

# Navigacija robota pomoću kamere

---

Šviglin, Hrvoje

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Bjelovar University of Applied Sciences / Veleučilište u Bjelovaru**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:144:025940>

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

Download date / Datum preuzimanja: **2025-02-20**



Repository / Repozitorij:

[Digital Repository of Bjelovar University of Applied Sciences](#)



VELEUČILIŠTE U BJELOVARU  
PREDDIPLOMSKI STRUČNI STUDIJ MEHATRONIKA

# **NAVIGACIJA ROBOTA POMOĆU KAMERE**

Završni rad br. 03/MEH/2022

Hrvoje Šviglin

Bjelovar, lipanj 2022.



Veleučilište u Bjelovaru  
Trg E. Kvaternika 4, Bjelovar

## 1. DEFINIRANJE TEME ZAVRŠNOG RADA I POVJERENSTVA

Student: **Šviglin Hrvoje**

JMBAG: **0023093229**

Naslov rada (tema): **Navigacija robota pomoću kamere**

Područje: **Tehničke znanosti**

Polje: **Strojarstvo**

Grana: **Proizvodno strojarstvo**

Mentor: **dr.sc. Tomislav Pavlic**

zvanje: **viši predavač**

**Članovi Povjerenstva za ocjenjivanje i obranu završnog rada:**

- 1. dr.sc. Zoran Vrhovski, predsjednik**
- 2. dr.sc. Tomislav Pavlic, mentor**
- 3. Danijel Radočaj, mag.inž.meh., član**

## 2. ZADATAK ZAVRŠNOG RADA BROJ: 03/MEH/2022

U sklopu završnog rada potrebno je:

1. Prikazati razvoj industrijskih robota i vizijskih sustava kroz povijest.
2. Opisati korištenu robotsku ćeliju.
3. Opisati proces učenja vizijskog sustava.
4. Prikazati i opisati simulaciju rada robotske ćelije.
5. Opisati rad realne robotske ćelije.
6. Analizirati razlike između simulacije i realnog sustava.
7. Prikazati rad kompletne robotske ćelije.

Datum: 30.03.2022. godine

Mentor: **dr.sc. Tomislav Pavlic**



## Sadržaj

<b>1. Uvod.....</b>	<b>1</b>
1.1 Sustav rada robota.....	1
<b>2. ROBOT.....</b>	<b>2</b>
2.1 Epson SCARA LS6-B702S (standard).....	3
2.2 Epson RC90.....	8
2.3 Epson CV2.....	12
2.3.1 Instalacija CV2 kontrolera i kamere.....	15
<b>3. KAMERA.....</b>	<b>21</b>
3.1 Basler acA1600-60gm.....	21
3.2 Objektiv kamere.....	23
3.2.1 Odabir objektiva kamere.....	24
3.3 Implementacija kamere.....	28
3.4 Postavke kamere.....	28
3.5 Kalibracija kamere.....	32
<b>4. RAD ROBOTA S KAMEROM.....</b>	<b>36</b>
4.1. Prikaz simulacije rada robotskog manipulatora.....	36
4.1.1 Programiranje robota.....	41
4.1.2 Učenje točaka.....	44
4.1.3 Funkcija Main.....	48
4.1.4 Funkcija InitRobot.....	50
4.1.5 Pokretanje programa.....	51
4.2 Prikaz stvarnog sustava robotskog manipulatora.....	53
4.2.1 Funkcije za vakuum.....	56
4.2.2 Funkcija za kontrolu slike.....	58
4.2.3 Funkcija za pretragu slike s kamere.....	60
4.2.4 Funkcija za pražnjenje palete.....	62
4.2.5 Funkcija za izmjenu palete.....	63
4.2.6 Funkcija za dozvolu okretnici.....	64
4.2.7 Funkcije za kretanje robota.....	65
4.2.8 Funkcija za automatski rad.....	66
4.2.9 Funkcija za osnovni položaj.....	67
4.2.10 Main funkcija.....	69
4.3 Analiza rada robotskog manipulatora.....	71
<b>5. ZAKLJUČAK.....</b>	<b>72</b>
<b>6. LITERATURA.....</b>	<b>73</b>
<b>7. OZNAKE I KRATICE.....</b>	<b>75</b>
<b>8. SAŽETAK.....</b>	<b>76</b>
<b>9. ABSTRACT.....</b>	<b>77</b>

## **1. Uvod**

Danas je nezamislivo realizirati bilo kakav moderan industrijski proces bez asistencije industrijskih robota. Roboti su precizniji, brži, učinkovitiji i jeftiniji od radnika u sve više radnih aktivnosti. Dokazano je da roboti poslodavcu predstavljaju manji trošak od radnika, najčešće obavljaju rutinske radnje točnije od čovjeka, poboljšavaju kvalitetu proizvodnje te minimiziraju potrošnju različitih resursa i ne stvaraju otpad. Danas je robotika ključni pokretač konkurentnosti u velikim proizvodnim industrijama. Robotika postaje okidač koji će potaknuti razvoj novih generacija autonomnih i kognitivnih uređaja, koji će svojim sposobnostima učenja neprimjetno komunicirati sa svijetom oko sebe i time omogućiti vezu koja danas nedostaje između digitalnog i fizičkog svijeta.

U ovom završnom radu biti će prikazano kako se povezuje industrijski robot s kamerom, kako bi omogućio spomenutu vezu između digitalnog i fizičkog svijeta.

### **1.1 Sustav rada robota**

U postojećem sustavu robotska ruka uzima proizvod iz kartonske kutije te ga stavlja na proizvodnu traku. Budući da je kutija s proizvodom kartonska, on nije uvijek na istoj poziciji uzimanja. Proizvod može biti pomaknut ili zakrenut svega nekoliko milimetara te već tada robotski manipulator ima problema s obavljanjem radnje umetanja proizvoda na proizvodnu traku. Ako robotski manipulator uzme proizvod u krivoj poziciji, može se dogoditi da prilikom transporta iz kutije prema proizvodnoj traci, isti ispadne te se polomi. Također se može dogoditi da je proizvod previše zakrenut ili pomaknut te se prilikom transporta slomi proizvod ili sama robotska ruka u putanji. Implementacijom industrijske kamere riješio bi se problem loše pozicioniranih proizvoda u kartonskoj kutiji. Kamera bi u stvarnom vremenu određivala položaj proizvoda te bi se pomoću toga robot navodio u poziciju koja je pogodna za nesmetan rad.

## 2. ROBOT

Prvi eksperimentalni primjerci robota napravljeni su u SAD-u 1940-ih na američkim sveučilištima. Suvremeni roboti nastali su 1950-ih u SAD-u, a razvoj je potaknut razvojem prvih automata te kasnijim razvojem alatnih strojeva, CNC. Istodobno su se razvijali i manipulatori koji su programiranjem postali djelomično samostalni u radu, a imali su odlike robota. Godine 1954. američki izumitelj George Devol programirao je prvu takvu konstrukciju (uređaj za programirano premještanje predmeta). S Josephom Engelbergerom započeo je komercijalnu proizvodnju robota početkom 1960-ih. Njihova tvrtka *Unimation* bila je prva tvornica za proizvodnju robota. Godine 1961. Engelberger prodaje prvi industrijski robot nazvan *Unimate* i zbog toga je prozvan ocem robotike. Iako njegovi roboti nisu pobudili veliko zanimanje u SAD-u, u Japanu su naišli na dobar prijem, gdje je 1970. godine u funkciju krenuo prvi industrijski robot koji je radio kao zavarivač u Nissanovoj tvornici. Tijekom sedamdesetih i osamdesetih godina robotika je izrazito napredovala zahvaljujući velikim investicijama automobilske industrije, no potkraj osamdesetih godina industrija je počela propadati. Zbog slabe isplativosti većina američkih proizvođača robota je propala. Jedina oaza robotike ostao je Japan, čije tvornice danas „zapošljavaju” više od polovine svjetskih robota. Od 2000. godine ponovno se bilježi porast u proizvodnji i razvoju robota. Roboti se počinju koristiti svugdje: u industriji, medicini, podvodnim i svemirskim istraživanjima, transportu, servisu, edukaciji itd.

Najčešće se pod pojmom robota podrazumijeva industrijski robot, kojeg još nazivamo robotski manipulator ili jednostavnije, i svima jasnije, robotska ruka. Robotske ruke najviše se koriste u proizvodnim procesima gdje je potrebno istu radnju obavljati neprekidno, jednakom brzinom te učinkovitošću. To može biti premještanje, smještanje ili nadograđivanje u hodu.

Robotska ruka sastoji se od:

- mehaničke strukture
- aktuatora koji služe za pogon
- senzora koji detektiraju status strukture, odnosno poziciju robotske ruke
- sistema upravljanja koji je najčešće računalo.

U ovom završnom radu korišten je robot proizvođača *EPSON*, model *Epson SCARA LS6-B702S* (standard). Navedeni je model *SCARA (Selective Compliant Articulated Robot for Assembly)* strukture, te ima tri stupnja rotacije i jedan stupanj translacije.

## 2.1 *Epson SCARA LS6-B702S (standard)*

Struktura robota je *SCARA* što je skraćena od *Selective Compliant Articulated Robot for Assembly*. Navedeni robot ima četiri stupnja slobode gibanja, tri rotacije i jedan translacije. Roboti serije *LS-B SCARA* razvijeni su za automobilsku i medicinsku industriju, laboratorijsku automatizaciju, elektroničke dijelove, potrošačke elektroničke uređaje i industrijska tržišta. Serija nastavlja primjenjivati metodiku prilagodljivosti Epsonovih robota *SCARA*, koja se temelji na vrhunskoj liniji robotičkih proizvoda srednje i početne klase, bez suvišnih funkcija.

Ovaj Epsonov robot osmišljen je za primjene koje zahtijevaju iznimno brzu izvedbu pri niskom trošku. Jedinica motora bez baterije maksimalno smanjuje vrijeme zastoja i smanjuje ukupan trošak vlasništva. Epsonova tehnologija bez suvišnih funkcija i niska potrošnja struje robota imat će pozitivne učinke na poslovanje.

Osmišljen kako bi pružio savršen omjer ravnoteže i izvedbe, robot *SCARA LS6-B702S* nudi: brže radne cikluse, niže postavljen kabelski kanal, ugrađen priključak za kamere za jednostavno postavljanje vizualnog sustava, novi izgled prilagođen gornjem dijelu ruke za poboljšanu upotrebu i jedinicu motora bez baterije za smanjenje zastoja i ukupnog troška vlasništva. Korisna nosivost ovog robota *SCARA LS-B* iznosi 6 kg, s rukom duljine 700 mm.



Slika 2.1: *Epson SCARA LS6-B702S*

Na slici 2.1 prikazan je model robota *LS6-B702S*, koji je ugrađen i radi. Zadatak robota u ovom je sustavu premještanje kućišta iz kartonske kutije na proizvodnu traku. Kućište koje robot premješta napravljeno je od keramike te je izrazito krhko. To kućište se koristi za izradu štapnog regulatora.

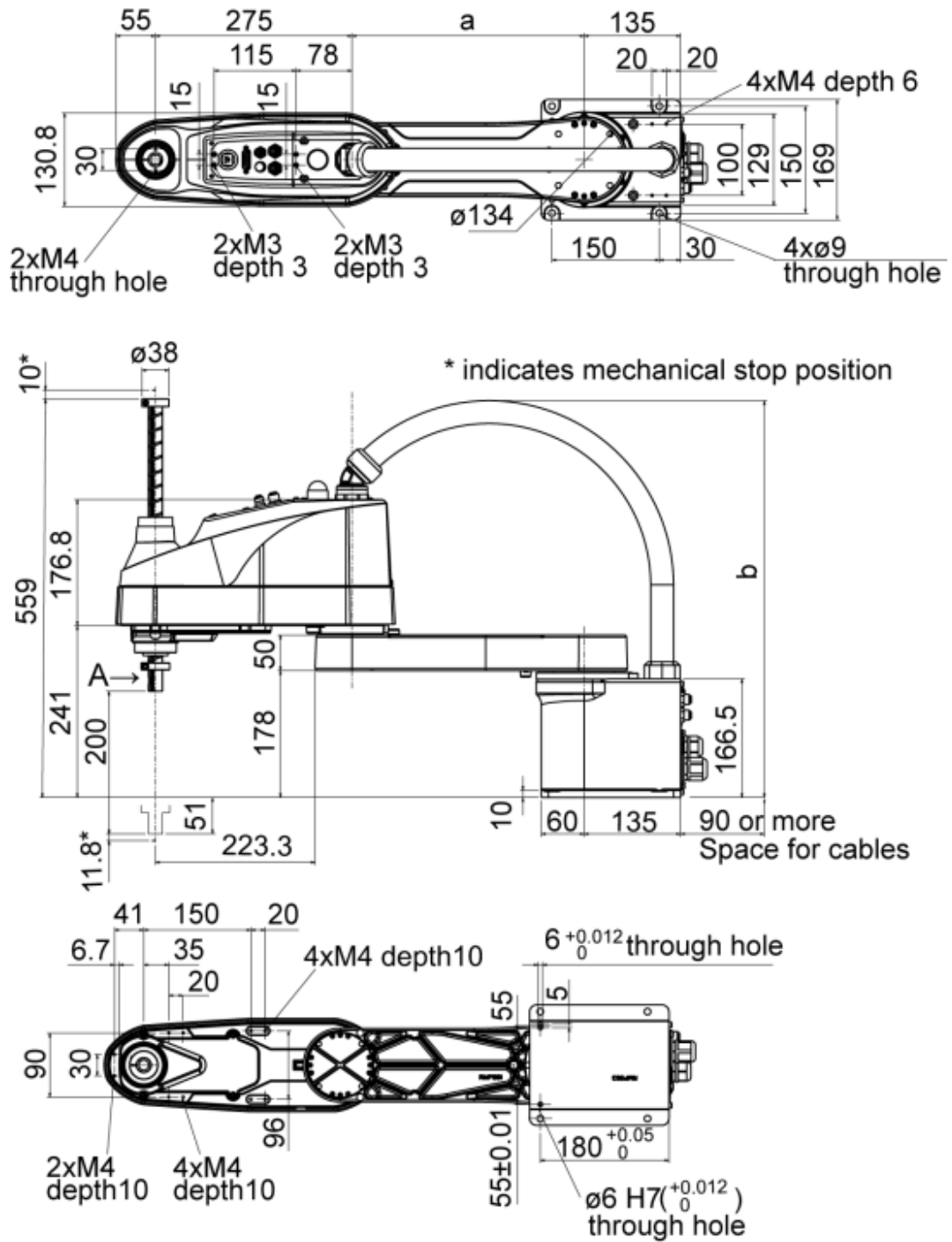
Karakteristike ovog robota jesu:

- konstrukcija: *SCARA* (robot s četiri osi)
- nosiva komponenta: nominalna 2 kg, maksimalna 6 kg
- vodoravni doseg: 700 mm
- uspravni domet: J3: 200 mm
- domet, usmjerenost: J4: +/- 360 °
- točnost ponavljanja, vodoravno: +/- 0,02 mm
- točnost ponavljanja, uspravno: J3: +/- 0,02 mm
- točnost ponavljanja, usmjerenost: J4: +/- 0,01 °



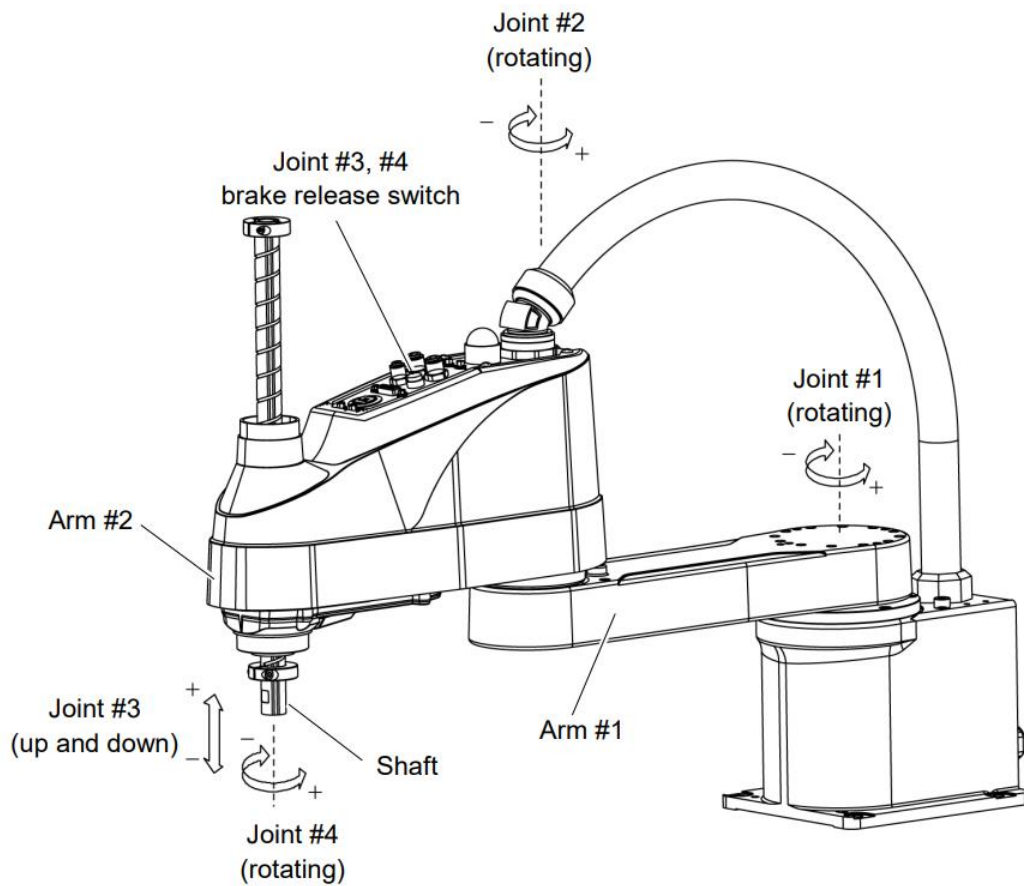
- maks. raspon operacija: J1: +/- 132 °, J2: +/- 150 °, J3: 200 mm, J4: +/- 360 °
- maks. brzina rada: J1,J2: +/- 8.590 mm/s, J3: +/- 1.100 mm/s, J4: +/- 2.000 °/s
- dozvoljeni moment tromosti: nom. 0,01 kg\*m<sup>2</sup>, maks. 0,12 kg\*m<sup>2</sup>
- električni kabeli za korisnike: 1x D-Sub 15 igla, *Equivalent to* 8 pin (RJ45) Cat.5e
- pneumatski kabeli za korisnike: 3 x (1 x 4 mm Ø & 2 x 6 mm Ø)
- os z: Ø, izvana: 20 mm / unutrašnji: 14 mm
- snaga pritiskanja: 100 N
- masa proizvoda: 18 kg
- upravljačka jedinica: RC90-B
- mogućnost montiranja: 3 m, 5 m, 10 m
- certifikati: Direktiva o elektromagnetskoj kompatibilnosti – EMC: EN 55011, ANSI/RIA: R15.06-1999, CE Mark, ANSI/RIA R15.06-2012, NFPA 79 (2007 *Edition*), KC Mark, KCs Mark.

## Standard Model (LS6-B\*02S)



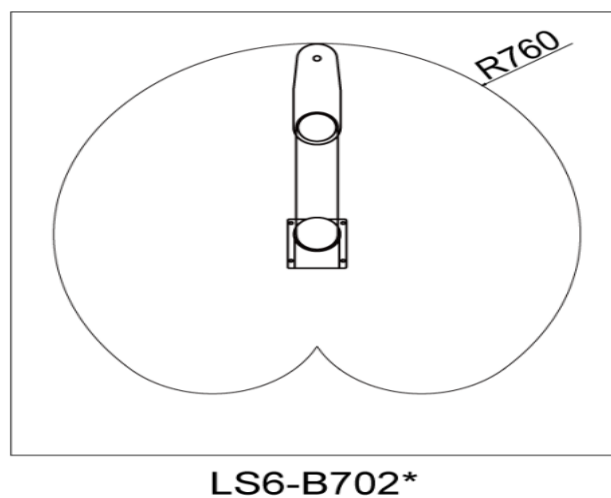
Slika 2.2: Epson SCARA LS6-B702S, prikaz dimenzija [2]

Na slici 2.2 moguće je vidjeti dimenzije robota, kao što su širina, dužina te visina.



Slika 2.2: *Epson SCARA LS6-B702S*, prikaz osi [2]

Na slici 2.2 prikaz je osi robota. Vidljivo je da su osi 1, 2, i 4 rotacijske, dok je os 3 translacijska.



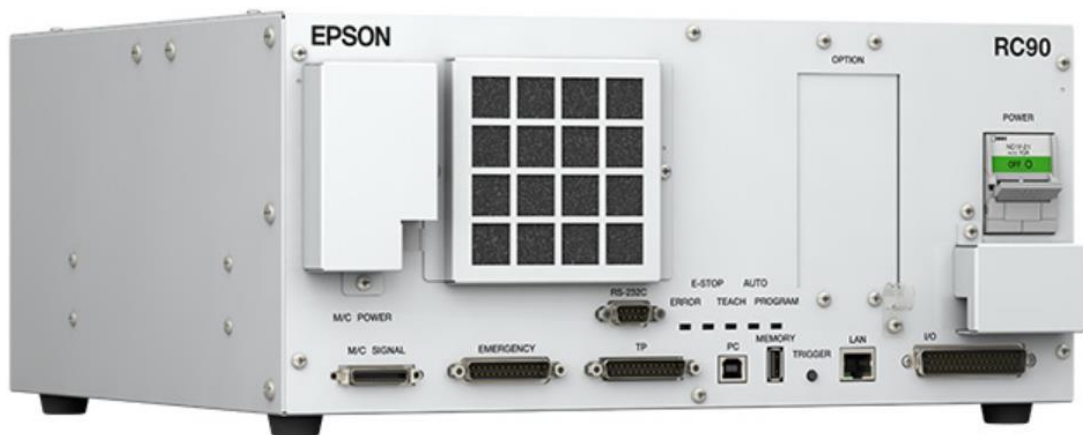
Slika 2.3: *Epson SCARA LS6-B702S*, područje rada [2]

Na slici 2.3 prikazano je područje rada *Epson SCARA LS6-B702S*, koje je radijusa do 760 mm u prikazanom smjeru.

## 2.2 *Epson RC90*

Robotom se upravlja pomoću *Epson RC90* kontrolera. *Epson RC90* kontroler koristi se za robote serije *Epson SCARA LS*. *Epson RC90* pruža vrhunsko iskustvo u jednostavnosti korištenja, po jako pristupačnoj cijeni. *RC90* može kontrolirati robote *Epson LS3* i *LS6 SCARA* i pruža superiornu *Power Drive* servokontrolu za super glatko kretanje, brzo ubrzanje ili usporavanje i brzo vrijeme ciklusa.

Kontroler dolazi sa softverom *Epson RC+ 7.0*, koji nam služi za upravljanje sustavom.



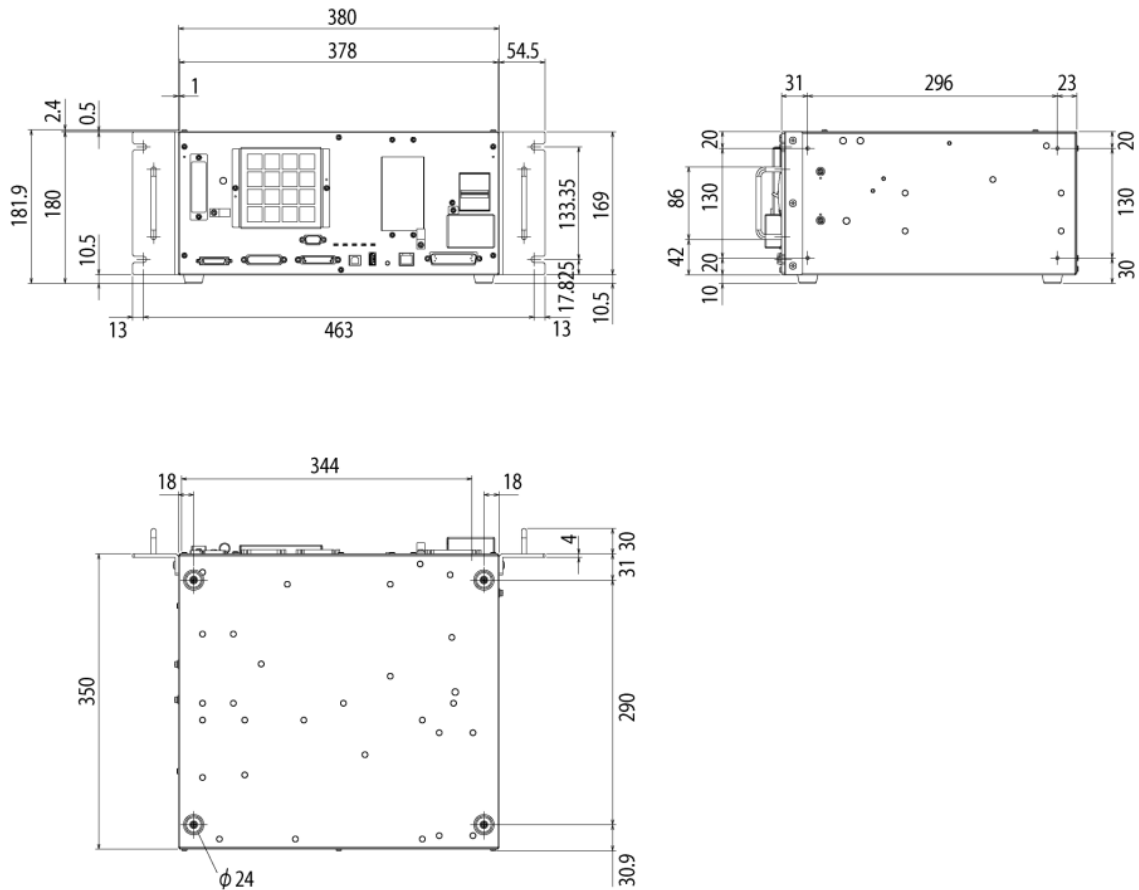
Slika 2.4: *Epson RC90*

Na slici 2.4 je prikaz *Epson RC90* kontrolera.

Karakteristike navedenog kontrolera su:

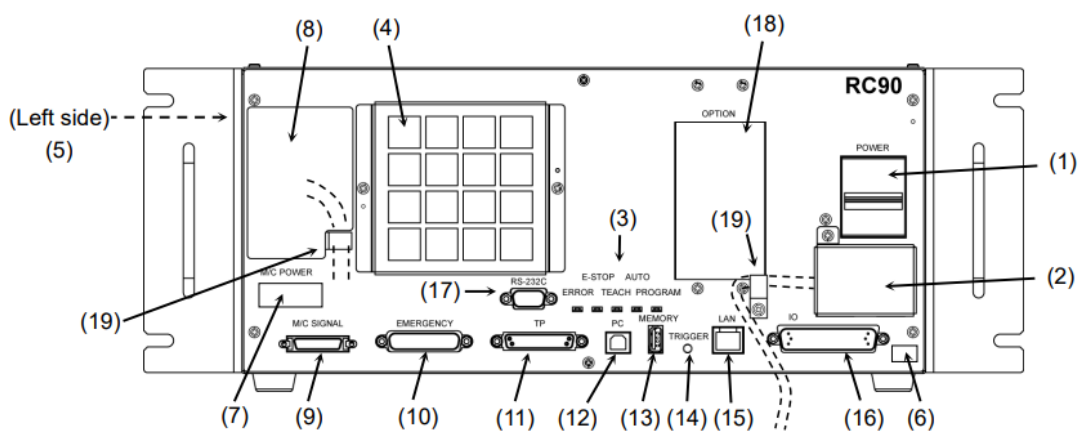
- CPU: 32 bita mikroprocesor
- broj osi kojima može upravljati: 4 (s AC servomotorima)
- kontrola robotskog manipulatora:
  - softver: *Epson RC+ 7.0*
  - kontrola osi: upravljanje s 4 osi istovremeno

- kontrola brzine:
  - od točke do točke: programabilno od 1 do 100 %
  - kontinuirano kretanje: programabilno
- kontrola pozicioniranja: od točke do točke (PtP), kontinuirano kretanje (CP)
- kapacitet memorije: maksimalna veličina objekta 4 MB
- metoda učenja točki: daljinska, direktna, ručni upis podataka
- unutrašnji ulazi/izlazi: input 24, output 16
- komunikacijsko sučelje: ethernet ( 1 kanal )
- RS-232C port: 1 port
- sigurnosne značajke:
  - prekidač za zaustavljanje u nuždi
  - ulaz za sigurnosna vrata
  - način rada male snage
  - dinamička kočnica
  - detekcija preopterećenja motora
  - otkrivanje nepravilnog momenta motora (manipulator izvan kontrole)
  - detekcija greške u brzini motora
  - prelijevanje pozicioniranja - greška servodetekcija
  - prekoračenje brzine - greška servodetekcija
  - detekcija CPU nepravilnost
  - otkrivanje pogreške kontrolnog zbroja memorije
  - detekcija pregrijavanja na modulu pogonskog motora
  - detekcija smanjenja napona napajanja izmjeničnom strujom
  - detekcija greške u temperaturi
  - detekcija greške ventilatora
- napajanje: AC 200 V to AC 240 V, *one-phase* 50/60 Hz
- maksimalna potrošnja energije: 2,5 kVA (ovisno o modelu manipulatora)
- težina: 7.5 kg.



Slika 2.5: *Epson RC90*, dimenzije [3]

Na slici 2.5 prikazane su dimenzije koje su ugrubo u rasponu 500 x 350 x 180. Uzevši u obzir dimenzije te da je težina *Epson RC90* kontrolera 7.5 kg, zaista se i može zaključiti da je ovaj kontroler kompaktan.



Slika 2.6: *Epson RC90*, opis [3]

Tablica 2.1: Oznake dijelova RC90

Redni broj	Naziv	Opis funkcije
1.	Tipka za napajanje	Uključuje ili isključuje kontroler.
2.	AC ulaz	Priključni blok za 200VAC ulaz.
3.	LED prikazuje trenutni način rada	ERROR, E-STOP, TEACH, AUTO, ili PROGRAM način rada.
4.	Filter ventilatora	Ispred ventilatora postavljen je zaštitni filter za filtriranje prašine.
5.	Oznaka s potpisom	Prikazani su serijski broj kontrolera i drugi podaci.
6.	Oznaka kontrolnog broja	Naveden je serijski broj kontrolera.
7.	Naljepnica za provjeru veze	Sadrži pojediniosti o manipulatoru.
8.	<i>M/C POWER</i> priključak	Priključak za napajanje manipulatora.
9.	<i>M/C SIGNAL</i> priključak	Priključak za signale.
10.	<i>EMERGENCY</i> priključak	Priključak za nužni isklop.
11.	<i>TP port</i>	Služi za učenje kretnji robota.
12.	<i>Development PC Connection Port</i>	Za spajanje RC90 s računalom.
13.	Memorijski port	<i>USB</i> port.
14.	<i>Trigger Switch</i>	Okidanje.
15.	<i>LAN</i> port	
16.	<i>I/O</i> priključak	
17.	RS-232C port	
18.	Opcijski utor	
19.	Stezaljka za kabel	

### 2.3 *Epson CV2*

*Epson CV2* vizualni je kontroler koji se koristi kao veza između robota i kamere. U kombinaciji s Epsonovim robotima i vizualnim kamerama, kontroler pruža jednostavno rješenje za vizualizaciju. Kao integrirani dio Epsonovog razvojnog okruženja *RC+ 7.0* pomaže smanjiti ukupno vrijeme razvoja aplikacije za vizualno usmjeravanje. Druge značajke uključuju: brzu komunikaciju putem sustava *Giga Ethernet*, podršku za do četiri kamere *GigE* i dvije *USB* kamere (uz visoku rezoluciju), jednostavno i intuitivno sučelje za jednostavnu registraciju vizualnih objekata povlačenjem i ispuštanjem. Testiranje procesa i sekvenciranje mogu se pripremiti prije ugradnje.



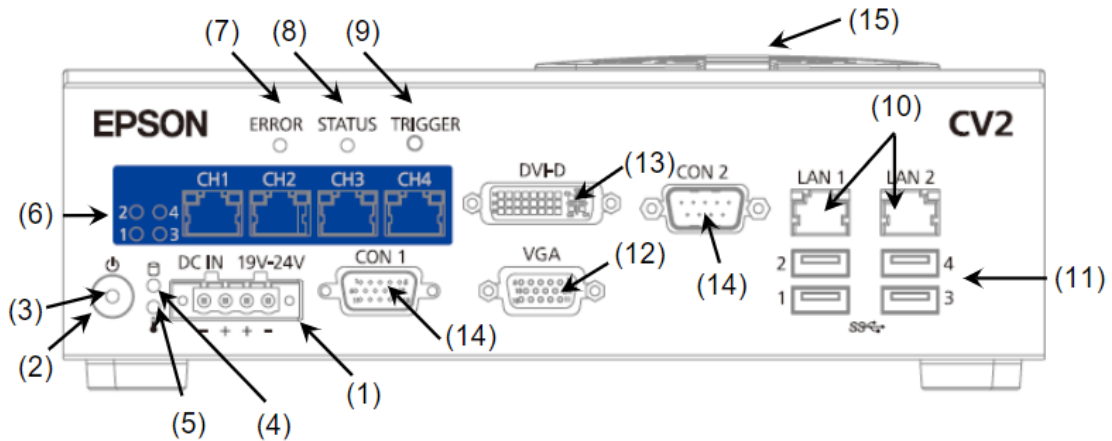
Slika 2.7: *Epson CV2* vizualni kontroler

Na slici 2.7 prikazan je vizualni kontroler CV2. Na slici se također vide primjeri industrijskih kamera koje se mogu koristiti sa kontrolerom.

Karakteristike *Epson CV2* kontrolera jesu:

- broj kamera koje se mogu priključiti: 4 *GigaE* kamere, 2 *USB* kamere
- napajanje: 24 V DC
- nazivna struja: 11.57A (pri naponu 19 V DC) do 9.16 A (pri naponu 24 V DC)
- težina: 2.1 kg
- sigurnosni standard.



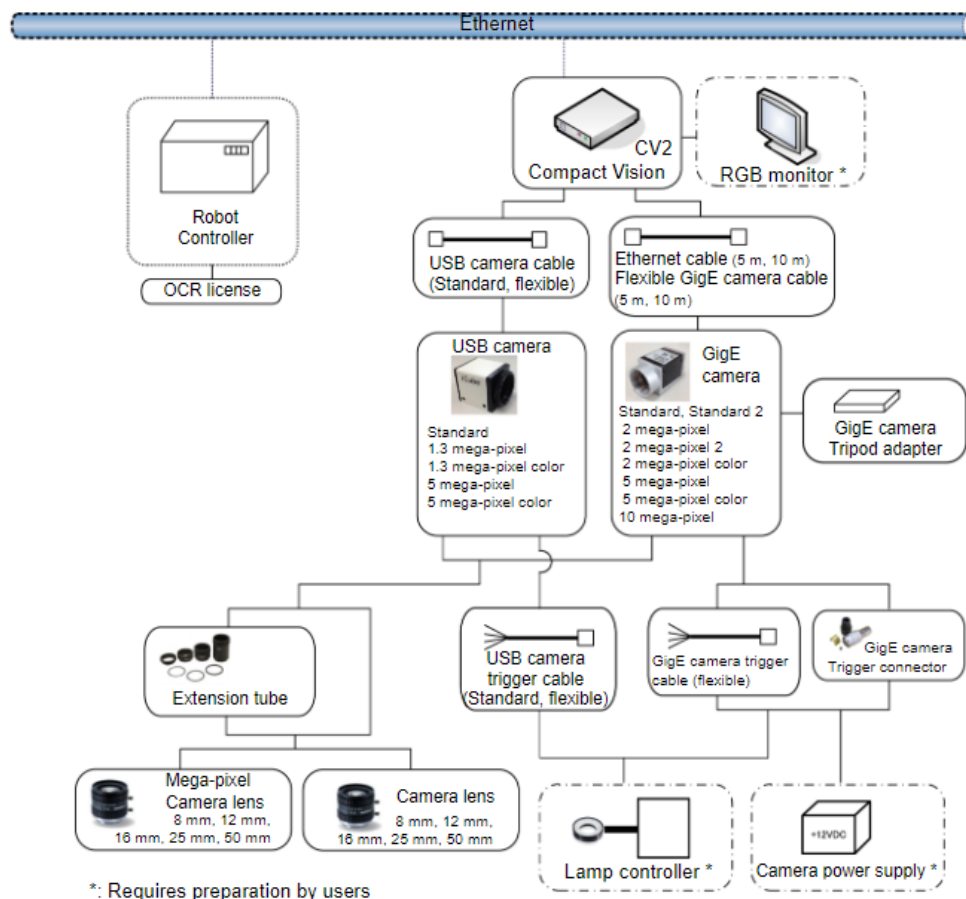


Slika 2.8: Epson CV2, opis uređaja [4]

Tablica 2.1: Oznake dijelova CV2

Redni broj	Naziv	Opis funkcije
1.	24 V ulazni konektor	Konektor koji služi za napajanje 24 V.
2.	Tipka za napajanje	CV2 se automatski pali kada dovedemo napajanje 24 V. Ako se dogodi da je baterija napajanja slaba CV2 je potrebno upaliti na tipku.
3.	Signalno svjetlo za napajanje	
4.	<i>Cfast Acces</i> signalno svjetlo	
5.	Signalno svjetlo za povišenu temperaturu	
6.	<i>PoE</i> signalno svjetlo (1 do 4)	Signalna svjetla za ethernet kablove.
7.	<i>PoE</i> konektori	
8.	Signalno svjetlo za grešku kontrolera	
9.	Signalno svjetlo za status kontrolera	

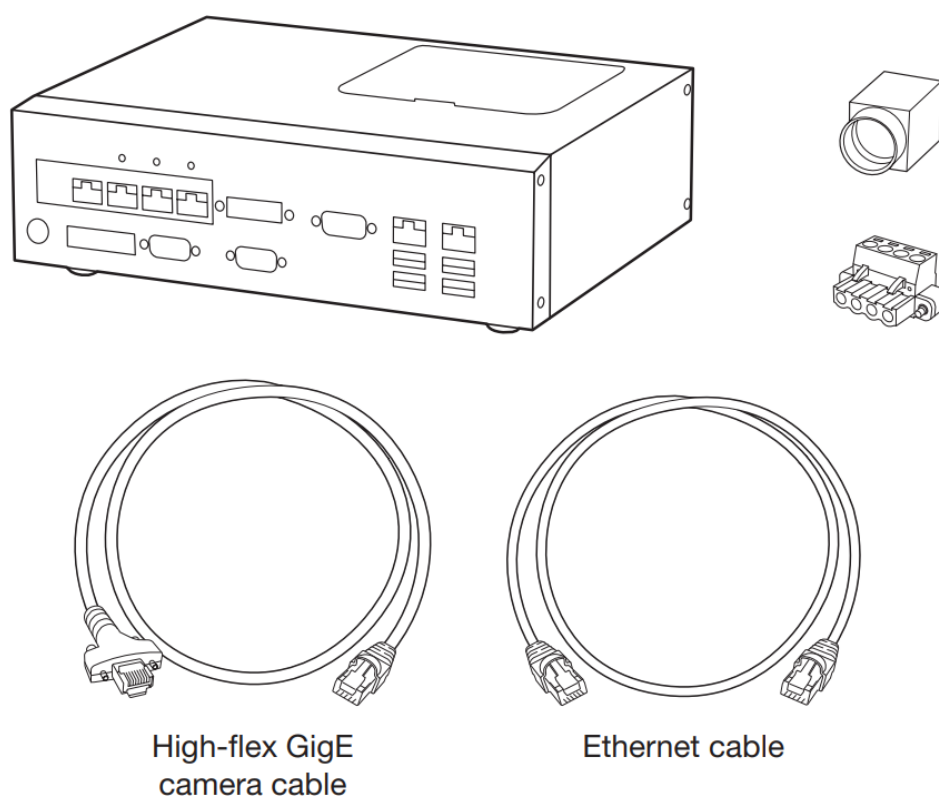
10.	Signalno svjetlo za signal okidanja kontrolera	
11.	LAN konektori	Konektori za switch ili hub te za robo kontrolere.
12.	USB konektori	Konektori za kamere, tipkovnicu te miš. Mogu biti spojene najviše dvije kamere.
13.	VGA konektor	Konektor za monitor.
14.	DVI-D konektor	Konektor za monitor.
15.	CON1 CON2	Funkcije nisu dostupne.
16.	Filter za zrak	



Slika 2.9: *Epson CV2*, dijagram konfiguracije

Na slici 2.9 prikazan je dijagram spajanja vizualnog kontrolera *Epson CV2*. Vizualni kontroler spojen je ethernet kablom s kontrolerom za robot, koji je u ovom slučaju *Epson RC90*. Na taj način ta dva kontrolera komuniciraju u realnom vremenu. Na kontroler *CV2* spajaju se kamere ethernet kablom za komunikaciju. Iako ima jako puno funkcija, kontroler *Epson CV2* ne služi za napajanje kamera.

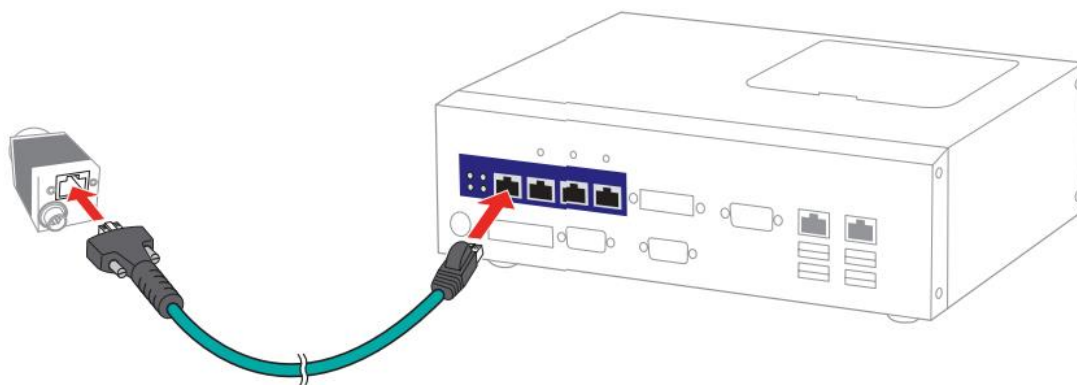
### 2.3.1 Instalacija *CV2* kontrolera i kamere



Slika 2.10: *Epson CV2*

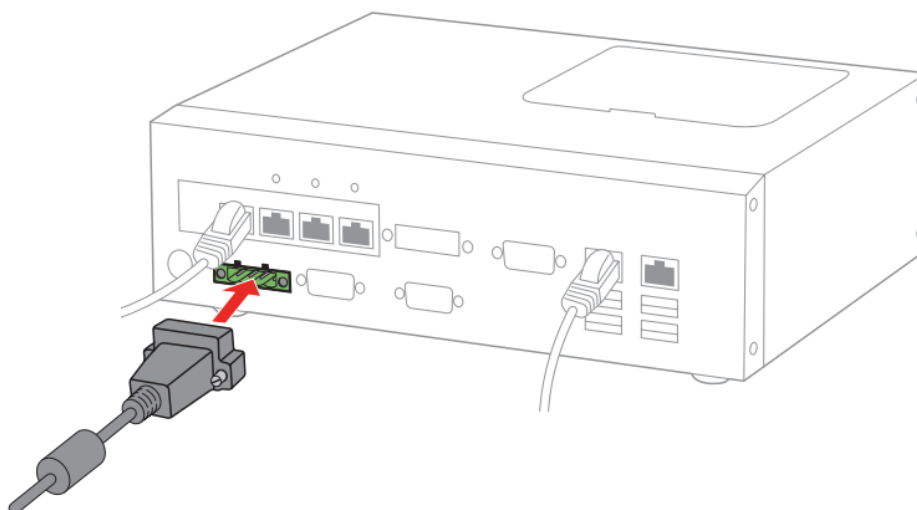
Na slici 2.10 prikazani su dijelovi nakon raspakiranja kontrolera.

Nakon što je sve raspakirano, spaja se *CV2* kontroler s kamerom posebnim ethernet kabelom *High Flex GigE*.



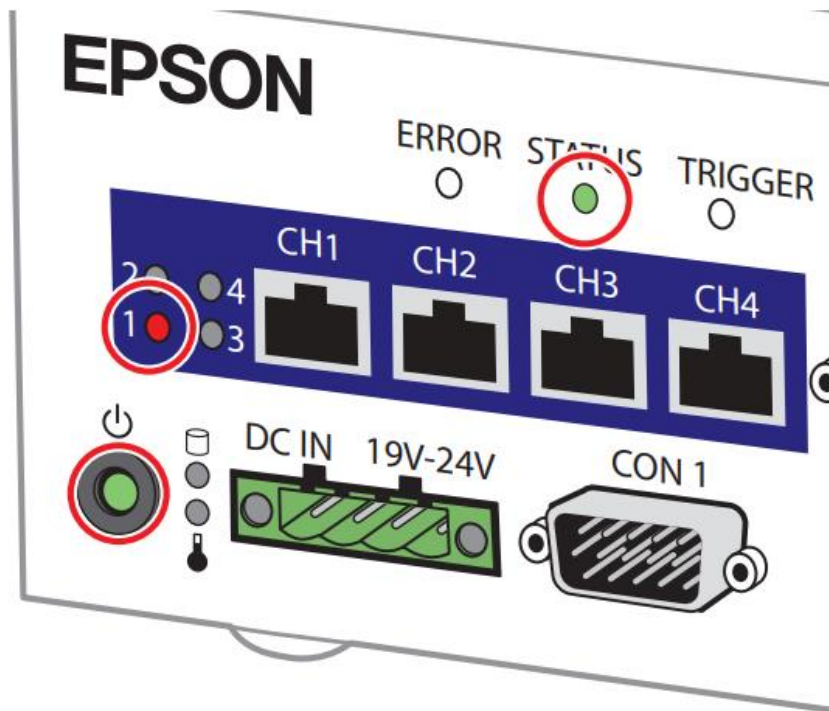
Slika 2.11: *Epson CV2*, spajanje kamere

Nakon što je spojeno napajanje kontrolera, te ako je napajanje upaljeno i kontroler bi se trebao odmah upaliti. Ako se kontroler ne upali potrebno je pritisnuti tipku za napajanje (*ON*).



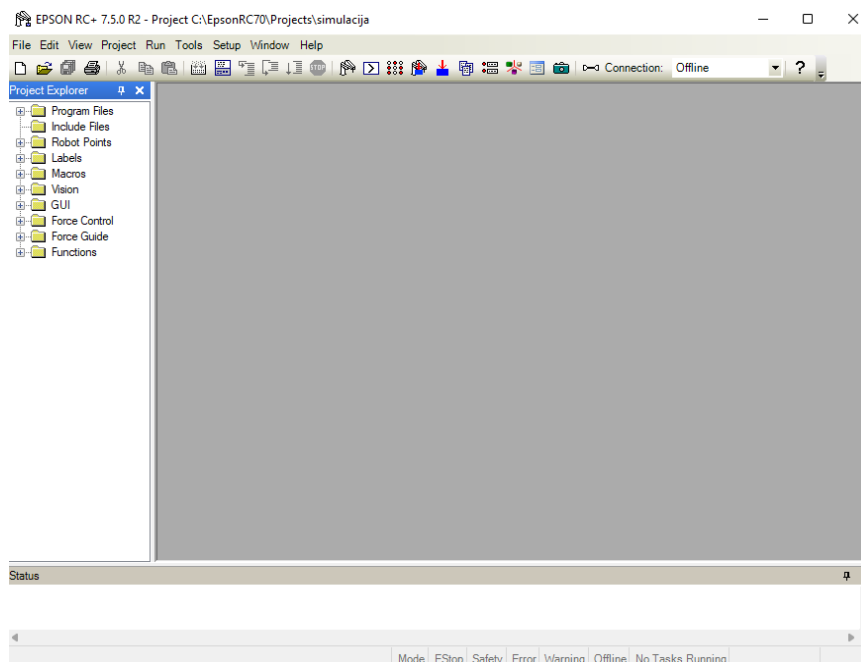
Slika 2.12: *Epson CV2*, spajanje napajanja

Kad je sve spojeno i upaljeno, trebala bi stalno svijetliti signalna svijetla od napajanja, *PoE (Power of Ethernet)* te bi status signalno svjetlo trebalo treperiti.



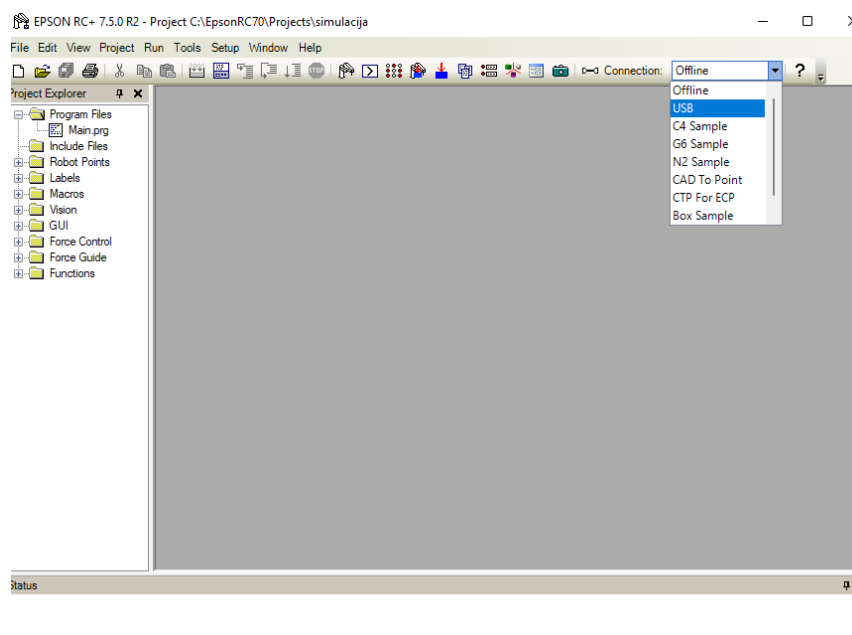
Slika 2.13: *Epson CV2*, prikaz signalnih svjetala

Nakon spajanja hardwarea, prelazi se na programski dio. Prije svega, potrebno je instalirati program *Epson RC+ 7.0* koji je dostupan na web-stranici proizvođača *Epson*. Zatim slijedi pokretanje programa, nakon uspješne instalacije.



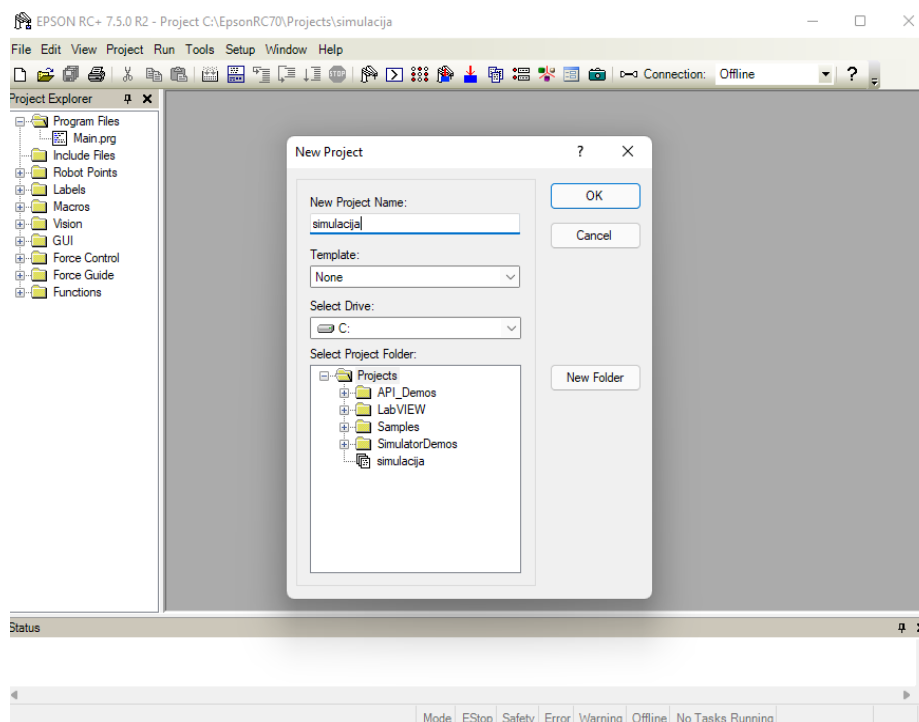
Slika 2.13: *Epson RC+*, prikaz početne stranice programa

Slika 2.13 prikazuje početnu stranicu programa *Epson RC+*.



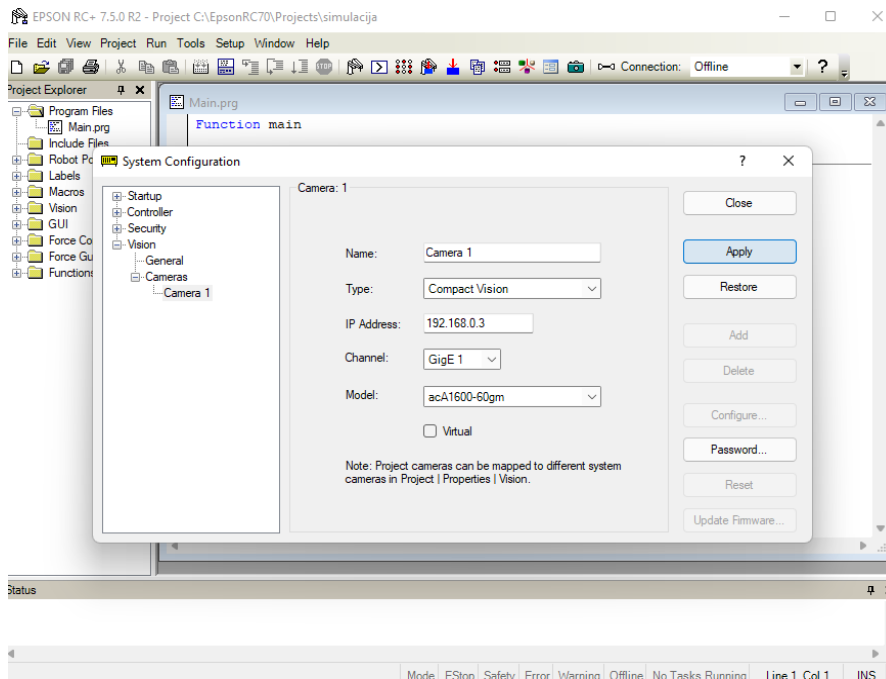
Slika 2.14: *Epson RC+*,

U izborniku se odabire opcija *Connection* i u padajućem izborniku opcija *USB*.



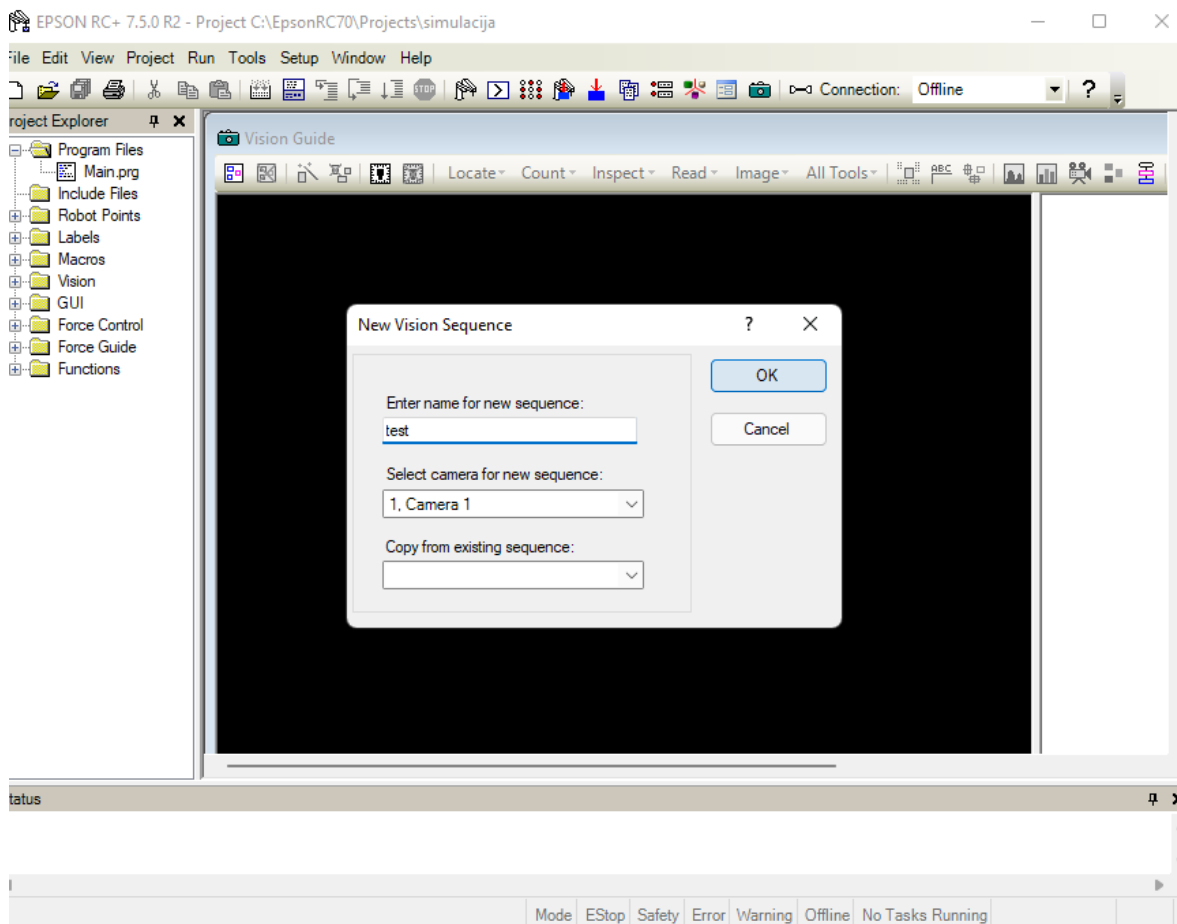
Slika 2.15: *Epson RC+*

U gornjoj alatnoj traci odabire se opcija *Project* i *New Project*. Unosi se ime projekta i pritišće *OK*.



Slika 2.16: *Epson RC+*

Nakon što je zadana naredba za novi projekt, odabire se opcija *Setup* te se odabire *System Configuration*. Na otvorenom prozoru proširuje se *Vision* te se pritišće tipkom miša na *Camera*. Nakon toga odabire se opcija *Add* te se odabire *Manually Configure a Camera*. Sada se mogu unijeti parametri kao na slici 2.16.



Slika 2.16: Epson RC+

Kada su uneseni svi parametri za kameru, može se pritisnuti ikonica kamere. Ona se nalazi gore desno, odmah do opcije *Connection*. Nakon što je pritisnuta ikonica kamere otvara se novi prozor unutar programa. U novom prozoru odabire se prva ikonica za konfiguriranje sekvence. Proizvoljno se unosi ime te se odabire kamera koja je već ranije konfigurirana s imenom *Camera 1*. Nakon toga bi trebala biti prikazana slika s kamere u realnom vremenu.



### 3. KAMERA

Osamdesetih godina prošlog stoljeća u svijetu istraživanja strojni vid počinje rasti, s novim teorijama i konceptima koji se pojavljuju. Jedan od njih je sustav optičkog prepoznavanja znakova (OCR). U početku su se koristili u raznim industrijskim aplikacijama za čitanje i provjeru slova, simbola i brojeva. Pametne kamere razvijene su kasnih 80-ih, što je dovelo do šire upotrebe i više aplikacija. Već 1990. strojni vid postaje sve češći u proizvodnim okruženjima što dovodi do stvaranja industrije strojnog vida, preko 100 tvrtki počinje prodavati sustave strojnog vida. Razvijena su LED svjetla za industriju strojnog vida, a postignut je napredak u funkciji senzora i upravljačkoj arhitekturi, čime se poboljšavaju sposobnosti sustava strojnog vida. Troškovi sustava strojnog vida počinju padati.

Industrijske kamere posebna su vrsta kamera koje su prilagođene za rad u teškim uvjetima kao što su visoke ili niske temperature, rad u područjima vibracija ili rad pod tlakom. Kamere za industrijsku uporabu značajno se razlikuju od običnih svakodnevnih kamera po visokoj ocjeni pouzdanosti. Takvi uređaji često imaju čvrsto kućište koje štiti unutarnje mehanizme od vlage, prašine, štetnih plinova i drugih čimbenika okoliša. Proizvođači industrijskih kamera za svoju proizvodnju koriste najtrajnije materijale (metali, ponekad ojačana plastika). Zbog svog ojačanog dizajna, takva oprema rješava probleme u nestandardnim ili teškim uvjetima. Industrijske kamere kao dio sustava računalnog vida danas služe za totalnu automatizaciju proizvodnje. Nalazimo ih u raznim područjima, posebno u praćenju proizvodnje i nizu kompliciranih mjernih zadataka. Kontrola kvalitete je još jedno polje koje se snažno oslanja na industrijsku obradu slike. Digitalne industrijske kamere općenito su robusnije od standardnih digitalnih fotoaparata koji se koriste u svakodnevnoj uporabi.

#### 3.1 *Basler acA1600-60gm*

Industrijska kamera koja se koristi je kamera proizvođača *Basler*, *Basler Ace acA1600-60gm*. *AcA1600-60gm* izgrađena je na *e2v CMOS* senzoru za isporuku 60 sličica u sekundi pri punoj rezoluciji 1600 x 1200. Budući da *Basler Ace GigE* kamera koristi isti otisak 29 x 29 mm koji je već dugi niz godina standardan za analogne kamere, zamjena analognih kamera jednostavna je.



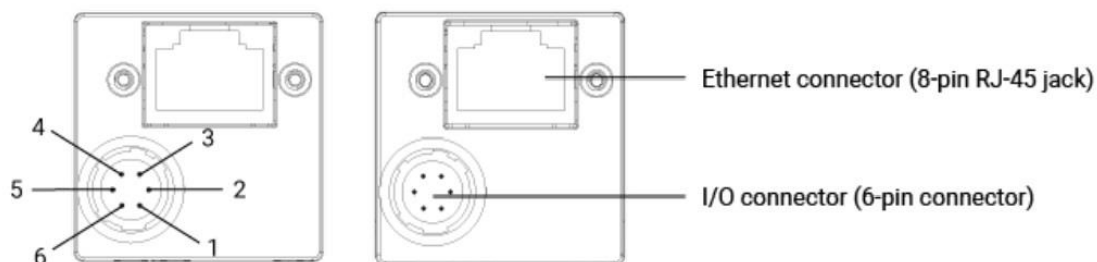
Slika 3.1: Basler Ace acA1600-60gm

Karakteristike *Basler Ace acA1600-60gm* jesu:

- nosač objektiva: C-držać
- rezolucija senzora: 1.600 (h) x 1.200 (v) pixela
- model senzora: e2v EV76C570ABT6
- boja: jednobojna
- vrsta zatvarača: globalno
- brzina kadrova (puna razlučivost): 60,0 sličica u sekundi
- tamni šum: 22.0 e-
- kapacitet zasićenja: 6,8 ke-
- dinamički raspon: 49,8 dB
- omjer signal-šum: 38,3 dB
- metoda ispitivanja: EMVA 1288
- podatkovno sučelje: *Ethernet GigE Vision*
- brzina podataka: gigabitni ethernet
- napajanje: *PoE* ili 12 VDC
- veličina kućišta (D x Š x V): 42 mm x 29 mm x 29 mm
- težina: 90 g.

*GigE* je skraćenica od *Gigabit Ethernet* te je najčešće sučelje za povezivanje kamere za vid u sustavima strojnog vida. *Gigabit Ethernet* kamera daje neobrađenu sliku preko mrežnog kabela na računalo. Zatim, softver za obradu slika na računalu analizira te slike i u našem slučaju određuje putanju robota.

Napajanje za ovu kameru može se dovesti preko etherneteta (*Power over Ethernet*) ili vanjskim putem. U navedenom slučaju napajanje je dovedeno preko ethernet kablova.



Slika 3.2: Basler Ace acA1600-60gm, prikaz priključaka

Na slici 3.2 je prikaz priključaka na kameri. Kamera ima ethernet priključak, te ulazno izlazni priključak.

Tablica 3.1: Oznake pinova I/O priključka na kameri

Broj pina	Funkcija
1.	12V napajanje za kameru
2.	Ulaz preko optokaplera
3.	/
4.	Izlaz preko optokaplera
5.	0 za I/O
6.	0V napajanje za kameru

### 3.2 Objektiv kamere

Objektiv kamere optički je sustav kojim se slika snimanog objekta projicira na fotoosjetljivu podlogu. Sastoji se od jedne ili obično više leća, mehanizma za prilagodbu količine svjetlosti koja se propušta u kameru prilikom eksponiranja, mehanizma za izoštravanje i dr. Kod nekih je kamera objektiv sastavni dio tijela, a kod drugih je samostalan i može se izmjenjivati. Ovisno o izboru i podešavanju objektivu, pri eksponiranju će na fotoosjetljivoj podlozi biti ocrtna slika različita vidnoga kuta, bit će potpuno oštra ili zamučena, pravilna ili iskrivljena. Zbog toga je objektiv osnovni dio fotografskog aparata, kojim se bitno utječe na izgled i kvalitetu slike, a time i znatno proširuju kreativne

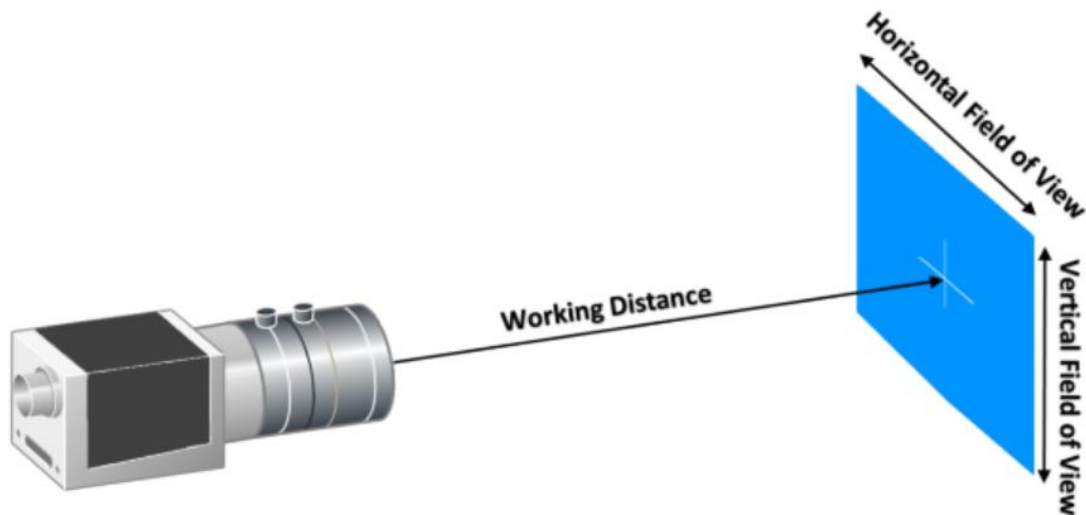
možnosti fotografiranja. Današnji su objektiv najčešće složeni sustavi leća, kombinacijom kojih su u potpunosti ili većim dijelom otklonjene optičke pogreške.



Slika 3.3: *Basler Ace* kamera s objektivom

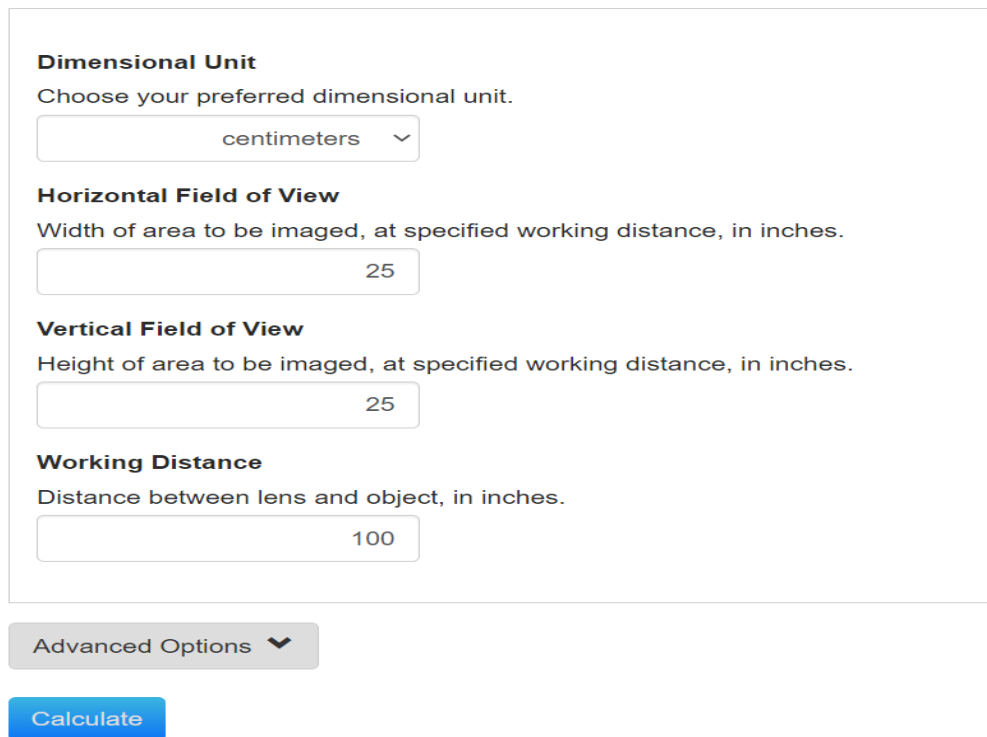
### 3.2.1 Odabir objektiva kamere

Objektiv kamere odabire se pomoću kalkulatora za izračunavanje leća, koji se nalazi na web-stranici proizvođača.



Slika 3.4: Kamera i vidno polje kamere

Na slici 3.4 prikazana je kamera, i vidno polje kamere. Za izračun tipa leća potrebne su dimenzije kao što su: udaljenost kamere od područja koje kamera gleda te širina i dužina radnog vidnog polja.



The image shows a web form for calculating lens types. It contains the following sections:

- Dimensional Unit**: A label "Choose your preferred dimensional unit." followed by a dropdown menu showing "centimeters" with a downward arrow.
- Horizontal Field of View**: A label "Width of area to be imaged, at specified working distance, in inches." followed by a text input field containing the number "25".
- Vertical Field of View**: A label "Height of area to be imaged, at specified working distance, in inches." followed by a text input field containing the number "25".
- Working Distance**: A label "Distance between lens and object, in inches." followed by a text input field containing the number "100".

Below the form is a grey button labeled "Advanced Options" with a downward arrow, and a blue button labeled "Calculate".

Slika 3.5: Kalkulator za leće

Na slici 3.5 prikazan je kalkulator za izračunavanje leća. Unose se već prije spomenute dimenzije, širina i dužina vidnog polja te udaljenost kamere od vidnog polja. Na *Advanced Options* (napredne postavke) mogu se odabrati karakteristike kamere, koje su već unesene, ako je na početku odabrana kamera za koju je potrebna leća.

## Design Inputs

Camera Type:	<u>CABR111 Basler Ace acA1600-60gm camera with e2v EV76C570ABT6 sensor</u>
Pixel Size:	4.50 $\mu\text{m}$
Camera Resolution:	1,600 x 1,200 pixels
Requested Field of View:	25.00 x 25.00 cm
Requested Working Distance:	100.00 cm
Lens Mount:	C-Mount

## Results

Ideal Focal Length:	21.6 mm
Ideal Magnification:	0.022
Ideal World Resolution:	0.2082 mm/pixel
Camera Field of View:	33.33 x 25.00 cm
Camera Image Circle:	9.00 mm
Suggested Area-of-Interest:	1,201 x 1,200 pixels
Suggested Image Circle:	7.64 mm

Slika 3.6: Rezultati nakon unosa podataka

Na slici 3.6 prikazani su rezultati kao što su: idealna dužina fokusiranja, povećanje, i rezolucija te područje vidnog polja kamere.

S dobivenim rezultatima odabrana je leća *Basler Lens C23-2518-5M-P f25mm*.

Karakteristike *Basler Lens C23-2518-5M-P f25mm* jesu:

- nosač objektiva: *C-mount*
- duljina: 37.1mm
- nit filtra: M27 x .5
- težina: 62 g
- iskrivljenje: < 0,92 %

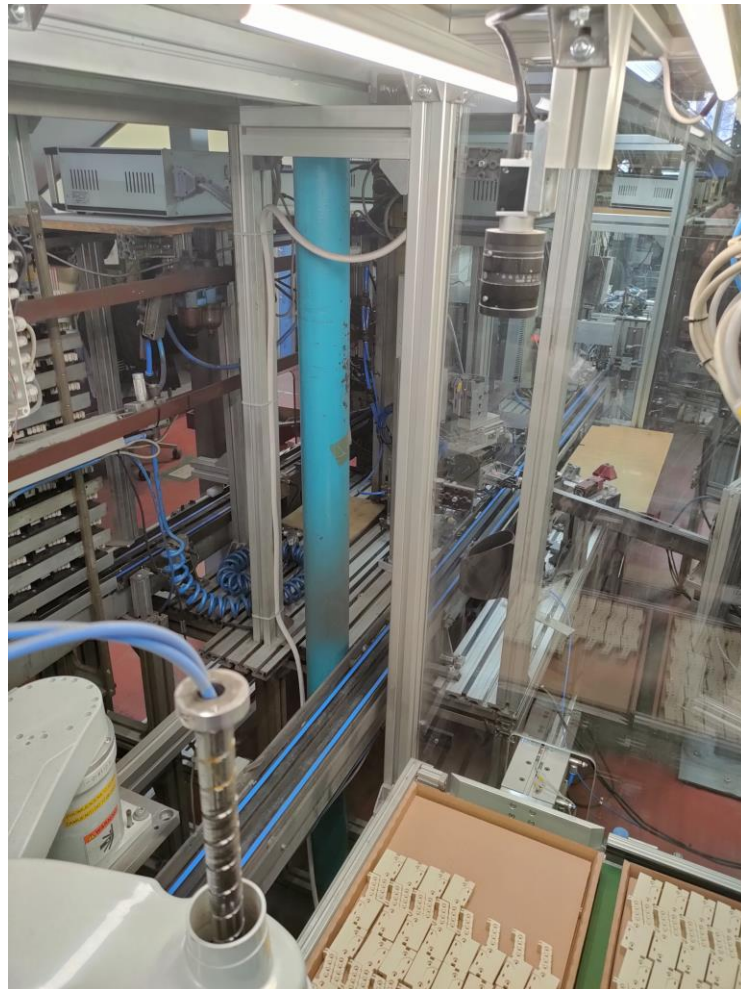
- snaga razlučivanja centra: 3,40  $\mu\text{m}$ , 145 lp/mm
- F#: 1,8 - 16,0
- krug slike: 11,00 mm (2/3")
- duljina fokusiranja: 25mm
- minimalna radna udaljenost: 200mm
- format senzora: 2/3".



Slika 3.7: *Basler Lens C23-2518-5M-P f25mm*

### 3.3 Implementacija kamere

Kamera u ovom sustavu služi za prepoznavanje pozicije komada proizvoda u paleti, odnosno kutiji. Nakon što kamera odredi poziciju komada proizvoda, robot se pozicionira u odnosu na prepoznatu poziciju. Korištena je kamera *Basler Ace acA1600-60gm*, a već ranije u poglavlju 2.3.1 opisana je instalacija iste. Kameru je potrebno ugraditi na mjesto koje gleda kutiju s proizvodima na mjestu s kojeg želimo da robotski manipulator uzima iste.

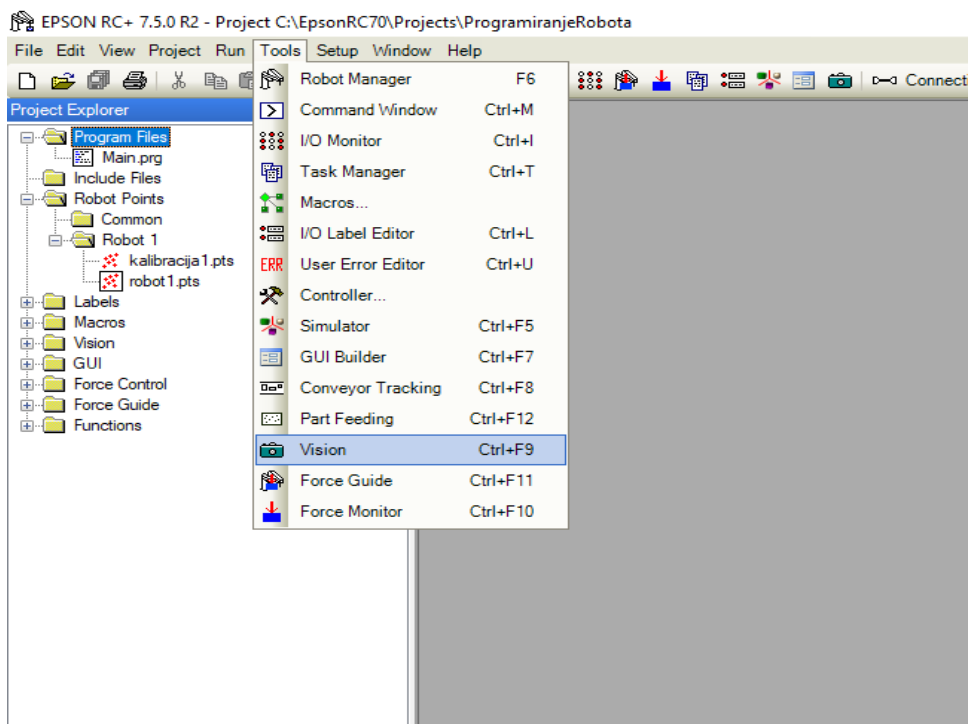


Slika 3.8: Kamera u sustavu

### 3.4 Postavke kamere

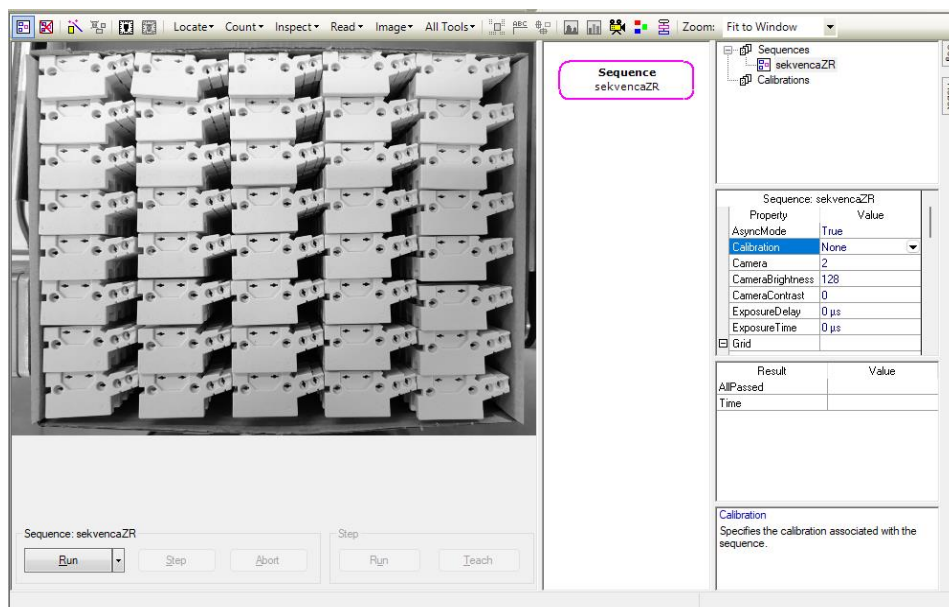
Nakon što je kamera ugrađena te je sve spojeno i upaljeno možemo vidjeti sliku koju producira kamera u realnom vremenu. Da bi se to ostvarilo, potrebno je pokrenuti program *Epson RC+*, otvoriti novi projekt te odabrati opciju *Tools* pa *Vision*. Postupak se može vidjeti na slici 3.9.





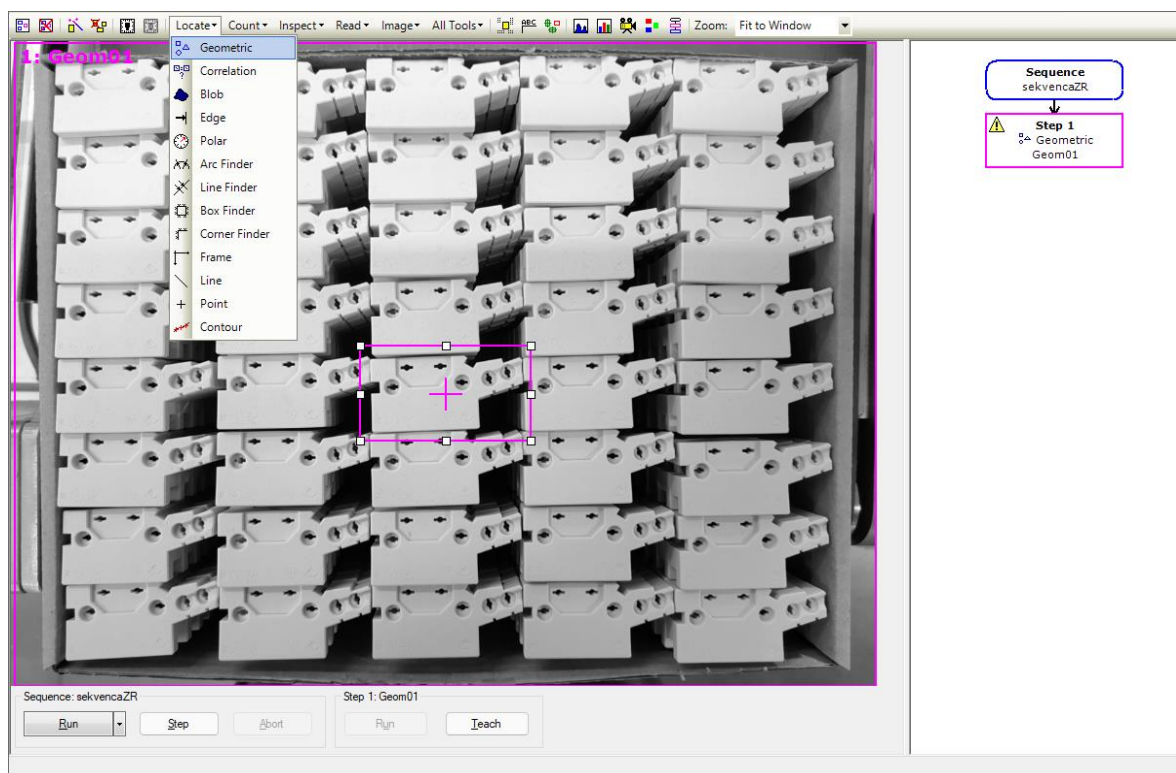
Slika 3.9: *Epson RC+*

Nakon što se napravi nova sekvenca, ako je sve spojeno dobro, trebala bi biti vidljiva slika kamere, kao na slici 3.10.



Slika 3.10: *Epson RC+, Vision Guide*

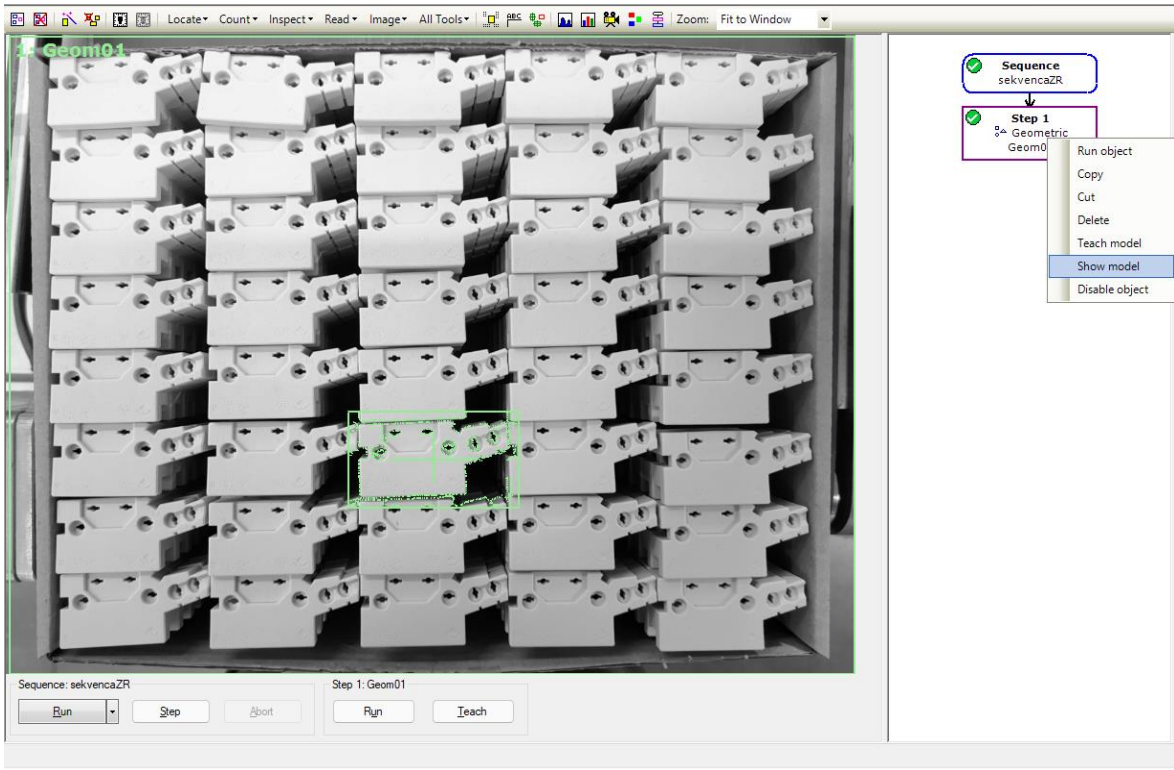
Da bi program prepoznao pojedine komade potrebno je odrediti značajke koje bi prepoznao na komadima proizvoda. To je moguće izvršiti koristeći opciju *Geometric*. Geometrijski objekt pronalazi model na temelju geometrijskih značajki. Koristi algoritamski pristup koji pronalazi modele, koristeći geometrijske značajke temeljene na obrubu nekog dijela. Geometrijski objekt obično se koristi za određivanje položaja i orijentacije objekta, lociranjem značajki na objektu. Ovo se obično koristi za pronalaženje položaja dijelova koji pomažu usmjeravanju robota do položaja podizanja i postavljanja.



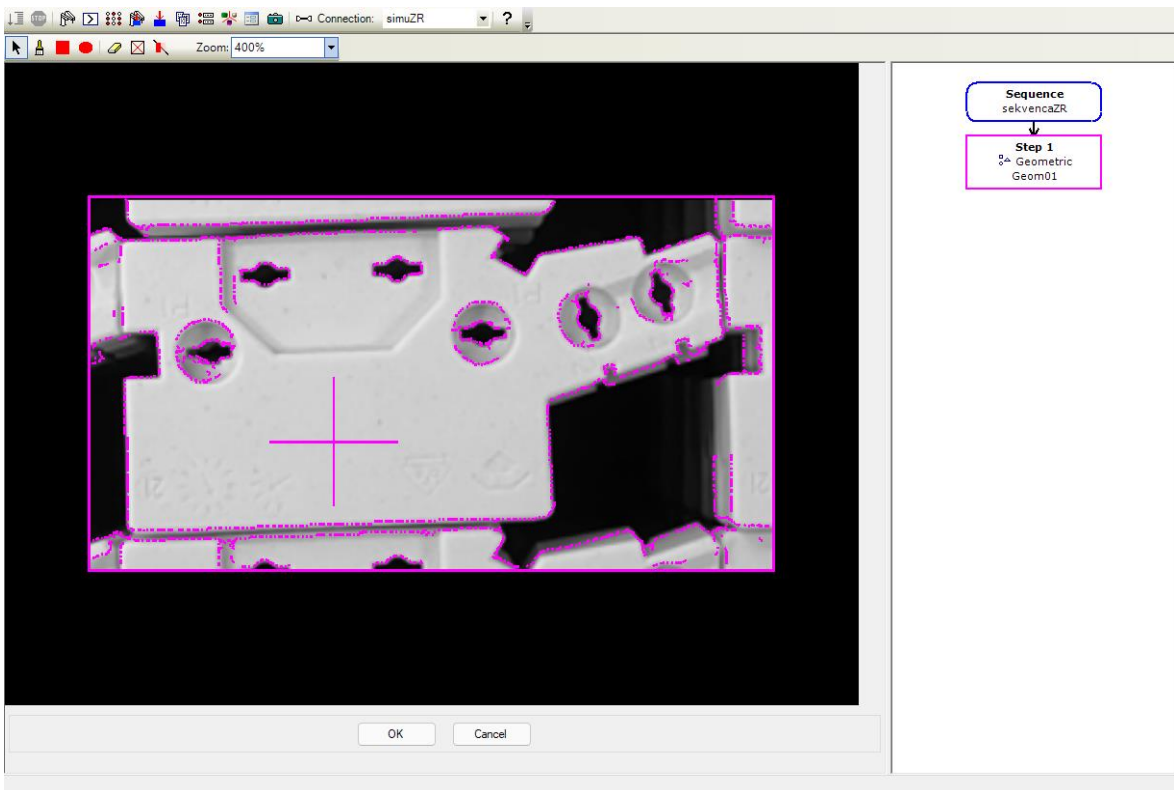
Slika 3.11: *Epson RC+, Vision Guide*

Do naredbe *Geometric* dolazi se pritiskom na *Locate* gdje se odabire *Geometric*, kao što je prikazano na slici 3.11. Kada je odabrana opcija *Geometric*, pojave se dva kvadrata. Veći kvadrat je područje pretraživanja, dok se s manjim označava predmet koji želimo prepoznavati. Oba kvadrata mogu se pomicati i povećavati ili smanjivati. Veći kvadrat koji služi za područje pretraživanja moguće je razvući preko cijele slike budući da je kutija s komadima proizvoda gotovo u cijelom području snimanja. Manji kvadrat se fokusira na jedan komad.

Može se pogledati što program pretražuje pritiskom desne tipke miša na *Step 1*, te odabirom *Show model*.

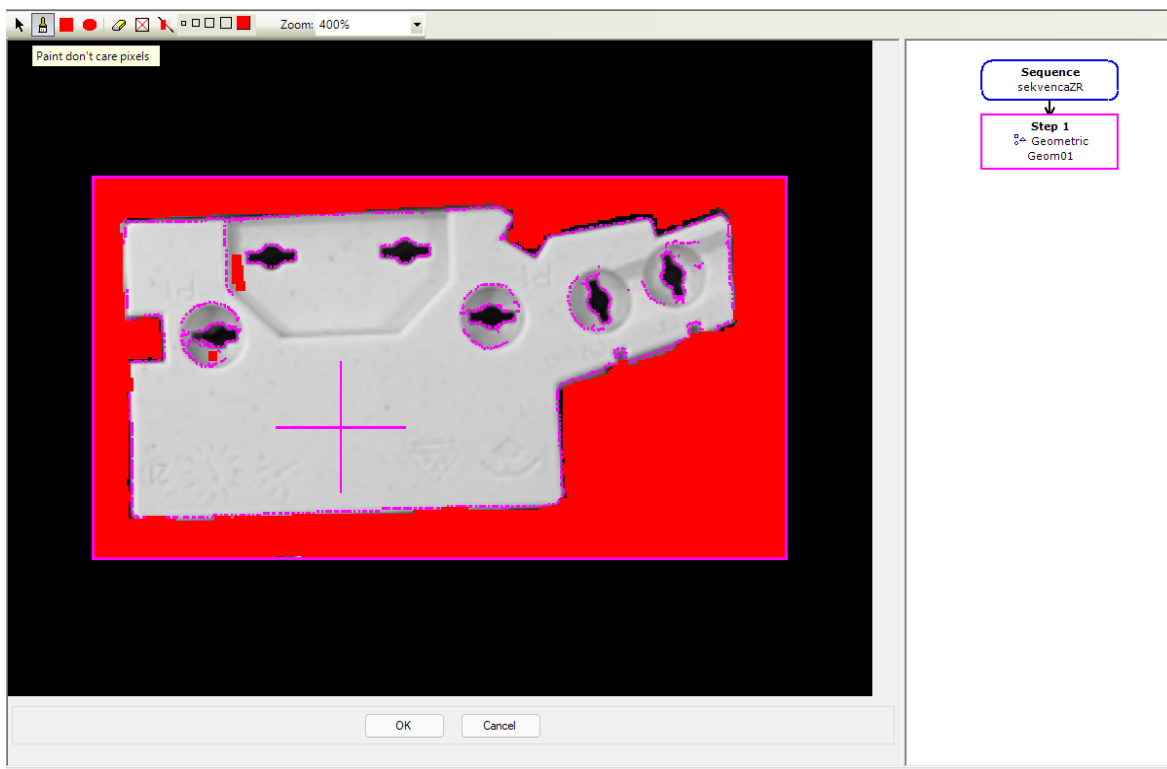


Slika 3.12: *Epson RC+, Vision Guide*



Slika 3.13: *Epson RC+, Vision Guide*

Na slici 3.13 vidljivo je što program pretražuje. Program pretražuje piksele označene ljubičastom bojom. Poželjno je pretraživati samo određene značajke pojedinog komada, zanemarujući susjedne komade proizvoda. Zato se uklanjaju pikseli i sve drugo što smeta u pretrazi. To se može napraviti tako što se jednostavno prekriju pikseli koji smetaju, naredbom *Paint don't care pixels*.

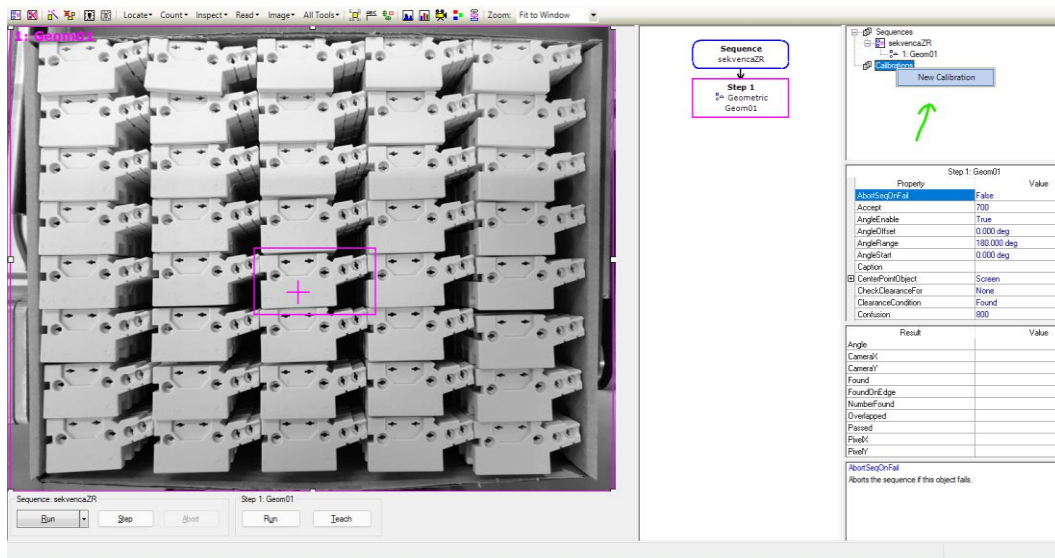


Slika 3.14: *Epson RC+, Vision Guide*

Na slici 3.14 vidimo prekrivene dijelove koji u ovom slučaju nisu bili potrebni. Također je moguće prekriti sve piksele koje nisu potrebni na dijelu koji se traži.

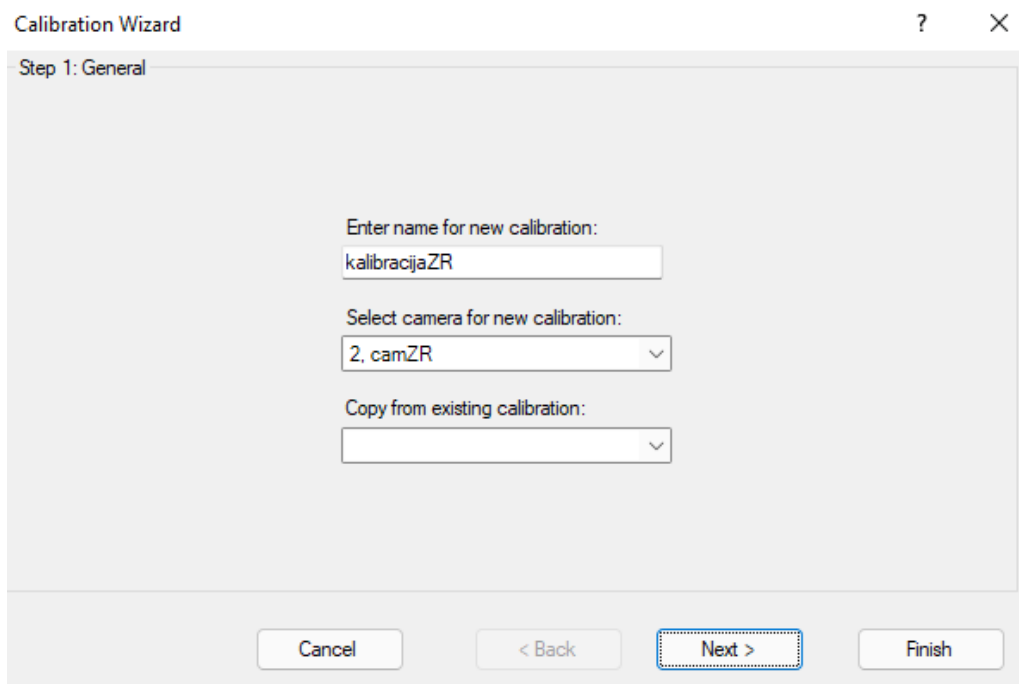
### 3.5 Kalibracija kamere

Kamera je montirana na jednom mjestu, s centrom u sredini kutije s proizvodima koje robot vadi. Zbog toga kamera vidi drugačije komade koji su u centru i oko njega, nego komade koji su uz rub kutije. Razlika možda je jako malena, ali uvelike otežava lociranje proizvoda u kutiji. Radi optimalnijih rezultata pretraživanja poželjno je kalibrirati kameru.



Slika 3.15: *Epson RC+, Vision Guide*

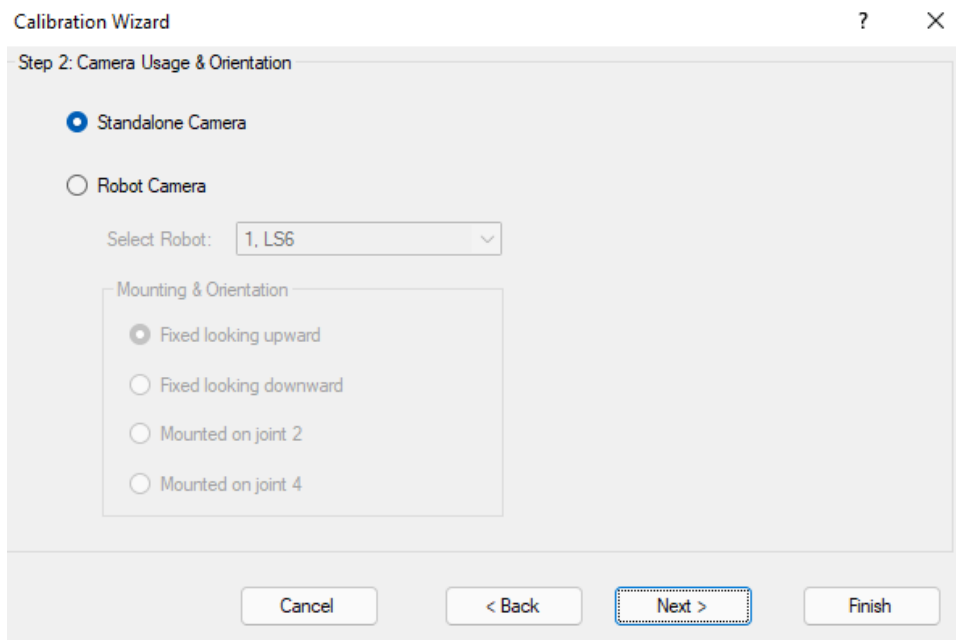
Kamera se kalibrira tako da se pritisne desna tipka na *Calibration* te *New Calibration*. Pojavljuje se novi prozor *Calibration Wizard*, u koji se upisuje naziv kalibracije te se odabire kamera koju je potrebno kalibrirati.



Slika 3.16: *Calibration Wizard*, prvi korak

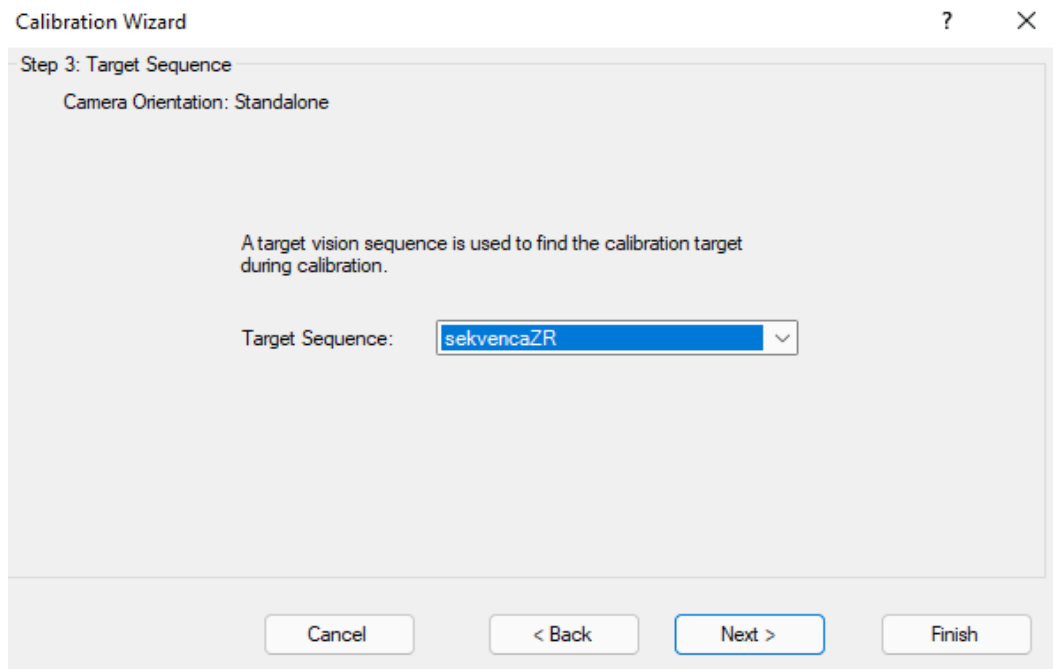
Zatim se odabire opcija *Next*. U sljedećem koraku važno je označiti kružić *Standalone Camera*. Opcija *Standalone* znači da je kamera montirana negdje drugdje, osim

na robotu. Kamera se također može montirati na robotski manipulator. Tada bi bilo potrebno odabrati kružić *Robot Camera*, te odabrati os na kojoj je kamera montirana.



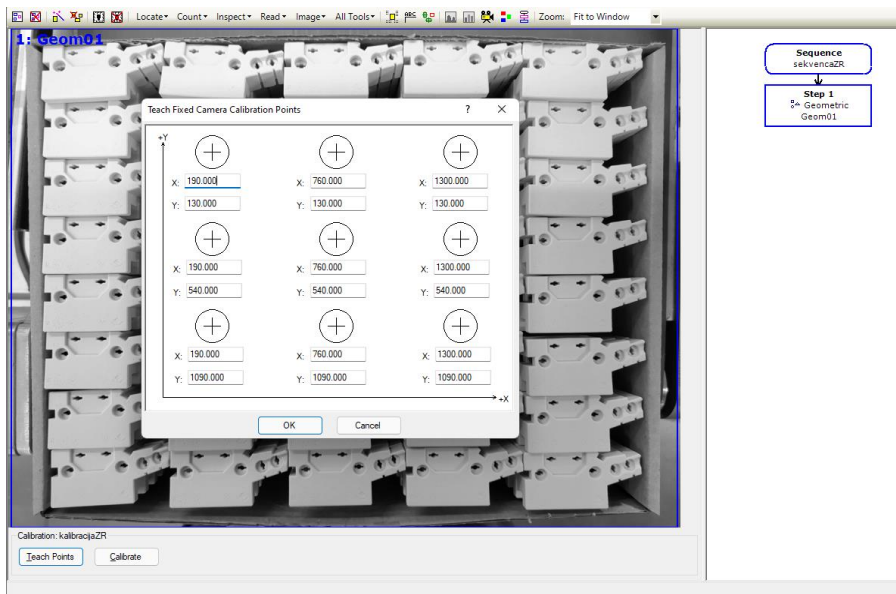
Slika 3.17: *Calibration Wizard*, drugi korak

U sljedećem koraku potrebno je odabrati sekvencu za koju radimo kalibraciju. Nakon što je odabrana sekvenca zatvara se prozor pritiskom na tipku *Finish*.



Slika 3.18: *Calibration Wizard*, treći korak

Nakon pritiska tipke *Finish*, potrebno je naučiti točke za kalibraciju. Točke za kalibraciju možemo naučiti pritiskom na *Teach Points*. Otvara se novi prozor naziva *Teach Fixed Camera Calibration Points*. U tom prozoru upisuju se koordinate devet točaka s odgovarajućim pozicijama kao na slici u otvorenom prozoru. Pritisne se tipka *OK*, zatim *Calibrate*.



Slika 3.19: Učenje točaka

Kalibracija se izvršava u devet točaka te zbog toga je potrebno imati naučenih devet koraka u sekvenci. Na slici 3.20 prikazane su nove vrijednosti nakon kalibracije.



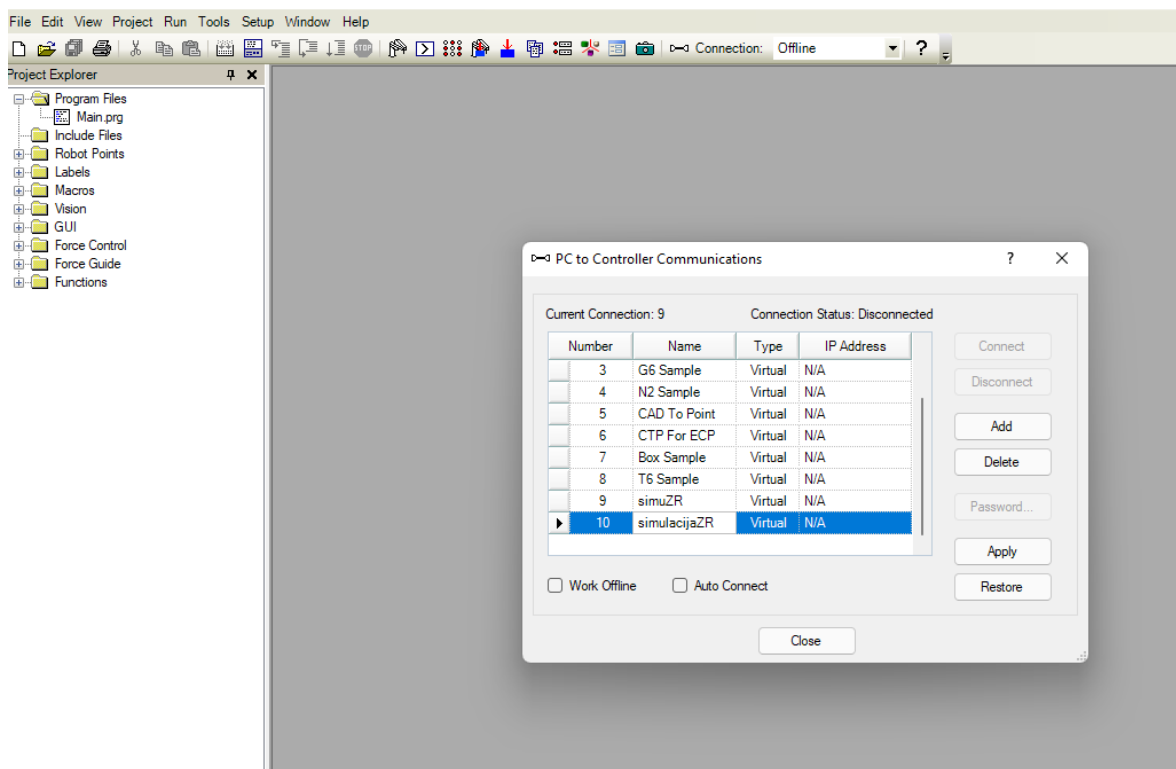
Slika 3.20: Kalibracija

## 4. RAD ROBOTA S KAMEROM

### 4.1. Prikaz simulacije rada robotskog manipulatora

U programu *Epson RC+* moguće je napraviti kompletnu simulaciju robotskog sustava. Moguće je dodati sve komponente koje se koriste u stvarnom sustavu (robotski manipulator, kameru, CAD elemente).

Nakon pokretanja programa potrebno je stvoriti novi projekt, te spojiti kontroler sa simulacijom. Tako da se napravi novi projekt, otvori *Setup* te prozor naziva *PC to Controller Communications*.

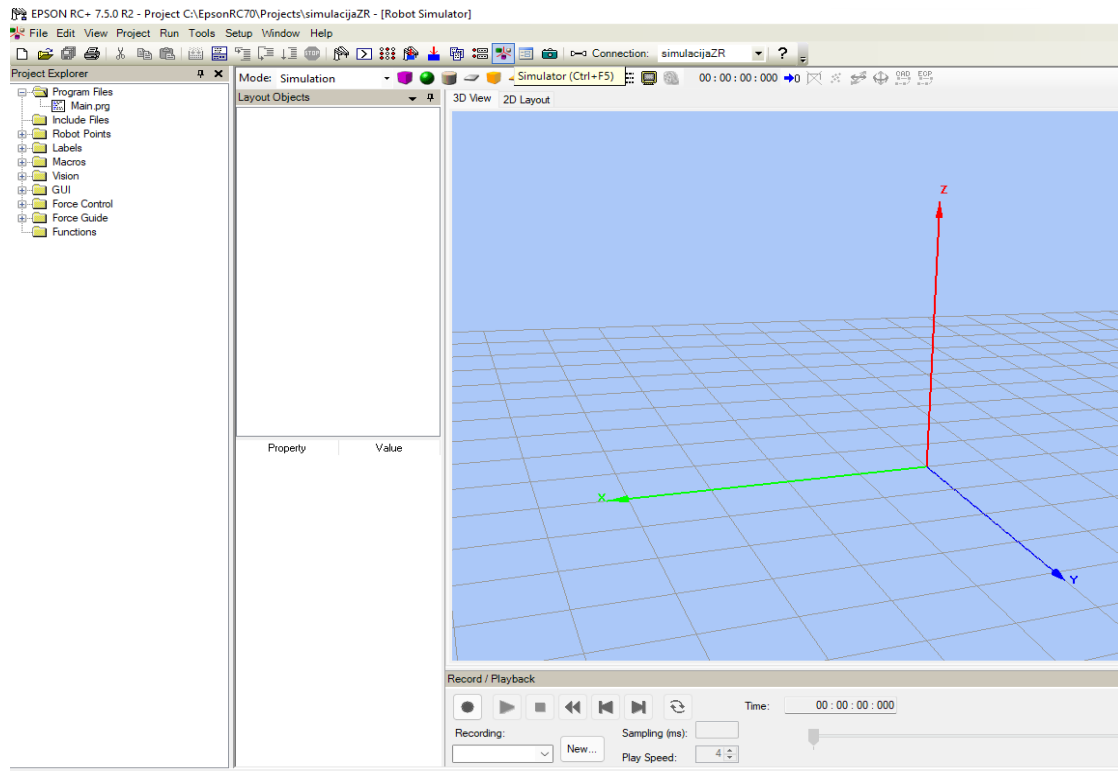


Slika 4.1: *Epson RC+*

Na slici 4.1 otvoren je prozor u kojem se odabire s čime se spojiti. U ovom slučaju kreirano je: *simulacijaZR*, tipa *Virtual* te je na to spojen kontroler. Nakon toga moguće je dodati robotski manipulator te ostale potrebne elemente u simulaciju.

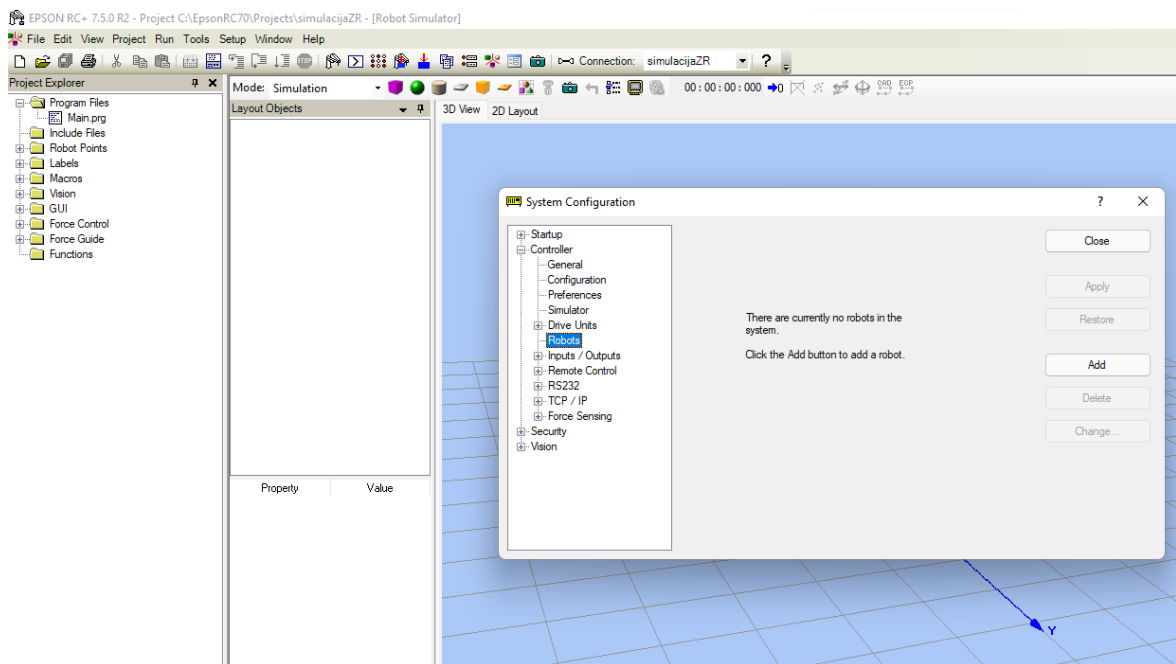
Da bi vidjeli okruženje u kojem radimo simulaciju potrebno je pritisnuti ikonicu kao što je prikazano na slici 4.2 ili pritisnuti tipke *Ctrl + F5*.





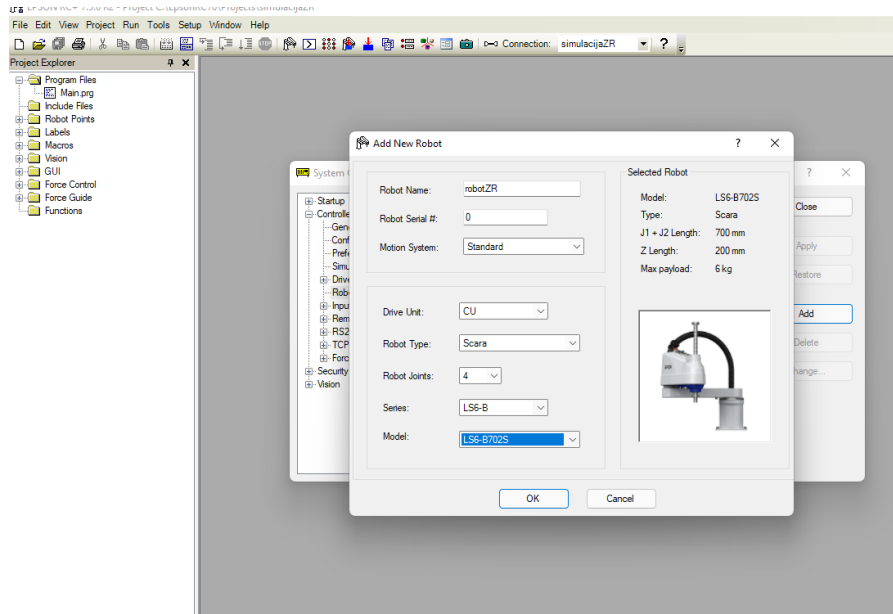
Slika 4.2: *Epson RC+*, simulator

Robotski manipulator dodaje se tako da se otvori *Setup* te pritiskom na *System Configuration*.



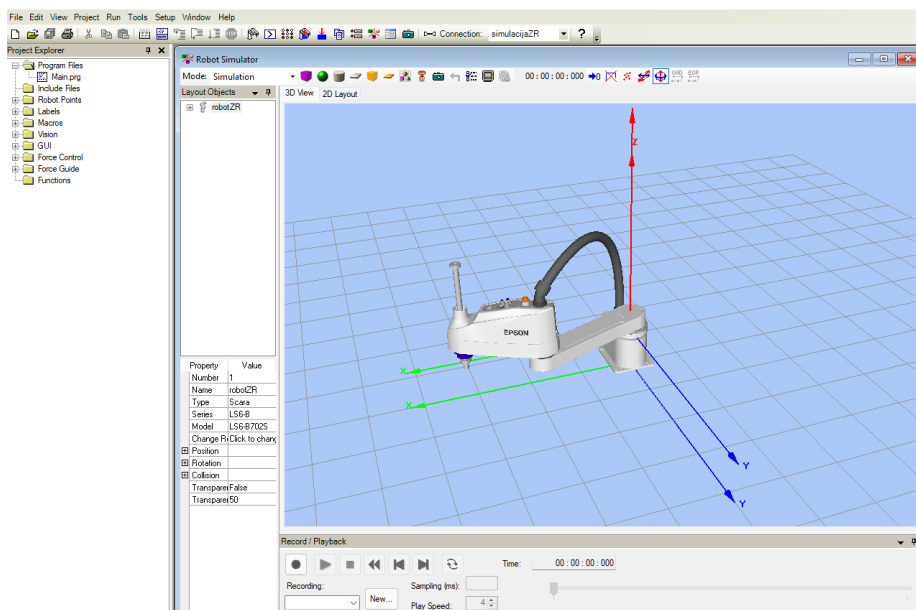
Slika 4.3: *Epson RC+*, simulator

Nakon što je otvoren novi prozor, kao na slici 4.3 proširuje se izbornik *Controller* te se odabire *Robots*. Za dodavanje robotskog manipulatora pritišće se tipka *Add*.



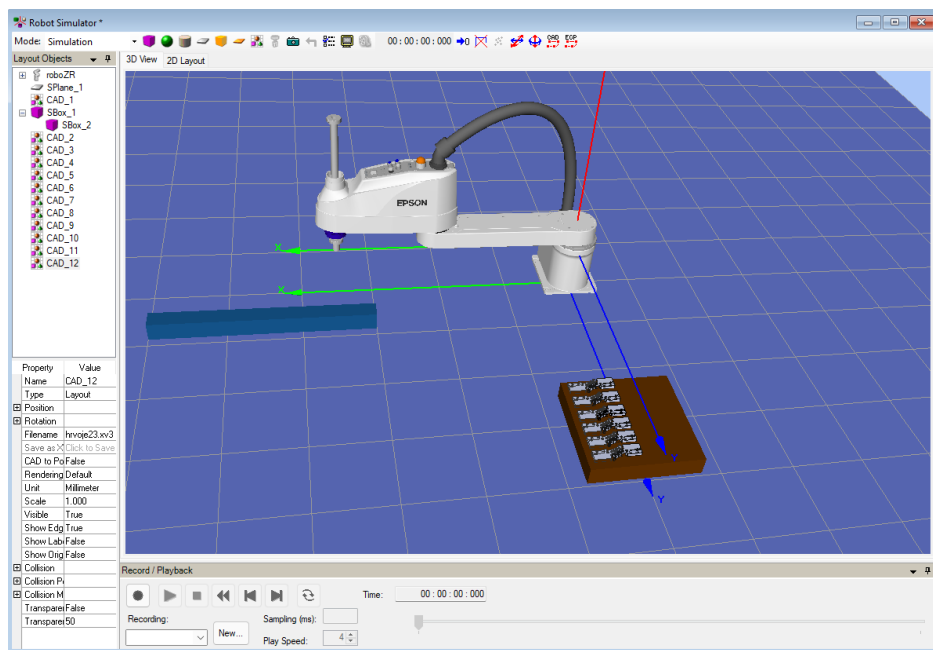
Slika 4.4: *Epson RC+*, odabir robotskog manipulatora

Na slici 4.4 vidljiv je odabrani robotski manipulator. Odabran je robotski manipulator koji je u stvarnom sustavu te je za potrebe simulacije nazvan *robotZR*.



Slika 4.5: *Epson RC+*, simulator

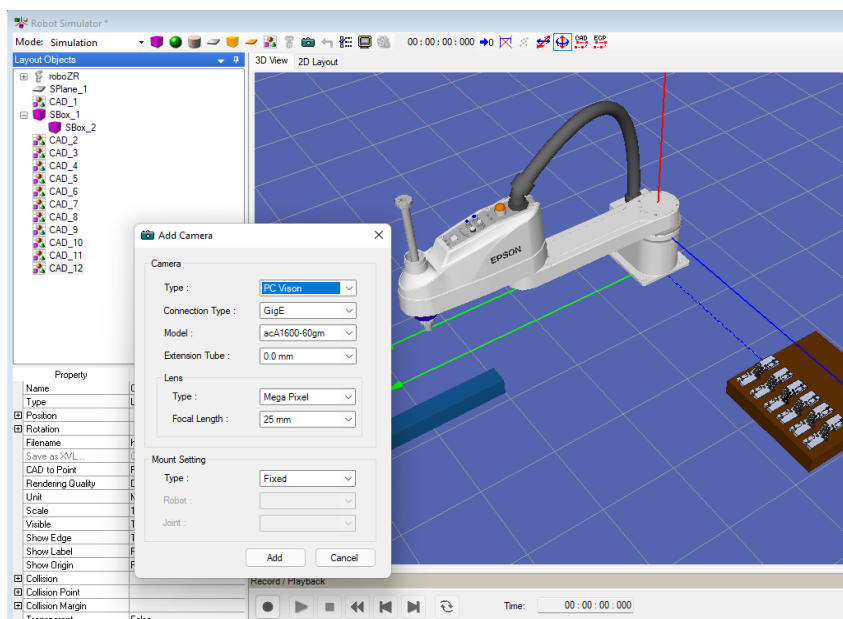
Na slici 4.5 prikazan je izabrani robot u simulacijskom okruženju. Nakon što je dodan robotski manipulator, potrebno je dodati elemente koji bi približili simulaciju realnom okruženju. To se radi tako da u simulacijskom prozoru u traci izbornika odabiremo objekte koje želimo dodati. To mogu biti geometrijski oblici kojima se mogu mijenjati dimenzije ili CAD elementi koje je moguće napraviti u nekom programu za 3D modeliranje. Za potrebe ovog završnog rada napravljen je 3D model proizvoda koji robotski manipulator prenosi iz palete na pokretnu traku.



Slika 4.6: *Epson RC+*, simulator

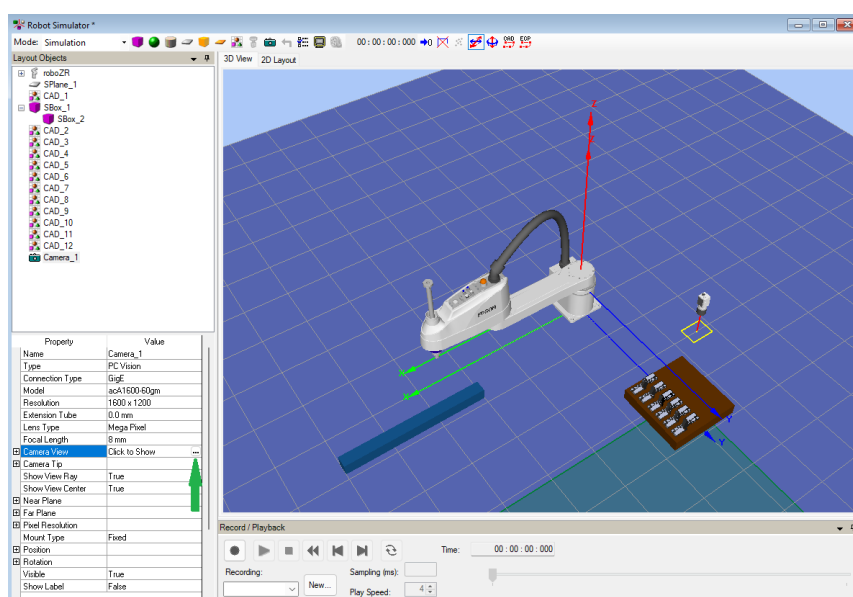
Na slici 4.6 stavljeni su elementi koji prikazuju paletu s 3D modelom proizvoda te element koji predstavlja pokretnu traku.

U simulaciju također možemo dodati kameru. Kameru dodajemo pritiskom na malu ikonicu kamere. Nakon što je pritisnuto otvara se novi prozor u kojem je moguće odabrati kameru te postavke kamere.



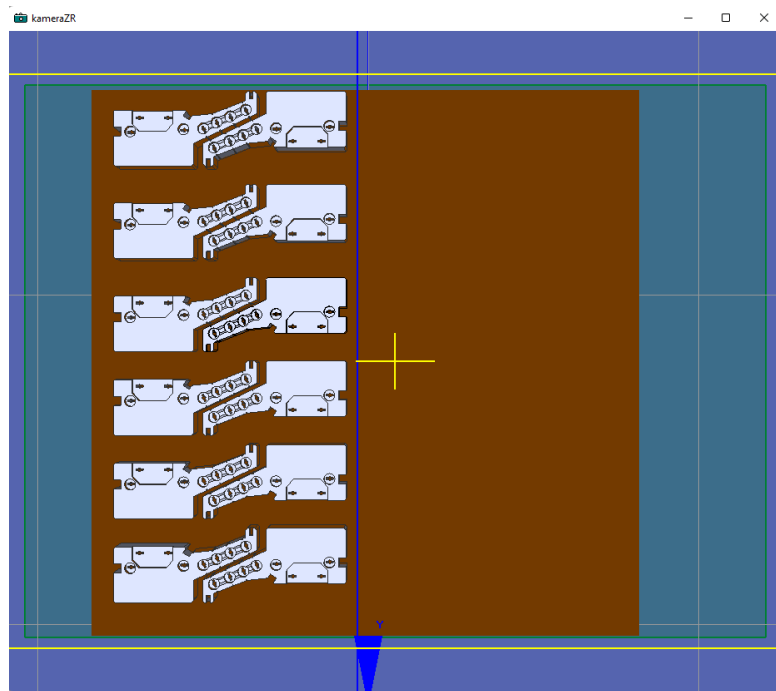
Slika 4.7: Epson RC+, dodavanje kamere

Na slici 4.7 otvoren je novi prozor *Add Camera*, u njemu se odabire kamera koja se koristi te odgovarajuće postavke.



Slika 4.8: Epson RC+, simulator

Na slici 4.8 vidljiva je kamera koja je dodana u simulaciju. Kamera u simulaciji gleda u realnom vremenu. Da bi se vidjelo što kamera gleda odabire se *Camera View* te pritiskom na tri točkice koje su na slici 4.8 označene zelenom strelicom.

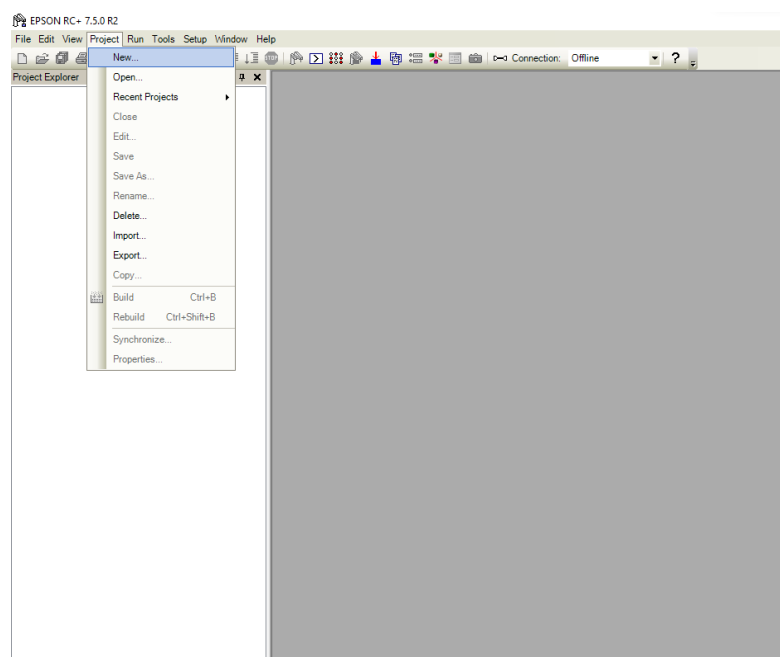


Slika 4.9: Prikaz pogleda kamere

#### 4.1.1 Programiranje robota

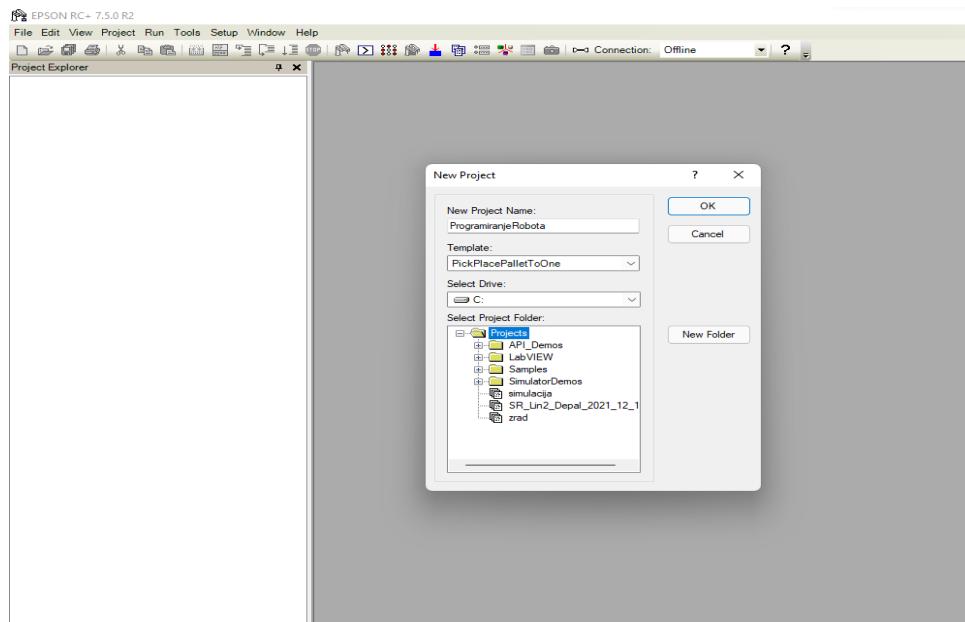
Robot se programira u programu *Epson RC+ 7.0*, a jezik koji se koristi prilikom programiranja zove se *SPEL+*.

Otvara se novi projekt pritiskom na *Project, New*.

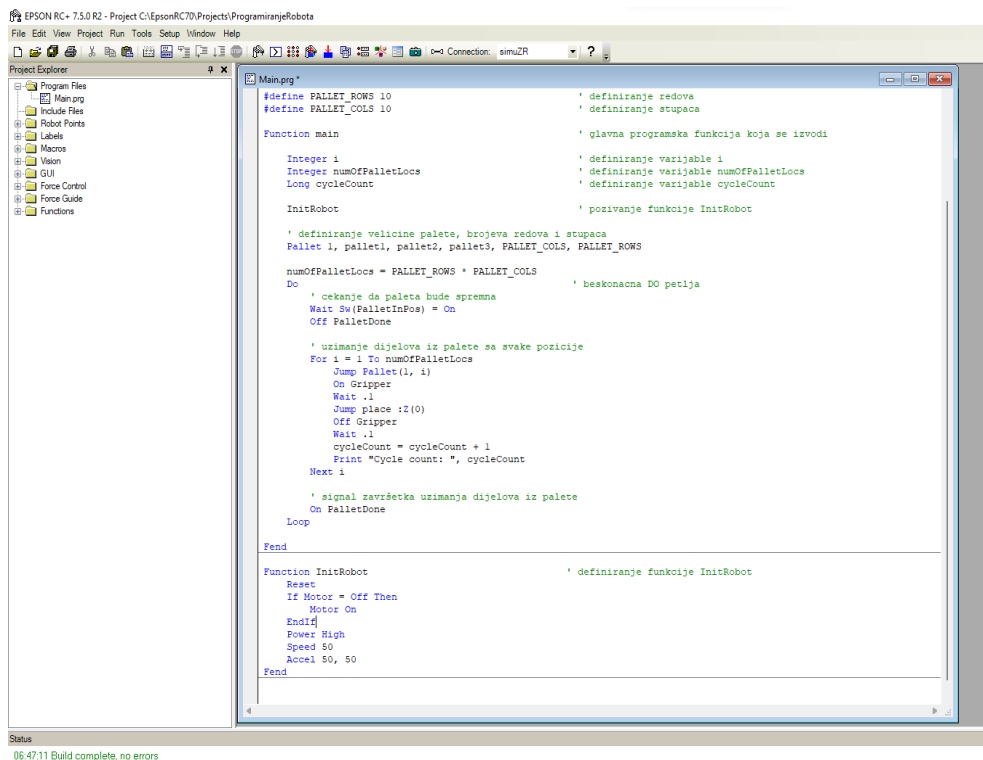


Slika 4.10: *Epson RC+ 7.0*, kreiranje novog projekta

Nakon toga otvara se novi prozor kao na slici 4.11. Ime se odabire proizvoljno, dok se pod predložak može, ali i ne mora odabrati jedna od ponuđenih opcija. Ponuđene opcije su: *Calibration*, *PickPlaceOneToOne*, *PickPlaceOneToPallet*, *PickPlacePalletToOne*. U ovom slučaju odabiremo *PickPlacePalletToOne* budući da robotski manipulator uzima proizvode iz palete te ih stavlja na jedno mjesto.



Slika 4.11: Epson RC+ 7.0, kreiranje novog projekta



Slika 4.12: *Epson RC+ 7.0, Main Program*

Nakon što je kreiran novi projekt otvara se prozor *Main Programa* kao na slici 4.12. Vidljive su dvije funkcije, funkcija *Main* koja se izvodi te funkcija *InitRobot* koja se izvodi ako je pozvana u *Main* funkciji. Radi lakšeg razumijevanja programa objašnjene su neke osnovne naredbe.

Input Commands	
<a href="#">In</a>	Reads one byte (eight bits) of input data.
<a href="#">InBCD</a>	Reads one byte of input bits in Binary Coded Decimal format.
<a href="#">InW</a>	Reads one word (sixteen bits) of input data.
<a href="#">Oport</a>	Reads one output bit.
<a href="#">Sw</a>	Read status of one bit of hardware inputs or memory inputs.
Output Commands	
<a href="#">Off</a>	Turns off one output bit.
<a href="#">On</a>	Turns on one output bit.
<a href="#">OpBCD</a>	Sets one byte of output bits in Binary Coded Decimal format.
<a href="#">Out</a>	Sets/reads one byte (eight bits) of output data.
<a href="#">OutW</a>	Sets/reads one word (sixteen bits) of output data.
Memory I/O Commands	
<a href="#">MemIn</a>	Reads one byte (eight bits) of Memory I/O.
<a href="#">MemInW</a>	Reads one word (sixteen bits) of Memory I/O.
<a href="#">MemOff</a>	Turns off one memory bit.
<a href="#">MemOn</a>	Turns on one memory bit.
<a href="#">MemOut</a>	Sets / reads one byte of memory bits.
<a href="#">MemSW</a>	Reads one bit of memory.

Slika 4.13: Osnovne naredbe *I/O*

Na slici 4.13 vidljive su naredbe za korištenje ulaza, izlaza i memorijskih bitova. Za ulaze najčešće se koristi naredba *Sw* pomoću koje se u programu čita stanje na ulazu. Izlazi se uključuju i isključuju naredbama *On* i *Off*. Na slici 4.13 također je vidljiva naredba za korištenje memorijskih bitova.

Command	Description
<a href="#">Go</a>	Move directly to a point.
<a href="#">Jump</a>	Jump to a point. First move up to the current <a href="#">LimZ</a> setting, the move over the destination point, then move to the point. The <a href="#">Arch</a> table settings determine the Jump profile.
<a href="#">Jump3</a>	Jump to a point in 3 dimensions. Move in a straight line with the same orientation until the recede point. The motion between the recede points is PTP motion.
<a href="#">Pass</a>	Move near one or more points.
<a href="#">TGo</a>	Move directly to a point in a tool coordinate system.
<a href="#">BGo</a>	Move in a PTP motion to the relative specified point in Base / Local coordinate system

Slika 4.14: Osnovne naredbe I/O

Na slici 4.14 vidljive su naredbe za pomicanje robotskog manipulatora. Naredba *Go* vodi robotski manipulator iz točke u točku, najdirektnijom putanjom. Naredba *Jump* prvo pomiče Z os prema gore pa onda ide do željene destinacije.

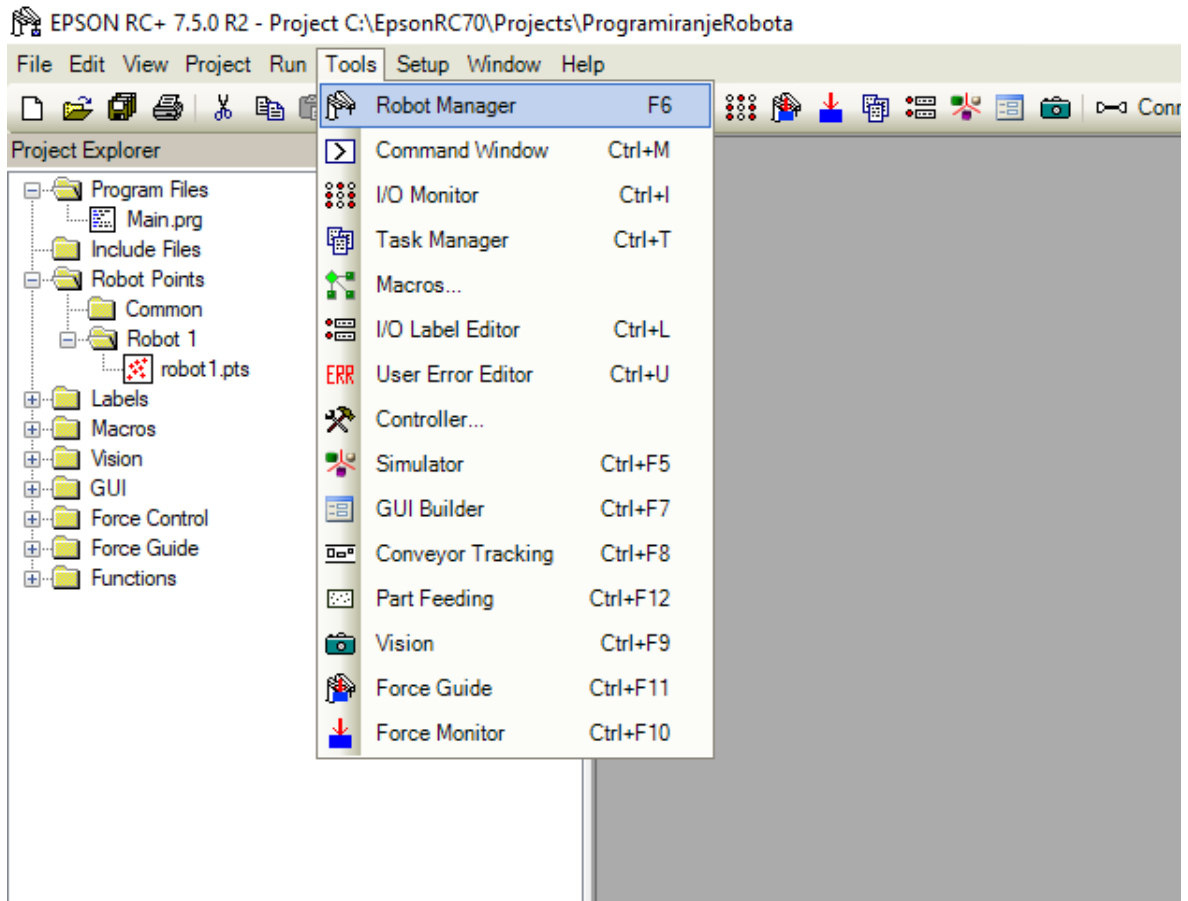
#### 4.1.2 Učenje točaka

Za uzimanje komada, kao i za ostavljanje komada, potrebno je znati njihove koordinate u prostoru. Točke uzimanja ili ostavljanja se uče tako što se robotski manipulator dovede u poziciju u kojoj je potrebno naučiti točku te se pritišće tipka *Teach* u programu.

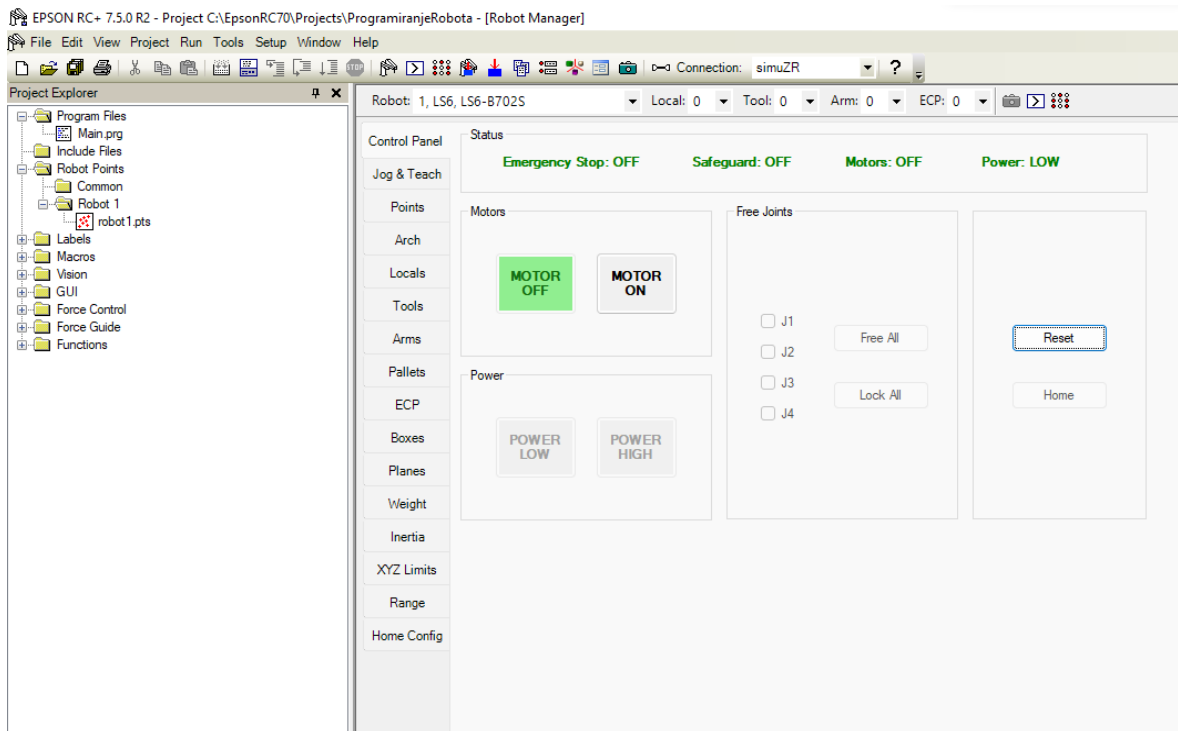
Postoji nekoliko načina za dovesti robotski manipulator u neku željenu poziciju. Mogu se upisivati koordinate te se robot pomiče sukladno njima, mogu se odabrati osi te pomicati robota grafičkim strjelicama u željenom smjeru ili se mogu osloboditi sve osi robota te ručno dovesti robotski manipulator u željenu poziciju.

Pritiskom na *Tools*, zatim na *Robot Manager* otvorit će nam se *Robot Manager* u novom prozoru. Postupak možemo vidjeti na slici 4.15.



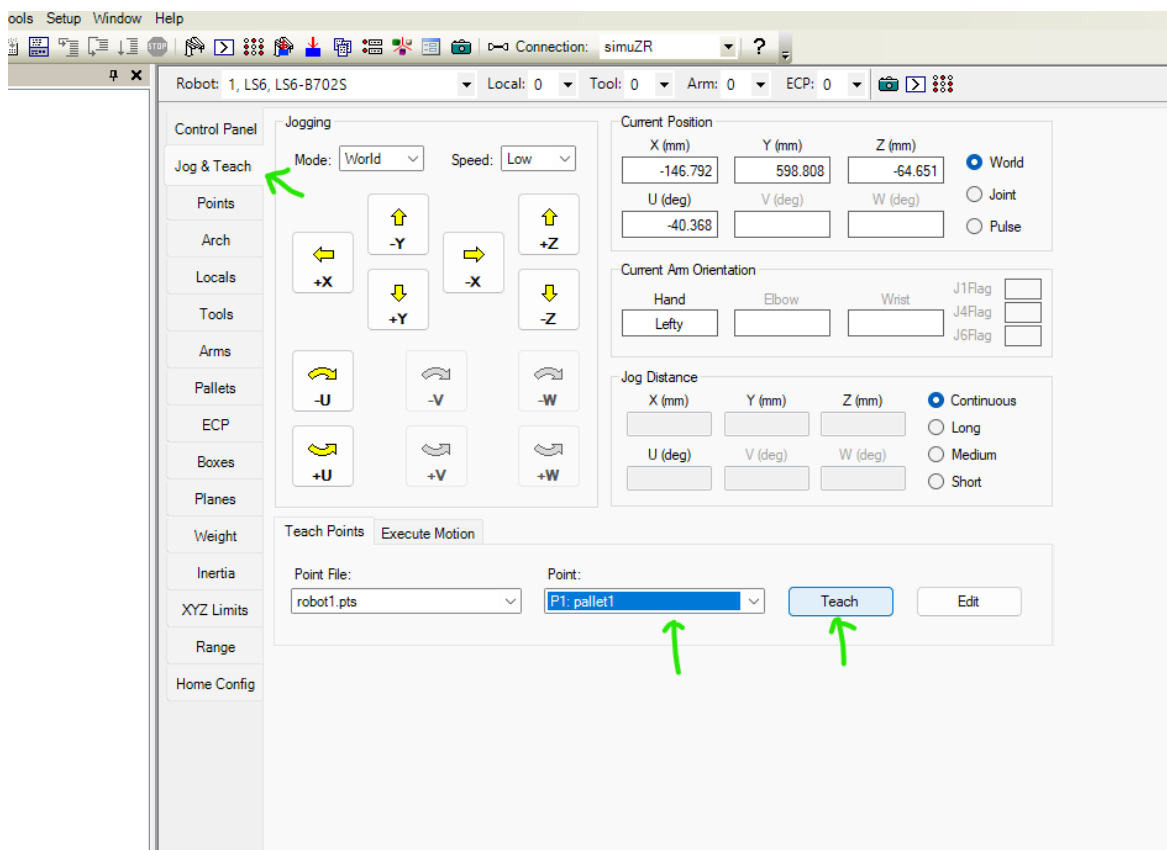


Slika 4.15: Epson RC+ 7.0, otvaranje Robot Managera



Slika 4.16: Epson RC+ 7.0, sučelje Robot Managera

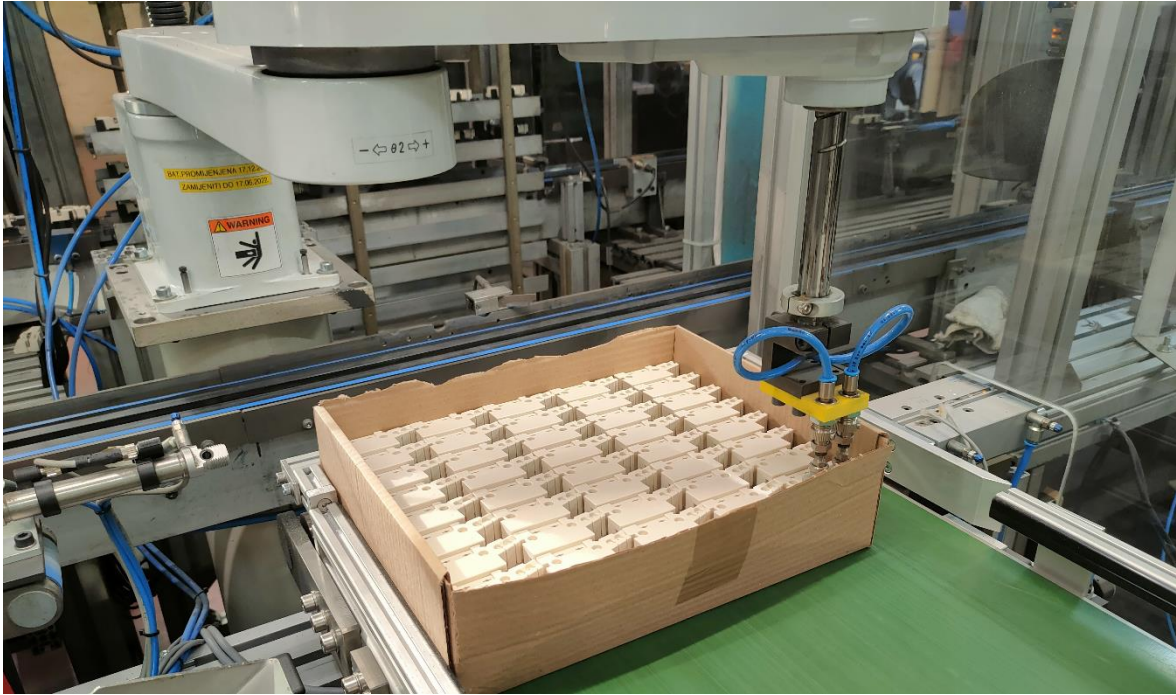
Nakon što je pokrenut *Robot Manager*, u *Control Panelu* uključuju se servomotori robota. Također moguće je resetirati robota, dovesti u *Home* poziciju te mijenjati režim rada sile. Ako bismo željeli robota pomicati ručno, nakon što bismo upalili motore, pritisnemo, u dijelu gdje su *Free Joints*, *Free All* kvadratić. Sada je moguće ručno pomicanje robota. Nakon što je robot doveden u željenu poziciju odabire se *Jog & Teach*.



Slika 4.17: *Epson RC+ 7.0*, sučelje *Jog & Teach*

Na slici 4.17 vidljivo je sučelje *Jog & Teach* koje je namijenjeno pomicanju robota te učenju točaka. Pritiskom na žute strelice pomičemo robota u željenom smjeru, po željenoj osi. Moguće je upisati koordinate te će se robot pomicati po danim koordinatama. Preporučljivo je to sve raditi u sporij brzini, da bismo smanjili šanse za stvaranje bilo kakve nezgode.

U našem slučaju potrebno je naučiti tri točke palete te poziciju ostavljanja proizvoda iz palete. Koristeći metodu ručnog pomicanja robot je doveden u točke koje je potrebno naučiti.



Slika 4.18: Robotski manipulator u poziciji učenja točke

Nakon što je robot u koordinatama koje su potrebne za učenje, pritiskom na tipku *Teach* pod *Point* se stavlja odgovarajuća pozicija. U ovom slučaju je to *pallet1*, što je označeno zelenom strelicom na slici 4.17.

Robot: 1, LS6, LS6-B702S    Local: 0    Tool: 0    Arm: 0    ECP: 0

Control Panel    Point File: robot1.pts

	Number	Label	X	Y	Z	U	Local	Hand
Points	0	place	310.842	254.072	-64.652	-144.368	0	Lefty
Arch	1	pallet1	-146.792	598.808	-64.651	-40.368	0	Lefty
	2	pallet2	-209.222	353.429	-64.651	-61.368	0	Lefty
Locals	3	pallet3	115.625	384.456	-64.652	-110.368	0	Lefty
	4							
Tools	5							
Arms	6							
	7							
Pallets	8							
ECP	9							
	10							
Boxes	11							

Slika 4.19: Prikaz naučenih točki

Ako pritisnemo *Points*, lijevo u izborniku otvorit će nam se prozor kao na slici 4.19. Na slici su vidljiva imena te koordinate naučenih točki. Ovdje se također mogu mijenjati koordinate točki.

### 4.1.3 Funkcija *Main*

*Main* funkcija je glavna funkcija koja se izvodi u programu. U njoj se mogu pozivati funkcije kao što je funkcija *InitRobot*.

```
#define PALLET_ROWS 8           ' definiranje redovi palete
#define PALLET_COLS 5         ' definiranje stupci palete

Function main                  ' glavna programska funkcija koja se izvodi

    Integer i                  ' definiranje varijable i
    Integer numOfPalletLocs    ' definiranje varijable numOfPalletLocs

    InitRobot                  ' pozivanje funkcije InitRobot

    ' definiranje velicine palete, brojeva redova i stupaca
    Pallet l, pallet1, pallet2, pallet3, PALLET_COLS, PALLET_ROWS

    numOfPalletLocs = PALLET_ROWS * PALLET_COLS
    Do                          ' beskonacna DO petlja
        ' cekanje da paleta bude spremna
        Wait Sw(PalletInPos) = On
        Off PalletDone

        ' uzimanje dijelova iz palete sa svake pozicije
        For i = 1 To numOfPalletLocs
            Jump Pallet(1, i)
            On Gripper
            Wait .1
            Jump place :Z(0)
            Off Gripper
            Wait .1

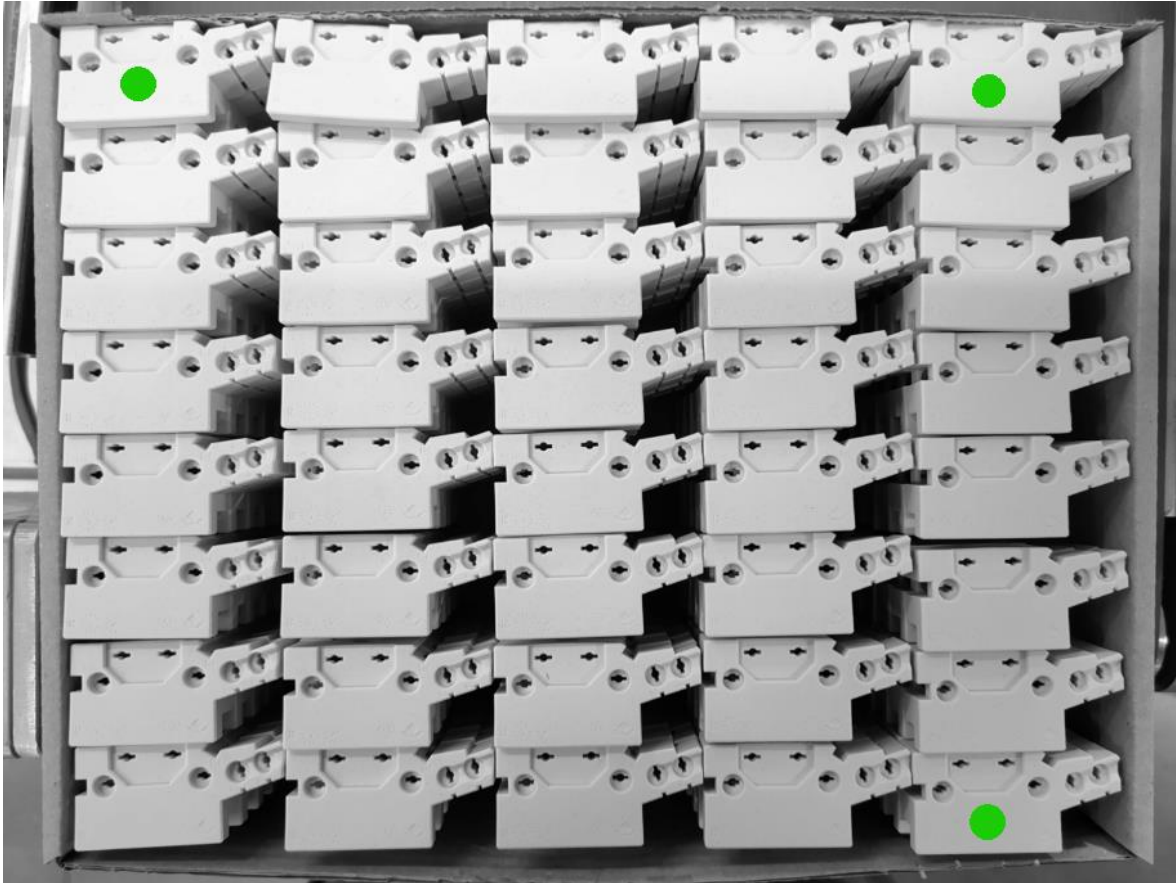
        Next i

        ' signal završetka uzimanja dijelova iz palete
        On PalletDone
    Loop

Fend
```

Slika 4.20: Funkcija *Main*

Slika 4.20 prikaz je *Main* funkcije programa. Definirano je *PALLET\_ROWS* i *PALLET\_COLS* te su im pridruženi brojevi. *PALLET\_ROWS* su brojevi redaka, a ima ih osam, dok su *PALLET\_COLS* brojevi stupaca kojih je pet. Prikaz palete moguće je vidjeti na slici 4.21.



Slika 4.21: Prikaz palete s 5 stupaca i 8 redova

U funkciji *Main* definirane dvije su varijable. Varijabla *I* te *numOfPalletLocs* koje su tipa *Integer*. Te varijable služe za petlju *For*. Varijabla *I* služi za brojanje, dok varijabla *numOfPalletLocs* služi za izračun broja komada proizvoda koji se nalaze u paleti.

U *Main* funkciji potrebno je definirati veličinu palete. Za to se uzimaju koordinate tri točke, koje su na slici 4.21 označene zelenom bojom. U programu se veličina palete definira naredbom *Pallet 1*. Naredbi *Pallet 1* se pridružuju točke prihvata: gornja lijeva (*pallet1*), gornja desna (*pallet2*), donja desna (*pallet3*) i broj redaka te stupaca (*PALLET\_COLS* i *PALLET\_ROWS*).

Nakon definiranja varijabli program počinje izvoditi *DO* petlju, koja se beskonačno vrti.

```

Do                                     ' beskonacna DO petlja
  ' cekanje da paleta bude spremna
  Wait Sw(PalletInPos) = On
  Off PalletDone

  ' uzimanje dijelova iz palete sa svake pozicije
  For i = 1 To numOfPalletLocs
    Jump Pallet(1, i)                 ' robot ide u poziciju uzimanja komada proizvoda
    On Gripper                        ' prihvati proizvod
    Wait .1                           ' cekanje 0.1 sekundu
    Jump place :Z(0)                  ' robot ide u poziciju gdje ostavlja proizvod
    Off Gripper                       ' puštanje proizvoda
    Wait .1                           ' cekanje 0.1 sekundu

  Next i                               ' i++

  ' signal završetka uzimanja dijelova iz palete
  On PalletDone
Loop

```

Slika 4.22: Prikaz *DO* petlje programa

Na slici 4.22 prikaz je *DO* petlje. U početku izvođenja *DO* petlje čeka se da paleta bude spremna za rad, naredbom *Wait Sw (PalletInPos) = On*. Nakon što je paleta spremna, robotski manipulator počinje uzimati komade redom. Rad robota u *DO* petlji opisan je slikom 4.22. Nakon što je uzet zadnji komad program čeka da nova paleta bude spremna.

#### 4.1.4 Funkcija *InitRobot*

U funkciji *InitRobot* definiramo postavke robota kao što su: brzina, sila, resetiranje motora, paljenje motora i slično.

```

Function InitRobot                    ' definiranje funkcije InitRobot
  Reset                               ' resetiranje servo motora
  If Motor = Off Then                ' uključenje napajanja
    Motor On
  EndIf
  Power High                          ' rezim rada velikom silom
  Speed 50                            ' brzina
  Accel 50, 50                       ' ubrzanje, usporavanje
Fend

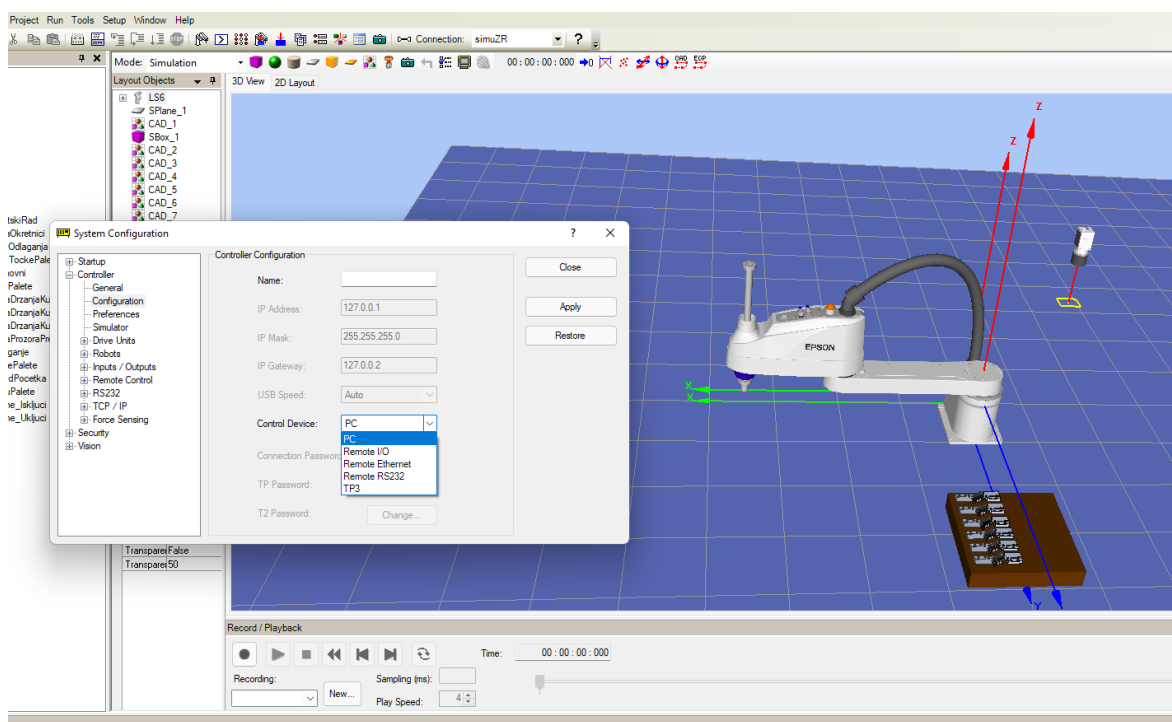
```

Slika 4.23: Funkcija *InitRobot*

Slika 4.23 prikazuje funkciju *InitRobot*. Potrebno je resetirati servomotore tako da prijašnje stanje ne utječe na daljnji razvoj programa. Nakon toga se pomoću *If* petlje pokreće motor. Daljnje izvođenje programa moguće je samo ako je motor pokrenut. *Power* se može mijenjati iz *Low* u *High*. *Low* je za manju silu prilikom rada robota, dok je *High* za veću silu. Budući da u okruženju robota nema nikakvih prepreka, odnosno ništa što bi robotski manipulator mogao udariti, odabrana je opcija *High* kako je bilo u izvornom kodu programa. Preostala je još brzina te ubrzavanje i usporavanje robotskog manipulatora.

#### 4.1.5 Pokretanje programa

Ovim robotskim manipulatorom može se upravljati na više načina. Neki od načina su upravljanje pomoću računala te pomoću ulaza koji mogu biti: obično tipkalo, senzor ili signal s nekog *PLC*-a.

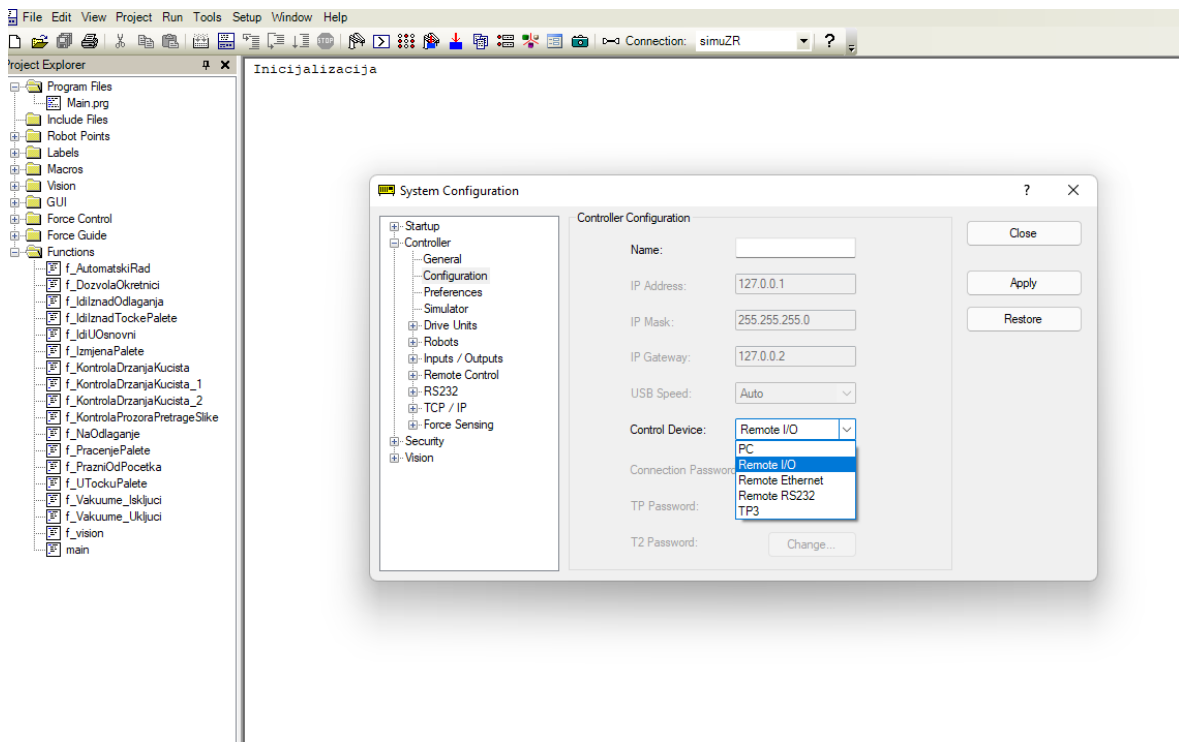


Slika 4.24: Odabir upravljanja robotom

Na slici 4.24 prikazano je što se može odabrati za upravljanje robotom. Posebno je zanimljivo upravljanje računalom preko *USB* kabela te upravljanje pomoću *I/O* ulaza, izlaza.

Nakon spajanja na kontroler *USB* kabelom, u programu *Epson RC+* u izborniku se odabire opcija *Run* te *Run Window*. Nakon toga se u novo otvorenom prozoru pomoću tipke *Run* se pokreće program. Robot će tada početi izvoditi *Main Program*.

Kada bismo željeli upravljati ulazima, u programu bi se trebao promijeniti uređaj kojim se želi upravljati robotom. To se može napraviti na način da se odabere opcija *Setup*, te *System Configuration*.

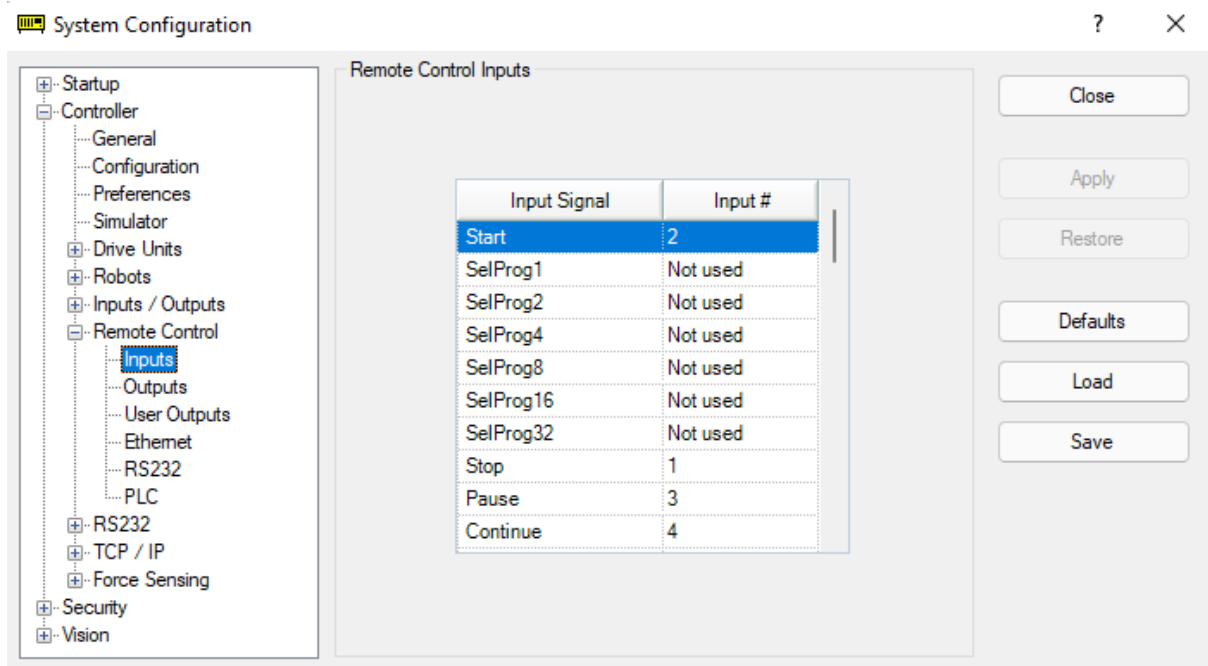


Slika 4.26: *System Configuration*

Na slici 4.26 vidljiv je novo otvoreni prozor *System Configuration*. U tom prozoru odabire se *Controller* te *Configuration*. Nakon toga se pod opcijom *Control Device* mijenja s *PC* na *Remote I/O*. Tako je omogućeno upravljanje robota pomoću ulaza.

Upravljanje pomoću ulaza funkcionira tako da, ako je u programu definiran neki ulaz koji služi za izvođenje programa, taj ulaz mora biti zadovoljen kako bi se program izvodio. Također su potrebni ulazi pomoću kojih bi se izvršavale glavne funkcije robota.





Slika 4.27: System Configuration, ulazi

Na slici 4.27 prikaz je ulaza te njihovih funkcija. Na ulazu 1 je *Stop*, dok je na ulazu 2 *Start*. Ulaz je napon od 24 V koji se može dovesti preko tipkala ili *PLC*-a.

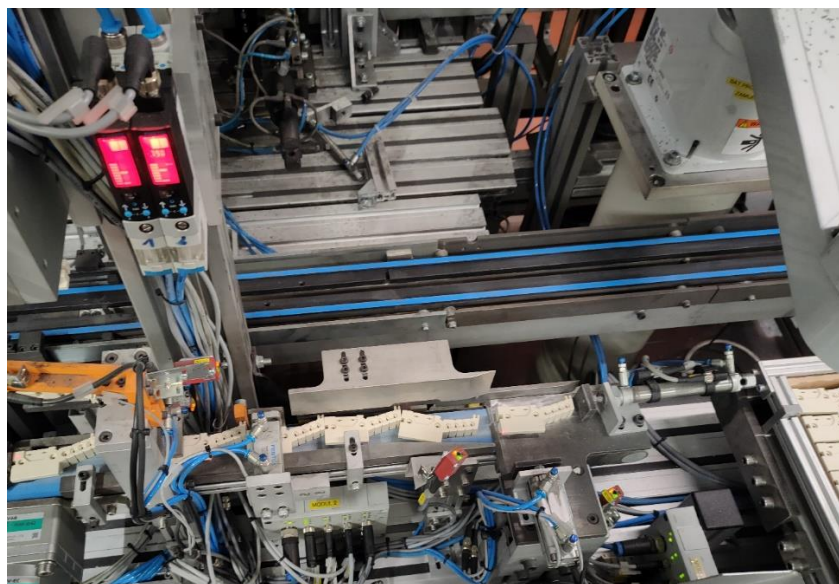
## 4.2 Prikaz stvarnog sustava robotskog manipulatora

Robotski manipulator *Epson SCARA LS6-B702S* u ovom sustavu služi za uzimanje proizvoda redom iz kutije. Robotski manipulator proizvod stavlja na pokretnu traku, uvijek u istu poziciju.



Slika 4.28: Robotski manipulator u sustavu

Na slici 4.28 prikazan je robotski manipulator koji uzima proizvod iz kutije te ga vodi prema točki ispuštanja. Robot je opremljen vakuumskim sisaljkaama koje služe za uzimanje proizvoda iz kutije. Nakon što se kutija isprazni, robotski manipulator uzima kutiju te ju stavlja na za to predviđeno mjesto.



Slika 4.29: Pozicija u kojoj robot ostavlja proizvod

Nakon što su robotski manipulator i kamera osposobljeni, potrebno je uskladiti jedno s drugim. To se radi u programu *Epson RC+*. U funkciji *Main* potrebno je pozvati funkcije koje služe za prepoznavanje orijentacije komada te ostale funkcije koje su potrebne za rad robota.

U programu se koriste ulazi koje šalje *PLC* (programabilni logički kontroler). To su ulazi kao: kutija s proizvodom za izdvajanje je spremna za rad, pokreni robotski manipulator, zaustavi, nastavi, resetiraj i slično.

The screenshot shows the 'I/O Label Editor (Ctrl+L)' window. On the left is a tree view of I/O categories. The main area is a table with columns for 'Input Bit', 'Label', and 'Description'. The table lists 24 input bits from 0 to 23.

Input Bit	Label	Description
0	iReset	St01: Robot <- reset emergency stopa ili greske programa robota (IN_00)
1	iStop	St01: Robot <- nalog za stopiranje programa robota (IN_01)
2	iStart	St01: Robot <- nalog za pokretanje programa robota (IN_02)
3	iPause	St01: Robot <- nalog za pauziranje programa robota (IN_03)
4	iContinue	St01: Robot <- nalog za nastavljjanje programa robota (IN_04)
5	iUOsnovni	St01: Robot <- nalog za odlazak u osnovni polozej (IN_05)
6	iSljedeciKorak	St01: Robot <- sljedeci korak u koracnom radu ili potpuno automatski (IN_06)
7	iUzmiKucista	St01: Robot <- nalog za uzimanje 1 komada iz kutije (IN_07)
8	iOdloziKucista	St01: Robot <- nalog za odlaganje 1 komada na stazu (IN_08)
9	iPrazniOdPocetka	St01: Robot <- nalog za depaletiranje iz nulte tocke (IN_09)
10	iSLOBODNO_10	St01: Robot komunikacija (IN_10)
11	iSLOBODNO_11	St01: Robot komunikacija (IN_11)
12	ilzbaciPraznu	St01: Robot <- nalog za izbacivanje prazne kutije (IN_12)
13	iSLOBODNO_13	St01: Robot komunikacija (IN_13)
14	iVAK_1_Drzi	IN_14: Vakuum 1 je ukljucen
15	iVAK_2_Drzi	IN_15: Vakuum 2 je ukljucen
16	iSLOBODNO_16	St01: Robot komunikacija (IN_16)
17	iSLOBODNO_17	St01: Robot komunikacija (IN_17)
18	iSLOBODNO_18	St01: Robot komunikacija (IN_18)
19	iSLOBODNO_19	St01: Robot komunikacija (IN_19)
20	iSLOBODNO_20	St01: Robot komunikacija (IN_20)
21	iSLOBODNO_21	St01: Robot komunikacija (IN_21)
22	iSLOBODNO_22	St01: Robot komunikacija (IN_22)
23	iSLOBODNO_23	St01: Robot komunikacija (IN_23)

Slika 4.30: Popis ulaza

*PLC* služi kao *Master* u ovom sustavu te se preko njega pokreće rad robotskog manipulatora. Pomoću *PLC*-a se pokreće, resetira, stopira i dovodi robot u osnovni položaj.

Output Bit	Label	Description
0	qRunning	St01: Robot vrti prog (moze primati i slati signale koji nisu sistem.) (OUT_00)
1	qPaused	St01: Robot je u pauzi (treba mu poslati continue da ode u running) (OUT_01)
2	qAlarm	St01: Robot javlja gresku ili upozorenje za bateriju ili podmazivanje (OUT_02)
3	qUOsnovnom	St01: Robot je u osnovnom položaju (OUT_03)
4	qIznadKutijeSKom	St01: Robot drži komade iznad kutije (OUT_04)
5	qOdlozioKucista	St01: Robot odložio 1 komad na traku (OUT_05)
6	qNovuKutiju	St01: Robot treba novu kutiju (OUT_06)
7	qIzbacioPraznu	St01: Robot izbacio praznu kutiju (OUT_07)
8	qDozvolaPremaSt01	St01: Robot daje dozvolu rada St01 (OUT_08)
9	qDozvolaPremaSt02	St01: Robot daje dozvolu rada St02 (OUT_09)
10	qOslobodiKutiju	St01: Robot zahtjeva oslobađanje trenutne kutije (OUT_10)
11	qNijePronadeno	St01: Kamera nije pronasla traženi objekt (OUT_11)
12	qSLOBODNO_12	St01: Robot komunikacija (OUT_12)
13	qSLOBODNO_13	St01: Robot komunikacija (OUT_13)
14	qVAK_1_ON	OUT_14 -> Vakuum 1 uključi
15	qVAK_2_ON	OUT_15 -> Vakuum 2 uključi

Slika 4.31: Popis izlaza

Memory Bit	Label	Description
0	mDrzi_1_VAK	
1	mDrzi_2_VAK	
2		
3		
4	mIzvadio_1_KUCIS	
5	mIzvadio_2_KUCIS	
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16	mUOsnovnoj	Robot je u osnovnoj poziciji
17	mIznadKutijeSKom	Robot je izvadio komad iz kutije i ceka iznad kutije
18	mNovuKutiju	Robot zahtjeva novu kutiju
19	mOslobodiKutiju	Robot zahtjeva oslobađanje trenutne kutije
20	mIzmjenaUToku	Izmjena palete je u toku
21	mDozvolaPremaSt02	St01: Robot daje dozvolu rada St02 - okretnici
22		
23		
24		
25		
26		

Slika 4.32: Popis memorijskih bitova

#### 4.2.1 Funkcije za vakuum

Korišteni robotski manipulator opremljen je vakuumskim sisaljka. Kada robotski manipulator dođe u poziciju uzimanja, vakuum se uključi te se tako uzima proizvod iz kutije.

Isto tako kada robot dođe u poziciju gdje treba ostaviti proizvod vakuum se isključuje te robotski manipulator ostavlja komad.



Slika 4.33: Vakuumske sisaljke

```
Function f_Vakuume_Ukljuci                                'ukljucivanje vakuma
  On qVAK_1_ON
  On qVAK_2_ON
  Wait 0.3
Fend

Function f_Vakuume_Iskljuci                              'iskljucivanje vakuma
  Off qVAK_1_ON
  Off qVAK_2_ON
  Wait 0.1
Fend
```

Slika 4.34: Funkcije za vakuum

Na slici 4.34 prikazane su funkcije za upravljanje vakuumom. Vakuumi se uključuju pomoću funkcije nazvane *f\_Vakuume\_Ukljuci*. U toj funkciji uključuju se izlazi *qVAK\_1\_ON* i *qVAK\_2\_ON*. Ti izlazi idu na *PLC* te *PLC* aktivira vakuume. Ostavljeno je vrijeme od 0.3 sekunde tako da sa sigurnošću vakuumske sisaljke prime komad.

Vakuumi se isključuju funkcijom naziva *f\_Vakuume\_Iskljuci*. U toj funkciji jednostavno se isključuju izlazi iz robotskog kontrolera te se vakuum gasi.

#### 4.2.2 Funkcija za kontrolu slike

Ova funkcija služi za određivanje pozicije prozora pretraživanja komada. Kada ova funkcija ne bi bila dostupna, robotski manipulator bi uzimao bilo koji pronađeni komad. Kako bi se to izbjeglo te kako bi robotski manipulator komade uzimao željenim redom, definirane su pozicije na slici. U tim pozicijama traže se geometrijski objekti koji su ranije definirani, kao na slici 3.14.



Slika 4.35: Slika palete

Na slici 4.35 prikaz je palete s komadima proizvoda koje robotski manipulator uzima. Razlikujemo komade koje ne treba okretati te komade koji su okrenuti za 180 stupnjeva. Komade koji su okrenuti za 180 stupnjeva robot uzima u drugačijoj poziciji.

```

Function f_KontrolaProzoraPretrageSlike

    g_brojacPalete = g_brojacPalete + 1

    j_180 = j_180 + 1
    j = j + 1

    If j_180 = 8 And i_180 = 0 Then      ' prvi stupac komada okrenutih za 180 stupnjeva
        i_180 = 1
        j_180 = 0
        i = 0
        j = 0
    EndIf
    If j = 8 And i = 0 Then            ' prvi stupac normalno orijentiranih komada
        i_180 = 1
        j_180 = 0
        i = 1
        j = 0
    EndIf
    If j_180 = 8 And i_180 = 1 Then   ' drugi stupac komada okrenutih za 180 stupnjeva
        i_180 = 2
        j_180 = 0
        i = 1
        j = 0
    EndIf
    If j = 8 And i = 1 Then           ' drugi stupac normalno orijentiranih komada
        i_180 = 2
        j_180 = 0
        i = 2
        j = 0
    EndIf
    If j_180 = 8 And i_180 = 2 Then   ' treci stupac komada okrenutih za 180 stupnjeva
        i_180 = 3
        j_180 = 0
        i = 2
        j = 0
    EndIf
    If j = 8 And i = 2 Then           ' treci stupac normalno orijentiranih komada
        i_180 = 0
        j_180 = 0
        i = 0
        j = 0
    EndIf

Fend

```

Slika 4.36: Funkcija za kontrolu pretrage slike

Na slici 4.36 prikazana je funkcija za kontrolu pretrage slike. U funkciji se pomoću brojača postavlja željeno mjesto pretraživanja.

Kada se u jednom stupcu izbroji osam redaka, prebacuje se u sljedeći stupac. Tako se broji do zadnjeg stupca. U paleti se nalazi šest stupaca (tri stupca komada zakrenutih za 180 stupnjeva i tri stupca normalno zakrenutih komada) te osam redaka.

### 4.2.3 Funkcija za pretragu slike s kamere

```
Function f_vision      ' funkcija za pronalazak komada i njihove orijentacije u kutiji
  Boolean obj_found

  obj_found = 0

  x0_slike_180 = 1280  ' pozicija u kojoj se nalazi prvi komad koji je okrenut za 180
  y0_slike_180 = 1045

  x0_slike = 1138     ' pozicija u kojoj se nalazi prvi komad normalne orijentacije
  y0_slike = 1045

  x_korak_slike = 455 ' udaljenost po x osi između komada
  y_korak_slike = 112 ' udaljenost po y osi između komada

  If paletal = 1 Then
    VSet SekvencaZR.ImageSource, VISION_IMAGESOURCE_CAMERA 'Vset naredba za postavke slike sa kamere, kalibracije i pronalazenja objekta
    VSet SekvencaZR.Calibration, "KalibracijaZR"
    VSet SekvencaZR.GeoM01.SearchWinCenterX, x0_slike - i * x_korak_slike
    VSet SekvencaZR.GeoM01.SearchWinCenterY, y0_slike - j * y_korak_slike
    VRun SekvencaZR ' VRun naredba za pokretanje sekvence
    VGet SekvencaZR.GeoM01.Found, obj_found ' Vget naredba za dobivanje vrijednosti svojstava i rezultata
  If obj_found = 1 Then
    Print "obj found"
    VGet SekvencaZR.GeoM01.RobotX, x_rob0
    VGet SekvencaZR.GeoM01.RobotY, y_rob0
    VGet SekvencaZR.GeoM01.RobotU, u_rob0
    z_rob0 = ZosVisinaVadenja
    Print "koordinate: ", x_rob0, y_rob0, z_rob0, u_rob0
  Else
    Print "nothing found"
    On qNijePronadeno
    Quit f_AutomatskiRad
  EndIf
EndIf
```

Slika 4.37: Funkcija za pretragu slike s kamere

Na početku funkcije vidljive na slici 4.37, deklarirana je nova varijabla *obj\_found* tipa *BOOL*. Ta varijabla služi za daljnji rad u funkciji. U kutiji se nalaze orijentirani komadi koji su od normalno orijentiranih, rotirani za 180 stupnjeva.





Slika 4.38: Komadi proizvoda u kutiji

Dalje su u funkciji postavljene koordinate početnih komada obje orijentacije te razmak između njih. Za pretragu slike s kamere koriste se naredbe *VSet*, *VRun* i *VGet*.

Nakon traženja slike koristi se *if* petlja, u kojoj se, ako je objekt pronađe, pridružuju koordinate nađenog geometrijskog oblika naredbama *VGet*. Naredbom *VGet SekvencaZR.Geom01.RobotX, x\_rob*, pridružuje se X koordinata u varijablu nazvanu *x\_rob*. Isto tako se radi i za Y os te za U os koja je rotacija glave robotskog manipulatora.

#### 4.2.3.1 Naredbe za pretragu slike

*VSet* se koristi za postavljanje vrijednosti svojstava za sekvence vida, kalibracije i objekata iz *SPEL+* jezika. Sekvence koje su napravljene u *Vision Guide* dijelu programa se ovdje koriste. Međutim, postoje slučajevi kada će se htjeti postaviti vrijednosti svojstava u

*SPEL+* programu prije pokretanja sekvence vizije. Kada se *VSet* pozove iz programa, promjene se vrše samo u memoriji i ne spremaju se. *VRun* se koristi za pokretanje sekvence. *VGet* se koristi za dobivanje vrijednosti pretrage slike.

```
VSet SekvencaZR.ImageSource, VISION_IMAGESOURCE_CAMERA      'Vset naredba za postavke slike sa kamere, kalibracije i pronalazenja objekta
VSet SekvencaZR.Calibration, "KalibracijaZR"
VSet SekvencaZR.Geom01.SearchWinCenterX, x0_slike - i * x_korak_slike  ' smjestanje podrucja pretrage slike po x osi
VSet SekvencaZR.Geom01.SearchWinCenterY, y0_slike - j * y_korak_slike  ' smjestanje podrucja pretrage slike po y osi
VRun SekvencaZR                                             ' VRun naredba za pokretanje sekvence
VGet SekvencaZR.Geom01.Found, obj_found                      ' Vget naredba za dobivanje vrijednosti svojstava i rezultata
```

Slika 4.39: Naredbe za pretragu slike

Na slici 4.39 prikazan je dio programa u kojem su pozvane naredbe za pretragu slike, pokretanje sekvence i dobivanje rezultata sa slike.

Za *VSet* naredbu potrebno je definirati odakle se dobiva slika te s kojom sekvencom se radi. U ovom slučaju to je kamera te se upisuje: *VISION\_IMAGESOURCE\_CAMERA*, dok je sekvenca sa kojom se radi nazvana *SekvencaZR*. Dalje se upisuje koja se kalibracija koristi te poslije kalibracije što se traži u sekvenci. U sekvenci se traži geometrijski oblik nazvan *Geom01*, koji se može vidjeti na slici 3.13. Područje pretrage slike određuje se naredbama *VSet SekvencaZR.Geom01.SearchWinCenterX, x0\_slike - i \* x\_korak\_slike*, te *SekvencaZR.Geom01.SearchWinCenterY, y0\_slike - j \* y\_korak\_slike*.

#### 4.2.4 Funkcija za pražnjenje palete

Robotski manipulator počinje prazniti paletu kada dobije određene uvjete.

```

Function f_PrazniOdPocetka
Do
  If Sw(iPrazniOdPocetka) = 1 And MemSw(mUOsnovnoj) = 1 Then
    Print "Praznjenje kutije od pocetka"

    Off qIzbacioPraznu
    Off qNovuKutiju
    MemOff mNovuKutiju

    g_brojacPalete = 1
    i = 0
    j = 0
    i_180 = 0
    j_180 = 0

    paletal = 0
    paletal_180 = 0

    kutija_prazna = 0
    kutija_prisutna = 1

    Wait MemSw(mIzmjenaUToku) = 0

  EndIf

  Wait 0.05 'kratko cekanje da procesor kontrolera ne vrtil na 100% mogucnosti

Loop
Fend

```

Slika 4.40: Funkcija za pražnjenje kutije

Na slici 4.40 prikazana je *DO* petlja u kojoj se pomoću *IF* uvjeta pokreće pražnjenje palete. Uvjeti koji se traže za pražnjenje palete su: signal iz *PLC*-a (*iPrazniOdPocetka*), te aktivan memorijski bit (*mUOsnovnoj*).

#### 4.2.5 Funkcija za izmjenu palete

Funkcija za izmjenu palete služi za micanje prazne palete. Robotski manipulator uzima praznu paletu te ju stavlja na za to predviđeno mjesto. To je napravljeno tako da se u funkciji pomiče robotski manipulator naredbom *GO* u već ranije naučene točke. Pri tome se mora paziti na prepreke između mjesta uzimanja i ostavljanja. Zbog toga je definirana varijabla *Zos\_sigurna\_visina*. Prilikom pozivanja te varijable robotski manipulator se pomiče u visinu koju je ranije definirana, a koja je dovoljna da se izbjegnu sve prepreke s manipulatorom.

```

Function f_IzmjenaPalete

  If g_brojacPalete = 0 And kutija_prazna = 1 And kutija_prisutna = 1 Then

    Go Here :Z(Zos_sigurna_visina)
    Go pPraznaKutija :Z(Zos_sigurna_visina)
    On qOslobodiKutiju
    MemOn mOslobodiKutiju

    Go pPraznaKutija
    Call f_Vakuume_Ukljuci
    Wait Sw(iVAK_1_Drzi) = 1 Or Sw(iVAK_2_Drzi) = 1

    Speed BrzinaKutija
    Accel AkceleracijaKutija, AkceleracijaKutija

    Go pOslobodiKutiju :Z(Zos_sigurna_visina)
    On qIzbacioPraznu

    kutija_prisutna = 0

    On qNovuKutiju
    MemOn mNovuKutiju

    Go pOdlaganjeKutije :Z(Zos_sigurna_visina)
    Go pOdlaganjeKutije
    Call f_Vakuume_Iskljuci

    Speed BrzinaKutija_NakonPustanja
    Accel AkceleracijaKutija_NakonPustanja, AkceleracijaKutija_NakonPustanja

    Go pOdlaganjeKutije :Z(Zos_sigurna_visina)
    Go pOsnovni
    MemOn mUOsnovnoj
    On qUOsnovnom

  EndIf
Fend

```

Slika 4.41: Funkcija za izmjenu palete

#### 4.2.6 Funkcija za dozvolu okretnici

```

Function f_DozvolaOkretnici
  Do
    If MemSw(mDozvolaPremaSt02) = 1 Then
      On qDozvolaPremaSt02
    Else
      Off qDozvolaPremaSt02
    EndIf

    Wait 0.05 'kratko cekanje da procesor kontrolera ne vrti na 100% mogucnosti
  Loop
Fend

```

Slika 4.42: Funkcija za dozvolu okretnici

Ova funkcija služi za dozvolu rada stanice dva. Stanica jedan je mjesto uzimanja sa palete, dok je stanica dva mjesto odlaganja robotskog manipulatora. Robot ne može odlagati

ako je prethodni komad još uvijek u točki odlaganja ili ako je stanica dva u tijeku micanja odloženog komada.

U *DO* petlji napravljen je uvjet koji čita memorijski bit *mDozvolaPremaSt02*. Ako je taj memorijski bit aktivan uključuje se izlaz *qDozvolaPremaSt02*, koji se šalje na *PLC*. *PLC* dalje obrađuje taj signal te daje stanici dva dozvolu za rad.

#### 4.2.7 Funkcije za kretanje robota

```
Function f_IdiIznadOdlaganja           ' robot iznad mjesta odlaganja
    Go pOdlaganjeKucista :Z(Zos_sigurna_visina)
Fend
```

---

```
Function f_NaOdlaganje                 ' robot u poziciji odlaganja
    Go pOdlaganjeKucista
Fend
```

---

```
Function f_IdiIznadTockePalete         ' robot iznad pozicije uzimanja
    If paletal = 1 Or paletal_180 = 1 Then
        Go XY(x_robota, y_robota, z_robota, u_robota) :Z(Zos_sigurna_visina)
    EndIf
Fend
```

---

```
Function f_UTockuPalete                ' robot u poziciji uzimanja
    If paletal = 1 Or paletal_180 = 1 Then
        Go XY(x_robota, y_robota, z_robota, u_robota) '
    EndIf
Fend
```

Slika 4.43: Funkcije za kretanje robota

Funkcije za kretanje robota napravljene su radi jednostavnijeg razumijevanja programa. U prve dvije funkcije ide se iznad točke odlaganja, s time da se robot kreće na sigurnoj visini po z osi te ide u poziciju odlaganja komada. U druge dvije funkcije robot ide u točku iznad pozicije uzimanja komada iz kutije te u točku uzimanja komada iz kutije. U tim funkcijama koordinate pozicioniranja robota dobivaju se iz funkcije za pretragu slike s kamere *f\_Vision*

## 4.2.8 Funkcija za automatski rad

Funkcija za automatski rad potrebna je zbog toga što se želi bez vanjske interakcije izvoditi program sve dok su uvjeti za rad robota ostvareni. To se radi u *DO* petlji koja se neprekidno izvodi sve dok su uvjeti u njoj zadovoljeni.

```
Function f_AutomatskiRad
  Print "Automatski rad"
Do
  If MemSw(mUOsnovnoj) = 1 And MemSw(mIznadKutijeSKom) = 0 And kutija_prazna = 0 And kutija_prisutna = 1 And (paleta1 = 1
  Or paleta1_180 = 1) And g_brojacPalete >= 1 And g_brojacPalete <= 48 And MemSw(mIzmjenaUToku) = 0 Then

    Print "Vadenje kucista iz kutije "

    Wait Sw(iUzmiKucista) = 1
    Off qDozvolaPremaSt01
    MemOff mUOsnovnoj
    Off qUOsnovnom
    Speed BrzinaRadna
    Accel AkceleracijaRadna, AkceleracijaRadna
    Call f_IdiIznadTockePalete
    Call f_UTockuPalete
    Call f_Vakuume_Ukljuci
    Wait Sw(iVAK_1_Drzi) = 1 Or Sw(iVAK_2_Drzi) = 1
    Wait 0.05 'pricekaj kratko vrijeme da se ukljuci vakuum
    Speed BrzinaDizanja
    Accel AkceleracijaDizanja, AkceleracijaDizanja
    Go Here :Z(Zos_sigurna_visina)
    On qIznadKutijeSKom
    MemOn mIznadKutijeSKom
    On qDozvolaPremaSt01
    Wait Sw(iOdlaziKucista) = 1
    MemOff mDozvolaPremaSt02
    Speed BrzinaRadna
    Accel AkceleracijaRadna, AkceleracijaRadna
    Call f_IdiIznadOdlaganja
    Off qIznadKutijeSKom
    MemOff mIznadKutijeSKom
    Speed BrzinaDizanja
    Accel AkceleracijaDizanja, AkceleracijaDizanja
    Call f_NaOdlaganje
    Call f_Vakuume_Iskljuci
```

Slika 4.44: Funkcija za automatski rad, 1. dio

Uvjeti za početak automatskog rada mogu se vidjeti odmah nakon otvaranja *DO* petlje. Kada su svi uvjeti zadovoljeni počinje izvođenje programa. Pozivaju se funkcije za dolazak manipulatora iznad točke uzimanja kućišta, u točku uzimanja kućišta te funkciju za uključivanje vakuuma. Tim funkcijama uzima se kućište iz palete. Dalje se pozivaju funkcije koje dižu manipulator te ga šalju u poziciju iznad i u točku odlaganja.

```

Call f_KontrolaProzoraPretrageSlike
Speed BrzinaRadna
Accel AkceleracijaRadna, AkceleracijaRadna
Go Here :Z(Zos_sigurna_visina)
On qOdlozioKucista
MemOn mDozvolaPremaSt02

If g_brojacPalete >= 1 And g_brojacPalete <= 48 Then
    Call f_IdiIznadTockePalete
Else
    Go pPraznaKutija :Z(Zos_sigurna_visina)
    paleta1 = 0
    paleta1_180 = 0
    g_brojacPalete = 0
    kutija_prazna = 1
EndIf

Off qOdlozioKucista
MemOn mUOsnovnoj
On qUOsnovnom
EndIf

If (g_brojacPalete < 1 Or g_brojacPalete > 48) And MemSw(mIzmjenaUToku) = 0 Then
    MemOn mIzmjenaUToku
    Call f_IzmjenaPalete
    Wait MemSw(mIzmjenaUToku) = 0 And g_brojacPalete = 1
    Call f_vision
EndIf
Wait 0.05
Loop
Fend

```

Slika 4.45: Funkcija za automatski rad, 2. dio

Nakon što je manipulator u položaju odlaganja, tj. nije više iznad palete, poziva se funkcija za kontrolu pretrage slike. Slika se pretražuje kad je robotski manipulator izvan vidnog polja kamere.

Nakon što se isprazne sva kućišta, kojih je na paleti 48, poziva se funkcija za izmjenu palete. Kada se paleta promijeni poziva se funkcija *f\_Vision* za pretragu kućišta na slici te robotski manipulator počinje izvoditi *DO* petlju ispočetka.

#### 4.2.9 Funkcija za osnovni položaj

Funkcija za osnovni položaj potrebna je radi postavljanja robota u neku osnovnu poziciju. U programu se gase sve trenutno aktivne funkcije, izlazi s kontrolera i memorijski bitove te se dovodi robot u poziciju koju je definirana kao *pOsnovni*.

```

Function f_IdiUOsnovni

Do
  If Sw(iUOsnovni) = 1 Then
    Print "U osnovni polozej robota"

    MemOff mIznadKutijeSKom
    MemOff mUOsnovnoj
    MemOff mDrzi_1_VAK
    MemOff mDrzi_2_VAK
    MemOff mNovuKutiju
    MemOff mOslobodiKutiju
    MemOff mIzmjenaUToku
    MemOff mDozvolaPremaSt02

    Off qUOsnovnom
    Off qOdlozioKucista
    Off qIzbacioPraznu
    Off qIznadKutijeSKom
    Off qNovuKutiju
    Off qOslobodiKutiju
    Off qDozvolaPremaSt01
    Off qNijePronadeno

    Quit f_AutomatskiRad
    Quit f_vision
    Quit f_Vakuume_Iskljuci
    Quit f_Vakuume_Ukljuci
    Quit f_IdiIznadTockePalete
    Quit f_UTockuPalete
    Quit f_IzmjenaPalete
    Quit f_IdiIznadOdlaganja
    Quit f_NaOdlaganje

```

Slika 4.46: Funkcija za osnovni položaj, 1. dio



```

Speed BrzinaOsnovni
Accel AkceleracijaOsnovni, AkceleracijaOsnovni

Go Here :Z(Zos_sigurna_visina)
Go pOsnovni

MemOn mUOsnovnoj
On qUOsnovnom
On qDozvolaPremaSt01
MemOn mDozvolaPremaSt02

Call f_Vakuume_Iskljuci

Print "Provjeriti jesu li otpustena kucista pala u kutiju"
Print "i ako jesu izvaditi ih van"

Wait Sw(iUzmiKucista) = 1
Wait 2
Call f_vision

Xqt f_AutomatskiRad
EndIf

Wait 0.05
Loop

Fend

```

Slika 4.47: Funkcija za osnovni položaj, 2. dio

Nakon što je sve ugašeno i robotski manipulator doveden u točku osnovne pozicije pokreće se funkcija *f\_AutomatskiRad* naredbom *Xqt*.

#### 4.2.10 *Main* funkcija

*Main* funkcija glavna je funkcija koja se izvodi. U njoj se pozivaju sve ostale funkcije koje je potrebno uključiti.

```

Global Integer AkceleracijaRadna, AkceleracijaOsnovni, AkceleracijaKutija
Global Integer AkceleracijaKutija_NakonPustanja, AkceleracijaDizanja

Global Real Zos_sigurna_visina
Global Real Visina_kucista
Global Real ZosVisinaVadenja

Global Integer x0_slike, y0_slike, x_korak_slike, y_korak_slike
Global Integer x0_slike_180, y0_slike_180

Global Preserve Integer i, j
Global Preserve Integer i_180, j_180

Global Preserve Real x_ robo, y_ robo, z_ robo, u_ robo

```

Slika 4.48: Popis deklariranih varijabli

```

Function main

    Print "Inicijalizacija"

    Motor On
    Power High

    BrzinaRadna = 70           'brzina u %(max=100)
    AkceleracijaRadna = 80    'akceleracija u %(max=120)

    BrzinaDizanja = 50
    AkceleracijaDizanja = 60

    BrzinaOsnovni = 10
    AkceleracijaOsnovni = 10

    BrzinaKutija = 10         'uzima praznu kutiju
    AkceleracijaKutija = 10

    BrzinaKutija_NakonPustanja = 30
    AkceleracijaKutija_NakonPustanja = 30

    ZosVisinaVadenja = -128   ' visina uzimanja
    Zos_sigurna_visina = -1

    pOsnovni = pTockaZaSnimanje :Z(Zos_sigurna_visina)  ':z apsolutna vrijednost

    pOslobodiKutiju = pPraznaKutija -X(4) -Y(5)

    Xqt f_PrazniOdPocetka
    Xqt f_PracenjePalete
    Xqt f_DozvolaOkretnici
    Xqt f_IdiUOsnovni

Fend

```

Slika 4.49: Main funkcija

### 4.3 Analiza rada robotskog manipulatora

Rad robotskog manipulatora prikazan je simulacijom. U simulaciji nije korištena navigacija pomoću kamere, budući da program *Epson RC+* ne podržava tu opciju. U simulaciji je moguće dodati kameru te u realnom vremenu gledati izvođenje programa robotskog manipulatora. Iako u simulaciji nije moguće navigirati robot pomoću kamere, svejedno se može simulirati okruženje te rad robota. Za to je potrebno naučiti točke uzimanja i odlaganja komada. Nakon što su točke u simulaciji naučene možemo vidjeti putanju robota koja će biti identična kao u stvarnom sustavu. Jedina bitna razlika između stvarnog i simulacijskog sustava je nemogućnost korištenja navigacije u simulaciji, dok su sve ostale funkcije poprilično jednake.

## 5. ZAKLJUČAK

U izradi proizvoda ugrađen je robotski manipulator koji je trebao ubrzati proizvodnju te ju učiniti pouzdanijom. Nakon ugradnje robotskog manipulatora primijećen je problem, odnosno nedostatak u radu robota. Kućišta koja robot vadi iz palete nisu uvijek u istoj poziciji, zbog toga što je mjesto iz kojeg robotski manipulator vadi kućišta obična kartonska kutija. Da se izbjegnu mogući problemi prilikom vađenja kućišta iz kutije, ugrađuje se kamera pomoću koje se robotski manipulator postavlja u točnu poziciju uzimanja kućišta. Nakon što je kamera instalirana napravljen je osnovni program koji je povezao kameru s robotskim manipulatorom. Taj program bio je vrlo jednostavan, ali s podosta mana. Kroz neko vrijeme program je doraden te su ubačene još neke funkcije koje su doprinijele boljem i pouzdanijem radu robota.

Ugradnjom robotskog manipulatora u potpunosti je izbačeno jedno radno mjesto. Budući da se radi o proizvodnji koja radi bez prestanka, nedvojbeno je da će se ova investicija ubrzo isplatiti.

## 6. LITERATURA

- [1] *Epson. Epson SCARA LS6-B702S/RC-90B*. 2017. Dostupno na: <https://www.epson.eu/products/robots/scara-ls-series/epson-scara-ls6-b702s-rc-90b/p/28435> (13.5.2022)
- [2] *Epson. SCARA ROBOT LS3-B/LS6-B series MANIPULATOR MANUAL*. 2022. Dostupno na: [https://files.support.epson.com/far/docs/epson\\_ls3-b\\_&\\_ls6-b\\_robot\\_manual\\_\(r7\).pdf](https://files.support.epson.com/far/docs/epson_ls3-b_&_ls6-b_robot_manual_(r7).pdf) (13.5.2022)
- [3] *Epson. ROBOT CONTROLLER RC90 / RC90-B*. 2022. Dostupno na: [https://files.support.epson.com/far/docs/epson\\_rc90\\_controller\\_manual\\_\(use\\_with\\_epson\\_rc\\_pl\\_7.0\)\\_\\_\(r20\).pdf](https://files.support.epson.com/far/docs/epson_rc90_controller_manual_(use_with_epson_rc_pl_7.0)__(r20).pdf) (27.6.2022)
- [4] *Epson. EPSON RC+ 7.0 Option Vision Guide 7.0*. 2022. Dostupno na: [https://files.support.epson.com/far/docs/epson\\_vision\\_guide\\_hardware-cv1\\_cv2\\_pv1\(v73r3\).pdf](https://files.support.epson.com/far/docs/epson_vision_guide_hardware-cv1_cv2_pv1(v73r3).pdf) (13.5.2022)
- [5] *Vex Robotics. Industrial Robotics*. 2022. Dostupno na: <https://education.vex.com/stemlabs/workcell/stemlab/industrial-robotics/what-are-industrial-robots?lng=en> (13.5.2022)
- [6] *MHI. Industrial Robots*. 2022. Dostupno na: <https://www.mhi.org/fundamentals/robots> (13.5.2022)
- [7] *Techtarget. Machine vision*. 2022. Dostupno na: <https://www.techtarget.com/searchenterpriseai/definition/machine-vision-computer-vision> (13.5.2022)
- [8] Łabudzki R. *Stanje razvoja strojnog vida*. Stručni rad. Poznanj: Tehnološko sveučilište u Poznanju; 2011. Dostupno na: <https://hrcak.srce.hr/file/127735> (13.5.2022)
- [9] *Keyence. History of Machine Vision : Factory Automation (FA)*. 2022. Dostupno na: <https://www.keyence.com/ss/products/vision/visionbasics/history/fa.jsp> (13.5.2022)
- [10] *Qualitas Technologies. The Evolution of Machine Vision : Factory Automation (FA)*. 2022. Dostupno na: <https://qualitastech.com/blog/quality-control-insights/evolution-of-machine-vision/> (13.5.2022)
- [11] *Epson. Vision Guide 7.0 Hardware & Setup*. 2022. Dostupno na: [https://files.support.epson.com/far/docs/epson\\_ls3-b\\_&\\_ls6-b\\_robot\\_manual\\_\(r7\).pdf](https://files.support.epson.com/far/docs/epson_ls3-b_&_ls6-b_robot_manual_(r7).pdf) (13.5.2022)

[12] *Basler. Lens Selector*. 2022. Dostupno na: <https://www.baslerweb.com/en/sales-support/tools/lens-selector/#option-camera;series=Basler%20ace;model=acA640-300gc>  
(13.5.2022)

## **7. OZNAKE I KRATICE**

*I/O – Input/Output (ulazi/izlazi)*

*PLC – Programmable Logic Controller (programabilni logički kontroler)*

*USB – Universal Serial Bus (univerzalni serijski ulaz)*

*SCARA - Selective Compliance Assembly Robot Arm*

*CPU - Central Processing Unit*

*GigE - Gigabit Ethernet*

## 8. SAŽETAK

**Naslov:** NAVIGACIJA ROBOTA POMOĆU KAMERE

U cilju poboljšanja kvalitete rada robota, ugrađuje se kamera za bolje pozicioniranje robotskog manipulatora. U ovom radu opisan je robotski manipulator, njegove komponente, te industrijske kamere. Također je opisana implementacija industrijske kamere u sustav rada robota. Pomoću programa *Epson RC+* usklađen je rad robotskog manipulatora s industrijskom kamerom.

**Ključne riječi:** robotika, industrija, kamera, *Epson*.



## 9. ABSTRACT

**Title:** Navigation of robot with help of camera

In order to improve the quality of robot work, camera is installed for better positioning of the robot manipulator. This seminar describes a robotic manipulator, its components, and industrial cameras. The implementation of an industrial camera in a robot operation system is also described. With the help of the *Epson RC +* program, the operation of the robotic manipulator is coordinated with the industrial camera.

**Keywords:** robotics, industry, camera, *Epson*.

## IZJAVA O AUTORSTVU ZAVRŠNOG RADA

Pod punom odgovornošću izjavljujem da sam ovaj rad izradio/la samostalno, poštujući načela akademske čestitosti, pravila struke te pravila i norme standardnog hrvatskog jezika. Rad je moje autorsko djelo i svi su preuzeti citati i parafraze u njemu primjereno označeni.

Mjesto i datum	Ime i prezime studenta/ice	Potpis studenta/ice
U Bjelovaru, <u>15. 7. 2022</u>	HRVOJE ŠVIGLIN	HRVOJE ŠVIGLIN

Prema Odluci Veleučilišta u Bjelovaru, a u skladu sa Zakonom o znanstvenoj djelatnosti i visokom obrazovanju, elektroničke inačice završnih radova studenata Veleučilišta u Bjelovaru bit će pohranjene i javno dostupne u internetskoj bazi Nacionalne i sveučilišne knjižnice u Zagrebu. Ukoliko ste suglasni da tekst Vašeg završnog rada u cijelosti bude javno objavljen, molimo Vas da to potvrdite potpisom.

Suglasnost za objavljivanje elektroničke inačice završnog rada u javno dostupnom nacionalnom repozitoriju

HRVOJE ŠVIGLIN

*ime i prezime studenta/ice*

Dajem suglasnost da se radi promicanja otvorenog i slobodnog pristupa znanju i informacijama cjeloviti tekst mojeg završnog rada pohrani u repozitorij Nacionalne i sveučilišne knjižnice u Zagrebu i time učini javno dostupnim.

Svojim potpisom potvrđujem istovjetnost tiskane i elektroničke inačice završnog rada.

U Bjelovaru, 15. 7. 2022

Hrvoje Švigin  
*potpis studenta/ice*