

# Upravljanje pneumatskim manipulatorom pomoću otvorene elektroničke platforme

---

**Poznanović, Dario**

**Undergraduate thesis / Završni rad**

**2019**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:640159>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-14**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Dario Poznanović**

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Željko Šitum, dipl. ing.

Student:

Dario Poznanović

Zagreb, 2019.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se obitelji na pruženoj podršci prilikom izrade ovoga završnog rada.

Dario Poznanović



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## ZAVRŠNI ZADATAK

Student: **Dario Poznanović** Mat. br.: 0035203647

Naslov rada na hrvatskom jeziku: **Upravljanje pneumatskim manipulatorom pomoću otvorene elektroničke platforme**

Naslov rada na engleskom jeziku: **Pneumatic manipulator control using open-source electronic platform**

Opis zadatka:

Pneumatski manipulator laboratorijski je uređaj čija namjena je prvenstveno edukacijska. Sastoji se od devet cilindara različite vrste, dvije vakumske prihvatnice te od jedanaest elektromagnetskih ventila za upravljanje cilindrima i vakumskim prihvatnicama. Cilindri omogućavaju dva translacijska i dva rotacijska gibanja manipulatora te jedno translacijsko gibanje za svaku od dvije palete. Palete služe za držanje predmeta koje manipulator prenosi iz jedne palete u drugu.

U ovom završnom zadatku potrebno je primijeniti upravljanje pneumatskim manipulatorom pomoću otvorene elektroničke i softverske platforme Arduino. Prethodno upravljanje bilo je realizirano pomoću industrijskog programabilnog logičkog kontrolera (PLC), koji više nije upotrebljiv. U zadatku također treba procijeniti funkcionalnost primjene jedne otvorene i jeftine elektroničke platforme poput Arduina na tipični primjer industrijskog upravljanja gdje se inače koriste PLC-ovi. Procjenu treba napraviti s obzirom na jednostavnost programiranja radnih zadataka, eventualno povezivanje na internet radi upravljanja i nadzora, ožičenje, napajanje i slično.

U zadatku je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:

29. studenog 2018.

Rok predaje rada:

1. rok: 22. veljače 2019.

2. rok (izvanredni): 28. lipnja 2019.

3. rok: 20. rujna 2019.

Predviđeni datumi obrane:

1. rok: 25.2. - 1.3. 2019.

2. rok (izvanredni): 2.7. 2019.

3. rok: 23.9. - 27.9. 2019.

Zadatak zadao:

Prof. dr. sc. Željko Šitum

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA.....	III
POPIS TEHNIČKE DOKUMENTACIJE .....	IV
POPIS OZNAKA .....	V
SUMMARY .....	VII
1. UVOD.....	1
2. ROBOTSKI MANIPULATORI.....	2
2.1. Općenito o manipulatorima.....	2
2.2. Paletizirajući manipulator .....	3
3. MAKETA MANIPULATORA .....	4
4. PNEUMATSKI POGON MANIPULATORA.....	6
4.1. Ventili.....	6
4.2. Prihvatnice .....	8
5. UPRAVLJAČKI UREĐAJ MANIPULATORA.....	9
5.1. Arduino Mega2560 .....	9
5.2. Arduino WiFi Shield.....	11
5.3. Arduino IDE.....	12
5.4. Povezivanje Arduina s računalom.....	13
5.5. Općenito o pisanju koda u Arduino IDE programu .....	14
5.6. Arduino Serial Monitor .....	14
5.7. Konfiguracija pinova.....	15
5.7.1. Visoka i niska vrijednost.....	15
5.7.2. Output i input način .....	16
6. PRILAGODBA UPRAVLJAČKOG SIGNALA .....	17
6.1. Prilagodba signala i galvanska zaštita.....	17
7. OPSKRBA SUSTAVA ENERGIJOM.....	18
7.1. Napajanje manipulatora .....	18
7.2. Napajanje kontrolera .....	19
7.2.1. Stabilizator napona.....	20
8. CIJENA I ISPLATIVOST .....	22
9. ZAKLJUČAK.....	23
LITERATURA.....	24
PRILOZI.....	25

---

**POPIS SLIKA**

Slika 1.	Gibanja manipulatora [2].....	2
Slika 2.	Pneumatski manipulator .....	5
Slika 3.	Kinematska struktura manipulatora [1].....	5
Slika 4.	Elektropneumatska shema manipulatora [1] .....	7
Slika 5.	Pneumatska shema vakuumskih prihvatnica [1] .....	8
Slika 6.	Arduino Mega2560.....	9
Slika 7.	WiFi modul ESP8266.....	11
Slika 8.	Arduino IDE sučelje .....	12
Slika 9.	Odabir pločice u Arduino IDE .....	13
Slika 10.	Namještanje serial monitora u Arduino IDE.....	15
Slika 11.	Definiranje input i output pinova .....	16
Slika 12.	8-segmentni relejni modul s optokaplerima .....	17
Slika 13.	Transformator 220V AC to 24V DC 4A .....	18
Slika 14.	Shema regulatora napona 7808 .....	20
Slika 15.	Stabilizator napona 7808 .....	21
Slika 16.	Shema stabilizatora 7808.....	21

## **POPIS TABLICA**

Tablica 1. Karakteristike ventila [5].....	6
Tablica 2. Tehničke specifikacije za Arduino Mega2560 [4] .....	10
Tablica 3. Cijena komponenata .....	22



## **POPIS TEHNIČKE DOKUMENTACIJE**

Shema spajanja

## POPIS OZNAKA

Oznaka	Jedinica	Opis
I	A	Jakost struje
U	V	Napon
p	Pa	Tlak
C	F	Kapacitet kondenzatora
L	H	Induktivitet zavojnice

## **SAŽETAK**

Ovaj završni rad je zapravo modernizacija 25 godina starog pneumatskog manipulatora koji je izrađen u tvrtki FESTO u edukacijske svrhe. Upravljanje je bilo izvedeno pomoću programabilnog logičkog kontrolera (PLC) proizvođača FESTO. S obzirom da navedeni PLC više nije u upotrebljivom stanju, potrebno ga je zamijeniti novim. Međutim, s vremenom je došlo do razvoja novih upravljačkih uređaja pomoću kojih bi se moglo izvesti samo upravljanje manipulatorom. Glavni razlog zašto bismo to radili jest cijena, a i s obzirom da se koristi na fakultetu, može služiti kao eksperiment koliko je to isplativo. PLC se u industriji zbog svoje kompaktnosti i praktičnosti koristi gotovo neizbježno, ali također ima i neke svoje zahtjeve. Stoga uz samu praktičnu izradu rad će se fokusirati i na usporedbu PLC-a i jednostavnijeg mikrokontrolera, a to će u ovom projektu biti Arduino Mega. Uz PLC analiza će se također baviti i ostalim komponentama koje su potrebne za upravljanje manipulatorom. Zbog starosti cijelog postava potrebno je ispitati svu elektroniku, te pneumatske uređaje poput ventila, prigušnica, vodova. S obzirom da novi mikrokontroler ne zadovoljava sve ono što PLC može, a to je u glavnom slučaju nemogućnost slanja i primanja 24-voltog signala, potrebno je i izraditi nove dijelove kućišta u svrhu kompaktnosti i urednosti cijelog postava.

Ključne riječi:

Elektropneumatika, manipulator, Arduino, PLC

## **SUMMARY**

This final paper is actually a modernization of a 25-year-old pneumatic manipulator that was made in FESTO company for educational purposes. The control was performed using the FESTO programmable logic controller (PLC). Since the specified PLC is no longer in serviceable condition, it must be replaced with a new one. However, over time, new control devices were developed that could only be used to manipulate the manipulator. The main reason why we do this is because of the cost, and given that it is used in college, it can serve as an experiment for how cost-effective it is. PLC is used almost inevitably in the industry because of its compactness and convenience, but it is also looking for some of its requirements. Therefore, with the practical development itself, this work will also focus on comparing PLCs and simpler microcontrollers, which in this project will be the Arduino Mega. In addition to the PLC, the analysis will also address the other components that are required to operate the manipulator. Due to the age of the entire setup, it is necessary to test all electronics and pneumatic devices such as valves, throttles, lines. Since the new microcontroller does not meet all that the PLC can do, and in the main case is the inability to send and receive a 24-volt signal, it is also necessary to create new housing components for the reason of compactness and tidiness of the entire setup.

Key words: Electropneumatics, manipulator, Arduino, PLC

## **1. UVOD**

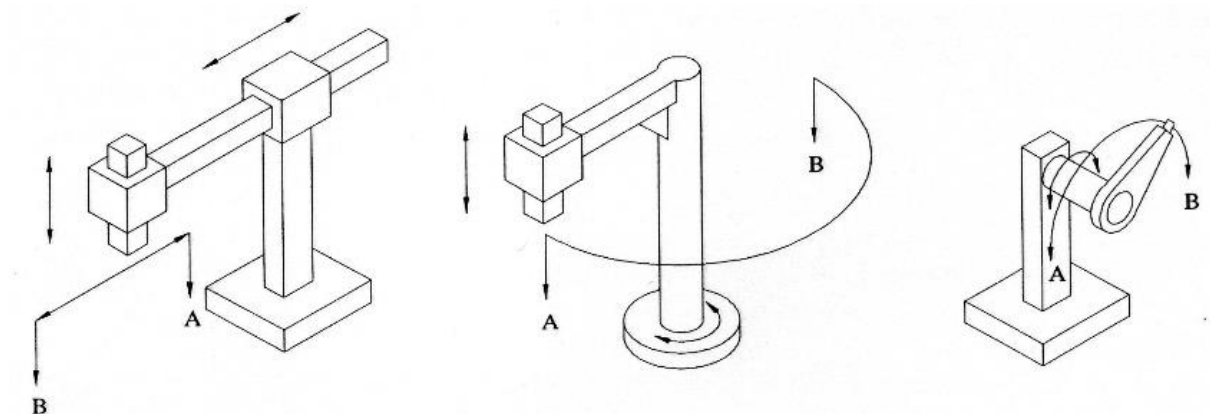
Svijet proizvođača osnovnih robotičkih komponenata brzo raste, pri čemu tvrtke uz hardver razvijaju i softvere za podršku raznih vrsta aplikacija. Pri tome se pojavio veliki broj raznih mikrokontrolera koji mogu izvršiti široku paletu funkcija uz vrlo male troškove. Na tržištu danas postoji veliki broj raznih pločica koja uključuju mikrokontrolere, FPGA uređaje, te računala s jednom pločicom. Među njima se najviše ističu Arduino i Raspberry Pi. Oba su open source, te su im komponente dostupne kod različitih proizvođača. Ipak, za njih je potrebna poprilično visoka razina programiranja, i snalažljivost prije nego što se mogu upotrijebiti u industriji. Zbog toga bi bilo potrebno ispitati može li se jedan industrijski PLC zamijeniti jednim takvim uređajem. Ako je originalni Arduino Mega moguće kupiti za nekih tristotinjak kuna, a njegove kopije i za znatno nižu cijenu, čemu trošiti tisuće kuna na PLC?

## 2. ROBOTSKI MANIPULATORI

### 2.1. Općenito o manipulatorima

Manipulatori su *mehanizmi s otvorenim kinematičkim lancem, pogonjeni pneumatskim, hidrauličkim ili električkim prigonima*. Manipulator uzima predmet s trake ili palete pomoću hvataljke koja može biti mehanička, vakuumska ili magnetna, ovisno o predmetu te ga premješta na drugo mjesto. Sam ciklus se odvija prema zadanom taktu ili s obzirom na upravljački signal.

Sama struktura manipulatora uzmi-stavi (eng. pick and place) manipulatora je određena brojem stupnjeva slobode gibanja. Stupnjevi slobode gibanja mogu biti rotacijski ili translacijski.



Slika 1. Gibanja manipulatora [2]

Mogućnost pozicioniranja hvataljke ovisi o kinematičkoj strukturi i načinu upravljanja manipulatora. Zbog toga je potrebno imati mogućnost prilagođavanja razmještaja opreme. U mnogim zadacima manipulatori mogu izvršiti posao jednako dobro kao i industrijski roboti, a imaju znatno nižu cijenu i zauzimaju manje prostora.

## **2.2. Paletizirajući manipulator**

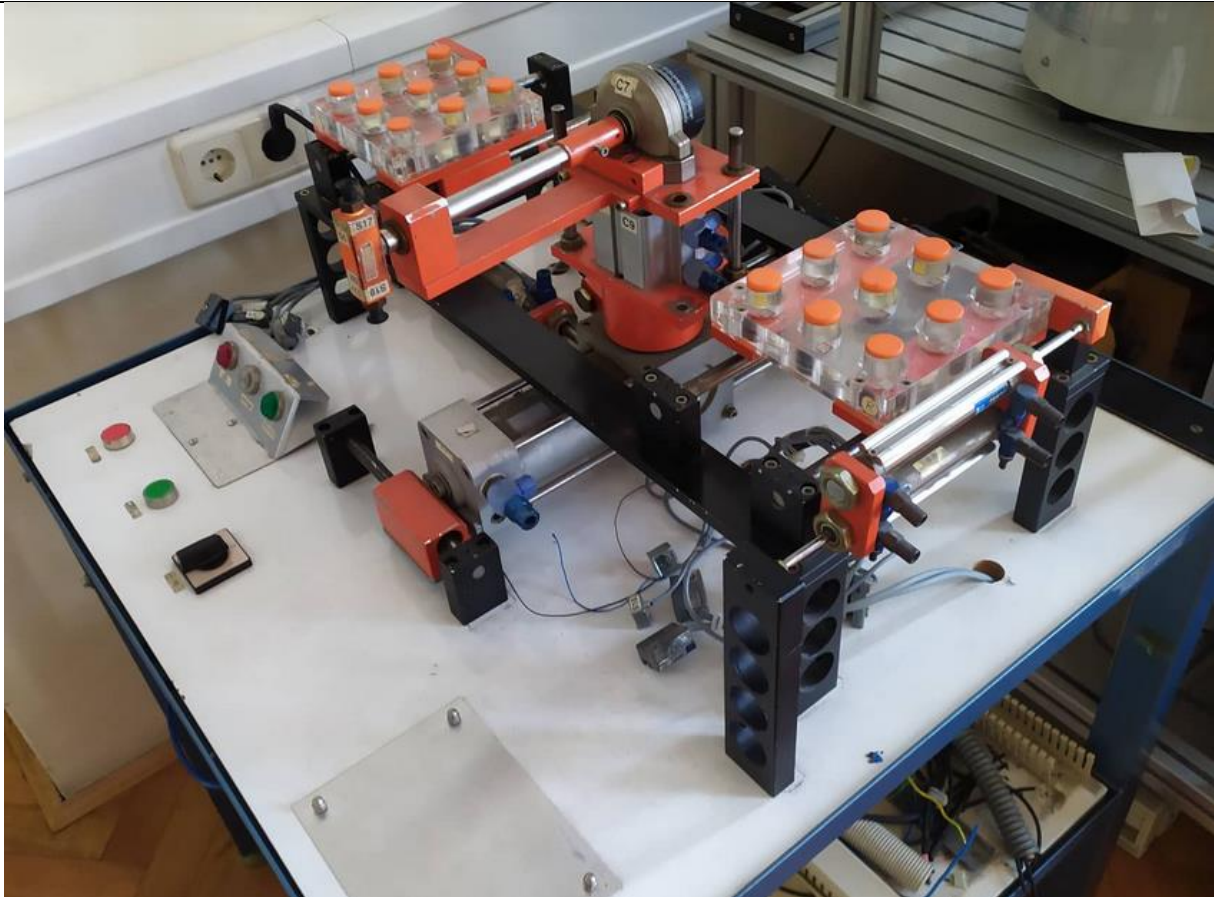
Paletizirajući robot je uobičajen tip manipulatora koji se često koristi u industriji za pomicanje teških tereta te izvršavanje monotonog rada umjesto da to obavlja čovjek. Ima mnogo prednosti poput fleksibilnosti, jednostavnog održavanja, malog prostora potrebnog za nesmetano djelovanje. Manipulator uz aktivne može imati i pasivne zglobove. Aktivni zglobovi su direktno vezani za aktuatore, dok se pasivni gibaju u ovisnosti o ostalim aktivnim i pasivnim zglobovima. Za posljedicu, matematički model takvih robota može biti iznimno kompliciran i težak za izvođenje. Za takve stvari je potrebno izraditi pojednostavljenu strukturu robota iz koje će se odrediti kinematske jednadžbe.

### 3. MAKETA MANIPULATORA

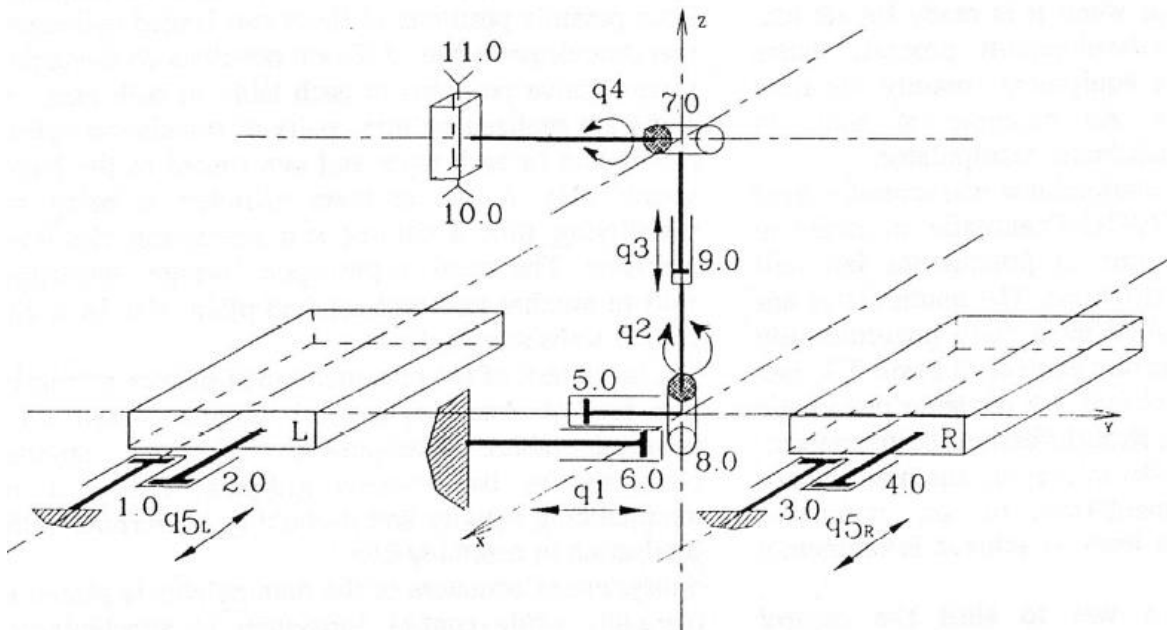
Sama maketa se može podijeliti u dva velika dijela. Prvi dio bi bila dva pokretna stolića na kojima stoje valjčići koji predstavljaju radne predmete. Na svakom stoliću može biti najviše 9 valjčića, poredanih u 3 reda. Svaki stol se može pomicati paralelno u odnosu na drugi, na način da su povezani s po dva cilindra u seriju, te time, s obzirom da su cilindri uvijek u krajnjim pozicijama, je moguće svakim stolom doći u tri pozicije. U prvoj poziciji su oba cilindra uvučena, u drugoj (srednjoj) je jedan od cilindara izvučen, a u trećoj su oba cilindra izvučena. Na taj način stolovi dozvoljavaju samom manipulatoru koji prebacuje predmete manji broj stupnjeva slobode gibanja.

Drugi dio se sastoji od same „ruke“ koja se sastoji od četiri stupnja slobode gibanja, dva translacijska i dva rotacijska. Za svaki stupanj slobode gibanja je zaslužan po jedan cilindar, osim za prvi translacijski stupanj koji je izveden slično kao i prije spomenuti stolići. Drugi stupanj slobode je rotacijski, te je izveden pomoću Festovog cilindra koji omogućava rotacijsko gibanje. Samo rotacijsko gibanje je izvedivo u rasponu od 180°. Zbog toga je bilo potrebno postaviti cilindar na takav način da u svojim krajnjim položajima bude točno iznad stolića. Treći stupanj slobode omogućuje vertikalno pomicanje prihvatnice. Kada god se neki od dijelova kreće potrebno je da taj cilindar bude potpuno izvučen, jer bi inače došlo do sudaranja predmeta sa stolićem ili s drugim predmetom. Zbog mogućih sudara potrebno je da taj cilindar gotovo uvijek tokom rada bude izvučen, osim prilikom uzimanja i ostavljanja dijelova. Posljednji stupanj slobode gibanja je rotacijski. Da bi se ubrzao cijeli proces prenošenja tih dijelova, na manipulatoru se nalaze dvije vakuumske prihvatnice koje se nalaze na suprotnim stranama. Zbog toga taj posljednji stupanj slobode omogućuje okretanje prihvatnica tako da nakon što prva prihvatnica uzme predmet, okreće se prema gore dok se druga okreće prema dolje te uzima drugi predmet.





Slika 2. Pneumatski manipulator



Slika 3. Kinematska struktura manipulatora [1]

## 4. PNEUMATSKI POGON MANIPULATORA

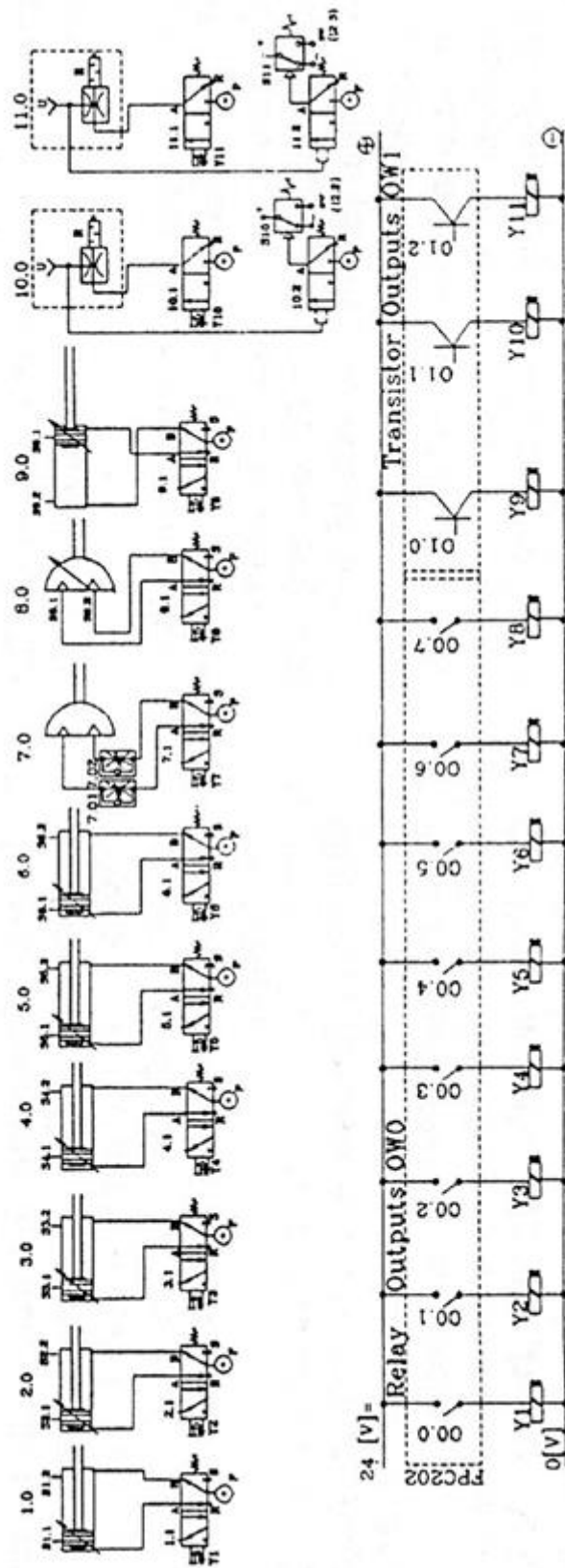
### 4.1. Ventili

Da bi mogli koristiti energiju stlačenog zraka, moramo taj stlačeni zrak usmjeravati u određenom smjeru. Najčešće se za to koriste pneumatski ventili. Pomoću njih možemo propuštati, mijenjati smjer i zaustavljati radni medij. Uglavnom su izvedeni s elektromagnetskim komponentama.

Ventili koje sadrži maketa su 9 elektromagnetskih FESTO-MFH-5/2 ventila, te dva 3/2 ventila. Sami ventili nisu bili u nekom zavidnom stanju, s obzirom da nisu korišteni dugo vremena. Prilikom njihovog ispitivanja neki od ventila se nisu mogli preklapati, pa ih je bilo potrebno otvarati jer elektromagneti nisu bili dovoljno jaki da ih preklapaju. Također ispitao sam i jesu li potpuno ispravni s obzirom na struju. U njihovom katalogu stoji da preklapaju na iznosu struje od oko 0.2A, što je i odgovaralo mjerenjima. Na svakom od njih se nalazi LED lampica pomoću koje se može vidjeti u kojem je stanju ventil. Također, otkrio sam da nije bitno kako je ventil spojen, jer ako dobiva struju on će preklapati, bez obzira na njen smjer. Što se tiče propuštanja zraka, tj. brtvljenja, sam 5/2 ventil nikad ne zapire zrak, tako da kod njih neće dolaziti do većih problema s obzirom na to da su dosta stari. S druge strane, 3/2 ventile bi možda bilo bolje zamijeniti jer bi protok zraka bio bolje zatvaran, ovako dolazi do propuštanja.

**Tablica 1. Karakteristike ventila [5]**

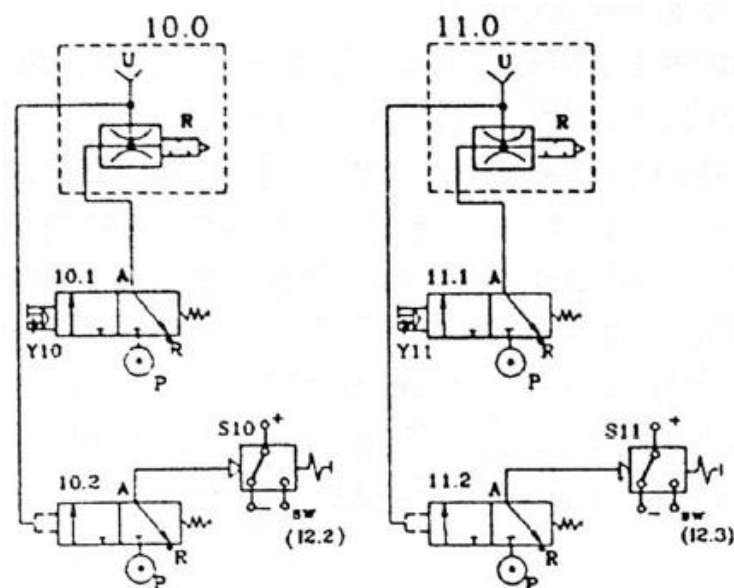
Tip ventila	5/2 monostabilni
Način prebacivanja	Električni
Protok [l/min]	1200
Radni tlak [bar]	2-10
Način vraćanja	Pneumatska opruga
Radni medij	Stlačeni zrak
Temperatura medija [°C]	-10 do 60
Radna temperatura [°C]	-5 do 40
Masa [kg]	0.4



Slika 4. Elektropneumatska shema manipulatora [1]

## 4.2. Prihvatnice

Za uzimanje predmeta s makete se koriste dvije vakuumske prihvatnice koje su povezane s prigušnicom te 3/2 ventilima. U ovom slučaju su ti ventili idealni, jer imaju položaj usisavanja ili zapiranja zraka, dok nije potrebno tjerati zrak u suprotnom smjeru.



Slika 5. Pneumatska shema vakuumskih prihvatnica [1]

Na slici iznad je prikazana pneumatska shema spajanja podsustava prihvatnica. Svaka vakumska prihvatnica pripada svom podsustavu. Svaki taj podsustav ima mogućnost detektiranja vrijednosti vakuuma, ili tlaka. To znači da ako vakumska prihvatnica dođe u doticaj s ravnom plohom, vakuum je u pozitivnom stanju te se očitava prihvaćanje predmeta. Element koji daje taj signal je pneumatski ventil koji otkriva vakuum (10.2 i 11.2). Taj signal je zatim potrebno pretvoriti u električki oblik. Za to se ovdje koriste mikrosklopke VPE-2N (S10 i S11) koje šalju digitalni signal prema kontroleru.

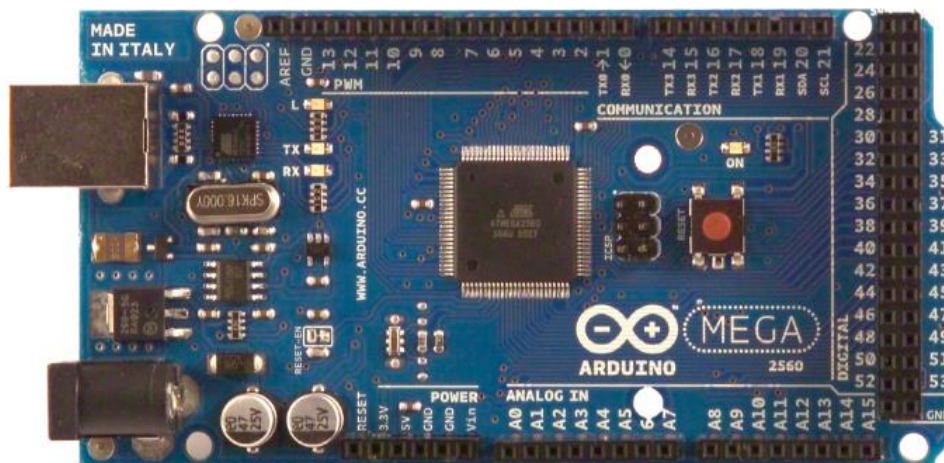
## 5. UPRAVLJAČKI UREĐAJ MANIPULATORA

Mikrokontroler Arduino, u ovom slučaju Arduino Mega, može se koristiti za različite upravljačke funkcije, ali postići da radi čak i u jednostavnijoj primjeni u industriji ipak nije toliko jednostavno. Arduino je „prazna ploča“, što je i prikladno njegovoj cijeni, ali ima široke mogućnosti ukoliko je moguće napisati program koji će odgovarati aplikaciji. Ovaj mikrokontroler nema funkcionalne blokove spremne za prijenos, tako da je za njega potrebno ispočetka stvoriti neki upravljački algoritam.

Također jedan veliki nedostatak kod Arduina je taj što radi s niskim naponima (0-5V) koji su često nedovoljni za upravljanje industrijskim uređajima.

### 5.1. Arduino Mega2560

Arduino Mega 2560 je mikrokontroler koji se temelji na mikroprocesoru Atmega2560. Ima 54 digitalnih ulazno izlaznih pinova, od koji se 15 može koristiti kao PWM izlazi, 16 analognih ulaza.



Slika 6. Arduino Mega2560

**Tablica 2. Tehničke specifikacije za Arduino Mega2560 [4]**

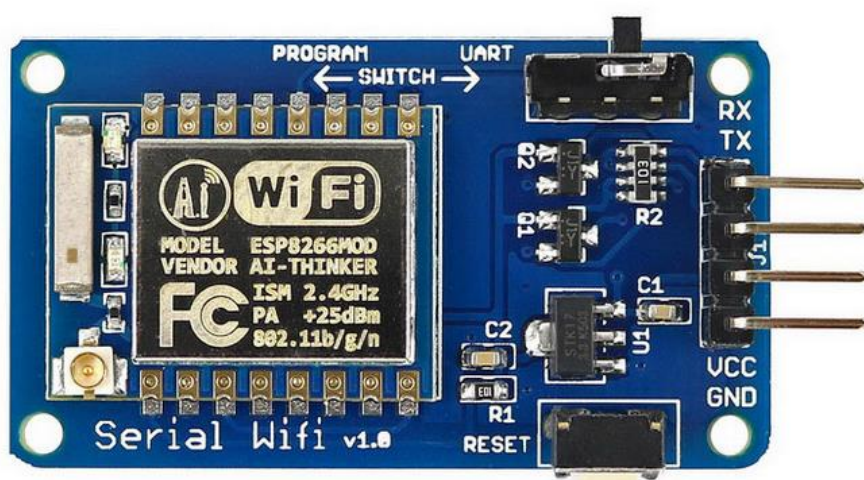
Mikroprocesor	Atmega2560
Radni napon	5V
Ulazni napon	7-12 V
USB Port	USB B
Broj digitalnih I/O pinova	54
PWM pinova	15
Broj Analognih pinova	16 (ujedno su to digitalni I/O pinovi)
Flash memorija	256 kB
WiFi	Ne sadrži
Kristal	16 MHz

Arduino Mega2560 je dizajniran tako da može biti resetiran pomoću softvera koji radi na računalu. Jedna od kontrolnih linija (DTR) je povezana s reset linijom na kontroleru. Kad ta linija dođe u nisko stanje, čip se resetira. Pomoću toga Arduino IDE može učitati kod jednostavnim klikom na gumb za učitavanje. Na taj način *bootloader* može imati kraći *timeout*.

Kada je Mega2560 povezan s ostalim operacijskim sustavom osim *Windowsa*, resetira se svaki puta kada se pomoću USBa poveže sa softverom. Kako je programiran da ignorira promijenjen sadržaj dok mu se vrti petlja, neće odmah prihvatiti novi kod. Zbog toga je i sam softver izrađen tako da pričekava neko kraće vrijeme da ne bi došlo do smetnji u radu prilikom izvršenja prethodnog koda.

## 5.2. Arduino WiFi Shield

Arduino Wifi Shield služi za povezivanje Arduino uređaja na bežičnu mrežu (*eng. wireless*). To povezivanje se ostvaruje tako da se WiFi Shield spoji na pin na zaglavljju Arduino pločice. WiFi Shield na sebi također ima i utor za umetanje *micro-SD* kartice pomoću koje se mogu pohranjivati datoteke koje se koriste za posluživanje preko mreže, zatim *Mini USB* priključak, *ICSP header* te četiri LED indikatora.



Slika 7. WiFi modul ESP8266

Svi Arduino kompatibilni WiFi moduli moraju biti napajani s 3.3 V. Potrebno je biti oprezan kada se koristi Arduino kontroler koji radi na 5V, jer modul može lako biti uništen spajanjem na taj napon. Zbog toga napajanje modula mora ići preko 3.3 V pina na mikrokontroleru.

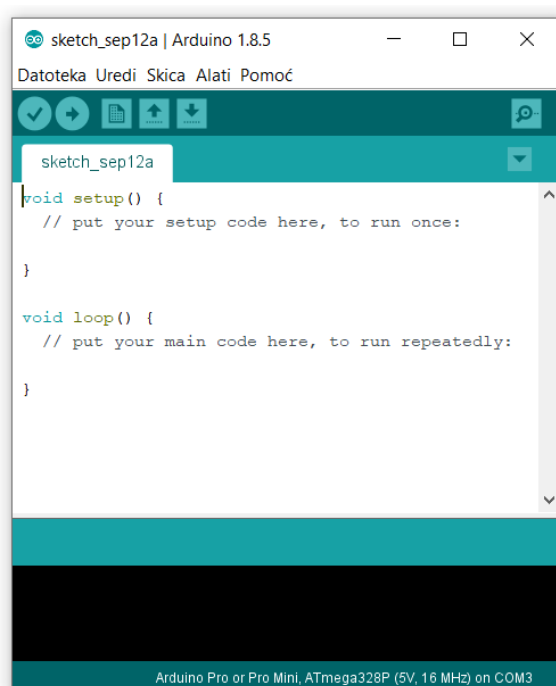
### 5.3. Arduino IDE

Arduino IDE je open source softver koji se uglavnom koristi za pisanje i prevođenje koda na Arduino uređaje. To je službeni Arduino softver koji kompajlira kod na način da svatko, bez nekog značajnijeg znanja programiranja može nešto isprogramirati.

Softver je dostupan za instalaciju na operacijske sustave Windows, MAC i Linux. Može se besplatno skinuti sa službene stranice Arduina. Radi na Java platformi te dolazi s ugrađenim funkcijama i naredbama koje igraju značajnu ulogu u uređivanju, otklanjanju grešaka i prevođenju koda.

Glavni kod, poznat i kao sketch, koji je stvoren na IDE platformi će imati svoju Hex datoteku koja će se zatim prebaciti i učitati na kontroler na pločici.

Samo IDE okruženje se uglavnom sastoji od dva glavna dijela: editora i kompajlera. U editoru pišemo i uređujemo kod, te otklanjamo greške koje nam i sam program može otkriti. Nakon toga kompajler taj kod prilagođuje i učitava na Arduino. IDE okruženje podržava C i C++ jezike.

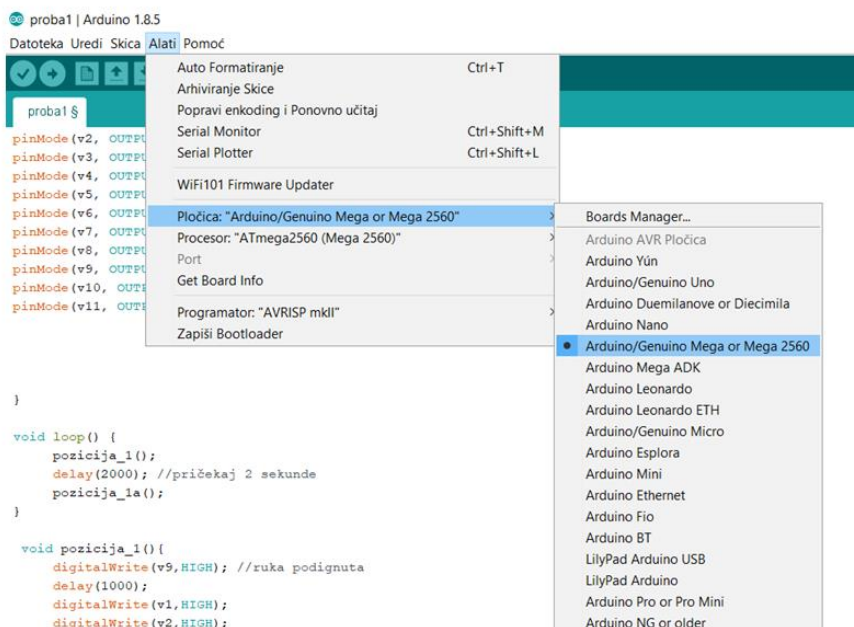


Slika 8. Arduino IDE sučelje



## 5.4. Povezivanje Arduina s računalom

Arduino se s računalom najjednostavnije povezuje pomoću USB kabela. Kod većih pločica poput Una ili Mege dovoljno je koristiti USB B kabel. Nakon što smo povezali i provjerili da je sve u redu, potrebno je u Arduino IDE programu namjestiti određene postavke. Prvo u meniju u izborniku alati je potrebno odrediti tip Arduino uređaja. Popis koji se nudi je poprilično velik, tako da je velika vjerojatnost da će jedan odgovarati. Također, ako pločica nije originalna, nego neka kopija, ne bi trebalo dolaziti do problema u radu.



Slika 9. Odabir pločice u Arduino IDE

Posljednja stvar koju je potrebno učiniti jest naći na kojem je COM portu povezan Arduino, te u programu za prebacivanje programa odabrati taj port. Obično ih bude više ponuđeno u izborniku alati, a za svaki slučaj, se točan port može naći u upravitelju uređajima koji se nalazi u postavkama računala.

## 5.5. Općenito o pisanju koda u Arduino IDE programu

Prilikom pisanja koda potrebno je imati na umu da je podatkovna memorija na kontroleru dosta malena. Zbog toga je potrebno koristiti minimalne veličine podataka kojima se može uspješno odraditi zadatak. Također je dobro izbjegavati alokaciju memorije u svrhu što boljeg skladištenja, na taj način će biti najbolje iskoristiva.

Postoje dvije glavne funkcije prilikom pisanja koda:

`setup()` - služi za inicijaliziranje trajnih postavki, npr određivanje ulaznih i izlaznih pinova. Sadržaj te funkcije će se izvršiti samo jednom te se neće mijenjati.

`loop()` - služi za pisanje koda koji će se izvršavati u petlji dok god je kontroler uključen.

Arduino platforma je vrlo praktična za korištenje periferije. Ako želimo očitavati stanje s nekog senzora, pogotovo ako radi s digitalnim signalom, možemo vrlo jednostavno pomoću određenih funkcija čitati njegovo stanje.

## 5.6. Arduino Serial Monitor

Ako testiramo neki novi kod potrebno je znati što se točno događa kada ga pokrećemo. Ali softver je „nevidljiv“. Zbog toga trebamo imati program koji će reći trenutno stanje, a ponekad samo želimo znati vrijednost varijabli koje se mijenjaju.

U `setup()` funkciji potrebno je upisati `Serial.begin(9600)`; Ta naredba govori Arduino da počne komunicirati putem teksta sa stopom od 9600 bitova u sekundi.

U `loop()` funkciji ćemo upisati `Serial.println(stanje_p1,stanje_p2)`; te , Pomoću toga će se ispisivati stanje vrijednosti na pinovima koji služe za detekciju vakuuma.

```
Serial.begin(9600); //inicijalizira serijsku komunikaciju s 9600 bitova/s

}

void loop() {

int stanje_p1 = digitalRead(p1);
int stanje_p2 = digitalRead(p2);

Serial.println(stanje_p1,stanje_p2); //ispisivanje stanja na ulaznim pinovima s prihvatnica
```

**Slika 10. Namještanje serial monitora u Arduino IDE**

## 5.7. Konfiguracija pinova

Arduino (Atmega) pinovi mogu biti konfigurirani kao ulazni ili izlazni, te postavljeni u visoku (HIGH) ili nisku (LOW) vrijednost.

### 5.7.1. Visoka i niska vrijednost

Kada je pin konfiguriran kao izlazni (OUTPUT) pomoću naredbe `pinMode()`, te u vrijednost HIGH pomoću `digitalWrite()`, na tom pinu napon će biti 5V, odnosno 3.3V (ako se radi o pločici koja radi na 3.3V, poput Arduino Due). U tom stanju se pločica ponaša kao izvor, te serijom s otpornikom se povezuje na ground pin.

Što se tiče ulaznih (INPUT) pinova, ako je vrijednost napona veća od 3.0V na 5V-verzijama ili 2.0V na 3.3V-verzijama, naredbom `digitalRead()` Arduino će prijaviti HIGH vrijednost na tom pinu.

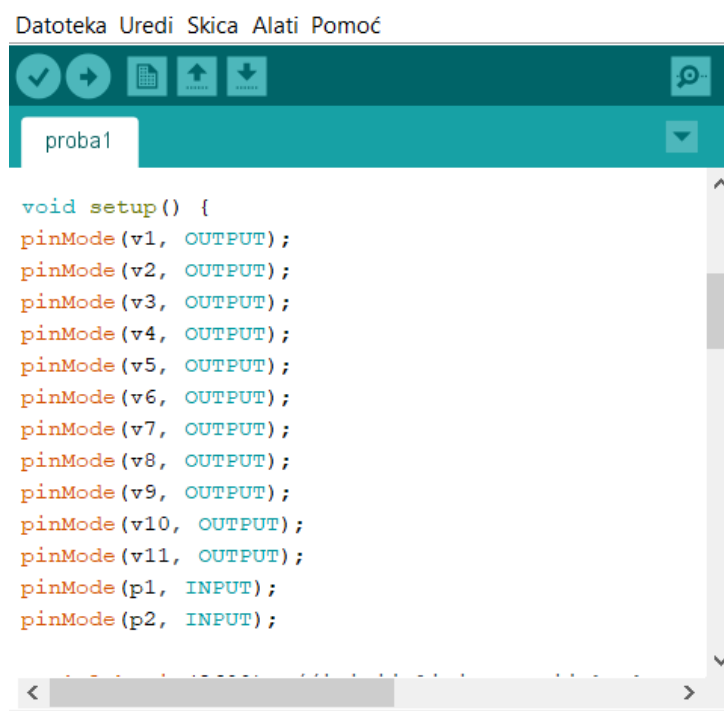
Značenje LOW također ovisi o tome je li pin postavljen kao ulazni ili izlazni. Ako je postavljen kao izlazni, na tom pinu će napon biti 0V. U ovom stanju može povlačiti struju ako je preko trošila povezan s visokim naponom.

Kod ulaznih pinova, Arduino će detektirati nisko stanje ukoliko je na pinu vrijednost napona manja od 1.5V, ili 1V kod 3.3V-verzija.

### 5.7.2. Output i input način

Pinovi konfigurirani kao OUTPUT pomoću naredbe `pinMode()` su u stanju niske impedancije. To znači da oni mogu pružiti znatnu količinu struje drugim krugovima. Atmega pinovi mogu davati ili primiti struju do 40 mA od drugih uređaja ili krugova. Za struje veće od 40mA je potrebno koristiti tranzistore ili ostale pojačivače struje.

Što se tiče INPUT pinova, oni su tako već i postavljeni, stoga ih nije potrebno posebno definirati kao takve. Njihovo stanje impedancije je visoko. To znači da će koristiti iznimno malo struje da prebace stanje na pinu iz jednog u drugo. Zbog toga su praktični u radu s kapacitivnim i analognim senzorima. To što radi s malim strujama ima posljedicu da ako pin koji je definiran kao input nije povezan s ničim, skupljanjem šuma sa susjednih pinova, može javljati znatne nasumične promjene stanja na tom pinu.

The image shows a screenshot of an IDE window titled "Datoteka Uredi Skica Alati Pomoć". The window contains a code editor with the following C++ code:

```
void setup() {  
  pinMode(v1, OUTPUT);  
  pinMode(v2, OUTPUT);  
  pinMode(v3, OUTPUT);  
  pinMode(v4, OUTPUT);  
  pinMode(v5, OUTPUT);  
  pinMode(v6, OUTPUT);  
  pinMode(v7, OUTPUT);  
  pinMode(v8, OUTPUT);  
  pinMode(v9, OUTPUT);  
  pinMode(v10, OUTPUT);  
  pinMode(v11, OUTPUT);  
  pinMode(p1, INPUT);  
  pinMode(p2, INPUT);  
}
```

The code is color-coded: keywords like `void` and `setup` are in green, `pinMode` is in orange, and `OUTPUT` and `INPUT` are in blue. The editor has a teal header bar with a tab labeled "proba1".

Slika 11. Definiranje input i output pinova

## 6. PRILAGODBA UPRAVLJAČKOG SIGNALA

### 6.1. Prilagodba signala i galvanska zaštita

Da bi bilo moguće uspješno slati i primiti signal između uređaja, potrebno ga je na neki način zaštititi od smetnji. Također nije poželjno da imamo galvanski povezane uređaje koji nisu istog napona. Može doći do proboja i oštećenja samog ponajprije kontrolera koji radi na nižem naponu. S obzirom da to da Arduino radi na 5V, a ventili na 24V, moramo nekako taj signal prenijeti s jednog na drugi uređaj. Za takve slučajeve se koriste releji s optocouplerima. Optocoupler je zapravo veza između dva kruga koji rade na uglavnom različitim naponima. Glavna prednost optocouplera jest električna izolacija između ulaznog i izlaznog kruga. Jedini kontakt između njih se postiže preko svjetlosne zrake diode. Zbog toga je moguće imati izolaciju između dva kruga reda gigaohma. Takve izolacije se koriste pri visokonaponskim uređajima gdje potencijal između dva kruga može biti i po nekoliko tisuća volti.

Dva glavna razloga zašto koristiti optoizolatore:

- Ako je kontroler jako udaljen pa ne dijeli zajedničko uzemljenje s relejnom pločicom (osim ako je povezano dugačkim kabelom). Korištenjem optoizolacije razlike u potencijalu uzemljenja neće imati utjecaj na signal.
- Napon na zavojnici releja nije nužno jednak naponu na kontroleru. Čak bi i mogao biti induciran iz nekog izvora koji nije izoliran. Stoga optoizolator izolira potencijalno neizolirane izvore od kruga samog kontrolera.

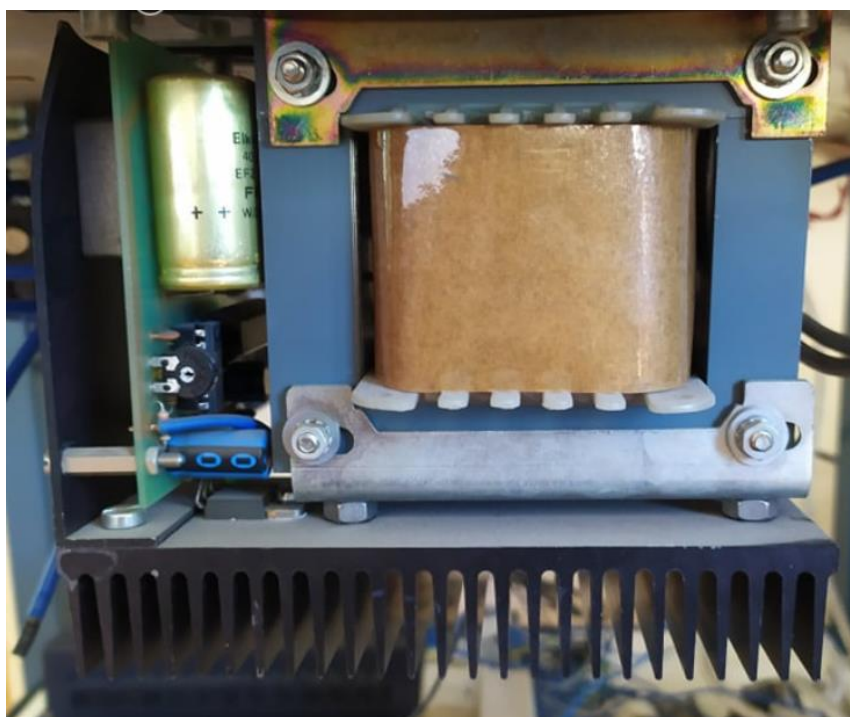


Slika 12. 8-segmentni relejni modul s optokaplerima

## 7. OPSKRBA SUSTAVA ENERGIJOM

### 7.1. Napajanje manipulatora

S donje strane stola se nalazi transformator koji pretvara 220V AC na 24V DC jakosti 4A. S obzirom na godine, više nije u mogućnosti raditi na nazivnoj struji jer dolazi do pregrijavanja. Naravno, potrebno je bilo uzeti u obzir da je prije PLC bio povezan direktno na 220V tako da nije koristio struju koja ide kroz transformator. Zbog toga je potrebno bilo reducirati broj potrošača koliko god je moguće iz razloga što nove komponente ne koriste ništa direktno s gradske mreže. Ventili troše relativno malo struje (oko 0.18A), a ta vrijednost pomnožena s njihovim brojem iznosi gotovo 2A, što je ipak znatno. Iako gotovo nikad neće biti uključeni svi ventili odjednom, ponekad će dolaziti do te vrijednosti. Prilikom izrade je razmatrana i nabavka novog napajanja, zbog toga što je i došlo do pregaranja osigurača na samom transformatoru, ali zasada je odlučeno da se ostane pri njemu. Ukoliko bude neophodno potreban, neće biti veliki problem ugraditi novi transformator, jer ima dovoljno mjesta na donjoj strani, a i vjerojatno će koristiti modernije osigurače.



Slika 13. Transformator 220V AC to 24V DC 4A

## 7.2. Napajanje kontrolera

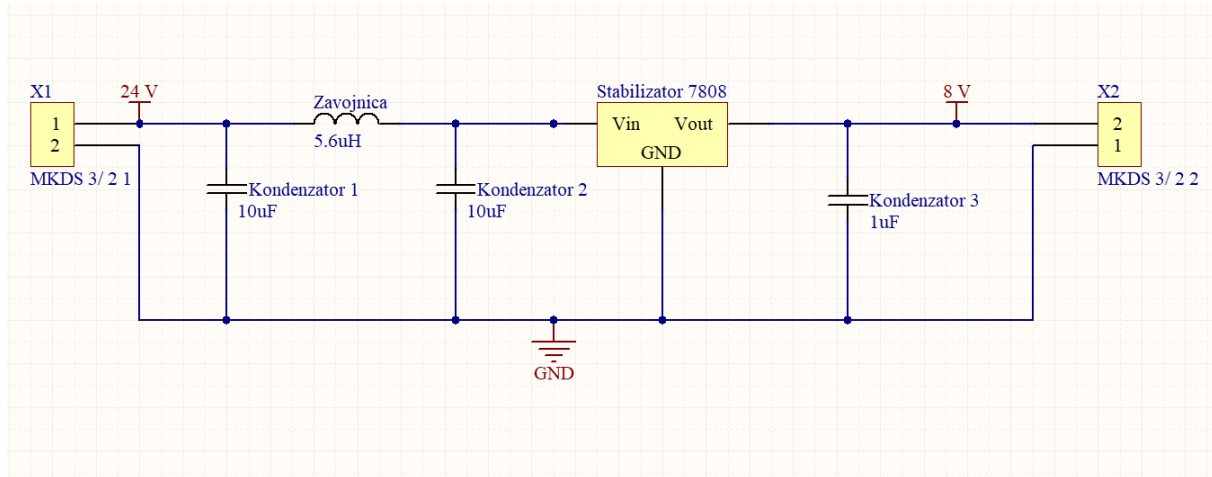
Veliki nedostatak Arduino pločica, kao i mnogih drugih kontrolera u odnosu na PLC je taj što rade na pre niskom naponu za veliki broj industrijskih postrojenja. Kao što je već spomenuto, Arduino radi na 5 ili 3.3 V, dok ulazi i izlazi na PLCu rade na 24 V. Također sam PLC se može spojiti direktno na napon 220 V, pa mu za napajanje nije potreban nikakav vanjski pretvarač napona. PLC ima vlastiti pretvarač napona koji pretvara 220 V ili 110 V AC, na 24 V DC.

Da bismo napajali Arduino mikrokontroler, nikako ga ne smijemo izravno spajati na gradsku mrežu. Arduino se može napajati samo s DC strujom, a, napon na koji ga možemo spajati iznosi maksimalno 12 V. I tih 12 V smijemo spajati samo na konektor napajanja. Dok primjerice, 5 V se može dovesti i na pin s oznakom 5 V, a zatim ga uzemljimo preko pina s oznakom GND.

Naravno u određenim slučajevima tih 5V je i prednost. Ako radimo s USB kablovima, radimo s tolikom naponom. U tom slučaju nisu potrebni nikakvi drugi kablovi, poput *Ethernet* kabla, koji nam je potreban kod PLCa. Kod mobilnih robota je također lakše raditi s baterijama od 5 V, jer su uglavnom kompaktnije, lakše i dostupne u različitim oblicima.

Ako želimo Arduino napajati s računala putem USB kabla, treba imati na umu da se može dogoditi na mu računalo neće davati dovoljno struje, pogotovo ako se koristi USB 3.0 utor na računalu. Zbog toga je, ako se već koristi napajanje putem računala, najbolje koristiti utor za USB 2.0. Iz iskustva također savjetujem da je i za prijenos koda također najbolje koristiti taj utor. Napajanje putem računala također nije toliko praktično ako nam je samo zbog njega računalo potrebno, tj. ako nam nije potrebno za očitavanja, povezivanja s Matlabom itd. U tom slučaju su mnogo praktičnije npr. baterije od 9V ili prijenosni punjač za punjenje mobitela.

Za ovu maketu će napajanje mikrokontrolera biti izvedeno pomoću stabilizatora napona L7808. Na ulaz će mu dolaziti napon 24 V, a izlazni napon će mu biti 8 V. Dobro je koristiti malo viši napon od potrebnog na ulazu zbog stabilnijeg rada mikrokontrolera, pogotovo iz razloga što ostali uređaji koji se napajaju iz transformatora mogu „povući“ struju pa se na ovaj način osiguravamo od pada napona, a samim time i od neočekivanog resetiranja mikrokontrolera.

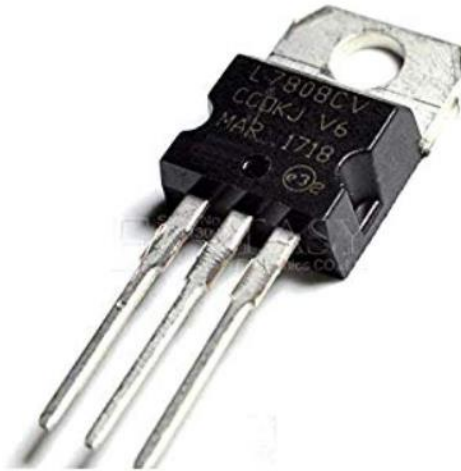


Slika 14. Shema regulatora napona 7808

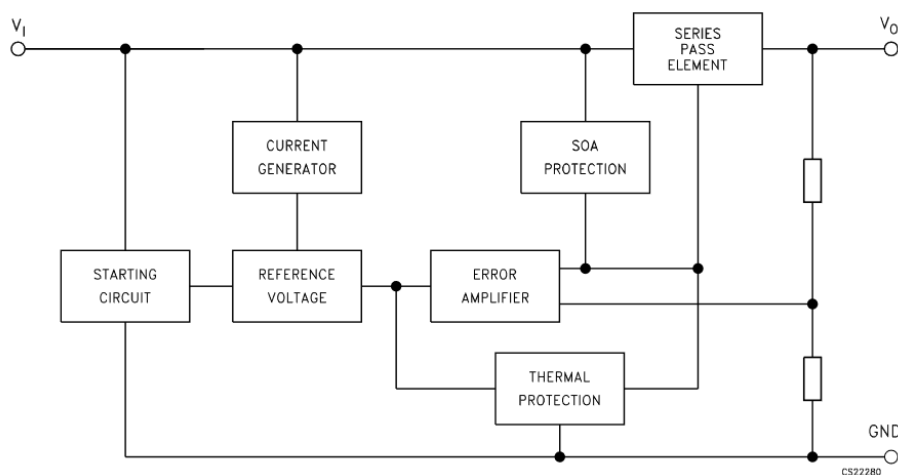
### 7.2.1. Stabilizator napona

Regulator napona iznad je baziran na stabilizatoru napona 7808. U paralelu s njim su spojeni kondenzatori od 10 i 1  $\mu\text{F}$ , zbog dodatne stabilnosti tu se nalazi i serijski spojena zavojnica od 5.6 mH. To je klasični način izrade regulatora napona koji se često upotrebljava u praksi. Sam stabilizator 7808 daje izlazni napon između 7.7 i 8.3V. Zbog toga je mu je potrebno u paralelu dodati kondenzatore koji će napon držati konstantnim oko 8V.





Slika 15. Stabilizator napona 7808



Slika 16. Shema stabilizatora 7808

Kao što je prikazano u shemi samog stabilizatora, on u sebi sadrži zaštitu i od prejake struje i temperature. Problem je kod mnogih komponenti koje se načelno bave provođenjem struje njihovo zagrijavanje. Zbog toga stabilizator s vanjske strane ima jedan metalni dio koji rasipa toplinu. S obzirom da će napajati Arduino, a preko njega i releje za ventile, moglo bi doći do znatnog zagrijavanja. Sam stabilizator ima nazivnu struju od 1.5mA što bi trebalo biti dovoljno, jer struja nikada neće biti jača od 1 A, čak ni kada su svi ventili u visokom stanju.

## 8. CIJENA I ISPLATIVOST

Uz sve to o čemu se u ovom radu govori, možda je najbitnije jedno pitanje: što je isplativije? Ovdje je bilo najviše napisano što i kako bi se to izvelo s Arduinoom, najviše iz tog razloga što PLC ipak ne zahtjeva toliko dodatnih komponenti, a da se i ne ponavljam s prethodnim radom o ovom manipulatoru. Naravno, PLC je mnogo jednostavniji za ugradnju, robusniji je, lakši za održavanje, ali je i mnogo skuplji. Isplati li se stoga raditi s Arduinoom? Pa rekao bih da ovisi. Ako se baziramo na glavne komponente finacijski trošak bi s opcijom koja koristi Arduino kao upravljački uređaj iznosila kao što je prikazano u donjoj tablici.

**Tablica 3. Cijena komponenata**

Arduino Mega2560	312 kn
Relejni moduli 2 kom	214 kn
Stabilizator	20 kn
WiFi modul za Arduino	30 kn
Tipkovnica 3x4	27 kn
Ukupno	603 kn

U tablici iznad su prikazane cijene originalnih uređaja u Chipoteci. Ako ne kupujemo originale, na ebayu bi sve skupa koštalo ne više od 200 kn.

Sam PLC s HMI-jem bi koštao više tisuća kuna. Naravno, radilo bi se o originalu. S obzirom da se radi o manipulatoru s kojim se radi na fakultetu, možda i nije toliko nužno plaćati toliko da bi se pokazala neka vježba. S druge strane, u industriji, gdje je bitno da ne dolazi do nikakvih zastoja, to može biti neznatna cijena.

## 9. ZAKLJUČAK

Dakle, što je bolje, Arduino ili PLC? Ako se uzmu u obzir samo troškovi hardvera za kontroler, pobjeđuje Arduino. No kada se dodaju sve pomoćne komponente potrebne da Arduino bude koristan u relativno jednostavnoj primjeni, razlika u troškovima hardvera će se smanjiti. Vrijeme potrebno za sastavljanje i programiranje Arduina je također veliko. Kad se ovo vrijeme izračuna po bilo čemu što je blizu uobičajenim inženjerskim stopama radnog vremena, PLC je jasan pobjednik u pogledu ukupnih troškova.

Za nekoga tko uči osnove pisanja koda i koncepata kontrole, Arduino i njegovi proizvodi pružaju zanimljive nastavne alate. Nužnost pisanja algoritama upravljanja ispočetka uzrokuje da korisnik razmotri sitnice kako se automatizacija obavlja. Dostupnost uređaja s ovom razinom sofisticiranosti uz tako male troškove prilično je izvanredna i korisna je onima koji imaju više vremena nego novca.

## LITERATURA

- [1] Nižetić, J., Petrić, J.: Pneumatic manipulator controlled by programmable logic controller, KoREMA Conference, Opatija, 1996.
- [2] Kunica, Z.: Roboti i manipulatori u montaži, nastava, FSB
- [3] <https://datasheetspdf.com/pdf-file/1162248/STMicroelectronics/L7808/1>
- [4] <https://store.arduino.cc/mega-2560-r3>
- [5] <https://www.festo.com/net/SupportPortal/Files/17267> (pristupljeno 2. 9. 2019.)
- [6] Reneker, D.: PLC vs. Arduino for industrial control, Control Design, 2017.

## **PRILOZI**

- I. CD-R disc
- II. Tehnička dokumentacija

```
int v1=1; // ventili 1-11
int v2=2;
int v3=3;
int v4=4;
int v5=5;
int v6=6;
int v7=7;
int v8=8;
int v9=9;
int v10=10;
int v11=11;
int p1=22; // prihvatnice 1-2
int p2=24;

void setup() {
  pinMode(v1, OUTPUT);
  pinMode(v2, OUTPUT);
  pinMode(v3, OUTPUT);
  pinMode(v4, OUTPUT);
  pinMode(v5, OUTPUT);
  pinMode(v6, OUTPUT);
  pinMode(v7, OUTPUT);
  pinMode(v8, OUTPUT);
  pinMode(v9, OUTPUT);
  pinMode(v10, OUTPUT);
  pinMode(v11, OUTPUT);
  pinMode(p1, INPUT);
  pinMode(p2, INPUT);

  Serial.begin(9600); //inicijalizira serijsku komunikaciju s 9600 bitova/s
}
```

---

```
}  
  
void pozicija_1a() {  
    podigni();  
    delay(1000);  
  
    digitalWrite(v1, LOW);  
    digitalWrite(v2, LOW);  
    digitalWrite(v5, LOW);  
    digitalWrite(v6, LOW);  
    digitalWrite(v8, LOW);  
}  
void pozicija_2a() {  
    pozicija_1a();  
    digitalWrite(v1, HIGH);  
}  
void pozicija_1b() {  
    digitalWrite(v9, HIGH); //ruka podignuta  
    delay(1000);  
    digitalWrite(v3, LOW);  
    digitalWrite(v4, LOW);  
    digitalWrite(v8, HIGH);  
    digitalWrite(v5, LOW);  
    digitalWrite(v6, LOW);  
    delay(2000);  
    digitalWrite(v9, LOW);  
}
```

---

```
void loop() {

int stanje_p1 = digitalRead(p1);
int stanje_p2 = digitalRead(p2);

Serial.println(stanje_p1,stanje_p2); //ispisivanje stanja na ulaznim pinovima s prihvatnica

    pozicija_1a();
    delay(2000); //pričekaj 2 sekunde
    spusti();
    delay(1000);
    if (digitalRead(p1) == HIGH){ //ako prihvatnica detektira predmet
        digitalWrite(v10,HIGH);
        delay(1000);
        digitalWrite(v7,HIGH); //okreni na drugu prihvatnicu
    }
    else pozicija_2a();
    pozicija_2a();
    delay(1000);
}
void podigni (){ // diže ruku u gornju poziciju
    digitalWrite(v9,HIGH);
    delay(1000);
}
void spusti() { //spušta ruku u poziciju za uzimanje i ostavljanje predmeta
    delay(2000);
    digitalWrite(v9,LOW);
    delay(1000);
}
```



A

A

B

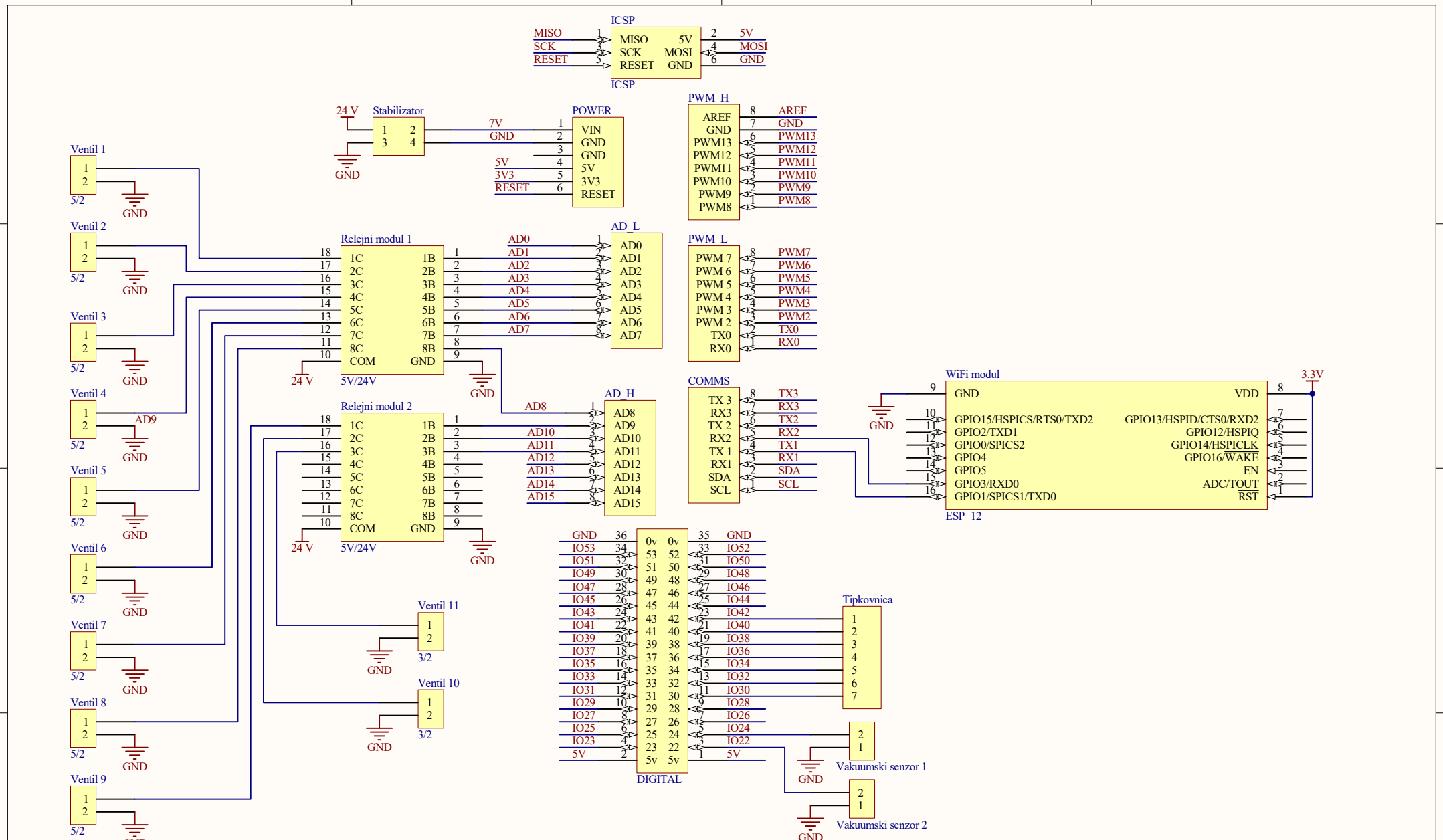
B

C

C

D

D



Naziv	Shema ožičenja manipulatora	Fakultet strojarstva i brodogradnje Sveučilište u Zagrebu Ivana Lučića 5 10 000 Zagreb	
Student	Dario Poznanović		
Datum:	15.9.2019.	Vrijeme:	22:44:10
File:	C:\Users\Korisnik\Desktop\Altium Designer\shema.SchDoc		