



Универзитет у Београду

Машински факултет

Мастер академске студије

Индустрија 4.0

КИБЕРНЕТСКО ФИЗИЧКИ СИСТЕМИ

ПРОЈЕКАТ

Оцена пројектног
задатка:

Предметни
наставници: проф. др Живана Јаковљевић

Предметни
сарадници: Душан Недељковић

Група: 3

Потпис наставника:

РБ	Презиме и име:	Бр.инд.	Потпис:
1.	Немања Јанев	4004/20	
2.	Јелена Цвијан	4010/20	
3.	Стефан Ковач	4003/20	
4.	Катарина Пантовић	4005/20	

Школска година: 2020/2021.

Садржај

Садржај.....	1
1. Увод	2
2. Опис компонената од којих се систем састоји.....	3
3. Шема повезивања компонената на mbed развојни систем.....	7
4. Избор протокола за комуникацију између паметних уређаја.....	9
5. Израда програма за управљање радом паметног сензора/актуатора	9
6. Физичка реализација система	13
7. Литература	17

1. Увод

У оквиру курса кибернетско физички системи, рађен је пројекат са циљем пројектовања и физичке реализације паметног обртног енкодера и паметног корачног мотора и на њима заснованог система за позиционирање (слика 1).

Katedra za proizvodno mašinstvo
KIBERNETSKO FIZIČKI SISTEMI

PROJEKTNI ZADATAK

GRUPA 3 Školska 2020/21 godina

Projektovanje i fizička realizacija pametnog obrtnog enkodera, pametnog koračnog motora i na njima zasnovanog sistema za pozicioniranje

Pametni koračni motor se sastoji od:

1. Koračnog motora Jiaxin JX57BYGH2115A
2. Drajvera koračnog motora DM556
3. Mikrokontrolera NXP LPC 1768 integrisanog u okviru mbed razvojnog sistema

Pametni enkoder se sastoji od:

1. Optičkog obrtnog enkodera Hedss ISC3806-003G-1000BZ3-5-24F
2. Mikrokontrolera NXP LPC 1768 integrisanog u okviru mbed razvojnog sistema

Koristeći navedene komponente, potrebno je da studenti izvrše projektovanje i izradu pametnog senzora i aktuatora i izvrše integraciju sistema upravljanja baziranog na ovim komponentama koji će da ciklično izvršava sledeću sekvencu obrtanja koračnog motora: 5 CW – 4 CCW – 2 CW – 3 CCW obrta.

Projektni zadatak treba da sadrži:

1. Opis komponentata od kojih se sistem sastoji;
2. Šemu povezivanja komponentata na mbed razvojni sistem;
3. Izbor protokola za komunikaciju između pametnih uređaja;
4. Izradu programa za upravljanje radom pametnog senzora/aktuatora;
5. Fizičku realizaciju sistema;
6. Testiranje i puštanje u rad sistema.

Opšte napomene:

1. Projekat radi grupa studenata; konačna ocena rada svakog studenta na projektu će biti izvedena na osnovu pojedinačne angažovanosti i motivisanosti studenta, kao i na osnovu ocene projekta u celini
2. Završeni projekat se predaje kao elaborat i realizovan sistem
3. Obavezno je redovno prisustvo na vežbama
4. Rok završetka projekta je kraj II semestra

Predmetni nastavnik: Prof. dr Živana Jakovljević

Слика 1- Задатак пројекта

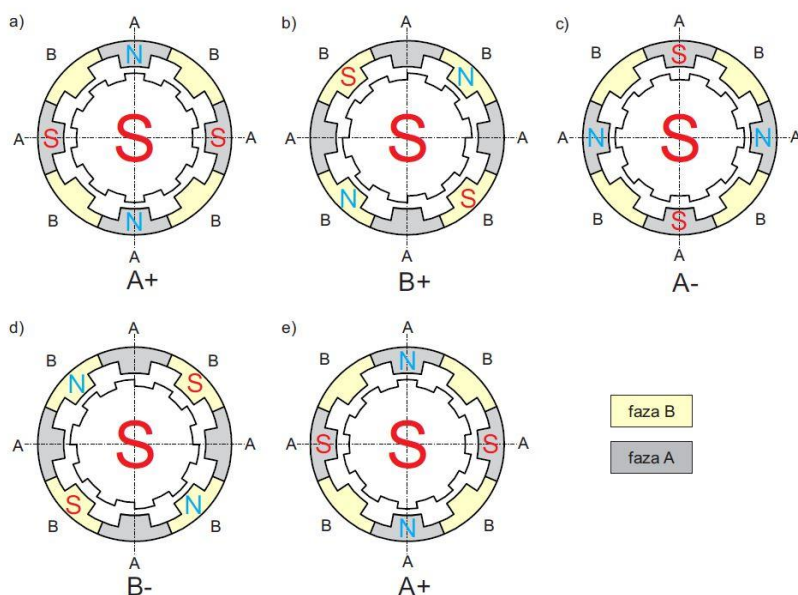
2. Опис компонената од којих се систем састоји

Као што је наведено у пројекту, паметни корачни мотор састоји се од 3 компоненте, а паметни енкодер од 2 компоненте и оне ће у наставку бити укратко описане.

а) Корачни мотор Jiaxin JX57BYGH2115

Сваки корачни мотор састоји се од статора који је назубљен и са намотајима и који најчешће има две фазе (А и Б) распоређене на 4 пола. Ротор је стални магнет и такође је назубљен. Одређени смер струје на фази А статора намагнетисаће вертикалне полове северно а хоризонталне јужно а супротни смер струје намагнетисаће их супротно. Слично је и за фазу Б. Ако се фаза А статора напоји позитивном струјом ротор ће заузети положај а) са слике 2. Када се искључи фаза А а фаза Б напоји позитивном струјом доћи ће до обртања ротора у смеру казаљке на сату за тачно одређен угао ротације. Секвенца облика А+Б+А-Б-А+ доведиће до континуалног обртања ротора у смеру казаљке на сату а секвенца А+Б-А-Б+А+ до обртања у супротном смеру. Угао ротације мотора (θ) зависи од броја зубаца на ротору (n) и броја пари полова на статору (p) и може се израчунати формулом 1:

$$\Theta = \frac{360^\circ}{np} \quad (1)$$



Слика 2- положаји корачног мотора у зависности од смера струје који пролази кроз фазе А и Б и редоследа активности фаза [слика преузета из презентације са предавања]

б) Драјвер корачног мотора D556

Драјвер је дизајниран да покреће двофазни или четворофазни мотор. Основне спецификације налазе се на слици 3.

Parameters	DM556T			
	Min	Typical	Max	Unit
Output Current	1.8	-	5.6(4.0 RMS)	A
Supply Voltage	20	24 - 48	50	VDC
Logic signal current	7	10	16	mA
Pulse input frequency	0	-	200	kHz
Minimal pulse width	2.5	-	-	μS
Minimal direction setup	5.0	-	-	μS
Isolation resistance	500			MΩ

Слика 3- Основна кофигурација драјвера корачног мотора DM556 [слика преузета из UserManuel-a за драјвер]

На слици 4 видимо описане функције пинова. Пин GND потребно је повезати на 0-ти потенцијал, али је ту битно водити рачуна да су GND свих компонената исти. Као што је предложено, драјвер се напаја извором од 24V из батерије (DC power supply). Пинови A+, A-, B+ и B- повезују се на мотор и сигнали са ових пинова ће довести до конфигурације истоимених фаза на корачном мотору. С обзиром да сигнал на драјвер треба да дође са микроконтролера чији је напон на излазу око 3.3V потребно га је повезати на оптокаплере чиме ће се, по појави сигнала на диоди оптокаплера, фотоелектричним ефектом активирати транзистор и тиме пинови на драјверу поставити на напон од око 24V.

Pin Function	Details
GND	Power supply ground connection.
+V	Power supply positive connection. Suggest 24-48VDC power supply voltage
A+, A-	Motor Phase A connections. Connect motor A+ wire to A+ Pin; motor A- wire to A-
B+, B-	Motor Phase B connections. Connect motor B+ wire to B+ Pin; motor B- wire to B-

Слика 4- функционалности пинова драјвера [слика преузета из UserManuel-a за драјвер]

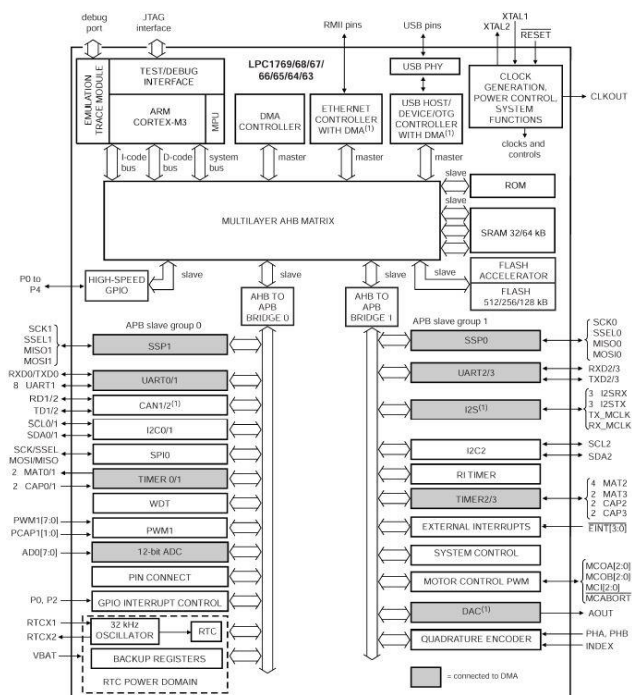
Напајање драјвера врши се помоћу слејв микроконтролера са PWM сигналом. Начин функционисања и зависности брзине и правца ротације мотора од вредности сигнала послатог са микроконтролера описан је у поглављу 5. Тај сигнал долази до пинова PUL+, PUL-, DIR+, DIR- (слика 5).

Pin Function	Details
PUL+	<u>Pulse signal:</u> Pulse active at rising edge; 4-5V when PUL-HIGH, 0-0.5V when PUL-LOW. Minimal pulse width of 2.5µs. Add a resistor for current-limiting at +12V or +24V input logic voltage (1K for +12V, 2k for +24V). The same as DIR and ENA signals.
PUL-	
DIR+	<u>DIR signal:</u> This signal has low/high voltage levels to represent two directions of motor rotation.. Minimal direction setup time of 5µs. Also swapping the connection of two wires of a coil (e.g. A+ and A-) to the drive will reverse motor direction.
DIR-	
ENA+	<u>Enable signal:</u> This signal is used for enabling/disabling the drive. High level +5V (NPN control signal) for enabling the drive and low level for disabling the drive. PNP and
ENA-	Differential control signals are on the contrary, namely Low level for enabling. By default it is left UNCONNECTED (ENABLED) .

Слика 5- функционалности пинова драјвера [слика преузета из UserManual-а за драјвер]

ц) Микроконтролер NXP LPC1768

У питању је 32-битни микроконтролер (слика 6). Као и сваки микроконтролер, располаже великим бројем различитих интерфејса за интеракцију са спољашним светом. Приступ функционалностима врши се коришћењем пинова микроконтролера. С обзиром да постоју пуно функционалности, пракса је да више интерфејса користи исти пин па је пре почетка коришћења потребно дефинисати која функционалност ће се вршити на одабраном пину. На слици 7 налазе се основне функционалности коришћеног микроконтролера.



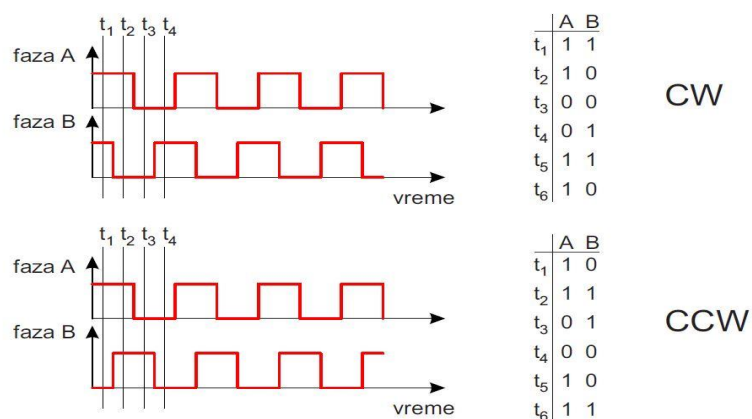
Слика 6- Блок дијаграм микроконтролера LPC 1768 [слика преузета из презентације са предавања]

Funkcionalnost	Karakteristike
Radna frekvenca	100MHz
Reseti	POR (Power-on reset) RESET pin Watchdog reset BOD (BrownOut Detection)
Fleš programska memorija	512KB
Memorija za podatke	64KB
Interapti	35
GPIO	70 (5 portova)
Tajmeri i brojači	4
PWM modul	1
Enkoderski ulaz	1
Serijska komunikacija	Ethernet MAC) USB 2.0 Universal Asynchronous Receiver Transmitter (UART) - 4 kom CAN (Controller Area Network) 2.0B - 2 kanala SPI (Serial Peripheral Interface) I ² C (Inter-Integrated Circuit) itd.
12-bitni analogno-digitalni konvertor	8 kanala
10-bitni digitalno-analogni konvertor	1 kanal
Analogni komparatori	2
Broj instrukcija	preko 100

Слика 7- Извод основних карактеристика микроконтролера LPC 1768 [слика преузета из презентације са предавања]

д) Оптички обртни енкодер

Садрже два пара емитера и детектора светлости, а сваки пар назива се фаза. Те две фазе називају се фаза А и фаза Б. У зависности од позиције диска, светлост ће се периодично детектовати што ће довести до електричне поворке импулса. Резолуција, односно број импулса по обрту који се може остварити уз помоћ једне фазе енкодера једнак је двоструком броју зареза. Ради повећања резолуције уводи се друга фаза те је укупан број импулса четворострук број зареза. Конкретно код енкодера коришћеног у пројекту, број зареза је 1000 те је резолуција 4000 импулса по једном кругу који направи диск енкодера (један корак мотора је ротација за $360^\circ/4000=0.09^\circ$). У односу на начин промене фаза А и Б могуће је одредити смер ротације мотора (слика 8).



Слика 8- смер ротације енкодера у зависности од комбинације импулса фаза [слика преузета из презентације са предавања]

Уколико се мало боље погледају таблице са вредностима, фаза за позитиван (CCW) и негативан (CW) смер ротације, може се лако приметити да у првом случају претходна вредност фазе А и фаза Б имају или 0 и 1 или 1 и 0 вредности а у случају CW 0 и 0 или 1 и 1. Дакле, ове две величине ће увек задовољавати логичку операцију EX ILI (слика 9). Ако је у питању позитивна операција резултат логичке операције ће бити 0 (false) а за негативан смер ротације 1 (true). Јасно је да, ако поставимо ову логичку операцију као услов if петље, програм ће извршити команде у петљи уколико је у питању негативан смер ротације и команде у else петљи ако је позитиван смер ротације. На овај начин је могуће након сваке ротације бележити број корака у одређеном смеру и на тај начин одредити колико је мотор корака па и кругова направио.

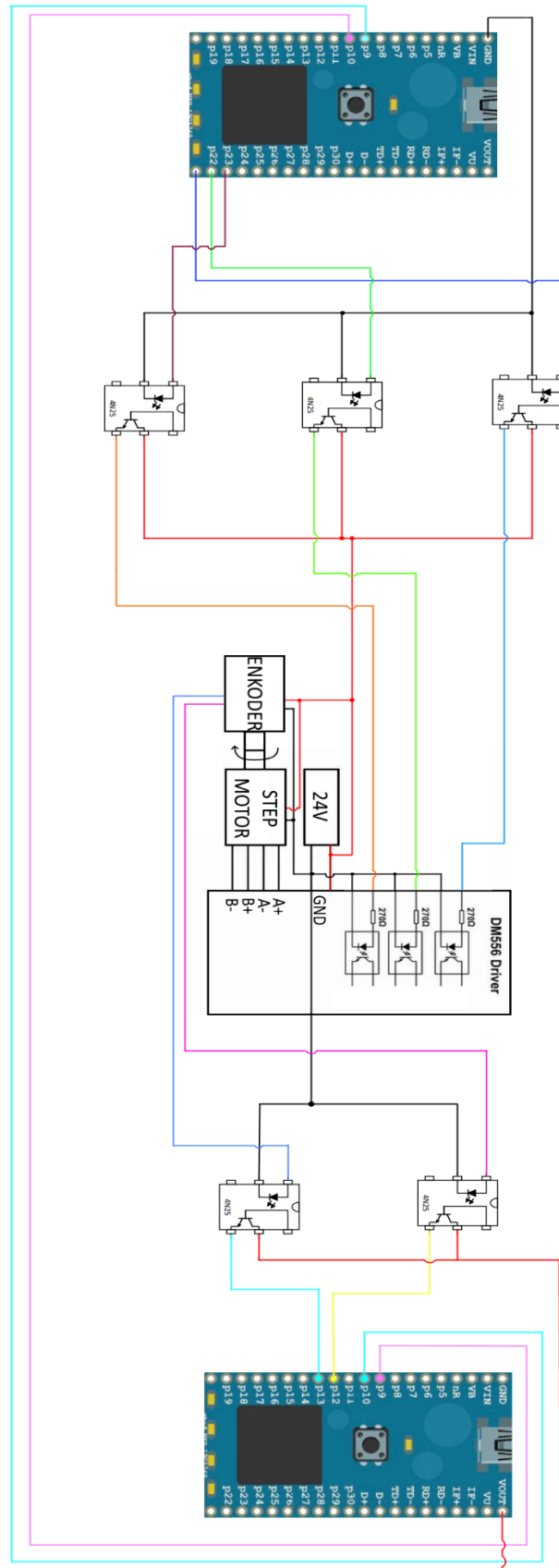
i	x_1	x_2	$y = x_1 \oplus x_2$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Слика 9- логичка операција EX ILI

3. Шема повезивања компонената на mbed развојни систем

На слици 10 приказана је шема повезивања компонената у електрично коло.

Микроконтролер којим управљамо драјвером је слејв, а онај везан за енкодер, мастер. С обзиром да драјвер ради на 24V, а излази из микроконтролера су на 3.3 V, управљање улазима у драјвер омогућено је помоћу три оптокаплера који ће пропустити струју кроз транзистор када диода на њима добије сигнал од микроконтролера. Транзистори оптокаплера, енкодер и мотор повезани су на спољашњи извор једносмерне струје (батерију) од 24V. Драјвер ће у зависности од улазних импулса на пиновима PUL и DIR, послати сигнал мотору преко пинова A+, A-, B+ B-. Мотор има на свом вратилу закачен енкодер који ће у зависности од стварне ротације мотора детектовати ротацију у позитивном и негативном смеру. Како је већ описано, енкодер ће имати за одређени смер ротације мотора одређену комбинацију 0 и 1 на пиновима А и Б што ће стизати преко оптокаплера на дигиталне улазе микроконтролера. Оптокаплери се у овом случају користе због тога што микроконтролер не сме на свом дигиталном улазу да има више од 3.3V а излазни сигнал из енкодера је свакако 24V јер он на том напону ради.



Слика 10- шема повезивања компоната у електрично коло

4. Избор протокола за комуникацију између паметних уређаја

Да би се остварила комуникација између мотора и енкодера потребно је увезати микроконтролере и тако креирати универзални асинхрони ресивер/трансмиситер (UART) тј. хардверски комуникациони уређај. UART се најчешће користи у комбинацији са неким комуникационим стандардом као што је, на пример, RS-232, RS485 RS422. У оквиру UART-а, подаци се преузимају са одговарајућих регистара и врши се њихово конвертовање у низ бита, затим њихов пренос и на крају, смештање информације у одговарајући регистар на пријемној страни.

Подаци које је потребно слати налазе се у TX регистру, а примљени подаци смештају се у RX регистар. У конкретној ситуацији, пинови 9 и 10 представљаће излаз из TX регистра, тј. слање у RX регистар другог контролера и улаз у RX регистар тј. примање података из TX регистра другог контролера.

С обзиром да се ради о асинхроној комуникацији (различит такт слања података) потребно је подесити и брзину комуникације тј baud rate. Комуникација мастер и слејв микроконтролера извршава се функцијама `getc()` која преузима податке `putc()` шаље податке.

5. Израда програма за управљање радом паметног сензора/актуатора

Свака промена на енкодеру, тј свака промена угла, забележена је преко промена фаза А и Б (поглавље 2). Да би се промене благовремено детектовале и бележиле потребно је да у тренутку дешавања промене програм изврши повећање бројача у ком смештамо број корака, тј. да изврши одговарајућу линију кода без обзира коју линију кода тренутно треба да изврши.

Да би се то остварило, користе се интерапти (прекиди). Када се догоди промена (улазна или силазна ивица сигнала на А или Б пину), она покреће захтев за интерапт и тада микроконтролер прекида извршавање програма без обзира у којој линији кода се налази и прелази на извршавање прекидне рутине. У оквиру прекидне рутине поставља се код који је потребно извршити када се догоди интерапт. Због тога се пинови на микроконтролеру 13, 12, и 11, који су повезани са пиновима енкодера на којима се читава сигнал промене фаза А Б и 3 дефинишу као `InterruptIn` што значи да се било каква промена на тим пиновима води као интерапт. (слика 11 и 12)

```

1  #include "mbed.h"
2  #include "Serial.h"
3
4  Serial pc(USBTX, USBRX);
5  Serial kontroler(p9, p10);
6
7  InterruptIn A (p13);
8  InterruptIn B (p12);
9  InterruptIn Z (p11);
10 int preth_A;
11 int br=0;
12
13 void funkcija() {
14
15     if(B^preth_A){
16         br++;}
17     else{
18         br--;}
19     preth_A=A;
20 }
21
22

```

Слика 11- дефинисање пинова и врсте комуникације на мастеру

```

1  #include "mbed.h"
2  #include "Serial.h"
3
4  Serial pc(USBTX, USBRX);
5  PwmOut voznja(p21);
6  DigitalOut direction(p22);
7
8  Serial kontroler(p9, p10);
9
10 int main() {
11     pc.baud(9600);
12     pc.format(8, Serial::None, 2);
13     kontroler.baud(9600);
14     kontroler.format(8, Serial::None, 2);
15
16     int primljeni_podaci;
17
18     primljeni_podaci=kontroler.getc();
19     if (primljeni_podaci==10){
20         pc.printf("Slave primio broj, %d\r\n",primljeni_podaci);
21     }
22
23     kontroler.putc(primljeni_podaci);
24     //SCW
25     direction = 0;
26     voznja.period(float(0.005));
27     voznja.write(0.3);
28
29

```

Слика 12- дефинисање пинова и врсте комуникације на слејву

Да би се проверила успостављеност комуникације између микроконтролера мастер-*slave*, у линијама кода 35-38 (слика 13 и 12) мастер микроконтролер треба да пошаље *slave* микроконтролеру неку информацију а затим *slave* треба да врати исту ту информацију. Када мастер микроконтролер прими информацију(која је смештена у променљивој с обзиром да се шаље број 10), на рачунару треба да се испише да је *slave* примио информацију, тј. број 10. Након тога почиње рад мотора и енкодера што ће бити описано у наставку.

```

23 int main() {
24
25     pc.baud(9600);
26     pc.format(8, Serial::None, 2);
27     kontroler.baud(9600);
28     kontroler.format(8, Serial::None, 2);
29     preth_A=A;
30
31
32
33     wait(3);
34
35     kontroler.putc(10);
36     int primljeni_podaci=kontroler.getc();
37     if (primljeni_podaci==10){
38         pc.printf("Slave_primio_broj, %d\r\n",primljeni_podaci);
39     }
40
41
42
43
44
45     A.fall(&funkcija);
46     A.rise(&funkcija);
47     B.fall(&funkcija);
48     B.rise(&funkcija);
49
50     //primljeni_podaci=kontroler.getc();
51

```

Слика 13- успостављање комуникације (мастер)

У овом пројекту било је потребно да се мотор окреће у различитим смеровима по некој секвенци која је задата преко броја корака: (-20000, 16000, -8000, 12000). На пример, 5 кругова у супротном смеру од казаљке на сату (5CCW) значи да број корака које мотор направи треба да буде -20000. Пошто енкодер има 2 фазе и 1000 зазора, један круг направиће након 4000 корака (4000 промена). Свака промена на улазима А и Б довешће преко интерапта до промене променљиве br (поглавље 2). Дакле, када br достигне вредност -20000 тј. када се мотор окрене 5 кругова супротно од казаљке на сату потребно је окренути смер ротације и извршити окретање по следећој вредности у секвенци. С обзиром да је мастер микроконтролер везан за енкодер, потребно је када се испуни ова вредност бројача на мастеру, послати информацију на слејв микроконтролер који управља радом мотора и на њему обрнути смер ротације (слика 14).

```

31 //4CCW
32 if (primljeni_podaci==1){
33     direction=1;
34     voznja.period(float(0.005));
35     voznja.write(0.3);
36     kontroler.putc(primljeni_podaci);}
37
38     primljeni_podaci=kontroler.getc();
39
40 //2CW
41 if (primljeni_podaci==2){
42     direction=0;
43     voznja.period(float(0.005));
44     voznja.write(0.3);
45     kontroler.putc(primljeni_podaci);}
46
47     primljeni_podaci=kontroler.getc();
48
49 //3CCW
50
51 if (primljeni_podaci==3){
52     direction=1;
53     voznja.period(float(0.005));
54     voznja.write(0.3);
55     kontroler.putc(primljeni_podaci);}
56
57     primljeni_podaci=kontroler.getc();
58
59 //kraj
60 if (primljeni_podaci==4){
61
62     voznja.write(0);
63     kontroler.putc(primljeni_podaci);}
64
65 }

```

Слика 14- обртање смера ротације на мотору помоћу слејв микроконтролера

На слејву се, у линијама кода 25-27 (Слика 14) успоставља правац и брзина окретања мотора. Наиме, променљива *voznja* се дефинише као PWM излаз и преко ње се подаци шаљу на драјвер у улаз који се зове PUL (pulse). Позивом *voznja.period* дефинишемо брзину ротације мотора (што је период доласка импулса краћи бржа је ротација мотора) а *voznja.write* позивом дефинишемо колики део једног периода има ненулти импулс (ако је вредност 1 онда ће импулс трајати све време, а ако је 0 неће бити импулса тако да се на тај начин обезбеђује гашење мотора). Правац окретања мотора додељује се пину *direction* који је дигитални излаз и шаље сигнал на пин DIR (*direction*) драјвера мотора.

У линијама кода 63-67 код мастера (Слика 15) дефинисан је улазак у *if* петљу уколико је достигнута вредност прве ротације, -20000. По уласку у петљу, слејву се шаље број 1. У слејв коду у бесконачној петљи, константно се проверавају услови променљиве у коју треба да стигне податак са мастера. Када је та променљива 1, слејв код ће ући у петљу и промениће правац окретања мотора променом променљиве *direction* са 0 на 1 и задавањем брзине окретања мотору. Такође ће послати мастеру информацију да је примио број 1 а онда ће мастер исписати на екрану да је извршена 1. ротација из секвенце и прећи ће на следећу задату вредност из секвенце. Све време *if* петље за проверу извршености ротација на енкодеру смештене су у бесконачну петљу тако да се и оне све време проверавају док се неки од услова не испуни. Након последње ротације, слејв зауставља мотор командом *voznja.write(0)* а мастер исписује на екрану да је крај програма.

```

52 int sekvenca[4] = {-20000, -4000, -12000, 0};
53
54 int brojacevic=0;
55 //SCW
56
57 int i=0;
58 while(i<4){
59     brojacevic++;
60     if(brojacevic%2000==0){
61         pc.printf("%d\r\n",br);
62     }
63 }
64 if(i%2==0){
65     if(br<=sekvenca[i]){
66         kontroler.putc(i+1);
67         primljeni_podaci=kontroler.getc();
68     }
69
70     if (primljeni_podaci==i+1){
71         pc.printf("Slave_primio_broj, %d\r\n",primljeni_podaci);
72         i++;
73     }
74 }
75 else{
76     if(br>=sekvenca[i]){
77         kontroler.putc(i+1);
78         primljeni_podaci=kontroler.getc();
79     }
80
81     if (primljeni_podaci==i+1){
82         pc.printf("Slave_primio_broj, %d\r\n",primljeni_podaci);
83         i++;
84     }
85 }
86 }
87 pc.printf("Kraj,%d\r\n",primljeni_podaci);

```

Слика 15- рачунање броја корака мотора преко енкодера и слање информације слејву за промену смера ротације

6. Физичка реализација система

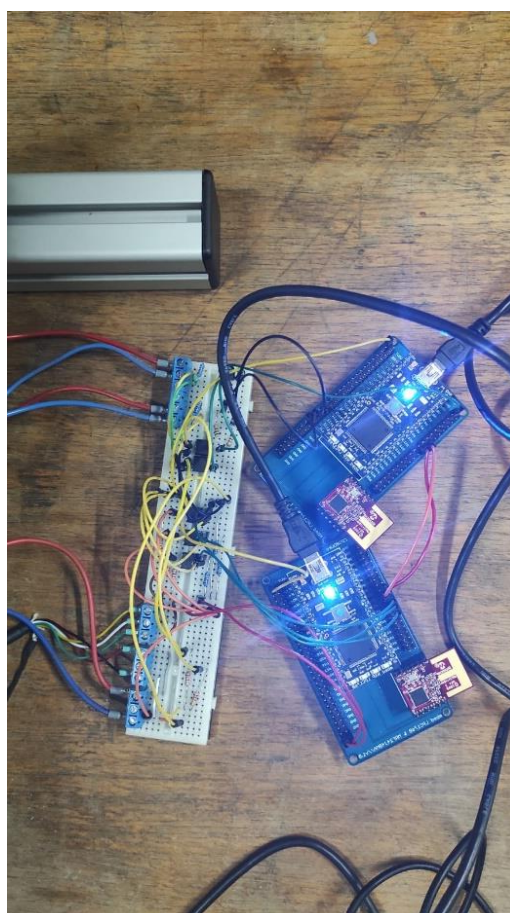
У овом поглављу налазе се фотографије експерименталне поставке паметног корачног мотора.



Слика 16- мотор



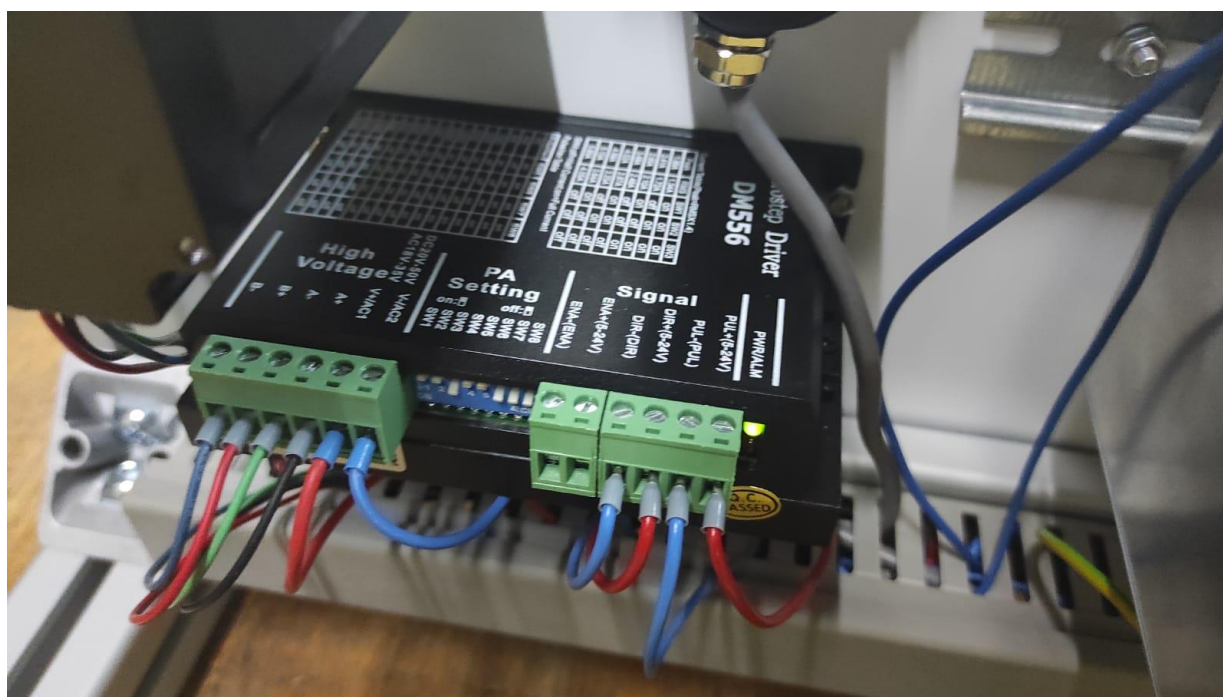
Слика 17- Извор напајања



Слика 18- Микроконтролери и електрична шема



Слика 18- енкодер, мотор и напајање



Слика 19- драјвер мотора



Слика 20- Енкодер



Слика 21- микроконтролери, енкодер и мотор

7. Литература

а) Прилог произвођача драјвера са сајта налази се овде. (<https://www.omc-stepperonline.com/download/DM556T.pdf>)

б) Живана Јаковљевић, Кибернетско физички системи- Изводи са предавања скинуто са сајта предмета. (<https://elearning.rcub.bg.ac.rs/moodle/enrol/index.php?id=1051>)