

**Факултет инжењерских наука
Универзитета у Крагујевцу**

Никола Д. Димитријевић

**Управљање мотора једносмерне струје
посредством Ардуино микроконтролера**

завршни рад

Крагујевац, 2020.

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Примењена механика и аутоматско управљање

Предмет: Сензори и актуатори

Број индекса: 126/2017

Никола Д. Димитријевић

Управљање мотора једносмерне струје посредством Ардуино микроконтролера завршни рад

Комисија за преглед и одбрану:

1. Проф. др Слободан Савић - ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____

Задатак рада је да се оствари управљање и надзор кретање једносмерног мотора. Произвољним концептом остварити управљање и надзор посредством микроконтролера Ардуино.

Циљ је управљање угаоном брзином једносмерног мотора и управљање позиције излазног вратила.

Препоручена литература:

1) Матијевић, М., Јакуповић, Г., Цар, Ј.: Рачунарски подржано мерење и управљање, Машински факултет у Крагујевцу, друго издање, Крагујевац, Септембар, 2008. – Преузето 10.08.2020. год.

2) Андрија С. Јестровић, „Пројектовање SCADA апликације и система заснованог на раду четири једносмерна и асинхрона мотора“, Дипломски - мастер рад, Факултет инжењерских наука Универзитета у Крагујевцу, Крагујевац, 2017. – Преузето 10.08.2020.год.

Крагујевац, 2020.

Ментор:

Проф. др Слободан Савић

САДРЖАЈ

РЕЗИМЕ.....	4
1. УВОД	5
1.1. ПИД контрола.....	5
2. ЈЕДНОСМЕРНИ МОТОР Escap 28D2R-219P	7
2.1. Енкодери Honto HTR-W-360-2-PP.....	7
2.2. Прилагодна електроника за једносмерне моторе.....	8
2.2.1. Техничко решење прилагодне електронике за једносмерне моторе	8
2.2.2. Израда електронике.....	14
2.2.3. Управљачки (PWM) сигнал	19
3. АРДУИНО РАЗВОЈНЕ ПЛАТФОРМЕ.....	21
4. УТИЦАЈ Н-МОСТА НА КАРАКТЕРИСТИКЕ ОБЈЕКТА УПРАВЉАЊА	22
4.1. Статичка карактеристика напон-брзина.....	22
4.2. Статичка карактеристика PWM-брзина	23
4.3. Статичка карактеристика PWM-напон.....	24
5. ПРИНЦИПИ ПРОГРАМИРАЊА КОНТРОЛЕРА И ПРИМЕР ЦЕЛОВИТОГ ПРОГРАМСКОГ РЕШЕЊА ЗА УПРАВЉАЊЕ И НАДЗОР МОТОРА	26
5.1. Пример целовитог програмског решења контролера на персоналном рачунару који је повезан са микроконтролером Ардуино и написан је у програмском језику Пајтон.....	28
5.2. Пример програмског решења у примени, извршења експеримента, обраде и презентације резултата експеримента	32
6. ЗАКЉУЧАК	35
7. ЛИТЕРАТУРА	36
Прилог А.....	37

РЕЗИМЕ

У раду су објашњени основни разлози аутоматизације једносмерног мотора као и основни принципи рада ПИД контроле.

Објашњено је како је састављена електроника за управљање радом једносмерног мотора, то јест техничко решење прилагодне електронике за једносмерни мотор, као и појам управљачког (PWM) сигнала. Као физичка рачунарска платформа која врши улогу контролера користи се „Arduino Uno R3“ развојна платформа. У даљем раду објашњен је утицај Н-моста на карактеристике објекта управљања. На крају су објашњени принципи програмирања контролера и пример целовитог програмског решења за управљање и надзор једносмерног мотора.

Кључне речи: ПИД, Ардуино, једносмерни мотор, управљање, PWM.

ABSTRACT

The paper explains the basic reasons for DC motor automation as well as the basic principles of PID control operation.

It is explained how the adapter electronics for a DC motor is assembled, that is, the technical solution of the adapter electronics for a DC motor, as well as the concept of the control (PWM) signal. The Arduino Uno R3 development platform is used as the physical computer platform that acts as a controller. In the further work, the influence of the H-bridge on the characteristics of the control object is explained. Finally, the principles of controller programming and an example of a complete software solution for control and supervision of a DC motor are explained.

Key words: PID, Arduino, DC motor, control, PWM.

1. УВОД

У данашње време потреба индустрије за аутоматизацијом индустријских процеса је веома велика. Један од основних актуатора који се користи у самим индустријским процесима јесте мотор једносмерне струје који спада у један од најбитнијих актуатора.

Како би једносмерни мотори могли да се користе на квалитетан начин потребно је остварити њихову прецизну контролу коришћењем неке врсте контролне петље, како би се од тог једносмерног мотора направио серво мотор.

Данас у индустрији машинске обраде и индустријски роботи покретани сервомоторима доприносе повећању продуктивности и квалитета обраде у практично свим гранама индустријске производње: аутомобилској индустрији, ваљаницама, текстилној индустрији, индустрији папира, прехранбеној индустрији и многим другим.

Постоје два облика контролне петље – отворена и затворена. Код отворене петље је контролер независан од излаза, док је код затворене петље управљање зависно од излаза. Практичан пример отворене спреге би био бојлер који греје воду на основу хронометра, док би затворена спрега била када би се додао термостат систему, како би бојлер грејао воду до жељене температуре, без обзира колико би му времена требало за то [1].



Слика 1. Повратна спрега [1]

1.1. ПИД контрола

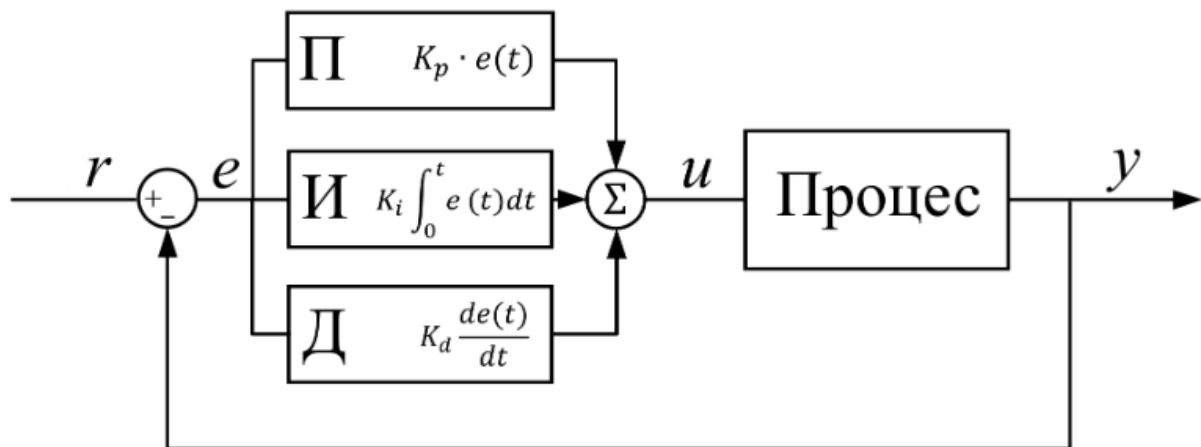
ПИД дејство, односно пропорционално, интегрално и диференцијално дејство представља врсту контроле која је јако заступљена у индустрији, где је процена да је више од 90% контролера засновано на ПИД управљању. Јако су популарни због просте структуре коју је лако применити у пракси. Предности су што се може користити и ако нам није познат математички модел објекта управљања и релативно лако је променити параметаре након његове инсталације унутар система аутоматског управљања. Данас је његова имплементација искључиво дигитална. Сложенији начини управљања се користи само када ПИД управљање није довољно да испуни задате спецификације [2].

Закон ПИД управљања је добијен искуствено и у идеализованом облику се може представити параметрима K - појачање, T_i - временска константа интегралног дејства и T_d - временска константа диференцијалног дејства. Закон управљања се састоји од три дејства – П, И и Д. П дејство представља пропорционално дејство прилагођава грешку у

тренутном времену, и ради једино ако грешка постоји. Повећањем K појачања се повећава брзина одзива и тачност система, али се губи на стабилности [2].

Интегрално дејство И генерише управљање на основу сигнала грешке прошлог времена. Уколико је сигнал грешке нула, то не значи да је управљање нула. И дејство доприноси елиминацији грешке стационарног стања, и користи се да би повећао тачност рада система у стационарном стању. Његово појачање T_i , уколико је превелико, може довести до дестабилизације система, или смањења релативне стабилности. И дејство повећава инертност и деградира динамичке перформансе одзива система. Са П дејством чини ПИ систем, који заједно са П регулатором, припада фамилији ПИД регулатора и често се користе у пракси [2].

Диференцијално дејство Д генерише управљање на основу будућег тренда сигнала грешке која се множи са T_d да би добили Д управљање. Д дејство повећава брзину и резерву стабилности система, али и осетљивост на високофреквенцијске сигнале, као што је мерни шум. Повећањем T_d је могуће у већој мери потиснути осцилације временског одзива система. На слици 2 је дат општи приказ ПИД контроле [2].



Слика 2. ПИД контрола [2]

2. ЈЕДНОСМЕРНИ МОТОР Escap 28D2R-219P

Једносмерни мотор има сталне магнете у статору. На ротор, односно арматуру, се доводи једносмерна струја, која индукује магнетно поље. Због деловања магнетног поља статора, ротор креће да се обрће. Брзина зависи од јачине магнетног поља, која зависи од напона струје. Ротор је везан на комутатор преко четкица које доводе струје на ротор. Како се индукује струја, један крај бива одбијен а други привучен и ротор прави полукруг. Комутатор сада окреће смер струје како би процес кренуо поново, јер у супротном би ротор стао [3].

Коришћени једносмерни мотор је Escap 28D2R-219P, који ради на максималном напону 12V и максималне брзине од 5800 обртаја у минути. У додатку је могуће наћи ширу документацију мотора. У себе има уграђен енкодер са 144 подеока, односно резолуције од 2.5° , који даје синусни излазни сигнал. Тај сигнал је потребно претворити у правоугаони. За управљање мотором нам је потребан и H мост. Рад обе компоненте ће бити објашњен у следећем поглављу [2].



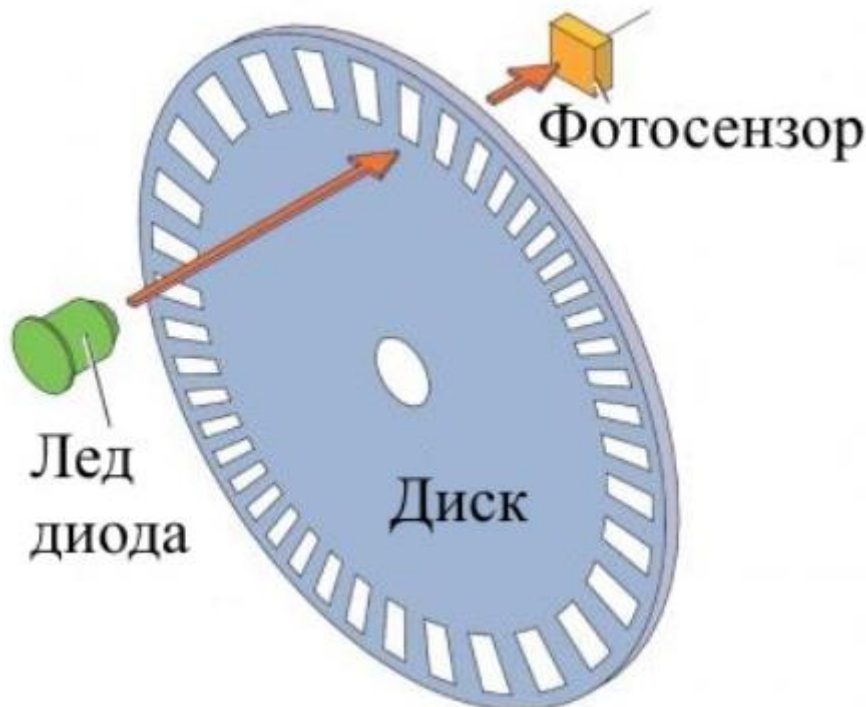
Слика 3. Escap 28D2R-219P мотор [2]

2.1. Енкодери Honto HTR-W-360-2-PP

Енкодери су сензори који служе за мерење угла окретања. Могу бити апсолутни и инкрементални. Апсолутни енкодери имају диск који се састоји од више концентричних прстенова при чему сваки следећи гледајући од центра ка ободу има двоструко више подеока. Вредност угла се одређује тако што извор светла осветљава део диска, тако да захвати све прстенове. Са друге стране се налазе фотодетектори који детектују да ли светлост пролази, и када пролази пуштају напон. На тај начин добијамо бинарни код на излазу енкодера који представља вредност измереног угла [4].

Инкрементални енкодери имају прстен да прорезима кроз које пролази светлост, која пада на фоторецептор. Пошто он генерише један импулс по прорезу, довољан је један диск са прстеновима. Ми добијамо само сигнал, тако да је потребно додати бројач који броји импулсе, на основу којих налазимо угао. Прво налазимо тачан положај тако што поделимо 360° са бројем прореза на диску, а затим помножимо са бројем добијених импулса. Да би смо одредили смер обртања потребан је још један прстен са фазно помереним прорезима у односу на први. Тако добијени сигнал касни или жури у односу

на први, зависно од смера, па га је могуће одредити. Сви енкодери које ми користимо су инкрементални [4].



Слика 4. Рад енкодера [4]

2.2. Прилагодна електроника за једносмерне моторе

Потребно је направити прилагодну електронику која ће синусоидни сигнал енкодера једносмерног мотора претворити у правоугаони. Излаз електронике би требало да ствара импулсе амплитуде за 24V или 5V, у зависности од напајања, да би иста електроника могла да се користи и за ПЛЦ и за Ардуино [2].

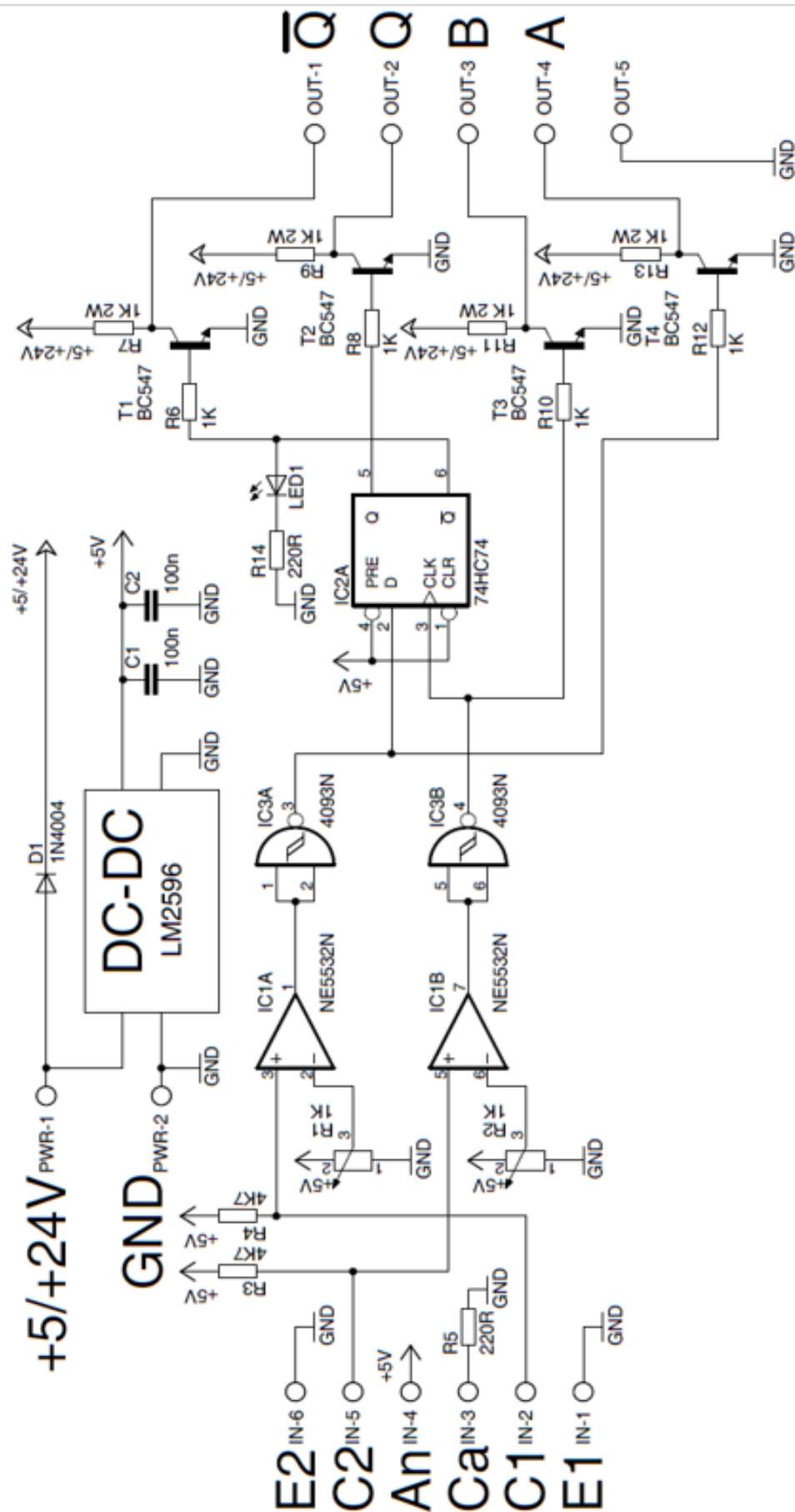
2.2.1. Техничко решење прилагодне електронике за једносмерне моторе

На слици 5 је дата шема кола прилагодне електронике. На енкодер доводимо напајање од 5V, и отпорник од 220 ома како би заштитили диоду. Излази са енкодера са доводе на потенциометре, а они су у суштини подешавајући отпорници. Сигнал са њих иде на операциони појачавач, који појачава разлику између два улазна сигнала. Ова два дела заједно чине компаратор, који нам синусни сигнал претвара у правоугаони. Он у суштини проверава да ли је напон већи или мањи од средње вредности коју добија и избацује максимум или минимум. На шеми се на плус доводи сигнал са енкодера, а на минус се доводи референтна вредност. Подешавањем потенциометра ми доводимо средњу вредност на средину амплитуде синусоиде. Након изласка са компаратора, сигнал водимо на Шмитов компаратор. Он ради на сличан начин као и обичан, са том разликом што има две границе уместо једне. Да би сигнал постао јединица, напон мора да буде већи од горње границе, а да би пао на нулу, мора да буде мањи од доње. Циљ овог поступка је елиминација шума. Код несавршене синусоиде, можемо имати брзо

померање горе-доле малих амплитуда, па би у кратком року имали брзо померање горе-доле на излазу пре него што би се стабилизовало на једном од положаја, сваки пут када би мењали положај. Шмит компаратор елиминише то понашање. Такође спушта вредност напона доње вредности на нулу [2].

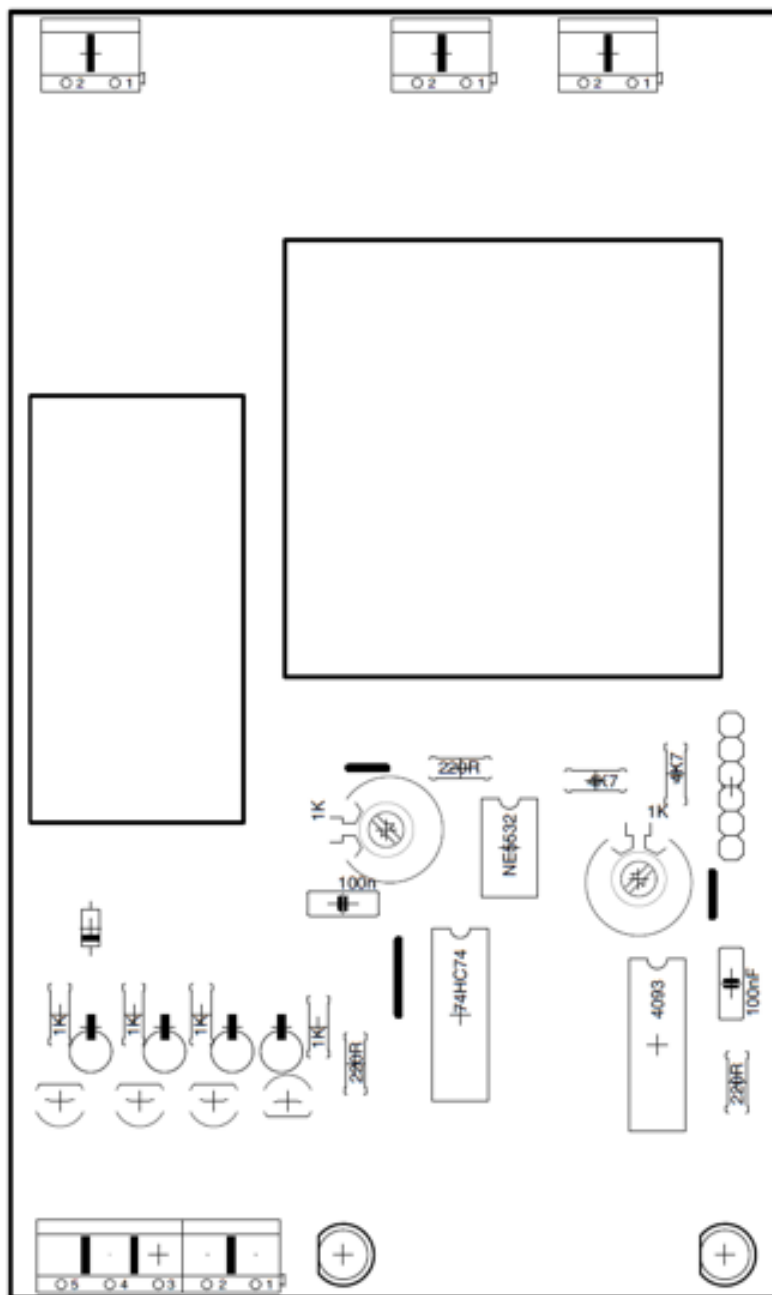
Сигнали се даље води на D флип-флоп. Принцип његовог рада је прост. Вредност доведена на улаз D се уписује у њега када на улаз CLK (Clock). На излаз се доводи уписана вредност, а не обрнута вредност. Вредности које он чува су бинарне. Излаз A је везан за улаз, а B је везан за CLK. Ако је A испред B због смера, прво се на улаз доводи јединица, а након тога и јединица на CLK па се та вредност уписује. Када A падне на нулу, и B пада. Када поново порасте на један, и B ће, па ћемо увек имати уписану јединицу ако се мотор врти у том смеру. У обрнутом смеру ће B ићи прво, па ће уписивати нулу са A. Када A порасте један неће бити уписано, јер је B већ на јединици. Сада добијемо да је излаз нула [2].

Излазе са енкодера и излазе са флип-флопа сада водимо на четири транзистора који повећавају њихов напон у зависности од улазног напона. Ако је напон 5V, крајњи излаз ће бити 5V, а ако је улаз 24V, крајњи ће бити 24V [2].

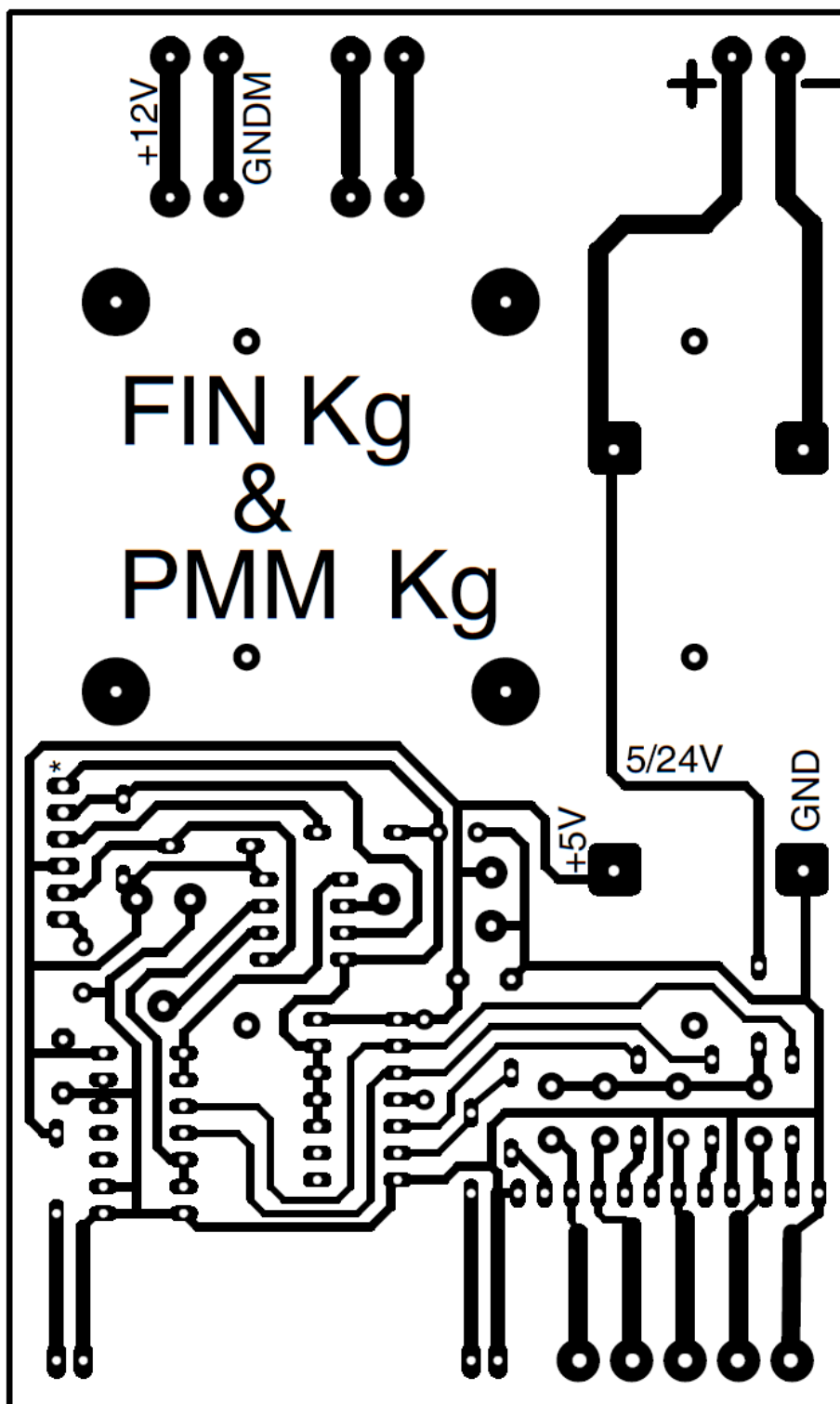


Слика 5. Шема прилагодне електронике [2]

Цела електроника, сем крајњих излаза ради на 5V, тако да морамо обезбедити потребно напајање без обзира на улазно. Томе служи DC-DC конвертер, који увек даје напон који смо му ми одредили без обзира на улаз. Излазни напон одређујемо преко потенциометра који се налази на њему. Отпорници везани у колу служе да заштите остале делове, а два кондензатора на улазу исправљају струју ради бољег рада. На сликама 6 и 7 су дати распореди компонената на површини и матрица за штампање плочице [2].

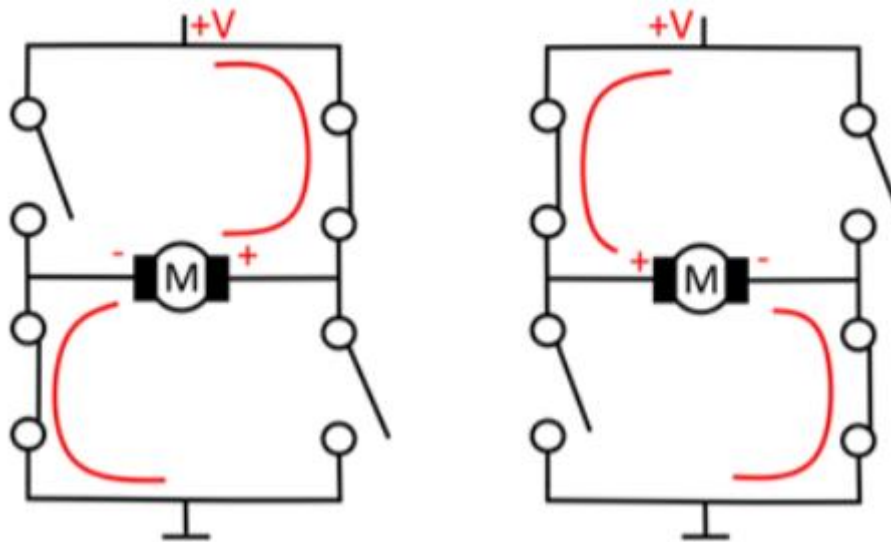


Слика 6. Распоред компоненти на горњој површини плочице [2]



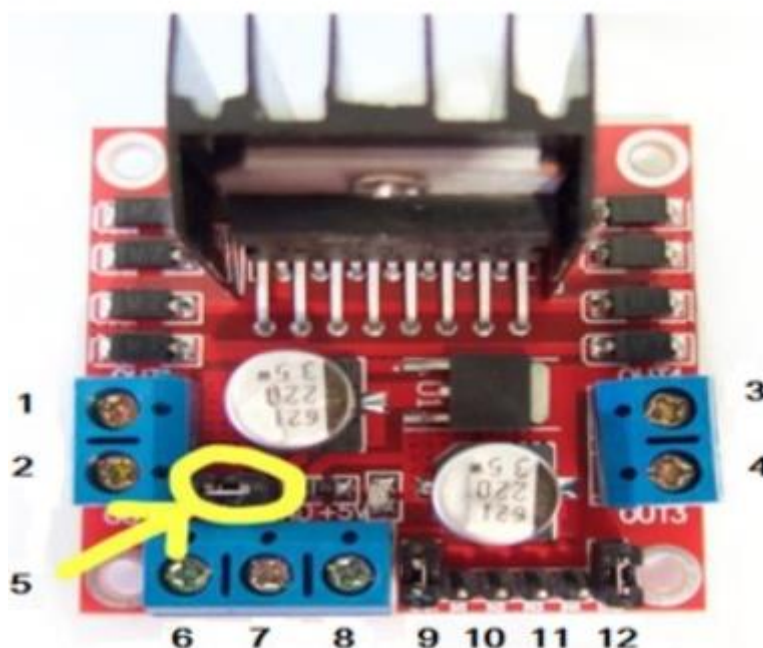
Слика 7. Матрица за штампање [2]

За управљање мотора потребан нам је Н мост уз помоћ којег можемо окретати смер мотора. На слици 8 је дат његов приказ. На једној дијагонали су прекидачи увек затворени, а на другој отворени. Избором дијагонале бирамо смер окретања мотора.



Слика 8. Н мост [2]

На слици 9 у нашем случају ми смер бирамо довођењем напона на један од пинова под бројем 10, а брзину PWM-ом. Н мост напајамо са 12V преко улаза 6 и 7. Краткоспојником спајамо пинове 5 како би исти напон користила и електроника самог моста, која може да издржи до 12V. Кратим спајањем пинова 9 ми обезбеђујемо да ће на излазе 1 и 2 ка мотору увек бити довођено 12V тако да PWM сигналом можемо да управљамо мотором. Када имамо сигнал, на мотор иде максимални напон, када га немамо, иде нула. Користимо одвојено напајање од остатке електронике, јер мотор зна да повуче струју, па би могао да спали цео склоп у том случају [2].



Слика 9. Електрична шема Н мостом [2]

Табела 1. *Излази Н моста*

1,2	Изводи за први мотор
3,4	Изводи за други мотор
5	Селектор извора напајања
6	Извор напајања за мотор
7	Заједничко уземљење
8	Извор напајања за управљачку логику
9	Сигнал за први мотор
10 (2 извода)	Избор смера првог мотора
11 (2 извода)	Избор смера другог мотора
12	Сигнал за други мотор

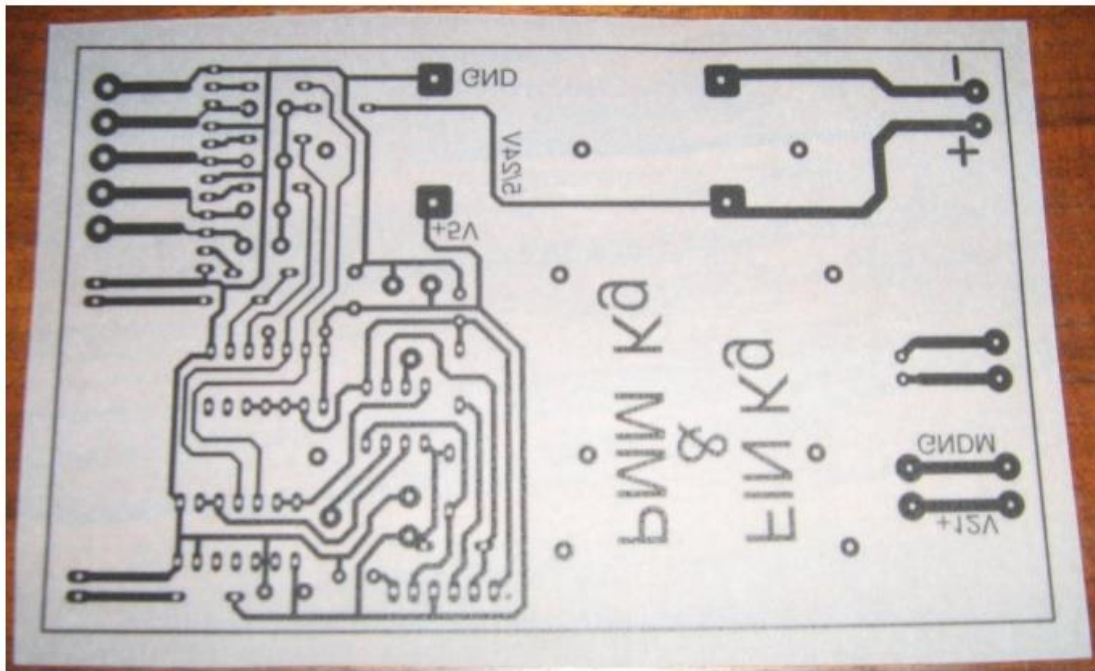
2.2.2. Израда електронике

Сада је потребно направити штампану плочицу. Њу правимо фото поступком. Прво сечемо једнострано каширани витропласт FR4, дебљине 1,55 милиметара и дебљине бакра 0,33 микрометра, на величину мало већу од наше плочице, која име димензије 13 пут 8 центиметара. Матрица коју ћемо да штампамо је приказана раније, а одрађена је у софтверу Eagle V5. Плочицу је потребно добро одмастити како би се отклониле нечистоће и масноће на њеној површини које би могле да сметају при изради [2].



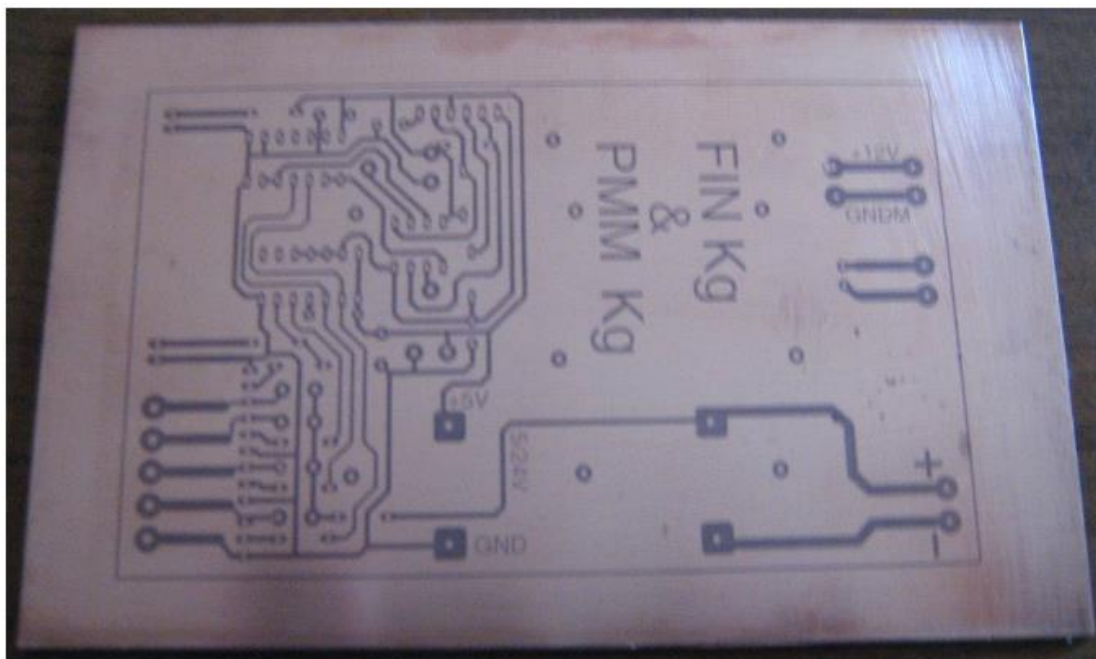
Слика 10. Очишћена плочица

Бакарну површину прекривамо слојем фотосензитивног спреја POSITIV 20, и затим печемо плочицу на 70° 20 минута, да би се слој спреја стегао. Затим на плочицу стављамо нашу матрицу одштампану на паус папиру, и стављамо под ултраљубичасто светло 20 минута. Овако осветљавамо слој спреја на свим местима сем под матрицом [2].



Слика 11. Одштампана матрица [2]

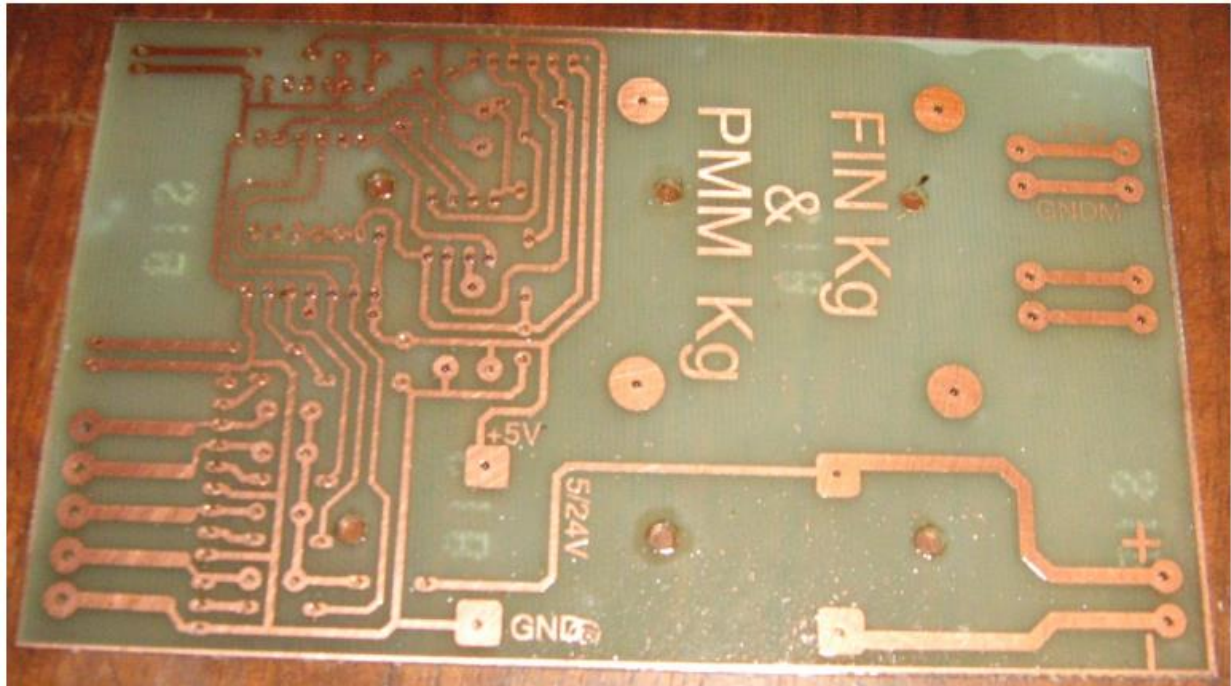
Након експозиције, отклањамо осветљени слој спреја користећи развијач - раствор од 7 промила натријум-хидроксида и воде [2].



Слика 12. Пластика пред нагризање бакра [2]

Након скидања боје, плочицу стављамо у растварач која нагриза непокривени бакар и уклања га. Киселина је раствор 15% хидроксида и 18% хлороводоничне киселине у односу 1 према 3. Након уклањања бакра остаје да се нитро разређивачем отклони остатак неосветљеног слоја спреја. Сада бушимо рупе на плочици бургијом од 1,1 милиметар, и

прекривамо бакарну страну калофонијумом раствореним у алкохолу који штити од корозије [2].



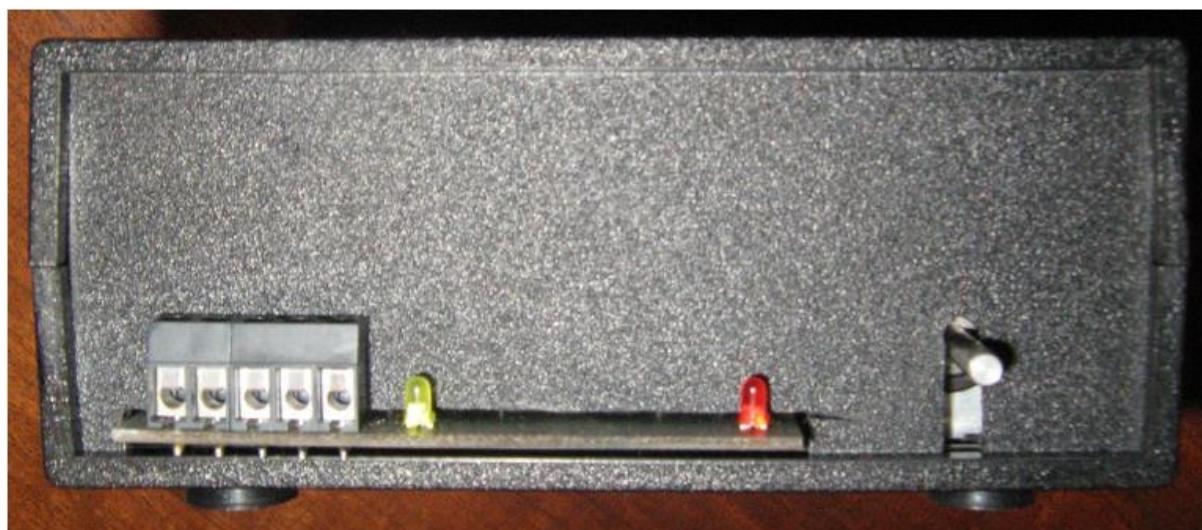
Слика 13. Готова штампана плочица [2]

Сада остаје да се на плочицу напакују компоненте и залеме према већ одрађеном техничком решењу. Онда подешавамо потенциометре тако да добијамо сигнал. Плочице и мотори морају бити упарени, јер постоје мале разлике између енкодера мотора, тако да лепо подешен компаратор код једног мотора не мора да ради и код другог. На крају их све заједно стављамо у једну кутију са излазима како би могли да повежемо мотор са остатком управљачког система [2].

На предњој страни кутије имамо пет излаза - маса, излаз А енкодера, излаз Б енкодера, смер окретања мотора, инверзна вредност смера окретања мотора. Позади имамо по три улаза. На први се доводи напајање од 12V за напајање Н моста и мотора, на други се доводе излази за PWM са Ардуина-а, и на задња два напајање електронике. На последњи улаз се доводи напајање, које може бити 5V или 24V у зависности од управљачког уређаја [2]. У овом раду се доводи 5V са Ардуина.

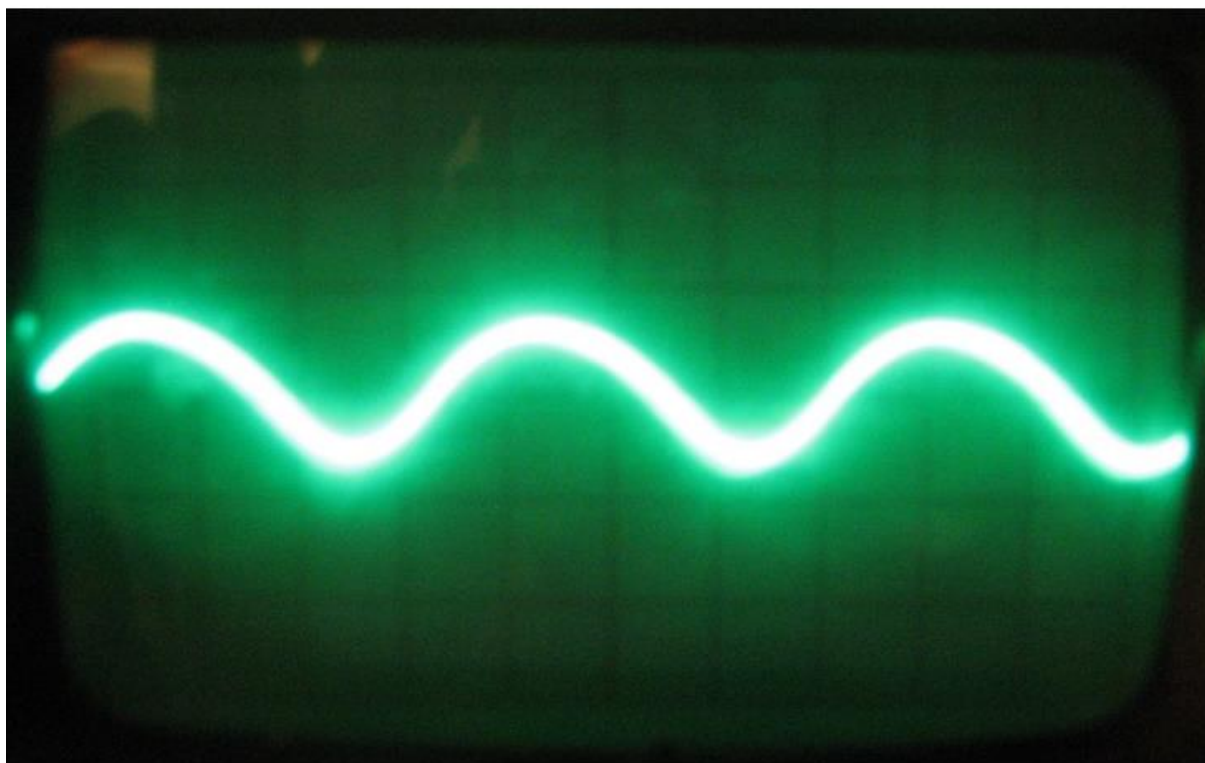


Слика 14. Електроника спакована у кутију [2]

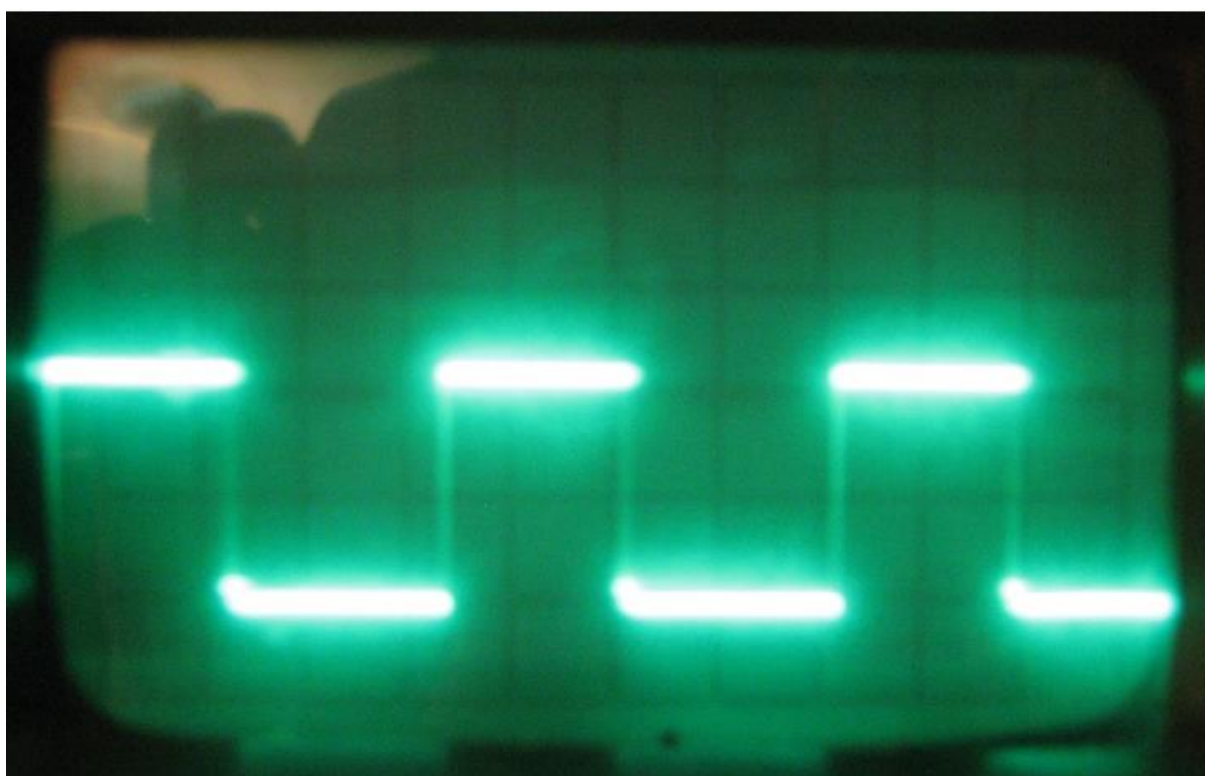


Слика 15. Изглед предњег дела кутије [2]

На сликама 16 и 17 се може видети излазни сигнал одмах са енкодера једносмерног мотора, и након проласка кроз прилагодну електронику [2].



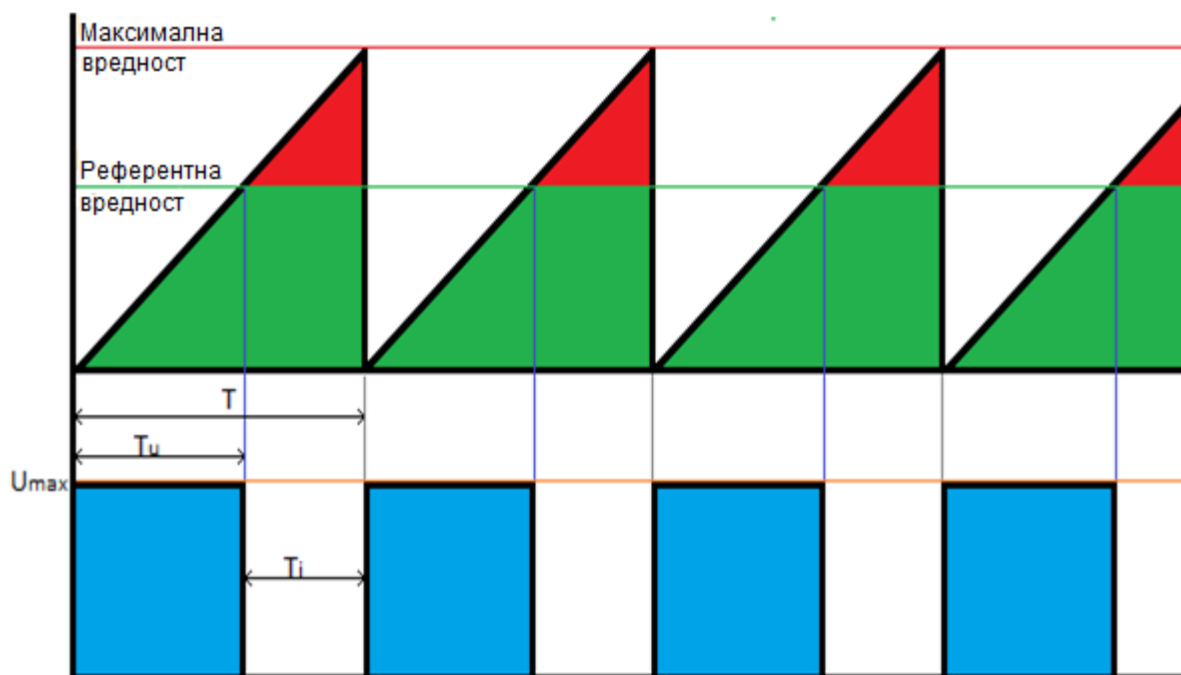
Слика 16. Синусоида са енкодера [2]



Слика 17. Сигнал након прилагодне електронике [2]

2.2.3. Управљачки (PWM) сигнал

Брзина мотора се контролише напонам на улазу мотора. Будући да је једносмерни мотор линеаран систем, брзина на вратилу ће бити сразмерна интензитету напона на улазу мотора. Напон на улазу може бити било које амплитуде између минималног и максималног напона. Будући да се напон може мењати у било ком тренутку и може бити било које амплитуде, произилази да је овакав вид контроле аналогни. Поставља се питање шта ако контролер не поседује дигитално/аналогне претвараче? У том случају, потребно је пронаћи неку другу технику управљања аналогним објектом. Из тога је произашао метод управљања аналогним објектом помоћу дигиталног (бинарног) сигнала варирањем дужине трајања импулса. Овај метод се назива Импулсно ширинска модулација (енгл. *Pulse Width Modulation*), и ради по следећем принципу: излазни сигнал може бити на напону од нула волти (пауза) или на максималном напону (импулс). Однос трајања ове две вредности диктира ефективни аналогни напон на излазу модулятора. Може се приметити да је овај сигнал и даље дигиталне природе, јер постоје само две могуће вредности сигнала, али ако је учестаност довољно висока, аналогни објекти управљања ће због својих филтерских особина 'видети' сигнал као ефективну аналогну вредност. Следеће питање је како се имплементира импулсно ширинска модулација? Најлакше је разумети концепт ако се замисли извор тестерастог таласа. Тестераста талас има својство да се његов интензитет повећава до неке предефинисане максималне вредности, и потом нагло пада на нулу. Ако се овај тестераста талас упоређи са неком произвољном референтном вредношћу, тако да све што је испод те вредности на излазу компаратора даје логичку јединицу, а све што је изнад логичку нулу, добили смо модулартор импулсно ширинске модулације (слика 18) [6].



Слика 18. Приказ PWM сигнала [6]

На слици 18, максимална вредност означава максималну вредност коју тестераста талас може да има, референтна вредност представља вредност за поређење, испод које се на излазу јавља импулс, а изнад које се јавља пауза. $U_{\max}$ представља вредност импулса,

T представља периоду тестерастог таласа, T_U представља дужину трајања импулса, а T_I дужину трајања паузе. Однос времена трајања импулса и паузе се назива испуна (енгл. *Duty Cycle*). Испуна је директно сразмерна ефективном напону. Софтверска имплементација овог модулятора се не разликује много од горе наведене методе. Потребно је направити бројач који у правилном временском интервалу додаје један на своју вредност. Када дође до максималне предефинисане вредности, враћа се на нулу. У истој рутини где се додаје један на вредност, потребно је упоредити тренутну вредност бројача са жељеном вредношћу. Ако је тренутна вредност испод жељене, на излаз поставити логичку јединицу, у супротном поставити логичку нулу. Максимална вредност бројача представља резолуцију модулятора (примера ради, ако је максимална вредност 4, могуће је имати само пет нивоа на излазу, резолуција 20%). Време за које бројач дође до максималне вредности представља периоду [6].

3. АРДУИНО РАЗВОЈНЕ ПЛАТФОРМЕ

Ардуино је физичка рачунарска платформа тзв. „отвореног кода“ (open-source) заснована на једноставној улазно/излазној плочи и развојном окружењу. У овом раду искоришћен је модел „Arduino Uno R3“ који има микроконтролерску плочу базирану на ATmega328P осмобитном микроконтролеру AVR архитектуре. Контролер поседује 14 дигиталних излазно/улазних пинова, од којих 6 могу да се користе као PWM (Pulse Width Modulation) излази, један десетобитни аналогно/дигитални претварач који преко мултиплексера омогућава 6 аналогних улаза, USB конектор за остваривање комуникације са рачунаром, прикључак за напајање, ICSP конектор и тастер за ресетовање. На слици 19 је дат приказ плоче Ардуино Уно са означеним изводима, док је детаљнија спецификација дата табелом 2 [5].



Слика 19. Ардуино Уно развојна плоча [5]

Табела 2 Техничка спецификација Ардуино Уно развојне плоче [5]

Микроконтролер	ATmega328P
Погонски напон	5V
Препоручени улазни напон	7-15 V
Дозвољени улазни напон	6-20 V
Дигитални И/У изводи	14
PWM дигитални И/У пинови	6
Аналогни улазни изводи	6
Једносмерна струја по У/И изводу	20 mA
Једносмерна струја за 3.3V извод	50 mA
Флеш меморија	32 KB, 0,5 KB се користи за bootloader
Статички рам	2 KB
ЕЕПРОМ	1 KB
Радни такт	16 MHz
Димензије	68,8 mm x 53,4 mm

Ардуино развојна плоча се програмира уз помоћ програмског језика C++, који је допуњен библиотекама за олакшан приступ улазима, излазима, тајмерима и прекидним рутинама [5].

4. УТИЦАЈ Н-МОСТА НА КАРАКТЕРИСТИКЕ ОБЈЕКТА УПРАВЉАЊА

У циљу сагледавања особина објекта управљања и утврђивања еквивалентности између управљања континуалним и управљања PWM сигналом извршено је експериментално испитивање и снимање статичких карактеристика мотора. Односно зависности брзине од континуалног улазног напона и зависности брзине од испуњености PWM-а [5].

4.1. Статичка карактеристика напон-брзина

За експериментално утврђивање статичке карактеристике мотора, односно одређивања брзине неоптерећеног мотора у зависности од улазног напона, коришћена је следећа експериментална поставка:

- Задавање улазног напона вршено је лабораторијским напајањем, повезаним директно на улаз мотора.
- Очитавање брзине вршено је очитавањем сигнала са енкодера, уз помоћ прилагодне електронике, ардуино развојне платформе и рачунара

Улазни напон је задаван у опсегу од -12V до 12V са кораком од 1V, узлазно и силазно. Резултати експеримента су приказани у табели 3.

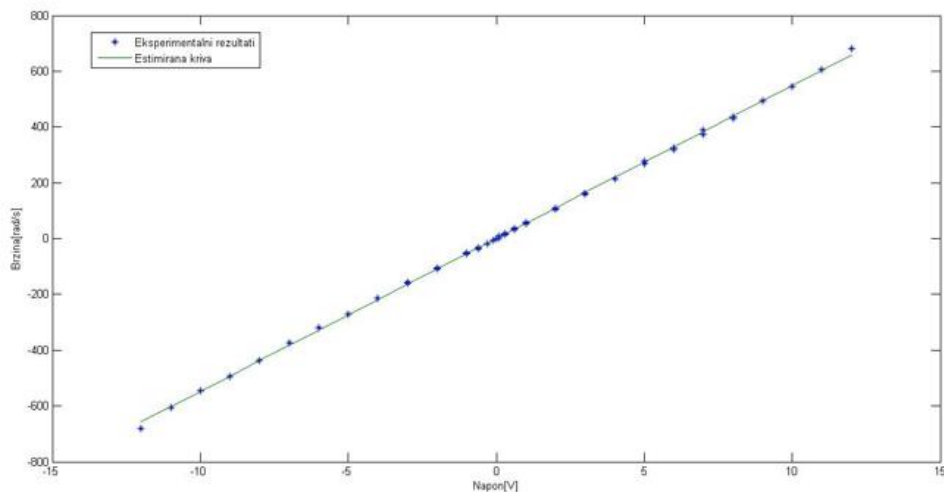
Табела 3. Експерименти резултати брзине у зависности од напона [5]

Напон[V]	Брзина[rad/s]- улазно	Брзина[rad/s]- силазно	Напон[V]	Брзина[rad/s]- улазно	Брзина[rad/s]- силазно
0	0	0	0	0	0
0.1	4.5	7.8	-0.1	-6.9	-7.5
0.3	14.5	18.2	-0.3	-19.3	-18
0.6	34.5	35.7	-0.6	-35	-33
1	54	56	-1	-52	-53
2	108	106	-2	-105	-109.5
3	161	162	-3	-160	-158
4	215	215	-4	-213	-213
5	276	268	-5	-272	-272
6	325	320	-6	-320	-320
7	390	375	-7	-375	-375
8	430	436	-8	-436	-436
9	495	495	-9	-495	-495
10	545	545	-10	-545	-545
11	605	605	-11	-605	-605
12	681	681	-12	-681	-681

Из математичког модела мотора следи да је статичка напонско-брзинска карактеристика мотора линеарна функција. На основу експерименталних резултата из табеле 3, коришћењем методе најмањих квадрата, естимирана је статичка карактеристика, дата једначином 1

$$\omega = 54.79 U + 0.377 \quad (1)$$

На слици 20 је дат приказ естимиране статичке карактеристике и експерименталних резултата [5].



Слика 20. Статичка карактеристика брзина-напон [5]

4.2. Статичка карактеристика PWM-брзина

За експериментално утврђивање брзина у зависности од испуњености PWM-а у стационарном стању, коришћена је следећа експериментална поставка:

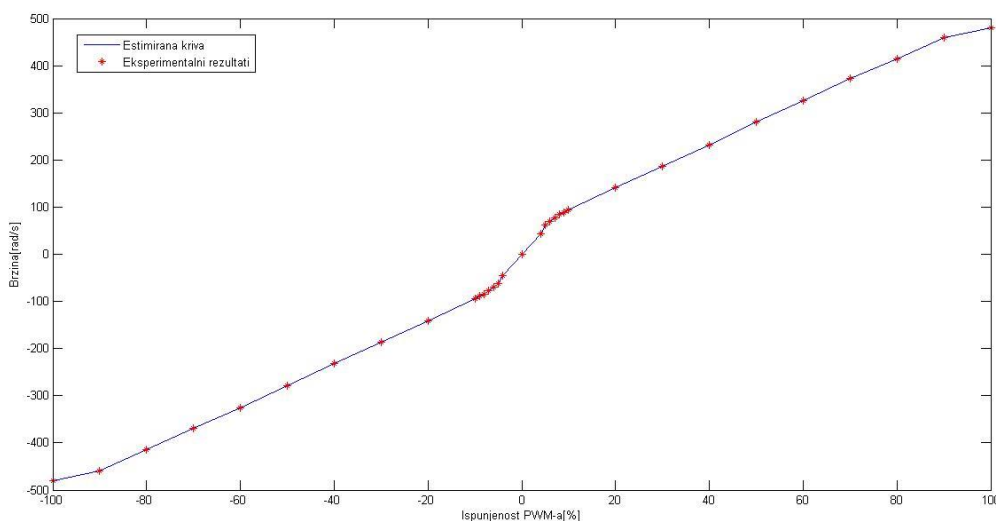
- PWM сигнал је задат коришћењем рачунара и Ардуиноплоче као улаз Н-моста , чије напајање је износило 10.6V
- Очитавање брзине вршено је очитавањем сигнала са енкодера, уз помоћ прилагодне електронике, ардуино развојне платформе и рачунара

Резултати експеримента су дати у табели 4.

Табела 4 Експериментални резултати брзине у зависности од испуњености PWM-а [5]

Испуњеност PWM-а[%]	Брзина[rad/s]	Испуњеност PWM-а[%]	Брзина[rad/s]
0	0	0	0
4	44	-4	-45
5	62	-5	-62
6	70	-6	-70
7	77	-7	-77
8	84	-8	-84
9	89	-9	-89
10	95	-10	-95
20	142	-20	-142
30	186	-30	-186
40	232	-40	-232
50	280	-50	-279
60	326	-60	-325
70	372	-70	-370
80	415	-80	-415
90	460	-90	-460
100	480	-100	-480

На основу експерименталних резултата из табеле 4 добијена је статичка карактеристика приказана на слици 21 [5].



Слика 21. Статичка карактеристика испуњеност PWM-а –брзина [5]

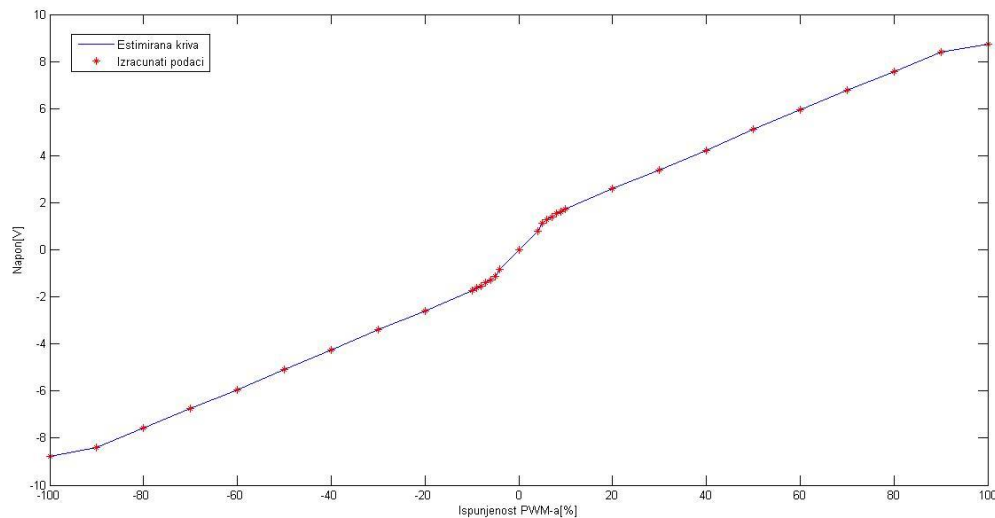
На основу статичке карактеристике PWM-брзина може се закључити да је зависност брзине од испуњености PWM-а линеарна у области рада од 10% до 90% испуњености за оба смера обртања, док се у области 0-10% и 90-100% јавља нелинеарност [5].

4.3. Статичка карактеристика PWM-напон

Коришћењем експерименталних резултата из табеле 4, као и релације 3, можемо добити зависност напона на улазу мотора од испуњености PWM-а. У табели 5 су приказани добијени подаци, док је статичка карактеристика приказана на слици 22 [5].

Табела 5. Израчунати подаци напона у зависности од испуњености PWM-а [5]

Испуњеност PWM-а[%]	Напон[V]	Испуњеност PWM-а[%]	Напон[V]
0	0	0	0
4	0.7962	-4	-0.8282
5	1.1248	-5	-1.1385
6	1.2708	-6	-1.2846
7	1.3968	-7	-1.4123
8	1.5263	-8	-1.5401
9	1.6176	-9	-1.6314
10	1.7271	-10	-1.7409
20	2.5850	-20	-2.5987
30	3.3881	-30	-3.4018
40	4.2277	-40	-4.2415
50	5.1038	-50	-5.0993
60	5.9434	-60	-5.9389
70	6.7830	-70	-6.7603
80	7.5679	-80	-7.5871
90	8.3893	-90	-8.4030
100	8.7543	-100	-8.7681



Слика 22. Статичка карактеристика испуњеност PWM-а -напон [5]

Напон лабораторијског напајања при управљању мотора *PWM* сигналом је износио 10.6 V, а губитак напона при максималној испуни *PWM* сигнала на Н мосту износи 1.85 V. Нелинеарност у доњем опсегу (и мртва зона од 4%) се јавља због немогућности остваривања максималног момента на мотору приликом кратке испуне *PWM* сигнала и Кулоновог трења које се јавља у мотору [5].

Добијени резултати указују на линеарну зависност између управљачког *PWM* сигнала и оствареног континуалног напона на улазу мотора у већем делу радног опсега, што остварује линеарност целокупног управљачког система [5].

5. ПРИНЦИПИ ПРОГРАМИРАЊА КОНТРОЛЕРА И ПРИМЕР ЦЕЛОВИТОГ ПРОГРАМСКОГ РЕШЕЊА ЗА УПРАВЉАЊЕ И НАДЗОР МОТОРА

Контролер лабораторијског модела је персонални рачунар сједињен (повезан) микроконтролером Ардуино, са сврхом да управљачки алгоритам буде програмиран у програмском језику Пајтон (на персоналном рачунару). Персонални рачунар, у сваком тренутку одабирања прослеђује израчунату управљачку променљиву микроконтролеру Ардуину на извршење даље ка актуатору. Такође, у сваком тренутку одабирања, персонални рачунар преузима са микроконтролера вредности угаоне брзине и позиције мотора [5].

У листингу испод, дат је изворни код програма за Ардуино микроконтролер који омогућава комуникацију DC мотор – РС(персонални рачунар). Ардуино комуницира са рачунаром путем USB порта, RS-232 протоколом, брзином од 250 000 бита по секунди. Ардуино прима од рачунара жељену вредност PWM сигнала, и прослеђује је мотору. Рачунару прослеђује тренутну брзину мотора. Период одабирања је једна милисекунда, и могуће је имплементирати ПИД регулатор директно на Ардуино платформи. Тада би рачунар слао само жељену вредност позиције или брзине, коју би мотор требало да оствари. ПИД регулатор је избачен из ове верзије, зарад прављења ПИД регулатора у програмском језику Python, који би се користио у сврху решавања задатка испитивања карактеристика и пројектовања контролера за DC мотор [5].

```
1.  /*
2.   * Definisanje relevantnih promenljivih
3.   * U globalnom domenu.
4.   */
5.  #include <TimerOne.h>
6.  volatile bool uslo = false;
7.  long kT=0;
8.  bool prvo = true;
9.  unsigned long previousMillis = 0;
10. const long interval = 50;
11. //brojac impulsa
12. //prefiks volatile oznacava da je promenljivu moguće promeniti
13. //jer se menja u prekidnoj rutini, te kompajler misli
14. //da ne može da se promeni, jer se ne zove u glavnoj rutini
15. //i tokom faze optimizacije, on je menja u konstantu ako nema
16. //prefiks volatile
17. long maks = 0;
18. volatile long broj = 0;
19. double pozicija = 0;
20. double frek = 0;
21. float pwm = 0;
22. String inputString = "";
23. float inte = 0;
24. float t = 0;
25. float sp = 0;
26. volatile int sp1 = 0;
27. int k = 0;
28. long impulsi;
```

```
29. float fp = 0;
30. long pc = 0;
31. int tB=0;
32. long ucestanost = 0;
33. int smer = 0;
34. int u = 0;
35. int zastavica = 0;
36. float filter[11];
37. void setup(){
38.   Timer1.initialize(50);
39.   Serial.setTimeout(1);
40.   inputString.reserve(200);
41.   //Podesavanje ulaza i izlaza
42.   //Pin 2 i 3 su ulazi
43.   //Sa aktiviranim internim pull-up otpornicima
44.   pinMode(3, INPUT_PULLUP);
45.   pinMode(2, INPUT_PULLUP);
46.   pinMode(8, OUTPUT);
47.   digitalWrite(8, HIGH);
48.   pinMode(9, OUTPUT);
49.   digitalWrite(9, LOW);
50.   pinMode(10, OUTPUT);
51.   digitalWrite(10, LOW);
52.   //Prilikom silaska signala sa logicke jedinice na nulu
53.   //na trecem digitalnom ulazu, poziva se prekidna rutina brojac
54.   attachInterrupt(digitalPinToInterrupt(3), brojac, FALLING);
55.   //Porteanje serijske komunikacije izmedju mikrokontrolera
56.   //i racunara na 250000 bauda.
57.   Serial.begin(250000);
58.   //delay(5000);
59.   Serial.println("0,0");
60.   Timer1.pwm(9,0);
61.   Timer1.pwm(10,0);
62.   previousMillis = millis();
63.   pc = micros();
64. }
65. void brojac(){
66.   uslo = true;
67.   ucestanost = micros() - pc;
68.   pc = micros();
69.   //prekidna rutinarta D i procitati
70.   //vrednost drugog ulaza
71.   //Ako je smer pozitivan
72.   if (PIND & (1<<PD2)){
73.     broj++; // Povecaj brojac za jedan
74.     //Ulaz za smer se ne cita uobicajnim funkcijama
75.     //za Arduino (digitalRead), jer oduzima oko 40 procesorskih
76.     //ciklusa, pristupicemo direktno registru po
77.     smer = 1;
78.   } else {smer = -1; broj--;} // U suprotnom, smanji za jedan
79. }
80. void loop() {
81.   noInterrupts();
82.   impuls = broj;
83.   broj = 0;
84.   interrupts();
85.   frek = (float) smer / (144 * (float) ucestanost / 1000000) * 6.28;
86.   frek = (ucestanost == 0)?0:frek;
87.   if (uslo){
88.     uslo = false;
89.   } else {
90.     frek = 0;
91.   }
92.   pozicija += impuls * 2.5;
93.   previousMillis = millis();
94.   if (maks < 60000){
```

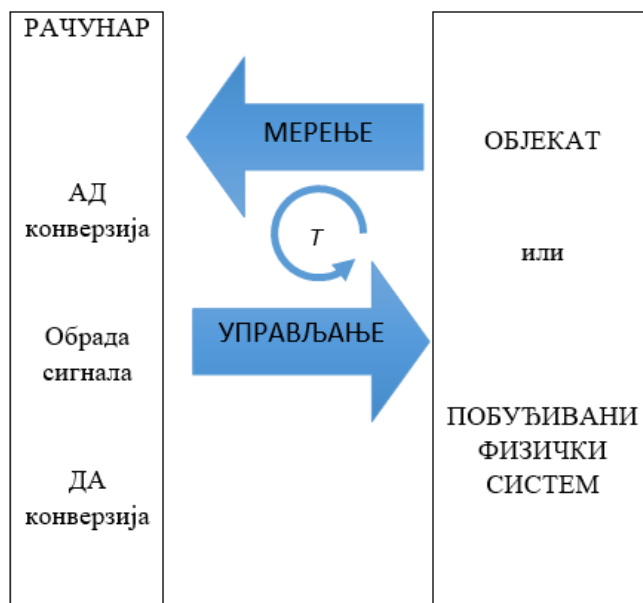
```
95.     if (sp1 > 0) {
96.         Timer1.setPwmDuty(9,0);
97.         Timer1.setPwmDuty(10,abs(sp1));
98.     }
99.     else {
100.         Timer1.setPwmDuty(10,0);
101.         Timer1.setPwmDuty(9,abs(sp1));
102.     }
103. }
104.     else {
105.         Timer1.setPwmDuty(10,0);
106.         Timer1.setPwmDuty(9,0);
107.     }
108.     while (previousMillis + 1 > millis()){
109.         continue;
110.     }
111.     if (maks < 60000) maks ++;
112.     else maks = 60000;
113. }
114.
115. void serialEvent(){
116.     while(Serial.available()){
117.         int kontrol=Serial.parseFloat();
118.         if (kontrol == 255.0){
119.             Serial.print(frek);
120.             Serial.print(',');
121.             Serial.println(pozicija);
122.         }
123.         if (kontrol == 128){
124.             Serial.println(pozicija);
125.         }
126.         else if (kontrol <= 100 && kontrol >= -100){
127.             sp1 = (int)(kontrol * 10.23);
128.         }
129.     }
130. }
```

5.1. Пример целовитог програмског решења контролера на персоналном рачунару који је повезан са микроконтролером Ардуино и написан је у програмском језику Пајтон

Корисник програма има могућност да програмски дефинише:

- Периоду одабирања T
- Време трајања експеримента t_{exp}
- Сигнал $r(t)$ – односно, задату (жељену или референтну) трајекторију система са затвореном повратном спрегом - угаоне брзине или позиције
- Сигнал $M_p(t)$ – односно, емулира дејство поремећаја
- Сигнал $u(t)$ – односно, напише алгоритам управљања
- Снима у фајл променљиве: време, задата вредност, управљачка променљива, угаона брзина обртања вратила мотора, угаона позиција вратила мотора, емулирани момент поремећаја

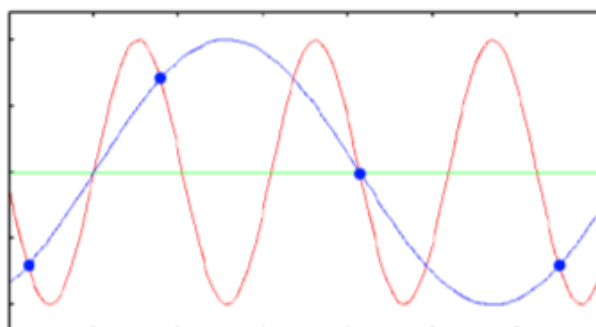
Да би се подсетили о којим величинама и сигнаlima је реч, следе илустративни блок дијаграм слика 23 [5].



Слика 23. Општи принцип функционисања експерименталне инсталације

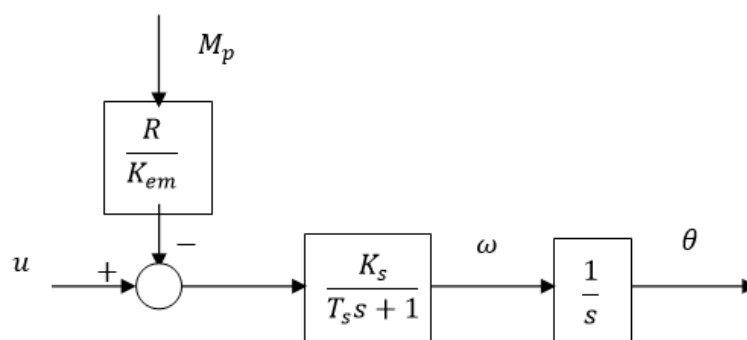
(на сваких T секунди ажурира се вредност сигнала управљања и прикупља нови податак мерења) [5]

Проблем избора адекватне периоде је илустрован на следећој слици 24.

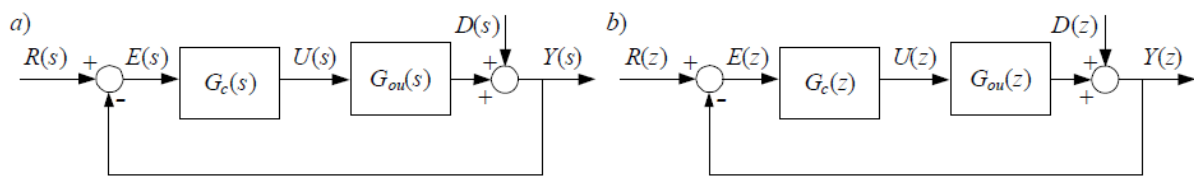


Слика 24. Илустрација аалисинг ефекта услед неадекватно изабране периоде одабирања [5]

Избор и параметризација алгорита управљања је увек прилагођена објекту управљања.



Слика 25. Структурни блок дијаграм ОБЈЕКТА УПРАВЉАЊА [5]



Слика 26. Општи структурни блок дијаграм система са затвореном повратном спрегом [5]

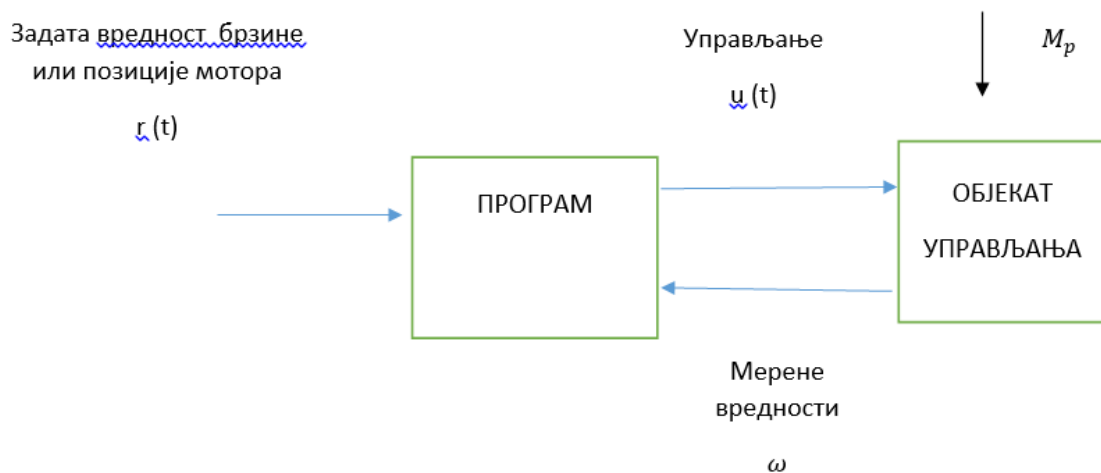
Општи ПИД закон управљања је облика:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s$$

$$u(kT) = K_p e(kT) + K_i \sum_{i=0}^k e(iT) + \frac{K_d}{T} (e(kT) - e[(k-1)T])$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Овде где је програм нацртати персонални рачунар + Ардуино и ставити програм у Пајтону + програм у С-у и илустровати променљиве које се дефинишу у програму [5].



Слика 27. Општи структурни блок дијаграм система са затвореном повратном спрегом [5]

Дефиниција периоде одабирања T , сигнала $r(t)$, $M_p(t)$, $u(t)$, се постиже изменама класе

```
1. #klasa motor
2. class Motor:
3.     #konstruktor klase, zove se pri pokretanju eksperimenta
4.     #sadrzi pocetne konstante
5.     def __init__(self):
6.         self.period = 0.05 #perioda odabiranja, obavezno polje, preporuceno 0.05 s
7.         self.duration = 15 #vreme trajanja eksperimenta u sekundama (ne vise od 60)
8.         #promenljive koje zadaje korisnik
```

```
9.         self.integral = 0
10.        self.kP = 0.5 #proporcionalno dejstvo
11.        self.kI = 3 #integralno dejstvo
12.        #upravljacka logika
13.        #dobija trenutno vreme, ugao (stepeni) i brzinu motora (rad/s)
14.        def loop(self, t, omega, teta):
15.            r = 60 #zadata vrednost 30 rad/s
16.            e = r - omega #greska
17.            self.integral += e * self.period #integral
18.            upravljanje = self.kP * e + self.kI * self.integral
19.            emP = 3 #emuliran poremećaj
20.            return upravljanje - emP #salje upravljanje motoru
```

док је целовито програмско решење дато линијама кода

```
1. #ucitavanje potrebnih biblioteka
2. import time #biblioteka za vreme
3. from skripta import * #ucitavanje programa koji je student napisao
4. import serial #biblioteka za serijsku komunikaciju
5. #Inicijalizacija serijske komunikacije
6. #Arduino se nalazi na portu COM3, brzina je 250 000 bita po sekundi
7. #Maksimalno cekanje podatka je 5 milisekundi
8. ser = serial.Serial('COM3', 250000, timeout=0.005)
9. try:
10.     #inicijalizacija skripte koju je student napisao
11.     motor = Motor()
12.     #ugasi motor
13.     u = 0
14.     #sacekaj dve sekunde da se arduino ukljuci
15.     time.sleep(2)
16.     #ukupno vreme od pocetka eksperimenta
17.     ukupnoVreme = 0
18.     pocetnoVreme = time.clock()
19.     protekloVreme = 0
20.     #perioda odabiranja
21.     periodaKorigovana = motor.period
22.     #kada se arduino ukljuci, salje nulu preko serijskog porta
23.     #ova komanda prima to i eksperiment pocinej
24.     ser.readline()
25.     #eksperiment traje do naznacene vrednosti
26.     while ukupnoVreme < motor.duration:
27.         ser.write('255') #uzimanje vrednosti brzine i pozicije
28.         brz = ser.readline()
29.         brz = brz.rstrip().split(',')
30.         #glavna upravljacka logika
31.         u = motor.loop(ukupnoVreme, float(brz[0]), float(brz[1]))
32.         #maksimalni pwm ne sme preci +/-100%
33.         if (u > 100): u=100
34.         if (u < -100): u = -100
35.         ser.write(str(int(u))) #slanje upravljanja arduinu
36.         #stampanje rezultata
37.         print str(ukupnoVreme) + ', ' + str(motor.r) + ', ' + str(u) + ', ' + str(brz
[0]) + ', ' + str(brz[1]) + str(motor.emP) #slanje podataka
38.         while (time.clock()-pocetnoVreme)-protekloVreme<= periodaKorigovana:
39.             pass
40.             protekloVreme = time.clock()
41.             ukupnoVreme += motor.period
42.         ser.write('0') #iskljucivanje
43.         #zatvaranje serijskog porta
44.         ser.close()
45. except:
46.     ser.close()
```

5.2. Пример програмског решења у примени, извршења експеримента, обраде и презентације резултата експеримента

Следи пример имплементације ПИ регулатора за синтезу брзинског сервомеханизма, са референтном (жељеном) брзином од 60 rad/s, и емулираним поремећајем од 0.3 V који се јавља у петој секунди експеримента. Класа *Motor* дефинише ПИ регулатор и услове експеримента (референтну вредност и поремећај) и дата је у листингу испод [5].

```
1. #klasa motor
2. class Motor:
3.     #konstruktor klase, zove se pri pokretanju eksperimenta
4.     #sadrzi pocetne konstante
5.     def __init__(self):
6.         self.period = 0.05 #perioda odabiranja, obavezno polje, preporuceno 0.05 s
7.         self.duration = 15 #vreme trajanja eksperimenta u sekundama (ne vise od 60)
8.
9.         self.upravljanje = 0
10.        self.emP = 0
11.        self.r = 0
12.        #promenljive koje zadaje korisnik
13.        self.integral = 0 #cuva vrednost integrala
14.        self.kP = 0.5 #proporcionalno dejstvo
15.        self.kI = 3 #integralno dejstvo
16.
17.        #upravljacka logika
18.        #dobija trenutno vreme, ugao (stepeni) i brzinu motora (rad/s)
19.        def loop(self, t, omega, teta):
20.            self.r = 60 #zadata vrednost 60 rad/s
21.            e = self.r - omega # trenutna greska
22.            self.integral += e * self.period #integral greske
23.            self.upravljanje = self.kP * e + self.kI * self.integral #PI zakon upravljan
24.            ja
25.        if (t > 5): #ako je proteklo pet sekundi od pocetka eksperimenta
26.            self.emP = 3 #emuliran poremećaj
27.        return self.upravljanje - self.emP #salje upravljanje motoru
```

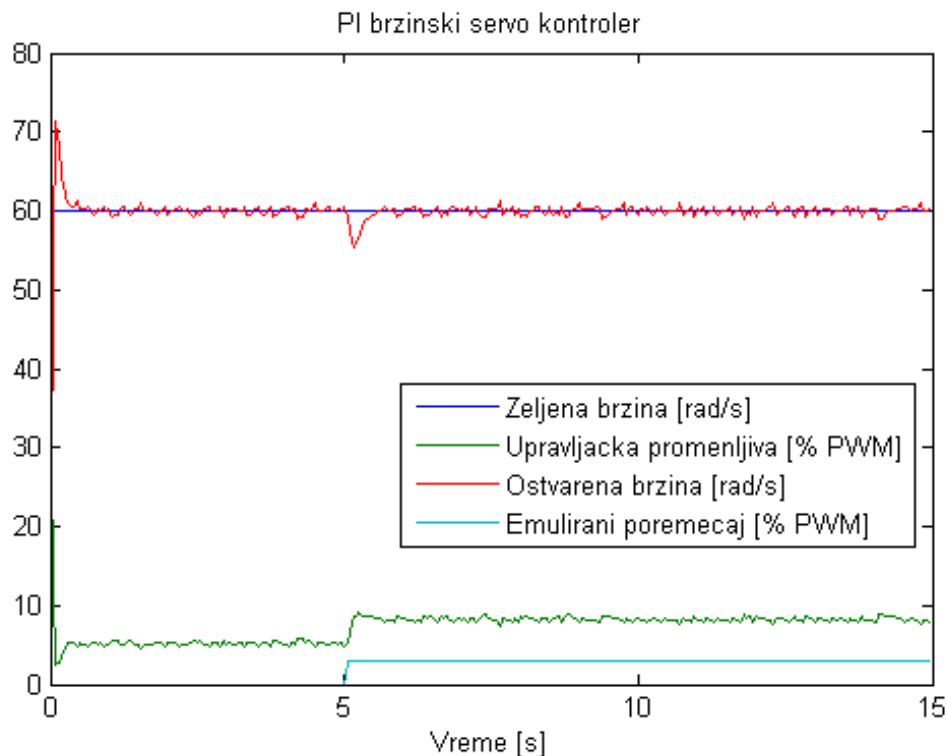
По завршетку експеримента, преузету датотеку је могуће отворити у било ком програму који подржава листе одвојене зарезом (*comma separated*). Следи пример обраде експерименталних података помоћу програмског пакета *MATLAB*. Командом *a = load('experiment.m');* добијени резултати се учитавају у променљиву *a*.

Сада је могуће иштампати добијене резултате на дијаграм коришћењем команде:

```
plot(a(:,1),a(:,2),a(:,1),a(:,3),a(:,1),a(:,4),a(:,1),a(:,6));
legend('Zeljena brzina [rad/s]', 'Upravljacka promenljiva [% PWM]', 'Ostvarena brzina [rad/s]', 'Emulirani poremećaj [% PWM]');
xlabel('Vreme [s]);
```

```
title('PI brzinski servo kontroler');
```

Резултат је слика 28 [5].



Слика 28. Експериментални резултати верификације брзинског сервомеханизма са ПИ регулатором [5]

Следећи пример се односи на позициони сервомеханизам са П регулатором. Задата жељена вредност износи 90 степени, емулирани ефекат поремећаја на управљачку променљиву је $0.5 V * h(t-5)$. Програмско решење је дато у листингу испод [5].

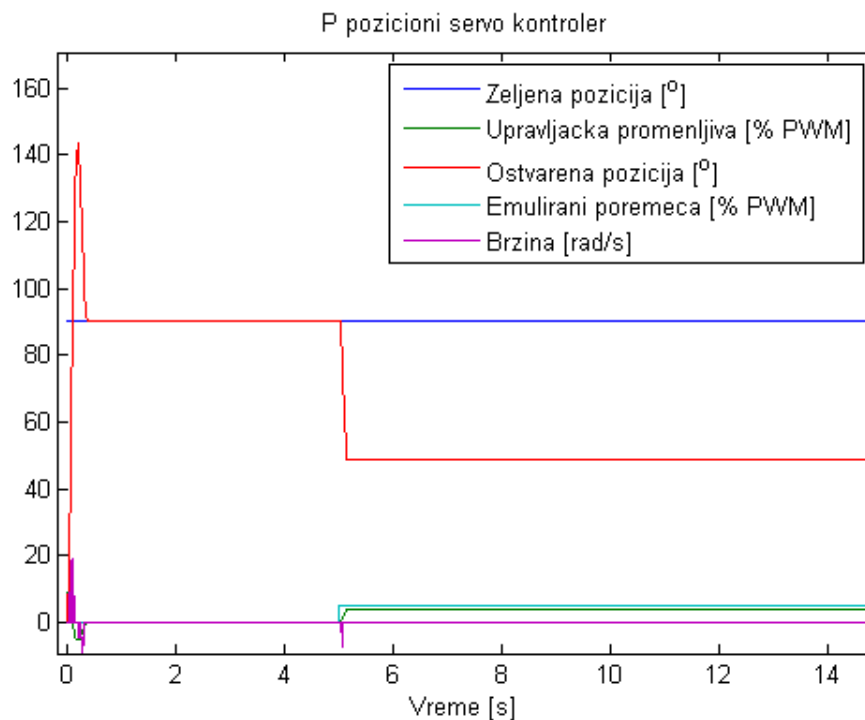
```

1. #klasa motor
2. class Motor:
3.     #konstruktor klase, zove se pri pokretanju eksperimenta
4.     #sadrzi pocetne konstante
5.     def __init__(self):
6.         self.period = 0.01 #perioda odabiranja, obavezno polje, preporuceno 0.05 s
7.         self.duration = 5 #vreme trajanja eksperimenta u sekundama (ne vise od 60)
8.         self.upravljanje = 0
9.         self.emP = 0
10.        self.r = 0
11.        #promenljive koje zadaje korisnik
12.        self.kP = 0.09 #proporcionalno dejstvo
13.        #upravljacka logika
14.        #dobija trenutno vreme, ugao (stepeni) i brzinu motora (rad/s)
15.        def loop(self, t, omega, teta):
16.            if (self.integral)>600:self.integral=600
17.            if (self.integral)<-600:self.integral=-600
18.            self.r = 90 #zadata vrednost 90 stepeni
19.            e = self.r - teta #greska
20.            self.upravljanje = self.kP * e
21.            if (t > 5):
22.                self.emP = 5 #emuliran poremećaj
23.            return self.upravljanje - self.emP #salje upravljanje motoru

```

Учитавањем података у *MATLAB*, и малом изменом предходног *MATLAB*кода (потребно је да се прикаже и позиција), могуће је добити дијаграм налик на слици 28. *MATLAB* код за штампање релевантних података позиционог сервомеханизма је:

```
b=load('experiment.m');  
plot(b(:,1),b(:,2),b(:,1),b(:,3),b(:,1),b(:,5),b(:,1),b(:,6),b(:,1),b(:,4));  
legend('Zeljena pozicija [°]', 'Upravljacka promenljiva [% PWM]', 'Ostvarena pozicija [°]',  
'Emulirani poremeća [% PWM]', 'Brzina [rad/s]');  
xlabel('Vreme [s]');  
title('P pozicioni servo kontroler');
```



Слика 29. Експериментални резултати верификације позиционог сервомеханизма са П регулатором [5]

6. ЗАКЉУЧАК

У овом раду су објашњени основни принципи рада ПИД управљања, Ардуино контролера. Изложене су све компоненте које чине систем и објашњава принципе рада енкодера, једносмерних мотора и управљачког (PWM) сигнала. За потребе контроле једносмерног мотора развијена је електроника која нам претвара синусни сигнал енкодера у правоугаони сигнал који управљачки уређај може да користи. Направљене су штампане плоче и цео систем је упакован у једну кутију, заједно са једносмерним мотором.

Даље је у раду објашњен утицај Н-моста на карактеристике објекта управљања. Објашњене и приказане су кроз графике и табеле статичке карактеристике:

- Напон-Брзина
- PWM-Брзина
- PWM-Напон

На крају рада објашњен је принцип програмирања контролера и пример целовитог програмског решења за управљање и надзор мотора. Одрађен је пример целовитог програмског решења контролера на персоналном рачунару који је повезан са микроконтролером Ардуино и написан је у програмском језику Пајтон, и објашњена је обрада и презентација резултата експеримента.

Цео овај систем представља основу на коју је касније могуће надоградити и развити системе. На пример, могуће је додати још један једносмерни мотор у циљу израде роботског аутомобила, а када би се додали различити сензори могуће је направити роботски ауто за праћење линије. За израду оваквог пројекта потребно је све делове повезати и физички направити саму структуру робота као и све то повезати на ардуион развојну платформу, и на крају потребно је извршити програмирање самог робота како би он могао да изврши задатак праћења линије. Додавањем разних сензора могуће је да се направе разне друге врсте робота за разне намене. Идеја коју сам навео као и предлог у вези додавања сензора и израде других врста робота за друге намене преставља путоказ за даљи развој овог система.

7. ЛИТЕРАТУРА

- [1] Матијевић, М., Јакуповић, Г., Цар, Ј.: Рачунарски подржано мерење и управљање, Машински факултет у Крагујевцу, друго издање, Крагујевац, Септембар, 2008. – Преузето 10.08.2020. год.
- [2] Андрија С. Јестровић, „Пројектовање SCADA апликације и система заснованог на раду четири једносмерна и асинхрона мотора“, Дипломски - мастер рад, Факултет инжењерских наука Универзитета у Крагујевцу, Крагујевац, 2017. – Преузето 10.08.2020.год.
- [3] Automatika - DC Motori, Интернет страница:<https://www.automatika.rs/baza-znanja/mehatronika/dc-elektromotori.html>, - Преузето 11.08.2020.год.
- [4] Machine Design- basic of Rotary Encodes: Overview and New Technologies, Интернет страница:<http://www.machinedesign.com/sensors/basics-rotary-encoders-overview-and-new-technologies-0>, - Преузето 11.08.2020.год.
- [5] Рачунарско управљање мотора једносмерне струје Escar 28D2R-219P – ОПИС ЛАБОРАТОРИЈСКОГ МОДЕЛА И ЊЕГОВЕ УПОТРЕБЕ Интернет страница: moodle.mfkg.rs/course/view.php?id=228, - приступљено 16.08.2020.год.
- [6] Д. Милошевић, „Пројектовање дигиталних система за управљање позиције и брзине DC мотора помоћу S7 1200 PLC-а и микроконтролера Atmega328P“, Дипломски - мастер рад, Факултет инжењерских наука Универзитета у Крагујевцу, Крагујевац, 2016. – приступљено – 16.08.2020.год.

Прилог А

Табела 6. Спецификације једносмерног мотора[2]

Величине	Вредност	Јединице
Максимални напонпобуде мотора ($-12\text{ V} \leq u(t) \leq 12\text{ V}$)	12	V
Максимална брзина неоптерећеног мотора	5800	o/min
Момент заочења (момент оптерећења под којим мотор не може да оствари било какву брзину)	94	mNm
Јачина струје неоптерећеног мотора (јачина струје у ротору при максималној брзини обртања неоптерећеног мотора)	44	mA
Стартни напон (минимални напон који је потребно довести да би се мотор покренуо – да би савладао Кулоново трење – величина мртве зоне)	0.15	V
Максимална континуална струја (максимално радно оптерећење струје ротора је 1.5 A)	1.5	A
Максимални континуални момент (максимално радно оптерећење вратила мотора према статичкој карактеристици произвођача, Сл.2.1)	28.4	mNm
Максимално угаоно убрзање	48	10^3 rad/s^2
Електрична константа ($K_{me} = 0.0196\text{ V}/(\text{rad/s}) = 0.0196\text{ Vs}$)	2.05	$\text{V}/(1000\text{ o/min})$
Механичка константа ($K_{em} = 0.0195\text{ Nm/A}$)	19.5	mNm/A
Отпорност арматуре ($R = 2.5\ \Omega$)	2.5	Ω
Индуктивност ротора ($L = 0.0003\text{ H}$)	0.30	mH
Момент инерције ротора ($J = 17.60 \cdot 10^{-7} \text{ kgm}^2$)	17.60	10^{-7} kgm^2
Механичка временска константа ($T_s = 0.012\text{ s}$)	12	ms
Коефицијент вискозног трења ($\beta = 1.41264797\text{e-6 Nm}/(\text{rad/s})$)	1.41264797e-6	Nm/(rad/s)
Број импулса по обртају енкодера Резолуција инкременталног енкодера је дакле 2.5 степена. Ради се о два сигнала која генеришу 144 импулса по обртају, а фазно су померена за 90 степени ради утврђивања смера обртања.	144	
Напон напајања енкодера	5	V
Пин +/- (прикључци за напајање мотора, и истовремено за управљање мотора напонским сигналом)	12VDC Motor	
Пин 1(прикључак за емитер транзистора 1 унутар енкодера)	Емитер 1	
Пин 2(прикључак за колектор транзистора 1 унутар енкодера)	Колектор 1	
Пин 3 (прикључак за катоду ЛЕ диоде унутар енкодера)	Катода	
Пин 4 (прикључак за аноду ЛЕ диоде унутар енкодера)	Анода	
Пин 5(прикључак за емитер транзистора 2 унутар енкодера)	Колектор 2	
Пин 6(прикључак за колектор транзистора 2 унутар енкодера)	Емитер 2	