

TEHNIČKO VELEUČILIŠTE U ZAGREBU
STRUČNI PRIJEDIPLOMSKI STUDIJ MEHATRONIKA/STROJARSTVO

Mario Dergez

JMBAG: 0246107888

Seminarski rad:
(pojednostavljena) Interpretacija procesnih
računala (PLC-a);
Projekt: Modularna upravljačka jedinica

Zagreb, veljača 2025

Sažetak

Ovime seminarskim radom ostvaruje se koncept koje će služiti kao podloga za završni rad. U ovome konceptu koriste se dva pojednostavljena modula; Modul 000-00 (gdje prve tri znamenke predstavljaju ulogu modula, u ovome slučaju uloga modula je obrada podataka te upravljanje ostalima modula (Modul 000-xx ponaša se kao „Master“ uređaj), te sljedeće dvije znamenke predstavljaju inačicu navedenoga tipa modula, u ovome slučaju „00“ predstavlja konceptnu verziju, te u stvarnosti ova verzija bi bila gotovo beskorisna (osim u edukacijske svrhe), te Modul 004-00 (uloga ovoga modula bi bila upravljanje aktuatora namijenjenog nekoj određenoj svrsi, a sama svrha modula ove inačice (00) je edukacijski koncept za upravljanje istosmjernog elektromotora bez četkica). Između ta dva modula se izvršava komunikacija pomoću „CAN bus“-a, gdje Modul 000-00 šalje, prima i zahtjeva podatke i uglavnom on pokreće komunikaciju, a Modul 004-00 samo prima i šalje podatke, te u ovome konceptu ne pokreće komunikaciju. Sva djelovanja (digitalni izlazi) vrše se putem Modula 004-00 (uključivanje/isključivanje LED, obični digitalni i PWM izlaz), kao što se i sve vanjske promjene prate istim modulom (pritisnuto tipkalo, struja koja djeluje na aktuator, praćenje promjena polariteta BEMF (protu elektromotorne sile))

Ključne riječi: koncept, podloga, Modul 004-00, Modul 000-00

Naslov na engleskom jeziku

(Simplified) Interpretation of Programmable Logic Controller (PLC-a);

Abstract

Prijevod sažetka na engleski.

Keywords: concept, template, Modul 004-00, Modul 000-00

Sadržaj

Popis oznaka i kratica	iv
Popis slika	iv
1. Uvod	i
2. Pojednostavljeni prikaz komponenata za Modul 000-00 i Modul 004-00	1
2.1. Modul 000-00	1
2.2. Modul 004-00	2
3. Postavljanje mikroupravljača za Modul 000-00	4
4. Postavljanje mikroupravljača za Modul 004-00	5
5. Opis rada Modula – KTM	8
5.1. Opis rada Modula 000-00 - KTM	8
5.1.1. Intervalno mijenjanje vrijednosti PWM pulsa, te slanje istoga	8
5.1.2. Intervalno slanje zahtjeva za povratne informacije/primanje informacija	9
5.1.3. Intervalno obrađivanje primljenih podataka	9
5.1.4. Intervalno primanje podataka	10
5.2. Opis rada Modula 004-00 - KTM	10
5.2.1. Intervalno slanje povratnih informacija	10
5.2.2. Intervalno obrađivanje primljenih podataka	10
5.2.3. Intervalno primanje podataka	11
6. Opis rada Modula 004-00 - URS	12
6.1. ADC – čitanje iznosa struje	12
6.2. Obrada PWM signala	12
6.3. Pogon aktuatora	12
6.3.1. Pokretanje (radno stanje)	12
6.3.2. Radno stanje	13
6.3.3. Zaustavljanje (koćenje)	13

6.3.4.	Zastoj (stanje mirovanja)	13
6.4.	Prikaz pomoću LED	13
7.	Funkcije	14
7.1.	(void) funkcije	14
7.1.1.	Normalan smjer radnje	15
7.1.2.	Obrnut smjer radnje	15
7.1.3.	Uključivanje kombinacija izlaza	15
7.1.4.	Pokretanje mjeritelja vremena	15
7.1.5.	Provjera polariteta.....	16
7.1.6.	Pronalazak promjene polariteta	16
7.2.	(callback) funkcije	16
7.2.1.	Pritisnuto tipkalo	16
7.2.2.	Promjena polariteta.....	16
8.	Zaključak	5

Popis oznaka i kratica

URS – Ugradbeni računalni sustavi

KTM – Komunikacijske tehnike u mehatronici

BEMF – eng. *Back Electromagnetic Force*

GND – uzemljenje (eng. *ground*)

LED – eng. *Light Emitting Diode*

CAN – eng. *Controller Area Network*

MOSFET – eng. *Metal Oxide Field Effect Transistor*

ADC – eng. *Analog to Digital Conversion*

BLDC – eng. *Brushless Direct Current*

PWM – Pulsno širinska modulacija (eng. *Pulse Width Modulation*)

Popis slika

Slika 1 - pojednostavljena shema Modula 000-00 -----1

Slika 2 - pojednostavljena shema Modula 004-00 -----2

Slika 3 - predviđeni prikaz promjene polariteta u vremenu -----14

1. Uvod

Cilj ovoga seminarskog uratka je izradba podloge (koncepta) za završni rad, te primjenjivanje znanja iz većine ishoda kolegija Ugradbeni računalni sustavi te kolegija Komunikacijske tehnike u mehatronici.

Cilj projekta za završni rad:

1. izraditi jedinice koje mogu zasebno raditi, neovisno o drugim jedinicama, ali za ispravan rad bile bi potrebne druge jedinice
2. pouzdanost
3. lako nadogradnji sustav (ukoliko jedinicu treba zamijeniti sa jedinicom iste namjene, ali sa možda učinkovitijim ili kvalitetnijim komponentama)
4. lako popravljiv sustav (laka i brza zamjena neispravne jedinice)
5. kompatibilnost (često su brendirani/markirani PLC-ovi ograničeni na dodatke određenih proizvođača - to ne želim)
6. prilagodljivost (mogućnost da se sa jedinicama sastave razni sustavi)

Područja kolegija koje ovaj seminarski rad/projekt obuhvaća:

URS:

1. digitalni ulazi (npr. tipkalo)
2. digitalni izlazi (npr. upravljanje tranzistorom (vodi ili ne vodi))
3. analogni ulazi (npr. mjerenje struje)
4. vremenski i događajno uvjetovani prekidni koncepti (npr. praćenje vremena između promjena BEMF-a, te prekidni koncepti pri navedenome)

KTM:

1. komunikacija između više mikroupravljača (CAN asinkrona digitalna serijska komunikacija između Modula 000-00 i Modula 004-00)

2. Pojednostavljeni prikaz komponenata za Modul 000-00 i Modul 004-00

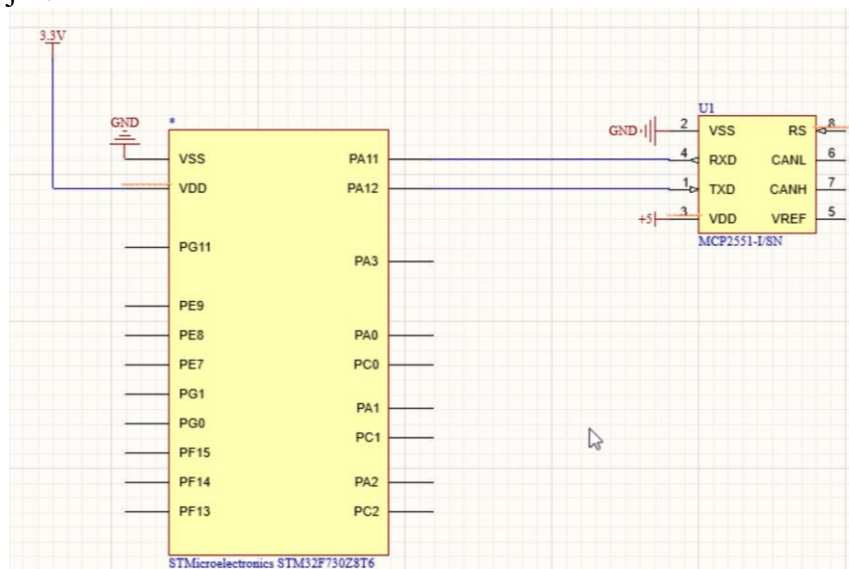
Oba modula bi sadržala potrebne pretvarače napona (nisu prikazani na pojednostavljenoj shemi), kao što i sadrže isti mikroupravljač; STM32F730Z8T6.

U ovome projektu koristimo mikroupravljač STM32F730Z8T6 umjesto mikroupravljača korištenog na laboratorijskim vježbama; STM32F446RET6, iz jednostavnog razloga, a to je da sam mikroupravljač nudi više mogućnosti, a nalazi se u sličnome cjenovnome pragu; poput drastično većeg broj pribadača (*eng. pins*), od čega je meni najbitniji pribadač Vref+ koji nam omogućuje dovoz vanjskog referentnog napona za precizniju analogno-digitalnu pretvorbu, što ga čini boljom podlogom za završni rad.

Za „CAN bus“ komunikaciju, oba modula koriste MCP2551 primopredajnik. Na primopredajnik se dovozi napon od +5V, spaja se na 0V (GND), te se pribadači „CANH“ i „CANL“ spajaju na dvije žice, istih oznaka, koje služe za komunikaciju, međusobno su isprepletene te pri krajevima su spojene raskidnim/završnim otpornicima od 120 Ohma. (na shemama će biti prikazan samo spoj mikroupravljača i komponenti koje se spajaju na ulaze i izlaze samoga mikroupravljača).

2.1. Modul 000-00

Pojednostavljena shema spojeva mikroupravljača sa ulazima i izlazima na sam mikroupravljač.



Slika 1 - pojednostavljena shema Modula 000-00

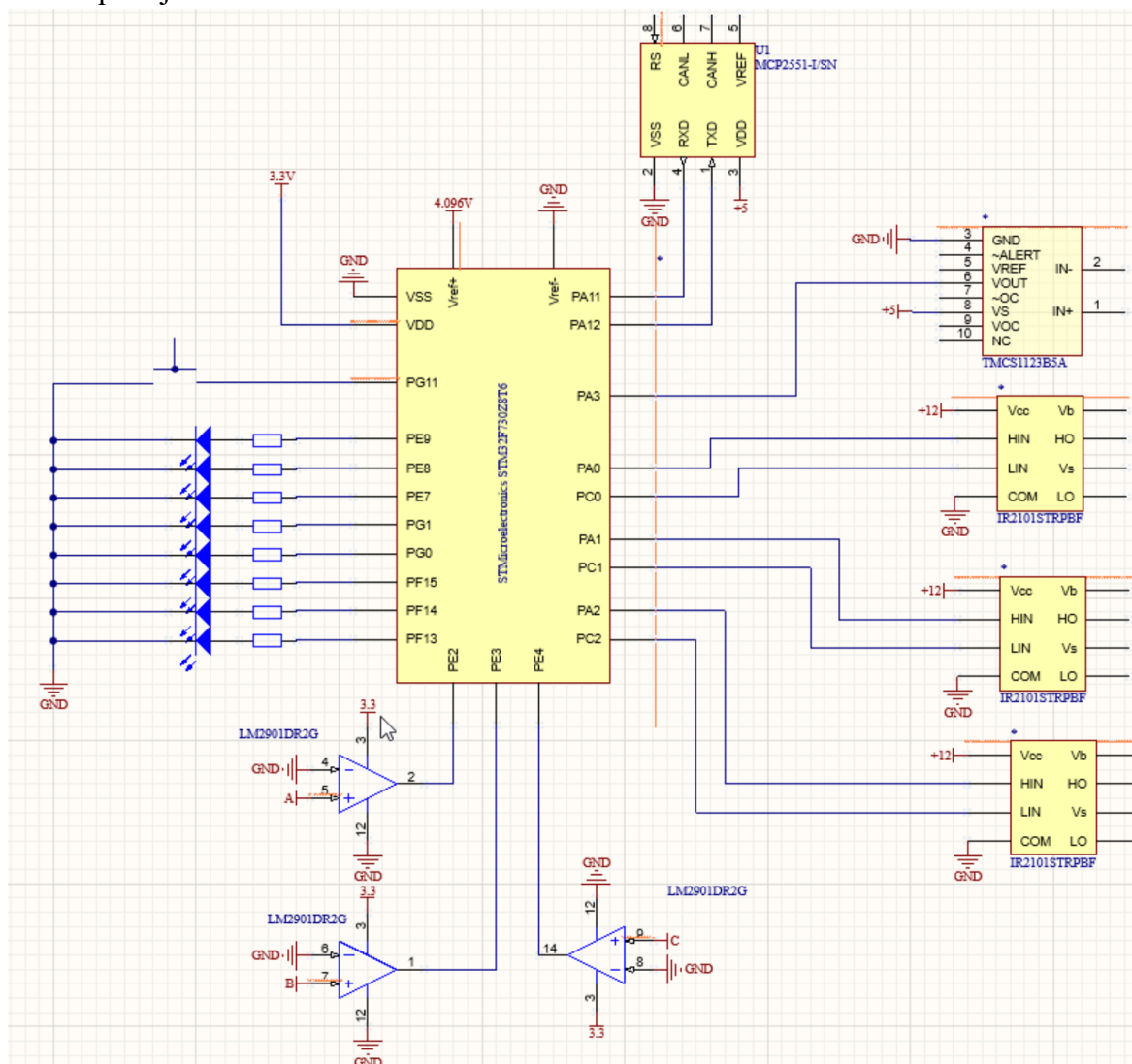
Kratki opis pojednostavljene sheme Modula 000-00 (Slika 1):

Na mikroupravljač se dovodi napon od +3.3V na pribadač „VDD“, te je uzemljen je preko spoja pribadača „VSS“ na „GND“.

Pribadači PA11 i PA12 postavljeni su kao digitalni izlazi i koriste se za primanje i slanje podataka od mikroupravljača na „CAN bus“ preko MCP2551 primopredajnika; PA11 služi za čitanje podataka te je spojen na „RXD“ pribadača primopredajnika, dok PA12 služi da stanje podataka i spojen je na „TXD“ pribadača primopredajnika.

2.2. Modul 004-00

Pojednostavljena shema spojeva mikroupravljača sa ulazima i izlazima na sam mikroupravljač.



Slika 2 - pojednostavljena shema Modula 004-00

Kratki opis pojednostavljene sheme Modula 004-00 (Slika 2):

Na mikroupravljač se dovodi napon od +3.3V na pribadač „VDD“, te je uzemljen je preko spoja pribadača „VSS“ na „GND“.

Pribadači PA11 i PA12 postavljeni su kao digitalni izlazi i koriste se za primanje i slanje podataka od mikroupravljača na „CAN bus“ preko MCP2551 primopredajnika; PA11 služi za čitanje podataka te je spojen na „RXD“ pribadač primopredajnika, dok PA12 služi da stanje podataka i spojen je na „TXD“ pribadač primopredajnika.

Pribadač PA3 postavljen je kao analogni ulaz i spojen je na izlazni napon senzora za struju („VOOUT“ pribadač senzora TMCS1123B5A).

Pribadač „Vref+“ spojen je na napon od +4.096V i služi za dovođenje referentnoga napona na sam mikroupravljač; pribadač „Vref-“ uzemljen je preko spoja na „GND“.

Pribadač PG11 postavljen je kao digitalni ulaz, spojen je na tipkalo i na njemu se koristi prekidna rutina s vanjskim izvorom; konfiguracija *push-pull*, *pull-up*.

Pribadači PE9, PE8, PE7, PG1, PG0, PF15, PF14 i PF13 postavljeni su kao digitalni izlaz i služe za uključivanje/isključivanje LED, spojeni su na LED; konfiguracija *push-pull*, *no-pull*.

Pribadači PA0, PC0, PA1, PC1, PA2 i PC2 postavljeni su kao digitalni izlazi, gdje se PA pribadači koriste za slanje PWM pulsa na upravljač tranzistora (eng. *gate driver*) (upravlja MOSFET-om koji upravlja dovod ulaznog napona na aktuator), dok se PC pribadači koriste za slanje visokog ili niskog stanja na upravljač tranzistora (upravlja MOSFET-om koji upravlja spajanje aktuatora na uzemljenje); PA0 je spojen na „HIN“ pribadač prvog upravljača, PC0 je spojen na „LIN“ pribadač prvog upravljača, PA1 je spojen na „HIN“ pribadač drugog upravljača, PC1 je spojen na „LIN“ pribadač drugog upravljača, PA2 je spojen na „HIN“ pribadač trećeg upravljača, PC2 je spojen na „LIN“ pribadač trećeg upravljača.

Pribadači PE2, PE3 i PE4 postavljeni su kao digitalni ulazi, služe za čitanje stanja polariteta BEMF-a i za pokretanje prekidnih rutina, spojeni su na usporednike (eng. *comparator*) čiji je izlazni signal visok ukoliko je polaritet BEMF-a veći od uzemljenja, a niski ukoliko je manji; PE2 je spojen na izlaz prvog usporednika, PE3 je spojen na izlaz drugoga usporednika, PE4 je spojen na izlaz trećega usporednika.

3. Postavljanje mikroupravljača za Modul 000-00

Postavljanje postavki mikroupravljača za Modul 000-00 izvađa se u programu „CubeMX“;

Nakon pokretanja programa „STM32CubeMX“, u odjeljenju „Novi projekt“ (eng. *New Project*) pritisnuti gumb „Pristup izborniku mikroupravljača“ (eng. *Access to MCU Selector*).

U gornjem lijevom kutu, u izborniku „Komercijalni Broj Dijela“ (eng. *Commercial Part Number*), upise se komercijalni broj našeg mikroupravljača; STM32F730Z8T6

Odaberemo nas mikroupravljač, te zatim u gornjem desnom kutu pritisnemo gumb „Pokreni Projekt“ (eng. *Start Project*).

U lijevom izborniku odaberemo odjeljenje „Mjeritelji vremena“ (eng. *timers*), te zatim pod mjeriteljem vremena „TIM3“, pod odjeljenjem „Način Rada“ (eng. *Mode*) na izborniku „Izvor Takta“ (eng. *Clock Source*) označimo „Unutarnji Sat“ (eng. *Internal Clock*), zatim, za isti brojač, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „15999“, a pod izbornikom „Razdoblje Brojanja“ (eng. *Counter Period*) unesemo broj „99“; time smo postavili mjerač vremena koji mjeri vrijeme svakih 0.1s (100ms) i to na način da tih odmjerenih 0.1s odmjeri u komadićima od 0.001s (1ms); frekvencija mjerača vremena je 10Hz, a frekvencija promjene vrijednosti brojača je 1000Hz.

U lijevom izborniku odaberemo odjeljenje „Povezanost“ (eng. *Connectivity*), te zatim pod vrstom povezanosti „CAN1“, pod odjeljenjem „Način Rada“ (eng. *Mode*) potvrdimo stanje kao aktivno (eng. *Activated*), zatim, za istu vrstu povezanosti, u odjeljenju „NVIC Postavke“ (eng. *NVIC Settings*), potvrdimo način rada (eng. *Enabled*) pod redcima „CAN1 TX interrupts“ i „CAN1 RX0 interrupts“, zatim, za istu vrstu povezanosti, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod odjeljenjem „Parametri Vremena Bita“ (eng. *Bit Timings Parameters*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „4“, pod izbornikom „Vremenski Kvanti u Bitnom Segmentu 1“ (eng. *Time Quanta in Bit Segment 1*) unesemo broj „13“, pod izbornikom „Vremenski Kvanti u Bitnom Segmentu 2“ (eng. *Time Quanta in Bit Segment 2*) unesemo broj „2“; time smo postigli brzinu prijenosa podataka „CAN bus“ komunikacijom od 250 kbit/s.

4. Postavljanje mikroupravljača za Modul 004-00

Postavljanje postavki mikroupravljača za Modul 004-00 izvađa se u programu „CubeMX“;

Nakon pokretanja programa „STM32CubeMX“, u odjeljenju „Novi projekt“ (eng. *New Project*) pritisnuti gumb „Pristup izborniku mikroupravljača“ (eng. *Access to MCU Selector*).

U gornjem lijevom kutu, u izborniku „Komercijalni Broj Dijela“ (eng. *Commercial Part Number*), upise se komercijalni broj našeg mikroupravljača; STM32F730Z8T6

Odaberemo nas mikroupravljač, te zatim u gornjem desnom kutu pritisnemo gumb „Pokreni Projekt“ (eng. *Start Project*).

U lijevom izborniku odaberemo odjeljenje „Analogno“ (eng. *Analog*), te zatim pod analognim „ADC1“, pod odjeljenjem „Način Rada“ (eng. *Mode*)) potvrdimo stanje kao ulaza broj 3 (označeno „IN3“); time smo postavili analogni ulaz na pribadač PA3.

U lijevom izborniku odaberemo odjeljenje „Mjeritelji vremena“ (eng. *timers*), te zatim pod mjeriteljem vremena „TIM2“, pod odjeljenjem „Način Rada“ (eng. *Mode*) na izborniku „Izvor Takta“ (eng. *Clock Source*) označimo „Unutarnji Sat“ (eng. *Internal Clock*), zatim, u izbornicima kanal 1, kanal 2 i kanal 3 (eng. *Channel 1*, *Channel 2*, *Channel 3*) odaberemo opciju „Stvaranje PWM CHx“ (eng. *PWM generation*) gdje „x“ predstavlja broj kanala, za isti brojač, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „3“, a pod izbornikom „Razdoblje Brojanja“ (eng. *Counter Period*) unesemo broj „99“; time smo postavili stvaranje PWM signala od dužine signala 25us koji je podijeljen na komadiće od 0.25us; frekvencija PWM signala je 40kHz.

U lijevom izborniku odaberemo odjeljenje „Mjeritelji vremena“ (eng. *timers*), te zatim pod mjeriteljem vremena „TIM3“, pod odjeljenjem „Način Rada“ (eng. *Mode*) na izborniku „Izvor Takta“ (eng. *Clock Source*) označimo „Unutarnji Sat“ (eng. *Internal Clock*), zatim, za isti brojač, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „15999“, a pod izbornikom „Razdoblje Brojanja“ (eng. *Counter Period*) unesemo broj „99“; time smo postavili mjerač vremena koji mjeri vrijeme svakih 0.1s (100ms) i to na način da tih odmjerjenih 0.1s odmjeri u komadićima

od 0.001s (1ms); frekvencija mjeraća vremena je 10Hz, a frekvencija promjene vrijednosti brojača je 1000Hz.

U lijevom izborniku odaberemo odjeljenje „Mjeritelji vremena“ (eng. *timers*), te zatim pod mjeriteljem vremena „TIM4“, pod odjeljenjem „Način Rada“ (eng. *Mode*) na izborniku „Izvor Takta“ (eng. *Clock Source*) označimo „Unutarnji Sat“ (eng. *Internal Clock*), zatim, za isti brojač, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „1599“, a pod izbornikom „Razdoblje Brojanja“ (eng. *Counter Period*) unesemo broj „9999“; time smo postavili mjerac vremena koji mjeri vrijeme svakih 1s i to na način da tih odmjerenih 1s odmjeri u komadićima od 0.0001s (0.1ms); frekvencija mjeraća vremena je 1Hz, a frekvencija promjene vrijednosti brojača je 10kHz.

U lijevom izborniku odaberemo odjeljenje „Mjeritelji vremena“ (eng. *timers*), te zatim pod mjeriteljem vremena „TIM5“, pod odjeljenjem „Način Rada“ (eng. *Mode*) potvrdimo „Unutarnji Sat“ (eng. *Internal Clock*), zatim, za isti brojač, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „1“, a pod izbornikom „Razdoblje Brojanja“ (eng. *Counter Period*) unesemo broj „3999999“; time smo postavili mjerac vremena koji mjeri vrijeme svakih 1s i to na način da tih odmjerenih 0.5s odmjeri u komadićima od 0.125us; frekvencija mjeraća vremena je 2Hz, a frekvencija promjene vrijednosti brojača je 8MHz.

U lijevom izborniku odaberemo odjeljenje „Povezanost“ (eng. *Connectivity*), te zatim pod vrstom povezanosti „CAN1“, pod odjeljenjem „Način Rada“ (eng. *Mode*) potvrdimo stanje kao aktivno (eng. *Activated*), zatim, za istu vrstu povezanosti, u odjeljenju „NVIC Postavke“ (eng. *NVIC Settings*), potvrdimo način rada (eng. *Enabled*) pod redcima „CAN1 TX interrupts“ i „CAN1 RX0 interrupts“, zatim, za istu vrstu povezanosti, u odjeljenju „Postavke Parametara“ (eng. *Parameter Settings*), pod odjeljenjem „Parametri Vremena Biti“ (eng. *Bit Timings Parameters*), pod izbornikom „Pretskaler“ (eng. *Prescaler*) unesemo broj „4“, pod izbornikom „Vremenski Kvanti u Bitnom Segmentu 1“ (eng. *Time Quanta in Bit Segment 1*) unesemo broj „13“, pod izbornikom „Vremenski Kvanti u Bitnom Segmentu 2“ (eng. *Time Quanta in Bit Segment 2*) unesemo broj „2“; time smo postigli brzinu prijenosa podataka „CAN bus“ komunikacijom od 250 kbit/s.

Na shematskom prikazu mikroupravljača (desna strana prozora), u padajućem izborniku koji se postigne pritiskom na pribadača, za pribadač PE2 se odabere opcija „GPIO_EXTI2“,

za pribadač PE3 se odabere opcija „GPIO_EXTI3“, za pribadač PE4 se odabere opcija „GPIO_EXTI4“ te za pribadač PG11 se odabere opcija „GPIO_EXTI11“, zatim, u lijevom izborniku odaberemo odjeljenje „Jezgra Sustava“ (eng. *System Core*), te zatim pod odjeljenjem „NVIC“ potvrdimo način rada (eng. *Enabled*) pod redcima „EXTI line2 interrupt“, „EXTI line3 interrupt“, „EXTI line4 interrupt“ i „EXTI line[15:10] interrupts“. Nakon toga, u lijevom izborniku odaberemo odjeljenje „Jezgra Sustava“ (eng. *System Core*), te zatim pod odjeljenjem „GPIO“, pod stupcem „GPIO Način Rada“ (eng. *GPIO mode*), za retke „PE2“, „PE3“ i „PE4“ u padajućem izborniku odaberemo opciju „External Interrupt Mode with Rising/Falling edge trigger detection“, a pod istim stupcem, ali redom „PG11“ u padajućem izborniku odaberemo opciju „External Interrupt Mode with Rising edge trigger detection“.

Na shematskom prikazu mikroupravljača (desna strana prozora), u padajućem izborniku koji se postigne pritiskom na pribadač, za pribadače PF13, PF14, PF15, PG0, PG1, PE7, PE8 i PE9 se odabere opcija „GPIO_Output“, zatim, u lijevom izborniku odaberemo odjeljenje „Jezgra Sustava“ (eng. *System Core*), te zatim pod odjeljenjem „GPIO“, pod stupcem „GPIO Pull-up/Pull-down“, za sve retke, u padajućem izborniku treba odabrati opciju „No pull-up and no pull-down“ osim za red PG11, gdje treba odabrati opciju „Pull-up“.

5. Opis rada Modula – KTM

U glavnome programu, pri inicijalizaciji kôda da „CAN bus“ komunikaciju, za oba modula se koristi isti kod, jedino što se razlikuje je da se za Modul 000-00 koristi ID većeg prioriteta (TxHeader.StdId = 127;), dok Modul 004-00 koristi ID manjega prioriteta (TxHeader.StdId = 255;)

Vrijedi napomenuti da se, radi jednostavnog primjera, tj. zbog toga što nema više od 2 mikroupravljača, ne koriste nikakvi filteri pri primanju podataka.

5.1. Opis rada Modula 000-00 - KTM

Uloge Modula 000-00, a da obuhvaćaju gradivo KTM-a, su;

5.1.1. Intervalno mijenjanje vrijednosti PWM pulsa, te slanje istoga

Unutar beskonačne while() petlje se nalazi logika kojom Modul 000-00 mijenja vrijednost PWM pulsa na način da kreće sa vrijednošću od 50%, a zatim svakih 10s se mijenja za +/- 25% na način da vrijednost PWM pulsa prati sljedeći slijed; 50%, 75%, 50%, 25%, ...(ponavljanje).

Pri svakoj promjeni vrijednosti PWM pulsa, trebaju se poslati sljedeće informacije Modulu 004-00; šalje se 8 bajta (64bita), zadnja dva bajta su vrijednost PWM pulsa (16bitna vrijednost u rasponu od 100-4000 gdje 100 predstavlja 0%, a 4000 predstavlja 100%), zatim narednih 4 bajta je 0, u sljedećem bajtu zadnja dva bita predstavljaju željeni smjer gibanja aktuatora (dva uzastopna ista bita, gdje visoko stanje znaci normalan smjere gibanja, dok nisko znaci obrnuti smjer gibanja), sljedeća dva bita su 0, a nakon toga ponovno dva bita koji predstavljaju poželjno stanje aktuatora (visoko stanje znaci da želimo da aktuator bude uključen, dok nisko stanje znaci da želimo da bude isključen), prvi bajt (preostali bajt, prvi bajt) je jednak 0b11110000 koji znaci da u poslanom paketu se sadrže varijable.

Primjer poslanog paketa;

0b11110000 0b00110011 0b00000000 0b00000000

0b00000000 0b00000000 0b00000000 0b01100100

Gdje je; PWM puls = 100 (0%), smjer gibanja = 11 (normalan), poželjno stanje aktuatora = 11 (uključen), prvi bajt = 11110000 (u paketu ima podataka).

5.1.2. Intervalno slanje zahtjeva za povratne informacije/primanje informacija

Unutar beskonačne while() petlje nalazi se logika kojom Modul 000-00 svakih 1.5s šalje zahtjev za povratne informacije Modulu 004-00.

Zahtjev se šalje na način da pošalje poruku od 8 bajta, gdje je prvi bajt jednak 0b00001111 (0b00001111 znaci da je paket prazan i da se od primatelja zahtjeva povratna informacija).

Primjer poslanog paketa;

0b00001111 0b00000000 0b00000000 0b00000000

0b00000000 0b00000000 0b00000000 0b00000000

Gdje je; prvi bajt = 00001111 (paket je prazan, zahtijevam povratne informacije).

5.1.3. Intervalno obrađivanje primljenih podataka

Unutar beskonačne while() petlje nalazi se logika kojom Modul 000-00 obrađuje podatke koje primi od Modula 004-00, svaki puta kada ih primi.

Pošiljka koju Modul 000-00 prima je uvijek 8 bajta, gdje zadnja dva bajta su vrijednost PWM pulsa (16bitna vrijednost u rasponu od 100-4000 gdje 100 predstavlja 0%, a 4000 predstavlja 100%), zatim narednih 4 bajta je vrijednost izmjerene struje [u mili Amperima], u sljedećem bajtu zadnja dva bita predstavljaju trenutni smjer gibanja aktuatora (dva uzastopna ista bita, gdje visoko stanje znaci normalan smjer gibanja, dok nisko znaci obrnuti smjer gibanja), sljedeća dva bita su smjer protoka struje (gdje visoko stanje predstavlja obrnuti smjer protoka, dok nisko predstavlja normalan), a nakon toga ponovno dva bita koji predstavljaju trenutno stanje aktuatora (visoko stanje znaci da je aktuator bude uključen, dok nisko stanje znaci da je isključen), prvi bajt (preostali bajt) je jednak 0b11110000 koji znaci da u poslanom paketu se sadrže varijable.

Primjer primljenog paketa;

0b11110000 0b00110011 0b00000000 0b00000000

0b00000000 0b01100101 0b00001000 0b00000010

Gdje je; PWM puls = 2050 (50%), smjer gibanja = 11 (normalan), smjer protoka struje = 00 (normalan), poželjno stanje aktuatora = 11 (uključen), prvi bajt = 11110000 (u paketu ima podataka).

5.1.4. Intervalno primanje podataka

Unutar callback() funkcije, koja se poziva svaki puta Modul 000-00 primi podatke, se nalazi kratka logika koja stavlja zastavicu (označava izvršenu radnju) koja predstavlja da je modul primio podatke da se treba provjeriti je li nakon primljene poruke treba obraditi podatke ili ih treba poslati.

5.2. Opis rada Modula 004-00 - KTM

Uloge Modula 004-00, a da obuhvaćaju gradivo KTM-a, su;

5.2.1. Intervalno slanje povratnih informacija

Nakon određenog intervala vremena Modul 004-00 dobiva zahtjev da pošalje informacije Modulu 000-00, unutar while() petlje se nalazi logika kojom to taj modul radi.

Svaki puta kada primi zahtjev za slanje povratnih informacija, Modulu 000-00 se šalje poruka od 8 bajta; zadnja dva bajta su vrijednost PWM pulsa (16bitna vrijednost u rasponu od 100-4000 gdje 100 predstavlja 0%, a 4000 predstavlja 100%), zatim narednih 4 bajta je vrijednost izmjerene struje [u mili Amperima], u sljedećem bajtu zadnja dva bita predstavljaju trenutni smjer vrtnje motora (dva uzastopna ista bita, gdje visoko stanje znaci normalan smjere gibanja, dok nisko znaci obrnuti smjer gibanja), sljedeća dva bita su smjer protoka struje (gdje visoko stanje predstavlja obrnuti smjer protoka, dok nisko predstavlja normalan), a nakon toga ponovno dva bita koji predstavljaju trenutno stanje aktuatora (visoko stanje znaci da je aktuator bude uključen, dok nisko stanje znaci da je isključen), prvi bajt (preostali bajt) je jednak 0b11110000 koji znaci da u poslanom paketu se sadrže varijable.

5.2.2. Intervalno obrađivanje primljenih podataka

Nakon određenog intervala vremena Modul 004-00 prima informacije koje mu Modul 000-00 šalje, unutar while() petlje se nalazi logika kojom to taj modul radi

Pošiljka koju Modul 004-00 prima je uvijek 8 bajta, gdje zadnja dva bajta su vrijednost PWM pulsa (16bitna vrijednost u rasponu od 100-4000 gdje 100 predstavlja 0%, a 4000 predstavlja 100%), zatim narednih 4 bajta 0, u sljedećem bajtu zadnja dva bita predstavljaju željeni smjer vrtnje motora (dva uzastopna ista bita, gdje visoko stanje znaci normalan smjere gibanja, dok nisko znaci obrnuti smjer gibanja), sljedeća dva bita 0, a nakon toga ponovno dva bita koji predstavljaju željeno stanje aktuatora (visoko stanje znaci da je željeno da aktuator bude uključen, dok nisko stanje znaci da je željeno da bude isključen), prvi bajt (preostali bajt) je jednak 0b11110000 koji znaci da u poslanom paketu se sadrže varijable.

5.2.3. Intervalno primanje podataka

Unutar `callback()` funkcije, koja se poziva svaki puta Modul 004-00 primi podatke, se nalazi kratka logika koja stavlja zastavicu (označava izvršenu radnju) koja predstavlja da je modul primio podatke da se treba provjeriti je li nakon primljene poruke treba obraditi podatke ili ih treba poslati.

6. Opis rada Modula 004-00 - URS

Glavna uloga Modula 004-00 je upravljanje aktuatorom (u ovome slučaju to je istosmjerni elektromotor bez četkica gdje su zavojnice spojene u trokut konfiguraciju ili zvijezda konfiguraciju sa 3 žice).

6.1. ADC – čitanje iznosa struje

Unutar while() petlje nalazi se logika koja svakih 10ms pokreće ADC i izmjeri iznos struje.

Vrijedi napomenuti da se za ADC koristi referentni napon od +4.096V, a ne +3.3V. Prema podatkovnom listu proizvođača senzora TMCS1123B5A, izlazni napon od 1.650V predstavlja protok struje od 0A, a osjetljivost senzora je 150mV/A, stoga se koristi prikladna logika za pretvorbu digitalnog signala u dekadsku vrijednost etalona, u našem slučaju, u mili Ampere.

6.2. Obrada PWM signala

Unutar beskonačne while() petlje nalazi se logika kojom u svakom ciklusu određujemo željenu vrijednost PWM pulsa.

Posto se iščekuje da nakon svakog određenog intervala željeni iznos vrijednosti PWM pulsa mijenja na način da Modul 004-00 primi informaciju o željenom iznosu vrijednosti PWM pulsa od Modula 000-00, u svakome ciklusu se izračunava željeni iznos vrijednosti PWM pulsa na način da primljenu vrijednost skaliramo na primjenjivu vrijednost, tj. primljena vrijednost bi se trebala nalaziti unutar intervala 100-4000, a primjenjiva vrijednost bi se trebala nalaziti u intervalu 0-100.

6.3. Pogon aktuatora

Aktuator može biti u 4 stanja; pokretanje, radno stanje, zaustavljanje i zastoje. Stoga je potrebna prikladna logika za omogućavanje tih stanja.

6.3.1. Pokretanje (radno stanje)

Unutar beskonačne while() petlje, ukoliko je to potrebno, nalazi se logika za pokretanje aktuatora.

Prvo se pokreće potrebni mjeritelj vremena, koji služi za mjerenje vremena između promjena polariteta BEMF-a (pomoću tog vremena, ukoliko su poznate karakteristike aktuatora, može se izračunati brzina; u našem slučaju brzina vrtnje za BLDC elektromotor

koja bi se računala formulom: $\frac{60}{t*2*n*p} = RPM$, gdje t oznacava nase izmjereno vrijeme, n broj polova, p broj parova polova i RPM okretaje u minuti). A nakon mjeritelja vremena se uključuje PWM izlaz sa iznosom pulsa od 10% na prvi izlaz, a drugi izlaz se uzemljuje. U teoriji, to bi trebalo pokrenuti dovoljan impuls da se nakon nekog kratkog vremena pokrene prekidna rutina za pokretanje daljnjih kombinacija izlaza.

6.3.2. Radno stanje

Unutar beskonačne while() petlje, ukoliko je to potrebno, nalazi se logika za radno stanje aktuatora u oba smjera (normalan i obrnut).

Ako je normalan smjer, priziva se funkcija za normalan smjer (vise objasnjeno u 8. Funkcije), a ako je smjer obrnut, priziva se funkcija za obrnut smjer (vise objasnjeno u 8. Funkcije).

6.3.3. Zaustavljanje (koćenje)

Unutar beskonačne while() petlje, ukoliko je to potrebno, nalazi se logika za zaustavljanje aktuatora.

Ako je potrebno zaustavljanje aktuatora, isključuju se svi PWM izlazi i svi izlazi se uzemljuju.

6.3.4. Zastoj (stanje mirovanja)

Unutar beskonačne while() petlje, ukoliko je to potrebno, nalazi se logika koja omogućuje ponovno pokretanje aktuatora.

Ako aktuator miruje 0.5s, stanje aktuatora se mijenja u „zastoj“, isključuje se mjeritelj vremena koji mjeri vrijeme između promjena polariteta BEMF-a, isključuju se svi izlazi (prijašnje su bili spojeni na uzemljenje, sada se stavljaju da „lebde“), te se omogućuje slijed za pokretanje.

6.4. Prikaz pomoću LED

Unutar beskonačne while() petlje, pri kraju petlje nalazi se logika kojom se uključuju LED sukladno s PWM pulsom.

Ovisno koliki je iznos PWM pulsa (0-100), uključuju se ili se isključuju 8 LED. Za svakih 12.5% PWM pulsa pali se dodatna LED kronološkim slijedom, tako da prikaz PWM pulsa izgleda kao ljestvica.

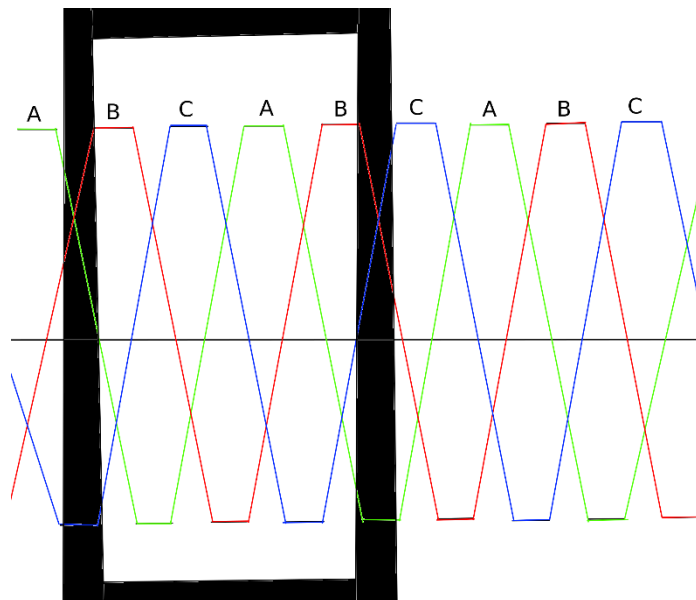
7. Funkcije

Unutar beskonačne while() petlje glavnog programa ima puno radnji koje se ponavljaju. Nisam sve, ali sam većinu takvih radnji definirao kao funkciju, tako da se, po potrebi, samo pozovu.

7.1. (void) funkcije

Unutar glavnog programa se nalaze nekoliko (void) funkcija; funkcija kod koje je definiran slijed uključivanja izlaza za normalan smjer gibanja, funkcija kod koje je definiran slijed uključivanja izlaza za obrnut smjer gibanja, funkcije za uključivanje određenih kombinacija izlaza, funkcije za pokretanje mjeritelja vremena, funkcija za provjeru polariteta BEMF-a, funkcija za pronalazak promjene polariteta BEMF-a.

Pomoću sljedeće slike je objasnjeno kako se određuje koja će se kombinacija izlaza koristiti.



Slika 3 - predviđeni prikaz promjene polariteta u vremenu

Na prijašnjoj slici, zelenom bojom (oznaka A) je označen polaritet (visok ili nizak, rampe predstavljaju prelazno stanje, u trenutku kada rampa sjeća srednju (neutralu) liniju, dešava se prekidna rutina) BEMF-a prvoga izlaza, isto tako, crvenom bojom (oznaka B) je označen drugi izlaz, a plavom (oznaka C) je treći izlaz. Za određivanje kombinacije izlaza koji bi trebali primati PWM puls, a koji bi trebali biti uzemljeni, prvo trebamo odrediti početnu točku u kojoj gledamo, za ovaj primjer objašnjavanja odabrao sam rastući brid trećega izlaza (plava boja (oznaka C) sjeća neutralnu). Nakon što odredimo točku gledanja, gledamo stanje

ostala dva izlaza (ignoriramo plavu boju (oznaka C) za ovaj primjer) u trenutku kada rastući brid sjeća neutralnu, te ako se izlaz nalazi iznad neutralne crte, taj izlaz je prije trenutka prelaza bio spojen na uzemljenje, a ako se nalazi ispod neutralne crte, taj izlaz je prije trenutka prelaza bio spojen na PWM puls; u primjeru A je primao PWM puls, a B je bio uzemljen. Tako se napravi za ostalih 5 slučajeva (sveukupno 6 kombinacija). Pomoću tih kombinacija se određuje smjer vrtnje. Primjer;

Tri uzastopna sjecanja neutralne crte;

1. A pada – C je PWM puls, B je uzemljen
2. C raste (već objašnjeno) – A je PWM puls, B je uzemljen
3. B pada – A je PWM puls, C je uzemljen

Ukoliko je željeni smjer normalan, a mi se nalazimo u trenutku kada C raste, prebacujemo se na kombinaciju A je PWM puls, C je uzemljen (kombinaciju prije B pada).

Ukoliko je željeni smjer obrnut, a mi se nalazimo u trenutku kada C raste, prebacujemo se na kombinaciju C je PWM puls, B je uzemljen (kombinaciju prije A pada).

7.1.1. Normalan smjer radnje

Funkcija koja ne prima nikakve argumente i izvršava primjerene kombinacije za normalan smjer vrtnje (objašnjeno u odjeljenju 8.1.).

7.1.2. Obrnut smjer radnje

Funkcija koja ne prima nikakve argumente i izvršava primjerene kombinacije za obrnut smjer vrtnje (objašnjeno u odjeljenju 8.1.).

7.1.3. Uključivanje kombinacija izlaza

U pitanju su 6 funkcija (za 6 mogućih kombinacija) koje kao argument primaju PWM puls, te sukladno o kombinaciji, jedan izlaz se spaja na PWM puls, neki drugi na uzemljenje, a preostali se ne spaja na ništa (ostavljen je da „lebdi“).

7.1.4. Pokretanje mjeritelja vremena

Za svaki mjeritelj vremena napravljena je funkcija koja ne prima nikakav argument, te služi samo da pokrene mjeritelj vremena, vrati vrijednost njegova brojača na 0, te u posebnu varijablu sprema se zastavica da je mjeritelj uključen.

7.1.5. Provjera polariteta

Funkcija ne prima nikakav argument, te svaki puta kada se pozove provjerava polaritete na pribadačima PE2, PE3 i PE4.

7.1.6. Pronalazak promjene polariteta

Funkcija ne prima nikakve argumente, a uspoređuje prijašnje i trenutne polaritete da pronade promjenu, prema kojoj se dalje određuje koja se kombinacija izlaza koristi.

7.2. (callback) funkcije

Unutar glavnog programa, pod (callback) funkciju, definirano je sto program radi dok je pokrenuta prekidna rutina; ili je pritisnut tipkalo ili je došlo do promjene polariteta BEMF-a.

7.2.1. Pritisnuto tipkalo

Ukoliko je pritisnuto tipkalo, pokreće se prekidna rutina gdje se nalazi logika za „debouncing“. Nakon „debouncing“-a promjenjuje se željeno stanje aktuatora.

7.2.2. Promjena polariteta

Ukoliko se desi promjena polariteta, prvo se mijenja način rada pribadača PE2, PE3 i PE4 iz vanjski uvjetovanog prekida u ulaz, zatim se se provjeri stanje polariteta, provjeri se koja se promjena desila, provjerava se stanje aktuatora; ako je stanje pokretanja neka se promjeni u radno, sprema se prijašnje stanje mjeritelja vremena i ponovo se pokreće, te se pri kraju način rada pribadača vrati u vanjski uvjetovane prekide.

8. Zaključak

Ovime seminarskim uratkom se pokriva veći dio gradiva kolegija Ugradbeni računalni sustavi, kao što se pokriva i osnove jedne vrste asinkrone serijske komunikacije. Seminarski uradak sam po sebi je poprilično dobar edukacijski primjerak. Glavni program sadrži logiku koja bi se relativno lako mogla implementirati u nadolazećim projektima, te je projekt sam po sebi interesantna mozgalica. Ovaj koncept je napravljen s ciljem obavezne modifikacije i nadogradnje kôda; mjesta gdje bi se kôd mogao poboljšati su poprilično uočljiva, te je sam po sebi izrađen s ciljem dodavanja više mogućnosti (npr. označen je i postavljen prostor za implementaciju regulatora). U konačnici, ovaj seminarski uradak će dobro služiti kao podloga za kompliciraniji i bolji završni rad.

9. Privítak

Github

link

za

kôd:

https://github.com/mdergez/Seminarski_URS_i_KTM_Mario_Dergez

Mario Dergez