

**Факултет инжењерских наука
Универзитета у Крагујевцу**



Назив студијског програма: Рачунарска техника и софтверско инжењерство

Ниво студија: Основне академске студије

Предмет: Основи електротехнике

Број индекса: 569/2015

Бобан Г. Срећковић
Дигитално управљање брзином и
позицијом серво мотора
дипломски рад

Комисија за преглед и одбрану:

1. **др Јасна Радуловић, ред. проф.**

2. _____

3. _____

Датум одбране: _____

Оцена: _____

У оквиру овог дипломског рада кандидат треба да истражи и проучи литературу на задату тему и на бази тога да детаљно и конкретно опише цео процес анализе и одабира електронских компоненти посматраног система. Потребно је да кандидат дефинише електро шеме целокупног система као и техничке спецификације одабраних компоненти, и уједно да пројектује и изради софтвер дигиталног управљања. Током израде система потребно је извршити подешавање разних параметара серво мотора као што су брзина и позиција помоћу поворке дигиталних импулса.

Препоручена литература:

- [1] Слободан Н. Вукосавић: Дигитално управљање електричним погонима, Академска мисао, Београд 2003.
- [2] Ендру С. Таненбаум: Архитектура и организација рачунара, Микро књига, Београд 2007.
- [3] Stuart R. Ball: Embedded Microprocessor Systems: Real World Design (Embedded Technology), Elsevier Inc., Oxford 2002.
- [4] Милић Р. Стојић: Дигитални управљачки системи, Академска мисао, Београд 2004.

Крагујевац, 16.05.2019. године

Ментор:
др Јасна Радуловић, ред. проф.

Садржај:

1. Увод	7
2. Теоријска разматрања	8
2.1 Електромотор	8
2.2 Енкодер	11
2.3 Микроконтролер	13
3. Практични део рада	15
3.1 Пројектовање и развој система	15
3.1.1 Анализа и дефинисање компоненти система	15
3.1.2 Блок дијаграм и терминали драјвера	20
3.2 Израда софтвера дигиталног управљања	22
3.2.1 Развојно окружење и програмски језик	22
3.2.2 Подешавање и одабир брзина	22
3.2.3 Подешавање и одабир позиције	26
3.2.4 Контрола брзине интерним подешавањем	28
3.2.5 Контрола позиције инкременталним енкодером	33
4. Закључак	36
Литература	37

Резиме:

У овом раду обрађена је тема дигиталног управљања серво мотора као и његова примена – практична употреба. Током анализе целокупног система и његових потреба, дошло се до закључка да серво мотор ознаке *MS-80ST* компаније *Xinje (Xinje Electronic)* задовољава наше потребе пошто је овај тип мотора предвиђен управо за примену у аутоматским управљачким системима који захтевају високе динамичке перформансе и прецизност. Како се врши дигитално управљање потребно је било одабрати и одговарајући драјвер (*driver*) који у потпуности пружа све неопходне карактеристике за развој једног оваквог система. Микроконтролер који је задужен за управљање наведеног система је *STM32F107VCT6* док се за израду софтвера управљања користи програмски језик *mikroC* као и развојно окружење *mikroC PRO for ARM* компаније *MikroElektronika*.

Кључне речи: Серво мотор, MS-80ST, Драјвер, Микроконтролер, ARM, mikroC.

Summary:

This paper deals with the topic of digital control of the servo motor and its application – practical use. During the analysis of the whole system and its needs, it was concluded that the *MS-80ST* servo motor produced by company *Xinje (Xinje Electronic)* meets our needs since this type of motor is intended for use in automatic control systems that require high dynamic performance and precision. In order to perform digital control, it was necessary to choose the appropriate driver that fully provides all the necessary characteristics for the development of such a system. The microcontroller in charge of managing the said system is *STM32F107VCT6* while programming language *mikroC* is used as well as the development environment *mikroC PRO for ARM*, products of company *MikroElektronika*.

Keywords: Servo motor, MS-80ST, Driver, Microcontroller, ARM, mikroC

1. Увод

Серво мотор (од лат. *servus*: слуга, роб; *motor*: покретач) је електромотор који према примљеном управљачком сигналу заузима одређени угаони положај или позицију на некој путањи, односно развија одговарајући обртни момент или силу. Покретањем и заустављањем серво мотора управља се управљачким сигналом мање снаге (електрични, механички, пнеуматски, хидраулички) који се потом претвара у одговарајуће помераје, најчешће са већом извршном механичком снагом – која се остварује помоћу редуктора. Углавном сваки серво мотор поседује и мерни инструмент (давач положаја или енкодер) који утврђује тачан положај (помак - угао) извршног дела али и који измерени положај по потреби и поправља (карактеристично за системе са повратном спрегом). [1]

Дигитално управљање система са серво моторима се остварује коришћењем микроконтролера. Његова главна улога је управљање уређајима и процесима, па у себи има интегрисан микропроцесор, меморију, дигитално - аналогне улазе и излазе, дигиталне сатове (тајмере), бројаче, осцилаторе, комуникационе интерфејсе и друге додатке у зависности од његове намене. [2]

Принцип рада самог управљања се заснива на томе да микроконтролер шаље поворку дигиталних импулса драјверу (управљачу) серво мотора, драјвер читава примљене импулсе и на основу њих управља серво мотором – промена брзине, положаја или смера окретања осовине мотора.

Серво мотори се користе самостално или као делови сервомеханизма. Најчешћа примена је у роботизи где се користе за одржавање позиције екстремитета, код модела авиона за померање крилаца, док се у аутомобилској индустрији може пронаћи унутар система за контролу скретања преко волана.

2. Теоријска разматрања

2.1 Електромотор

Електромотор је електрична машина која претвара електричну енергију у механичку енергију. Већина електро мотора раде на принципу електромагнетне индукције, али постоје и мотори који користе друге електромеханичке феномене, као што су електростатичка сила и пиезоелектрични ефекат. Фундаментални принцип на ком се заснивају електромагнетски мотори је механичка сила која делује на проводник кроз који протиче електрична струја и који се налази у магнетском пољу. Ова сила је описана Лоренцовим законом и њен правац је нормалан на проводник и магнетско поље. [3]

Већина електромагнетних мотора је ротационог типа, а постоје и линеарни мотори. У ротационом мотору, обртни део се назива ротор, а непомични се назива статор. Намотаји на ротору су постављени аксијално, а магнетско поље је радијално. Стога механичка сила делује тангентијално, па се развија обртни момент на осовини ротора.



Слика 1 – Делови електро мотора [4]

Електро мотори који раде на наизменичној струји могу бити монофазни или трофазни.

Синхрони мотор је машина која се окреће врло тачном брзином у складу са фреквенцијом напајања - електричне мреже или свог посебног (синхроно са мрежом).

Пример: Машина са 2 пола (1 пар полова) ће се окретати брзином од 50 r/s (ротација по секунди) односно 3000 o/min (обртаја по минути; код мреже од 50 Hz – као што је на пример градска мрежа), са четири пола 25 r/s односно 1500 o/min итд.

Асинхрони мотор је највише кориштена наизменична машина. Једноставне конструкције, јефтин и поуздан у раду, користи се свуда у индустрији за најразличитије задатке. Његова брзина ротације је донекле променљива зависно од оптерећења. На пример, асинхрони мотор са 4 пола (2 пара полова) са фреквенцијом мреже од 50 Hz ће имати радну брзину око 23-24 r/s, или око 1440 обртаја у минути.

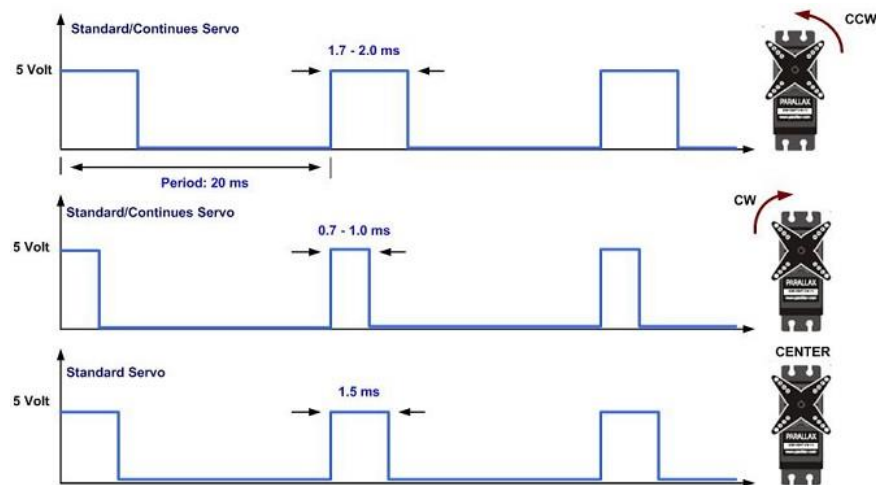
Серво мотори су такође једна врста електромотора за које се може рећи да су ротациони актуатори који се користе у процесима где је потребно управљање угаоном позицијом, брзином и убрзањем. [5]

Састоје се од одговарајућег мотора у комбинацији са сензором који даје информацију о позицији (потенциометар, енкодер), док такође могу садржати и релативно софистициране контролере (драјвере) који су углавном одвојен модул намењен за употребу са серво моторима.

Серво мотор поседује три прикључка, један управљачки и два преко којих се мотору доводи напајање. Управљачки сигнал се користи за управљање угаоном позицијом излазне осовине и најчешће је то *PWM* (*Pulse-Width Modulation* – Импулсно-Ширинска Модулација) сигнал.

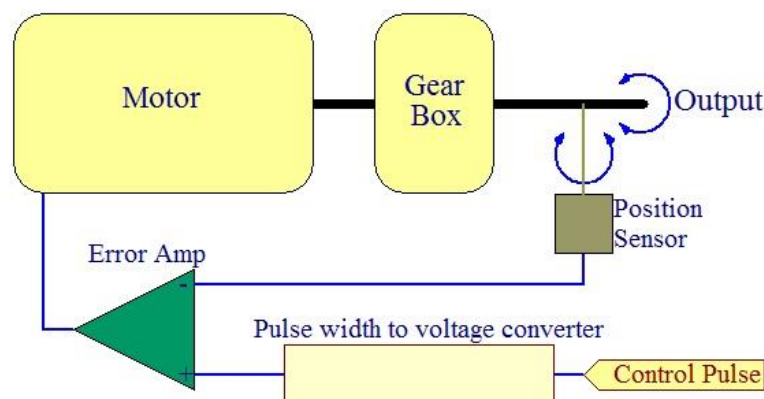
Трајање импулса *PWM* сигнала одређује позицију осовине. Док год је управљачки сигнал присутан, серво мотор задржава позицију своје излазне осовине. При појави сигнала, мотор ротира осовину у смеру у ком ће се најпре остварити задата позиција. Када се позиција оствари, осовина се зауставља и задржава позицију осовине. Управљачки сигнали серво мотора углавном имају период од 20 ms (милисекунди). Фактор испуне импулса унутар периоде ће се одразити на позицију излазне осовине.

На пример, ако желимо да осовину доведемо у угао од -45° потребно је да импулс управљачког сигнала траје 1 ms (фактором испуне од 5%); како би се осовина позиционирала у неутрални положај (0°), ширина импулса треба да је 1,5 ms (фактор испуне 7,5%); уколико пак желимо да осовину позиционирамо на угао од 45° потребна ширина импулса је 2 ms (фактор испуне 10%).



Слика 2 – Смер ротације серво мотора у зависности од ширине импулса управљачког сигнала [5]

Серво мотори су састављени из три дела: мотора са редуктором, уређаја повратне спреге и управљачке плоче. Уређај повратне спреге чини најчешће потенциометар. Мотор, преко низа зупчаника (који чине редуктор), окреће излазну осовину и потенциометар истовремено. Управљачки модул „очитава” отпорност потенциометра и на тај начин утврђује тренутну позицију излазне осовине.

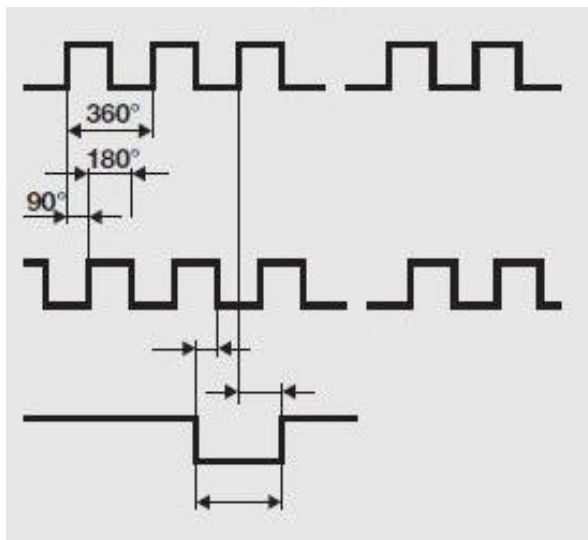


Слика 3 – Шематски приказ структуре серво мотора [5]

2.2 Енкодер

Енкодери су сензорски уређаји који се налазе на моторима и један су од њихових најважнијих делова. Дају нам могућност одређивања позиције осовине мотора што нам олакшава даље управљање жељеним системом. Најзаступљенији су инкрементални, магнетно-отпорни као и оптички енкодери.

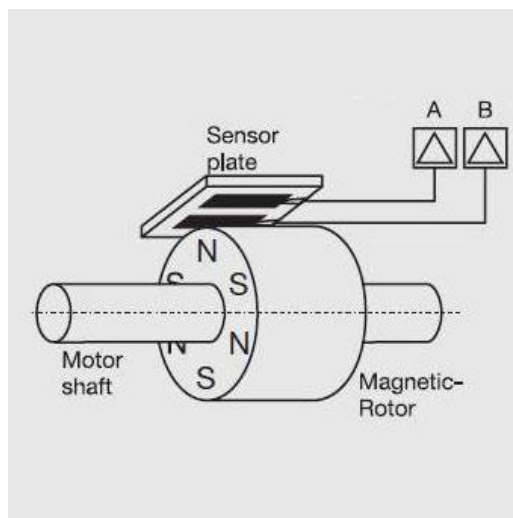
Инкрементални енкодер (бројач) обезбеђује једноставне квадратне сигнале за даљу обраду у контролном систему. Импулси могу бити избројени за тачну позицију или одређивање брзине. Канал А и Б подижу фазу потиснутих / подигнутих сигнала, који се упоређује са још једним одређивањем правца ротације. „Home” сигнал (преглед канала 1) може са користити као обавештење које тачније одређује угао ротације. Линијски погон производи комплементне сигнале А, Б, I који помажу да се елиминише сметња других сигналних линија. Такође, овај електронски погон инсталиран у бројачу побољшава квалитет сигнала помоћу великих сигналних граница. [6]



Слика 4 – Принцип рада инкременталног енкодера [6]

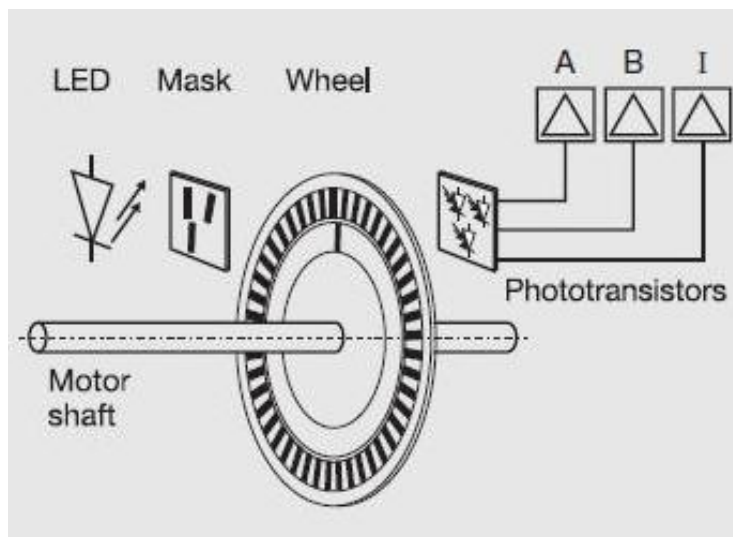
Код магнетно-отпорних енкодера, магнетни оквир диска мултипола на моторној осовини производи синусни напон у магнетном сензору. Класични бројач сигнала је направљен интерполацијом и пречишћавајем електронских сигнала.

Предности магнетно-отпорних енкодера се огледа у томе што заузимају мало простора, нема испупчених делова, може да запамти различите бројеве импулса као и пружање могућности прегледа канала и линијског погона.



Слика 5 – Блок дијаграм магнетно-отпорних енкодера [6]

Оптички енкодер функционише тако што се шаље *LED* (*Light-emiting diode* – Светлећа диода) светло кроз завршно кодно скенирање точка, што је крут (непомичан) оквир на моторној осовини. Прималац (фототранзистор) мења светло-тамне сигнале у одговарајуће електронске импулсе који су опширније објашњени и обрађени у електрониници.



Слика 6 – Принцип рада оптичког енкодера [6]

2.3 Микроконтролер

Намена микроконтролера је управљање уређајима и процесима, па у себи има интегрисан микропроцесор, меморију, дигиталне и аналогне улазе и излазе, дигиталне сатове (тајмере), бројаче, осцилаторе, комуникационе интерфејсе и друге додатке за које је некада био потребан низ посебних интегралних кола (чипова). Микроконтролер нормално ради у контролној петљи, дакле читава улазе и затим подешава излазе у складу са својим програмом. Петља се стално понавља док траје контрола процеса. Брзина микроконтролера се мери у посебним јединицама - флопсовима. [7]



Слика 7 – STM32 микроконтролер [8]

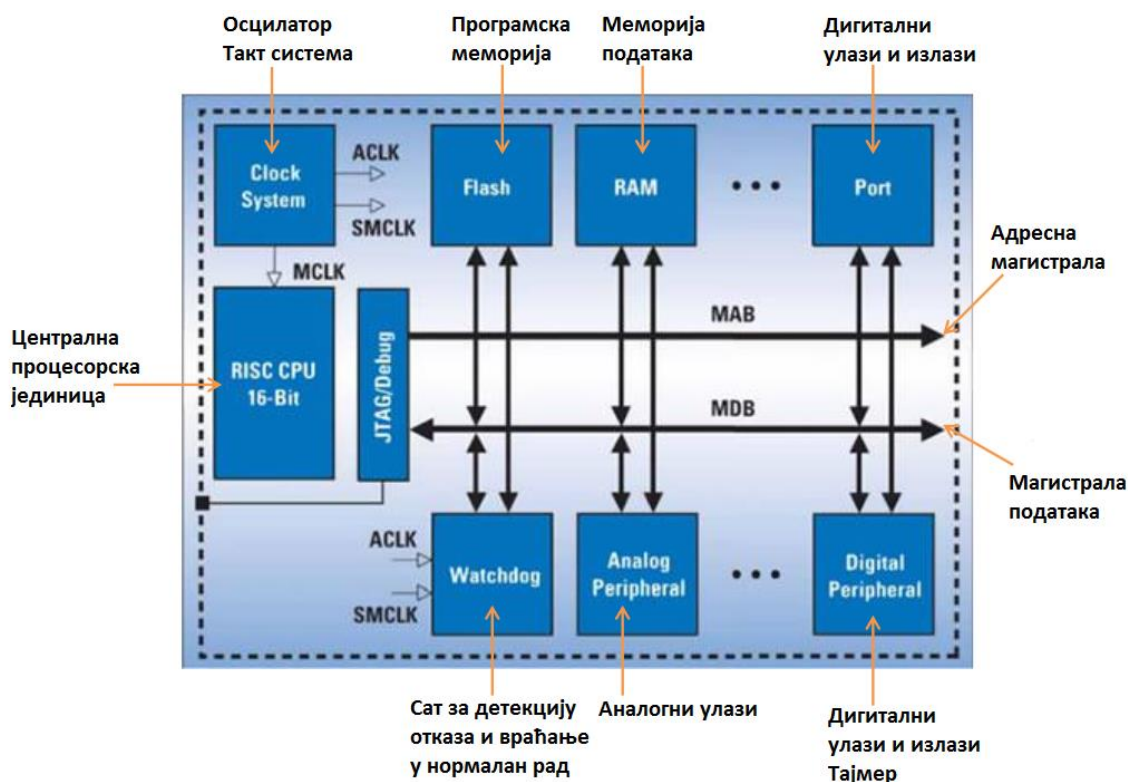
Са појавом првих микропроцесора 1971. године, почела је и њихова употреба у контролне сврхе. Међутим типичан систем је захтевао велики број додатних кола за рад, као што су били аналогно-дигитални претварачи (A/D конвертори), бројачи, осцилатори и друго. Временом је дошло до интегрисања потребних компоненти у једно коло, и тако је створен модерни микроконтролер. [9]

Разне врсте ће имати разне модуле интегрисане у микроконтролер. Ипак већина укључује као минимум следеће делове:

- Микропроцесор
- ROM (*Read Only Memory*), EPROM (*Erasable Programmable Read Only Memory*)
- RAM меморију (*Random Access Memory*)
- Осцилатор
- Бројач
- Дигиталне улазе и излазе

Већина уз горе наведено поседује и:

- Аналогне улазе и излазе
- Комуникационе интерфејсе
- Аналогне поређиваче напона (напонске компараторе)
- Модулатор ширине импулса (*PWM modulator*) за контролу мотора
- и друге.



Слика 8 – Архитектура микроконтролера (MSP430) [10]

Главна разлика између модерних микропроцесора и микроконтролера је да су први оптимизирани за брзину и перформансе код рачунарских програма, док су микроконтролери оптимизирани у правцу интеграције већег броја кола, управљања процесима у стварном времену (*real-time control*), масовну производњу, ниску цену, и малу потрошњу струје. Микроконтролери су отпорнији и на варијације напона, температуре, влажности, вибрације и тако даље. [11]

3. Практични део рада

У овом делу рада биће приказана практична реализација једног дигитално управљаног система серво мотора. Битно је напоменути да је овај систем само један део много већег и комплекснијег система који је своју практичну употребу пронашао унутар сложених борбених система – војна намена.

Управљање се извршава по команди члана борбеног система који за ту команду има надлежност, тако што се из главног рачунара шаље одређена шифрована порука (команда) директно на микроконтролер који је задужен за управљање одређеног елемента. Након тога микроконтролер у зависности од примљене команде и њене обраде, шаље дигиталну поворку импулса драјверу серво мотора где одмах након тога следи извршавање задатих акција – серво мотор почиње са радом.

Због комплексности самог система постоји велики број контролних елемената као што су разни сензори (у највећој мери је реч о индуктивним сензорима и микропрекидачима), али исто тако су присутни и остали системи (нпр. хидраулични, који је после серво система најзаступљенији).

3.1 Пројектовање и развој система

3.1.1 Анализа и дефинисање компоненти система

На почетку пројектовања и развоја система потребно је дефинисати и одабрати потребне компоненте које чине тај систем на основу анализе која се спроводи тако што се посматра шира слика намене и начина употребе наведеног система.

Конкретно за овај рад су потребне следеће компоненте:

- Серво мотор
- Драјвер серво мотора
- Микроконтролер
- Развојно окружење

Током одабира серво мотора, потребно је дефинисати његове параметре односно шта се све од њега очекује – најбитније ставке код одабира мотора су његова снага (изражава се у kW) као и његов обртни момент (изражава се у Nm).

У овом случају одабран је серво мотор *Xinje MS-80ST-M03520BZ-20P7*, номиналних вредности 0.75 kW и 3.5 Nm. Битно је напоменути да наведени мотор ради са наизменичном струјом напона 220 волти и да поседује електронску кочницу. [12]



Слика 9 – Серво мотор *MS-80ST-M03520BZ-20P7* [12]

Након одабира серво мотора потребно је одабрати и одговарајући драјвер серво мотора. Поред услова да одговара одређеном мотору потребно је обратити пажњу и на његове карактеристике као што су доступни начини повезивања комуникације са рачунаром али и доступност одређених интерних подешавања.

За драјвер серво мотора је одабран *Xinje DS2-20P7-AS Servo Driver* који у потпуности испуњава горе наведене услове. Намењен је за моторе од 0.7 kW и такође ради на наизменичном напону од 220 V. [13]

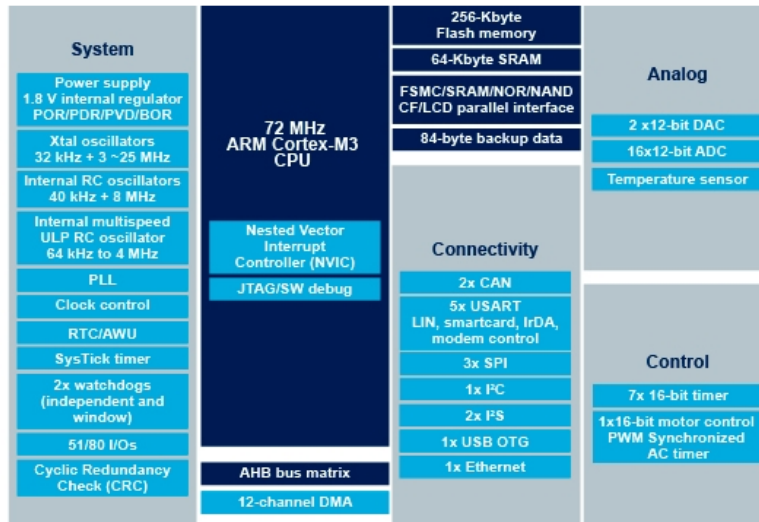


Слика 10 – Драјвер серво мотора DS2-20P7-AS [13]

Одабир микроконтролера је такође веома битна ставка јер уколико се изабере неадекватан микроконтролер, веома је тешко такав систем преправити или унапредити током развоја истог. Пракса је показала да је најбоље одабрати онај микроконтролер који је барем за 50% „бољи и јачи“ у односу на карактеристике које су се дефинисале током анализе.

Као што је већ наведено на почетку рада, у овом случају користимо *STM32F107VCT6* микроконтролер. Овај микроконтролер је заснован на *ARM Cortex M3* 32-битном *RISC* језгру радне фреквенције од 72 MHz. Представља компоненту високих перформанси која поседује 100 пинова, два 12-битна АД конвертора, меморије великих брзина (флеш и *SRAM*), 4 тајмера опште намене резолуције 16 бита као и PWM тајмер.

Што се тиче интерфејса комуникација, ту је реч о две I2C (*Inter-Integrated Circuit*), три SPI-a (*Serial Peripheral Interface*), пет USART-a (*Universal synchronous and asynchronous receiver-transmitter*), USB OTG FS (*On-The-Go*), 2 CAN-a (*Controller Area Network*) као и етернет интерфејс. [14]

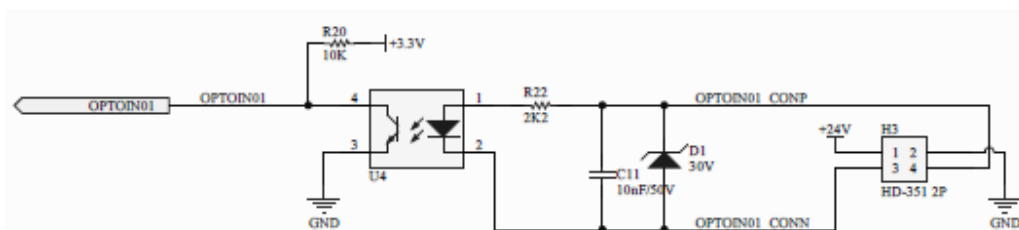


Слика 11 – Блок дијаграм ARM Cortex-M3 микроконтролера [14]

Радни напон се креће од 2.6 V до 3.6 V док микроконтролер може да ради без сметњи на температурама од -40 до +85 °C.

Када је реч о развојном окружењу, његова главна улога је да обезбеди максимално искоришћење микроконтролера односно да поседује довољан број компоненти за повезивање са пиновима микроконтролера, да поседује стабилно коло за напајање као и да пружи заштиту компонентама уколико се догоди некаква хаварија. [15]

Препорука је да се користе опто-изоловани улази и излази ради заштите целог склопа.

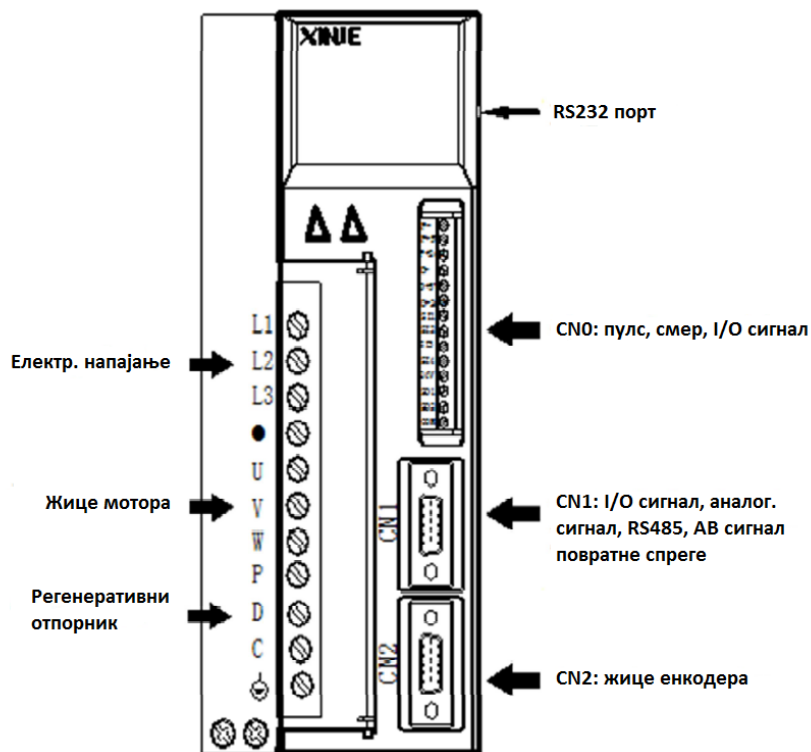


Слика 12 – Приказ опто-изолованог улаза

За реализацију овог рада користили смо већ направљено развојно окружење које се користи за целокупан систем што уједно представља реализовано идејно решење које најбоље одговара нашим потребама.

3.1.2 Блок дијаграм и терминали драјвера

Како би се драјвер серво мотора на исправан и правилан начин повезао, потребно је детаљно прегледати његове карактеристике, блок дијаграме као и терминале главног кола.



Слика 15 – Распоред терминала драјвера [16]

DS2-20P2-AS, DS2-20P4-AS, DS2-20P7-AS												
Terminal	Funkcija	Objašnjenje										
L1/L2/L3	Ulaz napajanja glavnog kola	1 ili 3 fazna AC 200 ~ 240V, 50/60Hz										
⊕	Uzemljenje	Poveziv. na terminal uzemljenja motora										
P+, PB	Regenerativni otpornik	Povezivanje regener.otpornika između P+ i PB										
U, V, W	Terminali motora	Povezivanje motora <table> <tr> <th>Terminal</th><th>Boja</th></tr> <tr> <td>U</td><td>braon</td></tr> <tr> <td>V</td><td>crna</td></tr> <tr> <td>W</td><td>plava</td></tr> <tr> <td>PE</td><td>žuto zelena</td></tr> </table>	Terminal	Boja	U	braon	V	crna	W	plava	PE	žuto zelena
Terminal	Boja											
U	braon											
V	crna											
W	plava											
PE	žuto zelena											
⊕	Uzemljenje	Poveziv.na terminal uzemljenja motora										

Слика 16 – Терминали главног вода [16]

■ CN0 terminali

Br.	Naziv	Objašnjenje	Br.	Naziv	Objašnjenje
1	P-	Impulsni ulaz PUL-	8	SI2	Ulaz 2
2	P+5V	5V diferencijal.ulaz	9	SI3	Ulaz 3
3	P+24V	OC ulaz	10	SI4	Ulaz 4
4	D-	DI ulaz DIR-	11	+24V	Ulaz +24V
5	D+5V	5V DI ulaz	12	SO1	Izlaz 1
6	D+24V	OC ulaz	13	SO2	Izlaz 2
7	SI1	Ulaz 1	14	COM	Uzemlj.izlaza

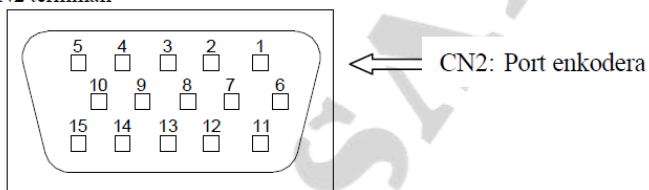
Слика 17 – CN0 терминали [16]

■ CN1 (DB15) terminali

Br.	Naziv	Objašnjenje	Br.	Naziv	Objašnjenje
1	NC	Rezervisan	9	Z-	Izlaz enkodera Z-
2	NC	Rezervisan	10	B+	Izlaz enkodera B+
3	SI5	Ulaz 5	11	T-REF	Analog.ulaz ob.mom.
4	SO3	Izlaz 3	12	V-REF	Analog.ulaz brzine
5	B-	Izlaz enkodera B-	13	GND	模拟量输入用 GND
6	A+	Izlaz enkodera A+	14	A	RS485+
7	A-	Izlaz enkodera A-	15	B	RS485-
8	Z+	Izlaz enkodera Z+			

Слика 18 – CN1 терминали [16]

■ CN2 terminali



Port drayvera	Port enkodera motora		Naziv	Port drayv.	Port enkodera motora		Naziv
	60, 80, 90 serija	110, 130, 180 serija			60, 80, 90 serija	110, 130, 180 serija	
1	9	4	A+	2	4	5	B+
3	7	6	Z+	4	6	10	U+
5	11	12	W+	6	13	7	A-
7	14	8	B-	8	5	9	Z-
9	8	13	U-	10	15	15	W-
11	1	1	Povez. zaštit. kablom	12	3	3	GND
13	2	2	5V	14	10	11	V+
15	12	14	V-				

Слика 19 – CN2 терминали [16]

3.2 Израда софтвера дигиталног управљања

На почетку израде софтвера потребно је претходно упознати се са техничким спецификацијама свим елемената који се користе. Најважније је добро разумевање пин спецификације односно утврђивање који се пин за шта може користити. Након тога следи одабир оних пинова који у потпуности одговарају дефинисаним наменама и који неће представљати сметњу уколико касније буде било потребе за унапређењем система или дела система.

3.2.1 Развојно окружење и програмски језик

За израду софтвера дигиталног управљања користи се интегрисано развојно окружење *mikroC PRO for ARM* које је, као што се види из назива, специјално намењено за рад са микроконтролерима архитектуре типа *ARM*. Поред њега користи се и *mikroC* програмски језик који је такође намењен за рад са микроконтролерима и који у великој мери олакшава разумевање и њихово коришћење.

3.2.2 Подешавање и одабир брзина

Подешавање и одабир брзине серво мотора коришћењем поворке дигиталних импулса се може поделити у две фазе. Прва фаза се односи на подешавање и одабир пинова који су потребни за наведени рад код микроконтролера као и подешавање драјвера где је потребно одабрати жељени тип (мод) рада, док се током друге фазе програмира софтвер управљања.

Код подешавања драјвера потребно је изабрати одговарајући режим рада, односно извршити избор контролног мода. *DS2* тип серво драјвера има један главни мод и и унутар њега се налазе под модови. У нашем случају потребно је приступити контролном моду *P0-01* и унутар њега подесити параметар 7 – контрола брзине серво мотора преко фреквенције импулса.

Parametar		Kontrolni mod
P0-00	0	Mod testiranje, korisnici ga ne mogu menjati!
P0-01 Pod mod1	0	Neaktivni mod Motor se neće uključiv. čak i ako se osposobi servo drajver.
	1	Kontrola obrtnog momenta (interna postavka) Kontrola izlaznog obrt.momenta servo motora preko operativnog panela ili komunikacionog porta.
	2	Kontrola obrtn.momenta (komanda analognog napona) Kontrola izlaz.obrtnog momenta servo motora preko analogne komande napona. Najčešće se koristi u sistemima zatvorene petlje za kontrolu obr.momenta sa gornjim uređajem, kao što je kontrola tenzije. (Ako servo drajver nema port za anal.ulaz, ovaj mod se ne može koristiti.)
	3	Kontrola brzine (postavka interne brzine) Upotrebite /SPD-D, /SPD-A, /SPD-B za izbor brzine koja je postav u servo drajveru. Servo drajver može postaviti 3 brzine.
	4	Kontrola brzine (komanda analognog napona) Kontrola brzine servo motora preko analog.komande napona. Najčešće se kor.u sistem.sa zatvor.petljom za kontrolu sa gornjim uređajem.
	5	Kontrola pozicije (komanda interne pozicije) Kontrola pozicije preko interne komande pozicije; može se postaviti količina impulsa i obrtni moment.
	6 (default vred.)	Kontrola pozicije (eksterna komanda za poziciju) Kontrola pozicije servo motora preko impulsne komande. Kontrola pozicije preko količine impulsa; kontr.brzine preko frekv. impulsa.
	7	Kontrola brzine (komanda frekvencije impulsa) Kontrola brzine servo motora preko frekv. impulsa, ali ne i kontrola pozicije.

Слика 20 – Избор контролног мода [16]

Управљање се извршава коришћењем три сигнала:

- *ENA (Enable - Дозвола)* – Сигнал који дозвољава рад серво мотору
- *DIR (Direction – Смер)* – Сигнал којим се бира смер окретања осовине серво мотора
- *PLS (Pulse - Импулс)* – Сигнал којим се покреће рад (окретање осовине) серво мотора

Потом се може прећи на подешавање и одабир пинова микроконтролера који служе за слање горе наведених сигнала – логичке јединице или нуле.

Како је реч о већ готовом развојном окружењу, можемо одабрати доступне излазне терминале (излазне пошто шаљемо сигнале од микроконтролера ка серво систему) који нам одговарају и на основу тог одабира подесити улогу и намену пинова микроконтролера који су повезани са одабраним терминалима.

Пин	I/O	Позиција	Назив	Опис
PA_0	Output	H - 31	DRVIN01-01	PLS
PC_3		H - 39	DRVIN01-02	ENA
PC_2		H - 30	DRVIN01-03	DIR

Слика 21 – Намена одабраних пинова

Одабрани пинови су:

- Пин A0 – Његова позиција на развојном окружењу је H-31 и он се користи за слање сигнала PLS
- Пин C3 – Његова позиција на развојном окружењу је H-39 и он се користи за слање сигнала ENA
- Пин C2 - Његова позиција на развојном окружењу је H-30 и он се користи за слање сигнала DIR

Након дефинисања пинова потребно је и подесити те пинове сходно њиховој намени. Како је реч о дигиталном управљању, потребно је поставити да се и пинови понашају као дигитални излази микроконтролера.

Команда за подешавање:

```
GPIO_Digital_Output(&GPIOx_BASE, _GPIO_PINMASK_y);
```

x – представља ознаку порт-а (A, B, C итд.)

y – представља ознаку пин-а (0, 1, 2 итд.)

Ради лакшег сналажења током израде софтвера није лоше да се одабрани пинови подесе и као статичке променљиве и то је најбоље учинити тако да нас њихове ознаке асоцирају на њихову намену:

```
static sbit PLS at GPIOA_ODR.B0;
static sbit ENA at GPIOC_ODR.B3;
static sbit DIR at GPIOC_ODR.B2;
```

Након тока може се почети са програмирањем софтвера дигиталног управљања. У наставку је приказан упрошћени изглед софтвера (део изворног кода) који се користи за контролу брзине код серво мотора где се одабир брзине заснива на фреквенцији дигиталних импулса односно брзини промене логичких стања сигнала *PLS*. Већа брзина се постиже смањењем времена између логичких стања наведеног сигнала и обрнуто.

```
void motor_start (){  
    ENA = 1;  
    DIR = 1;  
    PLS = 1;  
    Delay_Cyc(10);  
    PLS = 0;  
    Delay_Cyc(10); }
```

Сигнал *ENA* је потребно поставити на логичку јединицу да би сам серво драјвер омогућио (дозволио) рад серво мотора. Сигнал *DIR* се поставља на логичку јединицу или нулу у зависности од смера у коме је потребно да се осовина мотора креће гледано са стране оптерећења – унутар подешавања драјвера могуће је одабрати које логичко стање сигнала представља *CCW* (*Counter clockwise – Супротан смер казаљке на сату*) или *CW* (*Clockwise – Смер казаљке на сату*) смер окретања.

Сигнал *PLS* се користи директно за управљање брзином тако што промена његовог логичког стања представља „активно“ стање кретања осовине. Уколико је логичка јединица присутна – мотор добија команду да окреће осовину, док логичка нула представља команду за заустављање. Време односно пауза између два логичка стања дефинише брзину мотора и за ту намену се користи уграђена функција *Delay_Cyc* која креира паузу базирану на микроконтролерски осцилатор. Пауза траје 10 пута више у односу на унети параметар гледано из перспективе микроконтролера. На пример уколико смо поставили „10“ као параметар ове функције, то представља 100 циклуса микроконтролера и током тог времена логичко стање се не мења али је битно напоменути да серво мотор и током тог времена ради у зависности од претходног стања које се и даље налази на том излазном пину микроконтролера – уједно и на улазном пину драјвера серво мотора.

3.2.3 Подешавање и одабир позиције

Подешавање и одабир позиције серво мотора коришћењем поворке дигиталних импулса је веома слично као и код подешавања и одабира брзине. Прво је потребно одабрати одговарајуће пинове и подесити драјвер серво мотора у одговарајући режим рада, а потом се програмира софтвер управљања

Драјвер серво мотора у овом случају је потребно подесити на параметар 6 контролног мода P0-01 – контрола позиције серво мотора преко количине импулса (слика 20.).

Управљање се такође извршава коришћењем три сигнала:

- *ENA (Enable* - Дозвола) – Сигнал који дозвољава рад серво мотору
- *DIR (Direction* – Смер) – Сигнал којим се бира смер окретања осовине серво мотора
- *PLS (Pulse* - Импулс) – Сигнал којим се покреће рад (окретање осовине) серво мотора

Након дефинисања пинова може се почети са подешавањем истих, али како је поступак односно подешавање идентично као и у претходном делу, кренућемо са даљим радом – програмирањем софтера дигиталног управљања.

Програмирање софтера дигиталног управљања за одабир позиције се разликује у томе што је код њега потребно дефинисати начин помоћу којег ће се одређивати тачна угаона позиција серво мотора. Позиција се посматра у односу на осовину која се током рада мотора окреће и која условно речено прелази одређену дистанцу – пут.

У наставку је приказан упрошћени изглед софтера (део изворног кода) који се користи за контролу позиције код серво мотора где је примењена метода „бројање импулса“. Ова метода функционише на такав начин да након сваког завршетка целе периоде (трајање логичке нуле и јединице заједно) сигнала *PLS* повећава (инкрементује) бројач. У главном програму се подешава до ког броја бројач броји и када бројач дође до тог броја, серво мотор престаје са радом.

```

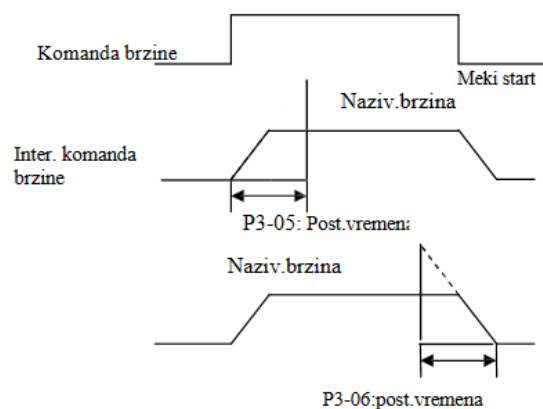
void motor_start(){
    ENA = 1;
    DIR = 1;
    PLS = 1;
    Delay_Cyc(10);
    PLS = 0;
    Delay_Cyc(10);
    brojac = brojac + 1;}

```

Пошто су нам познате све карактеристике микроконтролера, применом опште математике могуће је одредити тачну позицију серво мотора. Наравно битно је напоменути да је због механичких карактеристика могуће јављање проклизавања – проблем који се углавном решава применом убрзања (на почетку) и успорења (на крају) кретања мотора. У литератури се углавном проналази под називом „меки старт“ или „меки стоп“.

P3-05	Vreme ubrzanja mekog starta					
	Jedinica	Default	Opseg	Podesan mod	Izmena	Dejstvo
	1ms	0	0~65535	3, 4, 7	Servo OFF	Trenutno
P3-06	Vreme usporavanja mekog starta					
	Jedinica	Default	Opseg	Podesan mod	Izmena	Dejstvo
	1ms	0	0~65535	3, 4, 7	Servo OFF	Trenutno

Слика 22 – Контролни мод за меки старт-стоп [16]



Слика 23 – Принцип рада убрзања-успорења [16]

Све остале карактеристике у вези смера окретања и брзине важе исто као и код дигиталног управљања брзином серво мотора.

3.2.4 Контрола брзине интерним подешавањем

У данашње време код већине система за управљање серво мотора могуће је применити доступне напредне опције самог система – тачније те опције поседује драјвер серво мотора и у томе се највише и огледа квалитет једног таквог система. Да бисмо на правилан начин искористили доступне могућности, потребно је прво детаљно да се проучи један такав систем.

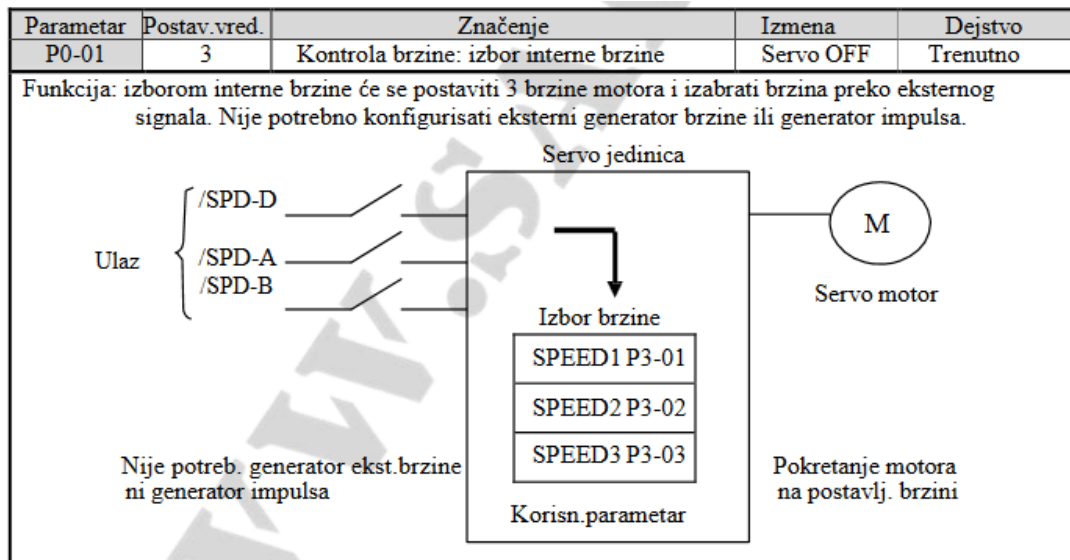
Најбитнији део је део који се тиче подешавања самог драјвера, јер је за то потребно веома добро познавање и разумевање области која се бави контролом и управљањем драјвера мотора.

За коришћење ове опције потребно је подесити следеће:

Osnovni parametar		
Parametar	Naziv	Referenca
P0-01	Izbor kontrolnog moda	5-6-1
P5-10	Servo ON signal /S-ON	5-2-1
P3-01	Interna brzina 1	5-6-2
P3-02	Interna brzina 2	
P3-03	Interna brzina 3	
P5-17	/SPD-D izbor smer interne brzine	5-6-3
P5-18	/SPD-A izbor interne brzine	
P5-19	/SPD-B izbor interne brzine	

Слика 24 – Опције за подешавање интерног мода [16]

Прво подешавање је одабир контролног мода режима рада драјвера (серво мотора). Унутар контролног мода потребно је поставити параметар 3 – контрола брзине (коришћењем интерних брзина – слика 20.).



Слика 25 – Опис контролног мода [16]

Потом је потребно подесити серво сигнал (S-0N) који дефинише стање рада мотора – у овом случају постављамо на параметар 0010 који предстаља активно стање рада мотора. Подешавање поставља систем на мод да увек ради што значи да нема потребе за довођењем сигнала на тај терминал.

Parametar	Signal	Postavka	Objašnjenje	Opseg
P5-10	/S-0N	n.0001 (default vredn.)	Kad je SI1 ON, servo motor se uključuje i osposobljava za pokretanje.	0001~0015
		n.0010	Uvek deluje, nema potrebe za ožičenjem.	

Слика 26 – Опис серво S-0N сигнала [16]

Затим се прелази на подешавање интерних брзина, односно вредности које представљају брзину у јединици - ротације по минути (*rotations per minute - rpm*). Максимална брзина је 5000 ротација по минути. Подешавање параметара се извршава унутар мода P3 и његових под модова – 01, 02 и 03. Као што се може закључити, имамо могућност подешавања као и коришћења три брзине.

P3-01	Interna brzina 1					
	Jedinica	Default	Opseg	Podesan mod	Izmena	Dejstvo
	1rpm	100	-5000 ~ +5000	3	Bilo koja	Trenutno
P3-02	Interna brzina 2					
	Jedinica	Default	Opseg	Suitable mode	Izmena	Dejstvo
	1rpm	200	-5000 ~ +5000	3	Bilo koja	Trenutno
P3-03	Interna brzina 3					
	Jedinica	Default	Opseg	Suitable mode	Izmena	Dejstvo
	1rpm	200	-5000 ~ +5000	3	Bilo koja	Trenutno

Слика 27 – Подешавање интерне брзине [16]

Потом је потребно дефинисати терминале који се користе као улазни сигнали драјвера (излазни сигнали микроконтролера).

Parametar	Signal	Default	Značenje	Izmena
P5-17	/SPD-D	n.0000	Potreb.distribuir.	Opseg: 0000-0015. Distribuirati na ulazni terminal kroz P5-17.
P5-18	/SPD-A	n.0000	Potreb.distribuir	Opseg: 0000-0015. Distribuirati na ulazni terminal kroz P5-18.
P5-19	/SPD-B	n.0000	Potreb.distribuir	Opseg: 0000-0015. Distribuirati na ulazni terminal kroz P5-19.

Слика 28 – Подешавање улазних сигнала [16]

Одабир интерне брзине односно реализација функције која поставља жељену брзину серво мотора се заснива на комбинацији улазних сигнала *SPD_D*, *SPD_A* и *SPD_B*.

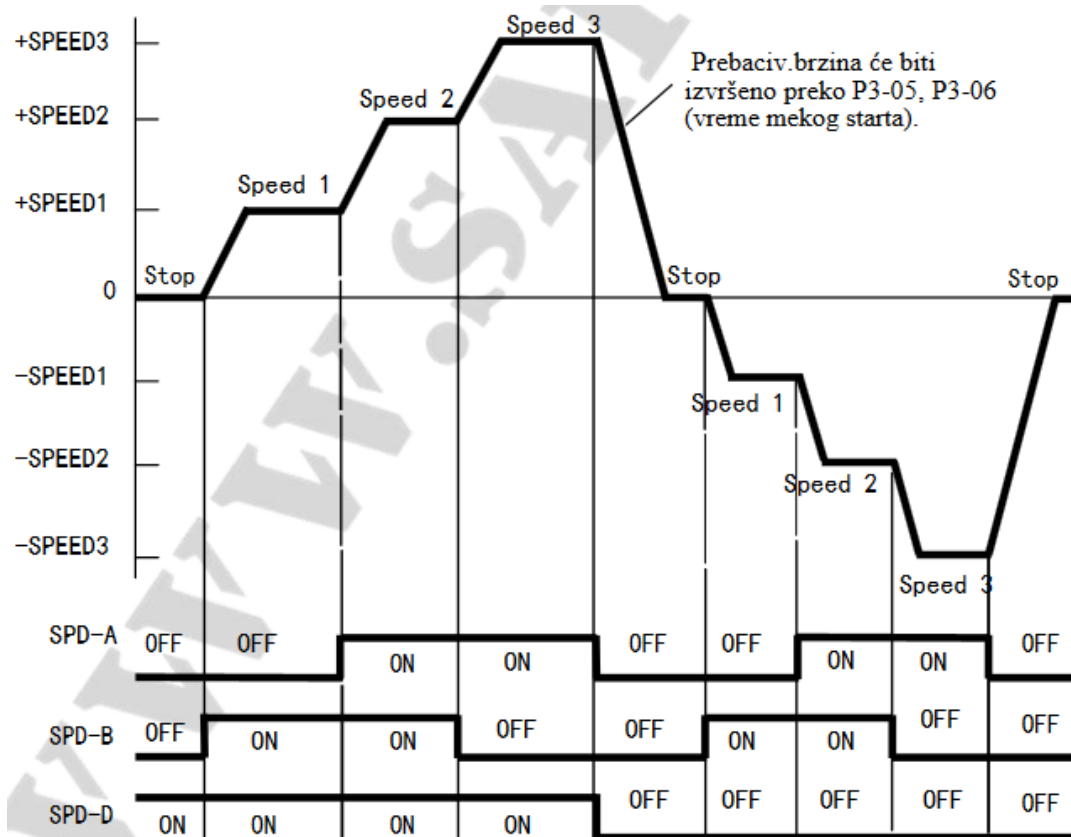
Сигнал *SPD_D* се користи за одабир смера кретања осовине мотора. док се сигнали *SPD_A* и *SPD_B* користе за одабир једне од три брзине као и могућност заустављања.

Комбинације улазних сигнала и реализација функције одабира подешавања на основу њих:

Ulazni signal			Smer	Brzina motora tokom rada
/SPD-D	SPD-A	SPD-B		
0	0	0	Forward (supr.od sm.kaz.na satu)	Interna brzina je nula
	0	1		P3-01: SPEED1 (BRZINA1)
	1	1		P3-02: SPEED2 (BRZINA2)
	1	0		P3-03: SPEED3 (BRZINA3)
1	0	0	Reverse (U smeru kaz.na satu)	Interna brzina je nula
	0	1		P3-01: SPEED1
	1	1		P3-02: SPEED2
	1	0		P3-03: SPEED3

Слика 29 – Комбинације улазних сигнала и намена [16]

Да би се на што бољи начин разумео принцип рада, на слици испод је приказан пример рада мотора током коришћења ове опције.



Слика 30 – Принцип рада у моду 3 [16]

Параметар	Вредност	Опис
P0-01	3	Избор контролног мода
P5-10	0010	Серво ON сигнал /S-0N
P5-12	6	Забрана покретања forward /P-OT
P5-13	7	Забрана покретања reverse /N-OT
P5-17	4	Смер интерне брзине /SPD-D, улаз SI4
P5-18	2	Избор интерне брзине /SPD-A, улаз SI2
P5-19	3	Избор интерне брзине /SPD-B, улаз SI3
P3-01	900	Интерна брзина 1
P3-02	2500	Интерна брзина 2
P3-03	300	Интерна брзина 3

Слика 31 – Подешене вредности

На слици 31. је приказан кратки преглед вредности параметара као и кратки опис свих подешених опција нашег система.

Такође је битно напоменути да се за сваку промену подешавања драјвера прво мора променити вредност параметра P5-10 (S-0N сигнал) на вредност 0001. Када је унос нових промена (вредности) завршен потребно је вратити претходно подешавање промењеног параметра – 0010.

Након завршеног подешавања драјвера серво мотора, можемо започети подешавање микроконтролера и програмирање софтвера за дигитално управљање.

Одмах на почетку вршимо одабир пинова за излазне сигнале:

Пин	I/O	Позиција	Назив	Опис
PC_0	Output	H - 38	DRVIN01-04	SPD_A
PC_13		H - 29	DRVIN01-05	SPD_D
PE_6		H - 37	DRVIN01-06	SPD_B

Слика 32 – Намена одабраних пинова

Одабрани пинови су:

- Пин C0 – Његова позиција на развојном окружењу је H-38 и он се користи за слање сигнала SPD_A
- Пин C13 – Његова позиција на развојном окружењу је H-29 и он се користи за слање сигнала SPD_D
- Пин E6 - Његова позиција на развојном окружењу је H-37 и он се користи за слање сигнала SPD_B

Након дефинисања пинова потребно је и подесити те пинове сходно њиховој намени, исто смо урадили и за претходна управљања. Такође и овде примењујемо добру праксу да пиновима доделимо називе који нас асоцирају на њихову улогу.

```
static sbit SPD_A at GPIOC_ODR.B0;
static sbit SPD_D at GPIOC_ODR.B13;
static sbit SPD_B at GPIOE_ODR.B6;
```

Након тока може се почети са програмирањем софтвера дигиталног управљања. Програмирање софтвера који управља радом серво мотора путем овог начина је веома једноставан.

Потребно је само довести горе наведене комбинације сигнала на излазе нашег микроконтролера – који су уједно и улазни сигнали за драјвер серво мотора.

```
void internal_CCW_1(){ void internal_CW_1(){
SPD_D = 0;             SPD_D = 1;
SPD_B = 1;             SPD_B = 1;
SPD_A = 0;             SPD_A = 0;
}                       }

void internal_CCW_2(){ void internal_CW_2(){
SPD_D = 0;             SPD_D = 1;
SPD_B = 1;             SPD_B = 1;
SPD_A = 1;             SPD_A = 1;
}                       }

void internal_CCW_3(){ void internal_CW_3(){
SPD_D = 0;             SPD_D = 1;
SPD_B = 0;             SPD_B = 0;
SPD_A = 1;             SPD_A = 1;
}                       }
```

Слика 33 – Комбинације за одабир брзине мотора

3.2.5 Контрола позиције инкременталним енкодером

Управљање позицијом серво мотора помоћу инкременталног енкодера представља можда и најзаступљенији начин путем којег се то управљање обавља. Скоро сви озбиљнији данашњи серво системи поседују већ уграђени давач положаја односно енкодер. У нашем случају DS2 серво драјвер поседује инкрементални енкодер резолуције 2500 ppr (*pulses per rotation* – импулса по ротацији). То значи да за један пређени пун круг (360°) осовине мотора, када прочитамо очитавање енкодера видећемо 2500 импулса, што нам помаже да одредимо тачну позицију осовине мотора (колико растојање-дистанцу је прешао).

Начин на који ћемо вршити покретање и заустављање мотора није од пресудног значаја када користимо енкодер. У овом случају поред излазних пинова (сигнала), микроконтролер користи и улазне пинове путем којих добија повратну информацију од енкодера – импулсе.

Како знамо да је 2500 импулса показатељ једног пуног круга, применом најједноставнијег математичког рачуна можемо веома лако доћи до броја који је потребан за неки одређени угаони померај.

Пример: Уколико нам механички склоп за померање неког елемента захтева 200 обртаја мотора, то значи да је потребно да држимо мотор активним (режим рада) све док нам на се на улазу микроконтролера не појави 500 хиљада импулса ($200 * 2500$). Наравно треба узети у обзир да ли се унутар механичког склопа налази редуктор и ако се налази потребно је и њега узети у обзир током рачуна.

Напомена: Све горе наведено подразумева идеалне елементе односно уређаје, али се у реалној примени користе како додатни модули тако и посебни софтвери који исправљају грешке углавном механичког порекла – проклизавање, трење, крутост материјала итд.

Приликом израде софтвера дигиталног управљања, одмах на почетку је потребно дефинисање улазних пинова микроконтролера. У овом примеру користили смо пин *D6* као улазни пин на којем се поставља сигнал енкодера који шаље повратну информацију у виду импулса. Да бисмо били сигурни да микроконтролер читава без грешке долазне сигнале (импулсе) поставили смо га да ради у режиму прекида (*interrupt*).

```
RCC_APB2ENR.AFIOEN = 1;
AFIO_EXTICR3 = 0x0300;
AFIO_EXTICR2 = 0x0300;
EXTI_FTSR = 0x00000440;
EXTI_IMR |= 0x00000440;
NVIC_IntDisable(IVT_INT_EXTI15_10);
NVIC_IntDisable(IVT_INT_EXTI9_5);
```

Слика 34 – Подешавање прекида

На слици 34. је приказано подешавање микроконтролера тако да пин *D6* функционише као екстерни прекид са доњом окидном ивицом.

Потом је потребно креирати функцију која долазне импулсе (са енкодера) пребројава и која ту вредност смешта унутар неке променљиве.

```
void ExtInt_2() iv IVT_INT_EXTI9_5 ics ICS_AUTO {  
    EXTI_PR.B2 = 1;  
    enkoder = enkoder + 1;  
}
```

Слика 35 – Функција за бројање долазних импулса

Сада на основу вредности те променљиве знамо колико је енкодер послао импулса односно колико је растојање серво мотор прешао.

На слици испод се налази упрошћени изворни код софтвера чија је намена да интерном брзином 1 мотор осовину окрене 12 пуних кругова, интерном брзином 2 осовина се окреће 8 пута док се интерном брзином 3 осовина мотора окреће 4 пута. Такође је потребно на почетку и крају функције поставити дозволу и забрану спољашњег прекида као и постављање променљиве на почетну вредност нула, да би софтвер радио са максималном прецизношћу.

```
void start_motor_enc(){  
    NVIC_IntEnable(IVT_INT_EXTI9_5);  
    enkoder = 0;  
    internal_1();  
    while(enkoder < 30000){  
        }  
    internal_2();  
    while(enkoder < 50000){  
        }  
    internal_3();  
    while(enkoder < 60000){  
        }  
    internal_stop();  
    enkoder = 0;  
    NVIC_IntDisable(IVT_INT_EXTI9_5);  
    enkoder = 0;  
}
```

Слика 36 – Примена енкодера

4. Закључак

Тема дипломског рада је била заснована на имплементацији дигиталног управљања серво мотора где је главни циљ био да се управљање једног таквог система у потпуности анализира и уједно реализује како у теоријском тако и у практичном смислу. Током анализе највише пажње је било усмерено на потребе - захтеве наведеног система као и на одабиру потребних елемената који су потребни да би овакав један систем било могуће реализовати. Реализација система је као примарни циљ имало примену у аутоматским управљачким системима који захтевају високе динамичке перформансе и прецизност. Практична реализација је захтевала израду система – прототипа који је требало испитати и што је више могуће унапредити у самом току рада, док су се сви недостаци који су се појављивали током коришћења наведеног система, морали отклонити – што је резултирало уједно и самим унапређењем система. Систем је током времена достигао одређене границе и уједно задовољио већину захтева који су били у почетку дефинисани.

Даљи развој и унапређења осталих система у наредном периоду проузроковаће и потребу да се наш систем мења сходно новим захтевима. Систем је показао високи ниво перформанси и могућности где је због тога добио високе оцене током практичне употребе. Концепт оваквог система задржаће се и у наредном периоду са веома малим изменама, док ће главни акценат бити на промени компонената које се користе. Промена има за циљ постизање што бољих перформанси и оцена – задовољавање претходно постављених али и будућих захтева.

Литература

- [1] Servomotor, Приступљено: 03.10.2019. године, линк:
<https://en.wikipedia.org/wiki/Servomotor>
- [2] Слободан Н. Вукосавић: Дигитално управљање електричним погонима, Академска мисао, Београд 2003.
- [3] Електромотор, Приступљено: 03.10.2019. године, линк:
<https://sh.wikipedia.org/wiki/Elektromotor>
- [4] Разни делови електромотора, Приступљено: 04.10.2019. године, линк:
http://www.eldon.rs/body_files/body_serbian/proizvodni_program_files/rde.htm
- [5] ПЦ Серво мотори, Приступљено: 03.10.2019. године, линк:
<https://www.automatika.rs/baza-znanja/mehatronika/rc-servo-motori.html>
- [6] Енкодери, Приступљено: 03.10.2019. године, линк:
<https://www.automatika.rs/baza-znanja/senzori/enkoderi.html>
- [7] Микроконтролер, Приступљено: 03.10.2019. године, линк:
<https://sr.wikipedia.org/sr-ec/mikrokontroler>
- [8] Introduction to Programming STM32 ARM Cortex-M 32-bit Microcontrollers, Приступљено: 03.10.2019. године, линк:
<https://predictabledesigns.com/introduction-to-programming-stm32-arm-cortex-m-32-bit-microcontrollers/>
- [9] Stuart R. Ball: Embedded Microprocessor Systems: Real World Design (Embedded Technology), Elsevier Inc., Oxford 2002.
- [10] Микроконтролери, Приступљено: 03.10.2019. године, линк:
<http://tnt.etf.bg.ac.rs/~oo1ue/predavanja/P6.pdf>
- [11] Ендрју С. Таненбаум: Архитектура и организација рачунара, Микро књига, Београд 2007.
- [12] MS-80ST-M03520BZ-20P7, Приступљено: 04.10.2019. године, линк:
http://www.sah.co.rs/ms-80st-m03520bz-20p7.html?_SID=U
- [13] DS2-20P7-AS, Приступљено: 04.10.2019. године, линк:
<http://www.sah.co.rs/ds2-20p7-as.html>
- [14] STM32F107VC, Приступљено: 04.10.2019. године, линк:
<https://www.st.com/en/microcontrollers-microprocessors/stm32f107vc.html>
- [15] Милић Р. Стојић: Дигитални управљачки системи, Академска мисао, Београд 2004.
- [16] DS2-AS, Приступљено: 04.10.2019. године, линк:
http://www.sah.co.rs/media/sah/techdocs/ds2-as_uputstvo.pdf