Konstruiranje makete sustava za pozicioniranje kamere u ravnini

Štampar, Tomislav

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

ZAVRŠNI RAD

POZICIONIRANJE KAMERE U RAVNINI

Voditelj rada:
Prof.dr.sc. Davor Zorc

Tomislav Štampar
4-MEHATRON
0035153927

Zagreb, 2010

IZJAVA

Izjavljujem da sam ovaj završni rad radio samostalno na Fakultetu strojarstva i brodogradnje u Zagrebu znanjem stečenim tijekom studija.

Tomislav Štampar

ZAHVALA

Zahvaljujem se mentoru prof. dr. sc. Davoru Zorcu na iskazanom povjerenju i pomoći tijekom izrade ovog rada.

Zahvaljujem se dr.sc.Danijelu Pavkoviću na pomoći oko rada i na stručnom vodstvu.

Također se zahvaljujem svim članovima Katedre za strojarsku automatiku i svim svojim prijateljima koji su mi pomagali.

Tomislav Štampar

Sadržaj

Sadržaj 3
1. UVOD 8
2. PRINCIPIJELNI PRIKAZ RADA MAKETE 9
3. OPIS RADA MAKETE 10
4. IGRAČI KONTROLER 15
4.1. Opis 16
5. USB PORT 17
5.1. Inačice 17
5.2. Način rada 17
5.3. USB signaliranje 18
5.4. Brzine USB uređaja 18
6. AUTOIT 19
6.1. Značajke AutoIT-a u detaljima 20
6.2. Objašnjenje načina rada skriptnog kôda 22
7. PARALELNI PORT 24
8. MIKROKONTROLERI 29
8.1. Općenito 29
8.2. Mikrokontroler PIC 16F628A 32
9. MICROCODE STUDIO 33
9.1. Objašnjenje načina rada programskog kôda 35
10. Koračni motor 39
10.1. Općenito 39
10.2. Upotreba 39

10.3. Konstrukcija 39
10.4. Podjela 41
10.5. Najčešći oblici pobude 42
10.6. Napajanje strujom 43
11. VIJČANI POGON 45
12. ZAKLJUČAK 46
LITERATURA 47
A PRILOG 48
A.1. CAD model 49
A.2. Shema spajanja 51
A.3. Podatkovna tablica koračnih motora 53
A.4. Karakteristike PIC16F628A 55
A.5. Podatkovne tablice naponskog regulatora 57
A.6. Skripta za inicijalizaciju igraćeg kontrolera 62
A.7. Programski kôd 69
A.8. Proračun trapeznog vretena s normalnim jednovojnim trapeznim navojem (HRN M.B0.060 do 064) 71

Popis slika

Slika 2. Principijelni prikaz rada makete 9

Slika 3.1. Prikaz pozicija analognog dijela igračeg kontrolera sa očitanim vrijednostima 10

Slika 3.2. Referenca analognog dijela igračeg kontrolera 11

Slika 3.1. Prikaz izlaznih napona iz paralelnog porta 12

Slika 3.3. Portovi korištenog mikrokontrolera PIC16F628A 13

Slika 3.4. Shematki prikaz naponskog regulatora ULN2068B 13

Slika 4.1. Upravljački igrači kontroler 15

Slika 4.2. Analogni dio upravljačkog igračeg kontrolera 16

Slika 4.1. Standardni USB utikač 17

Slika 5.2. Standardna USB A i B utičnica 18

Slika 5.3. Mini USB A i B utičnica 18

Slika 6.1. Izgled grafičkog korisničkog sučelja 21

Slika 6.2. Dijagram toka funkcija aplikacije za inicijalizaciju igračeg kontrolera i postavljanja izlaza parlelnog porta 22

Slika 7.1. Izgled paralelnog porta 24

Slika 7.2. Adresiranje paralelnog porta 26

Slika 8.1. Načelna blok-shema mikrokontrolera. 30

Slika 8.2. PIC 16F628A 32

Slika 9.1. Izgled editora MicroCode Studio 34

Slika 9.2. Dijagram toka funkcija unutar mikrokontrolera za pobudu jednog koračnog motora 35

Slika 10.1. Primjer koračnih motora 39

Slika 10.2. Podjela koračnih motora 41

Slika 10.3. Jednofazna pobuda za zakretanje četverofaznih koračnih motora 42
Slika 10.4. Principijelna shema pogonskog spoja 44
Slika 11.1. Trapezno vreteno 45

Popis tablica

Tablica 5.1. Standardni raspored USB utičnica 18

Tablica 5.2. Raspored Mini USB utičnica 18

Tablica 5.3. Brzine prijenosa podaka 18

Tablica 7.1. adresa paralelnog porta 25

Tablica 7.2 – Opis bitova data registra paralelnog porta 27

Tablica 7.3 – Opis bitova status registra paralelnog porta 27

Tablica 7.4 – Opis bitova control registra paralelnog porta 28

Tablica 10.1. Logički slijed dvofazne simultane uzbude namotaja unipolarnog četverofaznog koračnog motora 42

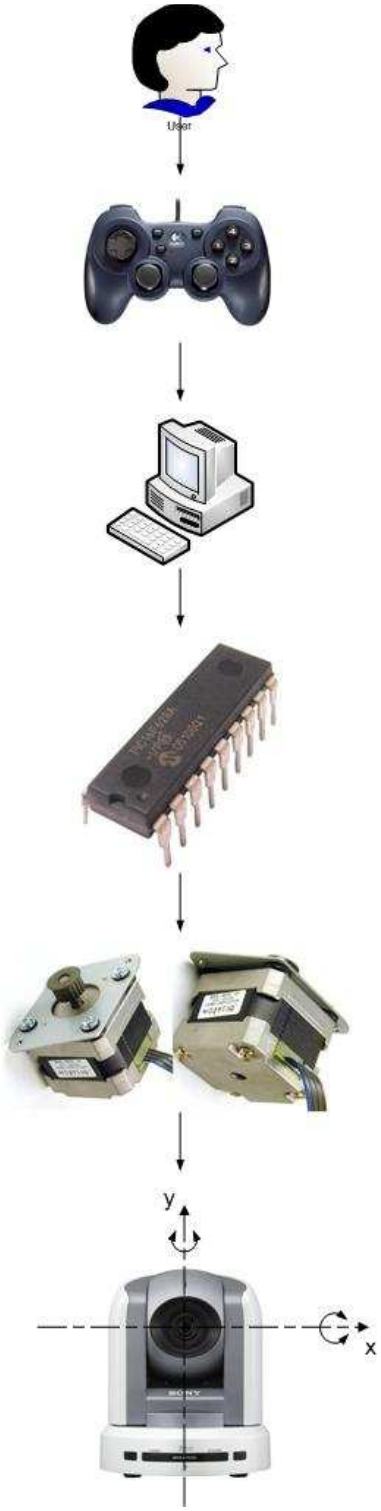
Tablica 10.2. Logički slijed mikrokorak (half-step) uzbude namotaja unipolarnog četverofaznog koračnog motora 43

1. UVOD

Ovaj rad obrađuje temu pozicioniranja kamere u ravnini u svrhu zamjene kamermana u studiu za snimanje, čime bi se pojednostavio i olakšao rad manjih studija. Rad opisuje sve segmente potrebne za izradu makete koja bi prikazala principijelni prikaz pozicnioniranja kamere u ravnini.

Izrađena maketa je sastavljena od 6 glavnih dijelova: igračeg kontrolera, osobnog računala, mikrokontrolera, koračnih motora, vijčanog pogona i kamere te opisuje protokole kojima ti dijelovi komuniciraju. Programiranje aplikacije koja obrađuje signal igračeg kontrolera i postavlja izlaze paralelnog porta pisana je u skriptnom jeziku AutoIt koji je baziran na BASIC-ovoj sintaksi dok je programiranje programskog kôda za mikrokontroler pisano u MicroCode Studio-u, također baziranom na BASIC-ovoj sintaksi. Mikrokontroler je programiran na razvojnom sistemu Mikroelektronike “EASYPIC 6” gdje je prvotno i testiran.

2. PRINCIPIJELNI PRIKAZ RADA MAKETE



Operater

Igrači kontroler

Osobno računalo

Sučelje

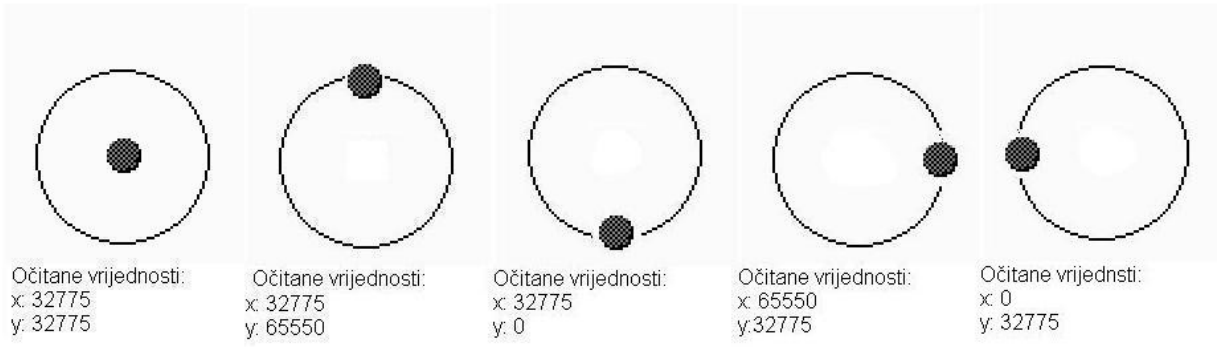
Koračni motori

Kamera

Slika 2. Principijelni prikaz rada makete

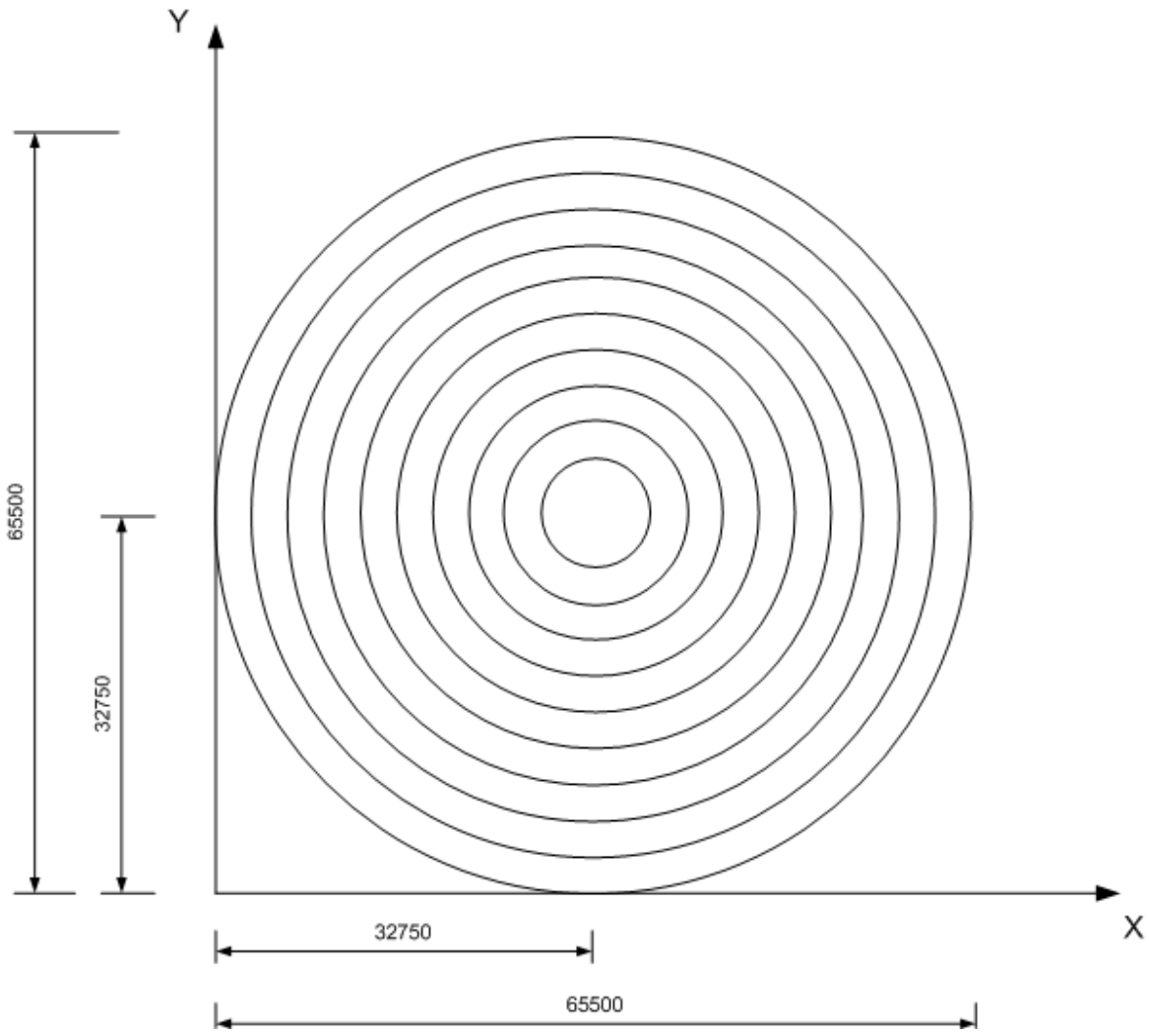
3. OPIS RADA MAKETE

Princip rada makete je prilično jednostavan: operater upravlja igračim kontrolerom koji šalje signal osobnom računalu preko USB porta. Osobno računalo očitava vrijednosti sa analognog dijela igračeg kontrolera te ih sprema u memoriju. Očitana vrijednost može biti između 0 i 65500 za x i y smjer ovisno o svojoj poziciji što je prikazano na slici 2.1.



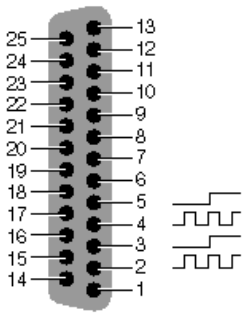
Slika 3.1. Prikaz pozicija analognog dijela igračeg kontrolera sa očitanim vrijednostima

Zatim se očitana vrijednost uspoređuje sa referencom koja je napravljena kako bi se dobio veći broj impulsa na paralelnom portu ovisno o većem kutu nagiba analognog dijela igračeg kontrolera. Referenca je definirana tako da je svaka os podijeljena u rezoluciji po 5% što je prikazano na slici 2.2. s time da je oko nultočke (inače koordinata 32750, 32750) postavljena veća tolerancija (10%). Ukoliko se nebi postavila ta tolerancija oko nultočke, zbog nepreciznosti izrade igračeg kontrolera ta vrijednost se u rijetkim slučajevima vrača u isti položaj, moglo bi doći do neželjnih pomicanja motora.



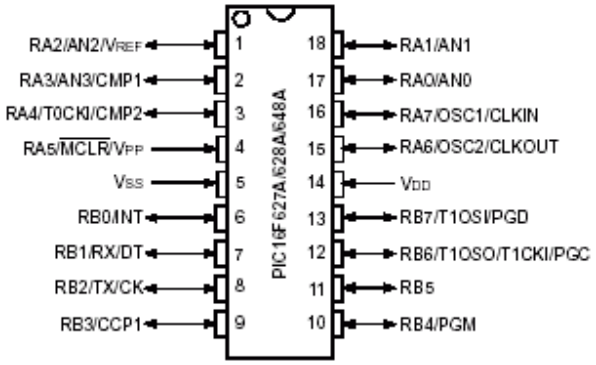
Slika 3.2. Referenca analognog dijela igračeg kontrolera

Obzirom da je potrebno kontrolirati 2 motora, korištena su 4 izlaza iz paralelnog porta i to za svaki motor po 2 izlaza. Jedan izlaz iz paralelnog porta (pin 2) služi kao izvor pravokutnog impulsa koji ujedno označava potrebnu promjenu koraka dok drugi izlaz (pin 3) označava smjer vrtnje motora i on može biti postavljen u stanje logične nule ili logičke jedinice. Što je veći kut zakreta analognog dijela igračeg kontrolera to će broj impulsa na izlazu iz paralelnog porta biti veći pa će se tako i motor brže okretati.



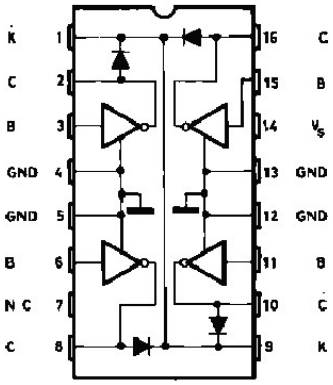
Slika 3.1. Prikaz izlaznih napona iz paralelnog porta

Na paralelni port je spojeno sučelje koje se sastoji od tri regulatora napona i mikrokontrolera. Jedan regulator napona služi za napajanje samog mikrokontrolera dok druga dva služe kao izlazni regulatori (tj. pretvara ulaznih +5V iz mikrokontrolera u izlaznih +24V za pogon motora). Mikrokontroler je isprogramiran tako da broji impulse na ulaznom portu (pin RB.0) koji je ujedno spojen na izlazni pin (pin2) iz paralelnog porta te tako postavlja svoje izlazne portove (pinovi RA.0-RA.3) u stanja logičkih jedinica i logičkih nula ovisno o izbrojanom broju ulaznih impulsa. Ukoliko bi ulazni port (pin RB.1), koji je spojen na izlazni pin paralelnog porta koji označava smjer vrtnje, bio postavljen u stanje logičke jedinice, tada bi program u mikrokontroleru oduzimao broj impulsa te postavljao izlaze u obrnuti smjer vrtnje. Opširnije o načinima uzbuđivanja koračnih motora opisano je u poglavlju 10. Koračni motor. Isti princip vrijedi i za drugi motor samo što se ovaj puta povezuju izlazi paralelnog porta (pin4 i pin5), ulazi na mikrokontroler (RB.2 i RB.3) i izlazi iz mikrokontrolera (RA.4 - RA.7). Prikaz portova korištenog mikrokontrolera PIC16F628A prikazan je na slici 2.3 a detaljnije o tom mikrokontroleru se može vidjeti u podatkovim tablicama mikrokontrolera u prilogu.



Slika 3.3. Portovi korištenog mikrokontrolera PIC16F628A

Izlazi iz mikrokontrolera (RA.0 – RA.7) spojeni su na 2 regulatora napona ULN2065B koji u sebi ima 4 Darlingtonova spoja. Oni rade tako da propuštaju uzemljenje na izlaznom pinu (pin 2, pin 7, pin 9, pin 16) ukoliko se na ulazni pin (pin 3, pin 6, pin 11, pin 14) dovede stanje logičke jedinice (+5V). S obzirom da su za izradu makete potrebna 2 motora (8 Darlingtonovih spojeva) potrebno je koristiti 2 regulatora napona (svakom po 4 Darlingtonova spoja). Prikaz ulaznih i izlaznih pinova prikazan je na slici 2.4. a detaljnije o tom regulatoru napona se može naći u podatkovnim tablicama u prilogu.



Slika 3.4. Shematki prikaz naponskog regulatora ULN2068B

Koračni motor na svakom regulatoru napona spojen je na njegova 4 izlaza te s obzirom da regulator napona može propustiti maksimalno 1.5A a motor troši 0.7A je i više nego dovoljno struje motoru za uzbudu. Pokretanje koračnih motora je opisano u poglavlju 10. Koračni motor a karakteristike korištenog motora se nalaze u podatkovnim tablicama u prilogu.

4. IGRAČI KONTROLER

Igrači kontroler ili gamepad (također zvan joypad ili control pad) vrsta je ulazne jedinice za igraće konzole i osobna računala, koja služi za upravljanje u videoigrama i koja se drži s obje ruke. Najčešće gamepad sadrži tipke za kretanje (gore, dolje, lijevo, desno, tzv. *D-pad*) te tipke na desnoj strani gamepada (obično označene slovima A, B, X i Y, znakovima ili brojevima). Dodatne tipke (*start*, *select* ili *mode*) obično su postavljene na središnjem ili stražnjem dijelu gamepada, dok se kod nekih kontrolera tipke nalaze i na gornjem dijelu (tzv. *shoulder buttons*, primjerice R1/R2 i L1/L2 kod kontrolera za PlayStation).

Gotovo svi moderniji upravljači imaju i analognu palicu sličnu igraćoj palici (joysticku), ali mnogo manjih dimenzija. Prvi se put analogna palica pojavila 1982. godine na igraćoj konzoli Arcadia 2001, američke tvrtke Emerson Radio, dok će se s vremenom takva vrsta upravljača pojaviti na svim gamepadovima poznatih igraćih konzola.

Korišteni gamepad na maketi je Logitec Dual Action.

Slika 4.1. Upravljački igrači kontroler

Slika 4.2. Analogni dio upravljačkog igračeg kontrolera

4.1. Opis

Kontroliranje

- Svestrano kontroliranje: Posjeduje 12 programibilnih tipki (uključujući 4 prednje tipke)
- Analogne preformanse: Preciznost iz dvije analogne palice s digitalnim gumbima i glatko, precizno djelovanje.
- Logitech® Profiler software (PC): pomoću programske podrške omogućeno je personalizirano programiranje svih tipki uz višeprofilnu mogućnost

Praktičnost

- *Plug-and-play* setup: Jednostavno ga uključite i počnite igrati.

5. USB PORT

USB je skraćenica za *Universal Serial Bus* i predstavlja tehnološko rješenje spajanja vanjskih uređaja s računalom. Podaci se razmjenjuju serijski, a brzina razmjene podataka za inačicu USB 2.0 iznosi 480 Mbps.

Slika 4.1. Standardni USB utikač

5.1. Inačice

- USB 1.0 FDR
- USB 1.0 objavljen u siječnju 1996.
- USB 1.1 objavljen u rujnu 1998.
- USB 2.0 objavljen je u travnju 2000.
- USB 3.0 objavljen u studenom 2008.

5.2. Način rada

Cilj USB tehologije jest rasterećivanje glavne sabirnice računala od posebnih kartica za proširenje, kao i olakšavanje umetanja i odvajanja vanjskih uređaja *(plug-and-play)* bez potrebe za ponovno pokretanja računala *(hot swapping)*.

USB je asimetričnog dizajna i sastoji se od kontrolera poslužitelja, kao i mnogostrukih jedinica koje se uključuju na poslužitelj kao grane preko posebnih uređaja *(hub)* i tako stvaraju stablasti oblik. Kod USB-a je moguće imati samo 5 nivoa grananja po svakom kontroleru poslužitelju, te je moguće priključiti 127 uređaja, umanjeno za svaki hub koji je priključen na isti USB poslužitelj.

5.3. USB signaliranje

Tablica 5.1. Standardni raspored USB utičnica

Pin	Funkcija
1	V_{BUS} (4,4 – 5,25 V)
2	D−
3	D+
4	Uzemljenje

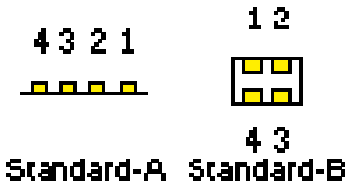
Tablica 5.2. Raspored Mini USB utičnica

Iglica	Funkcija
1	V_{BUS} (4,4 – 5,25 V)
2	D−
3	D+
4	ID
5	Uzemljenje

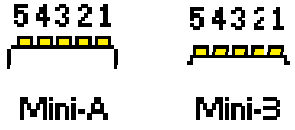
5.4. Brzine USB uređaja

Tablica 5.3. Brzine prijenosa podaka

Nazivne brzine prijenosa podataka		
Oznaka	**Brzina prijenosa**	**Podržano od verzije standarda**
Low Speed	1.5 Mbit/s	USB 1.0
Full Speed	12 Mbit/s	USB 1.0
HiSpeed	480 Mbit/s	USB 2.0
SuperSpeed	4.8 Gbit/s	USB 3.0



Slika 5.2. Standardna USB A i B utičnica



Slika 5.3. Mini USB A i B utičnica

6. AUTOIT

AutoIt v3 je besplatni skriptni jezik napravljen na bazi jednostavnog BASIC-a a sluzi za automatizaciju Windows GUI i općenitog skriptiranja.
Koristi kombinaciju simuliranih tipki, pokreta mišem i manipulaciju prozora/kontrola kako bi automatizirao zadatke koje nije moguće pouzdano napraviti u drugim jezicima (npr. VBScript, SendKeys…)
AutoIt također zauzima vrlo mali dio RAM memorije nakon instaliranja, samostalan je (eng. standalone) i raditi će na svim verzijama sustava.
AutoIt je prvotno dizajniran za PC "roll out" situacije da pouzdano automatizira i konfigurira tisuće računala. S vremenom je postao moćan jezik koji podržava složene izraze, korisničke funkcije, petlje i sve drugo što bi veterani skriptiranja mogli očekivati.

Zanačajke:

- Lako učenje sintakse slične BASIC-ovoj
- Simuliranje tipkanja na tipkovnici i pokretanja miša
- Manipuliranje prozorima i procesima
- Interakcija sa svim standardnim kontrolama prozora
- Skripte mogu biti kompajlirane u samostalne pokretljive programe (*.exe)
- Stvaranje grafičkog korisničkog sučelja (GUI)
- Podrška COM portova
- Regulari izrazi
- Direktno pozivanje vanjske DLL datoteke i Windows API funkcije
- Mogućnost skriptiranja funkcije “*RunAs*”
- Detaljna HELP datoteka i veliki forumi podrške
- Kompatibilan sa Windows 95 / 98 / ME / NT4 / 2000 / XP / 2003 / Vista / 2008
- Unicode i x64 podrška
- Digitalni potpis

- Radi sa Windows Vista's User Account Control (UAC)

AutoIt je dizajniran da bude što manji i što samostalniji bez vanjskih .dll datoteka ili unosa u registar Windowsa (eng. Windows registry) što ga čini vrlo sigurnim za rad sa serverima. Skripta moze biti kompajlirana u samostalan pokretljiv program (*.exe) sa Aut2Exe.

Također dolazi as kombiniranom COM i DLL verzijom AutoIt-a zvanom AutoItX koji omogućava dodavanje jedinstvenih značajka AutoIt-a u neki drugi skriptni ili programski jezik.

6.1. Značajke AutoIT-a u detaljima

6.1.1. Sintaksa slična BASIC-ovoj i bogat izbor funkcija

AutoIt ima sintaksu sličnu BASIC-ovoj što znači da većina ljudi koji su ikad napiasli skriptu ili se koriste naprednijim programskim jezikom nebi trebali imati poteškoća sa programiranjem u AutoIt-u.

Iako je njegova početna funkcija bila jednostavan alat za automatizaciju, AutoIt sada ima toliko funkcija i značajki da se može koristiti kao općeniti skriptni jezik. Značajke jezika uključuju:

- Uobičajeni elementi za visoki nivo programiranja (funkcije, petlje, rastavljanje izraza…)
- Zapanjujući iznos niza funkcija za rukovanje
- COM podrška
- Poziv na Win32 i DLL API

6.1.2. Ugrađen editor sa označavanjem sintakse

AutoIt dolazi sa prilagođenom *“lite”* verzijom SciTe koji olakšava skriptiranje. Korisnici mogu preuzeti cijelu verziju SciTe koji posjeduje više alata za još jednostavniju uporabu.

6.1.3. Međunarodna i 64-bitna podrška

AutoIt je potpuno svjestan Unicode-a i također podržava x64 verzije svih glavnih komponenata

6.1.4. Simulacija tipkovnice i miša

Mnogo vremena je potrošeno u optimizaciji funkcija simulacije tipki tastature i pokreta mišem da budu što preciznije za sve verzije Windowsa. Sve rutine miša i tipkovnice su visoko podesive i u pogledu simulacije brzine i u pogledu funkcionalnosti.

6.1.5. Prozorski menadžment

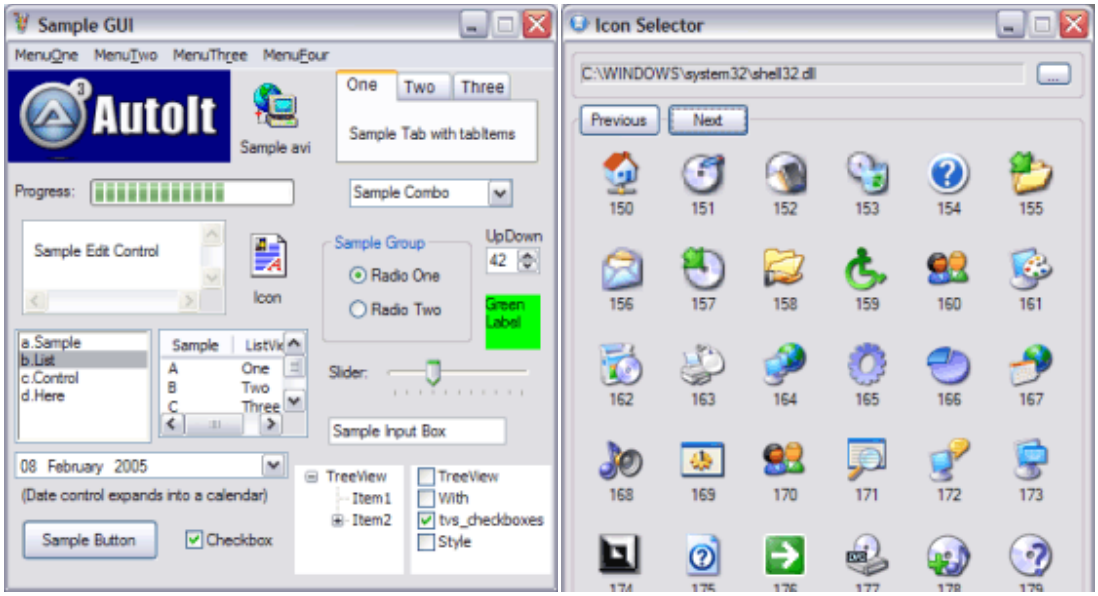
Moguće je pomicati, skrivati, pokazivati, mijenjati veličinu, aktivirati i zatvarati prozor. Prozor moze biti definiran sa naslovom, tekstu na prozoru, veličini, položaju, klasi, pa čak i unutarnjim Win32 API-em.

6.1.6. Kontrole

Direktno se mogu dobiti informacije sa editorskim okvirima, potvrdnim okvirima, popisnim okvirima, kombiniranim okvirima, funkcijskim tipkama, statusnom trakom bez rizika za gubljenje simulacije tipkovnice. Čak je moguće kontroliranje prozora koji nisu aktivni.

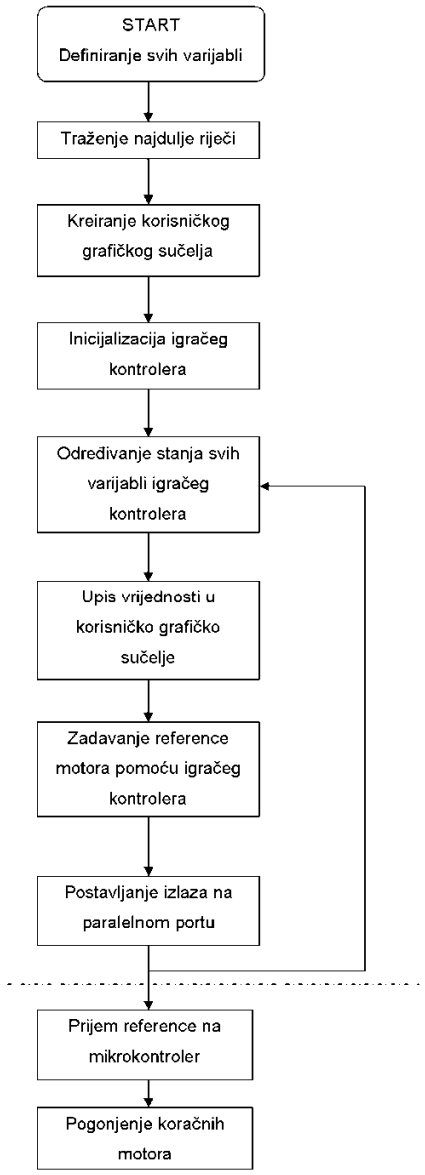
6.1.7. Grafičko korisničko sučelje (GUI)

U AutoIt-u je moguća izrada kompliciranih grafičkih sučelja kao što je prikazano na slici ispod:



Slika 6.1. Izgled grafičkog korisničkog sučelja

6.2. Objašnjenje načina rada skriptnog kôda



Slika 6.2. Dijagram toka funkcija aplikacije za inicijalizaciju igračeg kontrolera i postavljanja izlaza parlelnog porta

Prikazani dijagram toka funkcija aplikacije za inicijalizaciju igračeg kontrolera i postavljanja izlaza paralelnog porta opisuje rad aplikacije te objašnjava ulogu osobnog računala u ovom projektu. Igrači kontroler pomoću USB porta šalje vrijednosti svih svojih tipki osobnom računalu koje obrađuje dvije vrijednosti – koordinate analognog dijela igračeg kontolera. Ovisno o kutu zakreta analognog dijela i u kojem smjeru, računalo uspoređuje dobivene vrijednosti sa referencom koja je nacrtana u poglavlju 3. Opis rada makete na slici 3.2. Referenca analognog dijela igračeg kontrolera. Referenca analognog dijela je postavljena kako bi se dobio dojam analognosti odnosno kako bi se većim zakretom kuta moglo dobiti više impulsa na izlazu na paralelnom portu što bi rezultiralom bržim pomicanjem okretanjem koračnih motora a samim time i kamere. Nakon što se dobivena vrijednost sa igračeg kontrolera usporedila sa referencom, određuje se trajanje pozitivne i negativne poluperiode izlaznog signala i to tako da se pomakom prema vanjskom referentnom krugu koje označava krajnje vrijednosti vrijeme smanjuje. Tako se dobiva više impulsa u kraćem vremenu, koračni motor se brže zakreće i kamera se brže pomiće po osi ravnine.

6.2.1. Opis naredbi u skripnom kôdu

Skriptni kod se bazirao na rad sa .dll datotekama i baš zato se AutoIt pokazao vrlo zahvalnim skriptnim programom za pisanje ovakve vrste aplikacije. Korištene .dll datoteke su “Winmm.dll” i “input32.dll”.

“Winmm.dll” služi za određivanje varijabli stanja na igračem kontroleru te za zapisivanje tih varijabli u strukturu nazvanu “$lpJoy”.

Primjer u programu:

```
DllCall("Winmm.dll","int","joyGetPosEx", "int",$iJoy, "ptr",DllStructGetPtr($lpJoy))
```

„input32.dll“ služi za upravljanje raznim portovima osobnog računala dok je u ovom projektu iskorišten za postavljanje izlaza na paralelnom portu.

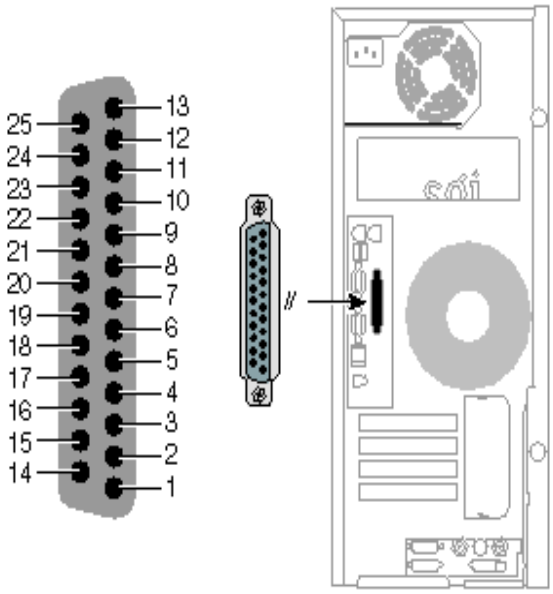
Primjer u programu:

```
DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
```

7. PARALELNI PORT

Paralelni port PC-a spada u najvažnije portove preko kojeg računala komuniciraju sa periferijom. Na ovaj port prije svega se priključuju printeri ali je on dostupan i drugim uređajima koji pravilno koriste njegove signale. Uređaji – sučelja koji se priključuju na ovaj port moraju biti programski podržani.

Paralelni port je prvi puta upotrijebio IBM na svojim PC XT računalima i bio je namijenjen za komunikaciju sa printerima. Taj standard je nazvan „Centonics printers interface“ i dugo se koristio kao takav. Kasnije je IEEE izdala standard 1284 koji definira 5 načina rada paralelnog porta.



Slika 7.1. Izgled paralelnog porta

- Kompatibilni način (Compatibility mode) koji je standardni način rada paralelnog porta poznat pod oznakom SPP
- Niblle način (Niblle mode) koristi 4 bita podataka na ovom portu i često ga koriste optički čitači
- Byte način (Byte mode) koji je uveo IBM uz svoj PS/2 standard i omogućio slanje jednog byte te se rijetko koristi
- EEP način (Enhanced Parallel Port Mode) je uveo dvosmjerni prijenos podataka preko paralelnog porta

- ECP način (Extended Capabilities Port Mode) omogućuje 8 bitni asinkroni prijenos uz upotrebu DMA što mu omogućuje velike brzine rada i danas se često koristi

SPP način rada omogućuje brzine prijenosa 50 do 150 kB/s dok ECP i EPP načini omogućuje brzine prijenosa i do 1 MB/s. 1284 standard uveo je dva nivoa kompatibilnosti sučelja i to tvz. I i II novo pri čemu se II nivo koristi za veće brzine prijenosa (ECP i EPP). Glavne karakteristike prvog nivoa su sljedeće:

- Visoki naponski nivo (logička jedinica) ne smije biti veća od +5.5V
- Niski naponski nivo (logička nula) ne smije biti manja od -0.5V
- Normalni visokonaponski nivo treba biti najmanje +2.4V uz struju od 14mA
- Normalni niskonaponski nivo ne smije biti veći od +0.8V uz struju od 14mA
- Otpor uređaja Ro priključenog na konektor treba biti 50±5 Ω na naponu od ˝ razlike napona sučelja visokog i niskog nivoa
- Vremenske promjene napona mora biti podržana u granicama 0.05-0.40 V/ns

Paralelni port PC računala se kontrolira preko tri 8-bitna memorijska registra sa sljedećim nazivima:

- Data port
- Control port
- Status port

Data port sadrži jedan byte podataka koji će se naći na izlaznim linijama paralelnog porta pod uvjetom da su zadovoljeni određeni zahtjevi. Control port kontrolira operaciju slanja podataka, dok Status port ukazuje na stanje uređaja.

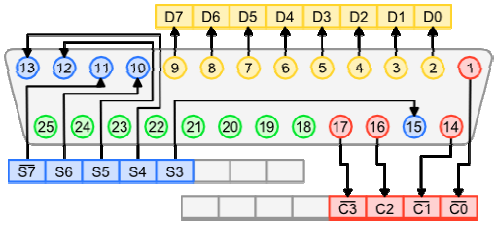
Tablica 7.1. adresa paralelnog porta

Port	Data Port	Status Port	Control port
LPT1	378h	379h	37Ah
LPT2	278h	279h	27Ah
LPT3	3BCh	3BDh	3BEh

Oznaka h uz adresu znači da je adresa izražena u heksadecimalnom brojevnom sustavu. U slučaju da je adresa Data porta nepoznata onda se ona može pronaći na sljedeći način:
Uđemo u MS Dos prompt
Upišemo naredbu DEBUG
Upišemo d 0040:0008 L6 i dobit ćemo sadržaj šest byta počevši od adrese 408h. Tada se dobije rezultat
debug
-d 0040:0008 L6
0040:0080 78 03 78 02 00 00

Iz sadržaja prikazanih byte-a možemo pročitat adrese Data porta LPT1 pri čemu se uvijek prikazuju niže pa više byte adrese.

Paralelni port ima izlaz preko ženskog DB-25 konektora. Signali na pojedinim pinovima ovog konektora prikazani su da donjoj slici.



Slika 7.2. Adresiranje paralelnog porta

U donjim tablicama ukratko su opisani pojedini pinovi i njihovi signali

Tablica 7.2 – Opis bitova data registra paralelnog porta

Signal	Pin	Opis	Vrsta Signala
D0	2	1. bit byta podataka	Izlazni
D1	3	2. bit byta podataka	Izlazni
D2	4	3. bit byta podataka	Izlazni
D3	5	4. bit byta podataka	Izlazni
D4	6	5. bit byta podataka	Izlazni
D5	7	6. bit byta podataka	Izlazni
D6	8	7. bit byta podataka	Izlazni
D7	9	8. bit byta podataka	Izlazni

Tablica 7.3 – Opis bitova status registra paralelnog porta

Signal	Pin	Opis	Vrsta Signala
S0		Ne koristi se	
S1		Ne koristi se	
S2		Ne koristi se	
S3	15	Impuls niskog nova koji signalizira grešku	Ulazni
S4	13	Potvrđuje prisutnost printera	Ulazni
S5	12	Paper Empty	Ulazni
S6	10	ACK – Potvrdi impuls	Ulazni
S7	11	Busy – Printer zauzet	Ulazni

Tablica 7.4 – Opis bitova control registra paralelnog porta

Signal	Pin	Opis	Vrsta Signala
C0	1	Strobe	Izlazni
C1	14	Auto Linefeed	Izlazni
C2	16	Initialize Printer (Reset)	Izlazni
C3	17	Select Printer	Izlazni
C4		Enable IRQ Via Ack Line	Izlazni
C5		Enable bi-directiona Port	Izlazni
C6		Ne koristi se	
C7		Ne koristi se	

Signali koji iznad svog naziva imaju negaciju standardno se nalaze na logičkoj jedinici dok su ostali na logičkoj nuli. U slučaju programskog slanja logičke nule na invertirane pinove oni će biti postavljeni na logičku jedinicu i obrnuto. Pinovi 18-25 na DB-25 konektoru koriste se za masu (ground). Iz prikazanih tablica može se vidjeti da svi bitovi na Control i Status portu nisu izvedeni na DB-25 konektoru, jer se oni programski kontroliraju.

8. MIKROKONTROLERI

8.1. Općenito

Mikrokontroler (microcontroller) je elektronički uređaj koji, slično kao i računalo, ima zadaću da zamjeni čovjeka u kontroli dijela proizvodnog procesa ili gotovo cijelog proizvodnog procesa.

Iz svakodnevne prakse pri uporabi već je poznato da je standardni ulaz tipkovnica i miš (eventualno igrača palica), dok je standardni izlaz monitor ili pisač. Teško je definirati što će biti standardni ulaz i izlaz mikrokontroleru. Razlog tome je što su mikrokontroleri uglavnom dizajnirani za specifične zadaće vrlo raznolike od slučaja do slučaja. Primjera ima mnogo, od jednostavne regulacije osvjetljenja, alarmnih sustav, pa do upravljanja robotima u industrijskim pogonima.

Ulazi mogu biti vrlo jednostavne izvedbe kao na primjer prekidač u sklopu plovka za nadzor najvećeg ili najnižeg nivoa tekućine u spremniku. Mikrokontroler tada ima za obradu samo dva stanja koje opisuje jedan bit. Složenije je praćenje ako treba pratiti stvarnu razinu nivoa tekućine u spremniku. Tada treba definirati koliko će se nivoa pratiti i tu će se uporabiti nekakav potenciometarski sklop koji će mikrokontroleru predati određenu analognu vrijednost koju će ovaj potom pomoću A/D pretvornika obraditi i isporučiti odredištu. Ako je pak povezan s fotoćelijom za brojanje predmeta po načelu prekidanja svjetlosnog snopa radi se o izravnom brojanju impulsa tijekom rada neovisno o vremenu.

Dakle, ulazi mogu biti analogni i digitalni i u suštini podatke će isporučivati nekakav mjerni pretvornik (senzor).

Izlazi iz mikrokontrolera također mogu biti analogne i digitalne prirode. Analogni izlazi, bilo naponski ili strujni, mogu se elektromehaničkim sklopovima pretvoriti u neku korisnu radnju kao promjena položaja nekog predmeta, povećanje brzine vrtnje motora i slično. Najjednostavniji primjer je lampica upozorenja koja upozorava čovjeka na promjenu ili neispravnost. Složeniji izlaz biti će kada se želi pratiti veličina promjene bilo kao analogni ili digitalni prikaz.

Naravno, ulaz i izlaz mikrokontrolera nije isključivo vezan na komunikaciju sa strojem. Vrlo rijetko kontroler nema neki vid komunikacije prema korisniku, na primjer s

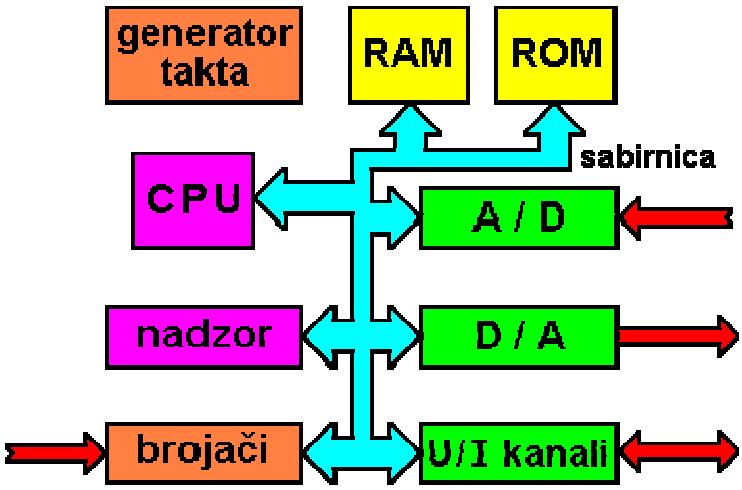
lampicama ili s digitalnim pokazivačem. No nisu rijetki slučajevi da se za komunikaciju sa čovjekom koristi računalo tipa PC.

Iz navedenog može se zaključiti da se mikrokontroleri prema načinu izrade i komunikacije s okolišem mogu svrstati u jednu od dvije osnovne kategorije:

Mikrokontroler kao samostalna upravljačka jedinica

Mikrokontroler kao osobita kartica u jednom od utora PC računala

U suštini mikrokontroler radi na načelu vrlo bliskom računalu. On je u istinu malo računalo, a složenost mu ovisi o složenosti zadaće koju ima nadzirati. Općenito blok shema mikrokontrolera mogla bi izgledati prema narednoj slici.



Slika 8.1. Načelna blok-shema mikrokontrolera.

Koje će sve elemente sadržavati i koliko moćne ovisiti će o njegovoj namijeni. CPU je jednostavniji od sklopova namijenjenih PC konfiguracijama, obično nekad popularni Zilog-Z80 ili neki iz porodice INTEL-ovih procesora. Svima su zajednička sljedeća svojstva:

- Relativno mali radni takt reda 10MHz
- Mali broj jednostavnih instrukcija, red veličine oko 100
- Radna memorija (RAM) reda KB
- Stalna memorija s programskim kodom u PROM ili EPROM izvedbi

- Brojači različitih namjena kao sat, brojač impulsa, BCD brojač i drugi
- Brojač za nadzor ispravnog rada - WDT (Watch Dog Timer)
- Ulazno/Izlazni kanali (port-ovi) za prihvat i slanje podataka
- A/D i D/A pretvornici razlučivosti prema namjeni, uobičajeno 8 bit-ni
- Širok raspon napona napajanja

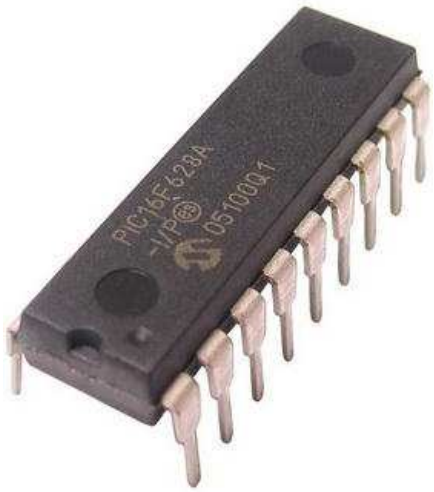
Svi navedeni elementi ne moraju biti nužno zastupljenu u mikrokontroleru. Od namjene mikrokontrolera ovisiti će njegov izbor te će jedni imati više U/I port-ova, a drugi više multipleksiranih A/D pretvornika i slično. Intel-ova serija mikrokontrolera MSC-48 ima 8 osnovnih predstavnika različitih po mogućnostima. Osim navedene postoji još nekoliko serija istog proizvođača. Ako se tome pribroje i ostali proizvođači (Motorola, Microship Technology, Siemens i drugi) izbor mikrokontrolera više je nego dovoljan.

Od značaja je WDT sustav nadzora. To je specijalno dizajnirano BROJILO. Ako program upisan u memoriju mikrokontrolera ispravno radi on će povremeno u prikladnim vremenski razmacima (do 100ms) vratiti kontrolno brojilo na početnu vrijednost. Ako pak brojilo u svom radu odbroji do kraja, zadnjim brojem stavlja na znanje CPU mikrokontrolera da je došlo do nepravilnosti u radu programa jer nije bio vraćen na početnu vrijednost. CPU pokreće programsku rutini za ponovnu inicijalizaciju mikrokontrolera ili se odvija neka druga predviđena zaštitna radnja. Na taj način onemogućava se duži nepravilan rad mikrokontrolera bez obzira na razloge uzroka nepravilnosti. Ova metoda nadzora višestruko povećava sigurnost sustava kojeg mikrokontroler nadzire.

Obično se u sklopu uređaja s mikrokontrolera nalazi i mala baterija kojoj vijek trajanja doseže i do 10 godina. Zajedno sa sustavom WDT za nadzor ispravnog rada mikrokontrolera, proizlazi da će podaci u RAM-u uvijek biti očuvani te da za redovnim servisom gotovo da nema potrebe. Naravno, baterija zamjenjuje i napajanje cjelokupnog sklopa u slučaju nestanka struje, slično kao što rade on-line UPS uređaji.

8.2. Mikrokontroler PIC 16F628A

Korišteni mikrokontroler pri izradi makete je PIC 16F628A njegove karakteristike su u prilogu. Programiran je na razvojnom sistemu Mikroelektronike „EasyPic6“ a program je pisan u MicroCode Studio-u (editor za PicBasic Pro).



Slika 8.2. PIC 16F628A

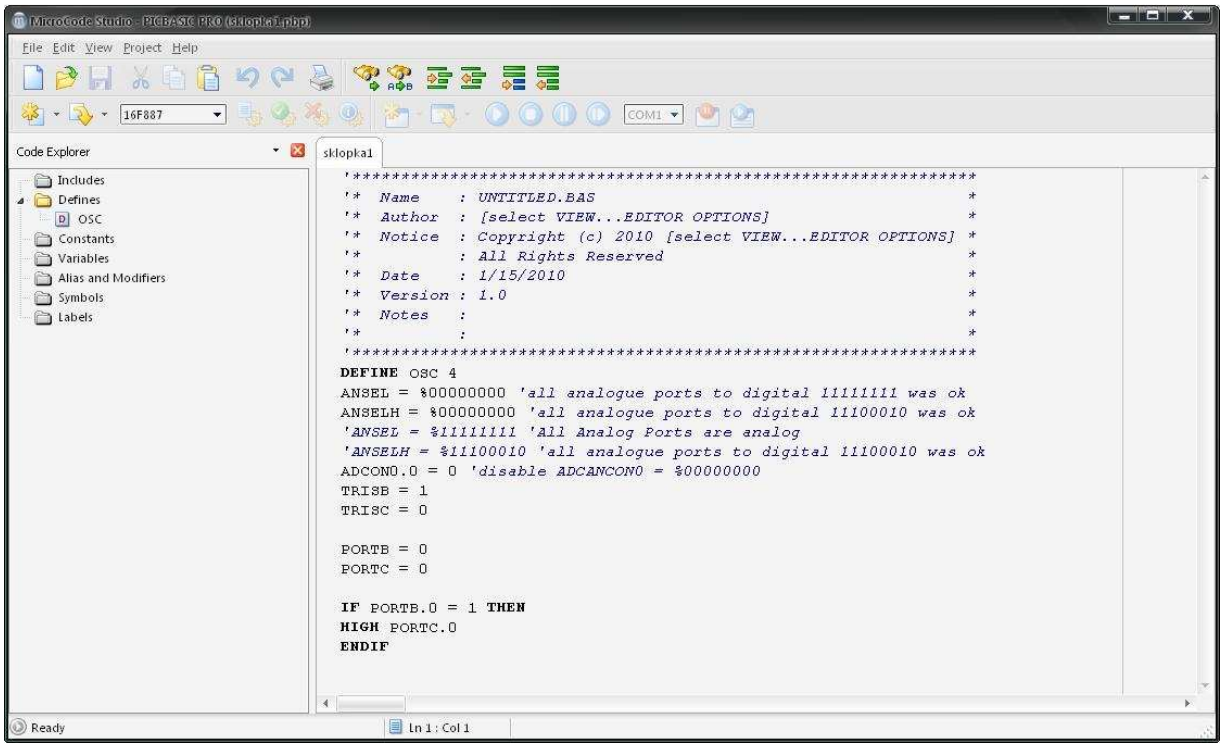
9. MICROCODE STUDIO

MicroCode Studio je vizualna integrirana razvojna okolina (IDE) sa mogučnošću traženja pogreške u krugu dizajnirana specijalno za microEngineering Labs PICBASIC™ i PICBASIC PRO™ kompajler.
Glavni editor pruža potpuno označavanje sintakse napisanog kôda koji je kontekstno osjetljiv na ključnu riječ iz Help-a te dodaje savjet kako nastaviti sintaksu. Pretraživač kôda omogućava korisniku da se automatski prebaci na uvrštene datoteke, definicije, konstante, varijable, pseudonime i modifikatore, simbole i natpise, koji su sadržani unutar svog izvornog kôda. Također su podržane funkcije izreži, kopiraj, zalijepi, poništi i “pronađi i zamjeni”.

MicroCode Studio sadrži EasyHID Čarobnjak, besplatan alat za generiranje kôda što omogućava korisniku brzu implementaciju dvosmjerne komunikacije između ugrađenog PIC mikrokontrolera i osobnog računala.

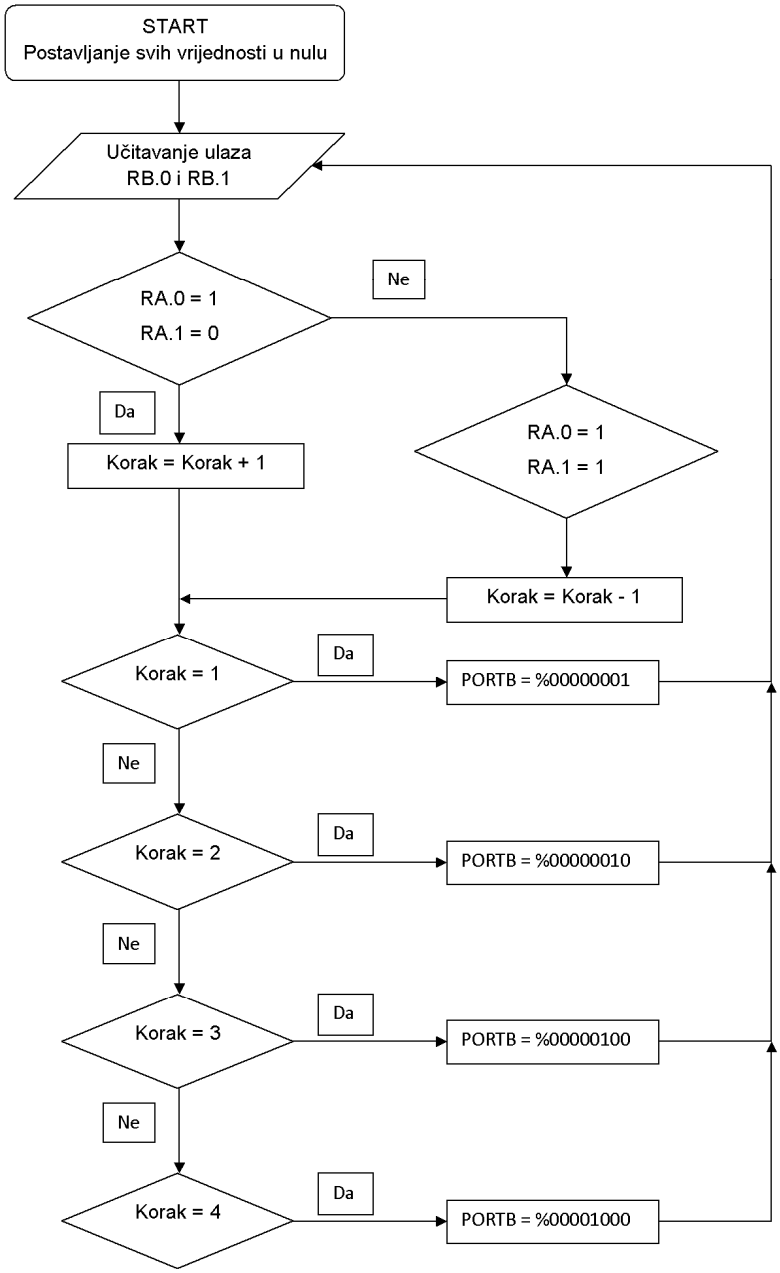
Jednostavno je postaviti kompajlerske, assemblerske i programatorske opcije ali moguće je dopustiti MicroCode Studio-u da odradi automatsko traženje postavki sa svojom integriranom funkcijom.

Kompajlerske i asemblerske greške je moguće lako identificirati i ispraviti koristeći prozor za pogreške. Samo jedan klik na kompajliranu grešku i MicroCode Studio će ovesti korisnika na pogrešan redak kôda.
Prozor za serijsku komunikaciju je također integriran što omogučuje ispravljanje pogreške i pregled serijskog izlaza iz ugrađenog mikrokontrolera.



Slika 9.1. Izgled editora MicroCode Studio

9.1. Objašnjenje načina rada programskog kôda



Slika 9.2. Dijagram toka funkcija unutar mikrokontrolera za pobudu jednog koračnog motora

Prikazani dijagram toka funkcija unutar mikrokontrolera (slika 9.2.) opisuje rad programskog kôda kojeg naš mikrokontroler izvršava. Na ulaze mikrokontrolera su spojena 2 para izlaza iz paralelnog porta (u svakom paru jedan ulaz je pravokutni impuls i on označava zahtjev koraka a drugi logička nula ili logička jedinica i označava smjer – lijevo ili desno). Ovisno o stanju ulaza, mikrokontroler određuje u koju stranu motor treba voditi te tako zbraja ili oduzima korake logičkog slijeda i postavlja svoje izlaze u stanja logičkih nula i logičkih jedinica u odnosu na broj koraka koje je izbrojio. Logički slijed načina pobude koračnog motora opisan je više u poglavlju 10. Koračni motor. Identičan dijagram toka funkcija se primjenjuje za drugi koračni motor samo što se za drugi motor koriste različita dva ulazna odnosno četiri izlazna pina.

Prilikom pisanja programa za navedeni mikrokontroler korištene su naredbe za početno definiranje i operacijske naredbe.

9.1.1. Naredbe za početno definiranje:

DEFINE OSC – ovom naredbom se definira taktna brzina procesora
Primjer u programu:
DEFINE OSC 8 %taktna brzina 8 MHz

CCP1CON – ovom naredbom se definira primanje/uspoređivanje režim rada
Primjer u programu:
CCP1CON = 0 %isključen je režim rada primanje/uspoređivanje

CMCON - ovom naredbom se definira komparator
Primjer u programu:
CMCON = 7 %isključen je komparator
OPTION_REG – ovom naredbom se postavlja opcijski registar za čitanje i pisanje koji sadrži razne kontrole bitova za konfiguraciju mjerila brojača, vanjskog prekidanja na RB.0,….

Primjer u programu:

```
OPTION_REG = %10000000      %bit 7 onemogućuje slabe dodatne otpore sa open-kolektorskom strukturom
```

INTCON – ovom naredbom se postavlja prekidački registar za čitanje i pisanje koji sadrži kontrolne bitove i zastavice za sve izvore prekida osim komparatorskog modula.

Primjer u programu:

```
INTCON = 0          %onemogućeni su svi prekidi
```

TRIS – registar kojim se definira da li je port ulaz ili izlaz

Primjer u programu:

```
TRISA = %11111111           %definirali smo cijeli port A ulaznim
TRISB =  %00000000          %definirali smo cijeli port B izlaznim
```

9.1.2. Operacijske naredbe

IF … THEN … ENDIF – ova naredba odlučuje što treba napraviti program ako je varijabla neke vrijednosti. ENDIF označava kraj odlučivanja

Primjer u programu:

```
if (portb.0 = 1) and (portb.1 = 0) then     %ako su ulazi portb.0=1 i portb.1=0
      steps1 = steps1 + 1                   %program uvećava vrijednost varijable za 1
   endif                                    %kraj naredbe
```

SELECT CASE … END SELECT – ova naredba cijelo vrijeme uspoređuje zadanu varijablu sa prije definiranim slučajevima te tako odlučuje što će napraviti ovisno da li je neki slučaj jedna promatranoj varijabli. END SELECT označava kraj uspoređivanja

Primjer u programu:

```
Select case steps1  %definiramo koju varijablu želimo uspoređivati
   CASE 0           %ako je definirana varijabla steps1 jednaka 0
      steps1 = 4            %postavi varijablu steps1 u 4
```

```
case 1
    y = 10
case 2
    y = 6
CASE 3
    y = 5
CASE 4
    y = 9
case 5
    goto start1
END SELECT      %ovdje definiramo kraj uspoređivanja
```

GOTO – naredba koja označava da program treba skočiti na zadanu oznaku

Primjer u programu:

```
GOTO main  %skoči na oznaku „main“
```

Kompletan programski kôd priložen je u prilogu A.5. Programski kôd

10. Koračni motor

10.1. Općenito

Koračni motori su takvi motori kod kojih se položaj rotora može mijenjati samo u diskretnim koracima. Najmanji korak za koji se rotor koračnog motora može pomaknuti naziva se jedinični pomak ili rezolucija (engl. increment)

Slika 10.1. Primjer koračnih motora

Ova vrsta motora omogućava precizno pozicioniranje tereta, a pozicioniranje motora se vrši direktno računalom, mikrokontrolerom ili programibilnim logičkim kontrolerom. Zbog svoje konstrukcije bez četkica, koračni motori su pouzdani, izdržljivi i ne zahtijevaju nikakvo održavanje.

10.2. Upotreba

Koračni motori se koriste u primjenama gdje je precizno pozicioniranje od značaja. To su precizne mašine, roboti, medicinska i naučna oprema, štampači i crtači (ploteri), i stariji modeli računalnih diskova.

10.3. Konstrukcija

Koračni motori imaju stator sa namotajima i neuzbuđeni rotor. Stator ima paran broj jednako razmaknutih polova (zubaca), svaki sa zavojnicom. Nasuprotni parovi statorskih zavojnica su spojeni u seriju, dakle kad je jedan u stanju sjevernog pola, drugi

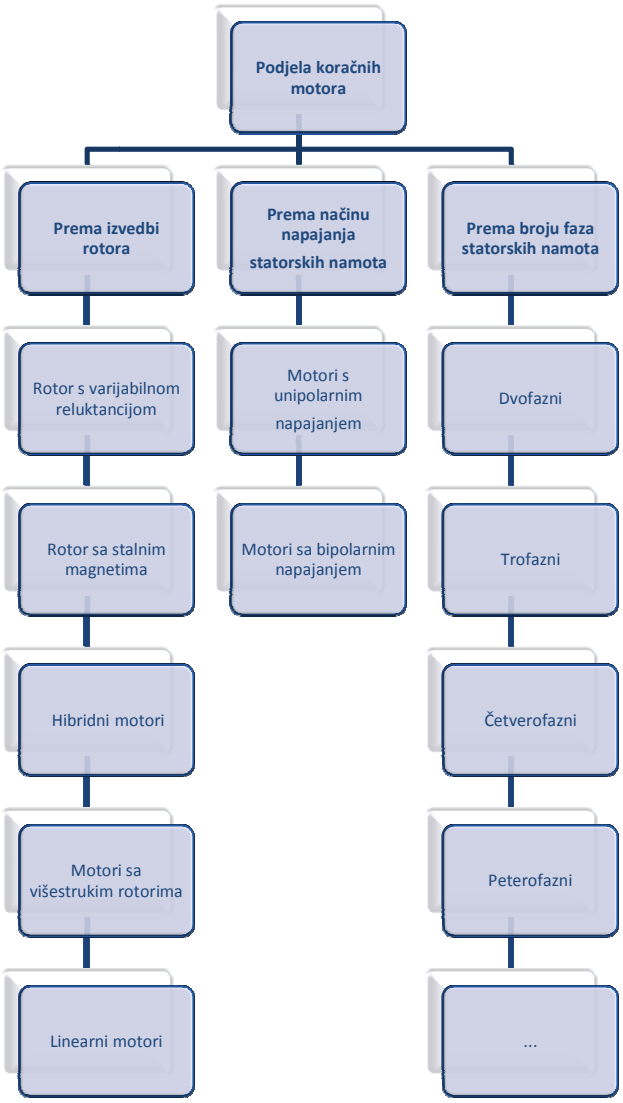
je u stanju južnog pola. Stator može imati 2,3 ili 4 nezavisna kruga ili faze, povezana sa sjever-jug parovima polova. Rotor ima vanjske zupce, jednako udaljene po periferiji, sa malim zračnim rasporom između statorskih i rotorskih zubaca.

Broj zubaca rotora, statora, i broj faza na statoru određuju veličinu koraka. Ovo se zove ugao koraka ili korački ugao, i za jednostavne motore se da izračunati:

Kut koraka = 360°/ (broj zuba rotora * broj faza statora)

Za primjer, koračni motor sa 16 zubaca rotora i 4 faze statora ima kut koraka od 7.5°.

10.4. Podjela

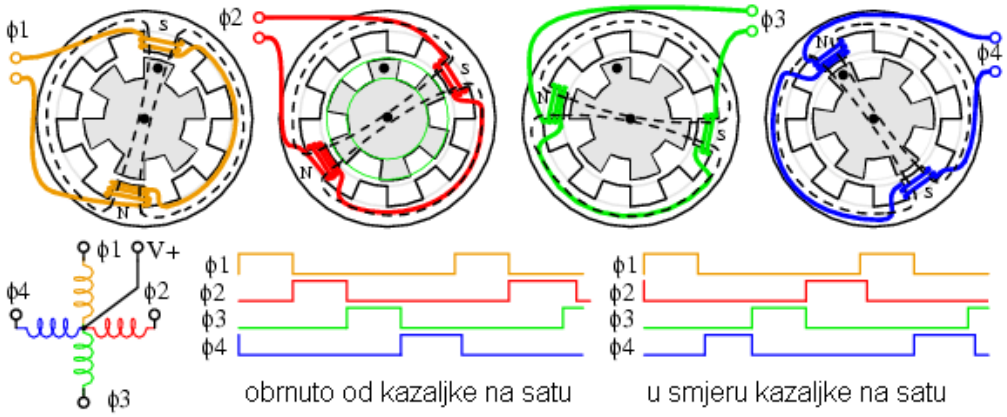


Slika 10.2. Podjela koračnih motora

10.5. Najčešći oblici pobude

- jednofazna
- dvofazna simultana
- tzv. half-step pobuda

Kod jednofazne pobude pojedini fazni namoti se uključuju odvojeno prema nekom logičkom slijedu. Primjer jednofazne pobude za zakretanje četverofaznog VR motora u oba smjera:



Slika 10.3. Jednofazna pobuda za zakretanje četverofaznih koračnih motora

Kod dvofazne simultane pobude u svakom trenutku bivaju ukljucena po dva fazna namota prema nekom logickom slijedu. U slijedećoj tablici se nalazi logički slijed dvofazne uzbude namotaja unipolarnog četverofaznog koračnog motora:

Tablica 10.1. Logički slijed dvofazne simultane uzbude namotaja unipolarnog četverofaznog koračnog motora

Korak	0	1	2	3
A	1	0	0	1
B	1	1	0	0
A\	0	1	1	0
B\	0	0	1	1

Mikrokorak je sličan dvofaznoj simultanoj pobudi ali je sad i vrijednost struje u zavojnici podesiva, pa pol rotora može zauzimati veliki broj diskretnih pozicija između polova statora.

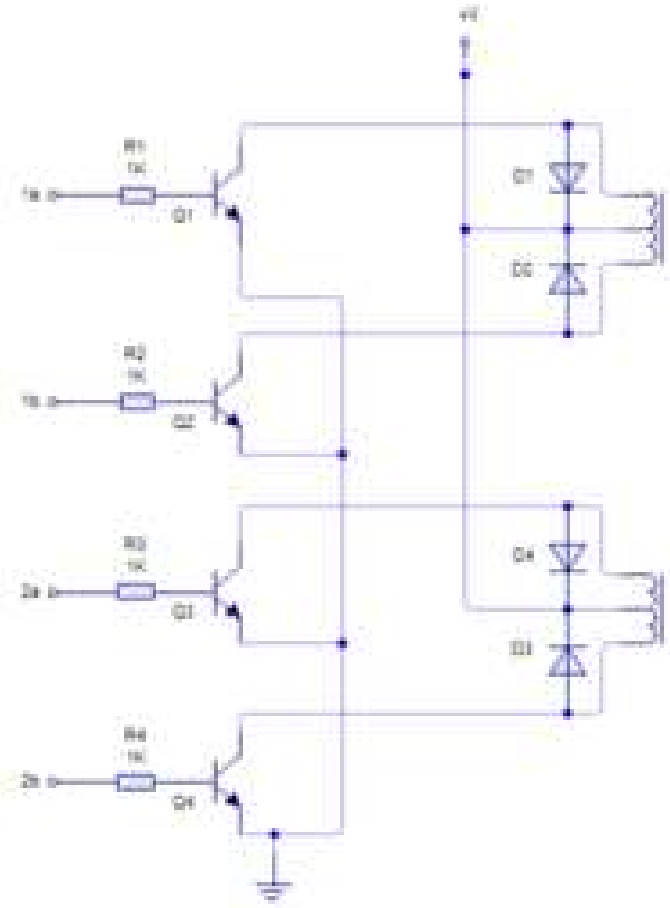
Tablica 10.2. Logički slijed mikrokorak (half-step) uzbude namotaja unipolarnog četverofaznog koračnog motora

Korak	0	1	2	3	4	5	6	7
A	1	0	0	0	0	0	1	1
B	1	1	1	0	0	0	0	0
A\	0	0	1	1	1	0	0	0
B\	0	0	0	0	1	1	1	0

10.6. Napajanje strujom

Rotor motora se pomiče u inkrementima (koracima), pa se zavojnice trebaju uključuivati jedna za drugom pod kontrolom mikroprocesora ili mikrokontrolera. Oni uključuju ili isključuju tranzistore putem baza. Tranzistori zatim propuštaju pulseve jače struje kroz zavojnice. Diode sprečavaju prenapone koji bi mogli uništiti tranzistore prilikom prekopčavanja.

Primjer spoja za kontrolu unipolarnog koračnog motora je prikazan na slici 7.15 .



Slika 10.4. Principijelna shema pogonskog spoja

11. VIJČANI POGON

Vijke pomoću kojih se okretno gibanje pretvara u uzdužno nazivamo vretena. Budući da oštri navoji imaju premale uspone, vijci za pokretanje najčešće dobivaju trapezni navoj kao što je u maketi korišten trapezni navoj Tr 12 x 3. Brže uzdužno gibanje matice vretena možemo postići viševojnim vretenima, kod njih se oko jezgre ovija više navoja (*n* navoja) jedan uz drugi. Proračun trapeznog vretena korištenog u maketi nalazi se u prilogu.

Slika 11.1. Trapezno vreteno

12. ZAKLJUČAK

Zanimljivost projektu „Pozicioniranje kamere u ravnini“ daje sinergija triju potpuno različitih područja znanosti – elektronika, strojarstvo i informatike ili jednom riječju mehatronika. Bez bilo koje od ove tri znanosti, projekt se ne bi mogao realizirati budući da svaka disciplina ima svoju ulogu u povezivanju mehaničkih i elektroničkih komponenata.

Svrha rada je pokazati kako povezivanjem spomenutih disciplina olakšati ljudski rad. Osim u zamjeni kamermana u studiju za snimanje, ovaj projekt bi mogao naći primjenu i u industriji (npr. pozicioniranje kranova, manipulatora, itd.). Da bi projekt imao stvarnu primjenu, zahtjevao bi jače motore i jači regulator izlaznog napona, te trapezno vreteno većih dimenzija (po mogućnosti viševojno).

Unaprijeđenje projekta, odnosno dizanje na višu razinu složenosti, zahtjevalo bi promjene u dva smjera – mehanički i programski dio.
Mehanički dio bi se mogao unaprijediti uvođenjem trećeg stupnja slobode gibanja (preporučeno os rotacije) ili nadogradnjom mobilnog postolja koje nalazi primjenu u području snimanja (njem. Kamera—ring).
Programski dio bi se mogao unaprijediti integracijom neuronskih mreža koje prepoznaju lica (eng. Face recognition) i/ili emocije (eng. Emotion recognition).

Osim što olakšava rad ljudima, može poslužiti i u situacijama kada ljudi nisu u mogućnosti obaviti određene radnje (npr. kamera na kolicima uz atletsku stazu koja prati trkače, itd.).

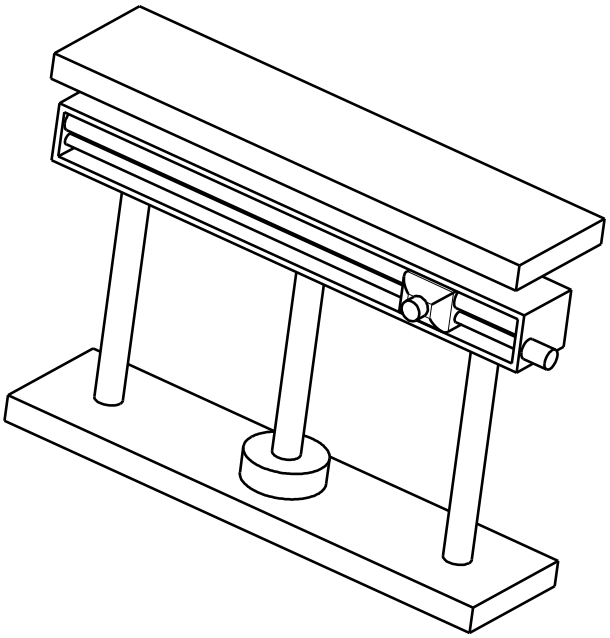
Razvojem ovih znanosti smanjuje se udio fizičkog rada ljudi a povećava se udio automatizacije raznih procesa kako u industriji tako i u ostalim djelatnostima što je uostalom i bila svrha ovog rada.

LITERATURA

- Wikipedia
- Riječnik elektronike, englesko-hrvatski i hrvatsko-engleski, A. Štambuk, M. Pervan, M. Pilković, V. Roje, LOGOS, Split, 1991.
- AutoIt Help datoteka
- MicroCode Studio Help datoteka
- Stručni forumi
- Elementi strojeva, Karl-Heinz Decker, GOLDEN MARKETING - TEHNIČKA KNJIGA, Zagreb
- Programski zadatak iz Elemenata konstrukcija I: Vijčana preša

A PRILOG

A.1. CAD model



	Datum	Ime i prezime	Potpis
Razradio	01.02.'10	Tomislav Štampar	
Projektirao	01.02.'10	Tomislav Štampar	
Crtao	01.02.'10	Tomislav Štampar	
Pregledao	01.02.'10	Tomislav Štampar	

FSB Zagreb

Objekt:

Objekt broj:

R. N. broj:

Napomena:

Kopija

Materijal:

Masa:

Naziv:

Model makete za pozicioniranje kamere u ravnini

Format:

Mjerilo orginala 1:5

Listova:

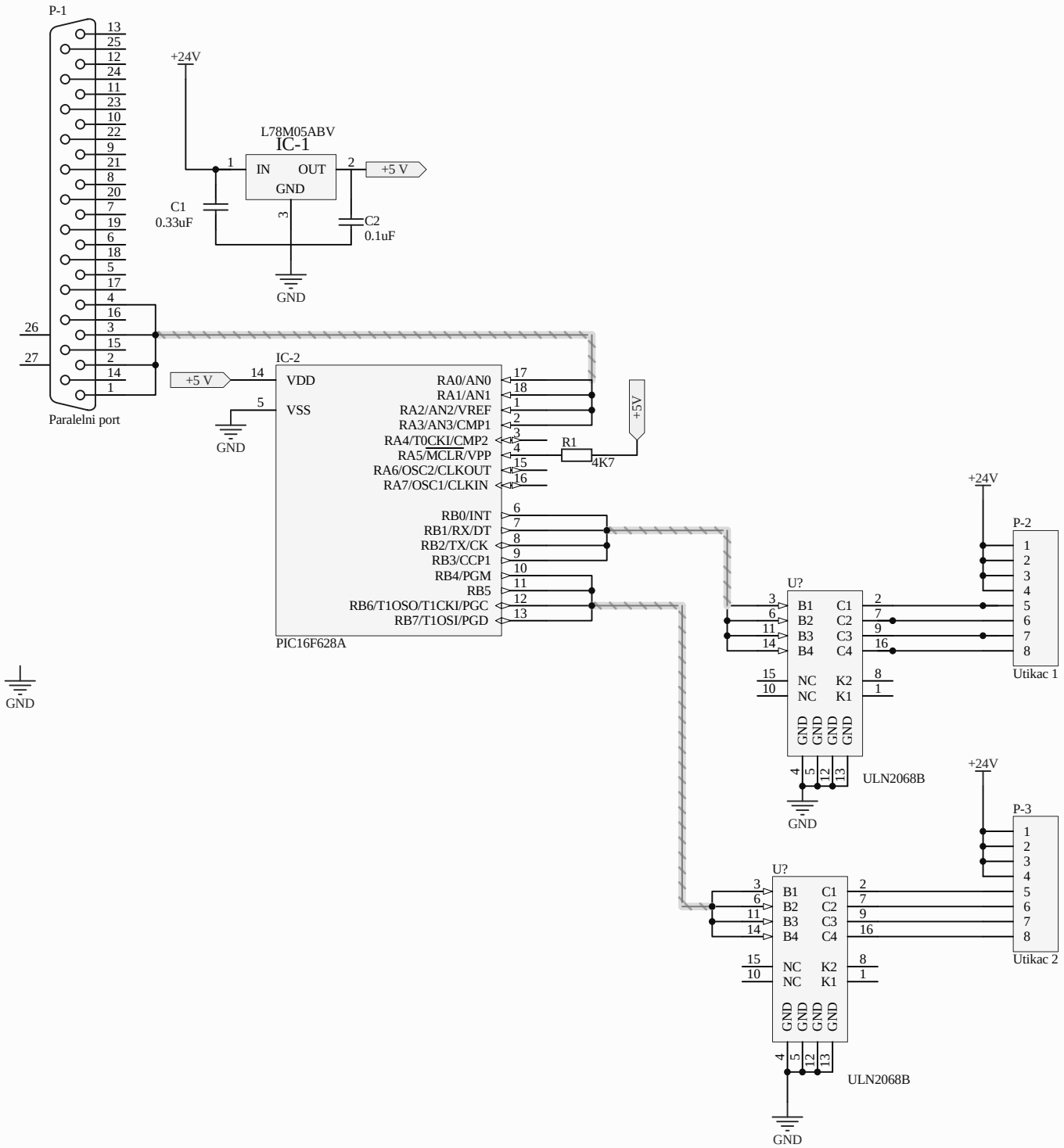
Crtež broj: 1

List:

Design by CADLab

F--stampar zavrsni.tif

A.2. Shema spajanja

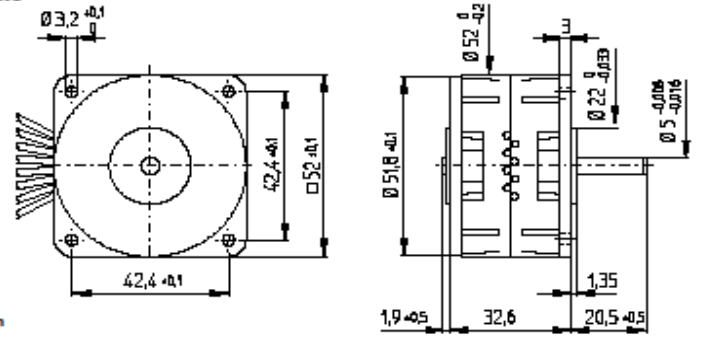


Naziv ***Shema spajanja***			
Format A4	Broj	Verzija 1	
Datum 9.2.2010	Time: 11:50:06	List 1 od 1	
File:			

A.3. Podatkovna tablica koračnih motora

P 532 — **Stepper motor**

Step angle: 3.6°



scale: 2/3
dimensions in mm
mass: 250g
lead wires: 200 ± 15 mm
0.25 mm² (AWG 24)

P532-258 - □□□ •10

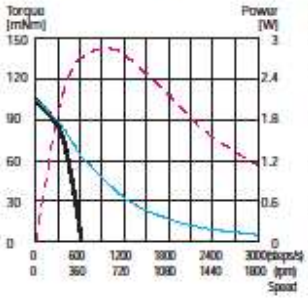
Windings available from stock □□□		**... - 012 • coils in series**			**... - 004 • coils in series**			**... - 004 • coils in parallel**			**... - 0.7 • coils in parallel**		
COIL DEPENDENT PARAMETERS		**min**	**typ**	**max**	**min**	**typ**	**max**	**min**	**typ**	**max**	**min**	**typ**	**max**
1 Phase resistance	ohm	24	27	29	8	8.8	9.5	2	2.2	2.4	0.3	0.34	0.38
2 Phase inductance (1 kHz)	mH		64			20			5			0.7	
3 Nominal phase current (2 ph. on)	A		0.4			0.7			1.4			3.7	
4 Nominal phase current (1 ph. on)	A		0.56			1			2			5.2	
5 Back-EMF amplitude	V/kst/s	17	21	25	10	12	14	5	6	7	1.8	2.3	2.8
COIL INDEPENDENT PARAMETERS [1]													
Torque parameters						min			typ			max	
6 Holding torque (nominal current)	mNm (oz-in)					174 (24.6)			205 (29)			236 (33.4)	
7 Holding torque (twice nominal current) [2]	mNm (oz-in)					306 (43.3)			360 (51)			414 (58.6)	
8 Detent torque amplitude and friction	mNm (oz-in)					14 (2)			28 (4)			40 (5.7)	
Thermal parameters													
9 Thermal resistance coil-ambient [3]	°C/W								7.3				
10 Coil temperature	°C											130	
11 Operating ambient temperature	°C					-20						50	
Angular accuracy													
12 Absolute accuracy (2 ph. on full-step mode)	% full-step								±3			±5	
Mechanical parameters													
13 Rotor inertia	kgm².10⁻⁷								12				
14 Radial load [4]	N											20	
15 Axial load [5]	N											30	
16 Radial shaft play (5 N)	µm								10			25	
17 Axial shaft play (5 N)	µm								10			25	
Other parameters													
18 Test voltage (1 min)	V_{RMS}								500				
19 Natural resonance frequency (nominal current)	Hz								330				
20 Electrical time constant	ms								2.3				
21 Angular acceleration (nominal current)	rad/s²								171000				
22 Power rate (nominal current)	kW/s								35				

[1] Bipolar driver
[2] The maximum coil temperature must be respected
[3] Motor unmounted
[4] Load applied at 12 mm from mounting face
[5] Shaft must be supported for press-fitting a pulley or pinion

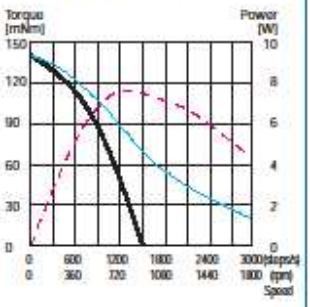
The P532 motor is also available from stock with the RG1/9 and K40 gearboxes (p. 101, 102).

Particular versions include options such as special shafts (hollow shaft), other gearboxes (R32, R40), optical encoders and so forth.

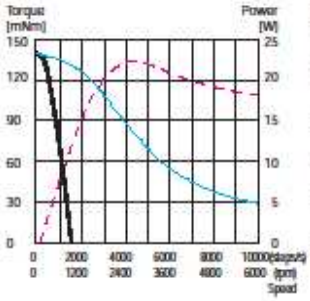
P532-258-004
Coils in series
escap® ELD-200
33Ω series resistor, 24V



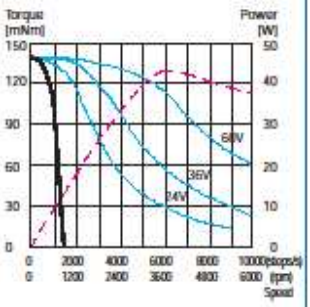
P532-258-012
Coils in series
escap® ELD-200
39Ω series resistor, 36V



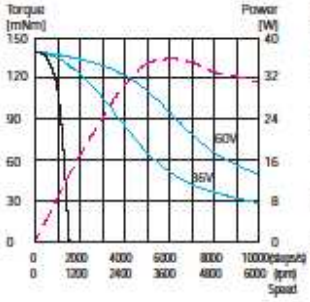
P532-258-004
Coils in parallel
escap® EDM-453, 34V, 2A



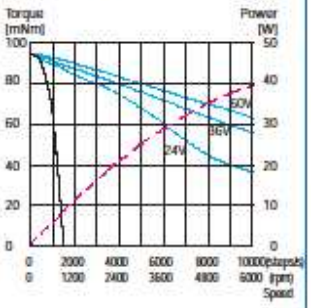
P532-258-004
Coils in parallel
escap® ESD-1200, I = 2A

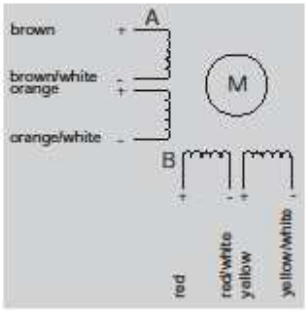


P532-258-0.7
Coils in series
escap® ESD-1300, I = 2.4A



P532-258-0.7
Coils in parallel
escap® ESD-1300, I = 3A





Motor connections

Pull-in range
Pull-out range
Power output

Notes
The low inertia, extended pull-in range, high peak speed and boost torque capability of this motor are benefits for fast incremental motion.
The speed scale is indicated in full-steps/s for all drive modes.
The motor is driven in half-steps unless otherwise specified.
The motor is energised with nominal current unless otherwise specified.
Pull-in is measured with a load inertia equal to the rotor inertia.
The following escap® drive circuits are recommended with the P532 motor, depending on the drive mode and the dynamic performance required: ELD-200, EDM-453, ESD-1200, ESD-1300.
Please refer to page 108 and 109 for more information on terminology and definitions.

A.4. Karakteristike PIC16F628A

• Radni takt do 20 MHz
• Mogućnost prekida
• 8 nivoa dubinskog sklopovnog stoga
• Direktno, intdirektno i relativno adresiranje
• 35 jednoriječnih instrukcija

Specijalne značajke mikrokontrolera:

•Mogućnost odabira unutarnjeg ili vanjskog oscilatora:
- Preciznost unutarnjeg oscilatora na 4 MHz je kalibrirana na ±1%
- Unutarnji 48 kHz oscillator male snage
- Vanjski oscilator podržava kritale i rezonatore
• Režim rada u mirovanju za štednju energije
• Programmable weak pull-ups on PORTB
• Multipleksiran pin za brisanje/ulaz
• Nadzorno brojilo (Watchdog Timer) sa neovisnim oscilatorom za pouzdan rad
• Nisko-naponsko programiranje
• Mogućnost programiranja serijskom komunikacijom na sklopu (pomoću 2 pina)
• Programibilna zaštita kôda
• Resetiranje memorije prilikom uključivanja napajanja
• Uključivanje brojila i oscilatora prilikom uključivanja napajanja
• Široki operacioni raspon napona (2.0-5.5V)
• Industrijski i povećani raspon temperatura
• Visoko izdržljiva Flash/EEPROM čelija:
- 100,000 pisanja u Flash memoriju
- 1,000,000 pisanja u EEPROM memoriju
- 40 godina sigurnog zadržavanja podataka

Značajke rada u režimu malih snaga:

• Struja u stanju pripravnosti:
- 100 nA @ 2.0V, tipično
• Struja u stanju rada:

- 12µA @ 32 kHz, 2.0V, tipično
- 120µA @ 1 MHz, 2.0V, tipično
• Struja nadzornog brojila:
- 1µA @ 2.0V, tipično
• Struja Timer1 oscilatora:
- 1.2µA @ 32 kHz, 2.0V, tipično
• Varijabilna frekvencija internog oscilatora:
- Radni takt selektivan između 4 MHz i 48 kHz
- 4µs vrijeme buđenja iz pripravnosti, 3.0V, tipično

Periferne značajke:

• 16 ulazno/izlaznih pinova sa individualnom kontrolom smjera
• Visoka strujna dobava za direktno vođenje LE-dioda
• Analogni komparatorni modul sa:
- Dva analogna komparatora
- Programabilna naponska referenca unutar čipa
- Selektivna unutarnja ili vanjska referenca
- Komparatorski izlazi su pristupni izvana
• Timer0: 8-bitni brojač/brojilo sa 8-bitnim programibilnim djeljiteljem
• Timer1: 16-bitni brojač/brojilo sa vanjskim kristalom/sposobnost generiranja taktova
• Timer2: 8-bitni brojač/brojilo sa 8-bitnim period-registrom, djeljiteljem i množiteljem
• Primanje i uspoređivanje PWM modula:
- 16-bitno primanje/uspoređivanje
- 10-bitni PWM
• Adresibilna univerzalna sinkrona/asinkrona prijemnik/predajnik komunikacija (USART/SCI)

A.5. Podatkovne tablice naponskog regulatora

Order this document by ULN2068/D

MOTOROLA

Quad 1.5 A Sinking High Current Switch

ULN2068

QUAD 1.5 A DARLINGTON SWITCH

SEMICONDUCTOR TECHNICAL DATA

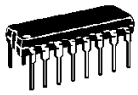
The ULN2068B is a high–voltage, high–current quad Darlington switch array designed for high current loads, both resistive and reactive, up to 300 W.

It is intended for interfacing between low level (TTL, DTL, LS and 5.0 V CMOS) logic families and peripheral loads such as relays, solenoids, dc and stepping motors, multiplexer LED and incandescent displays, heaters, or other high voltage, high current loads.

The Motorola ULN2068B is specified with minimum guaranteed breakdown of 50 V and is 100% tested for safe area using an inductive load. It includes integral transient suppression diodes. Use of a predriver stage reduces input current while still allowing the device to switch 1.5 Amps.

It is supplied in an improved 16–Pin plastic DIP package with heat sink contact tabs (Pins 4, 5, 12 and 13). A copper alloy lead frame allows maximum power dissipation using standard cooling techniques. The use of the contact tab lead frame facilitates attachment of a DIP heat sink while permitting the use of standard layout and mounting practices.

- TTL, DTL, LS, CMOS Compatible Inputs
- 1.5 A Maximum Output Current
- Low Input Current
- Internal Freewheeling Clamp Diodes
- 100% Inductive Load Tested
- Heat Tab Copper Alloy Lead Frame for Increased Dissipation



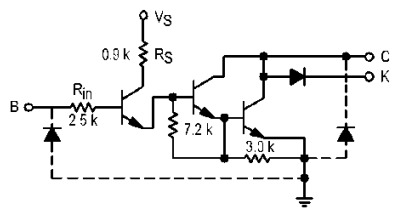
B SUFFIX
PLASTIC PACKAGE
CASE 648C

MAXIMUM RATINGS (T_A = 25°C and ratings apply to any one device in the package, unless otherwise noted)

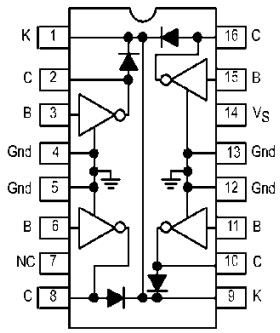
Rating	Symbol	Value	Unit
Output Voltage	V_O	50	V
Input Voltage (Note 1)	V_I	15	V
Supply Voltage	V_S	10	V
Collector Current (Note 2)	I_C	1.75	A
Input Current (Note 3)	I_I	25	mA
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	–55 to +150	°C
Junction Temperature	T_J	150	°C

NOTES: 1. Input voltage referenced to ground.
2. Allowable output conditions shown in Figures 11 and 12.
3. May be limited by max input voltage.

Partial Schematic



PIN CONNECTIONS



ORDERING INFORMATION*

Device	Operating Temperature Range	Package
ULN2068B	T_A = 0 to +70°C	Plastic DIP

*Other options of this ULN2060/2070 series are available for volume applications. Contact your local Motorola Sales Representative.

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Leakage Current (Figure 1) (V_{CE} = 50 V) (V_{CE} = 50 V, T_A = 70°C)	I_{CEX}	 – –	 – –	 100 500	µA
Collector–Emitter Saturation Voltage (Figure 2) (I_C = 500 mA (I_C = 750 mA (I_C = 1.0 A (I_C = 1.25 A } V_{in} = 2.4 V)	$V_{CE(sat)}$	 – – – –	 – – – –	 1.13 1.25 1.40 1.60	V
Input Current – On Condition (Figure 4) (V_I = 2.4 V) (V_I = 3.75 V)	$I_{I(on)}$	 – –	 – –	 0.25 1.0	mA
Input Voltage – On Condition (Figure 5) (V_{CE} = 2.0 V, I_C = 1.5 A)	$V_{I(on)}$	 –	 –	 2.4	V
Inductive Load Test (Figure 3) (V_S = 5.5 V, V_{CC} = 24.5 V, t_{PW} = 4.0 ms)	ΔV_{out}	 –	 –	 100	mV
Supply Current (Figure 8) (I_C = 500 mA, V_{in} = 2.4 V, V_S = 5.5 V)	I_S	 –	 –	 6.0	mA
Turn–On Delay Time (50% E_I to 50% E_O)	t_{PHL}	 –	 –	 1.0	µs
Turn–Off Delay Time (50% E_I to 50% E_O)	t_{PLH}	 –	 –	 4.0	µs
Clamp Diode Leakage Current (Figure 6) (V_R = 50 V) (V_R = 50 V, T_A = 70°C)	I_R	 – –	 – –	 50 100	µA
Clamp Diode Forward Voltage (Figure 7) (I_F = 1.0 A) (I_F = 1.5 A)	V_F	 – –	 – –	 1.75 2.0	V

TEST FIGURES

Figure 1.

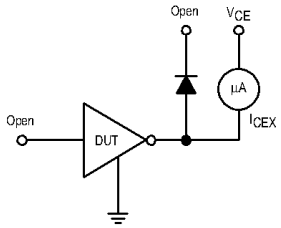


Figure 2.

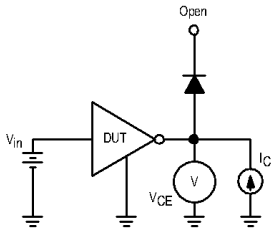


Figure 3.

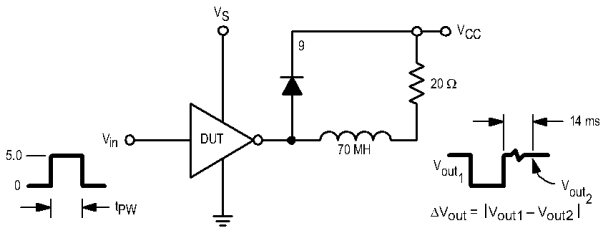
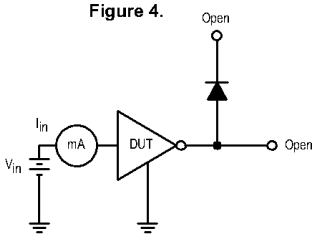


Figure 4.



TEST FIGURES (continued)

Figure 5.

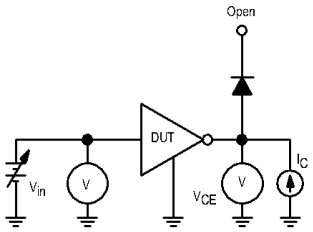


Figure 6.

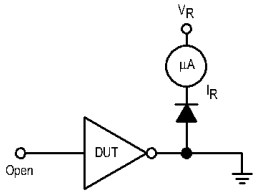


Figure 7.

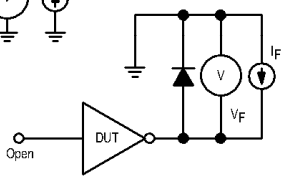
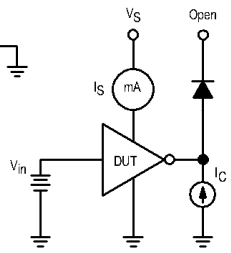


Figure 8.



TYPICAL CHARACTERISTIC CURVES – T_A = 25°C

Figure 9. Input Current versus Input Voltage

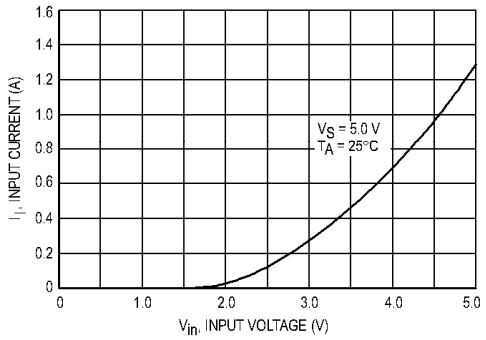


Figure 10. Collector Current versus Input Current

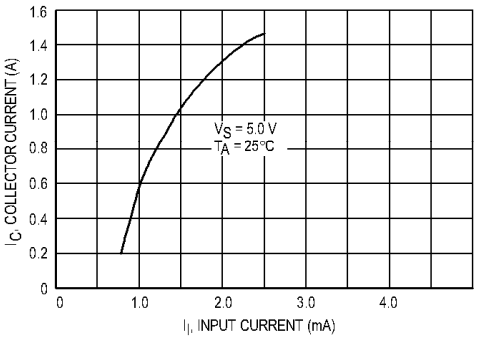


Figure 11. T_A = 70°C w/o Heat Sink

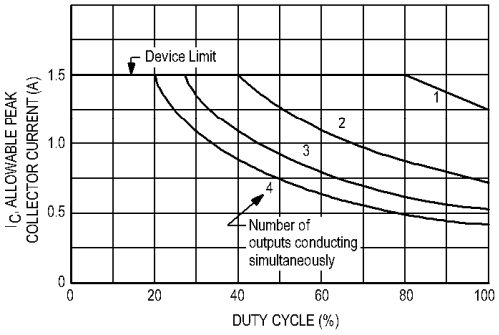


Figure 12. T_A = 70°C w/Staver V–8 Heat Sink (37.5°C/W)

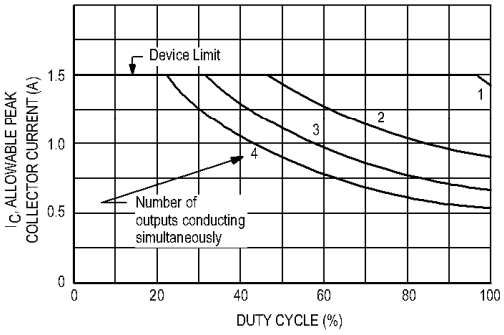


Figure 13. T_A = 70°C w/Staver V–7 Heat Sink (27.5°C/W)

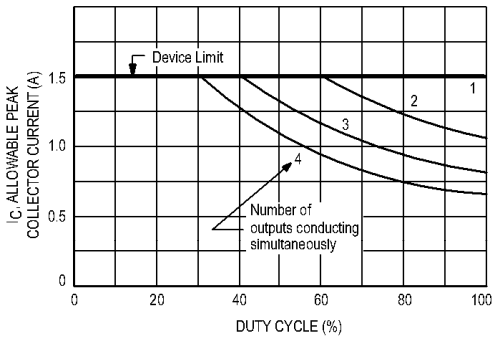


Figure 14. T_A = 50°C w/o Heat Sink

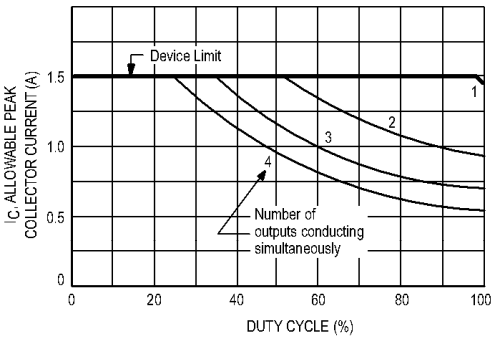


Figure 15. T_A = 50°C w/Staver V–8 Heat Sink (37.5°C/W)

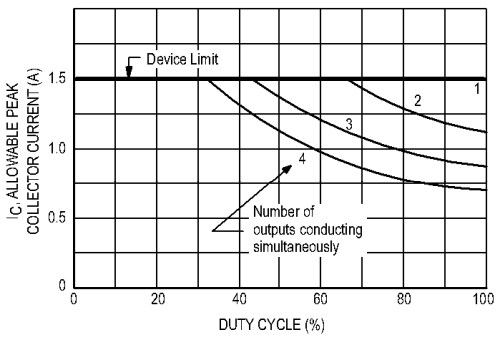
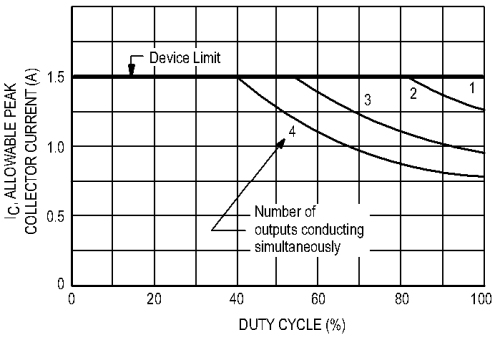
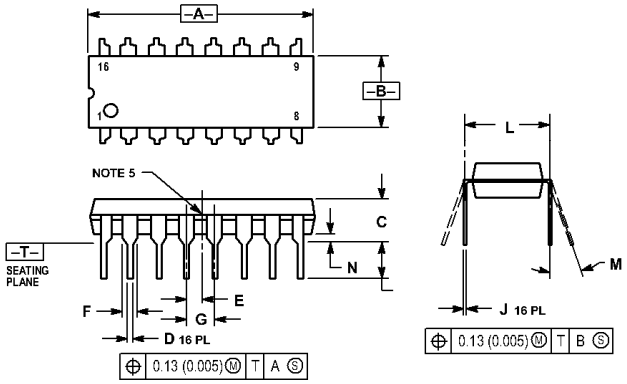


Figure 16. T_A = 50°C w/Staver V–7 Heat Sink (27.5°C/W)



OUTLINE DIMENSIONS

B SUFFIX
PLASTIC PACKAGE
CASE 648C–03
ISSUE C



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
5. INTERNAL LEAD CONNECTION BETWEEN 4 AND 5, 12 AND 13.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.740	0.840	18.80	21.34
B	0.240	0.260	6.10	6.60
C	0.145	0.185	3.69	4.69
D	0.015	0.021	0.38	0.53
E	0.050 BSC		1.27 BSC	
F	0.040	0.70	1.02	1.78
G	0.100 BSC		2.54 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.300 BSC		7.62 BSC	
M	0°	10°	0°	10°
N	0.015	0.040	0.39	1.01

A.6. Skripta za inicijalizaciju igraćeg kontrolera

```
 #include <GUIConstants.au3>
; SETUP
Local $joy,$coor,$h,$s,$msg
$joy    = _JoyInit()
dim $labels_text[10]=['X', 'Y', 'Z', 'R', 'U', 'V', 'POV', 'Buttons','Vrijeme X', 'Vrijeme Y']
dim $labels_no=UBound($labels_text)
dim $labels[$labels_no]
dim $labels_value[$labels_no]
dim $B
dim $C
dim $D
dim $E
dim $vrijeme_x_os
dim $vrijeme_y_os
; CONFIG
;Traženje najduže riječi
$label_len=0
for $text in $labels_text
    $len=stringlen($text)
    if $len>$label_len then
        $label_len=$len
    endif
next
$label_len*=10


; GUI
GUICreate('Joystick Test', 250, 250)
GUICtrlCreateLabel('Joystick', 40, 20, 100, 20)
for $i=0 to $labels_no-1
    GuiCtrlCreatelabel($labels_text[$i]&':', 10, 60+$i*12, $label_len, 12)
    $labels[$i]=GuiCtrlCreatelabel('', 10+$label_len, 60+$i*12, 70, 12)
    $labels_value[$i]=''
next
GUISetState()
```

```
;
while 1
    $coord=_GetJoy($joy,0)
    for $i=0 to UBound($coord)-1
        if $coord[$i]<>$labels_value[$i] then
            GUICtrlSetData($labels[$i], $coord[$i])
            $labels_value[$i]=$coord[$i]
        endif
    next
    sleep(10)
    $msg =GUIGetMSG()
    if $msg = $GUI_EVENT_CLOSE Then Exitloop
WEnd
$lpJoy=0 ; Joyclose
;====================================
; inicijalizacija igraceg kontrolera
;====================================
Func _JoyInit()
    Local $joy
    Global $JOYINFOEX_struct    = "dword[13]"
    $joy=DllStructCreate($JOYINFOEX_struct)
    if @error Then Return 0
    DllStructSetData($joy, 1, DllStructGetSize($joy), 1);dwSize = sizeof(struct)
    DllStructSetData($joy, 1, 255, 2)             ;dwFlags = GetAll
    return $joy
EndFunc
;====================================
Func _GetJoy($lpJoy,$iJoy)
    Local $coor,$ret

    Dim $coor[10]
    DllCall("Winmm.dll","int","joyGetPosEx", _
          "int",$iJoy, _
          "ptr",DllStructGetPtr($lpJoy))
    if Not @error Then
        $coor[0]    = DllStructGetData($lpJoy,1,3)
        $coor[1]    = DllStructGetData($lpJoy,1,4)
```

```
        $coor[2]    = DllStructGetData($lpJoy,1,5)
        $coor[3]    = DllStructGetData($lpJoy,1,6)
        $coor[4]    = DllStructGetData($lpJoy,1,7)
        $coor[5]    = DllStructGetData($lpJoy,1,8)
        $coor[6]    = DllStructGetData($lpJoy,1,11)
        $coor[7]    = DllStructGetData($lpJoy,1,9)
                    $coor[8]            = $vrijeme_x_os
                    $coor[9]            = $vrijeme_y_os
    EndIf

if DllStructGetData($lpJoy,1,9) = 2 then
                    DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", 0)
EndIf
;odredivanje varijable "vrijeme"
Select
case DllStructGetData($lpJoy,1,3) > 31000 and DllStructGetData($lpJoy,1,3)< 34500
        $vrijeme_x_os = 100
case DllStructGetData($lpJoy,1,4) > 31000 and DllStructGetData($lpJoy,1,4)< 34500
        $vrijeme_y_os = 100
;x os
            Case  DllStructGetData($lpJoy,1,3) > 27900 and DllStructGetData($lpJoy,1,3)< 31000
        $vrijeme_x_os = 100
            Case  DllStructGetData($lpJoy,1,3) > 24800 and DllStructGetData($lpJoy,1,3)< 27900
        $vrijeme_x_os = 90
            Case  DllStructGetData($lpJoy,1,3) > 21700 and DllStructGetData($lpJoy,1,3)< 24800
        $vrijeme_x_os = 80
            Case  DllStructGetData($lpJoy,1,3) > 18600 and DllStructGetData($lpJoy,1,3)< 21700
        $vrijeme_x_os = 70
            Case  DllStructGetData($lpJoy,1,3) > 15500 and DllStructGetData($lpJoy,1,3)< 18600
        $vrijeme_x_os = 60
            Case  DllStructGetData($lpJoy,1,3) > 12400 and DllStructGetData($lpJoy,1,3)< 15500
        $vrijeme_x_os = 50
            Case  DllStructGetData($lpJoy,1,3) > 9300 and DllStructGetData($lpJoy,1,3)< 12400
        $vrijeme_x_os = 40
            Case  DllStructGetData($lpJoy,1,3) > 6200 and DllStructGetData($lpJoy,1,3)< 9300
        $vrijeme_x_os = 30
            Case  DllStructGetData($lpJoy,1,3) < 6200
```

```
        $vrijeme_x_os = 20

            Case  DllStructGetData($lpJoy,1,3) > 34500 and DllStructGetData($lpJoy,1,3)< 37600
        $vrijeme_x_os = 100
            Case  DllStructGetData($lpJoy,1,3) > 37600 and DllStructGetData($lpJoy,1,3)< 40700
        $vrijeme_x_os = 90
            Case  DllStructGetData($lpJoy,1,3) > 40700 and DllStructGetData($lpJoy,1,3)< 43800
        $vrijeme_x_os = 80
            Case  DllStructGetData($lpJoy,1,3) > 43800 and DllStructGetData($lpJoy,1,3)< 46900
        $vrijeme_x_os = 70
            Case  DllStructGetData($lpJoy,1,3) > 46900 and DllStructGetData($lpJoy,1,3)< 50000
        $vrijeme_x_os = 60
            Case  DllStructGetData($lpJoy,1,3) > 50000 and DllStructGetData($lpJoy,1,3)< 53100
        $vrijeme_x_os = 50
            Case  DllStructGetData($lpJoy,1,3) > 53100 and DllStructGetData($lpJoy,1,3)< 56200
        $vrijeme_x_os = 40
            Case  DllStructGetData($lpJoy,1,3) > 56200 and DllStructGetData($lpJoy,1,3)< 59300
        $vrijeme_x_os = 30
            Case  DllStructGetData($lpJoy,1,3)  > 59300
        $vrijeme_x_os = 20
EndSelect
Select
;y os
    Case  DllStructGetData($lpJoy,1,4) > 27900 and DllStructGetData($lpJoy,1,4)< 31000
        $vrijeme_y_os = 100
            Case  DllStructGetData($lpJoy,1,4) > 24800 and DllStructGetData($lpJoy,1,4)< 27900
        $vrijeme_y_os = 90
            Case  DllStructGetData($lpJoy,1,4) > 21700 and DllStructGetData($lpJoy,1,4)< 24800
        $vrijeme_y_os = 80
            Case  DllStructGetData($lpJoy,1,4) > 18600 and DllStructGetData($lpJoy,1,4)< 21700
        $vrijeme_y_os = 70
            Case  DllStructGetData($lpJoy,1,4) > 15500 and DllStructGetData($lpJoy,1,4)< 18600
        $vrijeme_y_os = 60
            Case  DllStructGetData($lpJoy,1,4) > 12400 and DllStructGetData($lpJoy,1,4)< 15500
        $vrijeme_y_os = 50
            Case  DllStructGetData($lpJoy,1,4) > 9300 and DllStructGetData($lpJoy,1,4)< 12400
        $vrijeme_y_os = 40
```

```
        Case  DllStructGetData($lpJoy,1,4) > 6200 and DllStructGetData($lpJoy,1,4)< 9300
      $vrijeme_y_os = 30
        Case  DllStructGetData($lpJoy,1,4) < 6200
      $vrijeme_y_os = 20
       Case  DllStructGetData($lpJoy,1,4) > 34500 and DllStructGetData($lpJoy,1,4)< 37600
      $vrijeme_y_os = 100
        Case  DllStructGetData($lpJoy,1,4) > 37600 and DllStructGetData($lpJoy,1,4)< 40700
      $vrijeme_y_os = 90
        Case  DllStructGetData($lpJoy,1,4) > 40700 and DllStructGetData($lpJoy,1,4)< 43800
      $vrijeme_y_os = 80
        Case  DllStructGetData($lpJoy,1,4) > 43800 and DllStructGetData($lpJoy,1,4)< 46900
      $vrijeme_y_os = 70
        Case  DllStructGetData($lpJoy,1,4) > 46900 and DllStructGetData($lpJoy,1,4)< 50000
      $vrijeme_y_os = 60
        Case  DllStructGetData($lpJoy,1,4) > 50000 and DllStructGetData($lpJoy,1,4)< 53100
      $vrijeme_y_os = 50
        Case  DllStructGetData($lpJoy,1,4) > 53100 and DllStructGetData($lpJoy,1,4)< 56200
      $vrijeme_y_os = 40
        Case  DllStructGetData($lpJoy,1,4) > 56200 and DllStructGetData($lpJoy,1,4)< 59300
      $vrijeme_y_os = 30
        Case  DllStructGetData($lpJoy,1,4)  > 59300
      $vrijeme_y_os = 20


EndSelect

if DllStructGetData($lpJoy,1,3) < 28000 Then
                $E=2
                $D=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
                $E=0
                $D=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
        EndIf
if DllStructGetData($lpJoy,1,3) > 36000  then
```

```
                $E=2
                $D=1
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
                $E=0
                $D=1
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
        EndIf

if DllStructGetData($lpJoy,1,4) < 28000 Then
                $C=8
                $B=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
                $C=0
                $B=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
        EndIf

if DllStructGetData($lpJoy,1,4) > 36000  then
                $C=8
                $B=4
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
                $C=0
                $B=4
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
        EndIf

if DllStructGetData($lpJoy,1,4) < 36000 And DllStructGetData($lpJoy,1,4) > 28000 then
                $B=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
                $B=0
```

```
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_y_os)
        EndIf
if DllStructGetData($lpJoy,1,3) < 36000 And DllStructGetData($lpJoy,1,3) > 28000 then
                $D=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
                $D=0
                DllCall( "c:\inpout32.dll", "int", "Out32", "int", 0x378, "int", $E+$D+$B+$C)
                sleep ($vrijeme_x_os)
        EndIf

return $coor
EndFunc
```

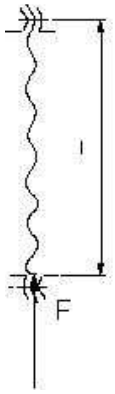
A.7. Programski kôd

```
DEFINE OSC 8
CCP1CON = 0 ' turn off capture/compare mode (set all digital I/O)
CMCON = 7 ' disable comparators
OPTION_REG = %10000000 ' bit 7 = 1 disables weak pull-ups on portB
INTCON = 0 ' disable interrupts
TRISA = %00000000
TRISB = %11111111
PORTA = 0
PORTB = 0
x VAR BYTE
y var byte
steps1 VAR WORD
steps2 var word
start1:

steps1 = 0
main:
   if (portb.0 = 1) and (portb.1 = 0) then
      steps1 = steps1 + 1
   endif
   if (portb.0 = 1) and (portb.1 = 1) then
      steps = steps - 1
      if steps1 = 0 then
         steps1 = 4
      endif
   endif
Select case steps1
   CASE 0
      Y = 0
   case 1
      y = 10
   case 2
      y = 6
   CASE 3
      y = 5
   CASE 4
      y = 9
   case 5
```

```
        goto start1
    END SELECT
pause 25

start2:
steps2 = 0
if (portb.2 = 1) and (portb.3 = 0) then
        steps2 = steps2 + 1
    endif
if (portb.2 = 1) and (portb.3 = 1) then
        steps2 = steps2 - 1
        if steps2 = 0 then
            steps2 = 4
        endif
    endif
Select case steps2
    CASE 0
        x = 0
    case 1
        x = 160
    case 2
        x = 96
    CASE 3
        x = 80
    CASE 4
        x = 144
    case 5
        goto start2
    END SELECT
pause 25

PORTA = x + y

GoTo main
```

A.8. Proračun trapeznog vretena s normalnim jednovojnim trapeznim navojem (HRN M.B0.060 do 064)

Potreban promjer jezgre vretena proračunava se prethodno prema Euler-u za elastično područje izvijanja:

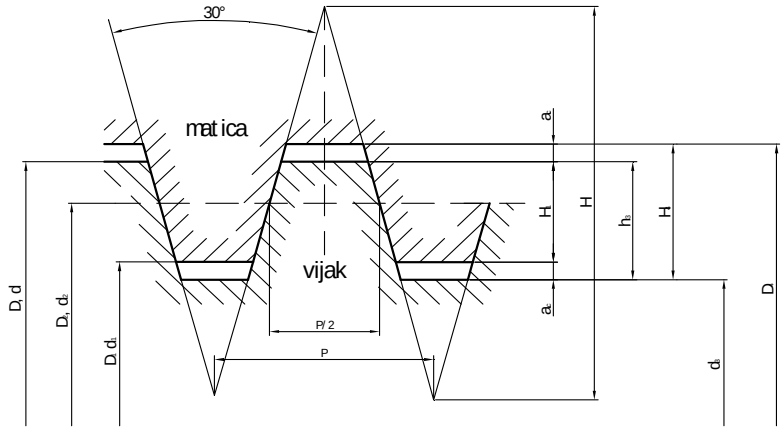


Za ovaj slučaj opterećenja: $l_0 = \frac{l}{2}$

$$d_3 = \sqrt[4]{\frac{64 \bullet F \bullet S \bullet l_0^{\,2}}{E \bullet \pi^3}}$$

Trapezni navoj:

Oznaka	P $[mm]$	d $[mm]$	$d_2 = D_2$ $[mm]$	d_3 $[mm]$	D_1 $[mm]$	D_4 $[mm]$	A_j $[mm^2]$
Tr 12x3	3	12	10,5	8,5	9	12,5	56,7



Slika Karakteristične dimenzije navoja

Mjere profila trapeznog navoja [*mm*]:

P	H_1	a_c	$h_3 = H_4$
3	1,5	0.25	1,25

F_k [*N*] – sila izvijanja

F [*N*] – tlačna sila na vreteno

S – sigurnost protiv izvijanja (8-10) – odabrana sigurnost S = 9

E [N/mm^2] – modul elastičnosti (za čelik, E = 210 000 N/mm^2, iz Bojan Kraut: Strojarski priručnik, Tehnička knjiga Zagreb, 1988., str. 112)

I_{mir} [mm^4] – najmanji aksijalni moment tromosti

l_0 [*mm*] – slobodna duljina izvijanja

d_3 [*mm*] – promjer jezgre vretena

d [*mm*] – nazivni promjer vretena

d_2 [*mm*] – srednji promjer vretena

P_h [*mm*] – uspon navoja vretena

P [*mm*] – korak navoja vretena

$$l_0 = \frac{l}{2} = \frac{390}{2} = 195\ mm$$

$$F = \frac{d_3^4 * E * \pi^3}{64 * S * l_0^2} = \frac{8{,}5^4 * 210000 * \pi^3}{64 * 9 * 195^2} = 1551{,}86N$$

$$I_{min} = \frac{d_3^4 \pi}{64} = 256{,}24mm^4$$

$$F_k = S \bullet F = \pi^2 \frac{E \bullet I_{min}}{l_0^2} = 13966{,}8\ N$$

Kontrola naprezanja

Vreteno složeno je opterećeno na tlak i torziju pa se računa reducirano naprezanje koje mora biti manje od dopuštenog naprezanja.

Tlačno naprezanje:

$$\sigma = \frac{F}{A_j} \ [N/mm^2]$$

σ $[N/\ mm^2]$ – tlačno naprezanje vretena

F $[N]$ – tlačna sila

A_j $[mm^2]$ – presjek jezgre vretena

$$\sigma = \frac{F}{A_j} = \frac{1551,86}{56,7} = 27,37\ N/mm^2$$

Torzijsko naprezanje:

$$\tau_t = \frac{T}{W_p} \ [N/\ mm^2]$$

$$T = F \bullet \frac{d_2}{2} tan(\alpha + \rho') \ [Nmm]$$

$$tg\ \alpha = \frac{P_h}{d_2 \pi} \ \rightarrow \ \alpha$$

$$Tg\ \rho' = \frac{\mu}{cos\beta} \ \rightarrow \ \rho'$$

- $\alpha < \rho'$ → navoj je samokočan
- $\alpha > \rho'$ → navoj nije samokočan

$$W_p = \frac{d_3{}^3 \pi}{16} \ [mm^3]$$

τ_t $[N/\ mm^2]$ – torzijsko naprezanje vretena

T $[Nmm]$ – torzijski moment navoja vretena

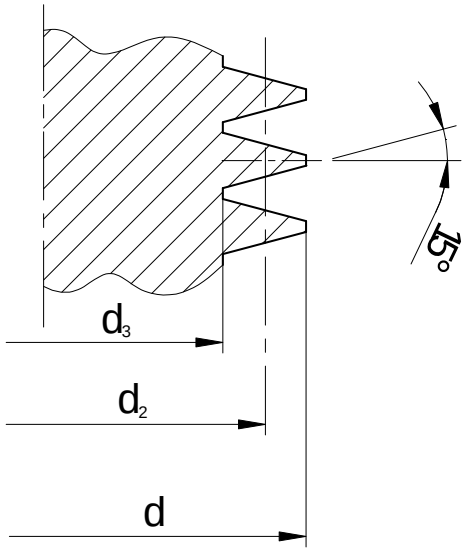
W_p $[mm^3]$ – polarni moment otpora

α [°] – kut uspona

ρ' [°] – korigirani kut trenja

μ = 0.1 – faktor trenja za čelik-broncu

β = 15° – polovina vršnog kuta navoja (vidi sliku do lje)



$$tg\,\alpha = \frac{P_h}{d_2\pi} = \frac{3}{10,5 * \pi} = 0.0909 \;\rightarrow\; \alpha = arctg\,0.0909 = 5,19\,^\circ$$

$$tg\,\rho' = \frac{\mu}{\cos\beta} = \frac{0.1}{\cos 15^\circ} = 0.1035 \;\rightarrow\; \rho' = arctg\,0.1035 = 5.9^\circ$$

$\alpha > \rho'$ → 5,19° < 5.9° → navoj JE samokočan!

$$W_p = \frac{d_3{}^3\pi}{16} = \frac{29^3 \bullet \pi}{16} = 120,58\ mm^3$$

$$T = F\frac{d_2}{2}\tan(\alpha + \rho') = 1551,86 * \frac{10,5}{2} * tg(5,13 + 5,91) = 1589,57\ Nmm$$

$$\tau_t = \frac{T}{W_p} = \frac{1589,57}{120,58} = 13,18\ N/mm^2$$

Reducirano naprezanje:

$$\sigma_{red} = \sqrt{\sigma^2 + 3\tau_t} \leq \sigma_{dop} \ [N/\ mm^2]$$

$$\sigma_{dop} = \frac{\sigma_{Dl}}{S_{potr}} \ [N/\ mm^2]$$

σ_{red} [N/mm^2] - reducirano naprezanje

$\sigma_{Dl}\ [N / mm^2]$ - istosmjerno promjenljiva trajna čvrstoća – za Č 0545 (prema Bojan Kraut: Strojarski priručnik. Tehnička knjiga Zagreb, 1988., str. 533.)

$\sigma_{Dl} = 220...270\ N / mm^2$, odabrano: $\sigma_{Dl} = 245\ N / mm^2$

$S_{potr} = 2$ - potrebna sigurnost

$$\sigma_{dop} = \frac{\sigma_{Dl}}{S_{potr}} = \frac{245}{2} = 122.5\ N / mm^2$$

$$\sigma_{red} = \sqrt{\sigma^2 + 3\tau_t^2} = \sqrt{27,37^2 + 3*13,18^2} = 35,64\ N / mm^2 \leq 122.5\ N/\ mm^2$$

Sigurnost u odnosu prema naprezanju na izvijanje σ_k

Faktor vitkosti vretena:

$$\lambda = \frac{l_0}{i}$$

$$i = \sqrt{\frac{I_{min}}{A_j}} \ [mm]$$

$$I_{min} = \frac{d_3^{\ 4}\pi}{64} \ [mm^4]$$

λ – faktor vitkosti

l_0 [mm] – slobodna duljina izvijanja

i [mm] – polumjer tromosti

I_{min} = 256,24 mm^4 – najmanji aksijalni moment tromosti vretena

A_j = 56,7 mm^2 – površina presjeka jezgre vretena

d_3 = 29 *mm* – promjer jezgre vretena

$$i = \sqrt{\frac{I_{min}}{A_j}} = \sqrt{\frac{256,24}{56,7}} = 2,126\ mm$$

$$\lambda = \frac{l_0}{i} = \frac{195}{2,126} = 91,72$$

- za mekani čelik → $\lambda_0 = 105$ (Bojan Kraut: Strojarski priručnik, Tehnička knjiga Zagreb, 1988., str. 121)

$\lambda < \lambda_0$ (91,72 < 105) → postoji neelastično izvijanje pa se proračun provodi prema Tetmajer-u:

$\sigma_k = 335 - 0.62\lambda$ [N/mm^2]

$$S = \frac{\sigma_k}{\sigma_{red}}$$

$\sigma_k = 335 - 0.62\lambda = 335 - 0.62 * 91,72 = 278,13\ N/mm^2$

Sigurnost protiv izvijanja:

$$S = \frac{\sigma_k}{\sigma_{red}} = \frac{278,13}{35,64} = 7.8$$