prikazuje sve uređaje koji su u dometu. Nakon što pronađe sve dostupne uređaje, prikazat će nam se i „Inovatic-Bluetooth“ kako je prikazano na slici.

Prikaz tijeka dodavanja novog uređaja



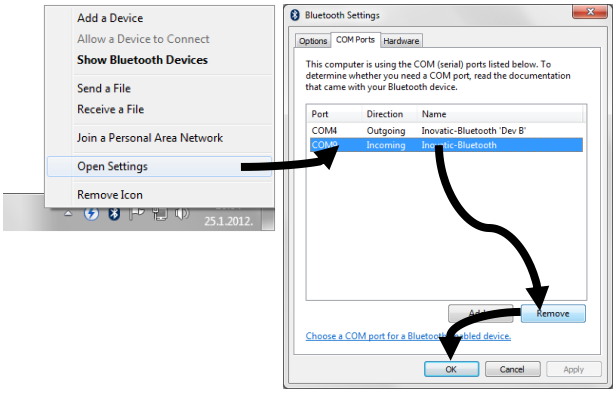
Odabirom „Inovatic-Bluetooth“ pronađenog uređaja i klikom na „Next“ otvara nam se slijedeći prozor u koji moramo upisati „Pairing Code“ (kod za uparivanje s Bluetooth sučeljem). Osnovni kod za uparivanje je „0000“. Odabirom „Enter the device's pairing code“ otvara nam se novi prozor u koji upišemo „0000“ i kliknemo na „Next“. Nakon toga Windowsi odrađuju svoj dio posla dodjeljivanja izlaznih i ulaznih portova, pa pričekamo da završe sa instalacijom.

Prikaz tijeka spajanja uređaja s računalom



Završetkom instalacije potrebnih drivera, desnom tipkom miša kliknemo na Bluetooth ikonu, odaberemo stavku „Open Settings“ i otvorimo tab pod nazivom „COM Ports“, kako je prikazano na sljedećoj slici. Odaberemo „Inocoming“ port i kliknemo na „Remove“ i onda na „OK“.

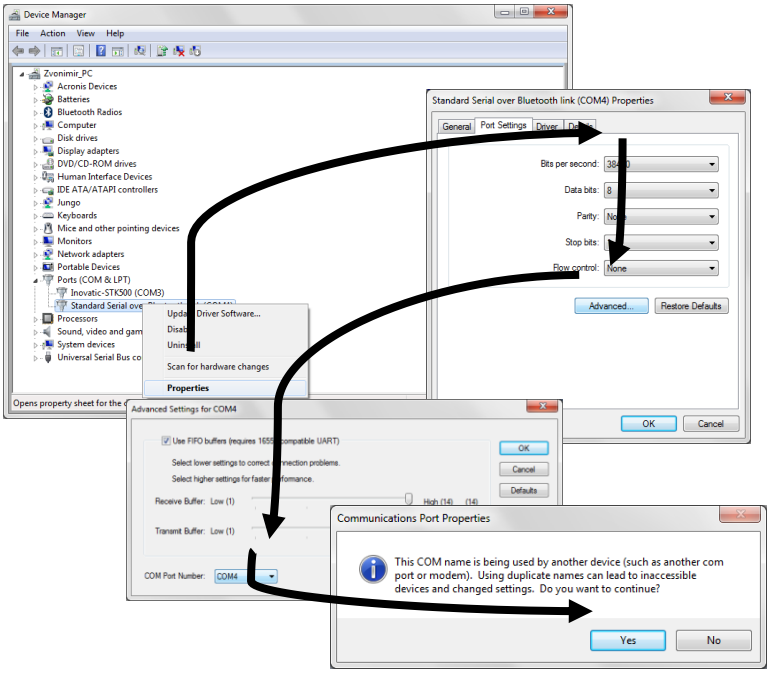
Prikaz tijeka podešavanja uređaja



Zadnji korak u podešavanju veze je ručno postavljanje „Outgoing“ COM Porta na manji od 8. Ako je već automatski podešen manji COM Port od COM8 (ili COM8), nije potrebno raditi sljedeće.

Otvoriti Device Manager (pritisnuti zajedno na tipkovnici znak „Windows“ (između Ctrl i Alt) i slovo „R“ → upisati „devmgmt.msc“ (bez navodnika) → OK) → Ports → Standard Serial over Bluetooth link → desni klik - Properties → odabir taba Port Settings → potrebno promijeniti Bits per second na „38400“ → Advanced → „COM Port Number“ → odabir COM Porta manjeg od 8 → OK → Yes → OK

Prikaz tijeka podešavanja COM portova

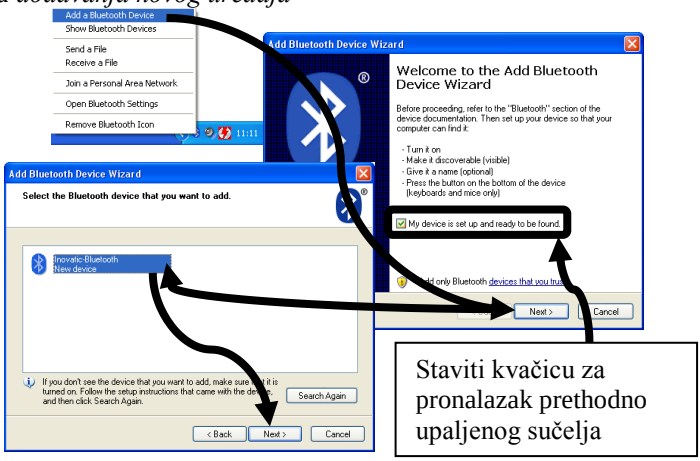


Nakon dodavanja uređaja ,završene instalacije potrebnih drivera, brisanja „Incoming COM Porta“ i podešavanja „Outgoing COM Porta“ možemo prijeći na upravljanje sučelja preko FreeBasic-a ili FlowDraw-a.

Povezivanje Bluetooth sučelja s računalom (XP)

Kako bismo povezali Bluetooth sučelje s računalom, potrebno je desnom tipkom miša kliknuti na Bluetooth ikonu u donjem desnom kutu, te odabrati stavku „Add a Bluetooth Device“. Odabirom te stavke otvara se novi prozor u kojem označimo kvačicu na „My device is set up and ready to be found“, te kliknemo na „Next“.

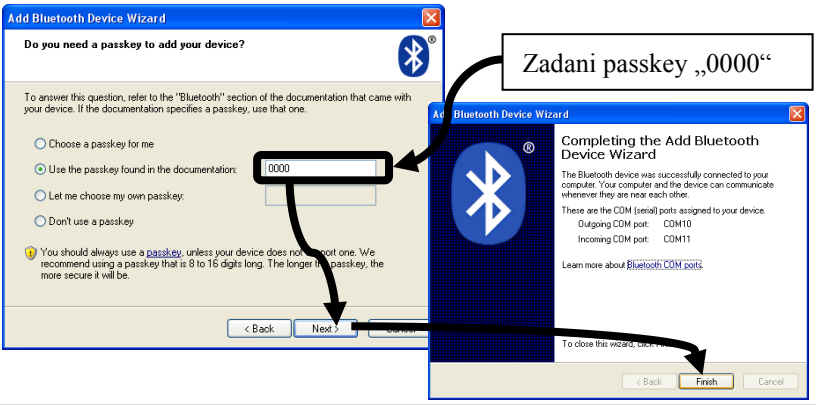
Prikaz tijeka dodavanja novog uređaja



U slijedećem prozoru Bluetooth prikazuje sve uređaje koji su u dometu. Nakon što pričekamo da pronađe sve dostupne uređaje, prikazat će nam se i „Inovatic-Bluetooth“.

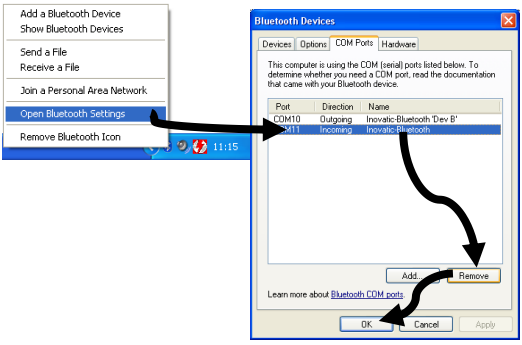
Odabirom „Inovatic-Bluetooth“ pronađenog uređaja i klikom na „Next“ otvara nam se sljedeći prozor gdje moramo upisati „Passkey“ (kod za uparivanje s Bluetooth sučeljem). Osnovni kod za uparivanje je „0000“. Odabirom „Let me choose my own passkey“, upisom „0000“ i klikom na „Next“ Windowsi odrađuju svoj dio posla dodjeljivanja izlaznih i ulaznih portova, pa pričekamo da završe instalaciju.

Prikaz tijeka spajanja uređaja s računalom



Završetkom instalacije potrebnih drivera, desnom tipkom miša kliknemo na Bluetooth ikonu, odaberemo stavku „Open Bluetooth Settings“ i otvorimo tab pod nazivom „COM Ports“, kao što je prikazano na sljedećoj slici. Odaberemo „Inocoming“ port i kliknemo na „Remove“ i onda na „OK“.

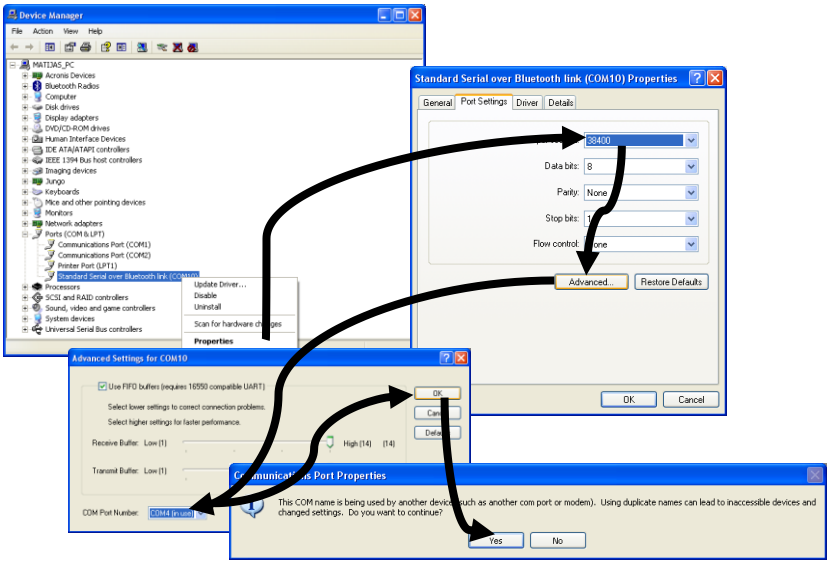
Prikaz tijeka podešavanja uređaja



Zadnji korak u podešavanju veze je ručno postavljanje „Outgoing“ COM Porta na manji od 8. Ako nam je već automatski podešen manji COM Port od COM8 (ili COM8), nije potrebno raditi sljedeće.

Otvoriti Device Manager (pritisnuti zajedno na tipkovnici znak „Windows“ (između Ctrl i Alt) i slovo „R“ → upisati „devmgmt.msc“ (bez navodnika) → OK) → Ports → Standard Serial over Bluetooth link → desni klik - Properties → odabir taba Port Settings → potrebno promijeniti Bits per second na „38400“ → Advanced → „COM Port Number“ → odabir COM Porta manjeg od 8 → OK → Yes → OK

Prikaz tijeka podešavanja COM portova



Programska podrška

4

4. PROGRAMSKA PODRŠKA

EMoRo kompleti mogu se koristiti/programirati na četiri načina, ovisno o prethodno stečenoj razini znanja.

1. Auto na daljinsko upravljanje
   - Nije potrebno prethodno znanje – robot postaje igračka. Ovaj oblik korištenja robota namijenjen je isključivo za upoznavanje djece/korisnika sa računalom, te za razvijanje motoričkih sposobnosti korištenjem tipkovnice i miša.

2. FlowDraw
   - Najjedostavniji način programiranja dijagramom tijeka. Dijagram tijeka je pomoćno sredstvo koje je neovisno o programskom jeziku i računalu, a vizualizira zadatak pa on postaje pregledniji. Dijagram tijeka sastoji se od niza jednostavnih geometrijskih likova spojenih usmjerenim crtama. Usmjerene crte pokazuju tijek rješavanja zadatka.
3. FreeBasic
   - Potrebno poznavanje osnovne sintakse Basic-a. Perfektno znanje ovog programskog jezika nije preduvjet da bi se moglo programirati već je samo potrebno znati značenje naredbi u tom programskom jeziku - tj. ključne riječi ili naredbe. Isto tako visoko znanje matematike ili znanosti nije također preduvjet da bi se znalo ili da bi osoba bila vična programiranju.
4. BascomAVR
   - Potrebno osnovno znanje o mikroračunalima. Programeru je čitava arhitektura mikroprocesora vidljiva preko simboličkih naziva, kao da se radi o "obicnim" programskim varijablama. To bitno pojednostavljuje kako programiranje, tako i razumijevanje načina rada ovih prilično složenih sklopova.
5. AVR Studio
   - Potrebno osnovno znanje o mikroračunalima, te poznavanje sintakse programskog jezika C. Korištenjem gotovih biblioteka cijela arhitektura mikroračunala vidljiva je preko simboličkih naziva, što bitno pojednostavljuje programiranje. Najnapredniji korisnici imaju mogućnost sami pisati svoje biblioteke i iskoristiti puni potencijal mikroračunala.

Sukladno prethodno stečenom znanju, tj. odabranoj trenutnoj razini poznavanja programiranja, potrebno instalirati željene programe koji su opisani u poglavlju 4. *"Programska podrška"* i pokrenuti željene primjere programa koji se nalaze u poglavljima 5, 6, 7 i 8. Primjeri programa također su priloženi i na DVD-u uz komplet.

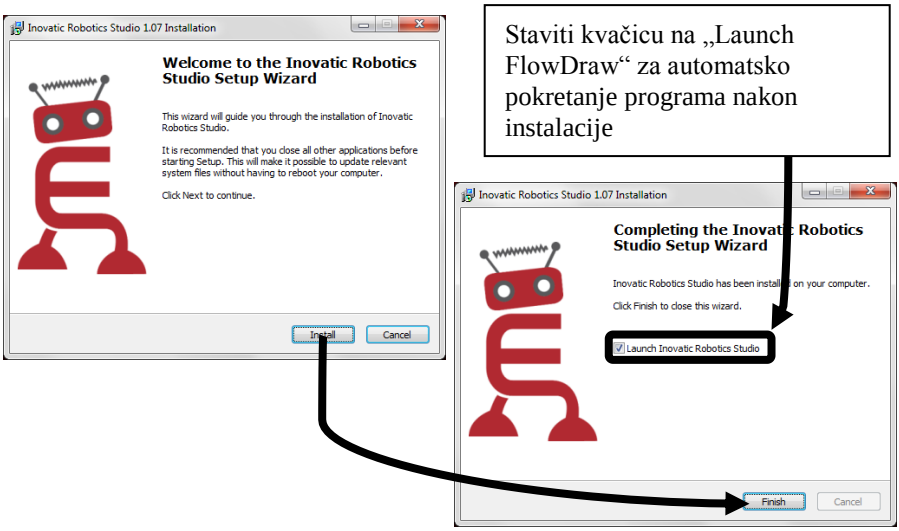
4.1 *FlawDraw*

FlowDraw je grafički softver za programiranje E-bot i EMoRo sučelja. FlowDraw nasljednik je Inovatic GUI-a. Služi za pokretanje programa na EMoRo sučelju/robotu. Za potrebe testiranja postoji simulacija u kojoj možemo pregledavati napravljene programe bez spajanja sučelja.

4.1.1 Instalacija

Pokrenuti instalaciju programa → Install → Finish

Prikaz tijeka instalacije

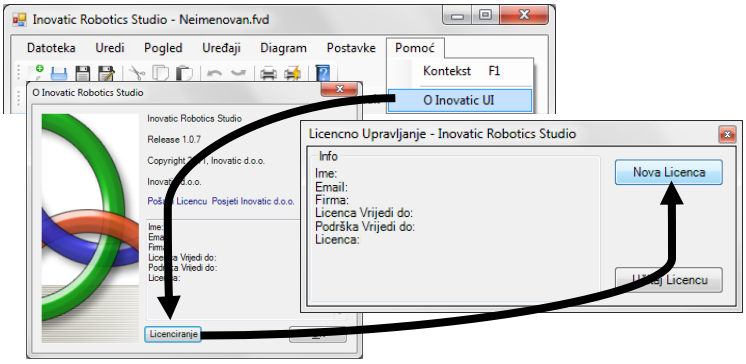


4.1.2 Registracija

Pokrenuti program → Pomoć → O Inovatic UI → Licenciranje → Nova Licenca → potrebno snimiti licencu u željenu mapu u definiranom obliku (npr. „OS Mate Lovrak Zagreb.lic“ ili „Pero Perić Osijek.lic“) i poslati sa podacima za registraciju (ime i prezime ili naziv ustanove, te hardware lock – naziv hardverskog dijela računala za koji se veže program, može biti čvrsti disk, bios, čak i USB flash što omogućava prijenos programa na drugo računalo pod istom licencom, ali uz uvjet da je USB flash uključen) na podrska@inovatic.hr.

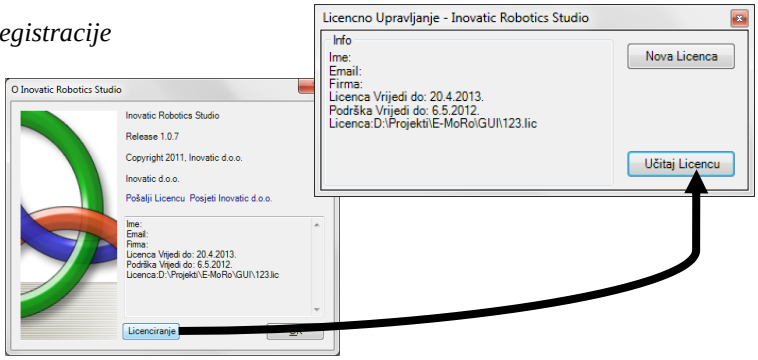
Bitno! Licenca mora biti generirana na računalu sa kojeg se pokreće program ili ako se licenca veže za USB flash (stick), flash mora biti na računalu u trenutku generiranja licence.

Prikaz tijeka registracije



Nakon primitka registrirane licence, potrebno snimiti licencu na računalo i pokrenuti program → Pomoć → O Inovatic UI → Licenciranje → Učitaj Licencu → Postaviti put do licence → OK

Prikaz tijeka registracije



Nakon pravilne registracije u prozoru „O Inovatic UI“ dobijete ispis poslanih podataka za registraciju, valjanost licence i podrške.

4.2 *FreeBasic IDE*

FBIde je besplatan, jednostavan za korištenje open source IDE za FreeBASIC prevodilac, napisan u C++. FBIde dolazi sa širokim rasponom značajki koje omogućuju jednostavno korištenje FBIde-a

Neki od njih su:

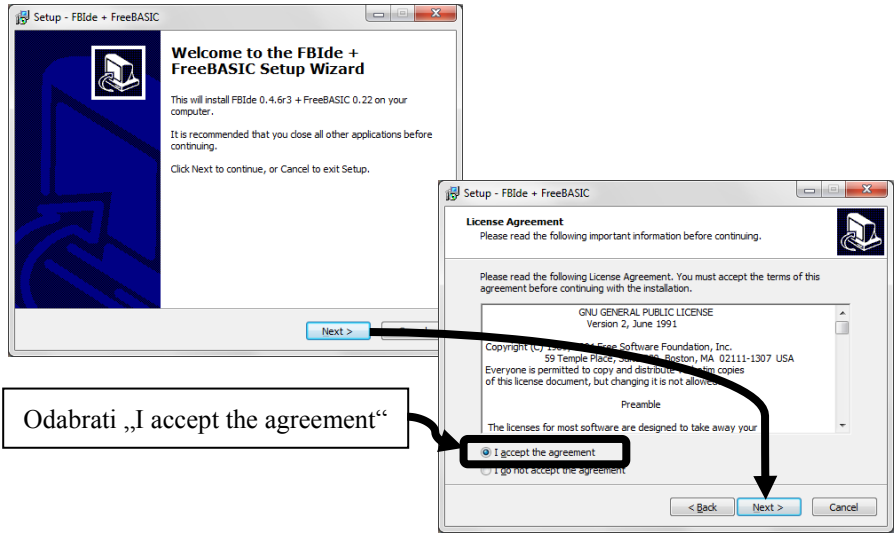
- Kartično okruženje
- Podesivo isticanje sintakse
- Izrada i provođenje izravno iz IDE
- Automatsko oblikovanje koda
- Višejezičnost
- Izvješća o pogreškama približena korisniku
- Pretraživač Funkcija i Sub
- Izvoz izvornog koda u HTML

4.2.1 Instalacija

Za instalaciju FreeBasic FBIDE-a potrebno je pokrenuti instalaciju programa sa DVD-a koji je priložen uz sučelje. Instalacija se provodi u nekoliko koraka.

Pokrenuti instalaciju programa → Next → potrebno odabrati „I accept the agreement“ → Next → „C:\FreeBasic“ → Next → Standard Installation → Next → Next → Next → Install → potrebno isključiti „Launch application“ → Finish

Prikaz tijeka instalacije



Lokacija može biti bilo gdje na tvrdom disku računala, samo je bitno zapamtiti lokaciju.

Setup - FBIde + FreeBASIC
Select Destination Location
Where should FBIde + FreeBASIC be installed?
Setup will install FBIde + FreeBASIC into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\FreeBasic
Browse...
At least 6,7 MB of free disk space is required.
< Back
Next >

Setup - FBIde + FreeBASIC
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Standard installation
Next >
Cancel

Setup - FBIde + FreeBASIC
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
FBIde
Browse...
< Back
Next >

Setup - FBIde + FreeBASIC
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing FBIde + FreeBASIC, then click Next.
Create Desktop Icon
Create Startmenu group
Associate .BAS and .BI files with FBIde
Next >
Cancel

Setup - FBIde + FreeBASIC
Ready to Install
Setup is now ready to begin installing FBIde + FreeBASIC on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\FreeBasic
Setup type:
Standard installation
Selected components:
FBIde
FreeBasic
Start Menu folder:
FBIde
< Back
Install
Cancel

Setup - FBIde + FreeBASIC
Completing the FBIde + FreeBASIC Setup Wizard
Setup has finished installing FBIde + FreeBASIC on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch application
Finish

Potrebno isključiti „kvačicu“ zbog dodatne instalacije biblioteka za sučelje

Prikaz tijeka instalacije potrebnih biblioteka

4.3 *BascomAVR (demo)*

Bascom AVR je razmjerno nov programski jezik. Njegov razvoj počeo je na kraju devedesetih godina prošlog stoljeća. Prvi put je izašao na tržište s verzijom BascomLT, zatim Bascom 8051, Bascom AVR. Bascom je kratica od BASic COMpiler, a to znači da se programi pišu u basicu, a zatim se prevode u oblik koji prepoznaje mikrokontroler. Tako Bascom možemo smatrati samo jednim dijalektom obitelji Basic-jezika (Basic, Q Basic, Visual Basic...).

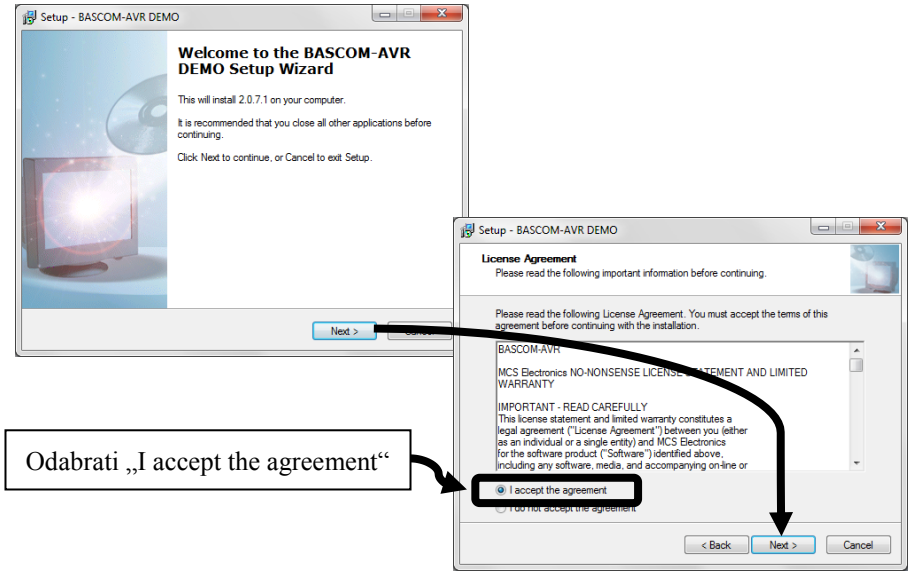
Bascom je originalan Windows prevodilac i namjena mu je stvaranje aplikacija koje će provoditi mikrokontroleri iz AVR obitelji.

4.3.1 Instalacija

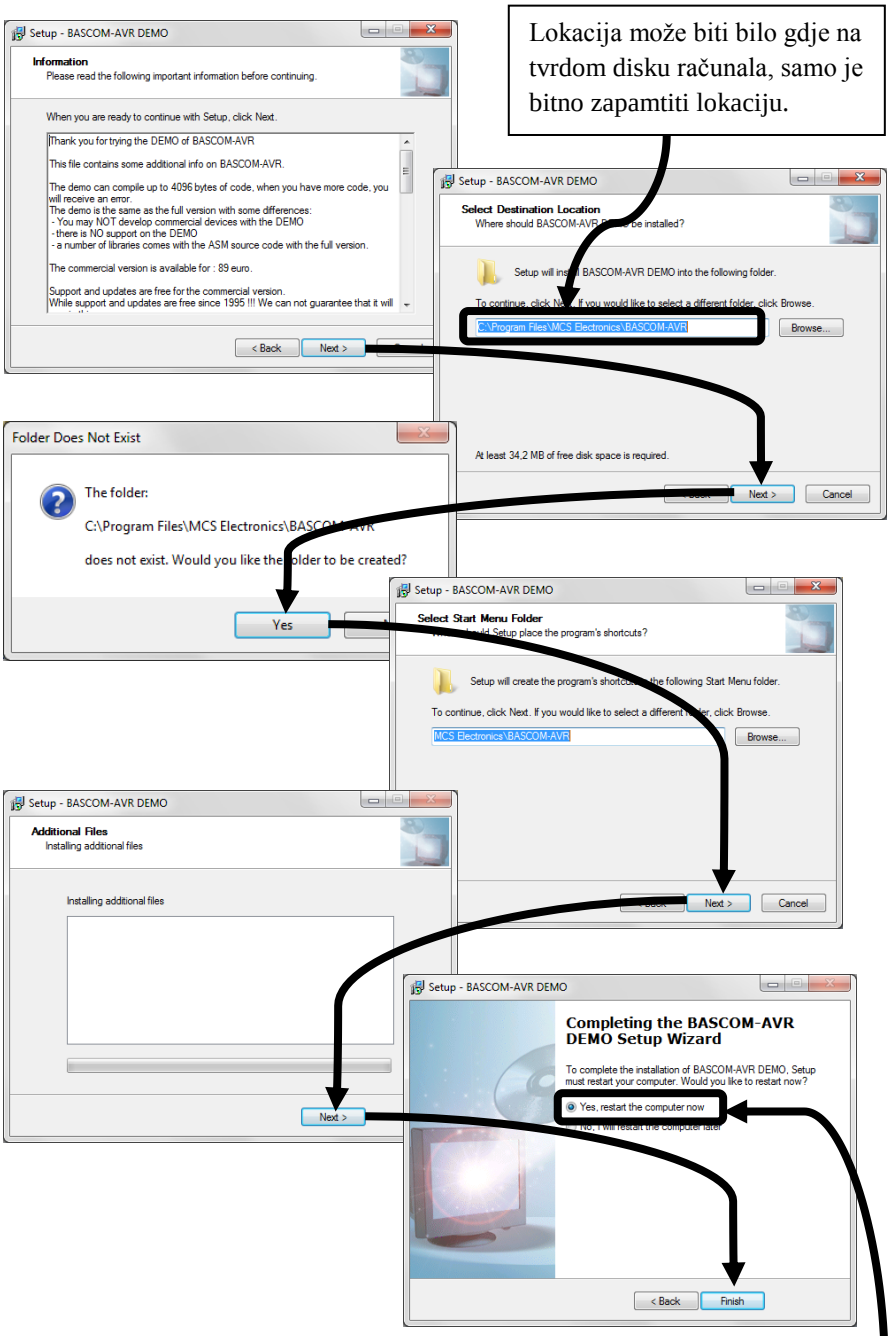
Za instalaciju demo verzije (besplatna inačica – maksimalno 4 kilobajta flash-a) BascomAVR potrebno je pokrenuti instalaciju programa s DVD-a koji je priložen uz sučelje. Instalacija se provodi u nekoliko koraka.

Pokrenuti instalaciju programa → Next → odabrati „I accept the agreement“ → Next → Next → „C:\Program Files („Program Files x86“ za 64-bitne sustave)\MSC Electronics\BASCOM-AVR“ → Next → Yes → Next → Next → odabrati„Yes, restart the computer now“ → Finish

Prikaz tijeka instalacije



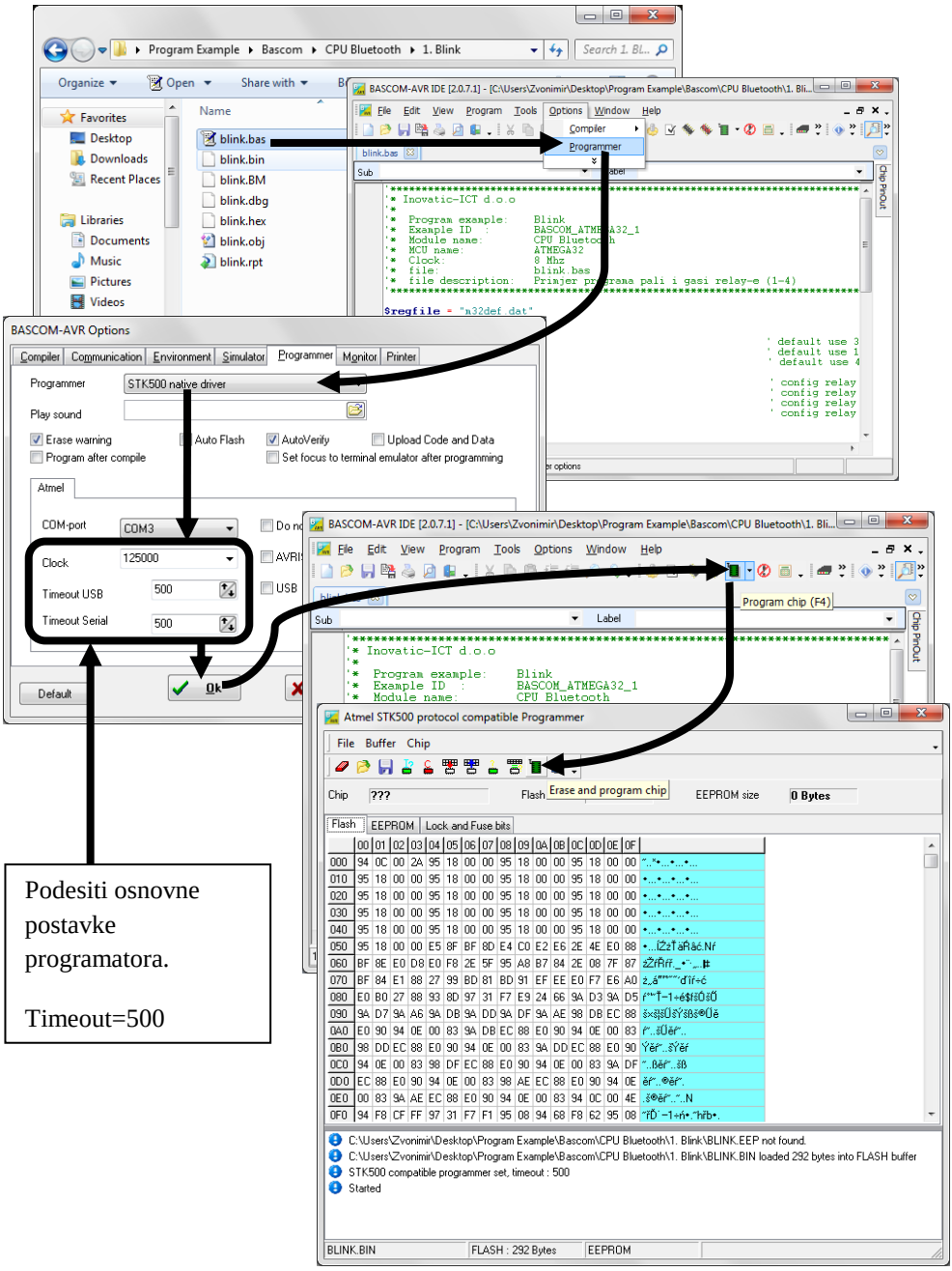
Lokacija može biti bilo gdje na tvrdom disku računala, samo je bitno zapamtiti lokaciju.



Potrebno ponovno pokrenuti računalo

4.3.2 Kratke upute za Inovatic STK500 programatora

Prikaz tijeka programiranja u BascomAVR-u sa Inovatic STK500 programatorom



Podesiti osnovne postavke programatora.

Timeout=500

4.4 *AVR Studio 4 i WinAVR*

AVR Studio 4 je IDE za ispravljanje pogrešaka u AVR softveru. AVR Studio omogućuje čip simulaciju i in-chip emulaciju za AVR obitelji mikrokontrolera. Korisničko sučelje je posebno dizajnirano za jednostavnu primjenu i daje cjelovit pregled informacija. AVR rabi isto korisničko sučelje i za simulaciju i za emulaciju, te time pruža eksponencijalnu krivulju učenja.

AVR Studio omogućuje:

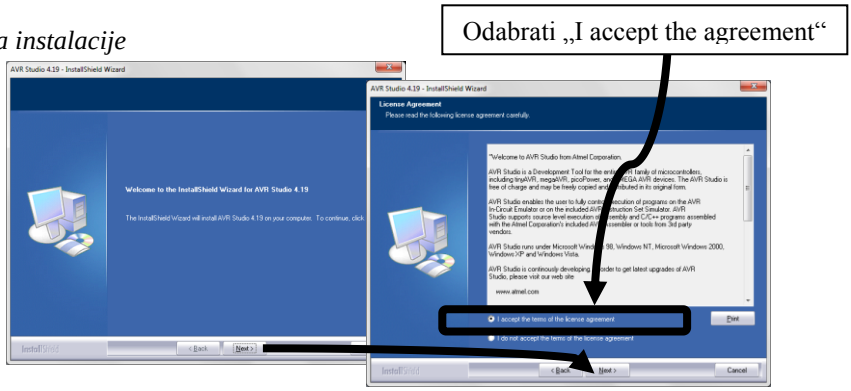
- Razvoj asemblerskih projekata za AVR mikroprocesore
  - Kreiranje, editiranje asemblerskih programa
  - asembliranje
- Emulaciju (simulaciju) izvođenja asemblerskog programa, tijekom simuliranog izvođenja (izvršavanja) asembliranog koda AVR Studio prikazuje :
  - tijek izvođenja
  - stanje registara (sadržaj)
  - Stanje i sadržaj memorije
  - Stanje i sadržaj ulaza i izlaza
  - Statusni registar
  - Registar - programski brojač
  - Registar - pokazivač adrese stoga
  - Adresne registre x,y,z

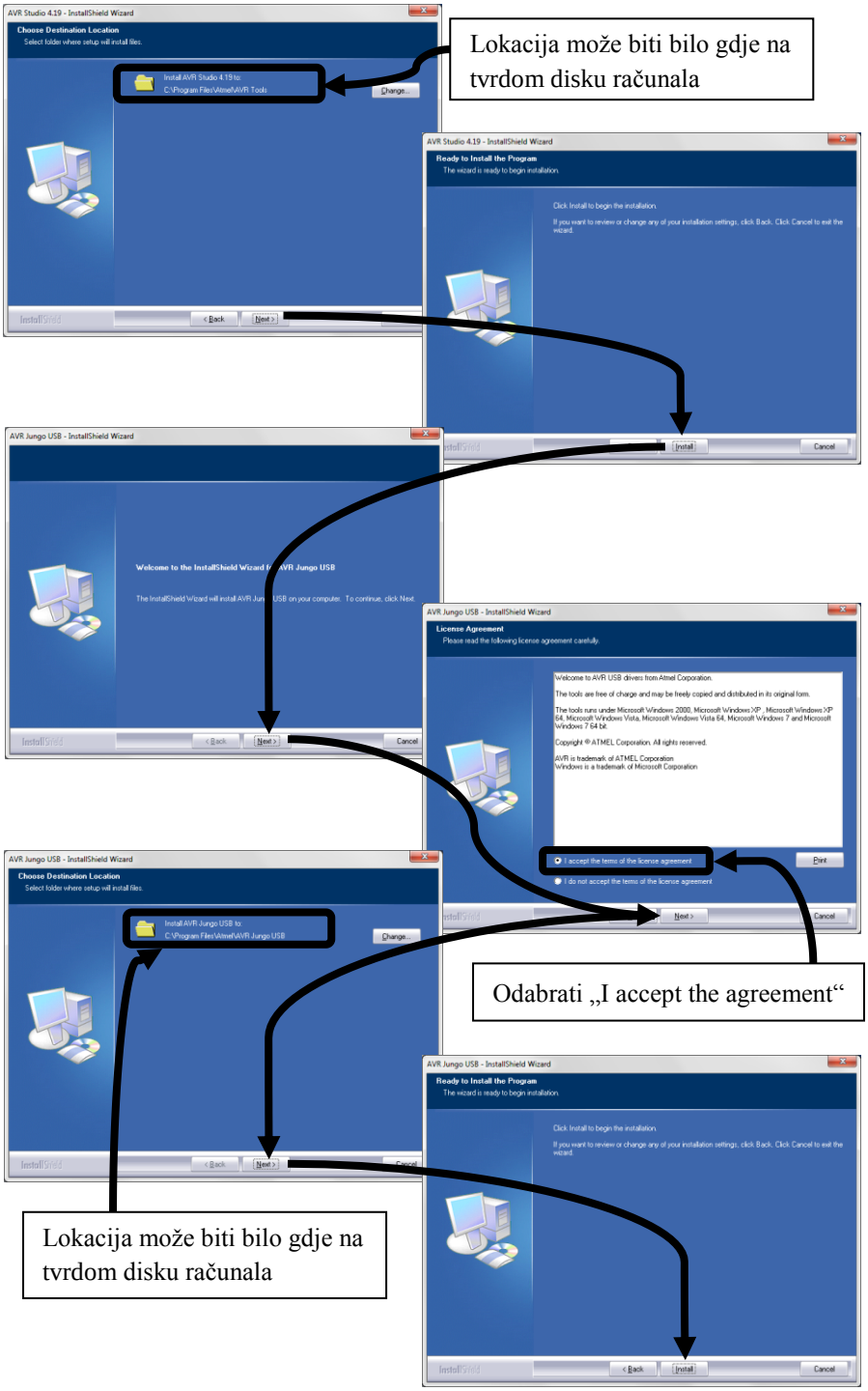
4.4.1 Instalacija

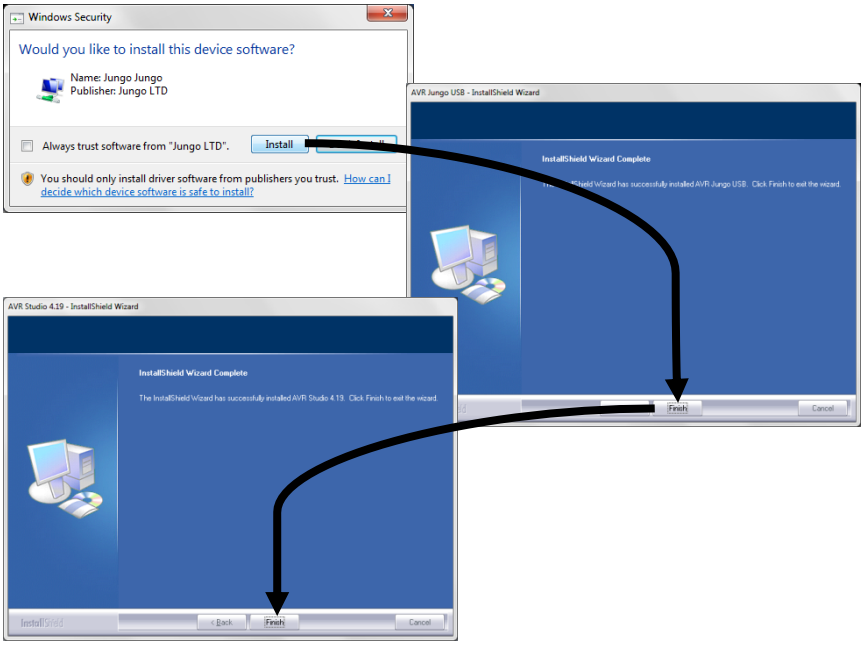
Za instalaciju AVR Studio 4 potrebno je pokrenuti instalaciju programa s DVD-a koji je priložen uz sučelje. Instalacija se provodi u nekoliko koraka.

Pokrenuti instalaciju programa → Next → odabrati „I accept the agreement“ → Next → „C:\Program Files („Program Files x86“ za 64-bitne sustave)\Atmel\AVR Tools“ → Next → Install → Next → odabrati „I accept the agreement“ → Next → „C:\Program Files\Atmel\AVR Jungo USB“ → Next → Install → Install → Finish → Finish

Prikaz tijeka instalacije




Lokacija može biti bilo gdje na tvrdom disku računala
Odabrati „I accept the agreement“
Lokacija može biti bilo gdje na tvrdom disku računala

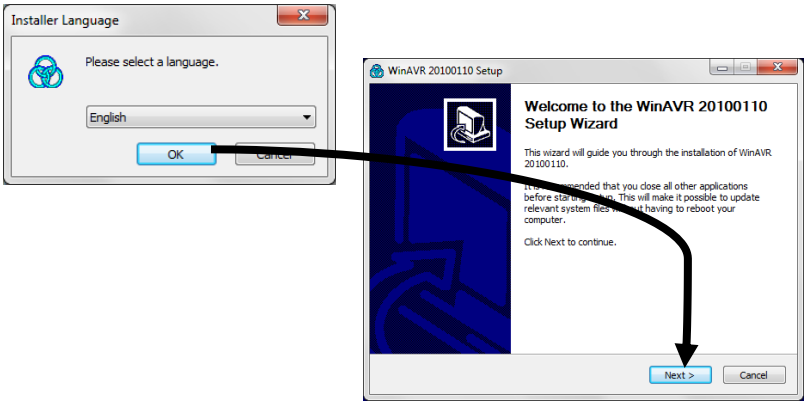


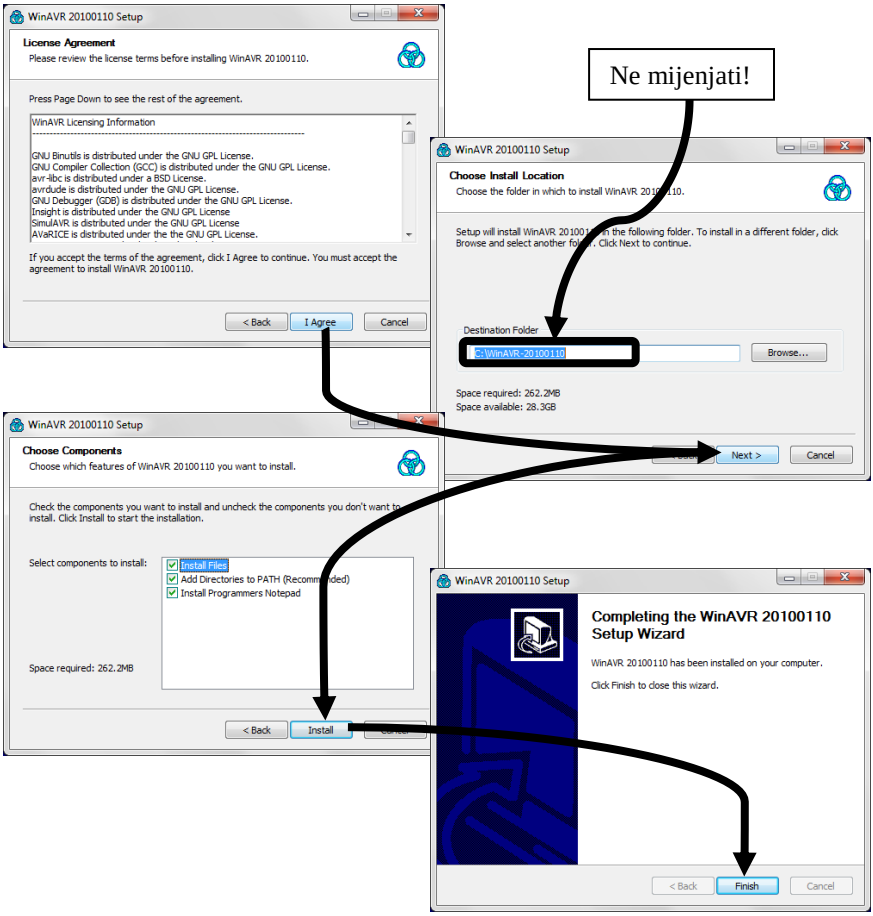
Nakon instalacije programa AVR Studio 4 potrebno je pokrenuti instalaciju WinAVR-a s DVD-a.

WinAVR je skup izvršnih, open source razvojnih alata za Atmel AVR serije RISC mikroračunala na Windows operativnom sustavu. Sadržava sve alate za razvoj na AVR-u. To uključuje AVR-gcc (prevodilac za C i C + +), avrdude (programer), AVR-gdb (debugger).

Pokrenuti instalaciju programa → odabrati jezik instalacije → OK → Next → I agree → „C:\WinAVR-20100110“ → Next → Install → Finish

Prikaz tijeka instalacije

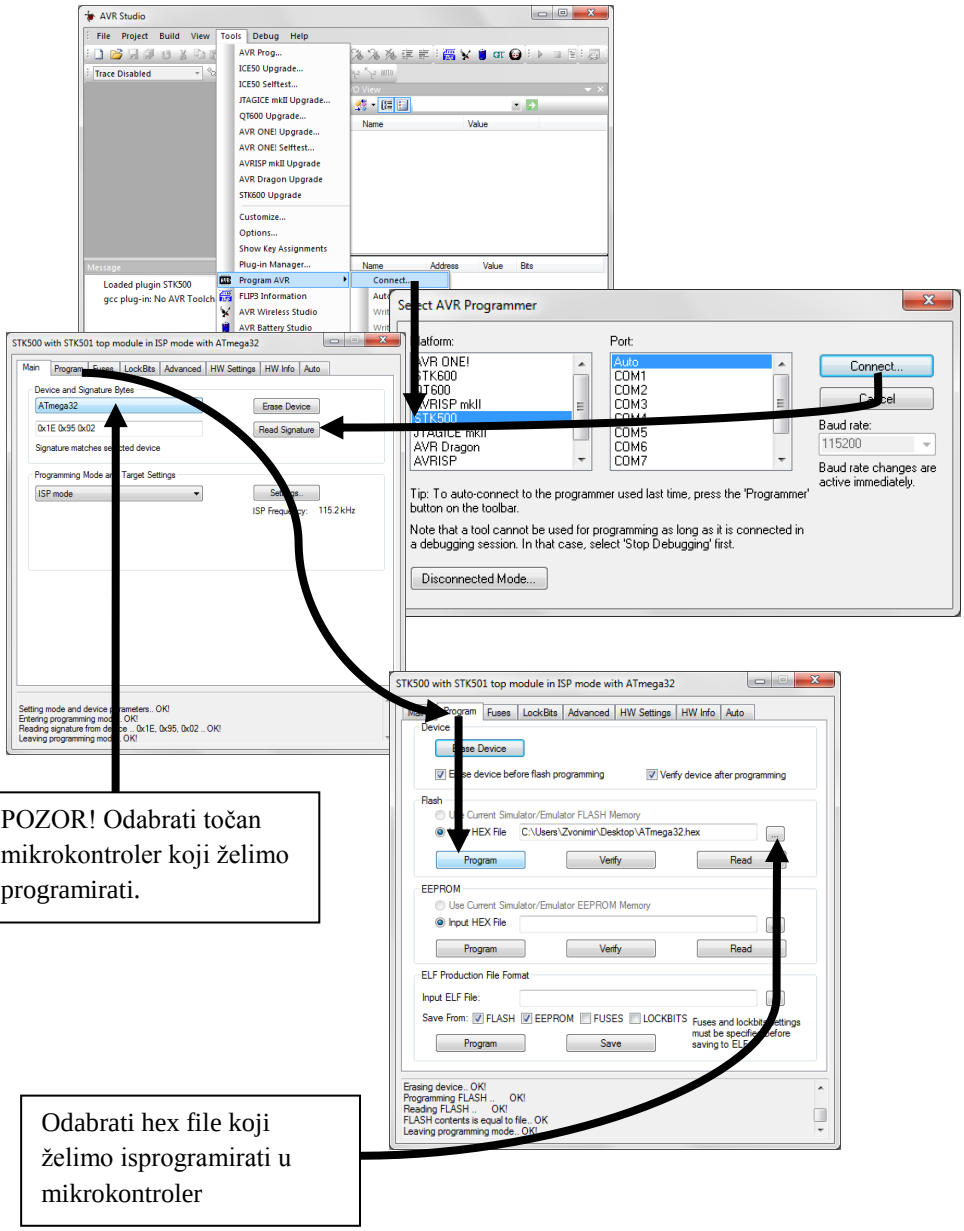



WinAVR 20100110 Setup
License Agreement
Please review the license terms before installing WinAVR 20100110.
Press Page Down to see the rest of the agreement.
WinAVR Licensing Information
GNU Binutils is distributed under the GNU GPL License.
GNU Compiler Collection (GCC) is distributed under the GNU GPL License.
avr-libc is distributed under a BSD License.
avrdude is distributed under the GNU GPL License.
GNU Debugger (GDB) is distributed under the GNU GPL License.
Insight is distributed under the GNU GPL License
SimulAVR is distributed under the GNU GPL License
AVaRICE is distributed under the the GNU GPL License.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install WinAVR 20100110.
< Back
I Agree
Cancel
Ne mijenjati!
WinAVR 20100110 Setup
Choose Install Location
Choose the folder in which to install WinAVR 20100110.
Setup will install WinAVR 20100110 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\WinAVR-20100110
Browse...
Space required: 262.2MB
Space available: 28.3GB
Next >
Cancel
WinAVR 20100110 Setup
Choose Components
Choose which features of WinAVR 20100110 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
Install Files
Add Directories to PATH (Recommended)
Install Programmers Notepad
Space required: 262.2MB
< Back
Install
WinAVR 20100110 Setup
Completing the WinAVR 20100110 Setup Wizard
WinAVR 20100110 has been installed on your computer.
Click Finish to close this wizard.
< Back
Finish
Cancel

Prikaz tijeka instalacije potrebnih biblioteka

4.4.2 Kratke upute za Inovatic STK500 programatora

Prikaz tijeka programiranja u AVR Studio-u sa Inovatic STK500 programatorom



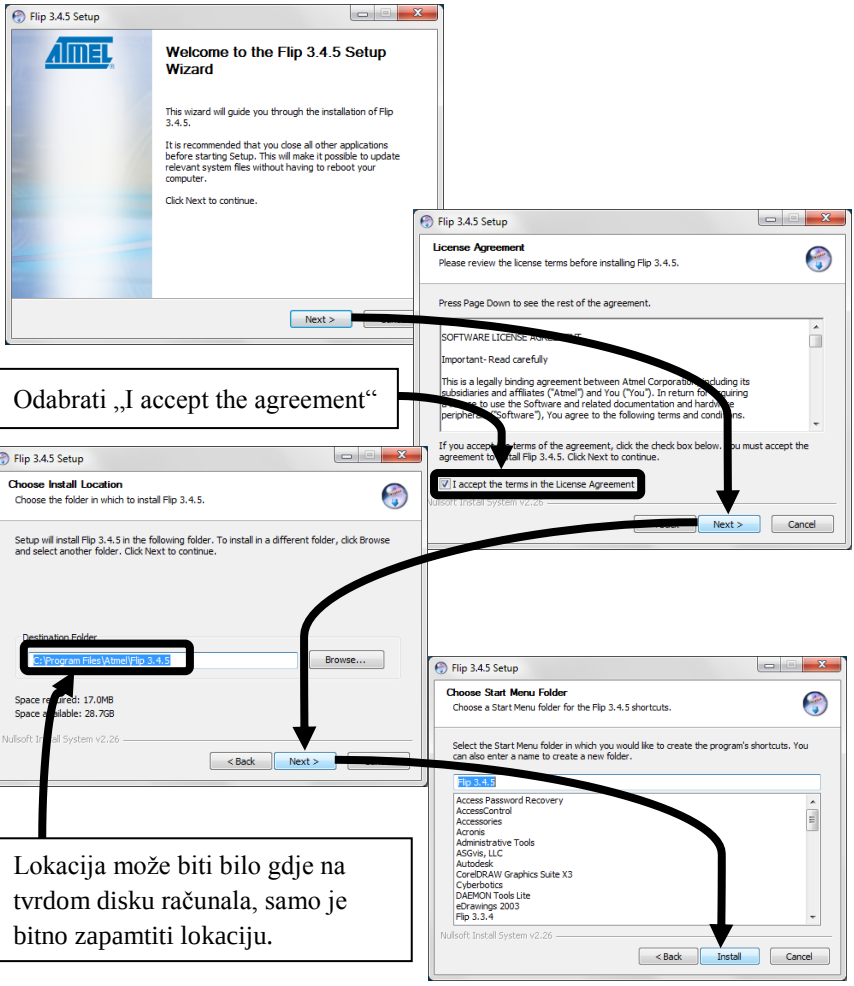
4.5 *Atmel Flip*

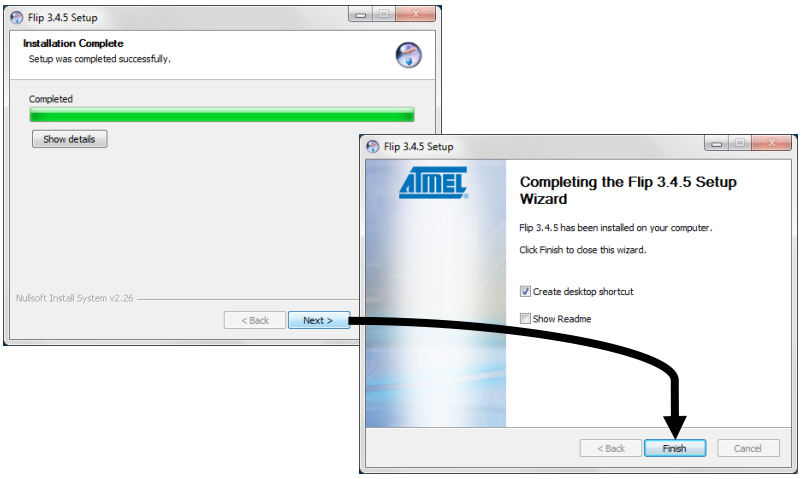
FLIP (fleksibilan in-system programator) je besplatan uslužni program koji vam omogućuje programiranje Flash-a Atmel mikrokontrolera izravno preko USB-a (bootloader).

4.5.1 Instalacija

Pokrenuti instalaciju programa → Next → odabrati „I accept the terms in the Licence Agreement → Next → „C:\Program Files („Program Files x86“ za 64-bitne sustave)\Atmel\Flip 3.4.5“ → Next → Install → Next → Finish

Prikaz tijeka instalacije





4.5.2 Instalacija Bootloader upravljačkih programa za USB sučelje

Postavljanje USB sučelja u Bootloader način rada:

1. Spojiti sučelje s računalom
2. Pritisnuti i držati tipku Reset
3. Pritisnuti tipku HWB (dalje držati Reset)
4. Pustiti tipku Reset
5. Pustiti tipku HWB

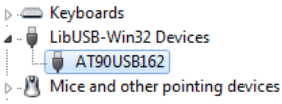
Otvoriti Device Manager (pritisnuti zajedno na tipkovnici znak „Windows“ (između Ctrl i Alt) i slovo „R“ → upisati „devmgmt.msc“ (bez navodnika) → OK) → desni klik na „*AT90USB 162 DFU*“ →

<u>Windows 7 i Vista</u>

→ Update Driver Software → Browse my computer for driver software → Browse → „C:\Program Files („Program Files x86“ za 64-bitne sustave)\Atmel\Flip 3.4.5“ → OK → Next → Install this driver software anyway → Close

<u>Windows XP</u>

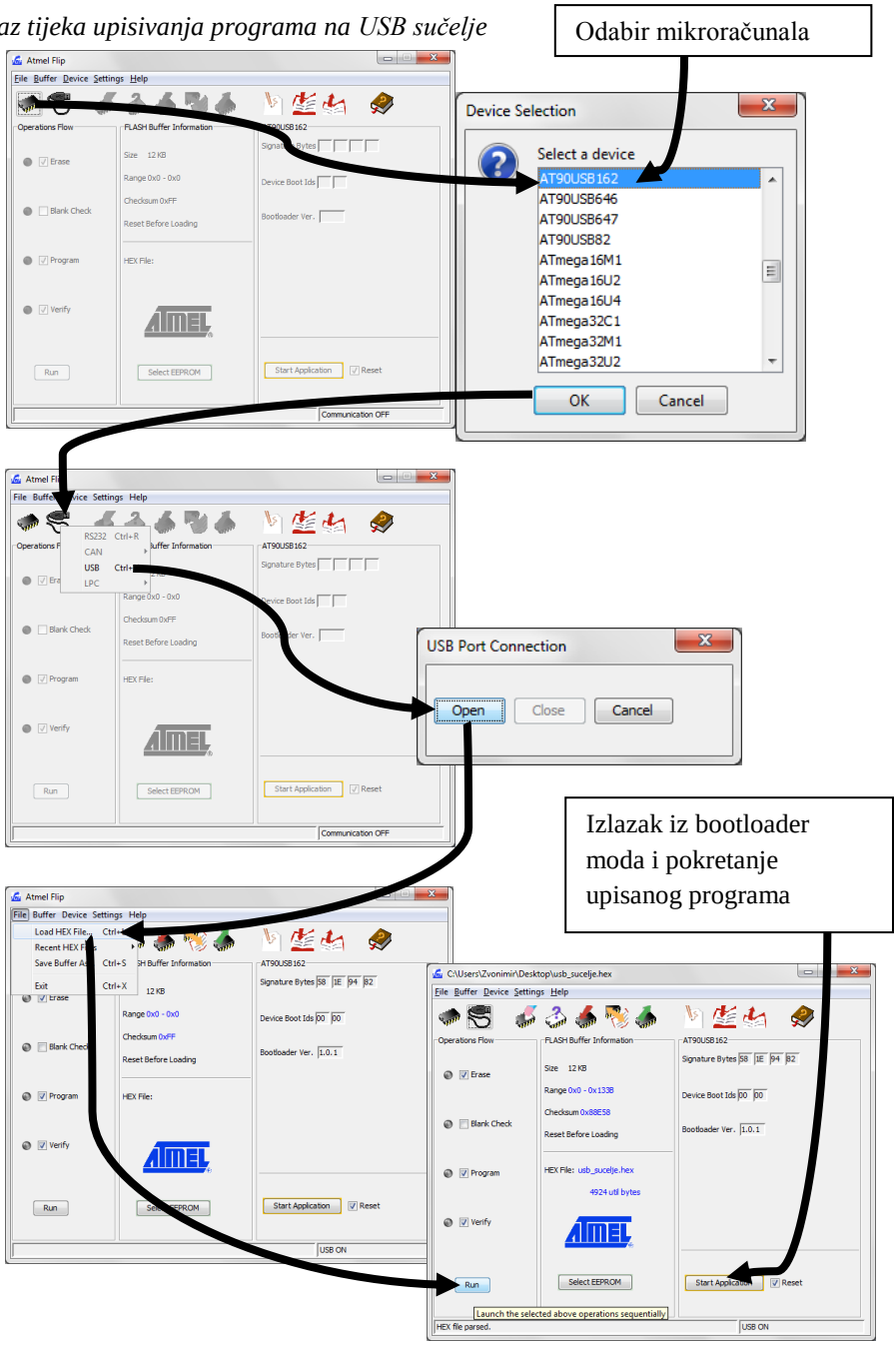
→ Update Driver → No, not this time → Install from specific location (Advanced) → Browse → „C:\Program Files („Program Files x86“ za 64-bitne sustave)\Atmel\Flip 3.4.5“ → OK → Next → Continue Anyway → Finish



Slika 73. Prikaz ispravno instaliranih upravljačkih programa

4.5.3 *Kratke upute za korištenje*

Prikaz tijeka upisivanja programa na USB sučelje



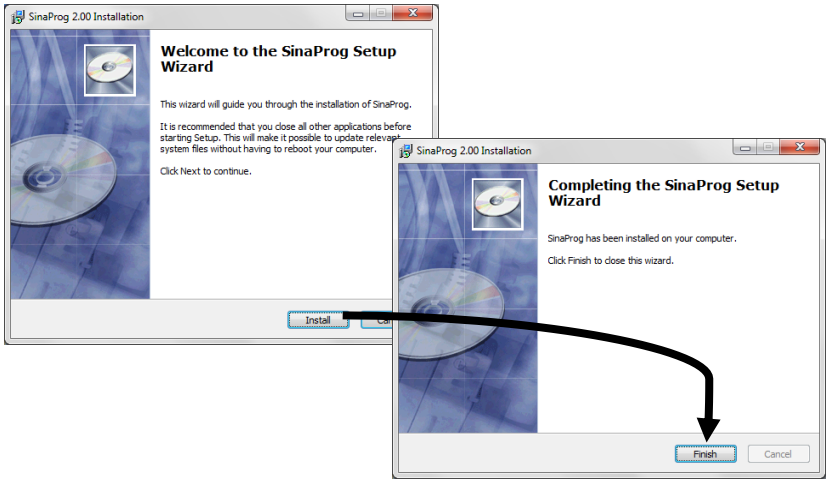
4.6 *SinaProg*

SinaProg je jednostavan program koji prepoznaje Inovatic STK500v2 u HID načinu rada i omogućuje najjednostavnije upisivanje programa (.hex) u mikroračunalo. Sinaprog prepoznaje Inovatic STK500 i u normalnom načinu rada za koji je potrebna instalacija upravljačkih programa, te skidanje kratkospojnika HID. Opširnije upute za instalaciju upravljačkih programa i podešavanje COM porta za programator nalaze se u poglavlju „*9.1.1 Instalacija upravljačkih programa*“

4.6.1 Instalacija

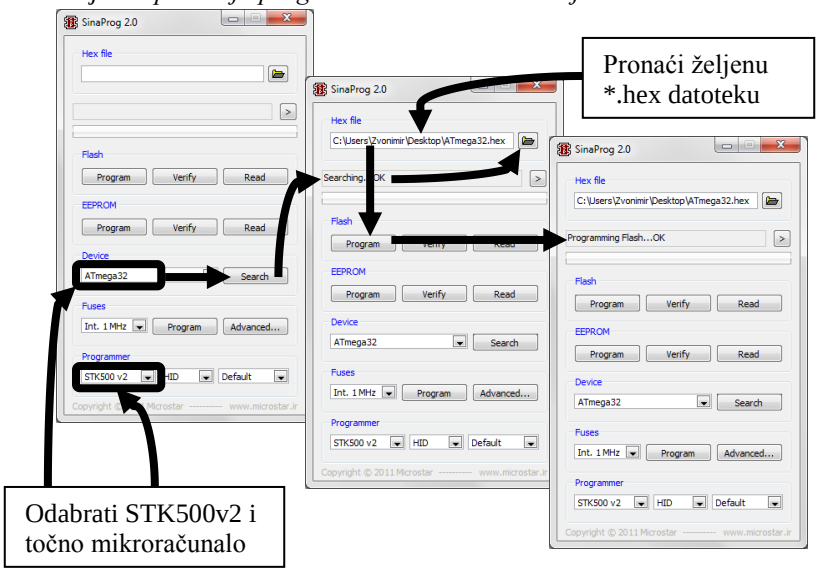
Pokrenuti instalaciju programa → Install → Finish

Prikaz tijeka instalacije

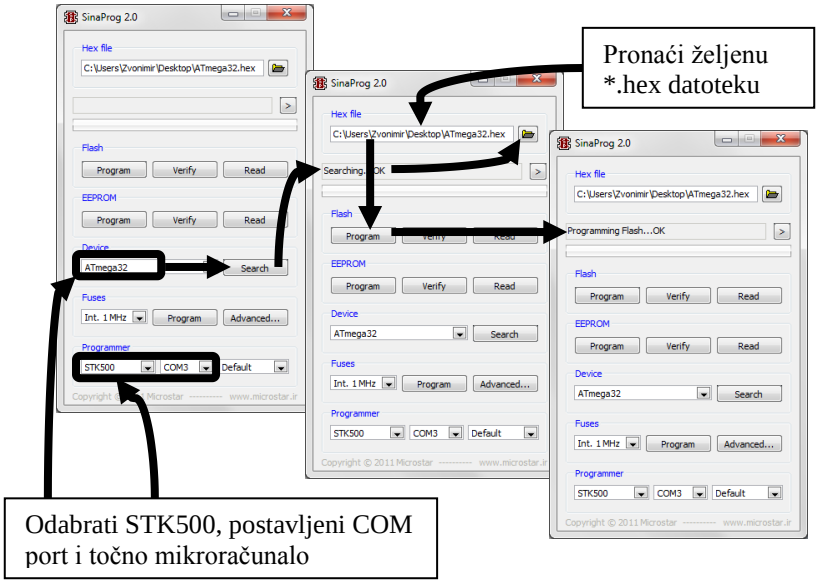


4.6.2 *Kratke upute za korištenje s Inovatic STK500 programatorom*

Prikaz tijeka upisivanja programa na CPU i BT sučelje HID način rada[26]



Prikaz tijeka upisivanja programa na CPU i BT sučelje u normanom načinu rada[27]



[26] HID način rada objašnjen je u poglavlju „*9.1.2 HID i Slow SCK način rada*“

[27] Instalacija upravljačkih programa i podešavanje COM porta objašnjeni su u poglavlju „*9.1.1 Instalacija upravljačkih programa*“

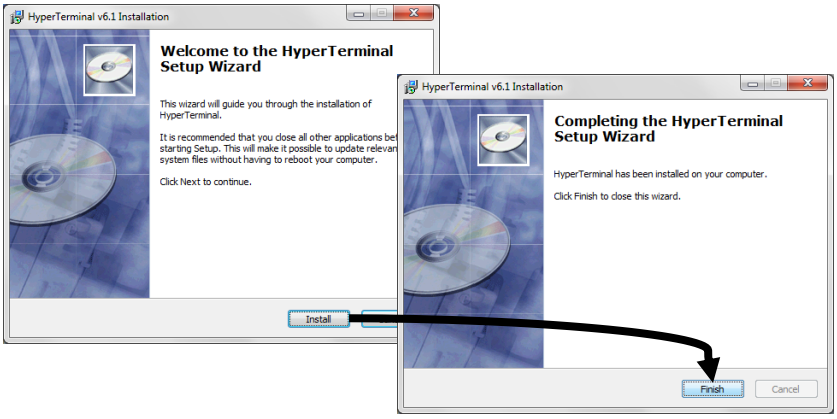
4.7 *HyperTerminal*

HyperTerminal je aplikacija koja dolazi sa sustavom Windows XP i omogućuje spojanje računala s drugim uređajima ili udaljenim računalima. HyperTerminal koristimo za komunikaciju CPU pro i BT pro sučelja s računalom. HyperTerminal dolazi sa Windows XP operativnim sustavom, ali nema ga na sustavima Vista i 7, pa ga treba naknadno dodati.

4.7.1 Instalacija

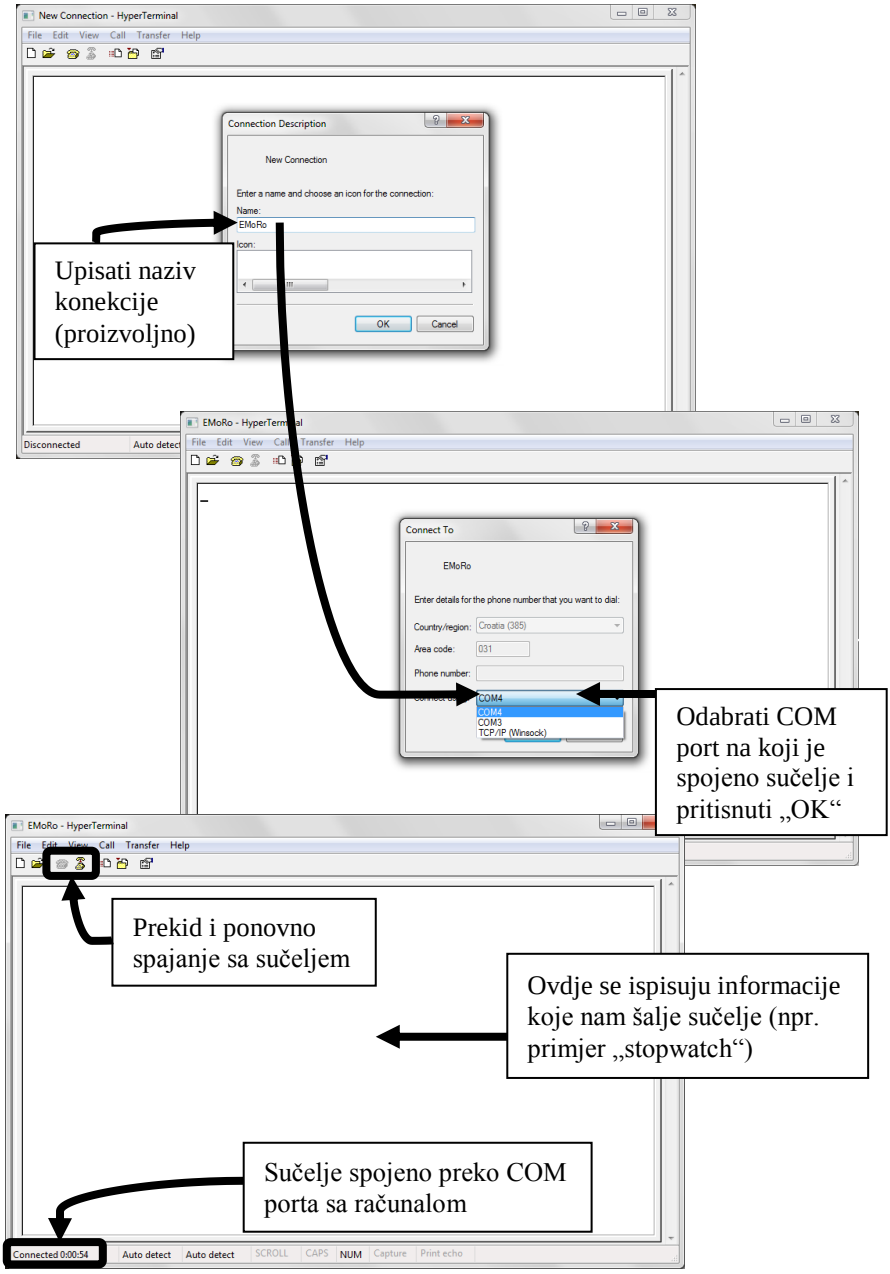
Pokrenuti instalaciju programa → Install → Finish

Prikaz tijeka instalacije



4.7.2 Kratke upute za korištenje

Prikaz tijeka spajanja EMoRo sučelja sa HyperTerminal-om



FlowDraw

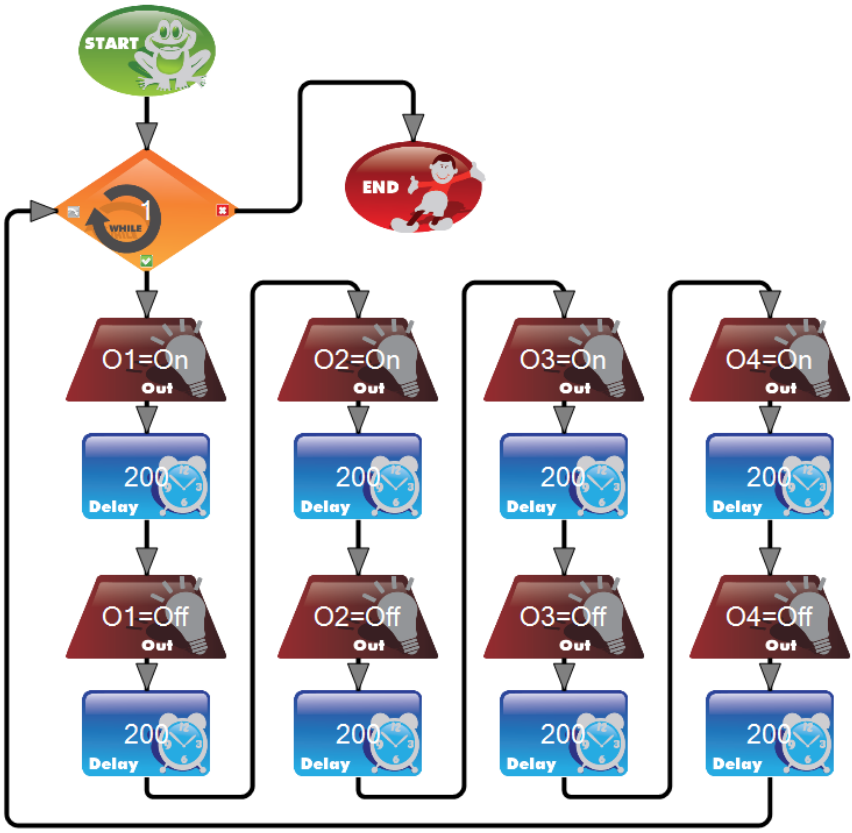
5

5. *FLOWDRAW*

Primjer 1

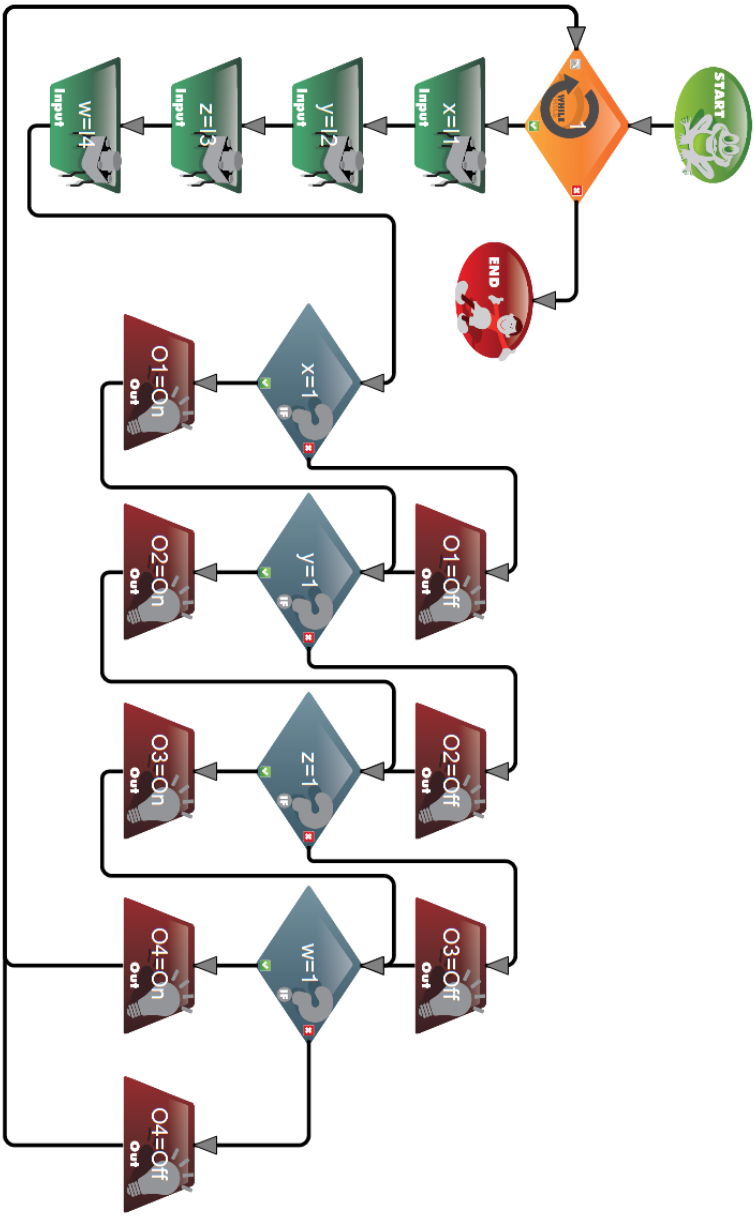
U ovom primjeru upoznajemo relejne izlaze EMoRo sučelja. Svaki relejni izlaz ima LED (svjetleću diodu) koja označava kad je izlaz aktivan. Korištenje izlaza 1 – 4 prikazat ćemo na primjeru „trćećeg svjetla“.

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **While** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza ponavlja sve dok ne zaustavimo program.



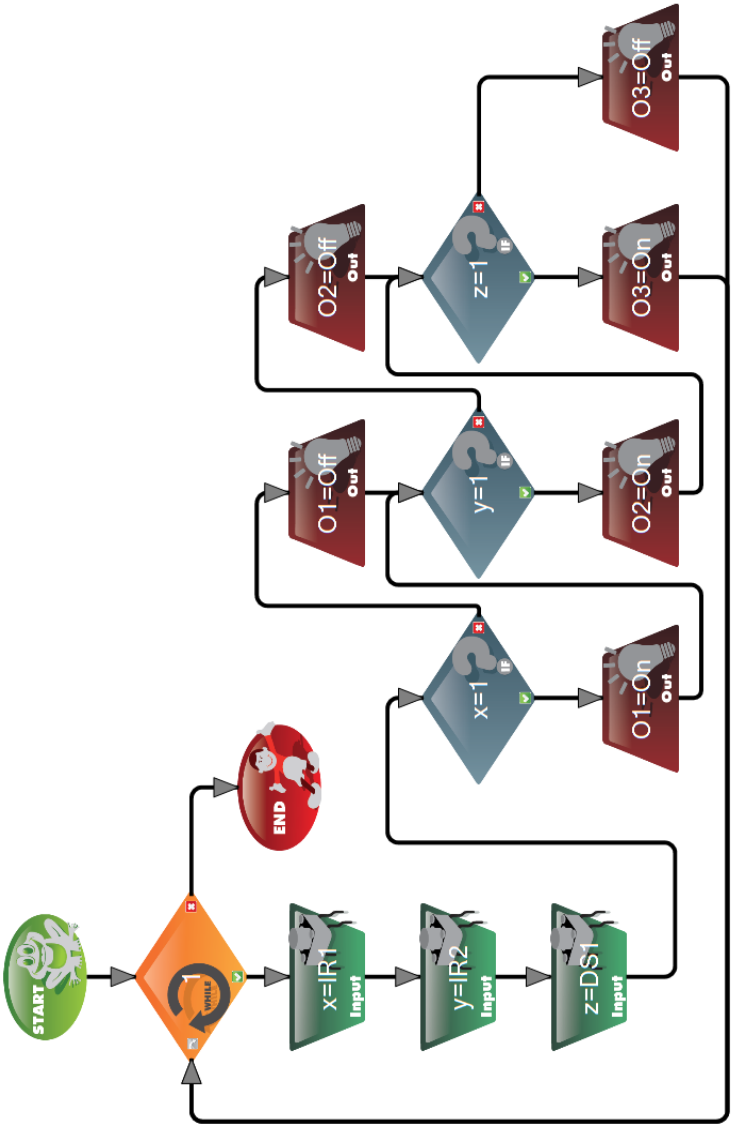
Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.



Primjer 3

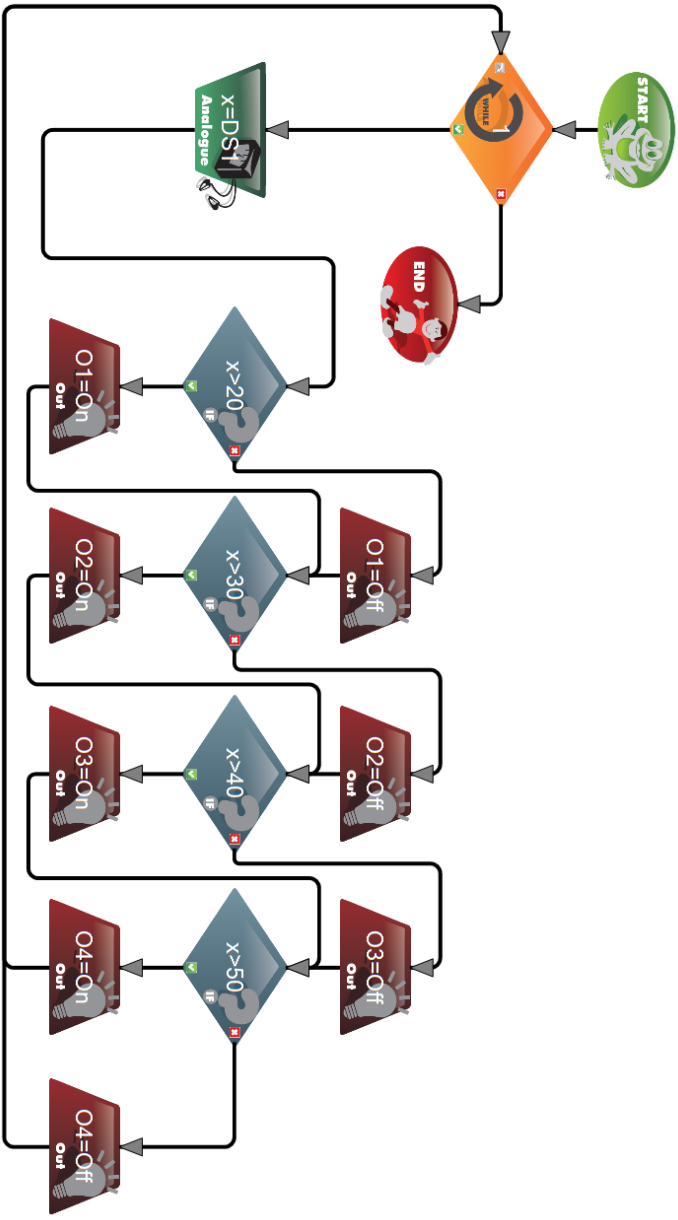
Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se Izlaz_3.



Primjer 4

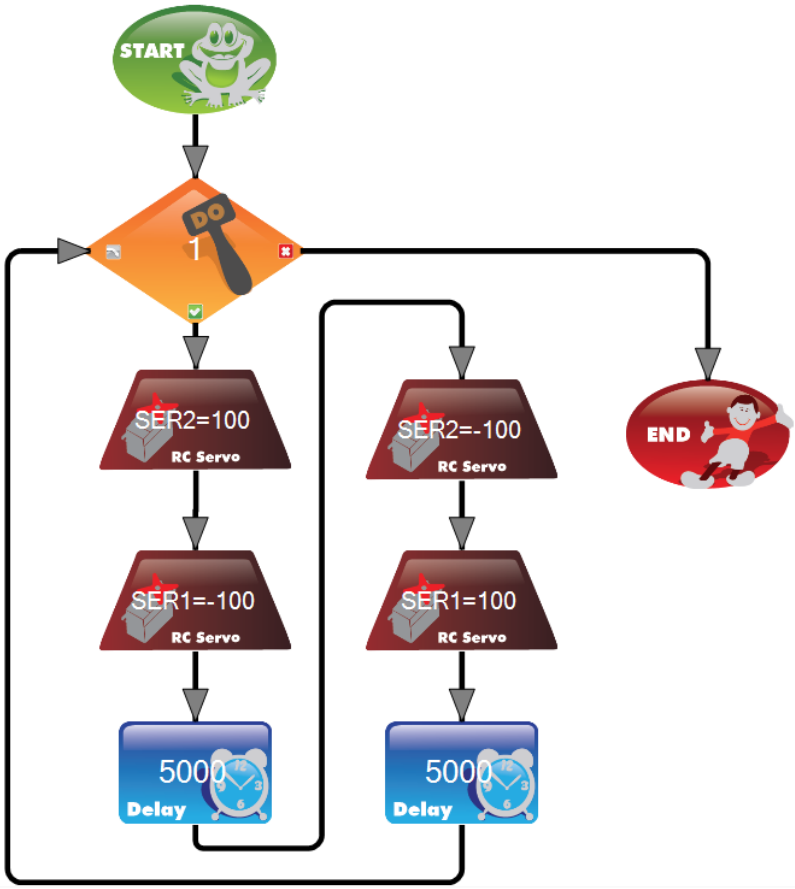
Testiranje senzora udaljenosti provodi se koristeći sljedeći primjer. Primjer se sastoji od čitanja analogne vrijednosti senzora udaljenosti koji je spojen na ulaz DS1, te ovisno o očitanoj vrijednosti aktivira izlaze 1 – 4.

Napomena! USB sučelje ne podržava čitanje analogne vrijednosti, jer ne sadržava analogno-digitalni pretvarač.

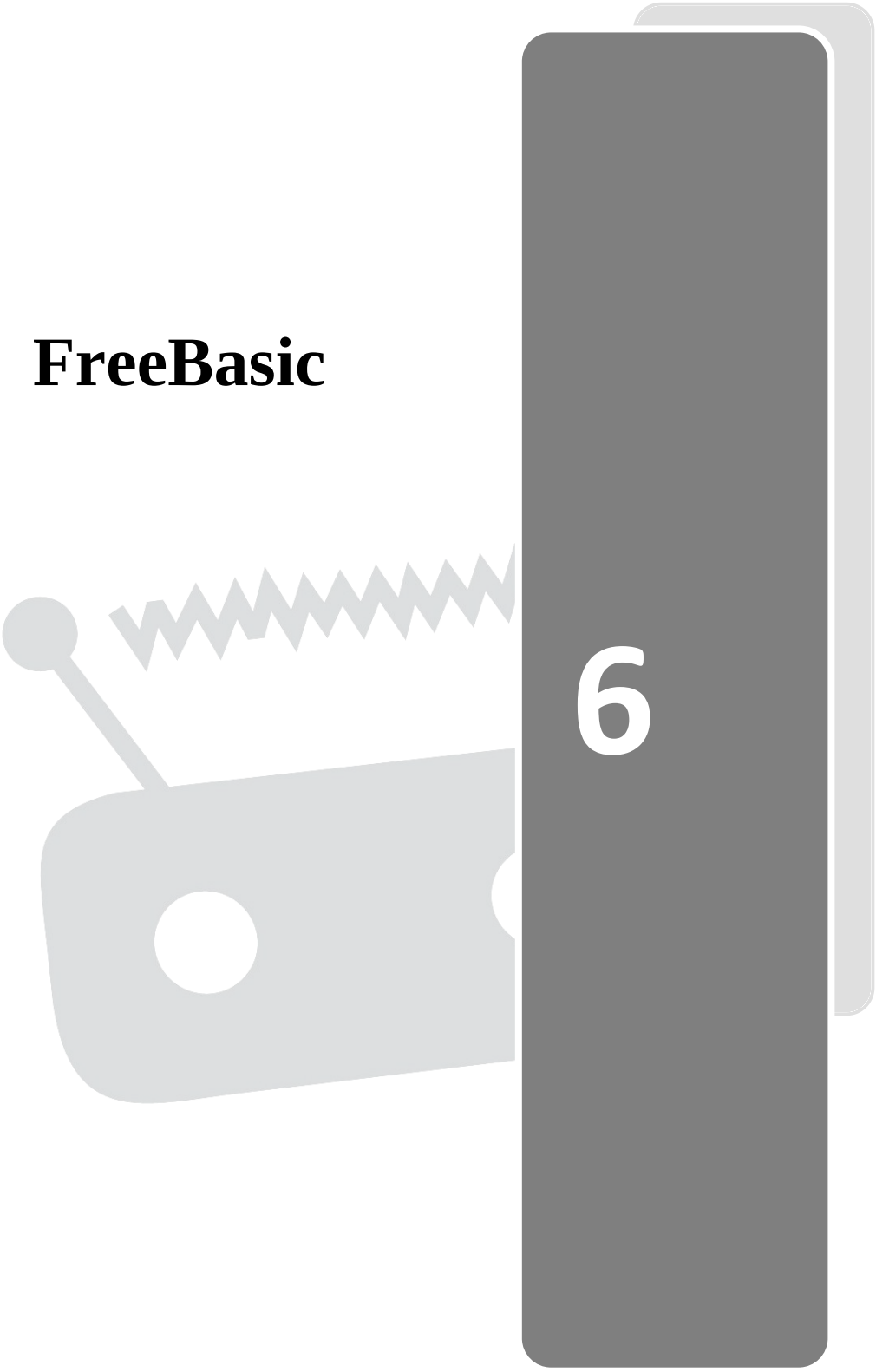


Primjer 5

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).



FreeBasic



6. FREEBASIC

Primjer 1

U ovom primjeru upoznajemo relejne izlaze EMoRo sučelja. Svaki relejni izlaz ima LED (svjetleću diodu) koja označava kad je izlaz aktivan. Korištenje izlaza 1 – 4 prikazat ćemo na primjeru „trčećeg svjetla“.

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **Do-Loop** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza ponavlja sve dok ne zaustavimo program. (Ctrl+c)

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka           '
#Include "fbgfx.bi"

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem     '

Do
   Relay1On()                       ' ukljuci relay 1                     '
   Sleep 200                        ' cekaj 200 mS                        '
   Relay1Off()                      ' iskljuci relay 1                    '
   Sleep 200                        ' cekaj 0.2 S                         '

   Relay2On()                       ' ukljuci relay 2                     '
   Sleep 200                        ' cekaj 200 mS                        '
   Relay2Off()                      ' iskljuci relay 2                    '
   Sleep 200                        ' cekaj 200 mS                        '

   Relay3On()                       ' ukljuci relay 3                     '
   Sleep 200                        ' cekaj 200 mS                        '
   Relay3Off()                      ' iskljuci relay 3                    '
   Sleep 200                        ' cekaj 200 mS                        '

   Relay4On()                       ' ukljuci relay 4                     '
   Sleep 200                        ' cekaj 200 mS                        '
   Relay4Off()                      ' iskljuci relay 4                    '
   Sleep 200                        ' cekaj 200 mS                        '
Loop
End
```

Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka            '
#Include "fbgfx.bi"

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem      '

Do
   If Input1() = 1 Then             ' pritisnuta tipka 1?                  '
      Relay1On()                    ' ukljuci relej 1                      '
   Else
      Relay1Off()                   ' iskljuci relej 1                     '
   End If

   If Input2() = 1 Then             ' pritisnuta tipka 2?                  '
      Relay2On()                    ' ukljuci relej 2                      '
   Else
      Relay2Off()                   ' iskljuci relej 2                     '
   End If

   If Input3() = 1 Then             ' pritisnuta tipka 3?                  '
      Relay3On()                    ' ukljuci relej 3                      '
   Else
      Relay3Off()                   ' iskljuci relej 3                     '
   End If

   If Input4() = 1 Then             ' pritisnuta tipka 4?                  '
      Relay4On()                    ' ukljuci relej 4                      '
   Else
      Relay4Off()                   ' iskljuci relej 4                     '
   End If
Loop
End
```

Primjer 3

Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se Izlaz_3.

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka            '
#Include "fbgfx.bi"

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem      '

Do
   If InputIR1() = 1 Then           ' IR1 aktivan?                         '
      Relay1On()                    ' ukljuci Relay 1                      '
   Else
      Relay1Off()                   ' iskljuci Relay 1                     '
   End If

   If InputIR2 = 1 Then             ' IR2 aktivan?                         '
      Relay2On()                    ' ukljuci Relay 2                      '
   Else
      Relay2Off()                   ' iskljuci Relay 2                     '
   End If

   If InputDS1() = 1 Then           ' DS1 aktivan?                         '
      Relay3On()                    ' ukljuci Relay 3                      '
   Else
      Relay3Off()                   ' iskljuci Relay 3                     '
   End If

   Sleep 100                        ' provjeti stanje IR-a svakih 100 mS   '
Loop
End
```

Primjer 4

Primjer programa inicijalizira analogno-digitalni pretvarač na ulazu DS1 na kojemu je spojen senzor udaljenosti, te ispisuje vrijednost udaljenosti u intervalu [0-1023] i napon na DS1 ulazu u intervalu [0-5V].

Napomena! USB sučelje ne podržava čitanje analogne vrijednosti, jer ne sadržava analogno-digitalni pretvarač.

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka            '
#Include "fbgfx.bi"

Dim distance As Unsigned Integer
Dim voltage As Double

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem      '
InitAdc(DS1)                        ' pokreni ADC na ulazu DS1             '

Do
   distance = GetAdc(DS1)           ' citaj senzor udaljenosti (DS1)       '
   voltage = (5*distance)/1024      ' izracunaj napon na DS1 ulazu         '

   Print Using "Distance = #### , DS1 Voltage = #.## V"; distance, voltage

   Sleep                            ' provjeti stanje DS1 svakih 300 mS    '
Loop
End
```

Primjer 5

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka            '
#Include "fbgfx.bi"

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem      '

InitServo(SERVO_1)                  ' pokreni servo SERVO_1                '
InitServo(SERVO_2)                  ' pokreni servo SERVO_2                '

Do
   SetServo(SERVO_1, 1000)          ' postavi Servo 1 pulse svakih 1mS     '
   SetServo(SERVO_2, 2000)          ' postavi Servo 2 pulse svakih 2mS     '

   Sleep 5000                       ' cekaj 5 sekundi                      '

   SetServo(SERVO_1, 2000)          ' postavi Servo 1 pulse svakih 2mS     '
   SetServo(SERVO_2, 1000)          ' postavi Servo 2 pulse svakih 1mS     '

   Sleep 5000                       ' cekaj 5 sekundi                      '
Loop
End
```

BascomAVR

7

7. BASCOMAVR

7.1 *USB sučelje – AT90USB162*

Primjer 1

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **Do-Loop** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza ponavlja sve dok ne zaustavimo program.

```
$regfile = "usb162.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portb.4 = Output                 ' postavi port releja 1 kao izlaz
Config Portd.6 = Output                 ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                 ' postavi port releja 3 kao izlaz
Config Portd.2 = Output                 ' postavi port releja 4 kao izlaz

Relay_1 Alias Portb.4
Relay_2 Alias Portd.6
Relay_3 Alias Portd.4
Relay_4 Alias Portd.2

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   Relay_1 = 0                          ' ukljuci relej 1
   Waitms 200                           ' cekaj 200 mS
   Relay_1 = 1                          ' iskljuci relej 1
   Waitms 200                           ' cekaj 200 mS

   Relay_2 = 0                          ' ukljuci relej 2
   Waitms 200                           ' cekaj 200 mS
   Relay_2 = 1                          ' iskljuci relej 2
   Waitms 200                           ' cekaj 200 mS

   Relay_3 = 0                          ' ukljuci relej 3
   Waitms 200                           ' cekaj 200 mS
   Relay_3 = 1                          ' iskljuci relej 3
   Waitms 200                           ' cekaj 200 mS

   Relay_4 = 0                          ' ukljuci relej 4
   Waitms 200                           ' cekaj 200 mS
   Relay_4 = 1                          ' iskljuci relej 4
   Waitms 200                           ' cekaj 200 mS
Loop

End
```

Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.

```
$regfile = "usb162.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portb.4 = Output                 ' postavi port releja 1 kao izlaz
Config Portd.6 = Output                 ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                 ' postavi port releja 3 kao izlaz
Config Portd.2 = Output                 ' postavi port releja 4 kao izlaz

Config Portd.1 = Input                  ' postavi port ulaza 1 kao ulaz
Config Portd.3 = Input                  ' postavi port ulaza 2 kao ulaz
Config Portd.5 = Input                  ' postavi port ulaza 3 kao ulaz
Config Portb.0 = Input                  ' postavi port ulaza 4 kao ulaz

Relay_1 Alias Portb.4
Relay_2 Alias Portd.6
Relay_3 Alias Portd.4
Relay_4 Alias Portd.2

Input_1 Alias Pind.1
Input_2 Alias Pind.3
Input_3 Alias Pind.5
Input_4 Alias Pinb.0

Portd.1 = 1                             ' ukljuci pull-up otpornik (PD1)
Portd.3 = 1                             ' ukljuci pull-up otpornik (PD3)
Portd.5 = 1                             ' ukljuci pull-up otpornik (PD5)
Portb.0 = 1                             ' ukljuci pull-up otpornik (PB0)

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   If Input_1 = 1 Then                  ' ulaz 1 aktivan?
      Relay_1 = 0                       ' ukljuci relej 1
   Else
      Relay_1 = 1                       ' iskljuci relej 1
   End If

   If Input_2 = 1 Then                  ' ulaz 2 aktivan?
      Relay_2 = 0                       ' ukljuci relej 2
   Else
      Relay_2 = 1                       ' iskljuci relej 2
   End If
```

```
   If Input_3 = 1 Then                  ' ulaz 3 aktivan?
      Relay_3 = 0                       ' ukljuci relej 3
   Else
      Relay_3 = 1                       ' iskljuci relej 3
   End If

   If Input_4 = 1 Then                  ' ulaz 4 aktivan?
      Relay_4 = 0                       ' ukljuci relej 4
   Else
      Relay_4 = 1                       ' iskljuci relej 4
   End If
Loop

End
```

Primjer 3

Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se Izlaz_3.

```
$regfile = "usb162.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portb.4 = Output                ' postavi port releja 1 kao izlaz
Config Portd.6 = Output                ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                ' postavi port releja 3 kao izlaz
Config Portd.2 = Output                ' postavi port releja 4 kao izlaz

Config Portb.6 = Input                 ' postavi port IR1 kao ulaz
Config Portb.5 = Input                 ' postavi port IR2 kao ulaz
Config Portc.4 = Input                 ' postavi port DS1 kao ulaz

Relay_1 Alias Portb.4
Relay_2 Alias Portd.6
Relay_3 Alias Portd.4
Relay_4 Alias Portd.2

Input_ir1 Alias Pinb.6
Input_ir2 Alias Pinb.5
Input_ds1 Alias Pinc.4

Portb.6 = 1                            ' ukljuci pull-up otpornik (PB6)
Portb.5 = 1                            ' ukljuci pull-up otpornik (PB5)
Portc.4 = 1                            ' ukljuci pull-up otpornik (PC4)

Relay_1 = 1                            ' iskljuci relej 1
Relay_2 = 1                            ' iskljuci relej 2
Relay_3 = 1                            ' iskljuci relej 3
Relay_4 = 1                            ' iskljuci relej 4

Do
   If Input_ir1 = 0 Then               ' ulaz IR1 aktivan?
      Relay_1 = 0                      ' ukljuci relej 1
   Else
      Relay_1 = 1                      ' iskljuci relej 1
   End If

   If Input_ir2 = 0 Then               ' ulaz IR2 aktivan?
      Relay_2 = 0                      ' ukljuci relej 2
   Else
      Relay_2 = 1                      ' iskljuci relej 2
   End If
```

```
    If Input_ds1 = 0 Then                ' ulaz DS1 aktivan?
       Relay_3 = 0                       ' ukljuci relej 3
    Else
       Relay_3 = 1                       ' iskljuci relej 3
    End If
Loop

End
```

Primjer 4

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).

```
$regfile = "usb162.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portc.5 , Servo2 = Portc.6 , Reload = 10

Config Portc.5 = Output                ' postavi port SERVO_1 kao izlaz
Config Portc.6 = Output                ' postavi port SERVO_2 kao izlaz

Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 50                       ' postavi SERVO_1 (1mS puls)
   Servo(2) = 130                      ' postavi SERVO_2 (2mS puls)
   Wait 3
   Servo(1) = 130                      ' postavi SERVO_1 (2mS puls)
   Servo(2) = 50                       ' postavi SERVO_2 (1mS puls)
   Wait 3
Loop

End
```

7.2 *CPU sučelje – ATmega168*

Primjer 1

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **Do-Loop** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza ponavlja sve dok ne zaustavimo program.

```
$regfile = "m168def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portc.2 = Output                 ' postavi port releja 1 kao izlaz
Config Portc.0 = Output                 ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                 ' postavi port releja 3 kao izlaz
Config Portd.7 = Output                 ' postavi port releja 4 kao izlaz

Relay_1 Alias Portc.2
Relay_2 Alias Portc.0
Relay_3 Alias Portd.4
Relay_4 Alias Portd.7

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   Relay_1 = 0                          ' ukljuci relej 1
   Waitms 200                           ' cekaj 200 mS
   Relay_1 = 1                          ' iskljuci relej 1
   Waitms 200                           ' cekaj 200 mS

   Relay_2 = 0                          ' ukljuci relej 2
   Waitms 200                           ' cekaj 200 mS
   Relay_2 = 1                          ' iskljuci relej 2
   Waitms 200                           ' cekaj 200 mS

   Relay_3 = 0                          ' ukljuci relej 3
   Waitms 200                           ' cekaj 200 mS
   Relay_3 = 1                          ' iskljuci relej 3
   Waitms 200                           ' cekaj 200 mS

   Relay_4 = 0                          ' ukljuci relej 4
   Waitms 200                           ' cekaj 200 mS
   Relay_4 = 1                          ' iskljuci relej 4
   Waitms 200                           ' cekaj 200 mS
Loop

End
```

Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.

```
$regfile = "m168def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portc.2 = Output                 ' postavi port releja 1 kao izlaz
Config Portc.0 = Output                 ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                 ' postavi port releja 3 kao izlaz
Config Portd.7 = Output                 ' postavi port releja 4 kao izlaz

Config Portb.0 = Input                  ' postavi port ulaza 1 kao ulaz
Config Portd.2 = Input                  ' postavi port ulaza 2 kao ulaz
Config Portd.3 = Input                  ' postavi port ulaza 3 kao ulaz
Config Portc.1 = Input                  ' postavi port ulaza 4 kao ulaz

Relay_1 Alias Portc.2
Relay_2 Alias Portc.0
Relay_3 Alias Portd.4
Relay_4 Alias Portd.7

Input_1 Alias Pinb.0
Input_2 Alias Pind.2
Input_3 Alias Pind.3
Input_4 Alias Pinc.1

Portb.0 = 1                             ' ukljuci pull-up otpornik (PB0)
Portd.2 = 1                             ' ukljuci pull-up otpornik (PD2)
Portd.3 = 1                             ' ukljuci pull-up otpornik (PD3)
Portc.1 = 1                             ' ukljuci pull-up otpornik (PC1)

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   If Input_1 = 1 Then                  ' ulaz 1 aktivan?
      Relay_1 = 0                       ' ukljuci relej 1
   Else
      Relay_1 = 1                       ' iskljuci relej 1
   End If

   If Input_2 = 1 Then                  ' ulaz 2 aktivan?
      Relay_2 = 0                       ' ukljuci relej 2
   Else
      Relay_2 = 1                       ' iskljuci relej 2
   End If
```

```
   If Input_3 = 1 Then                  ' ulaz 3 aktivan?
      Relay_3 = 0                       ' ukljuci relej 3
   Else
      Relay_3 = 1                       ' iskljuci relej 3
   End If

   If Input_4 = 1 Then                  ' ulaz 4 aktivan?
      Relay_4 = 0                       ' ukljuci relej 4
   Else
      Relay_4 = 1                       ' iskljuci relej 4
   End If
Loop

End
```

Primjer 3

Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se Izlaz_3.

```
$regfile = "m168def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Portc.2 = Output                 ' postavi port releja 1 kao izlaz
Config Portc.0 = Output                 ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                 ' postavi port releja 3 kao izlaz
Config Portd.7 = Output                 ' postavi port releja 4 kao izlaz

Config Portc.4 = Input                  ' postavi port IR1 kao ulaz
Config Portc.3 = Input                  ' postavi port IR2 kao ulaz
Config Portc.5 = Input                  ' postavi port DS1 kao ulaz

Relay_1 Alias Portc.2
Relay_2 Alias Portc.0
Relay_3 Alias Portd.4
Relay_4 Alias Portd.7

Input_ir1 Alias Pinc.4
Input_ir2 Alias Pinc.3
Input_ds1 Alias Pinc.5

Portc.4 = 1                             ' ukljuci pull-up otpornik (PC4)
Portc.3 = 1                             ' ukljuci pull-up otpornik (PC3)
Portc.5 = 1                             ' ukljuci pull-up otpornik (PC5)

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   If Input_ir1 = 0 Then                ' ulaz IR1 aktivan?
      Relay_1 = 0                       ' ukljuci relej 1
   Else
      Relay_1 = 1                       ' iskljuci relej 1
   End If

   If Input_ir2 = 0 Then                ' ulaz IR2 aktivan?
      Relay_2 = 0                       ' ukljuci relej 2
   Else
      Relay_2 = 1                       ' iskljuci relej 2
   End If
```

```
    If Input_ds1 = 0 Then                ' ulaz DS1 aktivan?
       Relay_3 = 0                       ' ukljuci relej 3
    Else
       Relay_3 = 1                       ' iskljuci relej 3
    End If
Loop

End
```

Primjer 4

Testiranje senzora udaljenosti provodi se koristeći sljedeći primjer. Primjer se sastoji od čitanja analogne vrijednosti senzora udaljenosti koji je spojen na ulaz DS1, te ovisno o očitanoj vrijednosti aktivira izlaze 1 – 4.

```
$regfile = "m168def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Adc = Single , Prescaler = Auto

Config Portc.2 = Output                ' postavi port releja 1 kao izlaz
Config Portc.0 = Output                ' postavi port releja 2 kao izlaz
Config Portd.4 = Output                ' postavi port releja 3 kao izlaz
Config Portd.7 = Output                ' postavi port releja 4 kao izlaz

Relay_1 Alias Portc.2
Relay_2 Alias Portc.0
Relay_3 Alias Portd.4
Relay_4 Alias Portd.7

Relay_1 = 1                            ' iskljuci relej 1
Relay_2 = 1                            ' iskljuci relej 2
Relay_3 = 1                            ' iskljuci relej 3
Relay_4 = 1                            ' iskljuci relej 4

Dim W As Word , Channel As Byte

Channel = 5                            ' Postavi port PC5 za ADC kanal (DS1)

Do
   W = Getadc(channel)                 ' Procitaj ADC

   If W > 200 Then                     ' ako je ADC vrijednost DS1 > 200
      Relay_1 = 0                      ' ukljuci relej 1
   Else
      Relay_1 = 1                      ' iskljuci relej 1
   End If

   If W > 300 Then                     ' ako je ADC vrijednost DS1 > 300
      Relay_2 = 0                      ' ukljuci relej 2
   Else
      Relay_2 = 1                      ' iskljuci relej 2
   End If

   If W > 400 Then                     ' ako je ADC vrijednost DS1 > 400
      Relay_3 = 0                      ' ukljuci relej 3
   Else
      Relay_3 = 1                      ' iskljuci relej 3
   End If
```

```
    If W > 500 Then                     ' ako je ADC vrijednost DS1 > 500
       Relay_4 = 0                      ' ukljuci relej 4
    Else
       Relay_4 = 1                      ' iskljuci relej 4
    End If
Loop

End
```

Primjer 5

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).

```
$regfile = "m168def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portb.2 , Servo2 = Portb.1 , Reload = 10

Config Portb.2 = Output                ' postavi port SERVO_1 kao izlaz
Config Portb.1 = Output                ' postavi port SERVO_2 kao izlaz

Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 50                       ' postavi SERVO_1 (1mS puls)
   Servo(2) = 130                      ' postavi SERVO_2 (2mS puls)
   Wait 3
   Servo(1) = 130                      ' postavi SERVO_1 (2mS puls)
   Servo(2) = 50                       ' postavi SERVO_2 (1mS puls)
   Wait 3
Loop

End
```

7.3 *Bluetooth sučelje – ATmega32*

Primjer 1

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **Do-Loop** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza ponavlja sve dok ne zaustavimo program.

```
$regfile = "m32def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Porta.3 = Output                 ' postavi port releja 1 kao izlaz
Config Porta.5 = Output                 ' postavi port releja 2 kao izlaz
Config Porta.7 = Output                 ' postavi port releja 3 kao izlaz
Config Portc.6 = Output                 ' postavi port releja 4 kao izlaz

Relay_1 Alias Porta.3
Relay_2 Alias Porta.5
Relay_3 Alias Porta.7
Relay_4 Alias Portc.6

Relay_1 = 1                             ' iskljuci relej 1
Relay_2 = 1                             ' iskljuci relej 2
Relay_3 = 1                             ' iskljuci relej 3
Relay_4 = 1                             ' iskljuci relej 4

Do
   Relay_1 = 0                          ' ukljuci relej 1
   Waitms 200                           ' cekaj 200 mS
   Relay_1 = 1                          ' iskljuci relej 1
   Waitms 200                           ' cekaj 200 mS

   Relay_2 = 0                          ' ukljuci relej 2
   Waitms 200                           ' cekaj 200 mS
   Relay_2 = 1                          ' iskljuci relej 2
   Waitms 200                           ' cekaj 200 mS

   Relay_3 = 0                          ' ukljuci relej 3
   Waitms 200                           ' cekaj 200 mS
   Relay_3 = 1                          ' iskljuci relej 3
   Waitms 200                           ' cekaj 200 mS

   Relay_4 = 0                          ' ukljuci relej 4
   Waitms 200                           ' cekaj 200 mS
   Relay_4 = 1                          ' iskljuci relej 4
   Waitms 200                           ' cekaj 200 mS
Loop

End
```

Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.

```
$regfile = "m32def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Porta.3 = Output                ' postavi port releja 1 kao izlaz
Config Porta.5 = Output                ' postavi port releja 2 kao izlaz
Config Porta.7 = Output                ' postavi port releja 3 kao izlaz
Config Portc.6 = Output                ' postavi port releja 4 kao izlaz

Config Portc.1 = Input                 ' postavi port ulaza 1 kao ulaz
Config Portc.7 = Input                 ' postavi port ulaza 2 kao ulaz
Config Porta.6 = Input                 ' postavi port ulaza 3 kao ulaz
Config Porta.4 = Input                 ' postavi port ulaza 4 kao ulaz

Relay_1 Alias Porta.3
Relay_2 Alias Porta.5
Relay_3 Alias Porta.7
Relay_4 Alias Portc.6

Input_1 Alias Pinc.1
Input_2 Alias Pinc.7
Input_3 Alias Pina.6
Input_4 Alias Pina.4

Portc.1 = 1                            ' ukljuci pull-up otpornik (PC1)
Portc.7 = 1                            ' ukljuci pull-up otpornik (PC7)
Porta.6 = 1                            ' ukljuci pull-up otpornik (PA6)
Porta.4 = 1                            ' ukljuci pull-up otpornik (PA4)

Relay_1 = 1                            ' iskljuci relej 1
Relay_2 = 1                            ' iskljuci relej 2
Relay_3 = 1                            ' iskljuci relej 3
Relay_4 = 1                            ' iskljuci relej 4

Do
   If Input_1 = 1 Then                 ' ulaz 1 aktivan?
      Relay_1 = 0                      ' ukljuci relej 1
   Else
      Relay_1 = 1                      ' iskljuci relej 1
   End If

   If Input_2 = 1 Then                 ' ulaz 2 aktivan?
      Relay_2 = 0                      ' ukljuci relej 2
   Else
      Relay_2 = 1                      ' iskljuci relej 2
   End If
```

```
   If Input_3 = 1 Then                 ' ulaz 3 aktivan?
      Relay_3 = 0                      ' ukljuci relej 3
   Else
      Relay_3 = 1                      ' iskljuci relej 3
   End If

   If Input_4 = 1 Then                 ' ulaz 4 aktivan?
      Relay_4 = 0                      ' ukljuci relej 4
   Else
      Relay_4 = 1                      ' iskljuci relej 4
   End If
Loop

End
```

Primjer 3

Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se Izlaz_3.

```
$regfile = "m32def..dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Porta.3 = Output                ' postavi port releja 1 kao izlaz
Config Porta.5 = Output                ' postavi port releja 2 kao izlaz
Config Porta.7 = Output                ' postavi port releja 3 kao izlaz
Config Portc.6 = Output                ' postavi port releja 4 kao izlaz

Config Porta.1 = Input                 ' postavi port IR1 kao ulaz
Config Porta.2 = Input                 ' postavi port IR2 kao ulaz
Config Porta.0 = Input                 ' postavi port DS1 kao ulaz

Relay_1 Alias Porta.3
Relay_2 Alias Porta.5
Relay_3 Alias Porta.7
Relay_4 Alias Portc.6

Input_ir1 Alias Pina.1
Input_ir2 Alias Pina.2
Input_ds1 Alias Pina.0

Porta.1 = 1                            ' ukljuci pull-up otpornik (PA1)
Porta.2 = 1                            ' ukljuci pull-up otpornik (PA2)
Porta.0 = 1                            ' ukljuci pull-up otpornik (PA0)

Relay_1 = 1                            ' iskljuci relej 1
Relay_2 = 1                            ' iskljuci relej 2
Relay_3 = 1                            ' iskljuci relej 3
Relay_4 = 1                            ' iskljuci relej 4

Do
   If Input_ir1 = 0 Then               ' ulaz IR1 aktivan?
      Relay_1 = 0                      ' ukljuci relej 1
   Else
      Relay_1 = 1                      ' iskljuci relej 1
   End If

   If Input_ir2 = 0 Then               ' ulaz IR2 aktivan?
      Relay_2 = 0                      ' ukljuci relej 2
   Else
      Relay_2 = 1                      ' iskljuci relej 2
   End If
```

```
    If Input_ds1 = 0 Then                ' ulaz DS1 aktivan?
       Relay_3 = 0                       ' ukljuci relej 3
    Else
       Relay_3 = 1                       ' iskljuci relej 3
    End If
Loop

End
```

Primjer 4

Testiranje senzora udaljenosti provodi se koristeći sljedeći primjer. Primjer se sastoji od čitanja analogne vrijednosti senzora udaljenosti koji je spojen na ulaz DS1, te ovisno o očitanoj vrijednosti aktivira izlaze 1 – 4.

```
$regfile = "m32def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

Config Adc = Single , Prescaler = Auto

Config Porta.3 = Output                ' postavi port releja 1 kao izlaz
Config Porta.5 = Output                ' postavi port releja 2 kao izlaz
Config Porta.7 = Output                ' postavi port releja 3 kao izlaz
Config Portc.6 = Output                ' postavi port releja 4 kao izlaz

Relay_1 Alias Porta.3
Relay_2 Alias Porta.5
Relay_3 Alias Porta.7
Relay_4 Alias Portc.6

Relay_1 = 1                            ' iskljuci relej 1
Relay_2 = 1                            ' iskljuci relej 2
Relay_3 = 1                            ' iskljuci relej 3
Relay_4 = 1                            ' iskljuci relej 4

Dim W As Word , Channel As Byte

Channel = 0                            ' Postavi port PA0 za ADC kanal (DS1)

Do
   W = Getadc(channel)                 ' Procitaj ADC

   If W > 200 Then                     ' ako je ADC vrijednost DS1 > 200
      Relay_1 = 0                      ' ukljuci relej 1
   Else
      Relay_1 = 1                      ' iskljuci relej 1
   End If

   If W > 300 Then                     ' ako je ADC vrijednost DS1 > 300
      Relay_2 = 0                      ' ukljuci relej 2
   Else
      Relay_2 = 1                      ' iskljuci relej 2
   End If

   If W > 400 Then                     ' ako je ADC vrijednost DS1 > 400
      Relay_3 = 0                      ' ukljuci relej 3
   Else
      Relay_3 = 1                      ' iskljuci relej 3
   End If
```

```
    If W > 500 Then                     ' ako je ADC vrijednost DS1 > 500
       Relay_4 = 0                      ' ukljuci relej 4
    Else
       Relay_4 = 1                      ' iskljuci relej 4
    End If
Loop

End
```

Primjer 5

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).

```
$regfile = "m32def.dat"
$crystal = 8000000
$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portd.4 , Servo2 = Portd.5 , Reload = 10

Config Portd.4 = Output                ' postavi port SERVO_1 kao izlaz
Config Portd.5 = Output                ' postavi port SERVO_2 kao izlaz

Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 50                       ' postavi SERVO_1 (1mS puls)
   Servo(2) = 130                      ' postavi SERVO_2 (2mS puls)
   Wait 3
   Servo(1) = 130                      ' postavi SERVO_1 (2mS puls)
   Servo(2) = 50                       ' postavi SERVO_2 (1mS puls)
   Wait 3
Loop

End
```

AVR Studio

8

8. AVR STUDIO

Primjer 1

U ovom primjeru upoznajemo relejne izlaze EMoRo sučelja. Svaki relejni izlaz ima LED (svjetleću diodu) koja označava kad je izlaz aktivan. Korištenje izlaza 1 – 4 prikazat ćemo na primjeru „trčećeg svjetla“.

Primjer programa uključuje i isključuje relejne izlaze svakih 200 mS krećući od releja 1 prema releju 4. Uključivanje i isključivanje relejnih izlaza smješteno je u beskonačnu **While(1)** petlju, te se ciklus uključivanja i isključivanja relejnih izlaza beskonačno ponavlja.

```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){                 /* glavni program                           */

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja    */

   while(1){                 /* beskonacna petlja                        */

      Relay1On();               /* upali relej 1                            */
      _delay_ms(200);           /* cekaj 200 mS                             */
      Relay1Off();              /* ugasi relej 1                            */
      _delay_ms(200);           /* cekaj 200 mS                             */

      Relay2On();               /* upali relej 2                            */
      _delay_ms(200);           /* cekaj 200 mS                             */
      Relay2Off();              /* ugasi relej 2
   */
      _delay_ms(200);           /* cekaj 200 mS                             */

      Relay3On();               /* upali relej 3                            */
      _delay_ms(200);           /* cekaj 200 mS                             */
      Relay3Off();              /* ugasi relej 3                            */
      _delay_ms(200);           /* cekaj 200 mS                             */

      Relay4On();               /* upali relej 4                            */
      _delay_ms(200);           /* cekaj 200 mS                             */
      Relay4Off();              /* ugasi relej 4                            */
      _delay_ms(200);           /* cekaj 200 mS                             */
   }
}
```

Primjer 2

Programski primjer u kojem obavljamo „interakciju“ s mikroračunalom prikazan je ispod. Ovisno o pritisku tipki 1 – 4 upravljamo relejnim izlazima 1 – 4. Mikrokontroler „čeka“ određenu akciju i onda, ovisno o aktiviranom ulazu, automatski aktivira određeni izlaz.

```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja     */

   while(1){                    /* beskonacna petlja                         */

      if(Input1() == 1)         /* pritisnuta tipka 1?                       */
         Relay1On();            /* ukljuci relej 1                           */
      else
         Relay1Off();           /* iskljuci relej 1                          */

      if(Input2() == 1)         /* pritisnuta tipka 2?                       */
         Relay2On();            /* ukljuci relej 2                           */
      else
         Relay2Off();           /* iskljuci relej 2                          */

      if(Input3() == 1)         /* pritisnuta tipka 3?                       */
         Relay3On();            /* ukljuci relej 3                           */
      else
         Relay3Off();           /* iskljuci relej 3                          */

      if(Input4() == 1)         /* pritisnuta tipka 4?                       */
         Relay4On();            /* iskljuci relej 4                          */
      else
         Relay4Off();           /* iskljuci relay 4                          */
   }
}
```

Primjer 3

Testiranje infracrvenih senzora sastoji se od čitanja ulaza IR1, IR2 i DS1, te ovisno o aktivnom ulazu mikroračunalo uključuje određeni izlaz. Ako je aktivan senzor koji je spojen na IR1, uključuje se Izlaz_1, za IR2 uključuje se Izlaz_2, a za DS1 uključuje se izlaz 3.

```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){                 /* glavni program                           */

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja    */

   while(1){                    /* beskonacna petlja                        */

      if(InputIR1() == 1)       /* IR1 senzor aktivan?                      */
         Relay1On();            /* ukljuci relej 1                          */
      else
         Relay1Off();           /* iskljuci relej 1                         */

      if(InputIR2() == 1)       /* IR2 senzor je aktivan?                   */
         Relay2On();            /* ukljuci relay 2                          */
      else
         Relay2Off();           /* iskljuci relay 2                         */

      if(InputDS1() == 1)       /* DS1 senzor aktivan?                      */
         Relay3On();            /* ukljuci relej 3                          */
      else
         Relay3Off();           /* iskljuci relej 3                         */
   }
}
```

Primjer 4

Testiranje senzora udaljenosti provodi se koristeći sljedeći primjer. Primjer se sastoji od čitanja analogne vrijednosti senzora udaljenosti koji je spojen na ulaz DS1, te ovisno o očitanoj vrijednosti aktivira izlaze 1 – 4.

Napomena! USB sučelje ne podržava čitanje analogne vrijednosti, jer ne sadržava analogno-digitalni pretvarač.

```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja     */
   InitAdc(DS1);                /* inicijaliziraj ADC na ulazu DS1           */

   while(1){                    /* beskonacna petlja                         */

      if(GetAdc(DS1) > 200)     /* vrijednost ADC-a veca od 200?             */
         Relay1On();            /* ukljuci relej 1                           */
      else
         Relay1Off();           /* iskljuci relej 1                          */

      if(GetAdc(DS1) > 300)     /* vrijednost ADC-a veca od 300?             */
         Relay2On();            /* ukljuci relej 2                           */
      else
         Relay2Off();           /* iskljuci relej 2                          */

      if(GetAdc(DS1) > 400)     /* vrijednost ADC-a veca od 400?             */
         Relay3On();            /* ukljuci relej 3                           */
      else
         Relay3Off();           /* iskljuci relej 3                          */

      if(GetAdc(DS1) > 500)     /* vrijednost ADC-a veca od 500?             */
         Relay4On();            /* ukljuci relej 4                           */
      else
         Relay4Off();           /* iskljuci relej 4                          */
   }
}
```

Primjer 5

Pogonske servo motore isprobavamo sljedećim primjerom. Nakon incijalizacije servo motora na kanalima SERVO_1 i SERVO_2 ulazimo u beskonačnu petlju u kojoj postavljamo servo motore na suprotne krajnje vrijednosti (robot se kreće naprijed 5s), nakon toga zamjenjujemo vrijednosti (robot se kreće unatrag 5s).

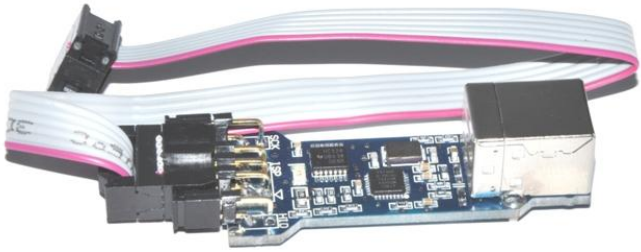
```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja     */
   InitServo(SERVO_1);          /* inicijaliziraj servo na izlazu SERVO_1    */
   InitServo(SERVO_2);          /* inicijaliziraj servo na izlazu SERVO_2    */

   while(1){                    /* beskonacna petlja                         */

      SetServo(SERVO_1, 1000);  /* postavi SERVO_1 pulse 1mS                 */
      SetServo(SERVO_2, 2000);  /* postavi SERVO_2 pulse 2mS                 */

      _delay_ms(5000);          /* cekaj 5S                                  */

      SetServo(SERVO_1, 2000);  /* postavi SERVO_1 pulse 2mS                 */
      SetServo(SERVO_2, 1000);  /* postavi SERVO_2 pulse 1mS                 */

      _delay_ms(5000);          /* cekaj 5S                                  */
   }
}
```

Programator

9

9. *PROGRAMATOR*

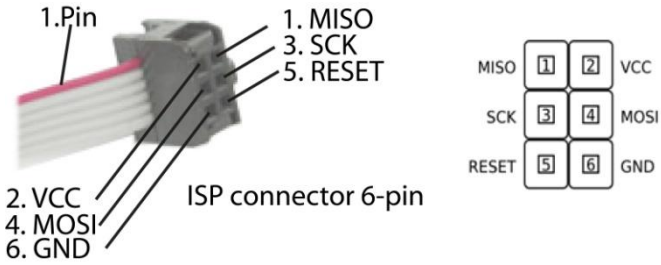
9.1 *USB Inovatic STK500v2 ISP*



Slika 74. Inovatic STK500 ISP programator

Podržana mikroračunala:

ATtiny12, ATtiny13, ATtiny13A, ATtiny15, ATtiny167, ATtiny22, ATtiny2313, ATtiny2313A, ATtiny24, ATtiny24A, ATtiny25, ATtiny26, ATtiny261, ATtiny261A, Attiny4313, ATtiny43U, ATtiny44, ATtiny44A, ATtiny45, ATtiny461, ATtiny461A, ATtiny48, ATtiny84, ATtiny85, ATtiny861, ATtiny861A, ATtiny87, ATtiny88, AT86RF401, AT89S51, AT89S52, AT90CAN128, AT90CAN32, AT90CAN64, AT90PWM1, AT90PWM2, AT90PWM216, AT90PWM2B, AT90PWM3, AT90PWM316, AT90PWM3B, AT90PWM81, AT90S1200, AT90S2313, AT90S2323, AT90S2343, AT90S4414, AT90S4433, AT90S4434, AT90S8515, AT90S8535, AT90SCR100H, AT90USB1286, AT90USB1287, AT90USB162, AT90USB646, AT90USB647, AT90USB82, ATA6289, ATmega103, ATmega128, ATmega1280, ATmega1281, ATmega1284, ATmega1284P, ATmega128A, ATmega128RFA1, ATmega16, ATmega161, ATmega162, ATmega163, ATmega164A, ATmega164P, ATmega164PA, ATmega165, ATmega165A, ATmega165P, ATmega165PA, ATmega168, ATmega168A, ATmega168P, ATmega168PA, ATmega169, ATmega169A, ATmega169P, ATmega169PA, ATmega16A, ATmega16HVA, ATmega16HVA2, ATmega16HVB, ATmega16M1, ATmega16U2, ATmega16U4, Atmega2560, Atmega2561, ATmega32, ATmega323, ATmega324A, ATmega324P, ATmega324PA, ATmega325, ATmega3250, ATmega3250P, ATmega325A, ATmega325P, ATmega328, ATmega328P, ATmega329, ATmega3290, ATmega3290P, ATmega329A, ATmega329P, ATmega329PA, ATmega32A, ATmega32C1, ATmega32HVB, ATmega32M1, ATmega32U2, ATmega32U4, ATmega32U6, ATmega48, ATmega48A, ATmega48P, ATmega48PA, ATmega64, ATmega640, ATmega644, ATmega644A, ATmega644P, ATmega644PA, ATmega645, ATmega6450, ATmega6450A, ATmega645A, ATmega649, ATmega6490, ATmega6490A, ATmega649A, ATmega649P, ATmega64A, ATmega64C1, ATmega64HVE, ATmega64M1, ATmega8, ATmega8515, ATmega8535, ATmega88, ATmega88A, ATmega88P, ATmega88PA, ATmega8A, ATmega8HVA, ATmega8U2.



Slika 75. ISP konektor

Za programiranje se koristi standardizirani 6-pinski Atmel ISP konektor sa shematskim prikazom na prethodnoj slici.

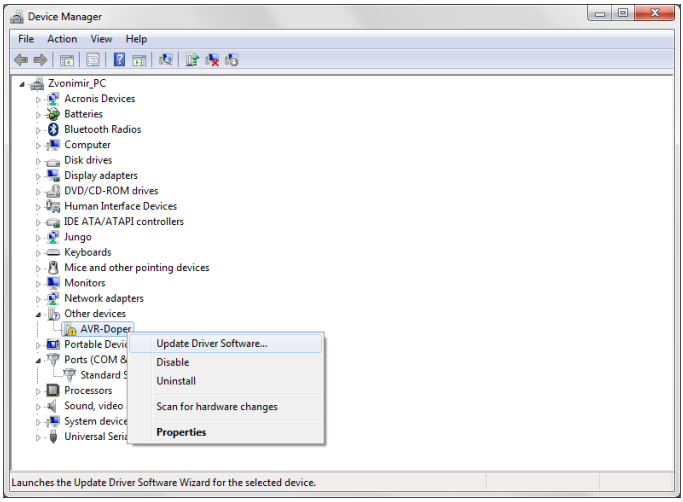
9.1.1 Instalacija upravljačkih programa

Ovisno o tome koristite li x86 ili x64 Windowse, postoje dvije vrste upravljačkih programa. Provjeru obavite na sljedeći način:

- Start → All Programs → Accessories → System Tools → System Information
- Pod stavkom „System type“ ispisan je tip operacijskog sustava

Instalacija upravljačkih programa:

- Provjeriti kratkospojnike (jumpere) HID i SCK (moraju biti odspojeni)
- Povezati programator USB kabelom s računalom
- Otvoriti Device Manager (pritisnuti zajedno na tipkovnici znak „Windows“ (između Ctrl i Alt) i slovo „R“ → upisati „devmgmt.msc“ (bez navodnika) → OK)



Slika 76. Instalacija upravljačkih programa – Inovatic STK500

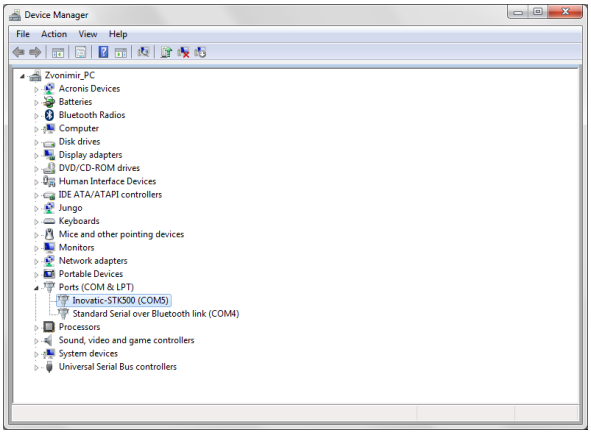
- <u>Windows Vista i 7</u> - (desni klik na AVR-Doper) → Update Driver Software → Browse my computer for driver software → Browse → (*DVD* → Drivers → Inovatic STK500 → x86_XpVista7 ili x64_XpVista7) → OK → Next → Install this driver software anyway → Close

- <u>Windows XP</u> - (desni klik na AVR-Doper) → Update Driver → No, not this time → Next → Install from specific location (Advanced) → Next → Include this location in the search → Browse → (*DVD* → Drivers → Inovatic STK500 → x86_XpVista7 ili x64_XpVista7) → OK → Next → Continue Anyway → Finish

Dodatno podešavanje upravljačkih programa na x64 operacijski sustav:

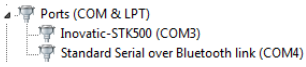
— Nakon instalacije upravljačkih programa potrebno je pokrenuti program deso13b.exe (Driver Signature enforcement Overrider) koji se nalazi u (*DVD* → Drivers → Inovatic STK500 → x64_XpVista7) → Next → Yes → Odaberite „Sign a System File“ → Next → dodajte adresu lowcdc.inf (*DVD*\Drivers\Inovatic STK500\x64_XpVista7\lowcdc.inf) → Ok → Odaberite ponovno „Sign a System File“ → dodajte adresu lowcdc.sys (*DVD*\Drivers\Inovatic STK500\x64_XpVista7\lowcdc.sys) → Ok → Enable Testmode → Next → ponovno pokrenite računalo (restart).

Nakon uspješne instalacije upravljačkih programa treba provjeriti COM port na kojem je instaliran (AVR Studio prepoznaje COM1, COM2, COM3 i COM4), te je potrebno prebaciti na neki niži slobodni (COM3 ili COM4)



Slika 77. Provjera instalacije i COM Port-a

— (desni klik na Inovatic-STK500) → Properties → Port Settings → Advanced → COM Port Number promijeniti u COM3 ili COM4 → OK → Yes → OK → (odspojiti i spojiti ponovno programator s računalom)

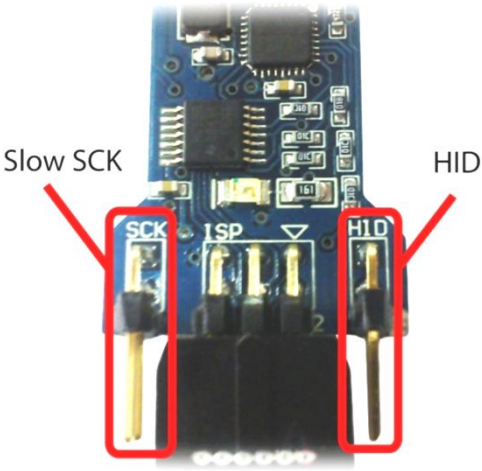


Slika 78. Uspješna instalacija programatora

<u>Uklanjanje „Test mode“ natpisa u desnom donjem kutu:</u>

— pokrenuti program RemoveWatermarkX64.exe za x64 operacijske sustave ili RemoveWatermarkX86.exe za x86 operacijske sustave koji se nalazi u (*DVD* → Drivers → Inovatic STK500 → x86_XpVista7 ili x64_XpVista7) → Run → pritisnuti tipku „Y“ → ponovno pokrenuti računalo (restart)

9.1.2 *HID i Slow SCK način rada*



Slika 79. HID i SCK način rada

Human Interface Device - HID

Postavljanjem kratkospojnika na HID letvice, programator ulazi u drugi način rada. Nije potrebna instalacija upravljačkih programa jer ga Windowsi prepoznaju kao Human Interface Device te ima mogućnost programiranja u AVRDude i SinaProg programima. Ovaj način rada „olakšava“ programiranje, tj. prebacivanje postojeće HEX datoteke na procesor, jer ne zahtjeva nikakve prethodne instalacije. Ako za programiranje rabite AVR Studio, ovaj kratkospojnik ne smije biti postavljen.

Slow SCK - SCK

Za programiranje mikroračunala/uređaja koji zahtjevaju nisku frekvenciju ISP programiranja (ako se upotrebljava softver koji ne podržava podešavanje ISP frekvencije). Podesiva ISP frekvencija omogućava programiranje na npr. 32kHz.

Kratkospojnik za napajanje - PWR

Slika 80. PWR – USB 5V izlaz

Kratkospojnik na donjoj strani tiskane pločice (slika 10.) omogućava napajanje uređaja izravno preko USB-a. Prema osnovnoj konfiguraciji programatora taj kratkospojnik je odspojen, pa ga je potrebno prelemiti, ako želimo napajati uređaj koji programiramo bez dodatnog vanjskog napajanja.

Bitno!!!

Ako se koristi napajanje preko programatora (potrosac - 5V i max 0.5A) ne smijemo upotrijebiti vanjsko napajanje, jer „slučajnim“ krivim okretanjem ISP konektora postoji mogućnost uništenja programatora.

Dodatak

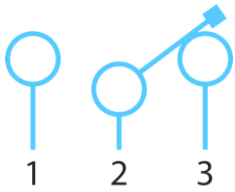
10

10. DODATAK

10.1 *Načelo rada preklopnih (relejnih) izlaza*

Načelo rada preklopnih kontakata može se shvatiti prema simbolu koji je otisnut u podnožju kontakta. Svaki izlaz je jedan relejni preklopni kontakt. Maksimalno opterećenje koje se može priključiti na relejni kontakt izlaza je 2A pri naponu od 24V.

Svaki izlaz ima tri priključka koji su veze na preklopni kontakt releja. Na slici ispod možete vidjeti shemu koja je otisnuta ispred svakog izlaza.



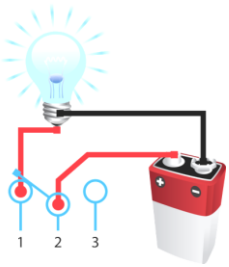
Slika 81. Shema preklopnog kontakta

Na slici vidimo da izlaz ima tri kontakta, 1, 2 i 3. Kada izlaz nije aktivan, relej je tako spojen da je ostvaren kontakt između priključka 2 i 3. Kada izlaz postane aktivan, programski uključimo izlaz, spoj između 2 i 3 prestaje, i u tom trenutku spaja se 1 i 2. Znači, kada je sučelje isključeno ili izlaz nije aktivan, spoj je između kontakata 2 i 3 i tuda može teći struja. Kada programski uključimo izlaz, kroz kontakte 2 i 3 struja više ne može teći, ali sada može teći kroz kontakte 1 i 2. Za kontakte 1 i 3 kažemo da su „statični“, a kontakt 2 je „preklopni“ jer kada izlaz nije aktivan spojen je s 3, a kad je izlaz aktivan, kontakt se preklapa i spaja se na 1.

Na slikama 82. i 83. shematski je prikazano uključivanje i isključivanje žaruljice s dodatnim vanjskim napajanjem.

Slika 82. Spajanje izlaza i žaruljice, izlaz sučelja nije aktivan

Kada izlaz nije aktivan, ne postoji spoj između kontakata 1 i 2 i žaruljica ne svijetli. Kada izlaz postane aktivan (logička jedinica), preklopni kontakt spoji se s kontaktom 1 i tako upali žaruljicu.

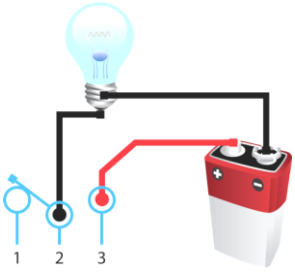


Slika 83. Izlaz sučelja je aktivan, žaruljica se upalila

Ako želimo suprotni učinak, tj. kad je izlaz aktivan želimo da se žaruljica ugasi. Shema spajanja prikazana je na slici ispod.

Slika 84. Spajanje izlaza i žaruljice, obrnuto upravljanje

Na slici 84. sada rabimo kontakte izlaza 2 i 3. Kada izlaz nije aktivan, žaruljica svijetli jer je ostvaren kontakt u releju između priključaka 2 i 3. Kada izlaz postane aktivan, gubi se spoj između 2 i 3, odnosno spaja se 1 i 2 i žaruljica prestaje raditi. Kada je izlaz aktivan, žaruljica se gasi kao što se vidi na slici 85.

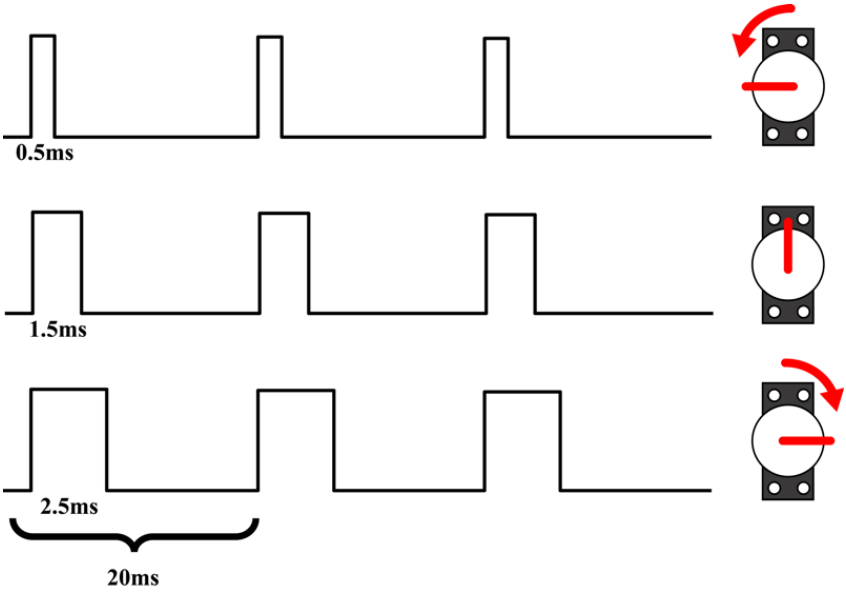


Slika 85. Izlaz sučelja pri aktivnom izlazu i obrnutom upravljanju

10.2 *Upravljanje servo motorima*

RC servo motori najčešće se upotrebljavaju kada trebamo savladati male sile. Uglavnom se primjenjuju u izradi malih robota, robotskih ruka i drugih manipulatora, te u modelarstvu. Osnovu RC servo motora čini DC motor koji je povezan s potenciometrom. RC servo motori ne zahtijevaju snažno napajanje i pouzdani su u radu.

Za kontrolu RC servo motora potrebno je generiranje PWM (pulsno-širinska modulacija) signala određene širine. Trajanje periode mora biti 20ms, a širina pulsa kreće se između 0.5ms i 2.5ms. Pri širini pulsa od 1.5ms servo motor zauzima središnji položaj (srednji položaj ili položaj mirovanja ako se radi o servo motorima s kontinuiranom rotacijom). Za 1ms servo se postavlja u krajnji lijevi, a za 2ms u krajnji desni položaj. Krajnji lijevi i desni položaj ovise o tipu servo motora i proizvođaču, pa krajnje pozicije treba podešavati u proširenom rasponu od 0.5ms do 2.5ms.

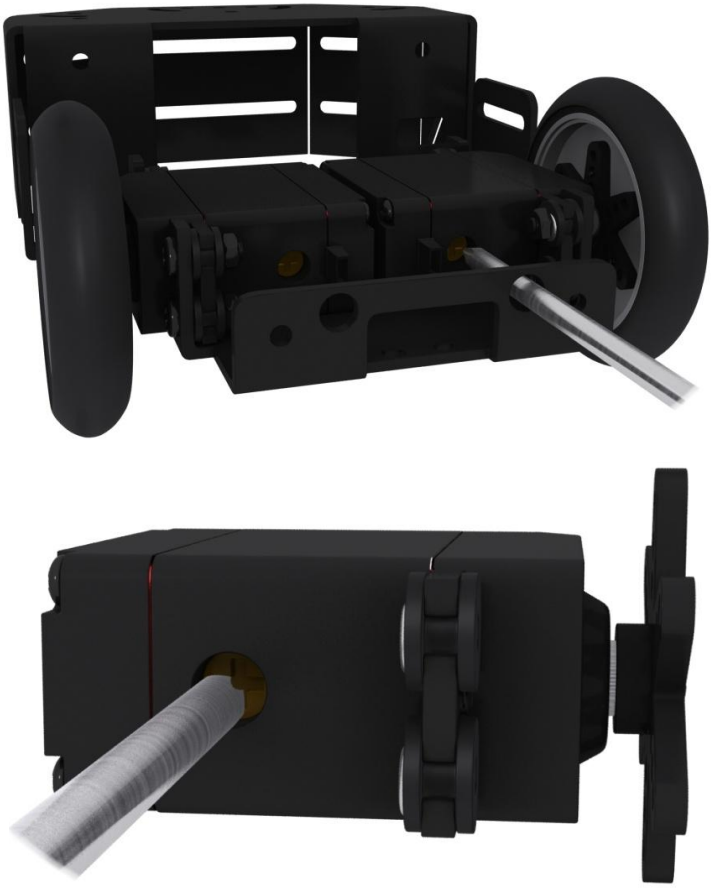


Slika 86. Servo motori – upravljanje širinom pulsa

10.3 *Centriranje servo motora*

Servo motori s konstantnom rotacijom (pogonski servo motori) nisu tvornički centrirani i zahtijevaju kalibraciju, tj. postavljanje na nultu poziciju. Ako motore ne centriramo, oni se mogu okretati i ako im zadamo nulti položaj (1.5ms). Od alata za centriranje potreban nam je odvijač kako bismo podesili potenciometar unutar servo motora. Servo motore možemo centrirati dok nisu postavljeni na donji dio šasije, ali i kad ih postavimo, jer postoji odgovarajuća rupa na šasiji kroz koju pristupamo potenciometru.

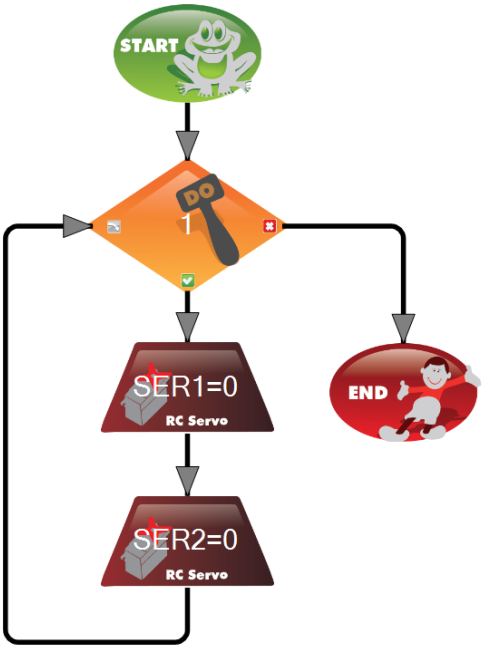
Pozor! Potenciometar je vrlo osjetljiv, stoga je potrebno pažljivo okretati i ne gurati i stiskati više nego što je potrebno, jer se servo motor može oštetiti.



Slika 87. Podešavanje potenciometra na servo motoru

Za centriranje servo motora treba pokrenuti sljedeći programski primjer (ovisno o programu kojim se koristite). U programu postavljamo servo na središnju poziciju (1.5ms) što znači da mora mirovati (bez zujanja i okretanja). Nakon pokretanja programa, polagano okretati potenciometar na servo motoru dok motor ne bude posve miran.

FlowDraw



FreeBasic FBIDE

```
#Include once "emoro.bi"            ' dodijeljivanje biblioteka                '
#Include "fbgfx.bi"

EmoroAutoConnect()                  ' otvori komunikaciju sa suceljem          '

InitServo(SERVO_1)                  ' pokreni servo SERVO_1                    '
InitServo(SERVO_2)                  ' pokreni servo SERVO_2                    '

Do
   SetServo(SERVO_1, 1500)          ' postavi Servo 1 pulse svakih 1.5mS       '
   SetServo(SERVO_2, 1500)          ' postavi Servo 2 pulse svakih 1.5mS       '
Loop
End
```

BASCOM AVR

AT90USB162

```
$regfile = "usb162.dat"
$crystal = 8000000

$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portc.5 , Servo2 = Portc.6 , Reload = 10

Config Portc.5 = Output                ' postavi port SERVO_1 kao izlaz
Config Portc.6 = Output                ' postavi port SERVO_2 kao izlaz


Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 90                       ' postavi SERVO_1 (1.5 mS puls)
   Servo(2) = 90                       ' postavi SERVO_2 (1.5 mS puls)
Loop

End
```

ATmega168

```
$regfile = "m168def.dat"
$crystal = 8000000

$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portb.2 , Servo2 = Portb.1 , Reload = 10

Config Portb.2 = Output                ' postavi port SERVO_1 kao izlaz
Config Portb.1 = Output                ' postavi port SERVO_2 kao izlaz


Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 90                       ' postavi SERVO_1 (1.5 mS puls)
   Servo(2) = 90                       ' postavi SERVO_2 (1.5 mS puls)
Loop

End
```

ATmega32

```
$regfile = "m32def.dat"
$crystal = 8000000

$hwstack = 32
$swstack = 10
$framesize = 40

'koristimo 2 servo motora sa razlucivosti od 10uS
Config Servos = 2 , Servo1 = Portd.4 , Servo2 = Portd.5 , Reload = 10

Config Portd.4 = Output                ' postavi port SERVO_1 kao izlaz
Config Portd.5 = Output                ' postavi port SERVO_2 kao izlaz


Enable Interrupts                      ' omoguci prekide

Do
   Servo(1) = 90                       ' postavi SERVO_1 (1.5 mS puls)
   Servo(2) = 90                       ' postavi SERVO_2 (1.5 mS puls)
Loop

End
```

AVR Studio

```
#include <inovatic\emoro.h>     /* dodijeljivanje biblioteka                */

int main(void){

   InitIO();                    /* inicijaliziraj ulaze i izlaze sucelja     */
   InitServo(SERVO_1);          /* inicijaliziraj servo na izlazu SERVO_1    */
   InitServo(SERVO_2);          /* inicijaliziraj servo na izlazu SERVO_2    */

   while(1){                    /* beskonacna petlja                         */

      SetServo(SERVO_1, 1500);  /* postavi SERVO_1 pulse 1.5 mS              */
      SetServo(SERVO_2, 1500);  /* postavi SERVO_2 pulse 1.5 mS              */

   }
}
```

Postavljanjem servo motora u središnje stanje mirovanja podesili smo tako da svaki put kad pozovemo to isto stanje zaustavimo motor. Napredno podešavanje [28]servo motora dodatno je objašnjeno u videolekcijama.

10

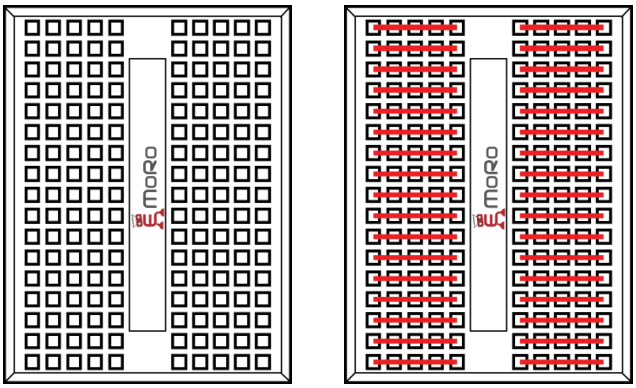
[28] Napredno podešavanje obavljamo u samom programu tako da korigiramo smjer. Npr. ako upravljamo robotom, potrebno je postaviti male korekcije na pravcu kretanja, što znači da ako jednom motoru zadamo smjer 1200 (1.2ms), a drugom 1800 (1.8ms), robot se najvjerojatnije neće kretati savršeno po pravcu. Stoga je potrebno jednostavnom korekcijom postaviti kretanje po pravcu (npr. jedan motor trebali biste umjesto na 1200 (1.2ms) postaviti na 1210 (1.21ms), a drugi ostaviti na 1800 (1.8ms))

10.4 *Eksperimentalna pločica*

Eksperimentalna pločica općenito služi za isprobavanje/testiranje određenih sklopova i shema. Ako imate gotovu shemu, prvo je treba temeljito iskušati na eksperimentalnoj pločici prije izrade tiskane pločice i lemljenja. Eksperimentalna pločica, kao što joj i samo ime govori, omogućuje eksperimentiranje s raznim projektima/shemama.

Eksperimentalna pločica nije nešto komplicirano; na njoj možete raditi i isprobavati sve vrste shema i sklopova. Možete povezivati više pločica u "jednu" ako dimenzije jedne nisu dovoljne za vaš projekt.

Glavno i jedino što morate znati za eksperimentalnu pločicu jest to da je pločica podijeljena na redove i stupce. Svaki redak na slici označen linijom je mjesto istog potencijala (kao da je ispod tih pet rupica provučen vodič).



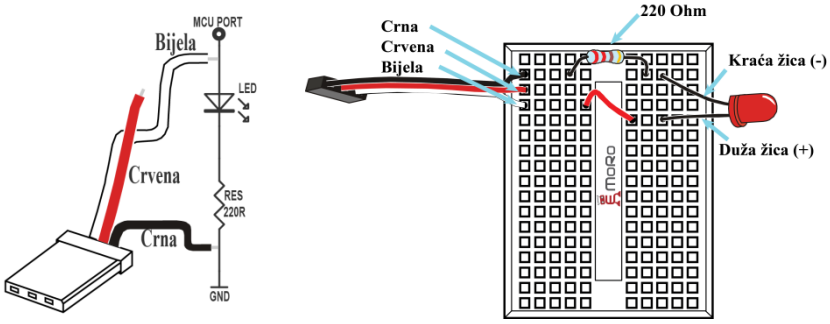
Slika 88. Testna pločica i shematski prikaz spojeva

U nastavku objasnit ćemo spajanje svjetleće diode (LED) i temperaturnog senzora DS1820.

Programski primjeri za provjeru spojenih elektroničkih strujnih krugova (LED i temp. senzor) na eksperimentalnoj pločici nalaze se na DVD-u priloženom uz sučelje.

Shema i načelo spajanja LED na eksperimentalnu pločicu

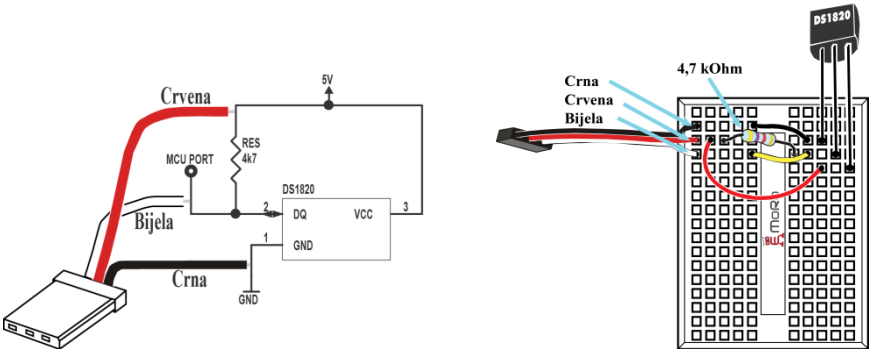
Za spajanje LE Diode na eksperimentalnu pločicu postoji shema. Potrebno je obratiti pozornost na polaritet (+ i -). LE Dioda je upravljana preko porta mikrokontrolera, stoga je potrebno spojiti trožilni kabel (JR-kabel) na port DS1 na sučelju (trokutić označava GND, tj. crnu žicu). U ovom primjeru potrebne su nam samo crna i bijela žica (crna žica – GND, bijela žica – port mikroračunala). Prvo je potrebno spojiti bijelu žicu s diodom (+, duža žica na diodi), nakon toga drugi kraj diode (-) spojiti na jedan kraj otpornika od 220 Ohm (boja - crvena/crvena/smeđa), a drugi kraj otpornika treba spojiti na crnu žicu. Shema i prikaz spajanja prikazani su na slici ispod.



Slika 89. Shema i prikaz spajanja na testnu pločicu (LED)

Shema i načelo spajanja temperaturnog senzora eksperimentalnu pločicu

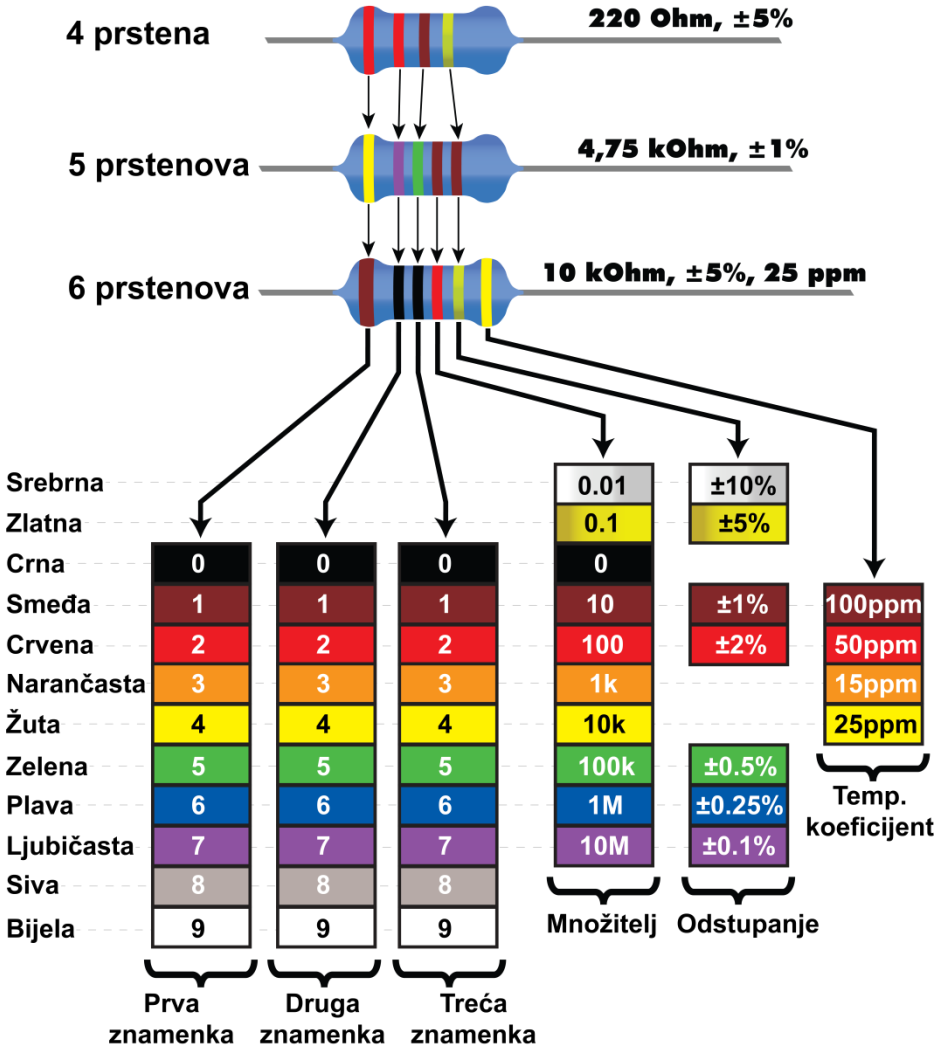
Za spajanje temperaturnog seznora DS1820 na eksperimentalnu pločicu imamo shemu prema kojoj spajamo (lijevo na slici ispod). Za spajanje temperaturnog senzora potrebni su nam spojni vodiči, JR kabel, otpornik od 4.7 kOhm, te temperaturni senzor. Shema i spajanje prikazani su na slici ispod.



Slika 90. Shema i prikaz spajanja na testnu pločicu (DS1820 – temperaturni senzor)

Otpornici

Otpornik je pasivna elektronička komponenta koja pruža otpor struji. Otpornici sa žičanim izvodima označeni su prstenovima različitih boja što nam olakšava raspoznavanje vrijednosti. Postoje otpornici s četiri, pet i šest prstenova kao što je prikazano na slici ispod. Za čitanje vrijednosti otpornika iz boja prstena, potrebno je uvijek krenuti od prstena bližeg rubu ili, ako su krajnja dva jednako udaljena od ruba, potrebno je čitati od šireg prstena.



Slika 91. Otpornici – shematski prikaz čitanja vrijednosti po bojama

10.5 *Popis funkcija u FreeBasic-u*

Prototip funkcije	Opis funkcije
Declare Sub EmoroAutoConnect()	Povezivanje sa EMoRo sučeljem
Declare Sub EmoroConnect(com_port **As** String)	Povezivanje sa EMoRo sučeljem na poznatom COM portu
Declare Sub EmoroDisconnect()	Odspajanje sa EMoRo sučelja
Declare Function Input1() **As** Unsigned Byte	Čitanje ulaza 1
Declare Function Input2() **As** Unsigned Byte	Čitanje ulaza 2
Declare Function Input3() **As** Unsigned Byte	Čitanje ulaza 3
Declare Function Input4() **As** Unsigned Byte	Čitanje ulaza 4
Declare Function InputIR1() **As** Unsigned Byte	Čitanje ulaza IR1
Declare Function InputIR2() **As** Unsigned Byte	Čitanje ulaza IR2
Declare Function InputDS1() **As** Unsigned Byte	Čitanje ulaza DS1
Declare Sub Relay1On()	Uključivanje releja 1
Declare Sub Relay2On()	Uključivanje releja 2
Declare Sub Relay3On()	Uključivanje releja 3
Declare Sub Relay4On()	Uključivanje releja 4
Declare Sub Relay1Off()	Isključivanje releja 1
Declare Sub Relay2Off()	Isključivanje releja 2
Declare Sub Relay3Off()	Isključivanje releja 3
Declare Sub Relay4Off()	Isključivanje releja 4
Declare Sub Relay1Toggle()	Promjena stanja releja 1
Declare Sub Relay2Toggle()	Promjena stanja releja 2
Declare Sub Relay3Toggle()	Promjena stanja releja 3
Declare Sub Relay4Toggle()	Promjena stanja releja 4
Declare Sub InitServo(ch **As** Unsigned Byte)	Inicijalizacija servo motora

Declare Sub StopServo(ch **As** Unsigned Byte)	Zaustavljanje servo motora
Declare Sub SetServo(ch **As** Unsigned Byte, _ pulse **As** Unsigned Integer)	Postavljanje pozicije servo motora
Declare Sub InitAdc(ch **As** Unsigned Byte)	Inicijalizacija AD pretvarača
Declare Sub StopAdc(ch **As** Unsigned Byte)	Zaustavljanje AD pretvarača
Declare Function GetAdc(ch **As** Unsigned Byte) _ **As** Unsigned Integer	Čitanje AD pretvarača
Declare Sub InitDcMotor()	Inicijalizacija DC motora
Declare Sub StopDcMotor()	Zaustavljanje DC motora
Declare Sub SetRightDcMotor(direction **As** Unsigned Byte, _ speed **As** Unsigned Byte)	Postavljanje brzine i smjera desnog DC motora
Declare Sub SetLeftDcMotor(direction **As** Unsigned Byte, _ speed **As** Unsigned Byte)	Postavljanje brzine i smjera lijevog DC motora
Declare Sub BluetoothChangeName(new_name **As** String)	Mjenjanje imena bluetooth modula
Declare Sub BluetoothChangePasskey(new_passkey **As** String)	Mjenjanje lozinke bluetooth modula

Prije upravljanja sučeljem potrebno je otvoriti komunikaciju potprogramom **EmoroAutoConnect()** ili **EmoroConnect()**

Potprogram traži EMoRo sučelje na COM portovima [0 – 20] te otvara komunikaciju s prvim pronađenim sučeljem.

```
Declare Sub EmoroAutoConnect()
```

Parametri:
nema

Povratna vrijednost:
nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' automatsko otvaranje komunikacije
End                                         ' kraj programa
```

Potprogram otvara komunikaciju s EMoRo sučeljem na poznatom COM portu.

```
Declare Sub EmoroConnect(com_port As String)
```

Parametri:
nema

Povratna vrijednost:
nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                  ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                       ' dodjeljivanje fbgfx biblioteke

EmoroConnect("COM2")                      ' otvaranje komunikacije na portu COM 2
End                                       ' kraj programa
```

Potprogram zatvara komunikaciju s EMoRo sučeljem.

```
Declare Sub EmoroDisconnect()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Funkcije čitaju ulaze (1, 2, 3, 4) i tipke (1, 2, 3, 4).

LED svakog ulaza prikazuje trenutačno stanje ulaza. Ako LED svijetli, ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli, ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

```
Declare Function Input1() As Unsigned Byte
Declare Function Input2() As Unsigned Byte
Declare Function Input3() As Unsigned Byte
Declare Function Input4() As Unsigned Byte
```

Parametri:

nema

Povratna vrijednost:

Unsigned Byte – Stanje ulaza (ulazi 1 – 4, tipke 1 – 4)

(0) – puštena tipka ili neaktivan ulaz – (Input High – 5V)

(1) – pritisnuta tipka ili aktivan ulaz – (Input Low – 0V)

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                     ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                      ' otvaranje komunikacije
Do                                      ' beskonačna petlja
   If Input1() = 1 Then                 ' ulaz 1 aktivan (pritisnuta tipka 1)?
      Relay1On()                        ' uključi relej 1
   End If

   If Input2() = 1 Then                 ' ulaz 2 aktivan (pritisnuta tipka 2)?
      Relay2On()                        ' uključi relej 2
   End If

   If Input3() = 1 Then                 ' ulaz 3 aktivan (pritisnuta tipka 3)?
      Relay3On()                        ' uključi relej 3
   End If

   If Input4() = 1 Then                 ' ulaz 4 aktivan (pritisnuta tipka 4)?
      Relay4On()                        ' uključi relej 4
   End If
Loop
EmoroDisconnect()                       ' zatvaranje komunikacije
End                                     ' kraj programa
```

Funkcije čitaju ulaze IR1, IR2 i DS1.

LED ulaza IR1 i IR2 prikazuje trenutačno stanje ulaza. Ako LED svijetli, ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli, ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

Navedene ulaze moguće je čitati samo ako na njima nije inicijalizirana druga funkcionalnost. Ulazima se može koristiti servodriver i adc_driver i u slučaju da je ovim portovima inicijalizirana druga funkcionalnost, ne mogu se čitati kao TTL ulazi.

Npr. izvršavanjem **InitServo(IR1); InitServo(IR2);** IR1 i IR2 bit će konfigurirani za funkcionalnost servomotora, te se funkcije **InputIR1()** i **InputIR2()** ne mogu upotrijebiti.

```
Declare Function InputIR1() As Unsigned Byte
Declare Function InputIR2() As Unsigned Byte
Declare Function InputDS1() As Unsigned Byte
```

Parametri:

nema

Povratna vrijednost:

Unsigned Byte – Stanje ulaza [IR1, IR2, DS1]

(0) – ulaz neaktivan – (Input High – 5V)

(1) – ulaz aktivan – (Input Low – 0V)

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
Do                                          ' beskonačna petlja
   If InputIR1() = 1 Then                   ' ulaz IR1 aktivan (0V)?
      Relay1On()                            ' uključi relej 1
   End If

   If InputIR2() = 1 Then                   ' ulaz IR2 aktivan (0V)?
      Relay2On()                            ' uključi relej 2
   End If

   If InputDS1() = 1 Then                   ' ulaz DS1 aktivan (0V)?
      Relay3On()                            ' uključi relej 3
   End If
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogrami uključuju relejne izlaze.

```
Declare Sub Relay1On()
Declare Sub Relay2On()
Declare Sub Relay3On()
Declare Sub Relay4On()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
Do                                          ' beskonačna petlja
   Relay1On()                               ' uključi relej 1
   Relay2On()                               ' uključi relej 2
   Relay3On()                               ' uključi relej 3
   Relay4On()                               ' uključi relej 4
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogrami isključuju relejne izlaze.

```
Declare Sub Relay1Off()
Declare Sub Relay2Off()
Declare Sub Relay3Off()
Declare Sub Relay4Off()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
Do                                          ' beskonačna petlja
   Relay1Off()                              ' isključi relej 1
   Relay2Off()                              ' isključi relej 2
   Relay3Off()                              ' isključi relej 3
   Relay4Off()                              ' isključi relej 4
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogrami mijenjaju stanje relejnih izlaza. Pozivom potprograma uključeni relejni izlaz će se isključiti, a isključeni relejni izlaz uključiti.

```
Declare Sub Relay1Toggle()
Declare Sub Relay2Toggle()
Declare Sub Relay3Toggle()
Declare Sub Relay4Toggle()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
Do                                          ' beskonačna petlja
   Relay1Toggle()                           ' promijeni stanje releja 1
   Relay2Toggle()                           ' promijeni stanje releja 2
   Relay3Toggle()                           ' promijeni stanje releja 3
   Relay4Toggle()                           ' promijeni stanje releja 4
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potrogram inicijalizira izlaz servo motora na kojemu servo driver generira servo pulse. Sučelje EMoRo podupire 7 servo motora na izlazima SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2 i DS1, a za upravljanje servo motorima potrebno je prvo obaviti inicijalizaciju servo izlaza. Nakon inicijalizacije servo motora zadano stanje izlaza postavljeno je na vrijednost 1500 što odgovara impulsu 1.5 mS. Treba obratiti pozornost na to da se nakon inicijalizacije servo motora na ovim portovima ne može upotrijebiti inicijalizacija za drugu funkcionalnost, kao ni funkcije koje čitaju port kao ulaz.

Npr. pozivom **InitServo(IR1)** IR1 port konfiguriran je kao servo izlaz, te se ne može rabiti funkcija čitanja ulaza **InputIR1()**, kao ni inicijalizacija analogno-digitalnog pretvarača na portu IR1.

DC motor driver koristi se zajedničkim portovima SERVO_2, PWM_1, PWM_2, i ako se jedan od navedenih izlaza konfigurira kao servo motor tada se DC motori ne mogu koristiti.

Inicijalizirani servo driver generira servopulse u intervalu [0,5 mS – 2,5 mS] svakih 20mS.

```
Declare Sub InitServo(ch As Unsigned Byte)
```

Parametri:

Unsigned Byte ch	– servo motor	[SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"          ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"               ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                ' otvaranje komunikacije
InitServo(SERVO_1)                ' pokreni servo motor na portu SERVO_1
InitServo(SERVO_2)                ' pokreni servo motor na portu SERVO_2
InitServo(PWM_1)                  ' pokreni servo motor na portu PWM_1
InitServo(PWM_2)                  ' pokreni servo motor na portu PWM_2
InitServo(IR1)                    ' pokreni servo motor na portu IR1
InitServo(IR2)                    ' pokreni servo motor na portu IR2
InitServo(DS1)                    ' pokreni servo motor na portu DS1
Do                                ' beskonačna petlja
Loop
EmoroDisconnect()                 ' zatvaranje komunikacije
End                               ' kraj programa
```

Potprogram isključuje izlaz servo motora i konfigurira port kao ulaz.

```
Declare Sub StopServo(ch As Unsigned Byte)
```

Parametri:

Unsigned Byte ch	– servo motor	[SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                     ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                      ' otvaranje komunikacije
InitServo(SERVO_1)                      ' pokreni servo motor na portu SERVO_1

Do                                      ' beskonačna petlja
   If Input1() = 1 Then                 ' pritisnuta tipka 1?
      StopServo(SERVO_1)                ' isključi servo motor na portu SERVO_1
   End If
Loop
EmoroDisconnect()                       ' zatvaranje komunikacije
End                                     ' kraj programa
```

Potprogram postavlja novu poziciju servo motora.

```
Declare Sub SetServo(ch As Unsigned Byte, pulse As Unsigned Integer)
```

Parametri:

Unsigned Byte ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Unsigned Integer position – nova pozicija servo motora [500 – 2500] µs

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"          ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"               ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                ' otvaranje komunikacije
InitServo(SERVO_1)                ' pokreni servo motor na portu SERVO_1

Do                                ' beskonačna petlja
   SetServo(SERVO_1, 1000)        ' postavi motor SERVO_1 pulse 1000uS (1mS)
Loop
EmoroDisconnect()                 ' zatvaranje komunikacije
End                               ' kraj programa
```

Potprogram inicijalizira analogno-digitalni pretvarač (ADC). Sučelje emoro podržava 3 analogno-digitalna pretvarača na ulazima IR1, IR2 i DS1, i prije čitanja ADC-a potrebno je obaviti inicijalizaciju **InitAdc()** potprogramom.

```
Declare Sub InitAdc(ch As Unsigned Byte)
```

Parametri:

Unsigned Byte ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitAdc(IR1)                                ' inicijaliziraj ADC na portu IR1
InitAdc(IR2)                                ' inicijaliziraj ADC na portu IR2
InitAdc(DS1)                                ' inicijaliziraj ADC na portu DS1

Do                                          ' beskonačna petlja
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogram zaustavlja analogno-digitalni pretvarač (ADC).

```
Declare Sub StopAdc(ch As Unsigned Byte)
```

Parametri:

Unsigned Byte ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitAdc(DS1)                                ' inicijaliziraj ADC na portu IR1

Do                                          ' beskonačna petlja
   If Input1() = 1 Then                     ' pritisnuta tipka 1?
      StopAdc(DS1)                          ' isključi ADC na portu DS1
   End If
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogram čita analogno-digitalni pretvarač. Prije čitanja ADC-a potrebno je obaviti inicijalizaciju AD pretvarača **InitAdc()** potprogramom.

Napon na ulazu pretvarača računa se formulom

U = (5 / 1024) x GetAdc(ch)

```
Declare Function GetAdc(ch As Unsigned Byte) As Unsigned Integer
```

Parametri:

Unsigned Byte ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

Unsigned Integer – (0 – 1023)

Vrijednost analogno-digitalnog pretvarača [0 – 1023]

Podržani moduli:

CPU 168-RS232, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                     ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                      ' otvaranje komunikacije
InitAdc(DS1)                            ' inicijaliziraj ADC na portu IR1

Do                                      ' beskonačna petlja
   If GetAdc(IR1) > 256 Then            ' napon na ulazu DS1 je veći od 1.25 V?
      Relay1On()                        ' uključi relej 1
   Else                                 ' napon je manji ili jednak 1.25V?
      Relay1Off()                       ' isključi relej 1
   End If
Loop
EmoroDisconnect()                       ' zatvaranje komunikacije
End                                     ' kraj programa
```

Potprogram inicijalizira izlaze DC motora Motor-R i Motor-L. Pozivom potprograma **InitDcMotor()** SERVO_2, PWM_1 i PWM_2 port konfigurirani su kao izlaz, te ih koristi DC motor driver i na njima se ne može inicijalizirati servo motor.

Za korištenje DC motora potrebno je odspojiti hardver s portova SERVO_2, PWM_1 i PWM_2.

Nakon inicijalizacije DC motori su postavljeni na brzinu „0“.

```
Declare Sub InitDcMotor()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitDcMotor()                               ' inicijaliziraj Motor-L i Motor-R

Do                                          ' beskonačna petlja
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogram zaustavlja DC motor driver.

```
Declare Sub StopDcMotor()
```

Parametri:

nema

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitDcMotor()                               ' inicijaliziraj Motor-L i Motor-R
SetLeftDcMotor(FORWARD, 170)                ' postavi lijevi motor max naprijed
SetRightDcMotor(FORWARD, 170)               ' postavi desni motor max naprijed

Do                                          ' beskonačna petlja
   If Input1() = 1 Then                     ' pritisnuta tipka 1?
      StopDcMotor()                         ' zaustavi DC motore
   End If
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogram postavlja novu brzinu desnog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore **InitDcMotor()** potprogramom.

```
Declare Sub SetRightDcMotor(direction As Unsigned Byte, speed As Unsigned Byte)
```

Parametri:

Unsigned Byte direction – smjer DC motora [FORWARD, REVERSE, STOP_DC]

Unsigned Byte speed – brzina DC motora [0 – 170]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitDcMotor()                               ' inicijaliziraj Motor-L i Motor-R

Do                                          ' beskonačna petlja
   SetRightDcMotor(FORWARD, 170)            ' postavi desni motor max naprijed
   Sleep 1000                               ' čekaj 1000 mS
   SetRightDcMotor(REVERSE, 170)            ' postavi desni motor max nazad
   Sleep 1000                               ' čekaj 1000 mS
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Potprogram postavlja novu brzinu lijevog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore **InitDcMotor()** potprogramom.

```
Declare Sub SetLeftDcMotor(direction As Unsigned Byte, speed As Unsigned Byte)
```

Parametri:

Unsigned Byte direction	– smjer DC motora	[FORWARD, REVERSE, STOP_DC]
Unsigned Byte speed	– brzina DC motora	[0 – 170]

Povratna vrijednost:

nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije
InitDcMotor()                               ' inicijaliziraj Motor-L i Motor-R

Do                                          ' beskonačna petlja
   SetLeftDcMotor(FORWARD, 170)             ' postavi lijevi motor max naprijed
   Sleep 1000                               ' čekaj 1000 mS
   SetLeftDcMotor(REVERSE, 170)             ' postavi lijevi motor max nazad
   Sleep 1000                               ' čekaj 1000 mS
Loop
EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Funkcija mijenja ime bluetooth modula.

```
Declare Sub BluetoothChangeName(new_name As String)
```

Parametri:

String new_name – Novo ime bluetooth modula.

Povratna vrijednost:

nema

Podržani moduli: CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije

BluetoothChangeName("My_New_Name")          ' promjeni ime u „My_New_Name“

EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

Funkcija mijenja lozinku bluetooth modula.

```
Declare Sub BluetoothChangePasskey(new_passkey As String)
```

Parametri:

String new_passkey – Nova lozinka bluetooth modula.

Povratna vrijednost:

nema

Podržani moduli: CPU Bluetooth

Primjer:

```
#Include once "emoro.bi"                    ' dodjeljivanje EMoRo biblioteke
#Include "fbgfx.bi"                         ' dodjeljivanje fbgfx biblioteke

EmoroAutoConnect()                          ' otvaranje komunikacije

BluetoothChangePasskey("2323")              ' promjeni lozinku u „2323“

EmoroDisconnect()                           ' zatvaranje komunikacije
End                                         ' kraj programa
```

10

10.6 *Popis funkcija u AVR Studio-u*

Prototip funkcije	Opis funkcije
void InitIO(void);	Inicijalizacija ulaza i izlaza
unsigned char Relay1State(void);	Čitanje stanja releja 1
unsigned char Relay2State(void);	Čitanje stanja releja 2
unsigned char Relay3State(void);	Čitanje stanja releja 3
unsigned char Relay4State(void);	Čitanje stanja releja 4
void Relay1On(void);	Uključivanje releja 1
void Relay2On(void);	Uključivanje releja 2
void Relay3On(void);	Uključivanje releja 3
void Relay4On(void);	Uključivanje releja 4
void Relay1Off(void);	Isključivanje releja 1
void Relay2Off(void);	Isključivanje releja 2
void Relay3Off(void);	Isključivanje releja 3
void Relay4Off(void);	Isključivanje releja 4
void Relay1Toggle(void);	Promjena stanja releja 1
void Relay2Toggle(void);	Promjena stanja releja 2
void Relay3Toggle(void);	Promjena stanja releja 3
void Relay4Toggle(void);	Promjena stanja releja 4
unsigned char Input1(void);	Čitanje ulaza 1 i tipkala 1
unsigned char Input2(void);	Čitanje ulaza 2 i tipkala 2
unsigned char Input3(void);	Čitanje ulaza 3 i tipkala 3
unsigned char Input4(void);	Čitanje ulaza 4 i tipkala 4
unsigned char InputIR1(void);	Čitanje ulaza IR1
unsigned char InputDS1(void);	Čitanje ulaza DS1

unsigned char InputSERVO_1(void);	Čitanje ulaza SERVO_1
unsigned char InputSERVO_2(void);	Čitanje ulaza SERVO_2
unsigned char InputPWM_1(void);	Čitanje ulaza PWM_1
unsigned char InputPWM_2(void);	Čitanje ulaza PWM_2
int InitServo(unsigned char ch);	Inicijalizacija servo motora
int StopServo(unsigned char ch);	Zaustavljanje servo motora
int GetServoState(unsigned char ch);	Čitanje stanja servo motora
int SetServo(unsigned char ch, int position);	Postavljanje pozicije servo motora
int GetServo(unsigned char ch);	Čitanje pozicije servo motora
int ConfigServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed, int position);	Konfiguracija brzina servo motora
int SetServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed);	Postavljanje servo motora na konfiguriranu brzinu
void InitDcMotor(void);	Inicijalizacija DC motora
void StopDcMotor(void);	Zaustavljanje DC motora
unsigned char GetDcMotorState(void);	Čitanje stanja DC motora
int SetRightDcMotor(unsigned char direction, unsigned char speed);	Postavljanje brzine i smjera desnog DC motora
int SetLeftDcMotor(unsigned char direction, unsigned char speed);	Postavljanje brzine i smjera lijevog DC motora
int GetRightDcMotor(unsigned char* direction, unsigned char* speed);	Čitanje brzine i smjera desnog DC motora
int GetLeftDcMotor(unsigned char* direction, unsigned char* speed);	Čitanje brzine i smjera lijevog DC motora
int SetDcMotorAcceleration(int acc);	Postavljanje akceleracije DC motora
int GetDcMotorAcceleration(void);	Čitanje akceleracije DC motora
void InitTimer(void);	Inicijalizacija hardverskog brojača
int InitDs1820(unsigned char ch);	Inicijalizacija temp. senzora DS1820.
int GetDs1820(unsigned char ch);	Čitanje temp. senzora DS1820

int InitAdc(unsigned char ch);	Inicijalizacija AD pretvarača
int StopAdc(unsigned char ch);	Zaustavljanje AD pretvarača
int GetAdcState(unsigned char ch);	Čitanje stanja AD pretvarača
int GetAdc(unsigned char ch);	Čitanje AD pretvarača
int InitUart(unsigned long baudrate);	Inicijalizacija RS232 komunikacije
int UartPutChar(unsigned char c);	Slanje jednog byte-a putem RS232 ili bluetooth komunikacije
unsigned char UartGetChar(void);	Čitanje jednog byte-a iz dolaznog RS232 ili bluetooth međuspremnika
unsigned char UartTestHit(void);	Čitanje stanja dolaznog RS232 ili bluetooth međuspremnika
void UartFlushRx(void);	Brisanje dolaznog RS232 ili bluetooth međuspremnika
void UartFlushTx(void);	Brisanje odlaznog međuspremnika
void InitBluetooth(void);	Inicijalizacija bluetooth komunikacije
int BluetoothChangeName(char* new_name);	Mjenjanje imena bluetooth modula
char* BluetoothGetName(void);	Čitanje imena bluetooth modula
int BluetoothChangePasskey(char* new_passkey);	Mjenjanje lozinke bluetooth modula
unsigned char BluetoothConnection(void);	Čitanje stanja bluetooth veze
unsigned char BluetoothT1(void);	Čitanje T1 tipke bluetooth modula
void BluetoothLedOn(void);	Uključivanje STATUS LED-a bluetooth modula
void BluetoothLedOff(void);	Isključivanje STATUS LED-a bluetooth modula
void BluetoothLedToggle(void);	Promjena stanja STATUS LED-a bluetooth modula

Funkcija inicijalizira ulaze i izlaze EMoRo sučelja i poziva se prije drugih inicijalizacija u programu. Nakon provedene funkcije releji su postavljeni kao izlazi, a SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2 i DS1 kao TTL ulazi. Inicijalizacija servo motora, analogno-digitalnog pretvarača, DC motora, temperaturnog senzora može promijeniti postavke ovih ulaza, te je zbog toga nužno prvo provesti **InitIO();** a nakon toga ostale inicijalizacije u programu.

```
void InitIO(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   while(1){                           /* beskonačna petlja                   */
   }
}
```

10

Funkcije čitaju stanje relejnih izlaza. Za čitanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom.

```
unsigned char Relay1State(void);
unsigned char Relay2State(void);
unsigned char Relay3State(void);
unsigned char Relay4State(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje relejnog izlaza

(0) – relej isključen

(1) – relej uključen

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>                /* dodjeljivanje EMoRo biblioteke   */

int main(void){                            /* glavni program                   */
   InitIO();                               /* inicijaliziraj ulaze i izlaze    */
   while(1){                               /* beskonačna petlja                */
      if(Relay1State() == 1) Relay1Off(); /* relej 1 uključen? Isključi relej */
      if(Relay2State() == 1) Relay2Off(); /* relej 2 uključen? Isključi relej */
      if(Relay3State() == 1) Relay3Off(); /* relej 3 uključen? Isključi relej */
      if(Relay4State() == 1) Relay4Off(); /* relej 4 uključen? Isključi relej */
   }
}
```

Funkcije uključuju relejne izlaze. Za uključivanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom.

```
void Relay1On(void);
void Relay2On(void);
void Relay3On(void);
void Relay4On(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   while(1){                       /* beskonačna petlja                     */
      Relay1On();                  /* uključi relej 1                       */
      Relay2On();                  /* uključi relej 2                       */
      Relay3On();                  /* uključi relej 3                       */
      Relay4On();                  /* uključi relej 4                       */
   }
}
```

Funkcije isključuju relejne izlaze. Za isključivanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom.

```
void Relay1Off(void);
void Relay2Off(void);
void Relay3Off(void);
void Relay4Off(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke         */

int main(void){                    /* glavni program                         */
   InitIO();                       /* inicijaliziraj ulaze i izlaze          */
   while(1){                       /* beskonačna petlja                      */
      Relay1Off();                 /* isključi relej 1                       */
      Relay2Off();                 /* isključi relej 2                       */
      Relay3Off();                 /* isključi relej 3                       */
      Relay4Off();                 /* isključi relej 4                       */
   }
}
```

Funkcije mijenjaju stanje relejnih izlaza. Za promjenu stanja relejnih izlaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom. Pozivom funkcije uključeni relejni izlaz će se isključiti, a isključeni relejni izlaz uključiti.

```
void Relay1Toggle(void);
void Relay2Toggle(void);
void Relay3Toggle(void);
void Relay4Toggle(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h]        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   while(1){                       /* beskonačna petlja                     */
      Relay1Toggle();              /* promijeni stanje releja 1             */
      Relay2Toggle();              /* promijeni stanje releja 2             */
      Relay3Toggle();              /* promijeni stanje releja 3             */
      Relay4Toggle();              /* promijeni stanje releja 4             */
   }
}
```

Funkcije čitaju ulaze (1, 2, 3, 4) i tipke (1, 2, 3, 4). Za čitanje ulaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom. LED svakog ulaza prikazuje trenutno stanje ulaza. Ako LED svijetli, ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli, ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

```
unsigned char Input1(void);
unsigned char Input2(void);
unsigned char Input3(void);
unsigned char Input4(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje ulaza (ulazi 1 – 4)

(0) – puštena tipka ili neaktivan ulaz (Input High – 5V)

(1) – pritisnuta tipka ili aktiviran ulaz (Input Low – 0V)

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   while(1){                           /* beskonačna petlja                   */
      if(Input1() == 1) Relay1On();    /* ulaz 1 aktivan, uključi relej 1     */
      if(Input2() == 1) Relay2On();    /* ulaz 2 aktivan, uključi relej 2     */
      if(Input3() == 1) Relay3On();    /* ulaz 3 aktivan, uključi relej 3     */
      if(Input4() == 1) Relay4On();    /* ulaz 4 aktivan, uključi relej 4     */
   }
}
```

Funkcije čitaju ulaze IR1, IR2 i DS1, SERVO_1, SERVO_2, PWM_1 i PWM_2. Za čitanje ulaza potrebno je inicijalizirati ulaze i izlaze sučelja **InitIO();** funkcijom. LED ulaza IR1 i IR2 prikazuje trenutno stanje ulaza. Ako LED svijetli ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli, ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

Navedene ulaze moguće je čitati samo u slučaju da na njima nije inicijalizirana druga funkcionalnost. Ulazima se može koristiti servo_driver, adc_driver, dc_motor_driver i ds1820_driver i ako je ovim portovima inicijalizirana druga funkcionalnost, ne mogu se čitati kao TTL ulazi.

Npr. obavljanjem **InitServo(SERVO_1); InitAdc(DS1);** SERVO_1 i DS1 će biti konfigurirani za funkcionalnost servo motora i analognog ulaza, te se funkcije **InputSERVO_1();** i **InputDS1();** ne mogu koristiti.

```
unsigned char InputIR1(void);
unsigned char InputIR2(void);
unsigned char InputDS1(void);
unsigned char InputSERVO_1(void);
unsigned char InputSERVO_2(void);
unsigned char InputPWM_1(void);
unsigned char InputPWM_2(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje ulaza [IR1, IR2, DS1, SERVO_1, SERVO_2, PWM_1, PWM_2]

(0) – ulaz neaktivan (Input High – 5V)

(1) – ulaz aktiviran (Input Low – 0V)

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>                /* dodjeljivanje EMoRo biblioteke   */

int main(void){                            /* glavni program                   */
   InitIO();                               /* inicijaliziraj ulaze i izlaze    */
   while(1){                               /* beskonačna petlja                */
      if(InputIR1()==1)      Relay1On();   /* IR1 aktivan, uključi relej 1     */
      if(InputIR2()==1)      Relay2On();   /* IR2 aktivan, uključi relej 2     */
      if(InputDS1()==1)      Relay3On();   /* DS1 aktivan, uključi relej 3     */
      if(InputSERVO_1()==1)  Relay4On();   /* SERVO_1 aktivan, uključi relej 4 */
      if(InputSERVO_2()==1)  Relay1Off();  /* SERVO_2 aktivan,isključi relej 1 */
      if(InputPWM_1()==1)    Relay2Off();  /* PWM_1 aktivan, isključi relej 2  */
      if(InputPWM_2()==1)    Relay3Off();  /* PWM_2 aktivan, isključi relej 3  */
   }
}
```

Funkcija inicijalizira izlaz servo motora na kojemu servo_driver generira servo pulse. Sučelje EMoRo podržava 7 servo motora na izlazima SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2 i DS1, a za upravljanje servo motorima potrebno obaviti inicijalizaciju servo izlaza. Nakon inicijalizacije servo motora zadano stanje izlaza postavljeno je na vrijednost 1500 što odgovara impulsu 1.5 mS. Treba obratiti pozornost da se nakon inicijalizacije servo motora na ovim portovima ne može koristiti inicijalizacija za drugu funkcionalnost, kao ni funkcije koje čitaju port kao ulaz.

Npr. pozivom **InitServo(IR1);** IR1 port konfiguriran je kao servo izlaz, te se ne može koristiti funkcija čitanja ulaza **InputIR1**.

DC motor driver upotrebljava zajedničke portove SERVO_2, PWM_1, PWM_2, i u slučaju da se jedan od navedenih izlaza konfigurira kao servo motor tada se DC motori ne mogu koristiti.

Inicijalizirani servo_driver generira servopulse u intervalu [0,5 mS – 2,5 mS] svakih 20mS.

```
int InitServo(unsigned char ch);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Servo izlaz uspiješno inicijaliziran

(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1 – 7]

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   InitServo(SERVO_1);             /* inicijaliziraj servo motor na SERVO_1 */
   InitServo(SERVO_2);             /* inicijaliziraj servo motor na SERVO_2 */
   InitServo(PWM_1);               /* inicijaliziraj servo motor na PWM_1   */
   InitServo(PWM_2);               /* inicijaliziraj servo motor na PWM_2   */
   InitServo(IR1);                 /* inicijaliziraj servo motor na IR1     */
   InitServo(IR2);                 /* inicijaliziraj servo motor na IR2     */
   InitServo(DS1);                 /* inicijaliziraj servo motor na DS1     */
   while(1){                       /* beskonačna petlja                     */
   }
}
```

Funkcija isključuje izlaz servo motora i konfigurira port kao ulaz.

```
int StopServo(unsigned char ch);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Servo izlaz uspiješno zaustavljen

(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1-7]

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitServo(SERVO_1);              /* inicijaliziraj servo motor na SERVO_1  */
   SetServo(SERVO_1, 1000);         /* postavi SERVO_1 na poziciju 1000 (1mS) */
   while(1){                        /* beskonačna petlja                      */
      if(Input1() == 1){            /* pritisnuta tipka 1?                    */
         StopServo(SERVO_1);        /* zaustavi servo motor SERVO_1           */
      }
   }
}
```

Funkcija čita stanje servo izlaza (servo izlaz pokrenut, servo izlaz zaustavljen).

```
int GetServoState(unsigned char ch);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(1) – Servo izlaz je inicijaliziran – (Servo pokrenut)
(0) – Servo izlaz nije inicijaliziran – (Servo zaustavljen)
(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1-7]

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitServo(SERVO_1);              /* inicijaliziraj servo motor na SERVO_1  */
   while(1){                        /* beskonačna petlja                      */

         /* pritisnuta tipka ulaza 1, a SERVO_1 je inicijaliziran?           */
      if((Input1()==1) && (GetServoState(SERVO_1)==1))
         StopServo(SERVO_1);        /* zaustavi servo na portu SERVO_1        */

         /* puštena tipka ulaza 1, a SERVO_1 nije inicijaliziran?            */
      if((Input1()==0) && (GetServoState(SERVO_1)==0))
         InitServo(SERVO_1);        /* inicijaliziraj servo motor SERVO_1     */
   }
}
```

Funkcija postavlja novu poziciju servo motora.

```
int SetServo(unsigned char ch, int position);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

int position – nova pozicija servo motora [500 - 2500] µs

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Servo izlaz uspješno postavljen na novu vrijednost
(-1) – Greška: Pozicija nije u intervalu [500 – 2500]
(-2) – Greška: Parametar servo motora „ch“ nije u intervalu [1 – 7]
(-3) – Greška: Servo motor nije inicijaliziran

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitServo(SERVO_1);          /* inicijaliziraj servo motor na SERVO_1     */
   while(1){                    /* beskonačna petlja                         */
      SetServo(SERVO_1, 1000);  /* postavi motor SERVO_1 pulse 1000 uS (1mS) */
   }
}
```

Funkcija vraća poziciju servo motora.

```
int GetServo(unsigned char ch);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int (500 – 2500) – Pozicija servo motora

(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1 – 7]

(-2) – Greška: Servo motor nije inicijaliziran

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   InitServo(SERVO_1);             /* inicijaliziraj servo motor na SERVO_1 */
   InitServo(SERVO_2);             /* inicijaliziraj servo motor na SERVO_2 */
   while(1){                       /* beskonačna petlja                     */

            /* postavi SERVO_2 na poziciju 2000 (2mS pulse)                 */
      SetServo(SERVO_2, 2000);

            /* postavi SERVO_1 na poziciju na kojoj se nalazi SERVO_2       */
      SetServo(SERVO_1, GetServo(SERVO_2));
   }
}
```

Funkcija konfigurira 5 servo brzina za smjer „FORWARD“ i 5 servo brzina za smjer „REVERSE“, te ih pohranjuje u EEPROM kako bi brzine servo motora ostale pohranjene i nakon isključivanja napona napajanja. Funkcija **ConfigServoSpeed();** predviđena je za pohranjivanje brzina servo motora s konstantnom rotacijom, ali se može koristiti i za pohranjivanje vrijednosti pulsa pozicijskih servo motora. Servo motori se postavljaju na konfigurirane brzine funkcijom **SetServoSpeed();**

```
int ConfigServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed,
                     int position);
```

Parametri:

unsigned char ch	– servo motor	[SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]
unsigned char dir	– smjer servo motora [FORWARD, REVERSE]	
unsigned char speed	– konfiguracijska brzina [1 – 5]	
int position	– brzina servo motora [500 – 2500] µs	

Povratna vrijednost:

int Rezultat funkcije:

(0) – Servo brzina uspješno konfigurirana
(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1 – 7]
(-2) – Greška: Brzina „speed“ nije u intervalu [1 – 5]
(-3) – Greška: Smjer nije u intervalu [FORWARD, REVERSE]
(-4) – Greška: Pozicija nije u intervalu [500 – 2500]

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   InitServo(SERVO_1);             /* inicijaliziraj servo motor na SERVO_1 */
   InitServo(SERVO_2);             /* inicijaliziraj servo motor na SERVO_2 */
   while(1){                       /* beskonačna petlja                     */

            /* konfiguriraj 5 brzina FORWARD za motor SERVO_1               */
      ConfigServoSpeed(SERVO_1, FORWARD, 1, 1450);
      ConfigServoSpeed(SERVO_1, FORWARD, 2, 1400);
      ConfigServoSpeed(SERVO_1, FORWARD, 3, 1350);
      ConfigServoSpeed(SERVO_1, FORWARD, 4, 1300);
      ConfigServoSpeed(SERVO_1, FORWARD, 5, 1000);

            /* konfiguriraj 5 brzina FORWARD za motor SERVO_2               */
      ConfigServoSpeed(SERVO_2, FORWARD, 1, 1550);
      ConfigServoSpeed(SERVO_2, FORWARD, 2, 1600);
      ConfigServoSpeed(SERVO_2, FORWARD, 3, 1650);
      ConfigServoSpeed(SERVO_2, FORWARD, 4, 1700);
      ConfigServoSpeed(SERVO_2, FORWARD, 5, 2000);
   }
}
```

Funkcija postavlja servo motor na konfigurirane brzine. Za uspješno postavljanje brzina servo motora potrebno je konfigurirati brzine **ConfigServoSpeed();** funkcijom. Konfiguraciju brzina potrebno je obaviti samo jedanput jer se brzine pohranjuju u EEPROM memoriju i ostaju pohranjene i nakon isključivanja sučelja s napona napajanja. Prije postavljanja servo motora na konfiguriranu brzinu potrebno je inicijalizirati servo izlaz **InitServo();** funkcijom.

```
int SetServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed);
```

Parametri:

unsigned char ch – servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]
unsigned char dir – smjer servo motora [FORWARD, REVERSE, STOP]
unsigned char speed – brzina servo motora [1 – 5]

Povratna vrijednost:

Int – Rezultat funkcije:
(0) – Servo brzina uspješno postavljena
(-1) – Greška: Servo brzina „speed“ nije konfigurirana.
(-2) – Greška: Parametar servo motora „ch“ nije u intervalu [1 – 7]
(-3) – Greška: Servo motor nije inicijaliziran
(-4) – Greška: Servo brzina „speed“ nije u intervalu [1 – 5]
(-5) – Greška: Smjer motora „dir“ nije u intervalu [FORWARD, REVERSE, STOP]

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitServo(SERVO_1);          /* inicijaliziraj servo motor na SERVO_1     */
   InitServo(SERVO_2);          /* inicijaliziraj servo motor na SERVO_2     */
   while(1){                    /* beskonačna petlja                         */

            /* postavi SERVO_1 brzinom 1 FORWARD                             */
      SetServoSpeed(SERVO_1, FORWARD, 1);

            /* postavi SERVO_2 brzinom 1 FORWARD                             */
      SetServoSpeed(SERVO_2, FORWARD, 1);
   }
}
```

Funkcija inicijalizira izlaze DC motora Motor-R i Motor-L. Pozivom funkcije **InitDcMotor();** SERVO_2, PWM_1 i PWM_2 port konfigurirani su kao izlaz, te ih koristi DC motor driver. Upozoravamo da se navedeni portovi ne mogu konfigurirati za drugu funkcionalnost, niti čitati kao ulazi.

Npr. inicijalizacijom DC motora SERVO_2, PWM_1 i PWM_2 portovi ne mogu se koristiti kao ulazi ili servo motor, ali se uz inicijalizaciju DC motora mogu inicijalizirati servo motori na portovima SERVO_1, IR1, IR2 i DS1.

Za korištenje DC motora potrebno je odspojiti hardver s portova SERVO_2, PWM_1 i PWM_2.

Nakon inicijalizacije DC motori postavljeni su na brzinu „0“.

```
void InitDcMotor(void);
```

Parametri:
nema

Povratna vrijednost:
nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>  /* dodjeljivanje EMoRo biblioteke                */

int main(void){              /* glavni program                                */
   InitIO();                 /* inicijaliziraj ulaze i izlaze                 */
   InitDcMotor();            /* inicijaliziraj DC motore Motor-L i Motor-R    */
   while(1){                 /* beskonačna petlja                             */
   }
}
```

10

Funkcija zaustavlja DC motor driver.

```
void StopDcMotor(void);
```

Parametri:
nema

Povratna vrijednost:
nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitDcMotor();                   /* inicijaliziraj Motor-L i Motor-R       */
   SetLeftDcMotor(FORWARD, 170);    /* postavi Motor-L max naprijed           */
   SetRightDcMotor(FORWARD, 170);   /* postavi Motor-R max naprijed           */

   while(1){                        /* beskonačna petlja                      */
      if(Input1() == 1)             /* pritisnuta tipka 1?                    */
         StopDcMotor();             /* zaustavi DC motor driver               */
   }
}
```

Funkcija čita stanje DC motor drivera. (DC motor driver pokrenut, DC motor driver zaustavljen).

```
unsigned char GetDcMotorState(void);
```

Parametri:

nema

Povratna vrijednost:

unsigned char – Stanje DC motor driver-a:

(1) – DC motor driver inicijaliziran - (DC motori pokrenuti)

(0) – DC motor driver nije inicijaliziran - (DC motori zaustavljeni)

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */

   while(1){                           /* beskonačna petlja                   */

         /* pritisnuta tipka ulaza 1 i DC motori su inicijalizirani?          */
      if((Input1()==1) && (GetDcMotorState() == 1))
         StopDcMotor();                /* zaustavi DC motore                  */

         /* puštena tipka ulaza 1 i DC motori nisu inicijalizirani?           */
      if((Input1()==0) && (GetDcMotorState() == 0))
         InitDcMotor();                /* inicijaliziraj DC motore            */
   }
}
```

Funkcija postavlja novu brzinu desnog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom.

```
int SetRightDcMotor(unsigned char direction, unsigned char speed);
```

Parametri:

unsigned char direction – smjer DC motora [FORWARD, REVERSE, STOP]

unsigned char speed – brzina DC motora [0 – 170]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno postavljena brzina i smjer desnog DC motora

(-1) – Greška: Brzina motora nije u intervalu [0 – 170]

(-2) – Greška: Smjer motora nije u intervalu [FORWARD, REVERSE, STOP]

(-3) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   InitDcMotor();                      /* inicijaliziraj Motor-L i Motor-R    */
   while(1){                           /* beskonačna petlja                   */
      SetRightDcMotor(FORWARD, 170);   /* postavi Motor-R max naprijed        */
      _delay_ms(1000);                 /* čekaj 1000 ms                       */
      SetRightDcMotor(REVERSE, 170);   /* postavi Motor-R max nazad           */
      _delay_ms(1000);                 /* čekaj 1000 ms                       */
   }
}
```

Funkcija postavlja novu brzinu lijevog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom.

```
int SetLeftDcMotor(unsigned char direction, unsigned char speed);
```

Parametri:

unsigned char direction – smjer DC motora [FORWARD, REVERSE, STOP]

unsigned char speed – brzina DC motora [0 – 170]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno postavljena brzina i smjer lijevog DC motora

(-1) – Greška: Brzina motora nije u intervalu [0 – 170]

(-2) – Greška: Smjer motora nije u intervalu [FORWARD, REVERSE, STOP]

(-3) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   InitDcMotor();                      /* inicijaliziraj Motor-L i Motor-R    */
   while(1){                           /* beskonačna petlja                   */
      SetLeftDcMotor(FORWARD, 170);    /* postavi Motor-L max naprijed        */
      _delay_ms(1000);                 /* čekaj 1000 ms                       */
      SetLeftDcMotor(REVERSE, 170);    /* postavi Motor-L max nazad           */
      _delay_ms(1000);                 /* čekaj 1000 ms                       */
   }
}
```

Funkcija čita postavljenu brzinu i smjer desnog DC motora. Prije čitanja brzine i smjera DC motora potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom.

```
int GetRightDcMotor(unsigned char* direction, unsigned char* speed);
```

Parametri:

unsigned char* direction – smjer DC motora [FORWARD, REVERSE, STOP]

unsigned char* speed – brzina DC motora [0 – 170]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno pročitana brzina i smjer desnog DC motora

(-1) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke         */

unsigned char my_speed, my_direction;

int main(void){                    /* glavni program                         */
   InitIO();                       /* inicijaliziraj ulaze i izlaze          */
   InitDcMotor();                  /* inicijaliziraj Motor-L i Motor-R       */
   SetRightDcMotor(FORWARD, 100);  /* postavi Motor-R na brzinu 100 naprijed */
   while(1){                       /* beskonačna petlja                      */

            /* čitaj smjer i brzinu Motor-R                                  */
      GetRightDcMotor(&my_direction, &my_speed);
            /* postavi Motor-L na brzinu i smjer kao Motor-R                 */
      SetLeftDcMotor(my_direction, my_speed);
   }
}
```

Funkcija čita postavljenu brzinu i smjer lijevog DC motora. Prije čitanja brzine i smjera DC motora potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom.

```
int GetLeftDcMotor(unsigned char* direction, unsigned char* speed);
```

Parametri:

unsigned char* direction – smjer DC motora [FORWARD, REVERSE, STOP]

unsigned char* speed – brzina DC motora [0 – 170]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno pročitana brzina i smjer lijevog DC motora

(-1) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke         */

unsigned char my_speed, my_direction;

int main(void){                    /* glavni program                         */
   InitIO();                       /* inicijaliziraj ulaze i izlaze          */
   InitDcMotor();                  /* inicijaliziraj Motor-L i Motor-R       */
   SetLeftDcMotor(FORWARD, 100);   /* postavi Motor-L na brzinu 100 naprijed */
   while(1){                       /* beskonačna petlja                      */

            /* čitaj smjer i brzinu Motor-L                                   */
      GetLeftDcMotor(&my_direction, &my_speed);
            /* postavi Motor-R na brzinu i smjer kao Motor-L                  */
      SetRightDcMotor(my_direction, my_speed);
   }
}
```

Funkcija postavlja novo ubrzanje DC motora. Prije postavljanja ubrzanja potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom. DC motor driver ne dopušta naglu promjenu smjera i brzine motora zbog zaštite DC motora, reduktora i sučelja, te će ubrzavati motore u skladu s podešenim ubrzanjem.

Zadano ubrzanje (acc) nakon inicijalizacije DC motora je vrijednost 128, a računa se prema formuli:

Ubrzanje(µs) = 32 x acc

Zadano ubrzanje DC motora nakon inicijalizacije (acc=128):

Promjena brzine svakih: 32(µs) x 128 = 4096(µs)

Maksimalnu brzinu DC motori će postići za: 4,096(ms) x 170 = 969,32(ms)

Maksimalno ubrzanje DC motora (acc=32):

Promjena brzine svakih: 32(µs) x 32 = 1024(µs)

Maksimalnu brzinu DC motori će postići za: 1,024(ms) x 170 = 174,08(ms)

Minimalno ubrzanje DC motora (acc=1024):

Promjena brzine svakih: 32(µs) x 1024 = 32768(µs)

Maksimalnu brzinu DC motori će postići za: 32,768(ms) x 170 = 5570,56(ms)

```
int SetDcMotorAcceleration(int acc);
```

Parametri:

int acc – ubrzanje DC motora [32 – 1023]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno postavljeno novo ubrzanje DC motora

(-1) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitDcMotor();                   /* inicijaliziraj Motor-L i Motor-R       */
   SetDcMotorAcceleration(32);      /* postavi ubrzanje: 32uS * 32 = 1024 uS  */
   SetLeftDcMotor(FORWARD, 170);    /* postavi Motor-L na max brzinu naprijed */
   while(1){                        /* beskonačna petlja                      */

   /* Motor-L dostiže max brzinu nakon: 1,024 [mS] * 170 [speed] = 174,08 mS */
   }
}
```

Funkcija čita ubrzanje DC motora. Prije čitanja ubrzanja potrebno je inicijalizirati DC motore **InitDcMotor();** funkcijom.

```
int GetDcMotorAcceleration(void);
```

Parametri:

nema

Povratna vrijednost:

int – Rezultat funkcije:

(32 – 1023) – Ubrzanje DC motora [32 – 1023]

(-1) – Greška: DC motori nisu inicijalizirani

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitDcMotor();                   /* inicijaliziraj Motor-L i Motor-R       */
   if(GetDcMotorAcceleration() != 32){    /* ubrzanje nije 32? (max)          */
      SetDcMotorAcceleration(32);   /* postavi ubrzanje: 32uS * 32 = 1024 uS  */
   }
   SetLeftDcMotor(FORWARD, 170);    /* postavi Motor_L na max brzinu naprijed */
   while(1){                        /* beskonačna petlja                      */

   /* Motor-L dostiže max brzinu nakon: 1,024 [mS] * 170 [speed] = 174,08 mS */
   }
}
```

Funkcija inicijalizira hardverski brojač koji povećava 10 varijabla svaku 1mS. Varijable brojača su unsigned int i mogu sadržavati vrijednosti [0 – 65535]. U slučaju inicijalizacije brojača, DC motori se ne mogu koristiti jer njihova inicijalizacija, zbog interne arhitekture mikrokontrolera, mora ugasiti hardverski brojač.

```
void InitTimer(void);
```

Varijable brojača:
extern volatile unsigned int timer_1; extern volatile unsigned int timer_2;
extern volatile unsigned int timer_3; extern volatile unsigned int timer_4;
extern volatile unsigned int timer_5; extern volatile unsigned int timer_6;
extern volatile unsigned int timer_7; extern volatile unsigned int timer_8;
extern volatile unsigned int timer_9; extern volatile unsigned int timer_10;

Parametri:
nema

Povratna vrijednost:
nema

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke        */

int main(void){                    /* glavni program                        */
   InitIO();                       /* inicijaliziraj ulaze i izlaze         */
   InitTimer();                    /* inicijaliziraj brojač  (1mS)          */
   while(1){                       /* beskonačna petlja                     */
      if(timer_1 >= 1000){         /* timer_1 je nabrojao 1000mS? (1s)      */
         Relay1Toggle();           /* okreni stanje releja 1                */
         timer_1 = 0;              /* postavi timer_1 na „0“                */
      }
   }
}
```

Funkcija inicijalizira temperaturni senzor DS1820. Sučelje EMoRo podržava 7 temperaturnih senzora na portovima SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2 i DS1.

DC motor driver koristi zajedničke portove SERVO_2, PWM_1 i PWM_2, te u slučaju inicijalizacije temperaturnog senzora na jedan od navedenih portova, DC motori ne mogu se koristiti.

```
int InitDs1820(unsigned char ch);
```

Parametri:

unsigned char ch – senzor DS1820 [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno inicijaliziran temperaturni senzor DS1820

(-1) – Greška: Senzor nedostupan na „ch“ portu

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>          /* dodjeljivanje EMoRo biblioteke      */

int main(void){                      /* glavni program                      */
   InitIO();                         /* inicijaliziraj ulaze i izlaze       */
   InitDs1820(IR1);                  /* inicijaliziraj DS1820 na portu IR1  */
   while(1){                         /* beskonačna petlja                   */
   }
}
```

10

Funkcija čita temperaturni senzor DS1820 sa razlučivosti od 0,5°C. Prije čitanja temperature potrebno je inicijalizirati temperaturni senzor **InitDs1820();** funkcijom.

```
int GetDs1820(unsigned char ch);
```

Parametri:

unsigned char ch – senzor DS1820 [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(-550 – 1250) – temperatura x 10 (od -55°C do 125°C)

(20000) – Greška: Senzor nedostupan na „ch“ portu

Podržani moduli: CPU 168-RS232, CPU 162-USB, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>      /* dodjeljivanje EMoRo biblioteke            */

int main(void){                  /* glavni program                            */
   InitIO();                     /* inicijaliziraj ulaze i izlaze             */
   InitDs1820(IR1);              /* inicijaliziraj DS1820 na portu IR1        */
   while(1){                     /* beskonačna petlja                         */
      if(GetDs1820(IR1) > 305)   /* temperatura je veća od 30,5 °C?           */
         Relay1On();             /* uključi relej 1                           */
      else                       /* temperatura je manja ili jednaka 30,5°C   */
         Relay1Off();            /* isključi relej 1                          */
   }
}
```

Funkcija inicijalizira analogno-digitalni pretvarač (ADC). Sučelje EMoRo podržava 3 analogno-digitalna pretvarača na ulazima IR1, IR2 i DS1, i prije čitanja ADC-a potrebno je obaviti inicijalizaciju **InitAdc();** funkcijom.

```
int InitAdc(unsigned char ch);
```

Parametri:

unsigned char ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Uspješno inicijaliziran analogno-digitalni pretvarač

(-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>          /* dodjeljivanje EMoRo biblioteke      */

int main(void){                      /* glavni program                      */
   InitIO();                         /* inicijaliziraj ulaze i izlaze       */
   InitAdc(IR1);                     /* inicijaliziraj ADC na portu IR1     */
   InitAdc(IR2);                     /* inicijaliziraj ADC na portu IR2     */
   InitAdc(DS1);                     /* inicijaliziraj ADC na portu DS1     */
   while(1){                         /* beskonačna petlja                   */
   }
}
```

Funkcija zaustavlja analogno-digitalni pretvarač (ADC).

```
int StopAdc(unsigned char ch);
```

Parametri:

unsigned char ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0) – ADC port uspješno zaustavljen

(-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>          /* dodjeljivanje EMoRo biblioteke      */

int main(void){                      /* glavni program                      */
   InitIO();                         /* inicijaliziraj ulaze i izlaze       */
   InitAdc(DS1);                     /* inicijaliziraj ADC na portu DS1     */
   while(1){                         /* beskonačna petlja                   */
      if(Input1() == 1)              /* pritisnuta tipka 1?                 */
         StopAdc(DS1);               /* zaustavi ADC na portu DS1           */
   }
}
```

Funkcija čita stanje analogno-digitalnog pretvarača (ADC pokrenut, ADC zaustavljen).

```
int GetAdcState(unsigned char ch);
```

Parametri:

unsigned char ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(1) – ADC ulaz je inicijaliziran - (ADC pokrenut)
(0) – ADC ulaz nije inicijaliziran - (ADC zaustavljen)
(-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   while(1){                        /* beskonačna petlja                      */

         /* pritisnuta tipka ulaza 1 i ADC ulaz IR1 je inicijaliziran?       */
      if((Input1()==1) && (GetAdcState(IR1) == 1))
         StopAdc(IR1);              /* zaustavi ADC na ulazu IR1              */

         /* puštena tipka ulaza 1 i ADC ulaz IR1 nije inicijaliziran?        */
      if((Input1()==0) && (GetAdcState(IR1) == 0))
         InitAdc(IR1);              /* inicijaliziraj ADC na ulazu IR1        */
   }
}
```

Funkcija čita analogno-digitalni pretvarač. Prije čitanja ADC-a potrebno je obaviti inicijalizaciju pretvarača **InitAdc();** funkcijom.

Napon na ulazu pretvarača računa se formulom

U = (5 / 1024) x GetAdc(ch)

```
int GetAdc(unsigned char ch);
```

Parametri:

unsigned char ch – ADC [IR1, IR2, DS1]

Povratna vrijednost:

int – Rezultat funkcije:

(0 – 1023) – Vrijednost analogno-digitalnog pretvarača [0 – 1023]
(-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]
(-2) – Greška: ADC nije inicijaliziran

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   InitAdc(DS1);                       /* inicijaliziraj ADC na portu DS1     */

   while(1){                           /* beskonačna petlja                   */
      if(GetAdc(DS1) > 256)            /* napon na DS1 je veći od 1.25 V?     */
         Relay1On();                   /* uključi relej 1                     */
      else                             /* napon je manji ili jednak 1.25V?    */
         Relay1Off();                  /* isključi relej 1                    */
   }
}
```

Funkcija inicijalizira serijsku komunikaciju (8 data bitova, 1 stop bit), te podržava brzinu komunikacije u intervalu [1200 – 57600] bps.

Za uspostavljanje komunikacije mikrokontrolerskog modula CPU 168-RS232 potrebno je inicijalizirati serijsku komunikaciju **InitUart();** funkcijom, a za uspostavljanje komunikacije CPU bluetooth modula potrebno je inicijalizirati bluetooth komunikaciju **InitBluetooth();** funkcijom.

```
int InitUart(unsigned long baudrate);
```

Parametri:

unsigned long baudrate – brzina komunikacije [1200 – 57600] bps

Povratna vrijednost:

int – Rezultat funkcije:

(0) – Serijska komunikacija uspješno inicijalizirana

(-1) – Greška: Parametar „baudrate“ nije u intervalu [1200 – 57600]

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      printf("Hello World\r\n");   /* pošalji pozdravnu poruku               */
   }
}
```

10

Funkcija šalje 1 byte putem RS232 ili Bluetooth komunikacije. Za slanje jednog bytea sa CPU 168-RS232 mikrokontrolerskog modula potrebno je inicijalizirati komunikaciju **InitUart();** funkcijom, a za slanje bytea putem bluetooth veze potrebno je inicijalizirati komunikaciju **InitBluetooth();** funkcijom.

```
int UartPutChar(unsigned char c);
```

Parametri:

unsigned char c – podatak za slanje [0 – 255]

Povratna vrijednost:

Int – Rezultat funkcije:

(0) – Podatak uspješno pohranjen u međuspremnik

(-1) – Greška: Komunikacija nije inicijalizirana

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer za CPU 168-RS232:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
   }
}
```

Primjer za CPU Bluetooth:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitBluetooth();             /* inicijaliziraj bluetooth                  */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
   }
}
```

Funkcija čita 1 byte iz međuspremnika bluetooth ili RS232 komunikacije. Za čitanje jednog bytea iz CPU 168-RS232 međuspremnika potrebno je inicijalizirati komunikaciju **InitUart();** funkcijom, a za čitanje jednog bytea iz bluetooth međuspremnika potrebno je inicijalizirati komunikaciju **InitBluetooth();** funkcijom.

UartGetChar(); funckija kao povratnu vrijednost uvijek vraća 1 byte, te je prije čitanja podatka iz međuspremnika potrebno provjeriti je li postoji primljeni podatak u međuspremniku funkcijom **UartTestHit();**

```
unsigned char UartGetChar(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Čitanje međuspremnika:

(0 – 255) – primljeni podatak

(0) – međuspremnik nema primljenog podatka

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer za CPU 168-RS232:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

unsigned char my_data;

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      if(UartTestHit() > 0){    /* provjeri podatake u dolaznom međuspremniku*/

         /* pročitaj 1 primljeni byte i pohrani ga u my_data varijablu       */
         my_data = UartGetChar();           }
   }
}
```

Primjer za CPU Bluetooth:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

unsigned char my_data;

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitBluetooth();             /* inicijaliziraj bluetooth                  */
   while(1){                    /* beskonačna petlja                         */
      if(UartTestHit() > 0){    /* provjeri podatake u dolaznom međuspremniku*/

         /* pročitaj 1 primljeni byte i pohrani ga u my_data varijablu       */
         my_data = UartGetChar();           }
   }
}
```

Funkcija provjerava međuspremnik primljenih podataka.

```
unsigned char UartTestHit(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje međuspremnika:

(0) – međuspremnik primljenih podataka je prazan

(1) – međuspremnik primljenih podataka nije prazan

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer za CPU 168-RS232:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      if(UartTestHit() > 0){    /* provjeri podatke u dolaznom međuspremniku */

            /* pročitaj 1 primljeni byte i vrati ga pošiljatelju             */
         UartPutChar( UartGetChar() );
      }
   }
}
```

Primjer za CPU Bluetooth:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitBluetooth();             /* inicijaliziraj bluetooth                  */
   while(1){                    /* beskonačna petlja                         */
      if(UartTestHit() > 0){    /* provjeri podatake u dolaznom međuspremniku*/

            /* pročitaj 1 primljeni byte i vrati ga pošiljatelju             */
         UartPutChar( UartGetChar() );
      }
   }
}
```

Funkcija briše sve dolazne podatke iz međuspremnika RS232 ili bluetooth komunikacije.

```
void UartFlushRx(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer za CPU 168-RS232:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartFlushRx();            /* obriši sve iz dolaznog međuspremnika      */
   }
}
```

Primjer za CPU Bluetooth:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitBluetooth();             /* inicijaliziraj bluetooth.                 */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartFlushRx();            /* obriši sve iz dolaznog međuspremnika      */
   }
}
```

Funkcija briše sve odlazne podatke iz međuspremnika RS232 ili bluetooth komunikacije.

```
void UartFlushTx(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU 168-RS232, CPU Bluetooth

Primjer za CPU 168-RS232:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitUart(38400);             /* inicijaliziraj RS232 na brzini 38400 bps  */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartFlushTx();            /* obriši sve iz odlaznog međuspremnika      */
      _delay_ms(1000);          /* čekaj 1000 mS                             */
   }
}
```

Primjer za CPU Bluetooth:

```
#include <inovatic\emoro.h>     /* dodjeljivanje EMoRo biblioteke            */

int main(void){                 /* glavni program                            */
   InitIO();                    /* inicijaliziraj ulaze i izlaze             */
   InitBluetooth();             /* inicijaliziraj bluetooth.                 */
   while(1){                    /* beskonačna petlja                         */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartPutChar(170);         /* pošalji broj 170 serijskom komunikacijom  */
      UartFlushTx();            /* obriši sve iz odlaznog međuspremnika      */
      _delay_ms(1000);          /* čekaj 1000 mS                             */
   }
}
```

Funkcija inicijalizira bluetooth komunikaciju.

```
void InitBluetooth(void);
```

Parametri:
Nema

Povratna vrijednost:
Nema

Podržani moduli: CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>            /* dodjeljivanje EMoRo biblioteke      */

int main(void){                        /* glavni program                      */
   InitIO();                           /* inicijaliziraj ulaze i izlaze       */
   InitBluetooth();                    /* inicijaliziraj bluetooth            */
   while(1){                           /* beskonačna petlja                   */
      printf("Hello World\r\n");       /* pošalji pozdravnu poruku            */
   }
}
```

Funkcija mijenja ime bluetooth modula. Za promjenu imena bluetooth modula potrebno je obaviti inicijalizaciju **InitBluetooth();** funkcijom.

```
int BluetoothChangeName(char* new_name);
```

Parametri:

char* new_name – Pokazivač na prvi član novog imena.

Povratna vrijednost:

int – Promjena imena:

(0) – Bluetooth ime uspješno promjenjeno

(-1) – Greška: Bluetooth modul ne podržava znakove novog imena

Podržani moduli:

CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>                /* dodjeljivanje EMoRo biblioteke   */

int main(void){                            /* glavni program                   */
   InitIO();                               /* inicijaliziraj ulaze i izlaze    */
   InitBluetooth();                        /* inicijaliziraj bluetooth.        */
   BluetoothChangeName("My_New_Name");     /* promjeni ime u „My_New_Name“     */
   while(1){                               /* beskonačna petlja                */
   }
}
```

Funkcija čita ime bluetooth modula. Za čitanje imena bluetooth modula potrebno je obaviti inicijalizaciju **InitBluetooth();** funkcijom.

```
char* BluetoothGetName(void);
```

Parametri:

Nema

Povratna vrijednost:

char* – Pokazivač na ime bluetooth modula

Podržani moduli:

CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>              /* dodjeljivanje EMoRo biblioteke   */

char *module_name;

int main(void){                          /* glavni program                   */
   InitIO();                             /* inicijaliziraj ulaze i izlaze    */
   InitBluetooth();                      /* inicijaliziraj bluetooth         */
   module_name = BluetoothGetName();     /* pročitaj ime bluetooth modula    */
   while(1){                             /* beskonačna petlja                */
   }
}
```

Funkcija mijenja lozinku bluetooth modula. Za promjenu lozinke bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth();** funkcijom.

```
int BluetoothChangePasskey(char* new_passkey);
```

Parametri:

char* new_passkey – Pokazivač na prvi član nove lozinke

Povratna vrijednost:

int – Promjena lozinke:

(0) – Bluetooth lozinka uspješno promjenjena

(-1) – Greška: Bluetooth modul ne podržava znakove nove lozinke

Podržani moduli: CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>              /* dodjeljivanje EMoRo biblioteke   */

int main(void){                          /* glavni program                   */
   InitIO();                             /* inicijaliziraj ulaze i izlaze    */
   InitBluetooth();                      /* inicijaliziraj bluetooth         */
   BluetoothChangePasskey("2323");       /* promjeni lozinku u „2323“        */
   while(1){                             /* beskonačna petlja                */
   }
}
```

Funkcija vraća stanje bluetooth veze (bluetooth spojen, bluetooth nije spojen). Za čitanje bluetooth veze potrebno je obaviti inicijalizaciju **InitBluetooth();** funkcijom.

```
unsigned char BluetoothConnection(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje bluetooth veze:

(1) – Bluetooth spojen

(0) – Bluetooth odspojen

Podržani moduli: CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>              /* dodjeljivanje EMoRo biblioteke   */

int main(void){                          /* glavni program                   */
   InitIO();                             /* inicijaliziraj ulaze i izlaze    */
   InitBluetooth();                      /* inicijaliziraj bluetooth         */

   while(1){                             /* beskonačna petlja                */

      while(BluetoothConnection() == 0); /* čekaj povezivanje bluetooth-a    */
      printf("Hello World\r\n");         /* pošalji pozdravnu poruku         */
      while(BluetoothConnection() == 1); /* čekaj odspajanje bluetooth-a     */
   }
}
```

Funkcija čita tipku bluetooth modula T1. Za čitanje stanja tipke potrebno je inicijalizirati bluetooth **InitBluetooth();** funkcijom.

```
unsigned char BluetoothT1(void);
```

Parametri:

Nema

Povratna vrijednost:

unsigned char – Stanje tipke T1:

(0) – puštena tipka

(1) – pritisnuta tipka

Podržani moduli: CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>        /* dodjeljivanje EMoRo biblioteke         */

int main(void){                    /* glavni program                         */
   InitIO();                       /* inicijaliziraj ulaze i izlaze          */
   InitBluetooth();                /* inicijaliziraj bluetooth komunikaciu   */

   while(BluetoothT1() == 0);      /* čekaj dok tipka T1 ne bude pritisnuta  */
   Relay1On();                     /* uključi relej 1                        */

   while(1){                       /* beskonačna petlja                      */
   }
}
```

Funkcije upravljaju STATUS LED-om bluetooth modula. Prije uključivanja, isključivanja ili promjene stanja STATUS LED-a potrebno je inicijalizirati bluetooth modul **InitBluetooth();** funkcijom.

```
void BluetoothLedOn(void);
void BluetoothLedOff(void);
void BluetoothLedToggle(void);
```

Parametri:

Nema

Povratna vrijednost:

Nema

Podržani moduli: CPU Bluetooth

Primjer:

```
#include <inovatic\emoro.h>         /* dodjeljivanje EMoRo biblioteke         */

int main(void){                     /* glavni program                         */
   InitIO();                        /* inicijaliziraj ulaze i izlaze          */
   InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciu   */

   while(1){                        /* beskonačna petlja                      */
      BluetoothLedOn();             /* uključi bluetooth STATUS LED           */
      _delay_ms(500);               /* čekaj 500 mS                           */

      BluetoothLedOff();            /* isključi bluetooth STATUS LED          */
      _delay_ms(500);               /* čekaj 500 mS                           */

      BluetoothLedToggle();         /* promjeni stanje bluetooth STATUS LED   */
      _delay_ms(500);               /* čekaj 500 mS                           */
   }
}
```

10.7 *Popis portova i sheme pojedinih modula*

Port konektor	Mikrokontroler pin	Funkcija
P1 (PORT1)	PA0	DISTANCE – 1
P2 (PORT2)	PD4/OC1B	PWM-16(1) – ALKAR
P3 (PORT3)	PD5/OC1A	PWM-16(2) / R_DIRECTION
P4 (PORT4)	PD6	L_DIRECTION
P5 (PORT5)	PB3/OC0	PWM-8(2) / R_PWM
P6 (PORT6)	PA1	IR – 1
P7 (PORT7)	PA2	IR – 2
P8 (PORT8)	PA3	RELAY – 1
P9 (PORT9)	PA4	INPUT – 4
P10 (PORT10)	PA5	RELAY – 2
P11 (PORT11)	PA6	INPUT – 3
P12 (PORT12)	PA7	RELAY – 3
P13 (PORT13)	PC7	INPUT – 2
P14 (PORT14)	PC6	RELAY – 4
P15 (PORT15)	PC1	INPUT – 1
P16 (PORT16)	PD7/OC2	PWM-8(1) / L_PWM
Vbat	-	ULAZNI NAPON
GND	-	GND
Vcc	-	NAPON 5V
GND	-	GND
-	PC0	BOARD CHECK
-	PB0	STATUS LED BT
-	PB1	TIPKA BT T1

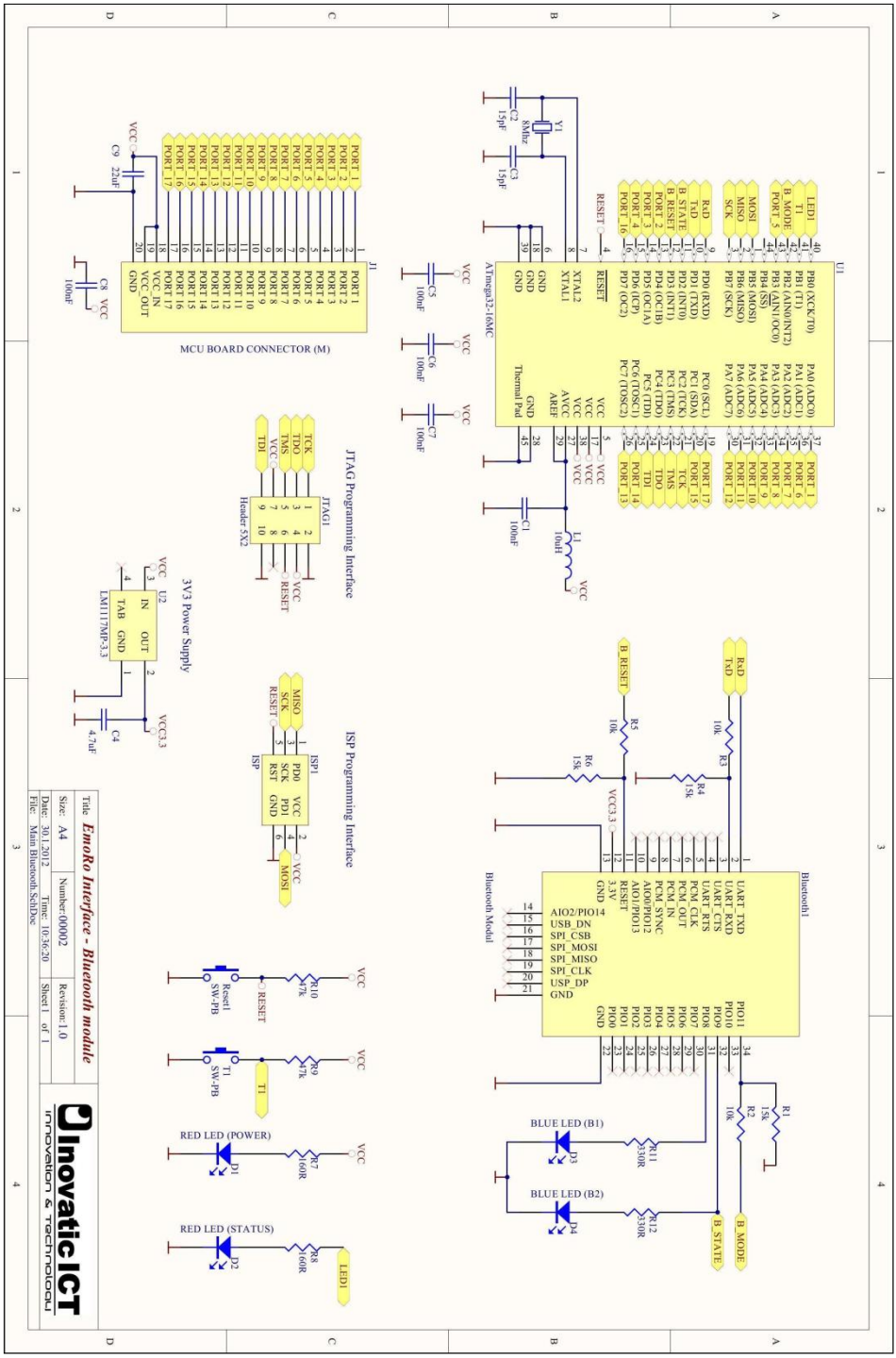
Tablica 5. ATmega32 mikrokontroler pinovi

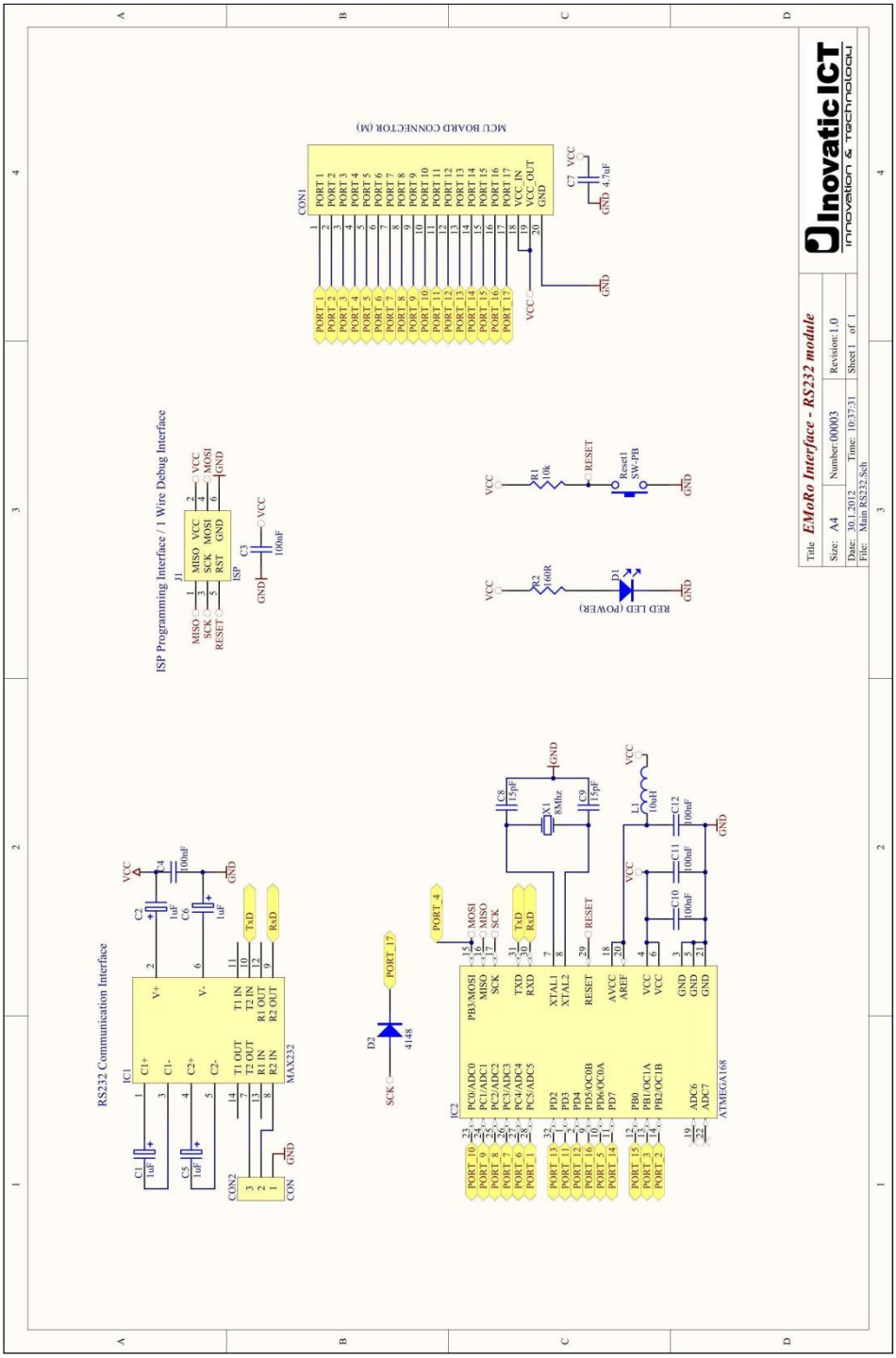
Port konektor	Mikrokontroler pin	Funkcija
P1 (PORT1)	PC5/ADC5	DISTANCE – 1
P2 (PORT2)	PB2/OC1B	PWM-16(1) – ALKAR
P3 (PORT3)	PB1/OC1A	PWM-16(2) / R_DIRECTION
P4 (PORT4)	PB3/MOSI	L_DIRECTION
P5 (PORT5)	PD6/OC0A	PWM-8(2) / R_PWM
P6 (PORT6)	PC4/ADC4	IR – 1
P7 (PORT7)	PC3/ADC3	IR – 2
P8 (PORT8)	PC2	RELAY – 1
P9 (PORT9)	PC1	INPUT – 4
P10 (PORT10)	PC0	RELAY – 2
P11 (PORT11)	PD3	INPUT – 3
P12 (PORT12)	PD4	RELAY – 3
P13 (PORT13)	PD2	INPUT – 2
P14 (PORT14)	PD7	RELAY – 4
P15 (PORT15)	PB0	INPUT – 1
P16 (PORT16)	PD5/OC0B	PWM-8(1) / L_PWM
Vbat	-	ULAZNI NAPON
GND	-	GND
Vcc	-	NAPON 5V
GND	-	GND

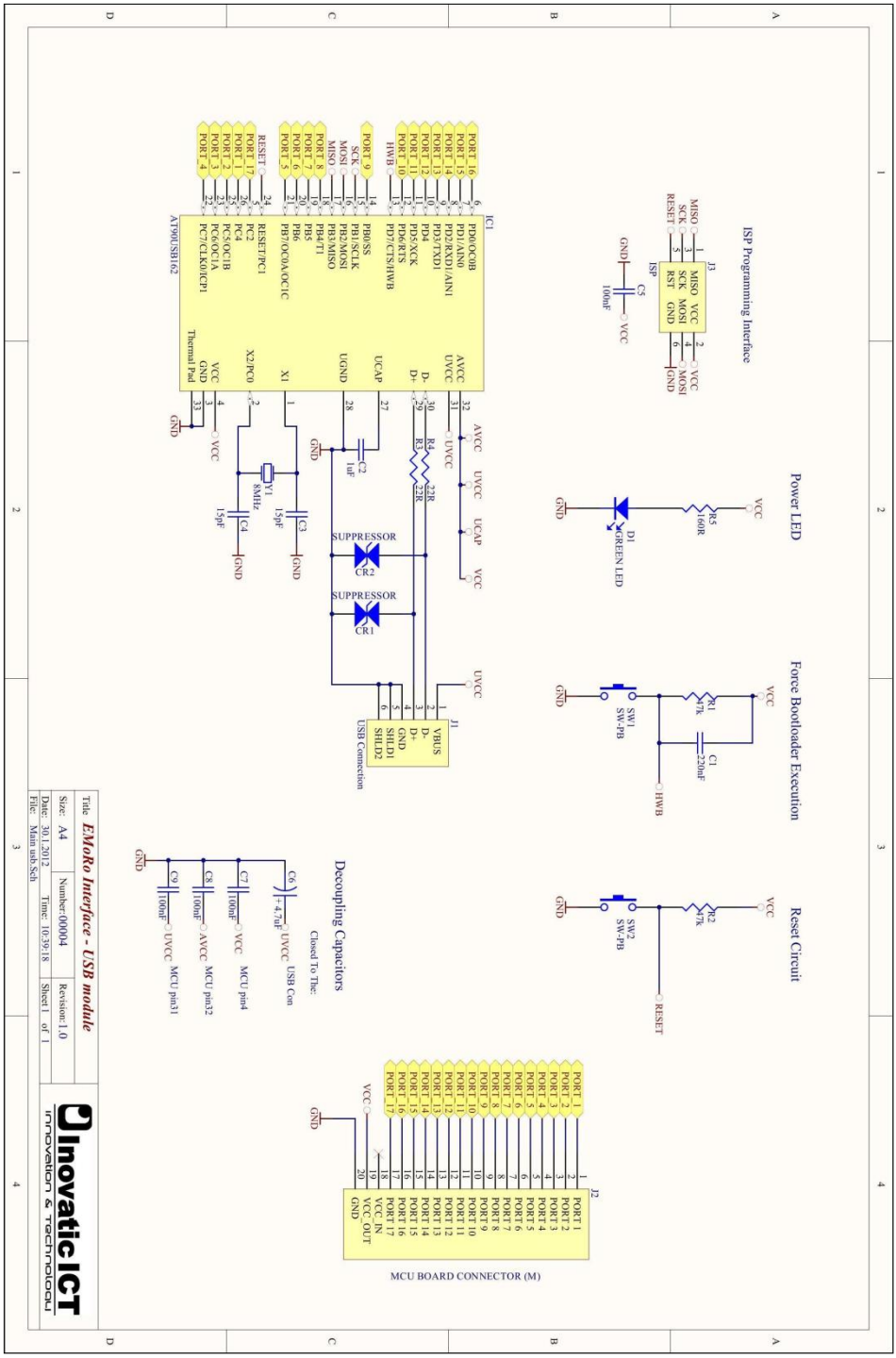
Tablica 6. ATmega168 mikrokontroler pinovi

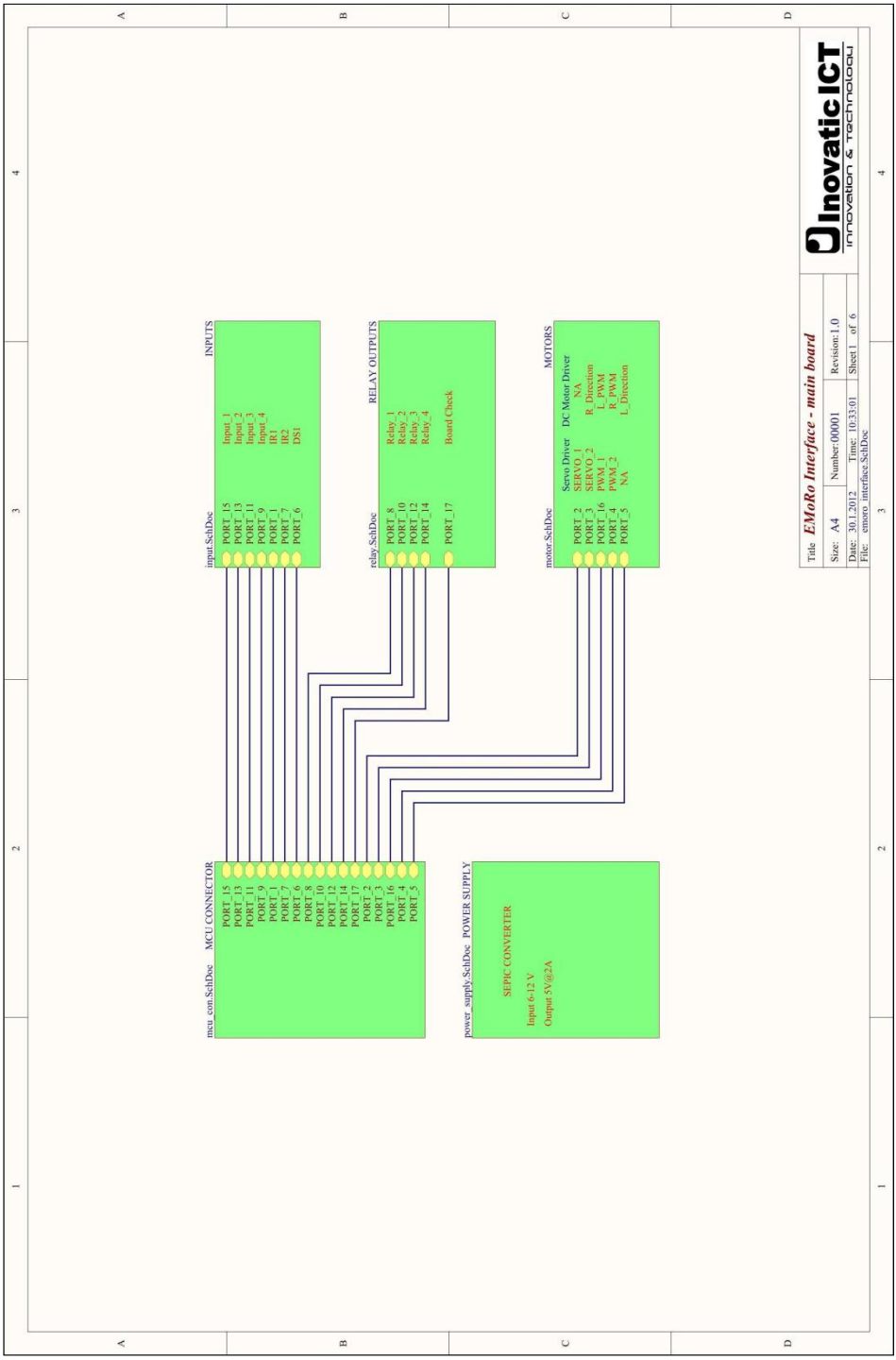
Port konektor	Mikrokontroler pin	Funkcija
P1 (PORT1)	PC4	DISTANCE – 1
P2 (PORT2)	PC5/OC1B	PWM-16(1) – ALKAR
P3 (PORT3)	PC6/OC1A	PWM-16(2) / R_DIRECTION
P4 (PORT4)	PC7	L_DIRECTION
P5 (PORT5)	PB7/OC0A/OC1C	PWM-8(2) / R_PWM
P6 (PORT6)	PB6	IR – 1
P7 (PORT7)	PB5	IR – 2
P8 (PORT8)	PB4	RELAY – 1
P9 (PORT9)	PB0	INPUT – 4
P10 (PORT10)	PD6	RELAY – 2
P11 (PORT11)	PD5	INPUT – 3
P12 (PORT12)	PD4	RELAY – 3
P13 (PORT13)	PD3	INPUT – 2
P14 (PORT14)	PD2	RELAY – 4
P15 (PORT15)	PD1	INPUT – 1
P16 (PORT16)	PD0/OC0B	PWM-8(1) / L_PWM
Vbat	-	ULAZNI NAPON
GND	-	GND
Vcc	-	NAPON 5V
GND	-	GND
-	PC2	BOARD CHECK

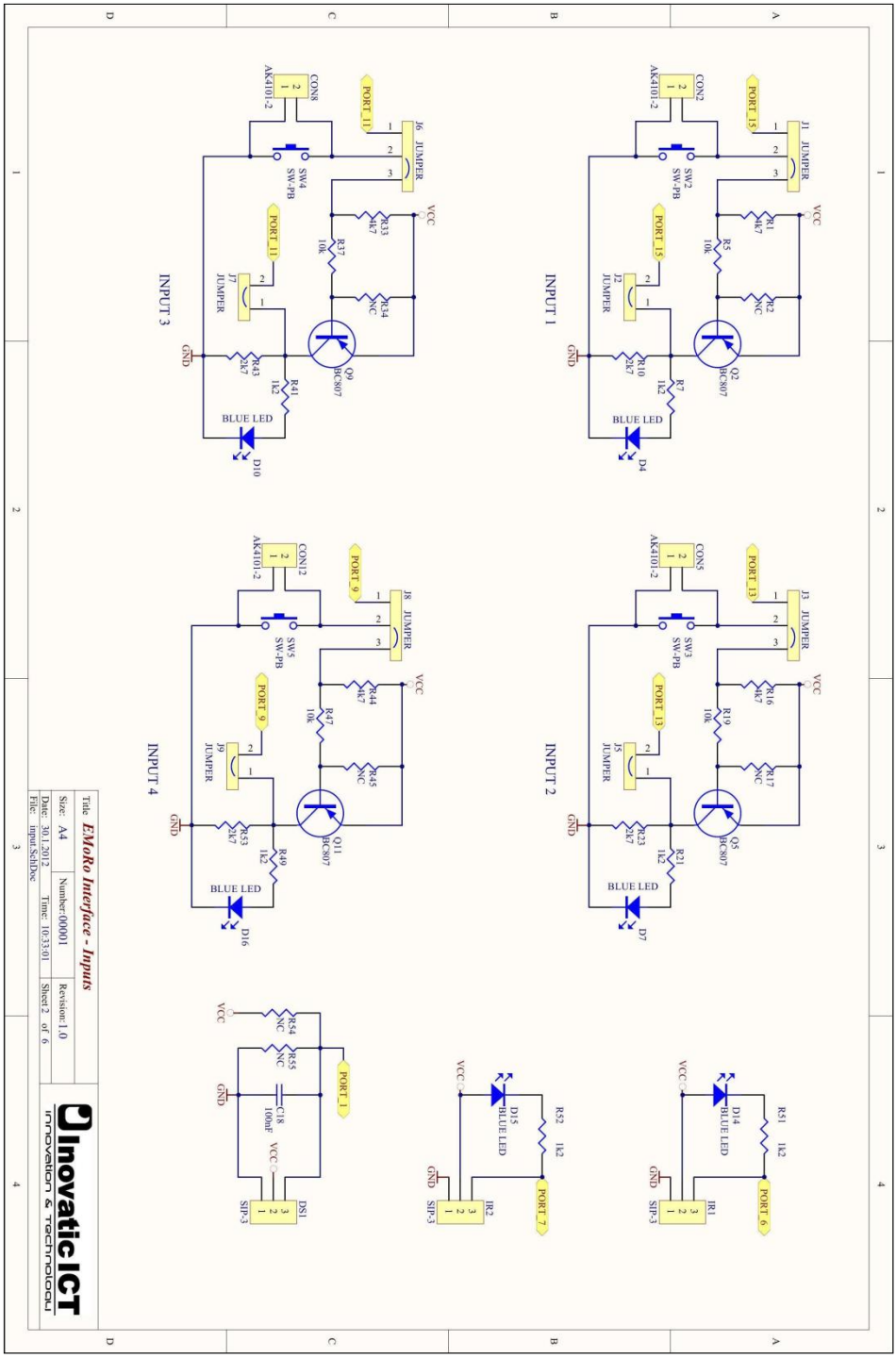
Tablica 7. AT90USB162 mikrokontroler pinovi

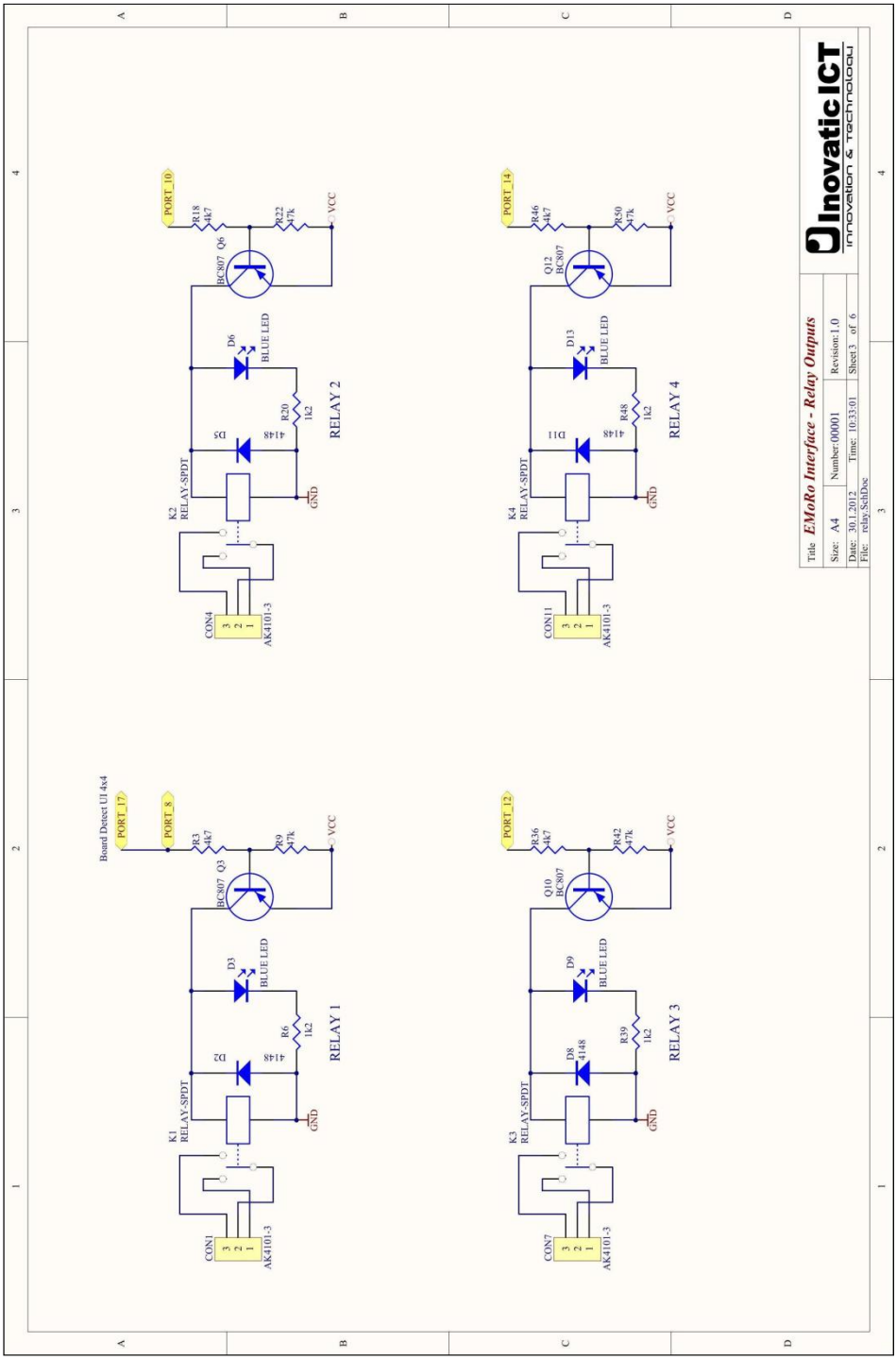

U1
ATmega32-16MC
MCU BOARD CONNECTOR (M)
J1
JTAG Programming Interface
Header 5X2
3V3 Power Supply
LM1117MP-3.3
ISP Programming Interface
Bluetooth Modul
Bluetooth1
BLUE LED (B1)
BLUE LED (B2)
RED LED (POWER)
RED LED (STATUS)
Reset1
SW-PB
T1
Title EmoRo Interface - Bluetooth module
Size: A4
Number: 00002
Revision: 1.0
Date: 30.1.2012
Time: 10:56:20
Sheet 1 of 1
File: Main Bluetooth.SchDoc
Inovatic ICT
Innovation & technology

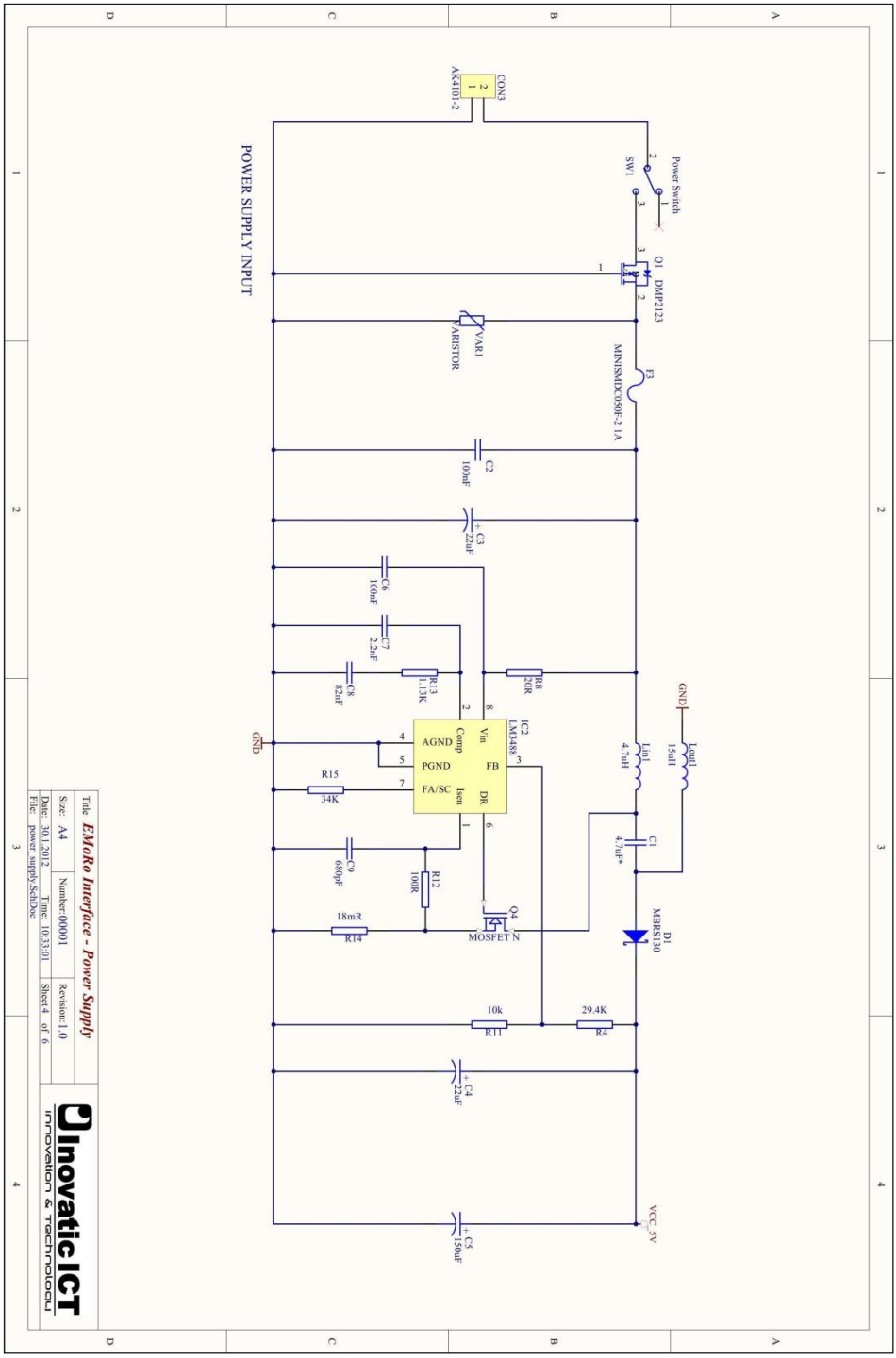

RS232 Communication Interface
ISP Programming Interface / 1 Wire Debug Interface
MCU BOARD CONNECTOR (M)
RED LED (POWER)
Title: EMoRo Interface - RS232 module
Size: A4
Number:00003
Revision:1.0
Date: 30.1.2012
Time: 10:37:31
Sheet 1 of 1
File: Main RS232.Sch
Inovatic ICT
innovation & technology

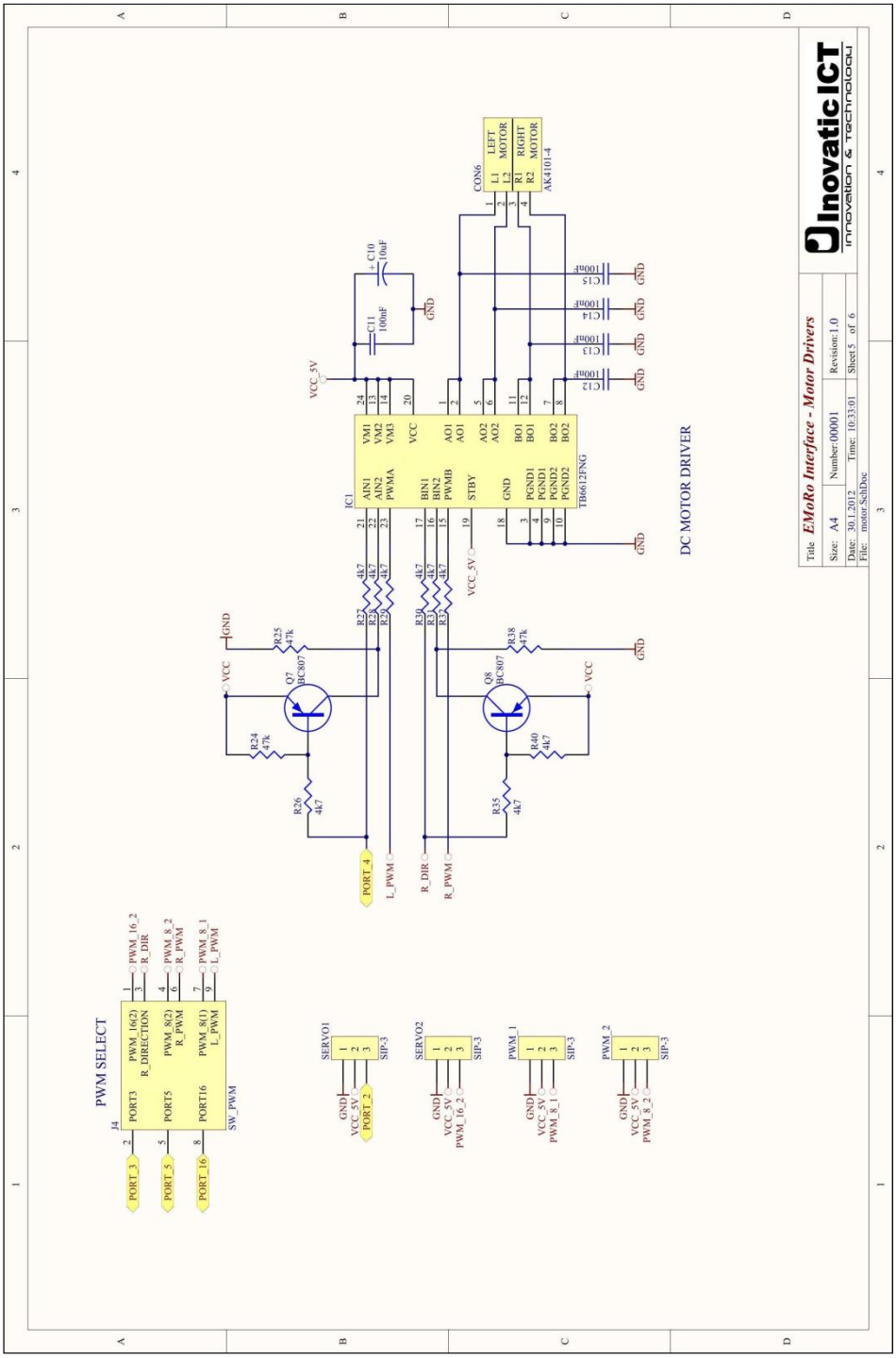

ISP Programming Interface
J3
MISO VCC
SCK MOSI
RST GND
MISO
SCK
RESET
VCC
MOSI
GND
ISP
C5
100nF
GND
VCC
Power LED
R5
160R
D1
GREEN LED
Force Bootloader Execution
R1
47k
C1
220nF
SW1
SW-PB
HWB
Reset Circuit
R2
47k
SW2
SW-PB
RESET
IC1
AT90USB162
PDO/OC0B
PD1/SDA/INT0
PD2/RXD1/AIN1
PD3/TXD1
PD4
PD5/XCK
PD6/RTS
PD7/CTS/HWB
PB0/SS
PB1/SCLK
PB2/MOSI
PB3/MISO
PB4/T1
PB5
PB6
PB7/OC0A/OC1C
RESET/PC1
PC4
PC5/OC1B
PC6/OC1A
PC7/CLKO/ICP1
Thermal Pad
AVCC
UVCC
D-
D+
UCAP
UGND
X1
X2/PC0
VCC
GND
R4
22R
R3
22R
C2
1uF
SUPPRESSOR
CR2
CR1
Y1
8MHz
C3
15pF
C4
15pF
J1
VBUS
D-
D+
GND
SHLD1
SHLD2
USB Connection
Decoupling Capacitors
Closed To The:
C6
4.7uF
USB Con
C7
100nF
MCU pin4
C8
100nF
MCU pin32
C9
100nF
MCU pin31
J2
PORT 1
PORT 2
PORT 3
PORT 4
PORT 5
PORT 6
PORT 7
PORT 8
PORT 9
PORT 10
PORT 11
PORT 12
PORT 13
PORT 14
PORT 15
PORT 16
PORT 17
VCC_IN
VCC_OUT
MCU BOARD CONNECTOR (M)
Title
EMoKo Interface - USB module
Size: A4
Number: 00004
Revision: 1.0
Date: 30.1.2012
Time: 10:39:18
Sheet 1 of 1
File: Main usb.Sch
Inovatic ICT
innovation & technology

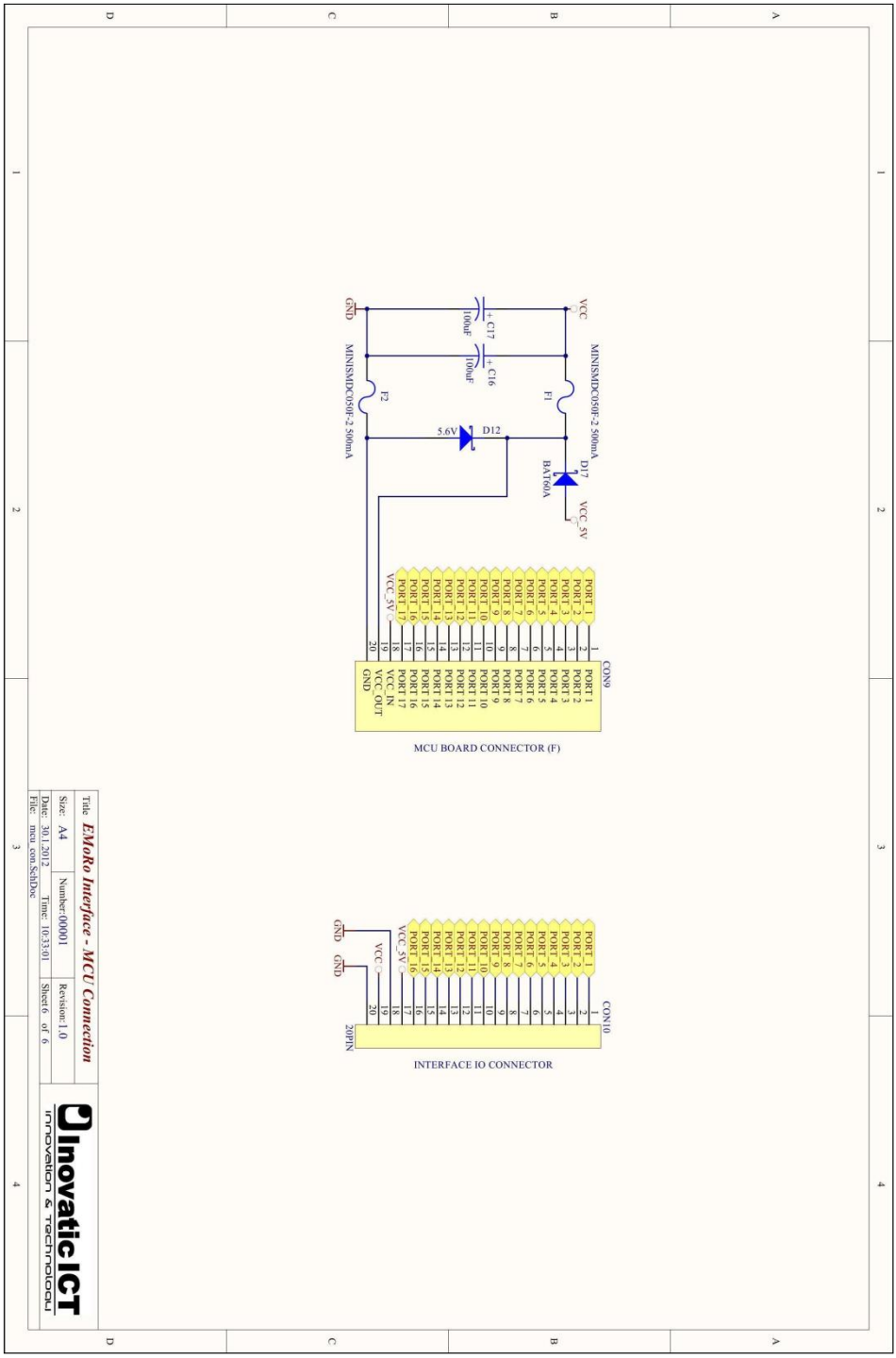

INPUTS
input.SchDoc
PORT_15
PORT_13
PORT_11
PORT_9
PORT_1
PORT_7
PORT_6
PORT_8
PORT_10
PORT_12
PORT_14
PORT_17
PORT_2
PORT_3
PORT_16
PORT_4
PORT_5
Input_1
Input_2
Input_3
Input_4
IR1
IR2
DS1
RELAY OUTPUTS
relay.SchDoc
PORT_8
PORT_10
PORT_12
PORT_14
PORT_17
Relay_1
Relay_2
Relay_3
Relay_4
Board Check
MOTORS
motor.SchDoc
PORT_2
PORT_3
PORT_16
PORT_4
PORT_5
Servo Driver
SERVO_1
SERVO_2
PWM_1
PWM_2
NA
DC Motor Driver
NA
R_Direction
L_PWM
R_PWM
L_Direction
MCU CONNECTOR
mcu_con.SchDoc
PORT_15
PORT_13
PORT_11
PORT_9
PORT_1
PORT_7
PORT_6
PORT_8
PORT_10
PORT_12
PORT_14
PORT_17
PORT_2
PORT_3
PORT_16
PORT_4
PORT_5
POWER SUPPLY
power_supply.SchDoc
SEPIC CONVERTER
Input 6-12 V
Output 5V@2A
Title EMoRo Interface - main board
Size: A4
Number:00001
Revision:1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 1 of 6
File: emoro_interface.SchDoc
Innovatic ICT
innovation & technology


INPUT 1
INPUT 2
INPUT 3
INPUT 4
BLUE LED
Title EMoKo Interface - Inputs
Size: A4
Number: 00001
Revision: 1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 2 of 6
File: input.SchDoc
Inovatic ICT
innovation & technology


Board Detect UI 4x4
PORT_17
PORT_8
R3 4k7
R9 47k
VCC
BC807 Q3
D3 BLUE LED
R6 1k2
D2 4148
K1 RELAY-SPDT
CON1 AK4101-3
GND
RELAY 1
PORT_10
R18 4k7
R22 47k
VCC
BC807 Q6
D6 BLUE LED
R20 1k2
D5 4148
K2 RELAY-SPDT
CON4 AK4101-3
GND
RELAY 2
PORT_12
R36 4k7
R42 47k
VCC
Q10 BC807
D9 BLUE LED
R39 1k2
D8 4148
K3 RELAY-SPDT
CON7 AK4101-3
GND
RELAY 3
PORT_14
R46 4k7
R50 47k
VCC
Q12 BC807
D13 BLUE LED
R48 1k2
D11 4148
K4 RELAY-SPDT
CON11 AK4101-3
GND
RELAY 4
Title EMoRo Interface - Relay Outputs
Size: A4
Number:00001
Revision:1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 3 of 6
File: relay.SchDoc
3
Innovatic ICT
innovation & technology


POWER SUPPLY INPUT
CON3
AK4101-2
Power Switch
SW1
Q1
DMP2123
VAR1
VARISTOR
F3
MINISMDC050F-2 1A
C2
100nF
C3
22uF
C6
100nF
C7
2.2nF
C8
82nF
R13
1.13K
R8
20R
IC2
LM3488
AGND
PGND
FA/SC
Comp
Isen
Vin
FB
DR
GND
R15
34K
C9
680pF
R12
100R
Q4
MOSFET N
18mR
R14
10k
R11
29.4K
R4
C4
22uF
C5
150uF
VCC_5V
Lin1
4.7uH
C1
4.7uF*
D1
MBRS130
Lout1
15uH
Title
EMoKo Interface - Power Supply
Size: A4
Number: 00001
Revision: 1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 4 of 6
File: power supply.SchDoc
Inovatic ICT
innovation & technology


PWM SELECT
J4
PORT3
PORT5
PORT16
PWM_16(2)
R_DIRECTION
PWM_8(2)
R_PWM
PWM_8(1)
L_PWM
SW_PWM
PORT_3
PORT_5
PORT_16
PWM_16_2
R_DIR
PWM_8_2
R_PWM
PWM_8_1
L_PWM
SERVO1
SERVO2
PWM_1
PWM_2
SIP-3
GND
VCC_5V
PORT_2
PWM_16_2
PWM_8_1
PWM_8_2
IC1
TB6612FNG
AIN1
AIN2
PWMA
BIN1
BIN2
PWMB
STBY
VM1
VM2
VM3
VCC
AO1
AO2
BO1
BO2
GND
PGND1
PGND2
C10
10uF
C11
100nF
C12
C13
C14
C15
CON6
AK4101-4
LEFT MOTOR
RIGHT MOTOR
L1
L2
R1
R2
Q7
BC807
Q8
BC807
R24
47k
R25
47k
R26
4k7
R27
4k7
R28
4k7
R29
4k7
R30
4k7
R31
4k7
R32
4k7
R33
4k7
R34
4k7
R35
4k7
R38
47k
R40
4k7
VCC
PORT_4
L_PWM
R_DIR
R_PWM
DC MOTOR DRIVER
Title: EMoRo Interface - Motor Drivers
Size: A4
Number: 00001
Revision: 1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 5 of 6
File: motor.SchDoc
Innovatic ICT
innovation & technology


MINISMDC050F-2 500mA
F1
VCC
D17
BAT60A
VCC_5V
C17
100uF
C16
100uF
5.6V
D12
F2
MINISMDC050F-2 500mA
GND
CON9
PORT 1
PORT 2
PORT 3
PORT 4
PORT 5
PORT 6
PORT 7
PORT 8
PORT 9
PORT 10
PORT 11
PORT 12
PORT 13
PORT 14
PORT 15
PORT 16
PORT 17
VCC_5V
VCC_IN
VCC_OUT
GND
MCU BOARD CONNECTOR (F)
CON10
20PIN
VCC
INTERFACE IO CONNECTOR
Title
EMoKo Interface - MCU Connection
Size: A4
Number:00001
Revision:1.0
Date: 30.1.2012
Time: 10:33:01
Sheet 6 of 6
File: mcu_con.SchDoc
Inovatic ICT
innovation & technology

10.8 *Uklanjanje pogrešaka (Troubleshooting)*

Ponovno stavljanje / nadogradnja programa na mikrokontroler

Programiranje USB sučelja
- Pronalazak .hex datoteke na „(DVD) → Program Example → AVR Studio → CPU162-USB → EMoRo_Interpreter → cdc.hex

Programiranje CPU sučelja
- Pronalazak .hex datoteke na „(DVD) → Program Example → AVR Studio → CPU168-RS232 → EMoRo_Interpreter → default → EMoRo_Interpreter.hex

Programiranje Bluetooth sučelja
- Pronalazak .hex datoteke na „(DVD) → Program Example → AVR Studio → CPU Bluetooth → EMoRo_Interpreter → default → EMoRo_Interpreter.hex

Nakon pronalaska željene .hex datoteke potrebno je otići na poglavlje 4.6 ako koristite USB sučelje, tj. na poglavlje 4.7 ako koristite CPU ili Bluetooth sučelje i slijediti upute za programiranje sučelja.

Spajanje Bluetooth sučelja

Računalo ne prepoznaje USB Bluetooth dongle:
- Odspojiti i ponovno spojiti dongle na računalo
- Promijeniti USB port na koji spajamo dongle (neispravan port)

Objašnjenje LED signalizacije
- LED B1 i B2 ne svijetle, a POWER svijetli
  - Potrebno ponovno staviti ispravan program sa protokolom komunikacije na mikrokontroler
- LED B1 brzo blinka, a B2 ne svijetli
  - Sučelje je spremno za povezivanje sa računalom i vidljivo je u pronađenim bluetooth uređajima, ali nije povezano s računalom
- LED B1 sporo blinka, a B2 ne svijetli
  - Sučelje povezano sa računalom, „čeka“ na pokretanje programa u FreeBasic-u ili otvaranje COM porta preko HyperTerminal-a
- LED B1 blinka, a B2 konstantno svijetli
  - Sučelje je spojeno sa računalom i otvoren je COM port i ispravno je uspostavljena komunikacija (pokrenut programski primjer)

Povratak na početne postavke (ime i passkey)
- Povratak na poglavlje 3.3, te ponovno spajanje sučelja

Puca veza sa računalom
- Potrebno punjenje ili zamjena baterija
- Prevlika udaljenost između bluetooth dongla i sučelja
- Na sučelje spojen prevelik potrošač koji baterije ne mogu dovoljno napajati

Spajanje USB sučelja

Računalo ne prepoznaje spojeno sučelje
- Zamjena USB porta na računalu (neispravan port)
- Potrebno ponovno staviti ispravan program sa protokolom komunikacije na mikrokontroler

Nepravilna instalacija upravljačkih programa
- Povratak na poglavlje 3.1, te ponovna instalacija potrebnih upravljačkih programa i spajanje sučelja sa računalom

Servo motori i senzori ne rade ispravno
- Nije spojeno dodatno vanjsko napajanje (baterije). Preko USB porta na računalu možemo napajati samo ulaze i izlaze (releje), s tim da bilo koji dodatni potrošač (servo motor i sl.) zahtjeva više struje nego što možemo dobiti iz porta, stoga je potrebno spojiti dodatno napajanje.
- Potrebno punjenje ili zamjena baterija

Spajanje na hyper terminal

Nemogućnost otvaranja zadanog COM porta
- Odabran krivi COM port za spajanje, ponoviti spajanje
- Potrebno ponovno staviti ispravan program sa protokolom komunikacije na mikrokontroler (Bluetooth i CPU sučelje)

Provjera ispravnosti ulaza i izlaza

LED signalizacija i program
- Pokrenuti Primjer 2 (FlowDraw, FreeBasic, BASCOM AVR ili AVR Studio)
    - Pritiskom na tipku Tipka_1 svijetli LED za Ulaz-1 i Izlaz_1 i čuje se preklapanje releja (klik)
    - Pritiskom na tipku Tipka_2 svijetli LED za Ulaz-2 i Izlaz_2 i čuje se preklapanje releja (klik)
    - Pritiskom na tipku Tipka_3 svijetli LED za Ulaz-3 i Izlaz_3 i čuje se preklapanje releja (klik)
    - Pritiskom na tipku Tipka_4 svijetli LED za Ulaz-4 i Izlaz_4 i čuje se preklapanje releja (klik)
- Provjeriti konfiguraciju kratkospojnika ulaza na donjoj strani sučelja (J1-J8), zadane postavke označene su na shematskom prikazu s „Normal“

Provjera napajanja

Sučelje ne radi – PWR led na modulu ne svijetli
- Crvena i crna žica za napajanje obratno su okrenute, potrebno provjeriti. Crvena žica ide na 6-12V, a crna žica na GND.
- Krivo postavljene baterije u kućište
- Neispravne ili potpuno prazne baterije

AVR Studio, BASCOM AVR i SinaProg ne prepoznaju programator

Provjera postavki kratkospojnika (jumperi)
- HID i SCK
    - Povratak na poglavlje 9.1.2

Instalacija upravljačkih programa
- Povratak na poglavlje 9.1.1, te ponovna instalacija potrebnih upravljačkih programa i spajanje programatora s računalom

Pronađen programator, ali ne prepoznaje mikrokontroler
- Odabran krivi mikrokontroler
- Nije spojeno napajanje na sučelje
- Krivo postavljen ISP konektor
    - Povratak na poglavlje 9.1

Pokretanje robota

Robot ide unatrag – pogon servo motori
- Zamijeniti SERVO_1 i SERVO_2 konektore

Robot ide unatrag – pogon DC motori
- Provjeriti spajanje plavih i zelenih žica, povratak na poglavlje 2.5.3

Robot se vrti u krug – pogon DC motori
- Utvrditi koji se kotač okreće unatrag i zamijeniti žice tog motora (npr. ako se kotač kojeg pokreće motor za zelenim žicama okreće unatrag, onda je potrebno obratno spojiti žice na konektoru Motor-R)

Spajanje konektora za servo motore i senzore

Na svim konektorima na sučelju označen je prvi pin trokutićem. Prvi pin na sučelju uvijek je GND (crni ili smeđi), drugi je VCC – 5V (crveni), a treći pin je signalni (bijeli ili narančasti), kao što je prikazano na slici ispod.

Slika 92. Konektor za servo/senzore

Indeks

11

11. INDEKS

Popis slika:

Slika 1. Plastični dio za pritezanje matica 7
Slika 2. Kotač za pogonske servo motore 8
Slika 3. Sastavljeni kotač za pogonske servo motore 8
Slika 4. Kotač za elektromotorni prijenos 9
Slika 5. Sastavljeni kotač za elektromotorni prijenos 9
Slika 6. Svesmjerni kotač - Omniwheel 10
Slika 7. Svesmjerni kotač - Omniwheel 10
Slika 8. Sastavljeni kotač za elektromotorni prijenos 11
Slika 9. Svesmjerna kugla - BallCaster 12
Slika 10. Postavljanje kugle i metalnih osovina u donji plastični dio 12
Slika 11. Postavljanje gornje plastike 12
Slika 12. Sastavljena svesmjerna kugla na šasiji 13
Slika 13. Pogonski servo motor 14
Slika 14. Postavljanje gumica i metalnih čahurica na servo motor 14
Slika 15. Postavljanje pogonskog servo motora na šasiju 15
Slika 16. Sastavljeni pogonski servo motori i kotači na šasiji 15
Slika 17. Prijenos za elektromotore – Twin Motor Gearbox (brzina C) 16
Slika 18. Postavljanje zupčanika na elektromotor 16
Slika 19. Podešavanje visine matice na osovini pogona 17
Slika 20. Sastavljanje desnog dijela prijenosa 01 17
Slika 21. Sastavljanje desnog dijela prijenosa 02 18
Slika 22. Shematski prikaz spajanja desnog dijela prijenosa 18
Slika 23. Sastavljanje lijevog dijela prijenosa 19
Slika 24. Sastavljen prijenos - gearbox 19
Slika 25. Postavljanje elektromotora na prijenos 19
Slika 26. Postavljanje elektromotornog pogona na šasiju 20
Slika 27. Sastavljen elektromotorni pogon sa prijenosom i kotačima na šasiji 20
Slika 28. Prijenos za elektromotore – Twin Motor Gearbox (brzina A) 21
Slika 29. Postavljanje zupčanika na elektromotor 21
Slika 30. Podešavanje visine matice na osovini pogona 22
Slika 31. Podešavanje visine zupčanika na osovini pogona 22
Slika 32. Sastavljanje desnog dijela prijenosa 01 23
Slika 33. Sastavljanje desnog dijela prijenosa 02 23
Slika 34. Shematski prikaz spajanja desnog dijela prijenosa 24
Slika 35. Sastavljanje lijevog dijela prijenosa 24

Slika 36. Sastavljen prijenos - gearbox 25
Slika 37. Postavljanje elektromotora na prijenos 25
Slika 38. Postavljanje elektromotornog pogona na šasiju 26
Slika 39. Sastavljen elektromotorni pogon sa prijenosom i kotačima na šasiji 26
Slika 40. Postavljanje sučelja na gornji dio šasije 27
Slika 41. Spajanje gornjeg i donjeg dijela šasije plastičnim nitnama 27
Slika 42. Postavljanje senzora udaljenosti na metalni dio 28
Slika 43. Spajanje servo roga sa senzorom udaljenosti 28
Slika 44. Postavljanje servo motora na metalni nosač za radar 28
Slika 45. Kompletiranje radar seta 29
Slika 46. Postavljanje radar seta na šasiju 29
Slika 47. Spajanje senzora udaljenosti na sučelje 30
Slika 48. Postavljanje gumica i servo roga za alkar set 31
Slika 49. Sastavljanje donjeg servo motora za alkar set 31
Slika 50. Postavljanje metalnog nosača za drugi servo motor 32
Slika 51. Postavljanje drugog servo motora 32
Slika 52. Sastavljanje koplja za alkara 33
Slika 53. Kompletiranje alkar seta 33
Slika 54. Postavljanje alkar seta na šasiju 34
Slika 55. Spajanje napajanja na sučenje 35
Slika 56. Spajanje adaptera za servo motore 35
Slika 57. Spajanje servo motora na sučelje 36
Slika 58. Spajanje elektromotora na sučelje 36
Slika 59. Postavljanje taktilnih senzora na šasiju 37
Slika 60. Spajanje taktilnih senzora na sučelje 37
Slika 61. Postavljanje foto senzora na šasiju 38
Slika 62. Spajanje foto senzora na sučelje 38
Slika 63. Postavljanje IR senzora na šasiju 39
Slika 64. Spajanje IR senzora na sučelje 39
Slika 65. USB sučelje – modularni prikaz 43
Slika 66. Instalacija upravljačkih programa – USB sučelje 45
Slika 67. Uspješna instalacija USB sučelja 46
Slika 68. CPU sučelje – modularni prikaz 47
Slika 69. Konektor za RS232 - pinovi 49
Slika 70. Bluetooth sučelje – modularni prikaz 50
Slika 71. Pronađen USB bluetooth adapter 53
Slika 72. Prikaz ispravno instaliranih upravljačkih programa 76
Slika 73. Inovatic STK500 ISP programator 124
Slika 74. ISP konektor 124
Slika 75. Instalacija upravljačkih programa – Inovatic STK500 125

Slika 76. Provjera instalacije i COM Port-a 126
Slika 77. Uspješna instalacija programatora..... 126
Slika 78. HID i SCK način rada 127
Slika 79. PWR – USB 5V izlaz..... 128
Slika 80. Konektor za servo/senzore 221
Slika 81. Izgled legende preklopnog kontakta 131
Slika 82. Spajanje izlaza i žaruljice, izlaz sučelja nije aktivan 131
Slika 83. Izlaz sučelja je aktivan, žaruljica se upalila 132
Slika 84. Spajanje izlaza i žaruljice, obrnuto upravljanje 132
Slika 85. Izlaz USB sučelja prilikom aktivnog izlaza i obrnutog upravljanja ... 132

Popis tablica:

Tablica 1. Usporedba EMoRo sučelja 42
Tablica 2. USB sučelje – mogućnosti 44
Tablica 3. CPU sučelje - mogućnosti 49
Tablica 4. Bluetooth sučelje – mogućnosti..... 52
Tablica 5. ATmega32 mikrokontroler pinovi 208
Tablica 6. ATmega168 mikrokontroler pinovi 208
Tablica 7. AT90USB162 mikrokontroler pinovi..... 209

ZABILJEŠKE