

# Čitanje QR koda pomoću kamere robotskog sustava

---

**Babić, Tomislav**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University North / Sveučilište Sjever**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:122:607263>

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

*Download date / Datum preuzimanja:* **2024-11-27**



*Repository / Repozitorij:*

[University North Digital Repository](#)





**Sveučilište  
Sjever**

**Završni rad br. 038/MEH/2024**

**Čitanje QR koda pomoću kamere robotskog sustava**

**Tomislav Babić, 0336039917**

Varaždin, rujan 2024. godine





# Sveučilište Sjever

Odjel za Mehatroniku

Završni rad br. 038/MEH/2024

## Čitanje QR koda pomoću robotskog sustava

### Student

Tomislav Babić, 0336039917

### Mentor

Zoran Busija, dipl. ing. strojarstva

Varaždin, rujan 2024. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODIEL: Odjel za mehatroniku

STUDIJ: preddiplomski stručni studij Mehatronika

PRISTUPNIK: Tomislav Babić

JMBAG: 0336039917

DATUM: 09.09.2024.

KOLEBIJ: Robotika

NASLOV RADA: Čitanje QR koda pomoću kamere robotskog sustava

NASLOV RADA NA ENGL. JEZIKU: QR code reading using the camera of the robot system

MENTOR: Zoran Busija, dipl. ing. stroj.

ZVANJE: predavač

ČLANOVI POVJERENSTVA

1. Siniša Švoger, mag.ing.mech, predavač
2. prof. dr. sc. Ante Čikić
3. Zoran Busija, dipl.ing.stroj, predavač
4. Josip Srpak, dipl.ing.el, viši predavač
- 5.

## Zadatak završnog rada

BRJ: 038/MEH/2024

OPIS:

U završnom radu potrebno je:

- opisati robotski sustav s kamerom.
- posebno istražiti i opisati mogućnosti kamere.
- potrebne objekte oblikovati pomoću 3D CAD alata.
- izgraditi robotsko okruženje za simulaciju sustava pomoću softvera RobotStudio.
- objasniti pojam QR kod i podesiti (naučiti) kameru za prepoznavanje koda.
- izraditi primjer programa koji na temelju različitih kodova izvršava određene akcije.
- testirati program unutar softvera RobotStudio.

ZADATAK UBUČEN

16.09.2024



Busija Zoran

## **Predgovor**

Temu završnoga rada sam odabrao zbog toga što sam kroz stručnu praksu imao priliku vidjeti probleme sortiranja unutar skladišta jednog poduzeća. U moderno doba sve se više koristi dostava paketa, materijala, odjeće, obuće. Razmišljao sam kako riješiti problem kada stigne velika količina paketa u skladište, kako je najlakše i najbrže sortirati. Ideja je bila da bi se problem mogao najlakše riješiti pomoću kamere robotskog sustava. Na temelju tog razmišljanja i iskustva odabrao sam ovu temu za svoj završni rad.

Posebne zahvala mentoru Zoranu Busiji, dipl. ing. stroj. na njegovom strpljenju, savjetima kroz izradu ovog rada. Velike zahvale kolegama koji su također bili spremni za suradnju te savjete.

Posebno se želim zahvaliti svojoj obitelji te bliskim prijateljima koji su bili moja najveća podrška te motivacija tijekom mojeg preddiplomskog studija.

Tomislav Babić

## Sažetak

Rad prikazuje implementaciju vizijskog sustava u radnu okolinu industrijskog robota. Kroz rad se upoznaje način rada i povezivanje ABB robota sa industrijskom kamerom. To se izvodi kroz sučelje softvera "RobotStudio-a", te "*In-sight*" programskog sučelja koje služi za uparivanje Cognex kamere sa upravljačkom jedinicom "*IRC5 Compact*". Također se pokazuje kako se oblikuje računalni 3D model objekta koristeći softver "SolidWorks". Nakon postavljanja radnog okruženja unutar "RobotStudio-a" pokazano je kako se upravlja virtualnim signalima unutar programa, a oni zapravo simuliraju signale realnog sustava.

Ključne riječi: IRB 120, IRC5 Compact, Cognex 7402, SolidWorks, Signali, "Smart Component"

## **Summary**

The topic of the paper is the implementation of a vision system in the working environment of an industrial robot. Through the work, the method of operation and connection of the ABB robot with the industrial camera is introduced. This is done through the "RobotStudio" software interface, and the "In-sight" program interface, which serves to pair the Cognex camera with the "IRC5 Compact" control unit. It also shows how to create a computer 3D model of an object using the "SolidWorks" software. After setting up the working environment within "RobotStudio", it was shown how to manage virtual signals within the program itself, and they actually simulate the signals of a real system.

Keywords: IRB 120, IRC5 Compact, Cognex 7402, SolidWorks, Signals, "Smart Component"



## **Popis korištenih kratica**

ABB	ASEA Brown Boveri
IRB	Industrial Robot
IRC	Industrial Robot Controller
TCP	Tool Centar Point
QR	Quick Response kod
ON	Uključeno
OFF	Isključeno
CAD	Computer Aided Design

# Sadržaj

1.	Uvod.....	1
2.	Upoznavanje elemenata sustava .....	2
2.1.	Robot i upravljačka jedinica.....	2
2.2.	Kamera i njena svojstva .....	4
2.3.	QR kod .....	5
3.	Oblikovanje dijelova u SolidWorks-u .....	6
4.	Programski alat RobotStudio za simulaciju sustava .....	11
4.1.	Postavljanje robota i njegovog okruženja .....	11
4.2.	Kreiranje putanje robota.....	17
4.3.	In-Sight Eyplorer programsko sučelje.....	20
5.	Opis objekta „SmartComponent“ .....	27
6.	Dopremanje paketa na kliznu stazu .....	28
7.	Zaključak.....	40
8.	Literatura.....	41
	Popis slika .....	42

## 1. Uvod

U današnje vrijeme industrija se sve više automatizira pomoću robotskih postrojenja zbog velikih količina proizvoda koje tržište zahtijeva. Čovjeka se sve više miče iz same proizvodnje te ga se nastoji zaštititi od zamornih, ponavljajućih poslova. To se i postiže upravo implementacijom robota u sustave proizvodnje. Robot se u proizvodnji ne umara, ne radi greške, a može raditi 24 sata na dan dok čovjek to ne može. Naravno, čovjek je još uvijek potreban za nadzor takvih sustava. Upravo time se postiže veća efikasnost te učinkovitost same tvornice, proizvodnje te skladišta, ovisno o tome gdje će se sustav implementirati. Zbog širokog spektra robota te njihove opreme, danas ih je moguće implementirati gotovo u svaki sustav. Može zamijeniti čovjeka gotovo u svakom pogledu. U autoindustriji kod sastavljanja, dopreme dijelova, vizualne inspekcije i slično. Može se desiti da zbog umora čovjeka ili samim time što nije uočio neku grešku, proizvod ode na tržište neispravan ili oštećen. Ako su parametri robota te njegove dodatne opreme kao što su kamere, pištolj za zavarivanje, te razni prihvatni dobro podešeni, mogućnosti pogrešaka svedene su na minimum. Naravno, uvjet za takav rad robota je redovno održavanje sustava. Ako se takav sustav ne održava, to jest ne servisira, nakon nekog vremena će se pojaviti greške. Zato je jako važno voditi dobro nadziranje ovakvog sustava.

Elementi koji se koriste kroz ovaj rad su: industrijski robot ABB IRB 120, upravljačka jedinica ABB IRC5 Compact, ABB Smart gripper te Cognex 7402 kamera. Pomoću stolnog računala se programira te upravlja sustavom robota i sustavom kamere. Kroz rad su detaljno opisani pojedini elementi sustava te je pokazano kako su oni međusobno povezani.

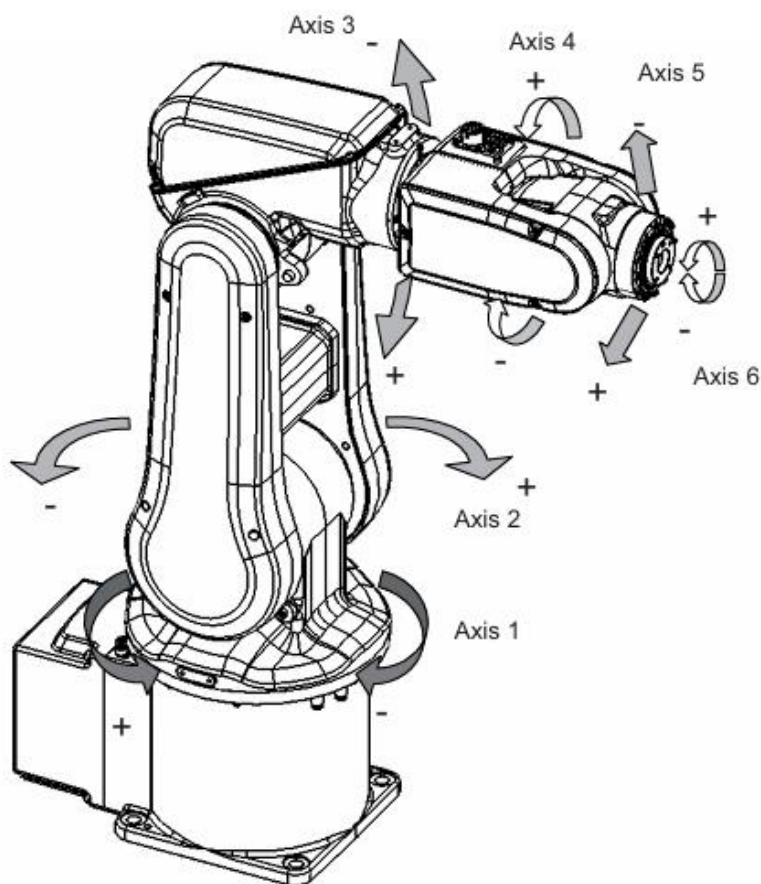
Najprije se opisuje sam robot te neka njegova ograničenja. Zatim njegova upravljačka jedinica IRC 5 Compact, te na kraju kamera Cognex 7402. Za svaki pojedini dio su navedene njegove osnovne specifikacije. Softver za gibanje robota je rađen u RobotS-studiju. Zaseban softver *In-Sight explorer* služi kao virtualna kamera.

## 2. Upoznavanje elemenata sustava

Sustav se sastoji od više pojedinih elemenata pa je opis zbog toga podijeljen u više cjelina. Prva cjelina opisuje robot i njegove značajke kao što su: doseg, nosivost, masa robota, potrošnja i slično. Druga cjelina opisuje kameru i njezine značajke. Treća cjelina opisuje izradu simulacije rada cijelog sustava.

### 2.1. Robot i upravljačka jedinica

Puni naziv robota koji se koristi je IRB 120 3-58. Proizvođač je ABB (ASEA Brown Boveri). U imenu robota su ujedno i njegove specifikacije. U ovom slučaju: IRB 120 „Industrial Robot 120“, sljedeći broj je „3“-taj broj označava propisanu nosivost od strane proizvođača u kilogramima, a broj „58“ njegovu maksimalnu horizontalnu udaljenost od središta robota, što bi u ovom slučaju bilo 0.58 metara. Robot ima 6 stupnjeva slobode gibanja koji su prikazani na slici 1. Svaki stupanj slobode je rotacijski, te svaki stupanj pogoni po jedan elektro motor na izmjeničnu struju. Zbog toga što su svi (više od 3 ) stupnjevi slobode gibanja rotacijski, robot je smješten u kategoriju zglobnih ili artikuliranih robota.[1][2]



Slika 2.1 Prikaz robota sa označenim osima (stupnjevi slobode gibanja).[1]

IRB 120 je jedan on prvih industrijskih robota, tako male nosivosti, koje je ABB proizveo. Zbog male nosivosti predviđen je za preciznu montažu sitnih komponenti, te također za podizanje i premještanje predmeta (engl. *Pick and place*). Funkcija premještanja predmeta predviđena je na primjer za farmaceutsku industriju pri manipuliranju lijekovima. Robot zadovoljava standard DIN EN ISO 14644-1 koji određuje klasu čistoće zraka u smislu broja čestica izraženih kao koncentracija u volumenu zraka. Zbog toga se može upotrijebiti u farmaceutskoj, prehrambenoj i u drugim industrijama gdje se traži velika čistoća.[2]

IRB 120 specifikacija	Podatak
Nosivost	3 kg
Masa robota	25 kg
Doseg robota	0.58 m
Energetska potrošnja (ISO kocka, maksimalna brzina )	0.24 kW
Energetska potrošnja-kočnice ON	0.095 kW
Energetska potrošnja-kočnice OFF	0.173 kW

Tablica 1 – Osnovni tehnički podatci robota IRB 120 3-58

IRB 120 dolazi u kompletu sa upravljačkom jedinicom IRC 5 (*Industrial Robot Controller*, slika 2.3). Može se reći da je to mozak ili računalo robota. Unutar njega se nalaze komponente kao i u računalu: procesor, radna memorija, memorija za pohranu podataka, ali za razliku od klasičnog računala ima i napajanje elektro motora robota te dodatne opreme robota. Robot smije spajati samo tehničar koji je ovlašten od strane proizvođača. Njegov zadatak je postaviti robot na željenu poziciju kupca. Također, potrebno je povezati robot s upravljačkom jedinicom odgovarajućim kablovima. Svu dodatnu opremu također je potrebno povezati. U dodatnu opremu spadaju senzori, kamere te *FlexPedant*.



Slika 2.2 IRC 5 upravljačka jedinica

## 2.2. Kamera i njena svojstva

Kamera koja se koristi za ovu simulaciju je Cognex 7402. U tablici 2. se nalaze njezine tehničke informacije. Informacija koja je bitna za ovu simulaciju je rezolucija slike koju kamera ima. Rezolucija slike je 1280 x 1020 piksela.

Podatak	Vrijednost
Programska memorija	512 MB
Radna memorija	256 MB
Rezolucija slike	1280 x 1020
Brzina zatvarača	Od 16 $\mu$ s do 950 ms
Potrošnja energije	24V, 2A
Kućište	Aluminij
Masa	220g
Radna temperatura	Od 0°C do 45°C
Spremišna temperatura	Od -30°C do 80°C
Maksimalna sila udarca	80G
Maksimalna sila vibracija	10G od 10Hz do 500Hz

Tablica 2 – Tehnički podaci kamere Cognex 7402 [4]

Zbog svojih karakteristika, kamera se koristi u različitim sektorima industrije. Primjena ove kamere kreće se od optimizacije logističkih operacija preko sortiranja paketa pa sve do inspekcije pojedinih komponenti unutar neke proizvodnje kao što je inspekcija komponenti u autoindustriji. Raznolikost primjene ove kamere je u njenoj mogućnosti čitanja raznih kodova kao što su barkod, poštanski kod, QR kod, uglavnom svih kodova u 1 i 2 dimenzije. U ovoj simulaciji koristi se mogućnost usporedbe slike, to jest QR koda. Kameri se zadaju slike QR kodova. Nakon očitavanja odgovarajuće slike, kamera šalje signal upravljačkoj jedinici robota. Svaki signal predstavlja određeni zadatak. Na primjer, kada kamera očita QR kod „zagreb\_qr“, šalje se signal upravljačkoj jedinici robota. Kada signal dođe u upravljačku jedinicu, robot izvršava određeni zadatak. Zadatak je sljedeći: nakon očitavanja te dolaska signala u upravljačku jedinicu, robot se kreće zadanom putanjom prema prostoru za odlaganje. Nakon što robot dođe u određenu poziciju, ostavlja objekt na zadanoj poziciji te se vraća u „Home position“, to jest početnu točku.

### 2.3. QR kod

QR (Quick response kod) je vrsta dvodimenzionalnog koda koji se koristi za pohranu informacija. Popularnost QR koda se znatno povećala posljednjih godina zbog njegove efikasnosti i jednostavnosti. Za njegovo korištenje je potreban uređaj s kamerom te softverom za čitanje QR koda. Unutar koda mogu biti zapisane razne informacije. Može služiti za plaćanje, marketing, zdravstvo, ulaznice i razne druge opcije.

QR kod se sastoji od crnih i bijelih kvadratića koji su raspoređeni u određenom uzorku. Ti kvadratići predstavljaju binarne podatke. Svaka boja označava neku informaciju. Crni kvadratići označavaju binarno 1, a bijeli kvadratići binarno 0. Nakon što se QR kod formira te ga se želi pročitati, potrebno ga je dekodirati. Softver za čitanje QR kodova analizira raspored pojedinih kockica, to jest 0 i 1 binarnog koda. Ovi podaci se zatim interpretiraju prema kodiranom formatu na primjer: tekst, broj telefona, link poveznica i slično.

QR kodovi su generirani pomoću online softvera. Način na koji se kreiraju je sljedeći: u web tražilicu je potrebno upisati sljedeću web adresu „[www.qr.io](http://www.qr.io)“, zatim se otvara izbornik za kreiranje QR kod po želji korisnika. QR kodovi za ovu simulaciju su zapravo na službe stranice Sveučilišta Sjever, Grada Zagreba, Marije Bistrice (paket sa greškom). Nakon što su kodovi generirani, potrebno ih je spremiti u obliku .JPG, a zatim ih premjestiti u željenu mapu na disku iz koje će se kasnije tijekom rada pozivati u SolidWorks softver.

### 3. Oblikovanje dijelova u SolidWorks-u

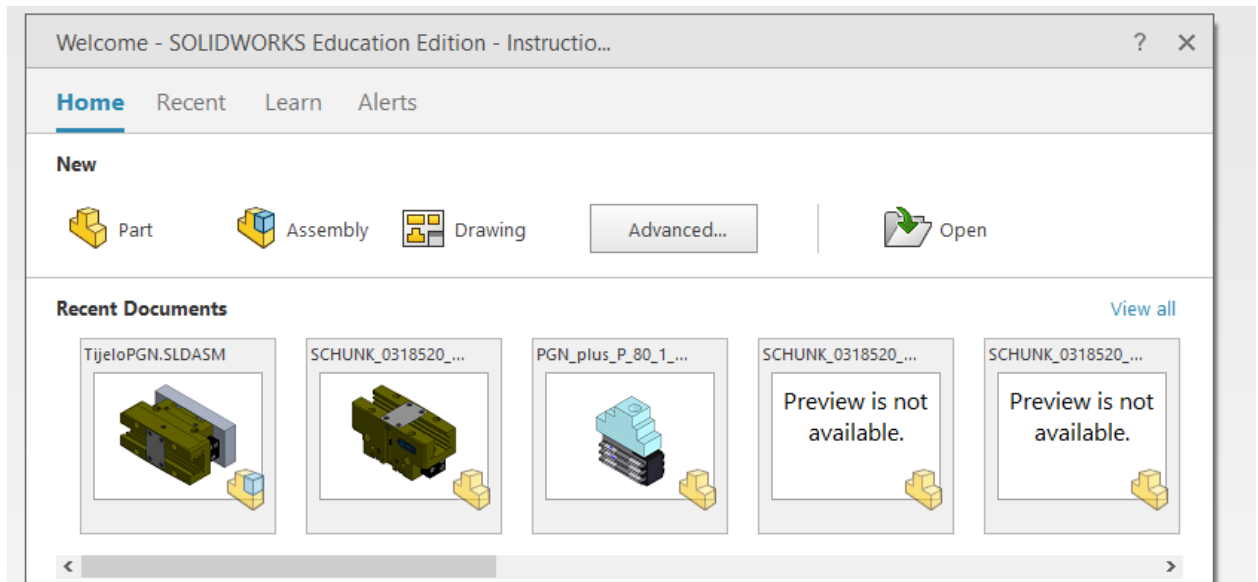
Kako bi se realizirala simulacija sortiranja paketa pomoću industrijskog robota, potrebno je oblikovati pojedine komponente. Za oblikovanje komponenti koristi se CAD programski sustav SolidWorks koji omogućuje 3D oblikovanje.

U programskom sustavu SolidWorks oblikovane su sljedeće komponente :

- Paket Sjever
- Paket Zagreb
- Paket greška
- Staza
- Osigurač
- Klizna staza
- Profili držača kamere te njihov sklop

Komponente u programskom sustavu SolidWorks oblikovane su na sljedeći način:

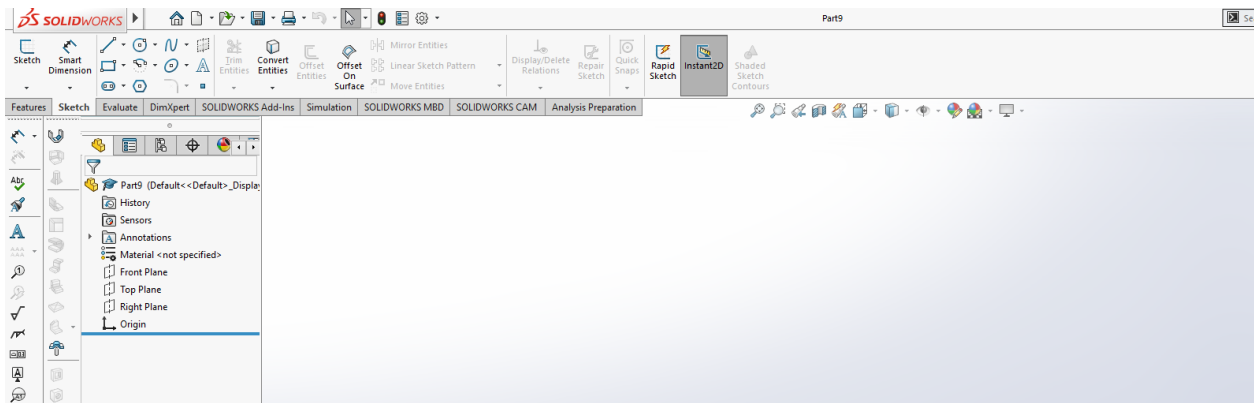
Pokretanjem SolidWorks programa pojavljuje se početni izbornik u kojem se odabire što se želi oblikovati. Opcije oblikovanja koje se nude su: Part, Assembly i Drawing. Obzirom da se objekt sastoji od samo jednog dijela, odabire se opcija Part (slika 3.1).



Slika 3.1. Kreiranje novog dijela (Part)

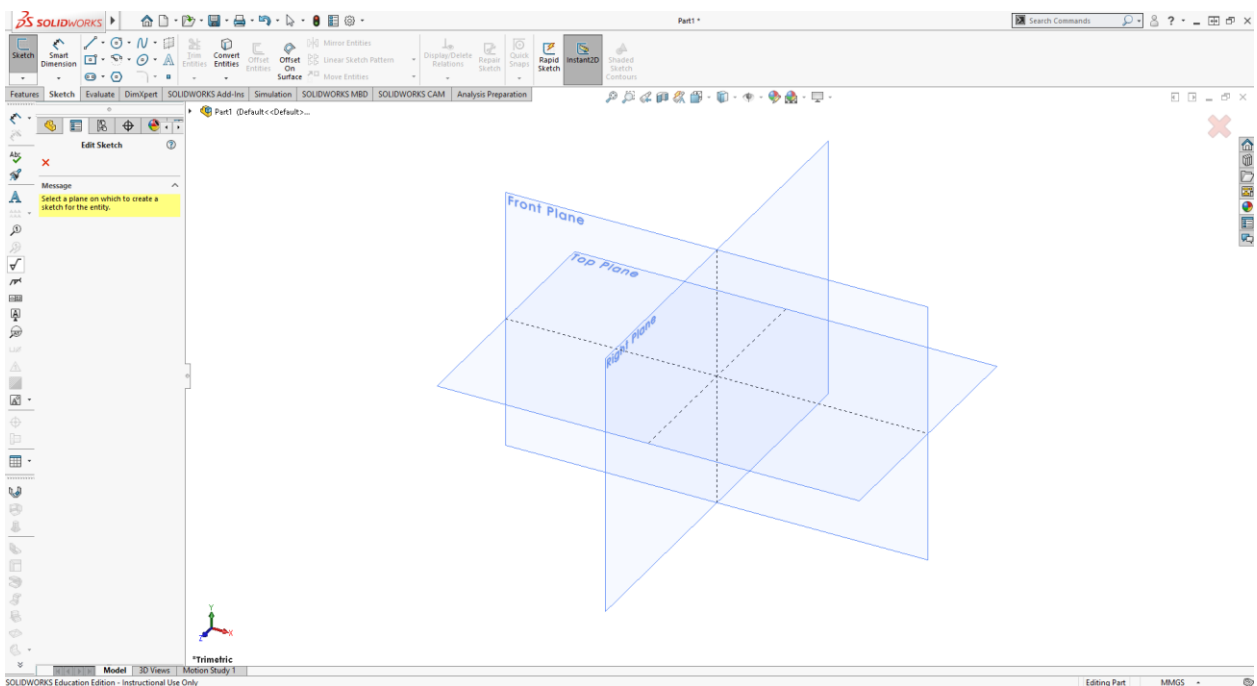
Nakon odabira opcije *Part* otvara se radna površina SolidWorks-a. Oblikovanje komponente započne se klikom na karticu "Sketch" i zatim na opciju "Sketch" iznad nje (Slika 3.2).





Slika 3.2 Odabir kartice „Sketch“ i odabir alata „Sketch“ u gornjem lijevom kutu

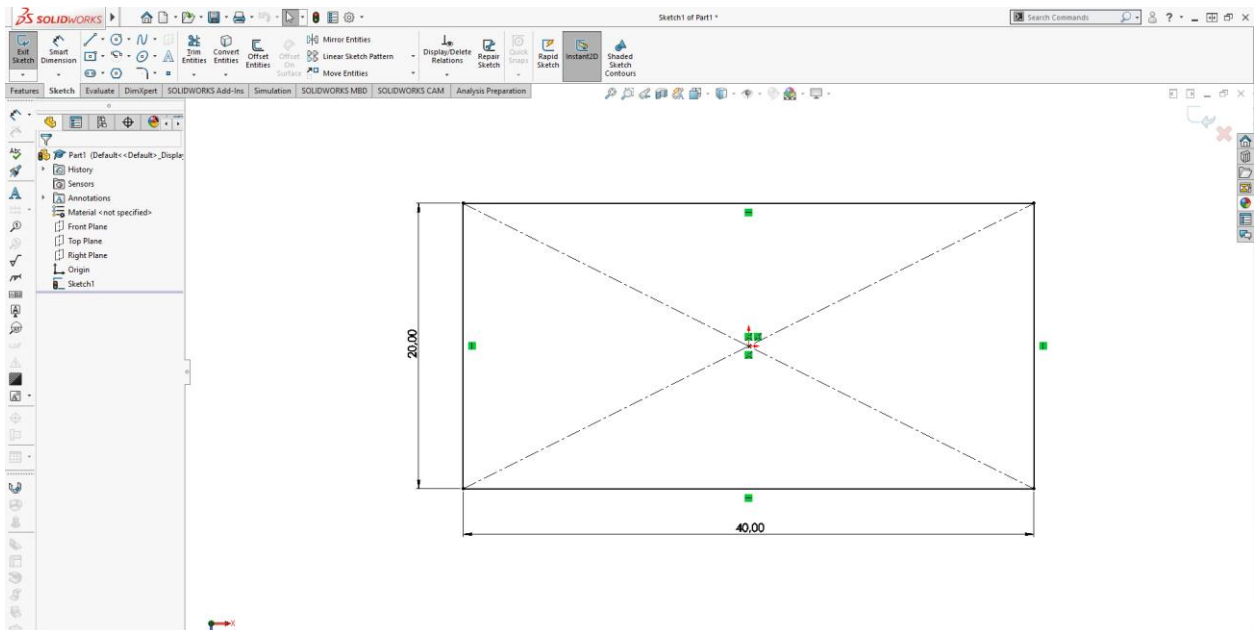
Nakon odabira opcije “Sketch” na radnoj površini se prikažu 3 ravnine (engl. *Front, Top; Right Plane*) kao što prikazuje slika 3.3. U nastavku se odabire ravnina na kojoj će se konstruirati skica. U ovom slučaju odabire se “Top plane”.



Slika 3.3 Odabir ravnine za kreiranje skice „Top Plane“

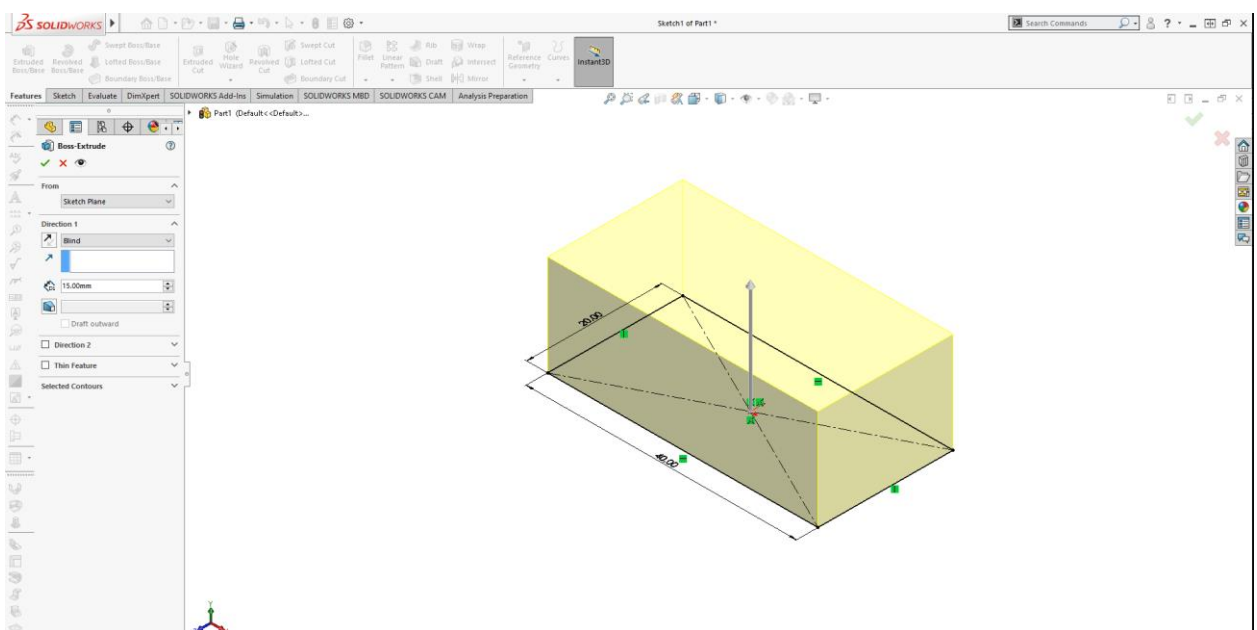
Skica započinje odabirom opcije “Centar Rectangle”(slika 3.4.) na kartici alata “Sketch”. Odabirom te opcije aktivira se alat za skiciranje pravokutnika. Zatim se odabire centar, ishodište pravokutnika. Poželjno je da se definira ishodište pravokutnika u centru koordinatnog sustava kako bi se lakše definirali parametri linija pravokutnika. Nakon što je pravokutnik skiciran, odabire se opcija “Smart dimension”(slika 3.4.). Alatom “Smart dimension” te zatim odabirom stranice pravokutnika definira se duljina pojedine stranice te se upisuju željene dimenzije. Dimenzije ovog

pravokutnika su visina od 20mm te širina 40mm. Kada je skica potpuno definirana, ona postaje crna, a prije potpunog definiranja je plave boje.



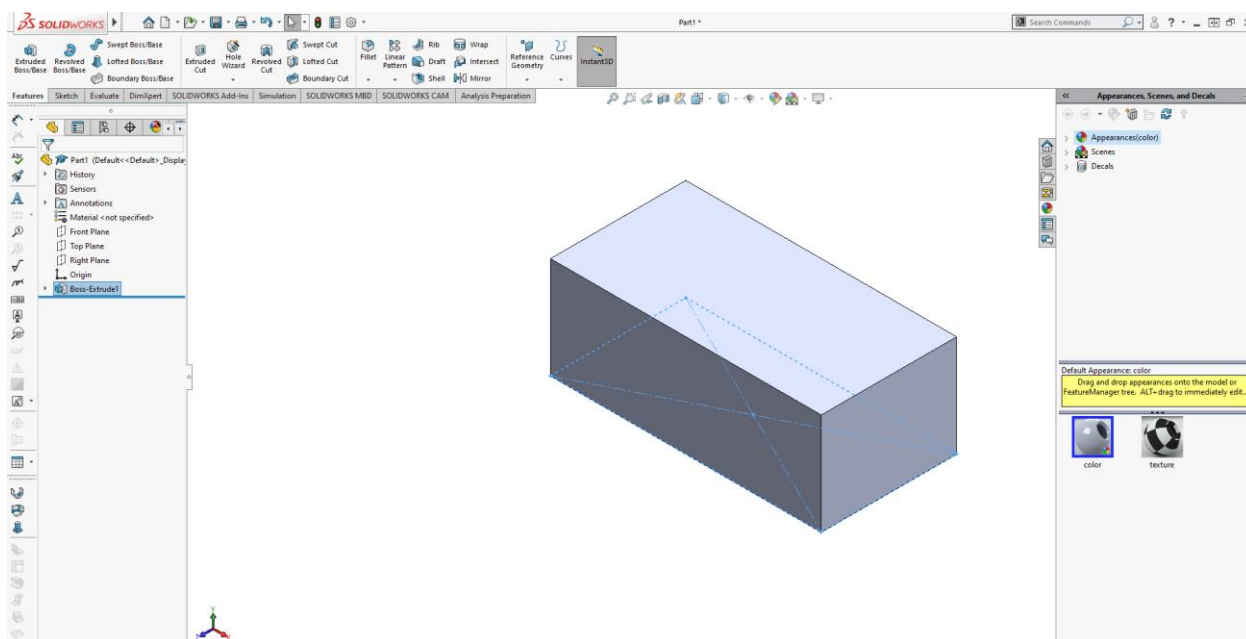
Slika 3.4 Odabir opcije za skiciranje pravokutnika te njegovo definiranje

Nakon potpunog definiranja pravokutnika potrebno je odabrati “Features” na kartici alata. Zatim klikom na opciju “Extruded Boss/Base”(slika 3.5.) može se zadati treću dimenziju kvadratu i stvoriti 3D model. Odabiremo visinu kvadru 15mm te smjer u koju stranu će se on formirati (slika 3.5).



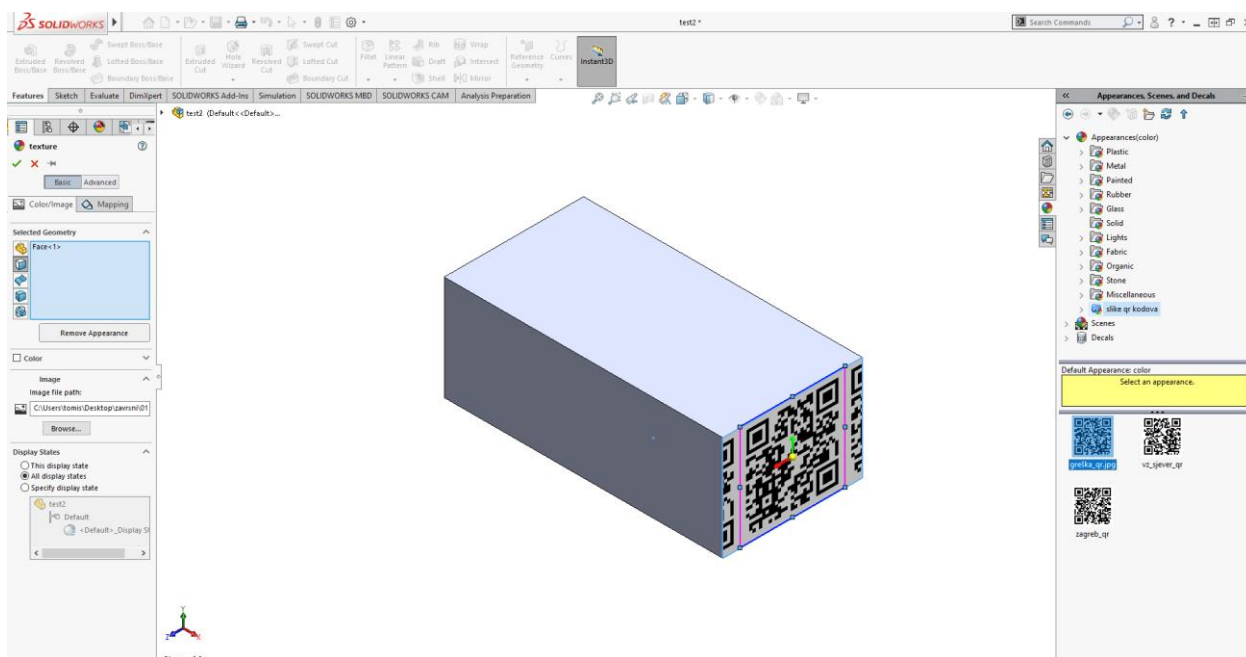
Slika 3.5 Odabir opcije „Extruded Boss/Base“ za kreiranje tijela pravokutnika

Nakon što je kvadar formiran odaberemo opciju “appearance” (slika 3.6). Zatim se odabire/dodaje lokacija iz koje će se uzimati uzorci. Nakon toga se pojave uzorci koji se mogu zalijepiti na sam objekt. Odabire se ili cijeli objekt ili pojedina ploha bjekta, ovisno o tome što se želi urediti.



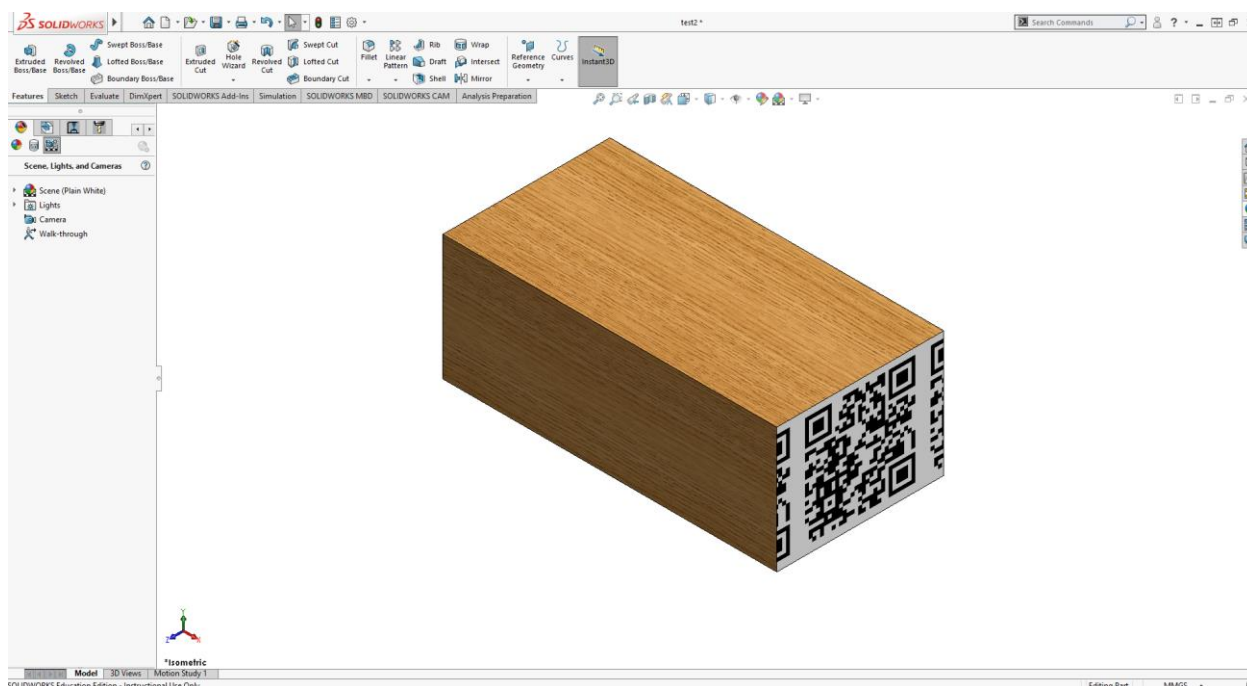
Slika 3.6 Odabir opcije „appearance“ te dodavanje izvora za uzorke

U ovom slučaju odabire se bočna stranica te se na nju zalijepi uzorak po želji iz izbornika (slika 3.7).



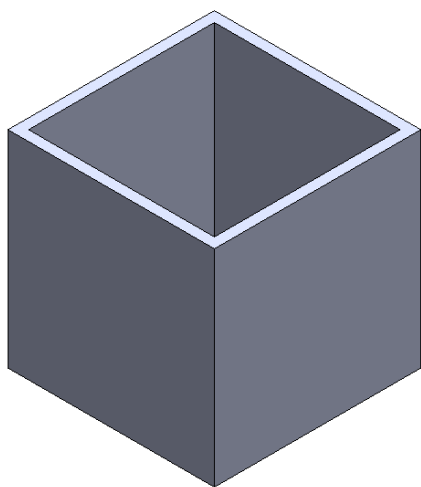
Slika 3.7 Odabir plohe tijela koje se želi implementirati - slika iz izbornika

Zatim se definira završni izgled objekta koji nam simulira kasnije paket u simulaciji unutar “RobotStudio-a” (slika 3.8).

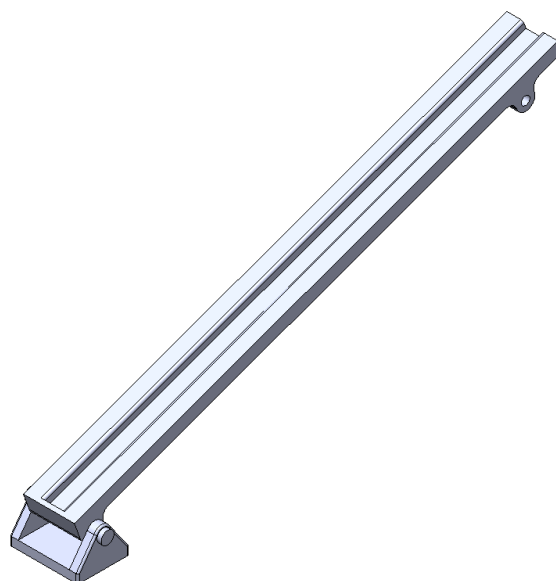


*Slika 3.8 Objekt „Paket s greškom“*

Istim postupkom se izrađuju i ostali objekti koji su potrebni za ovu simulaciju. Potrebno je izraditi spremnik za odlaganje paketa (slika 3.9) i kliznu stazu (slika 3.10) po kojoj se paketi dostavljaju.



*Slika 3.9 Spremnik za odlaganje paketa*



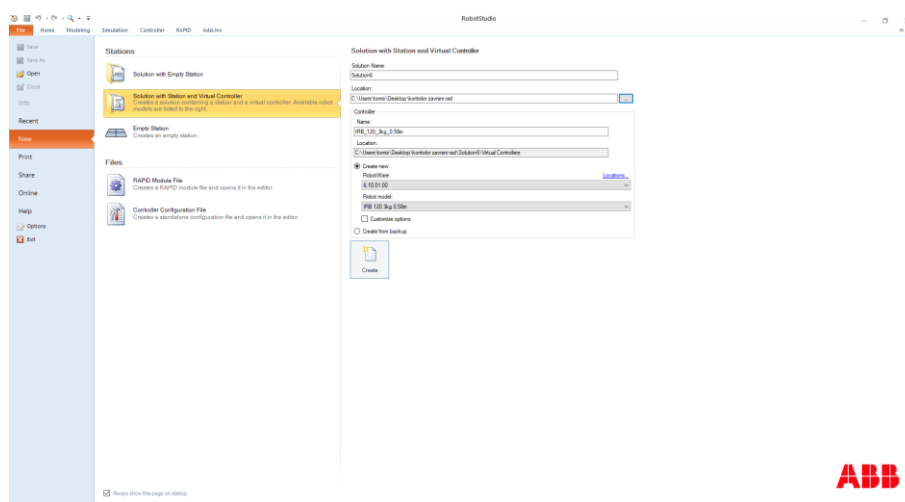
*Slika 3.10 Klizna staza*

## 4. Programski alat RobotStudio za simulaciju sustava

U daljnjem tekstu detaljno je opisano kako se postavi robot i ostale komponente u radno okruženje RobotStudio-a. Opisano je kako se izvodi simulacija kamere pomoću dodatnog programskog sustava “In-Cognex” te kako implementirati “In-Cognex” programski sustav da simulira kameru unutra “RobotStudio-a”.

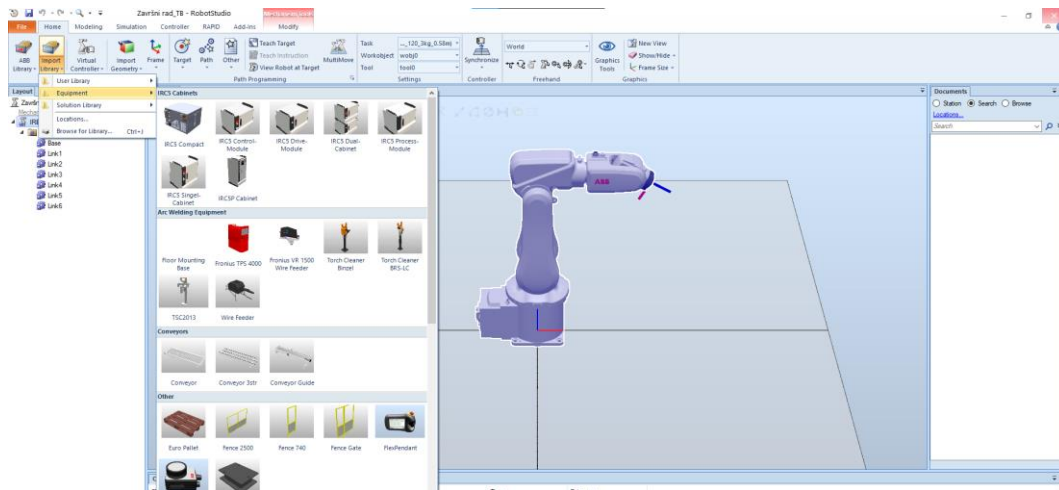
### 4.1. Postavljanje robota i njegovog okruženja

Nakon pokretanja programskog alata RobotStudio otvara se početni izbornik programa. Odabire se “Solution with Station and Virtual Controller” (slika 4.1). Zatim se odrede ostali parametri kao što su naziv okruženja robota i mjesto kamo će se okruženje pohraniti.



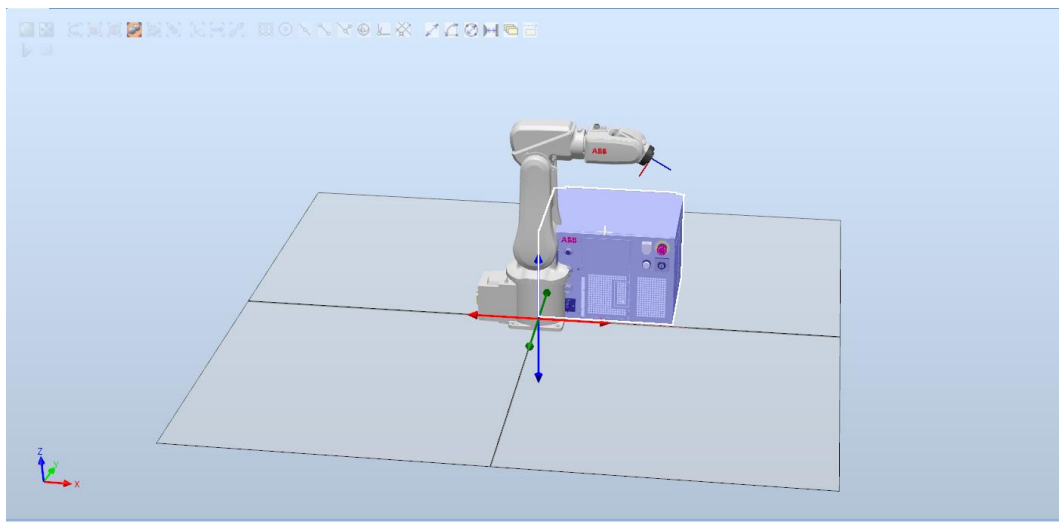
Slika 4.1 Početni izbornik RobotStudio-a

Nakon toga se otvara radno okruženje unutar “RobotStudio-a” sa odabranim robotom “IRB 120”, a zatim se odabire opcija “Import Library” (slika 4.2) te opcija “Equipment” (slika 4.2), tu se nalaze razni alati. Zatim se odabire “IRC5 Compact” (slika 4.2).



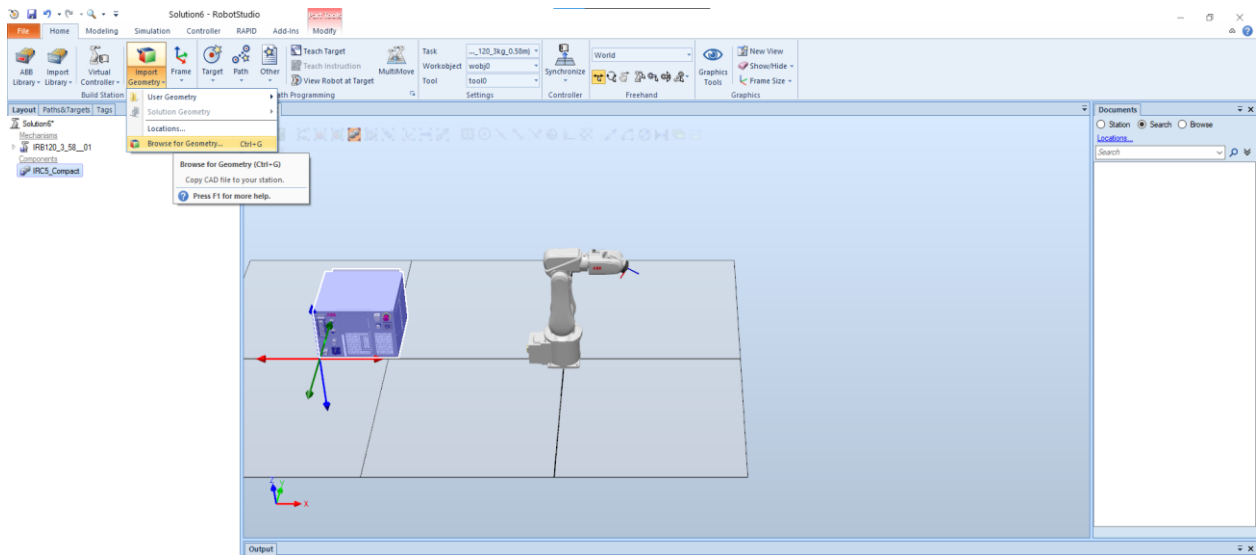
Slika 4.2 Implementacija komponenta

Nakon pritiska na “IRC5 Compact” on se pojavi u radnom okruženju u ishodištu koordinatnog sustava (slika 4.3). Pritiskom na opciju “Move”, te odabirom objekta “IRC5 compact” pojavi se ishodište objekta. Pritiskom i držanjem lijevog klika na mišu na pojedinu os koordinatnog sustava objekta može se micati po odabranoj osi.



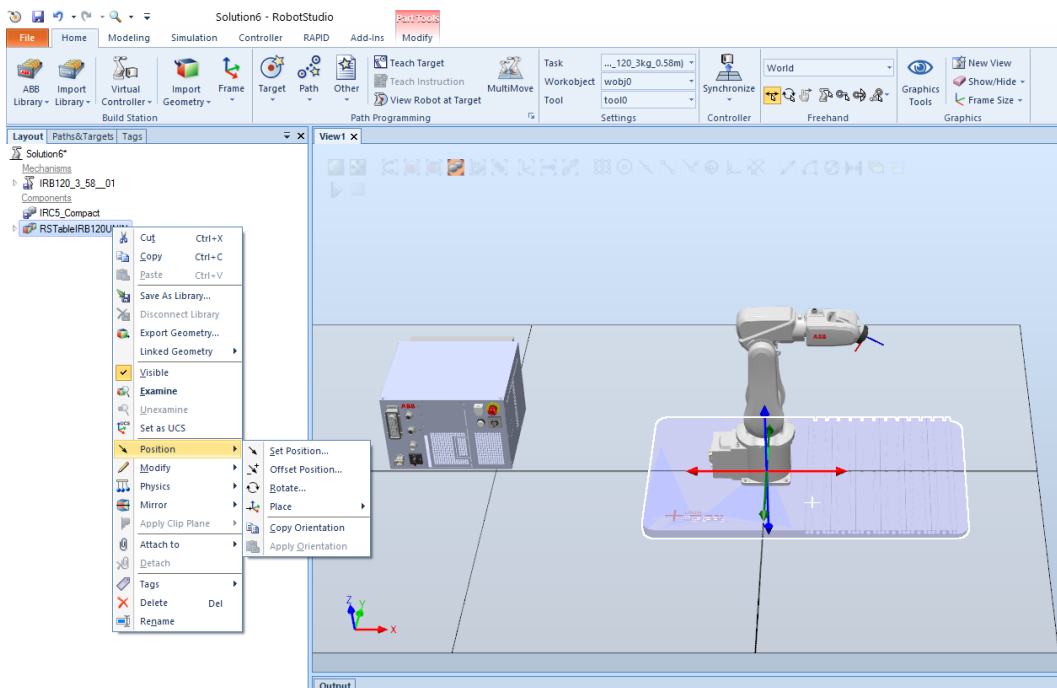
Slika 4.3 Pomicanje komponente IRC5

Da bi se dodali drugi objekti koji se ne nalaze u samom programskom sustavu odabira se opcija “Import geometry” te pod opcija “Browse for geometry”(slika 4.4.). Zatim se pojavi izbornik te se odabire izvor po želji iz kojeg se mogu implementirati drugi objekti. Svi objekti nakon implementacije se pojave u samom središtu izvornog koordinatnog sustava te ih je potrebno pomaknuti na željenu poziciju.

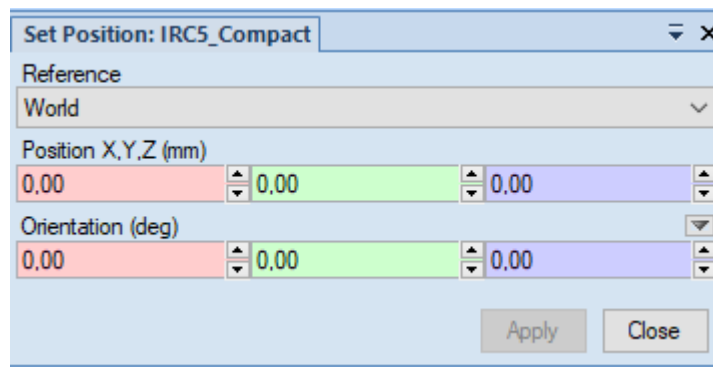


Slika 4.4 Implementacija objekata izvan RobotStudio-a

To se može odraditi na više načina. Pritiskom desnog klika miša na ime objekta ili na sami objekt otvara se dodatni izbornik (slika 4.5). Opcija “position” ima još i dodatne opcije. Korištenjem dodatnih opcija i uz pomoću koordinatnog sustava (slika 4.6) pojedini objekti se postave na željeni položaj unutar radnog prostora.

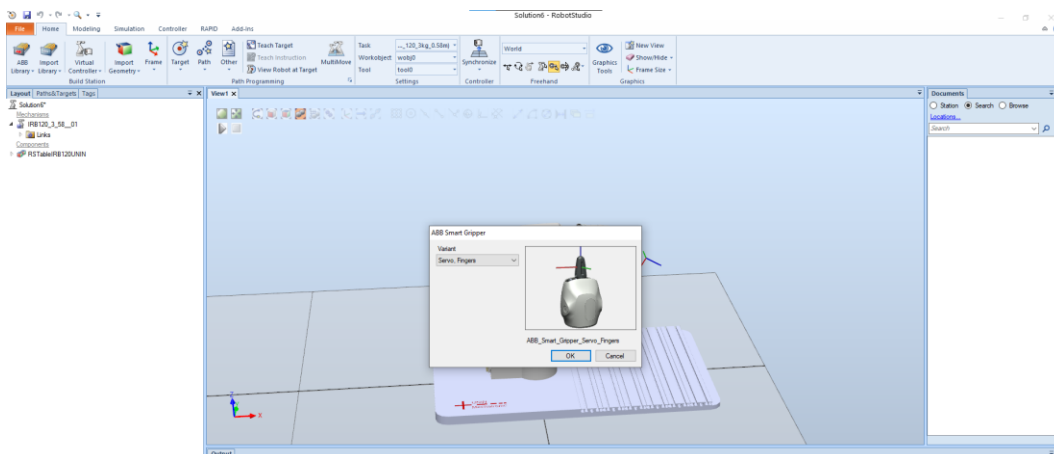


Slika 4.5 Pomicanje objekta pomoću opcije Position



*Slika 4.6 Koordinatni sustav za pomicanje objekta*

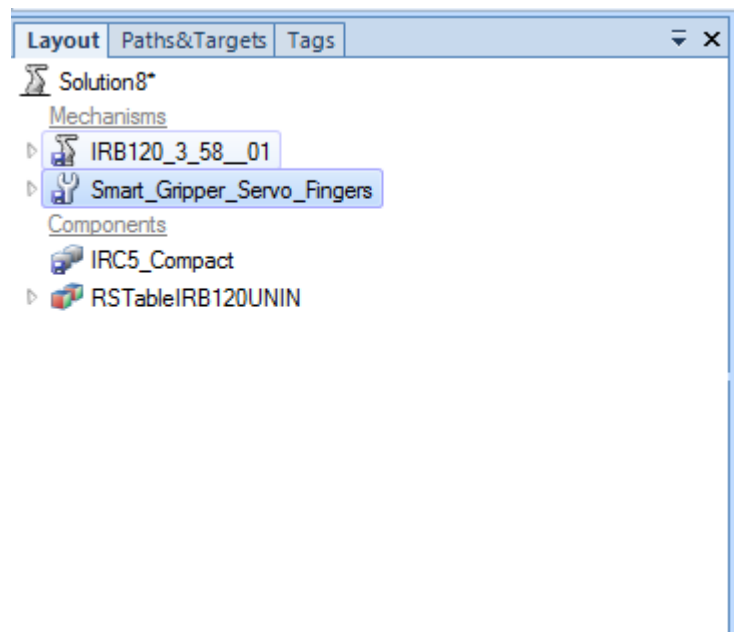
Dodavanje robotske prihvatnice se izvodi na sljedeći način. Otvori se „Import Library“ te pod opcija „Equipment“ zatim pod sekcijom „Tools“ nalazi se „Smart\_Gipper\_Servo\_Fingers“. Taj prihvat dolazi s više opcija. U ovom slučaju odabire se „Servo, Fingers“(slika 4.7).



*Slika 4.7 Dodavanje robotske prihvatnice*

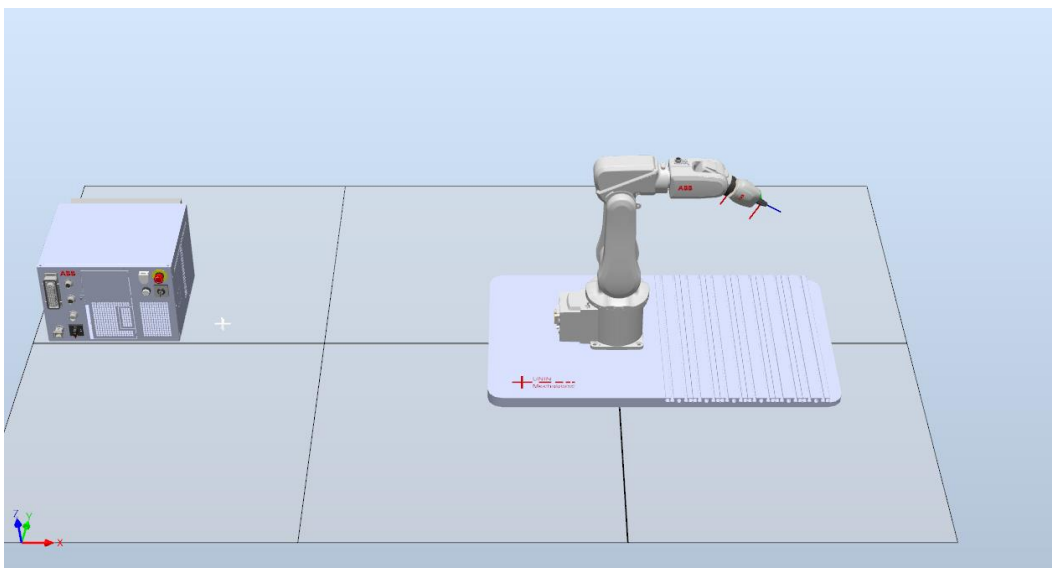
Nakon što je dodan alat u radno okruženje, potrebno ga je postaviti na robot. Sa lijeve strane nalazi se „Layout“ kartica te su na njoj nalaze svi dijelovi unutar radnog prostora. Pritiskom lijevog klika i držanjem na „Smart\_Sgipper\_Servo\_Fingers“ povlači se na ime robota „IRB120\_3\_58\_01“ (slika 4.8).





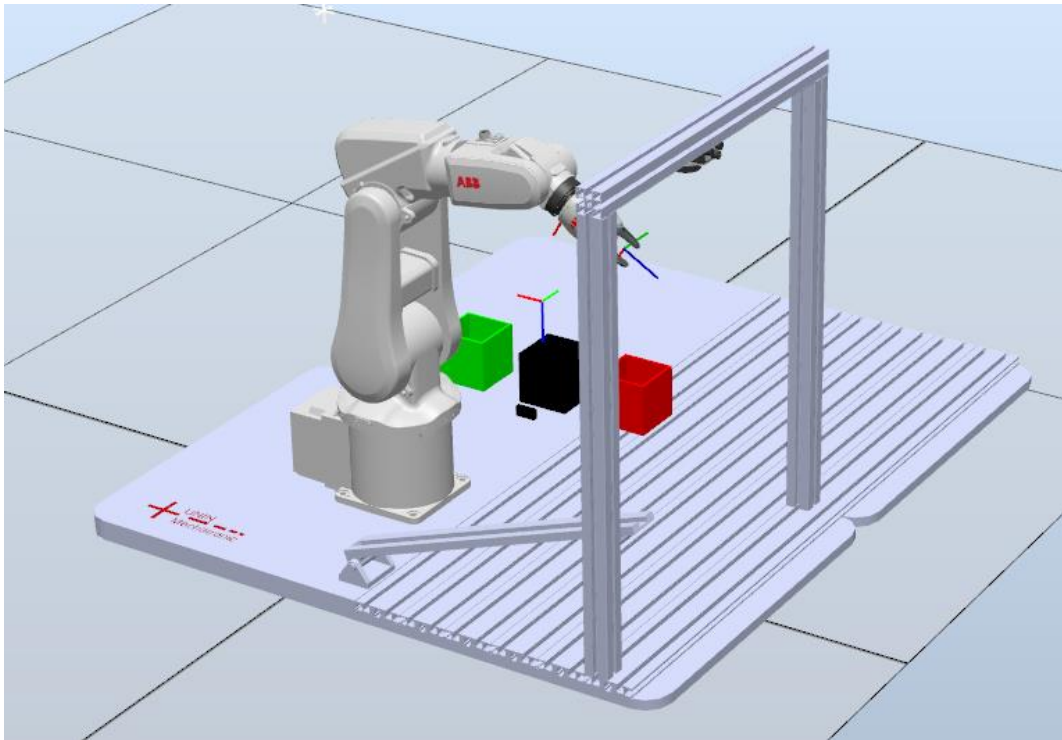
*Slika 4.8 Spajanje robotske prihvatnice na robot*

Zatim se pusti lijevi klik miša te se alat postavi na vrh robota (Slika 4.9).



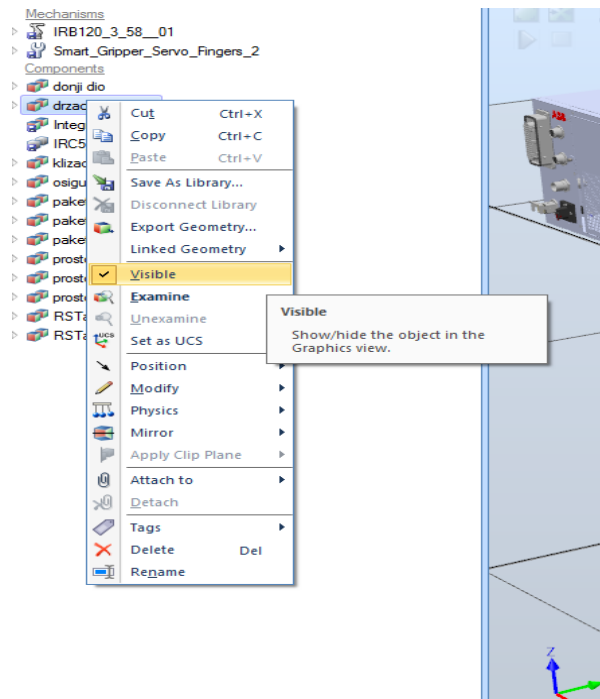
*Slika 4.9 Sklop robot – alat (robotska prihvatnica)*

Implementiraju se i ostali potrebni segmenti potrebni za simulaciju istim načinom kao što je u prethodnim koracima pokazano. Zatim ih se postavlja na željene pozicije pomoću koordinatnog sustava (slika 4.10).



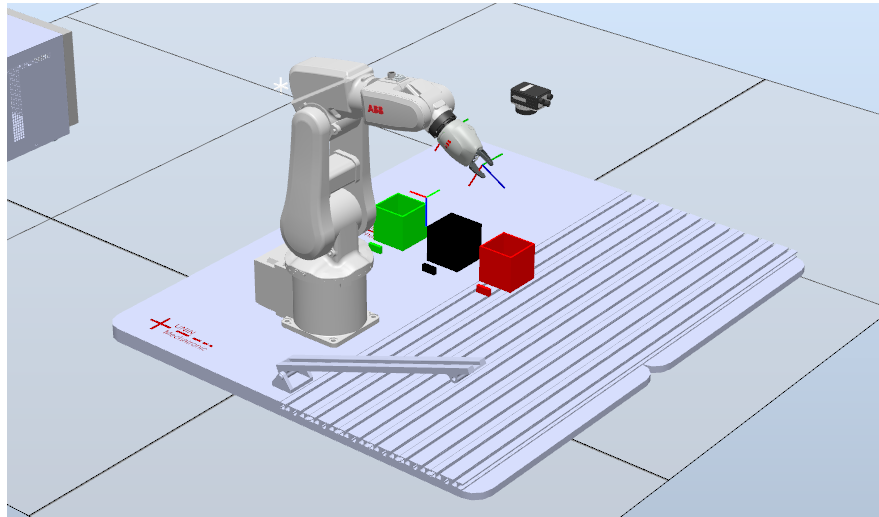
Slika 4.10 Radni prostor simulacije

Držač kamere smeta kod izvođenja same simulacije te njezinog kreiranje te ga je potrebno sakriti. Nosač kamere se označi u „Layout-u“ te se desnim klikom miša otvori dodatni izbornik i odabiremo opciju „Visible“(slika 19.).



Slika 4.11 Sakrivanje objekta zbog preglednosti

Držac kamere je zbog svoje pozicije narušavao preglednost simulacije zbog toga je sakriven. Na slici 24 se može vidjeti koliko je radni prostor pregledniji nakon sakrivanja držača kamere. Nakon što je držac kamere sakriven te su svi objekti na željenim pozicijama, kreće se s izradom putanji robota.



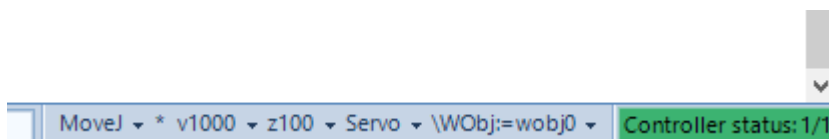
*Slika 4.12 Radni prostor nakon sakrivanja držača kamere*

## **4.2. Kreiranje putanje robota**

Nakon što je radni prostor izgrađen, potrebno je izraditi putanje robota. Zadatak robota je kretanje iz „Home position“ početne točke prema kliznoj stazi na kojoj se nalaze paketi. Robot se pozicionira iznad klizne staze te se spušta prema poziciji za prihvat paketa. Zatim uzima paket. Nakon što je paket uhvaćen pomoću robotske prihvatnice, ponovno se podiže. Robot kreće prema kameri na skeniranje paketa. Kamera skenira paket te šalje signal kontroleru. Sve se ovo odvija na putanji „path\_scan“. Na temelju tog istog signala robot kreće po jednoj od 3 zadane putanje, a potom se vraća u „Home position“ te kreće po novi paket. Postoje putanje „path\_Vz“, „path\_Gr“, „path\_Zg“. Crveni pravokutnik označava paket Vz, zeleni pravokutnik paket Zg te crni paket označava paket Gr. VZ paket predstavlja pakete koji se šalje u Varaždin, Zg paket pakete koji se šalju u Zagreb, a 3 paket GR predstavlja paket s greškom. Prije nego što se krene na kreiranje signala te upravljanja potrebno je kreirati putanje. U daljem tekstu je objašnjeno kako se kreiraju točke putanje te konačna putanja.

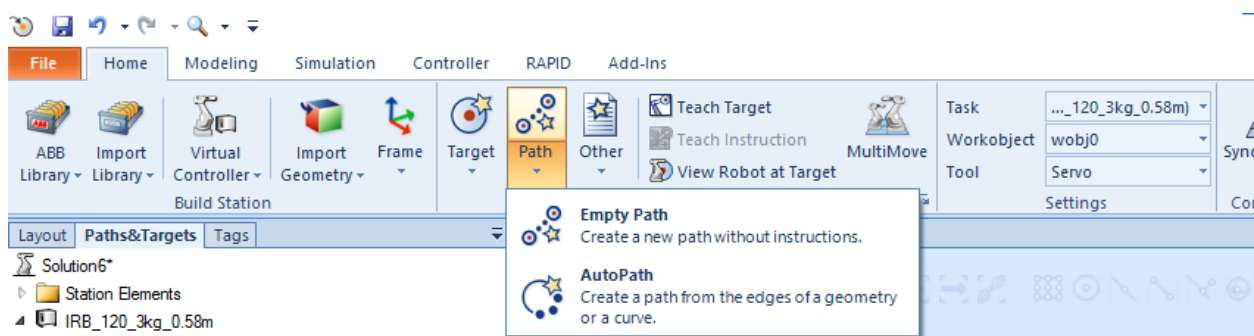
Prije kreiranja same putanje potrebne su zasebne točke putanje. U ovoj simulaciji se koristi metoda „Od točke do točke“ eng. „Point to point“ metoda. Ova metoda se najčešće koristi kod premještanja predmeta, točkastog zavarivanja, bušenja te slično. Metoda od točke do točke funkcionira na sljedeći način. Robot se pomiče u zadanu poziciju, obavi zadatak npr. prihvati paket

te zatim kreće u sljedeću zadanu točku. Za ovu metodu gibanja koristi se „MoveJ“ (slika 4.13) opcija. Ova opcija omogućuje da se svi zglobovi robota kreću u isto vrijeme kako bi robot došao u zadanu poziciju u najkraćem roku dok se kod „MoveL“ opcije robot kreće linearno između zadanih točaka.



Slika 4.13 Odabir kretnje robota

Potrebno je kreirati praznu putanju, a to se izvodi na sljedeći način (slika 4.14) Nakon što je kreirana prazna putanja potrebno je kreirati točke tj. „Target“ putanje. Potrebno je prethodno dovesti robota u željeni položaj pomoću opcija koje se nalaze na alatnoj traci sekcija „Freehand“ (slika 4.15). Kada ja robot doveden u željenu poziciju, potrebno je pritisnuti opciju „Teach Target“ (slika 4.14) kako bi se stvorila točka koja će kasnije biti dio neke putanje. Kako bi se točke stvorile na vrhu alata (TCP), potrebno je odabrati alat „Tool“ (slika 4.14). Odabire se alat koji se koristi u simulaciji, a to je alat „Servo“.



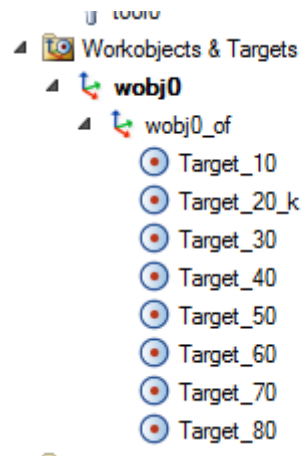
Slika 4.14 Kreiranje prazne putanje

Unutar „FreeHand“ alatne trake nalaze se svi potrebni alati koji se koriste za postavljanje robota u željenu poziciju. Robot se može micati svakim zglobom posebno ili pomicanjem vrha alata. Ako se pomiče vrh alata, miču se svi zglobovi zajedno kako bi se postavio u željeni položaj.

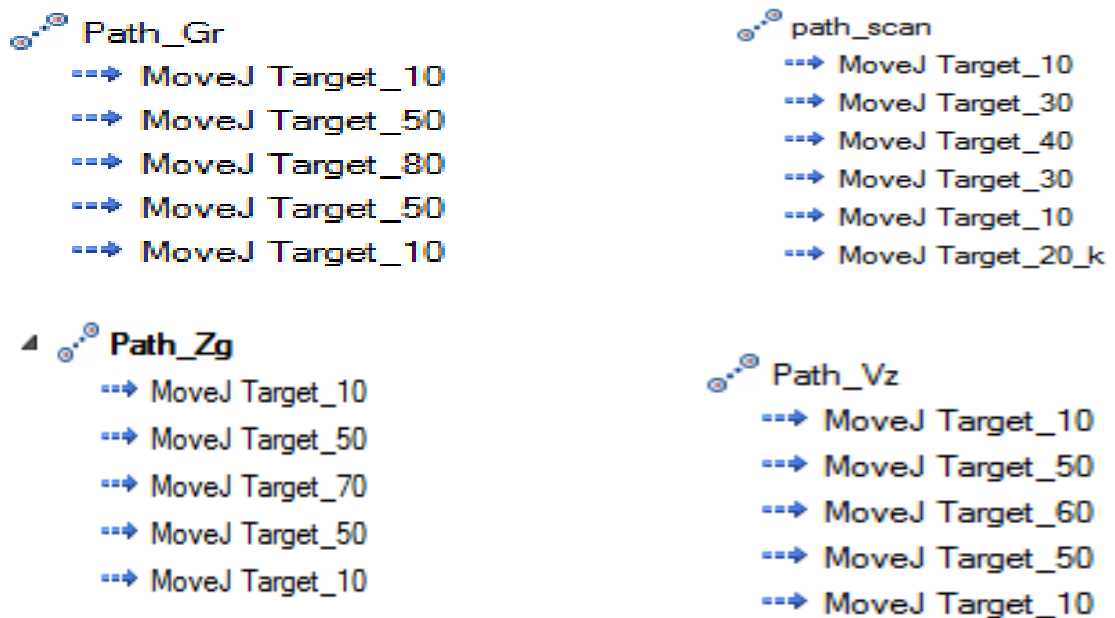


Slika 4.15 Freehand sekcija alata za pomicanje robota

Kada su kreirane točke putanje (slika 4.16), potrebno ih je staviti u određenu putanju. Točke putanje se mogu koristiti u više putanja. Točke možemo prebaciti u putanju tako što se na nju pritisne lijevim klikom miša te se povuče u željenu putanju. Potrebno je obratiti pažnju na to koja se točka gdje nalazi da bi se kreirala željena putanja (slika 4.17). Da bi se provjerilo je li putanju moguće izvesti desnim klikom miša, na ime putanje otvara se dodatni izbornik te opcijom „*Move along path*“ robot se kreće zadanom putanjom. Ako se ne može ostvariti putanja zadanim točkama, robot staje i javlja grešku.



Slika 4.16 Kreirane točke putanje

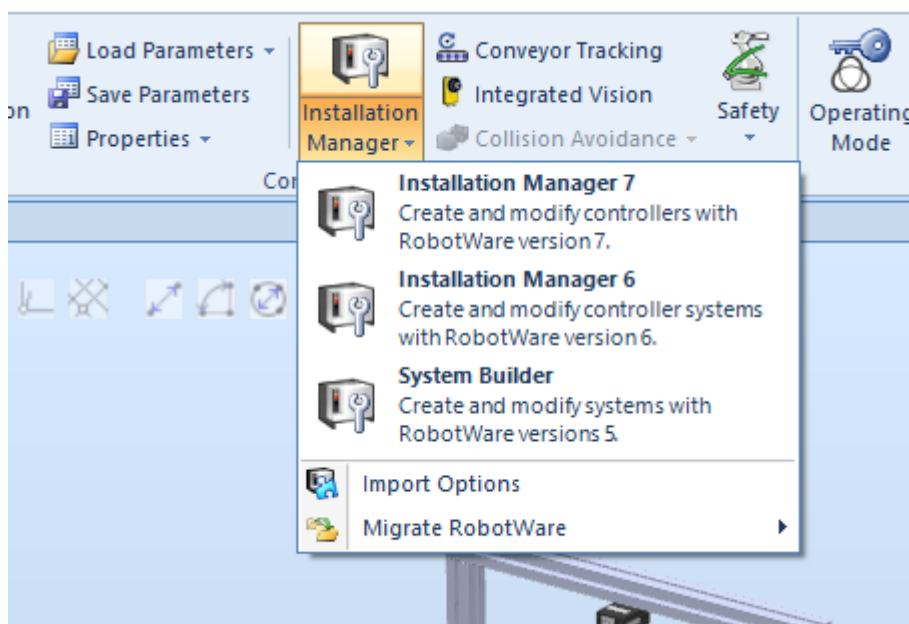


Slika 4.17 Putanje pojedinog paketa

### 4.3. In-Sight Eyplorer programsko sučelje

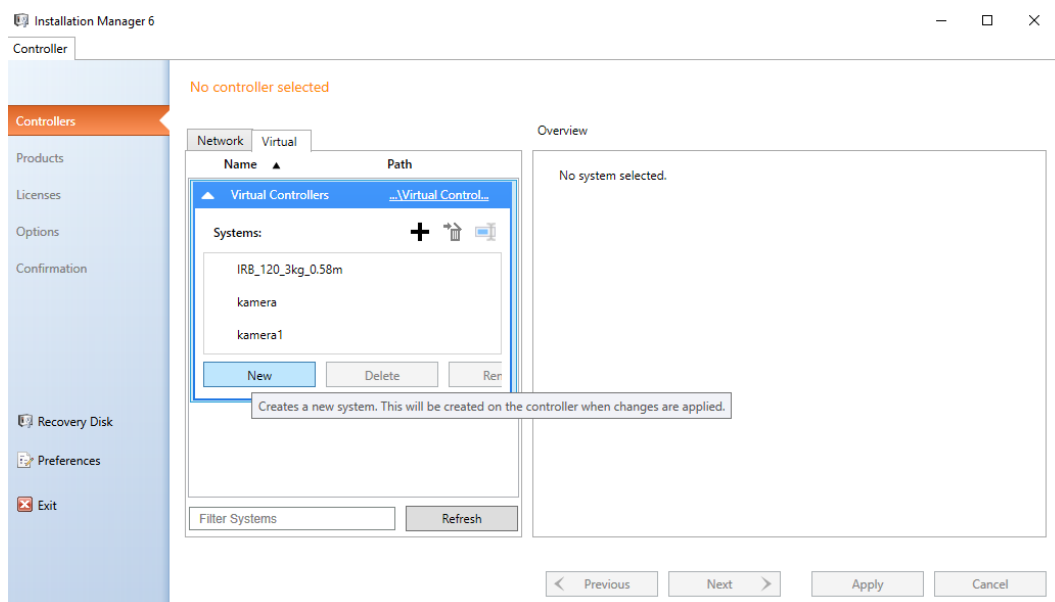
Pomoću *In-Sight* softvera simulira se stvarna kamera. Zbog izrade simulacije kod kuće ili izvan laboratorija kada nije moguće spojiti fizičku kameru, može se koristiti virtualna kamera. Također, pomoću ovog softvera moguće je programirati kameru. To znači da se nakon spajanja kamere unutar softvera nalaze integrirani alati pomoću kojih je moguće kreirati zadatke za kameru. Nakon što su zadaci kreirani, prenose se na radnu memoriju kamere te joj softver nije potreban za rad već ona sama odrađuje svoje zadatke te šalje rezultate pomoću izlaznih signala. U ovoj simulaciji softver nam služi kako bi samo simulirao kameru kojoj se zatim zadaju zadaci unutar RobotStudio-a. Spajanje te zadavanje zadataka virtualnoj kameri pomoću *In-sight Explorer* programa te RobotStudio-a se izvodi se na sljedeći način.

Prvi korak je pokretanje RobotStudio-a te kreiranje simulacije to jest radno okruženje. Nakon što je kreirano radno okruženje potrebno je kreirati virtualni kontrolor kamere. To se izvodi pomoću opcije "*Installation Manager 6*" koja se nalazi unutar izbornika "*Installation Manager*" (slika 4.18).



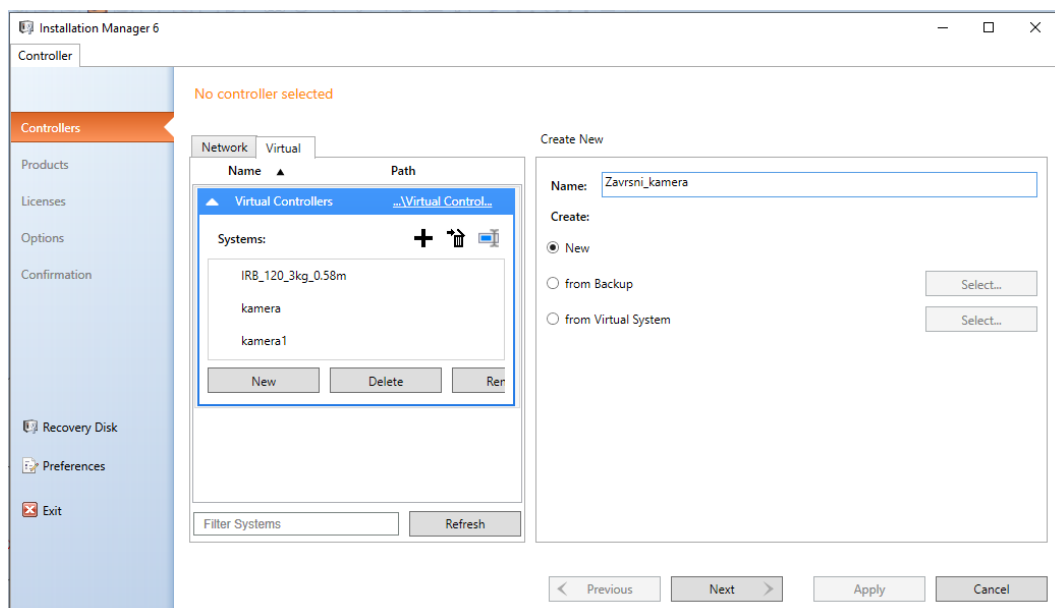
Slika 4.18 Kreiranje kontrolera za kameru

Nakon što je odabrana opcija "*Installation Manager 6*" otvara se izbornik za kreiranje novog kontrolera. Kontroler je potrebno podesiti kako je prikazano u daljem tekstu (slike 4.19-4.22). Potrebno je obratiti pozornost na lokaciju gdje je spremljen sam kontroler unutar računala.



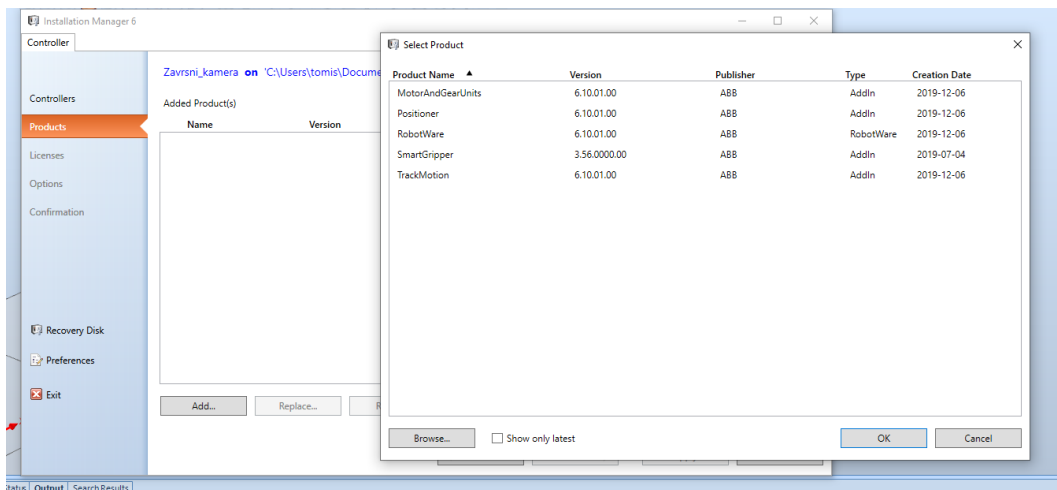
Slika 4.19 Izbornik za kreiranje virtualnog kontrolera za kameru

Nakon pritiska na opciju „New“ otvara se izbornik za definiranje imena virtualnog kontrolera (slika 4.20).



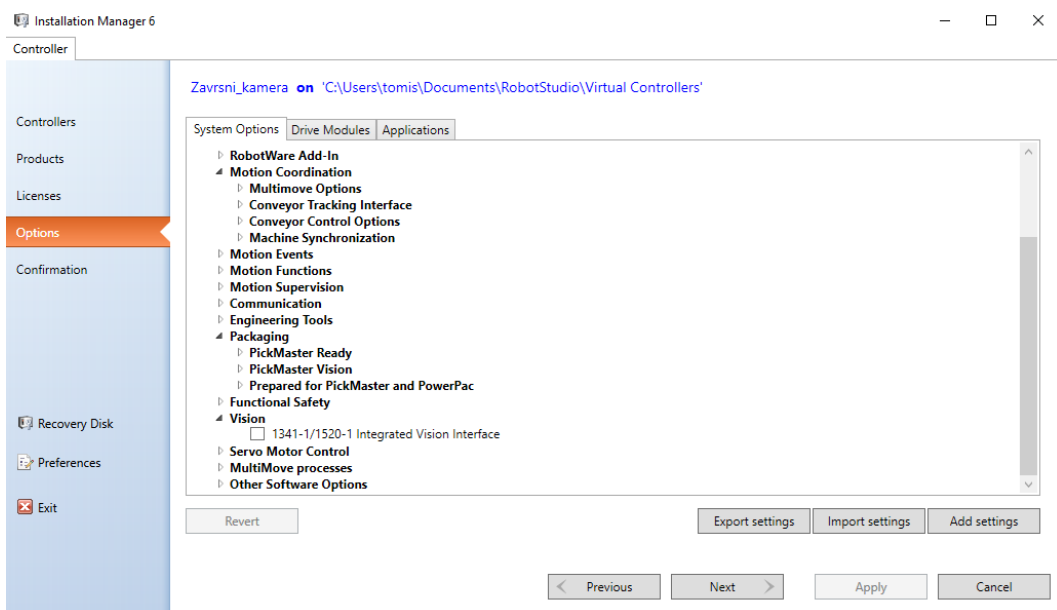
Slika 4.20 Upisivanje imena novog virtualnog kontrolera

Nakon upisivanja imena potrebno je pritisnuti „Next“. Zatim je potrebno pritisnuti „Add...“ kako bi se odabrao novi alat kojim će kontroler upravljati. Odabire se „RobotWare“ (slika 4.21) koji će upravljati virtualnom kamerom. Nakon toga potrebno je potvrditi odabir na „OK“. Izbornik se zatvara te se može krenuti na sljedeći korak pritiskom na opciju „Next“ dva puta.



Slika 4.21 Prikaz elemenata okruženja kojima će novi virtualni kontroler upravljati

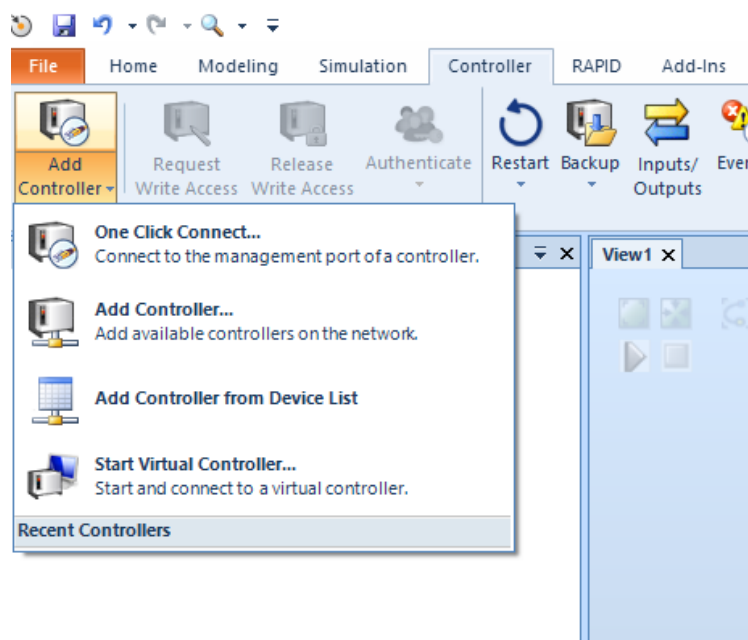
Tada je potrebno kod dodatnih opcija uključiti „*Integrated Vision Interface*“ (slika 4.22) kako bi se kasnije u simulaciji mogla koristiti virtualna kamera. Potvrđuje se odabir na „*Next*“ te nakon toga dodatna potvrda svih postavki na opciju „*Apply*“.



Slika 4.22 Odabir dodatne opcije „*Integrated Vision Interface*“

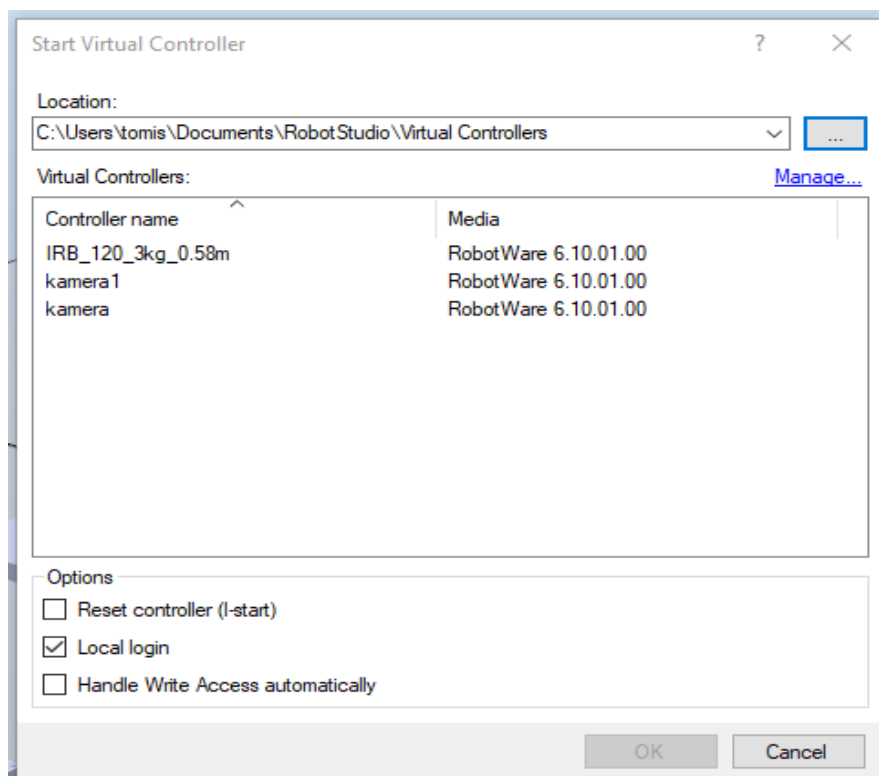


Nakon što je kontroler kreiran, potrebno ga je pokrenuti (slika 4.23).



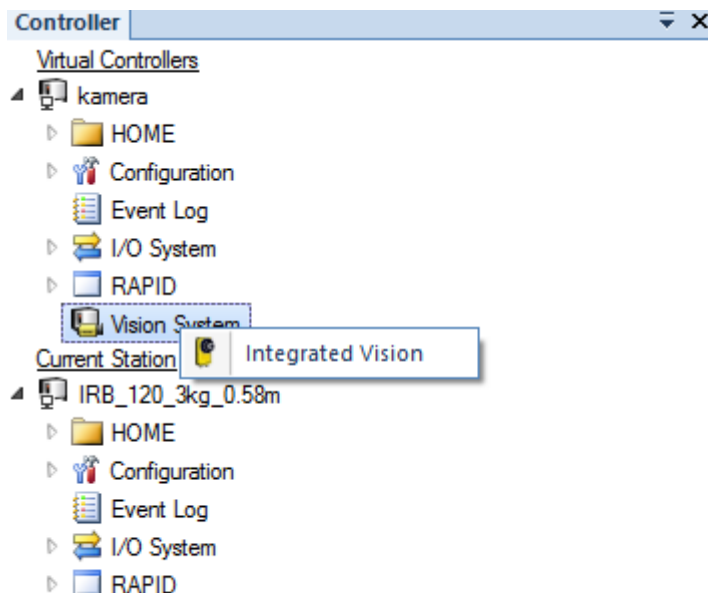
Slika 4.23 Pokretanje virtualnog kontrolera za kameru

Odabire se lokacija kontrolera koji je kreiran prethodno (slika 4.24) te je potrebno potvrditi odabir pritiskom na „OK“.



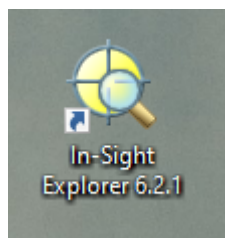
Slika 4.24 Odabir virtualnog kontrolera

Sa lijeve strane u kartici „*Controller*“ pojavi se novi virtualni kontroler sa kamerom. Desnim klikom miša na „*Vision system*“ otvara se dodatni izbornik te se odabire opcija „*Integrated Vision*“ (slika 4.25).



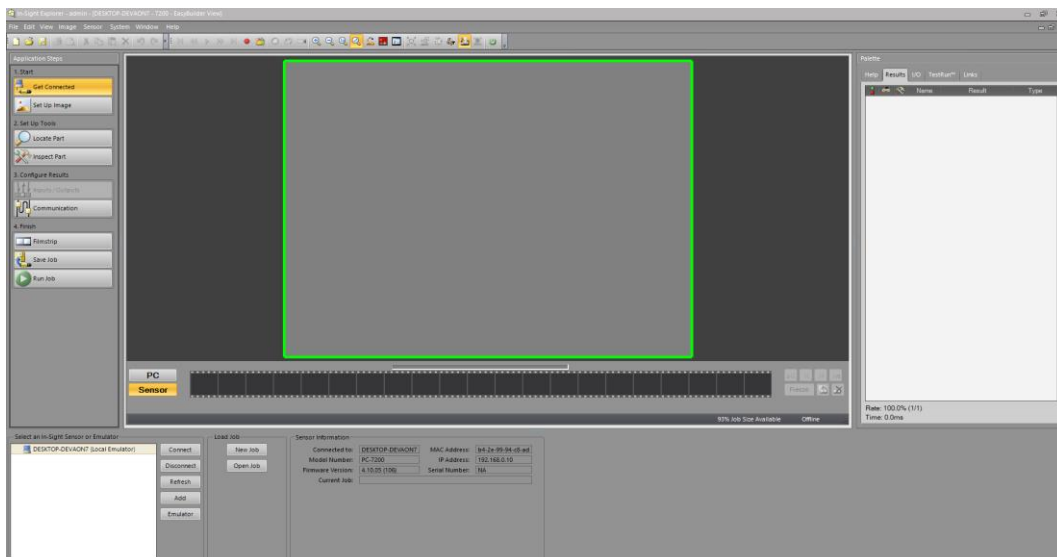
Slika 4.25 Pokretanje kamere

Zatim je potrebno pokrenuti dodatni program za simulaciju „*In-Sight Explorer*“ (slika 4.26).



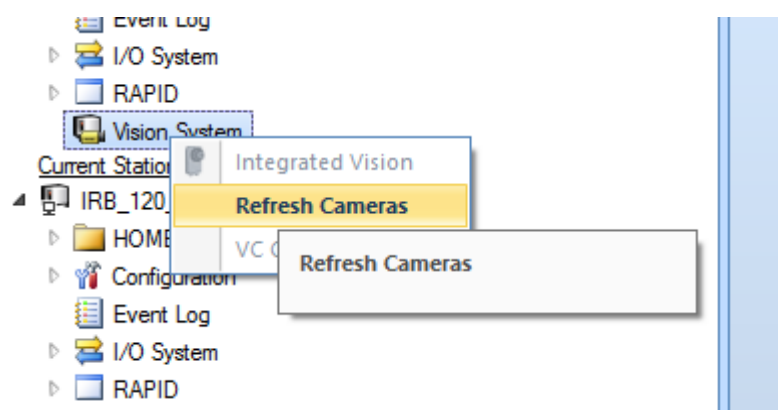
Slika 4.26 Dodatni softver za simuliranje kamere „*In-Sight Explorer*“

Nakon pokretanja otvara se početni izbornik softvera (slika 4.27). Početni izbornik može izgledati i drugačije ako nije pokrenut RobotStudio.



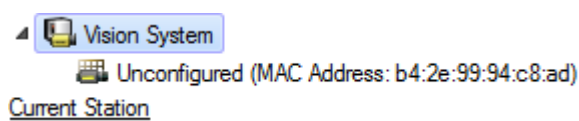
Slika 4.27 Početni izbornik unutar „In-Sight Explorer-a“

Nakon pokretanja „In-Sight Explorer-a“ potrebno je pritisnuti desni klik na „Vision System“ (slika 4.28). Sada postoje dodatne opcije te se odabire opcija „Refresh Cameras“ (slika 4.28). Ponekad program ne želi prepoznati kameru pa su potrebni dodatni koraci. Ako kamera nije prepoznata, odabire se opcija „VC Configuration“. Nakon toga potrebno je odabrati istu IP adresu koja piše u „In-Sight Exploreru“ (slika 4.27).



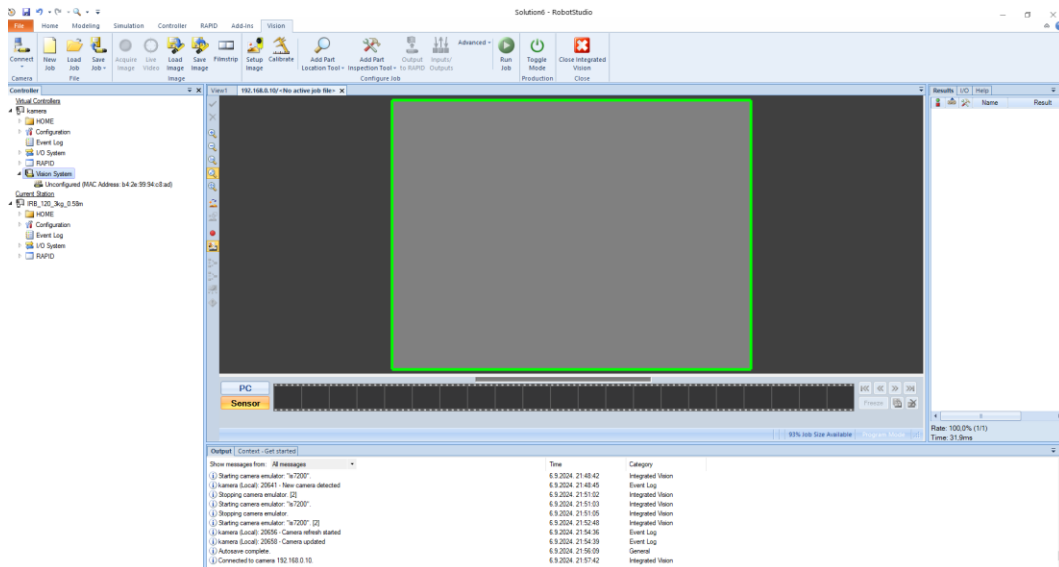
Slika 4.28 Opcija „refresh Cameras“

Nakon odabira opcije „refresh cameras“ trebala bi se pojaviti kamera.



Slika 4.29 Virtualna kamera

Dvostrukim klikom na kameru otvara nam se sučelje kamere unutar RobotStudio-a (slika 4.30). Unutar toga zadaju se zadaci kameri i podešavaju ostali parametri kamere za izradu simulacije. Kasnije tokom ovog rada će se izvršiti podešavanje te zadavanje zadataka.



Slika 4.30 Virtualna kamera unutar RobotStudio-a

## 5. Opis objekta „SmartComponent“

„*SmartComponent*“ je opcija koja služi da nekom objektu unutar „RobotStudio-a“ daje mogućnost da objekt dobije fizička svojstva. Također da se objektom može manipulirati robotom te samim signalima tijekom simulacije.

Pomoću ove opcije omogućuju se složenija ponašanja objekta. Pod složenije ponašanje smatra se gibanje objekta po transporterima, u ovom slučaju klizna staza paketa, gibanje prihvatnice robota i slično. Unutar softvera RobotStudio-a kontrolerima se definiraju virtualni signali koji upravljaju samom simulacijom. Upravo ti virtualni signali daju mogućnost da se sama simulacija izvede što vjerodostojnije tj. realnije. Nazivamo ih virtualni signali, a zapravo simuliraju realne signale koji bi upravljali realnim sustavom. Kontrolor šalje signal robotu, on prihvaća paket te ga pozicionira na određeno mjesto ispod kamere. Pošto se koristi virtualna kamera, sam korisnik simulira signal kamere. Na temelju tog signala robot odnosi paket na određeno mjesto.

„*SmartComponent*“ ima ulazne i izlazne signale te na taj način komunicira s ostalim elementima unutar simulacije. Postoji više vrsta signala unutar RobotStudio-a :

DigitalInput

DigitalOutput

DigitalGroupInput

DigitalGroupOutput

AnalogInput

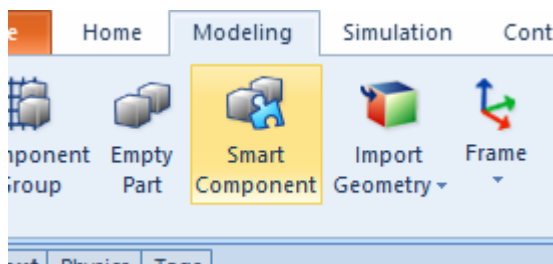
AnalogOutput

Tijekom idućeg poglavlja pokazano je kako se koriste signali te kako se njima upravlja.

## 6. Dopremanje paketa na kliznu stazu

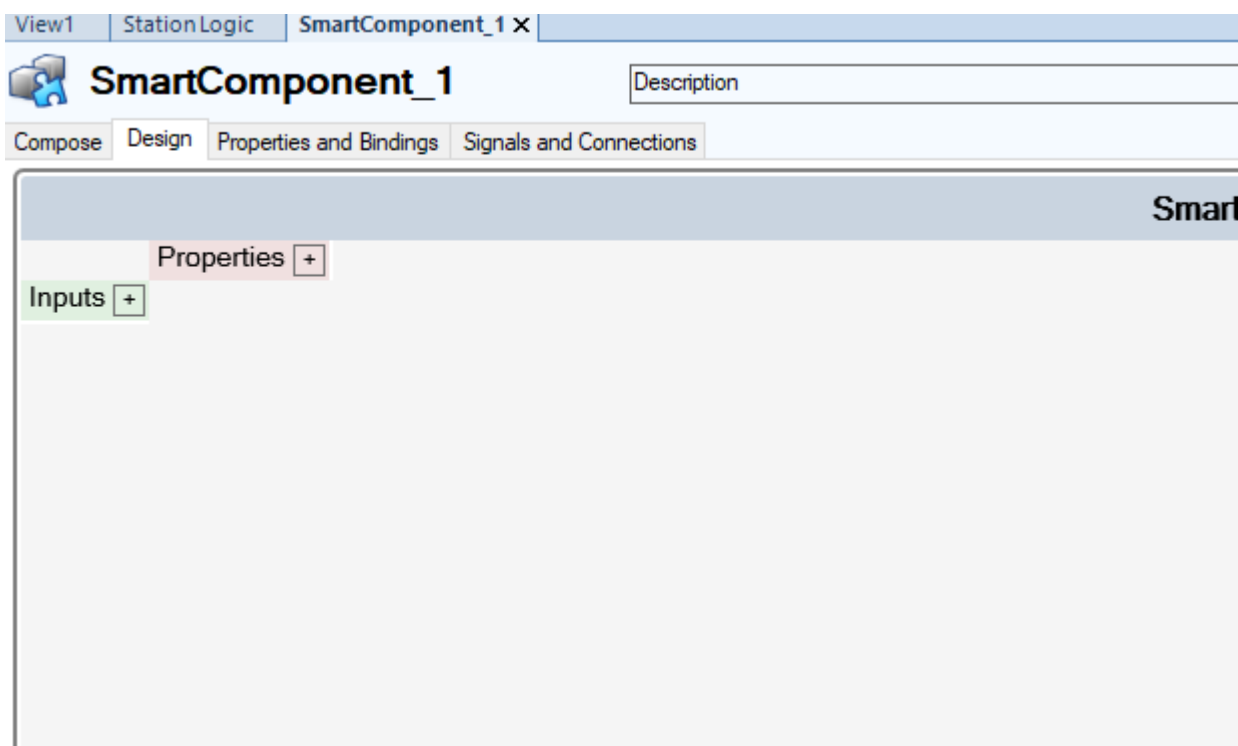
U daljem tekstu će se pokazati princip kojim se paketi definiraju na kliznu stazu te način upravljanja signalima da se paketi stvore na određenom mjestu kada se izvrši neka radnja. Signal koji će doći na kraju izvođenja neke radnje generira novi paket.

Kako bi se stvorio pojedini paket na kliznoj stazi, potrebno je za svaki paket stvoriti „*Smart Component*“ (slika 6.1). Stvaranje novog „*Smart Componenta*“ se izvodi na sljedeći način. Potrebno je odabrati karticu alata „*Modeling*“ te opciju „*Smart Component*“.



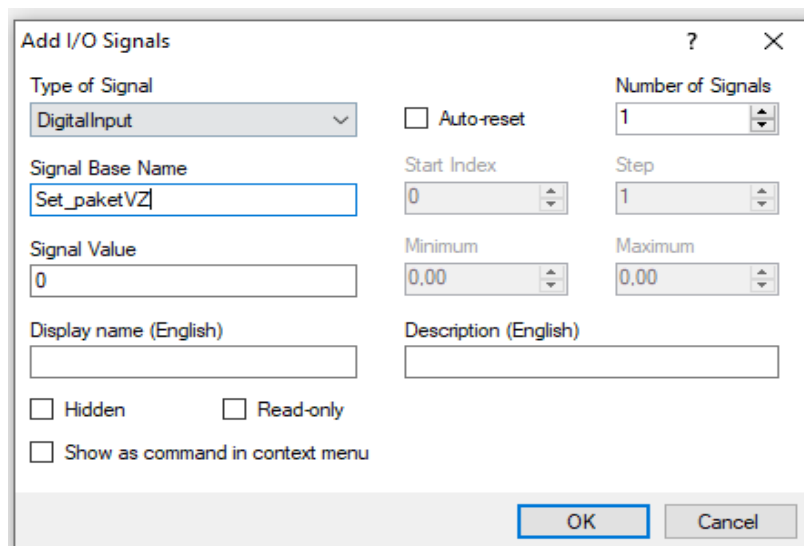
Slika 6.1 Stvaranje „*Smart Component*“ objekta

Nakon odabira opcije otvara se izbornik za definiranje svih parametara „*Smart Componenta*“ (slika 6.2). Pod „*Design*“ karticom odabire se opcija za kreiranje novog ulaznog signala „*Inputs*“.



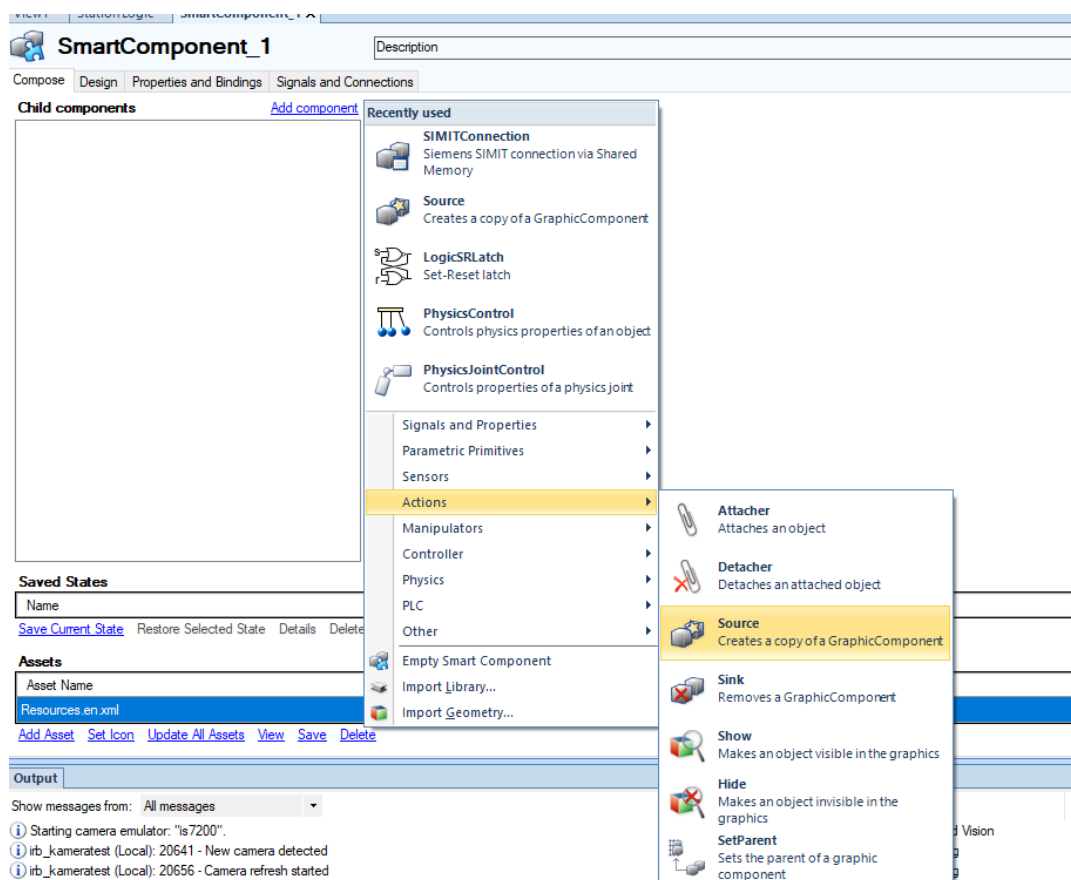
Slika 6.2 Stvaranje ulaznog signala koji će dopreмати paket na kliznu stazu

Zatim se otvara izbornik za definiranje ulaznog signala. Ulazni signal je potrebno definirati kao što je prikazano na slici 6.3.



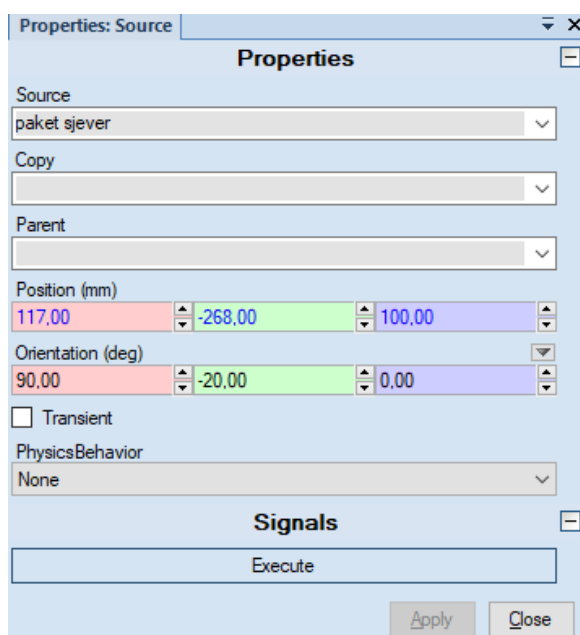
Slika 6.3 Izbornik za definiranje parametara ulaznog signala

Idući korak je dodavanje izvora paketa. To znači da je potrebno odabrati objekt koji će se kopirati. To se izvodi na sljedeći način. Odabire se kartica „Compose“ (slika 6.4), a zatim pritiskom na „Add component“ se dodaju ostali parametri. Potrebno je dodati sljedeće komponente : *Source*, *Attacher* i *Detacher*.



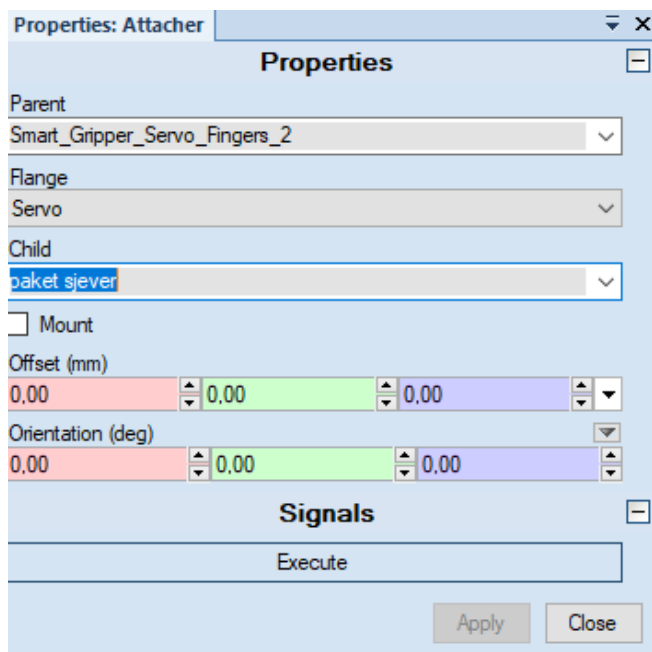
Slika 6.4 Stvaranje izvora paketa

Nakon dodavanja „Source“ komponente potrebno ju je definirati prema slici 6.5. Vrlo važno je odabrati odgovarajući „Source“, to jest objekt koji će se kopirati, te mu zadati točne koordinate na koje ga se želi kopirati.



Slika 6.5 Definiranje parametara „Source“

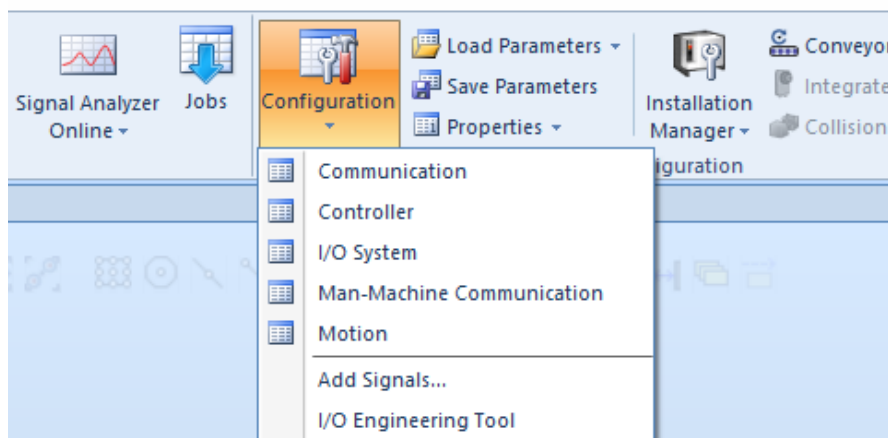
Zatim je potrebno definirati parametre „Attacher“ prema slici 6.6. Ta opcija služi za povezivanje hvataljke robota s paketom, to jest da ga se može micati po zadanim putanjama robota.



Slika 6.6 Definiranje parametara „Attacher“

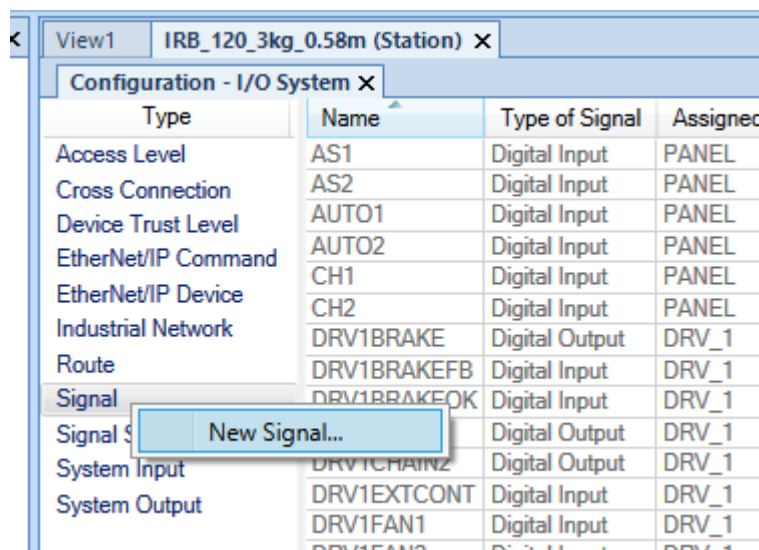


Idući korak je dodavanje novih signala koji će se kasnije koristiti za upravljanje sustavom. Potrebno je odabrati opciju „*Configuration*“ te zatim pod opciju „*I/O system*“ (slika 6.7).



Slika 6.7 Otvaranje ulazno izlaznih signala kontrolera

Idući korak je kreirati pojedine signale za upravljanje. Desnim klikom na „*Signal*“ otvara se dodatna opcija „*New Signal...*“ (slika 6.8).



Slika 6.8 Dodavanje novog signala u kontroleru

Nakon odabira opcije „*New Signal*“ otvara se izbornik za definiranje novog signala. Potrebno je definirati novi signal prema slici 6.9. Ovdje je prikaz kako je potrebno definirati signal koji će kasnije označiti početak gibanja robota. Nakon što su definirani svi parametri potrebno je potvrditi klikom na OK.

Name	Value	Information
Name	start	Changed
Type of Signal	Digital Input	Changed
Assigned to Device		
Signal Identification Label		
Category		
Access Level	Default	
Default Value	0	
Filter Time Passive (ms)	0	
Filter Time Active (ms)	0	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	

**Value (string)**  
 The changes will not take effect until the controller is restarted.

OK Cancel

*Slika 6.9 Definiranje signala koji će pokrenuti gibanje robota*

Potrebno je kreirati i dodatne signale. Sljedeći signal koji se dodaje je signal za prihvat paketa (slika 6.10) . Taj signal je ovisan o položaju robota. Taj signal je potrebno staviti na mjesto kada robot dođe u poziciju za prihvat paketa. Tada dolazi signal u kontrolnu jedincu te se izvodi prihvat paketa. U daljem tekstu će biti dodatno pojašnjena točna pozicija signala.

Name	Value	Information
Name	prihvat	Changed
Type of Signal	Digital Output	Changed
Assigned to Device		
Signal Identification Label		
Category		
Access Level	Default	
Default Value	0	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Safe Level	DefaultSafeLevel	

**Value (string)**  
 The changes will not take effect until the controller is restarted.

OK Cancel

*Slika 6.10 Definiranje signala za prihvat paketa*

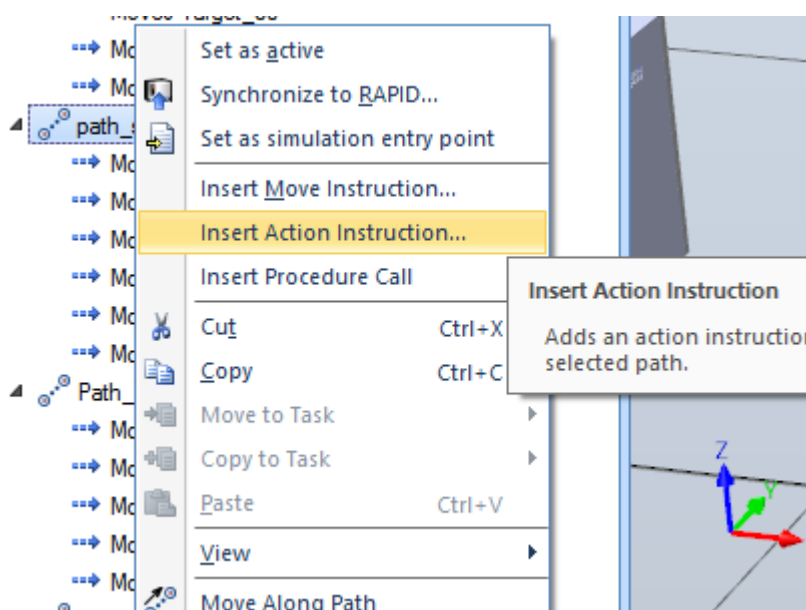
Kako je potrebno kreirati signal za prihvat paketa, tako je potrebno kreirati i signal za otpuštanje paketa nakon što je robot došao na željenu poziciju na putanji. Signal za otpuštanje paketa potrebno je definirati prema slici 6.11.

Name	Value	Information
Name	otpustanje	Changed
Type of Signal	Digital Input	Changed
Assigned to Device		
Signal Identification Label		
Category		
Access Level	Default	
Default Value	0	
Filter Time Passive (ms)	0	
Filter Time Active (ms)	0	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	

*Slika 6.11 Definiranje signala za otpuštanje paketa*

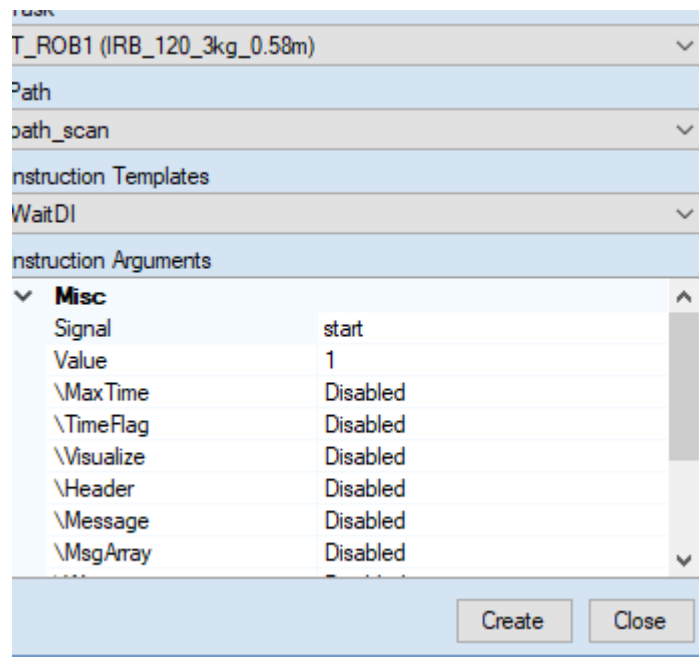
Potrebno je izraditi i dodatne signale koji će simulirati izlazne signale kamere. Zbog toga što se koristi virtualna kamera, ne možemo dobiti njezine signale u kontrolor. Zbog toga ih je potrebno samostalno kreirati i kontrolirati. Signali se kreiraju istim postupkom kako su kreirani i prethodni signali. Signale koji su se dodatno kreirali su nazvani „*scan\_vz*“, „*scan\_zg*“, „*scan\_gr*“. Svaki od tih signala će simulirati kad je koji paket skeniran na kameri tijekom simulacije.

Idući korak je definirati instrukcije unutar putanje robota pomoću kojih se upravlja signalima u željenom trenutku. Pomoću „*Insert Action Instruction*“ (slika 6.12) se kreiraju instrukcije unutar putanje. One služe aktiviranju pojedinog signala kada robot dođe u željenu poziciju na putanji.



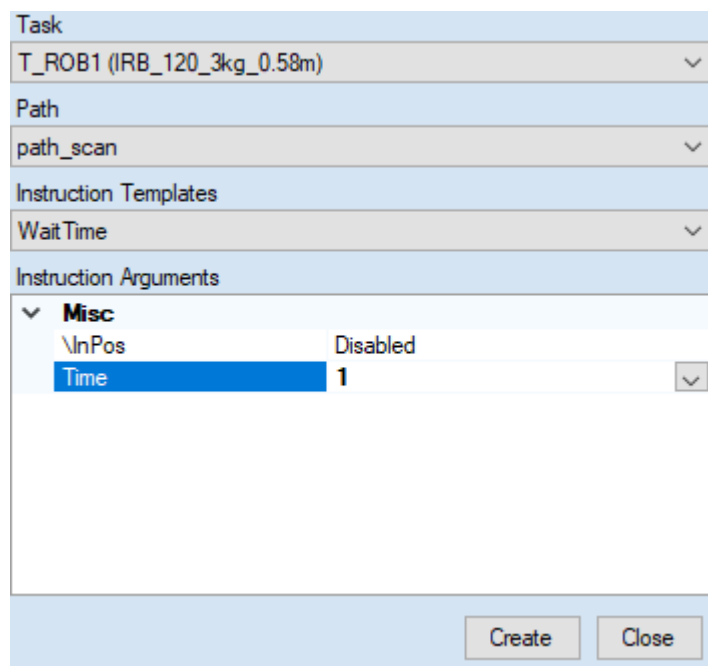
Slika 6.12 Opcija „*Insert Action Instruction*“

Nakon što je instrukcija kreirana unutar putanje, potrebno ju je definirati u potpunosti (slika 6.13 ). Potrebno je odabrati putanju u koju želimo implementirati instrukciju te na koji način se upravlja instrukcijom. Odabire se opcija „*Wait DI*“ unutar opcije „*Instruction Templates*“. Ova instrukcija funkcionira na način da čeka da signal postigne vrijednost „1“, a kada instrukcija postigne zadanu vrijednost, pušta robota da se dalje kreće po putanji. U ovom slučaju potrebno je odabrati signal „*start*“ koji će omogućiti da se robot počne kretati.



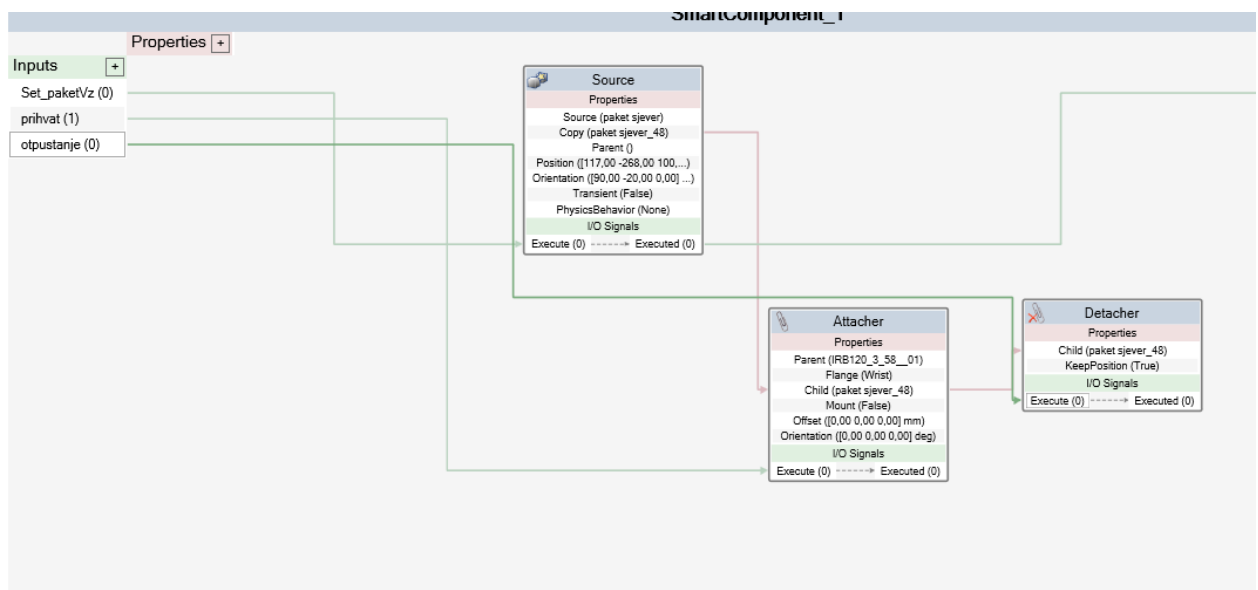
Slika 6.13 Definiranje instrukcije unutar putanje robota

Idući korak je implementiranje instrukcija „WaitTime“, a zatim „Set“. „Set“ instrukcija nam služi da nam željeni signal postavi u vrijednost po želji: 1 ili 0. Unutar definiranja instrukcije „Set“ izabire se signal kojim se hoće upravljati te vrijednost koja se želi postići. „WaitTime“ instrukcija služi smanjivanju mogućnosti pogreške. Na primjer da se robot zaustavi, to jest zaustavi sve motore, u potpunosti stane te tek onda krene sa prihvatom paketa. Na slici 6.14 je prikazano kako je potrebno definirati instrukciju čekanja te odabir vremena koje mora proći da bi se nastavila putanja robota.



Slika 6.14 Definiranje instrukcije čekanja vremena u trajanju od 1 sekunde

Nakon što su postavljene sve instrukcije, potrebno se vratiti u „*SmartComponent\_1*“ koji je prethodno kreiran. Potrebno je dodati signale na isti način na koji su bili rađeni i prije. Idući korak je povezivanje pojedinih komponenata, a to je potrebno učiniti kako je prikazano na slici 6.15.

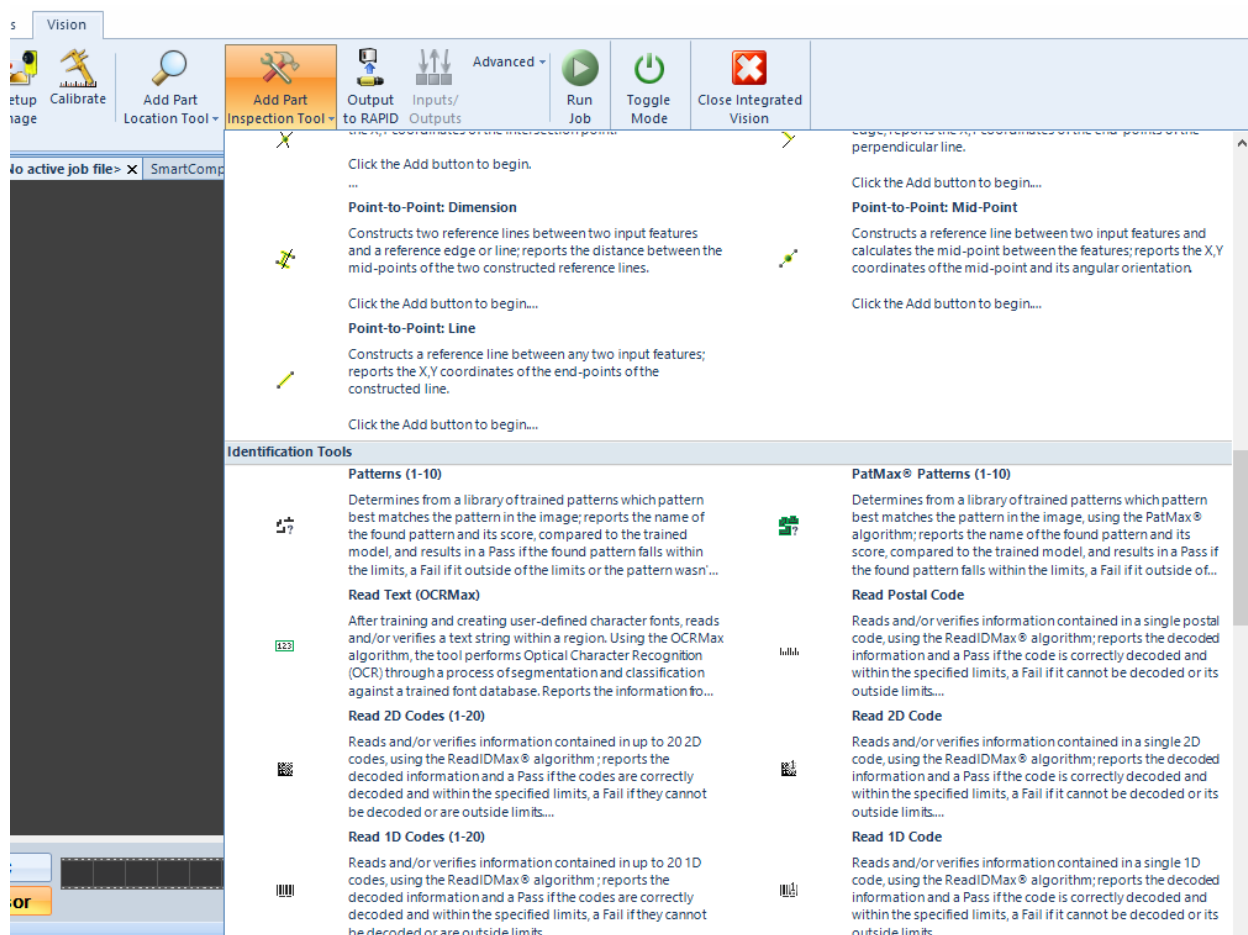


Slika 6.15 Dopuna *SmartComponenta\_1*

Kada je izvršeno povezivanje komponenata unutar „*SmartComponenta\_1*“, potrebno je dodatno kreirati izlazni signal (slika 6.16). Izlazni signal služi za dalje upravljanje sustava. Taj signal će služiti kretanju kamere u rad te skeniranju paketa koji se nalazi ispod nje.

Slika 6.16 Definiranje izlaznog signala *SmartComponenta\_1*

Sljedeći korak je dodavanje zadatka kameri. Kao što je ranije spomenuto u tekstu, nažalost zbog korištenja dodatnog programa za simuliranje kamere, nije dostupna opcija „*Inputs/Outputs*“. Zbog toga su se ranije kreirali signali koji će simulirati signale skeniranja kamere. No svejedno će se u daljnjem dijelu pokazati kako odabrati alat za usporedbu slike QR koda. Pod karticom „*Vision*“ odabire se opcija „*Add Part Inspection Tool*“. Tamo se odabire zadatak kamere. Opciju koju je potrebno odabrati je „*Patterns*“ (1-10)“ (slika 6.17).



Slika 6.17 Odabir alata za prepoznavanje slike QR koda

Zatim se ubacuje slika koju se želi naučiti kameru. To se izvodi na način da se odabire opcija „*Load Image*“. Dimenzije slike moraju biti 800x600 piksela kako bi se u potpunosti vidjela. Zatim je potrebno raširiti rozi kvadrat koji se nalazi na slici QR koda do punog raspona. Tada je potrebno potvrditi na OK. Zatim se odabire naziv zadatka, u ovom slučaju to je QR kod „*paketa\_vz*“ pa se zato naziva „*scan\_vz*“. Zeleno svijetlo u gornjem desnom kutu slike 6.18 prikazuje signal da se naučena te postavljena slika podudaraju.



Slika 6.18 Definiranje zadatka kamere

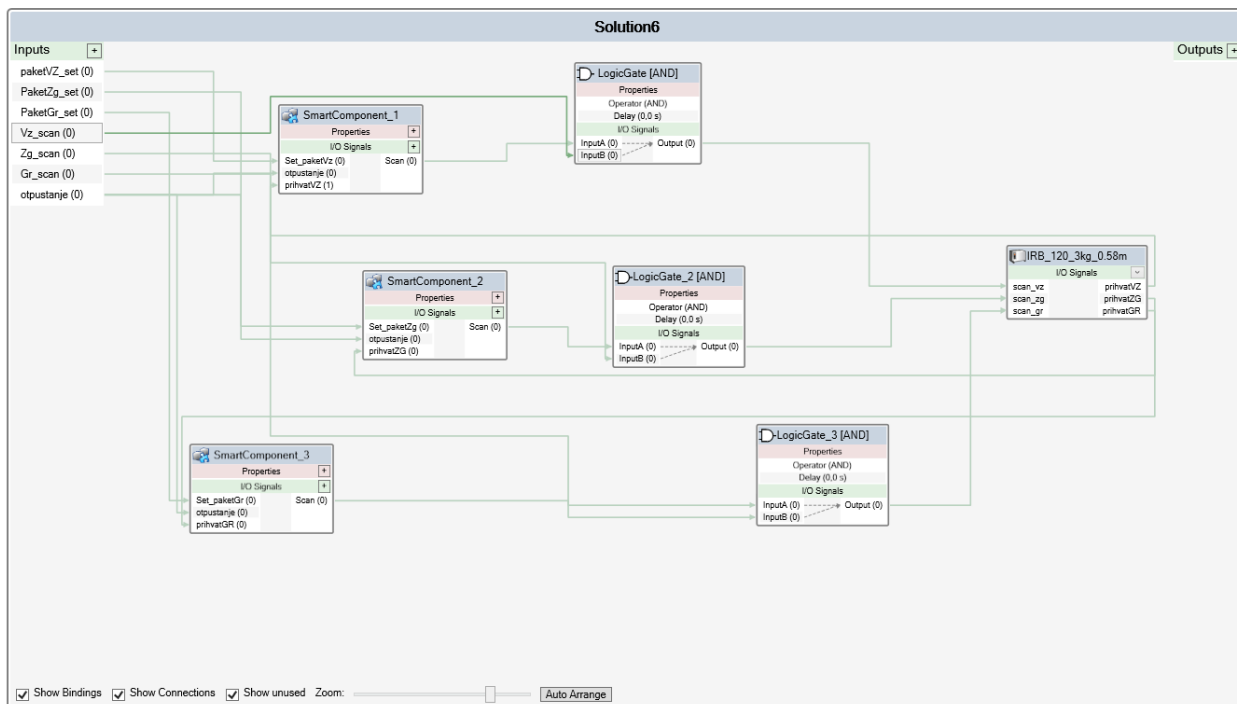
Na slici 6.19 u gornjem desnom kutu vide se izlazni signali skeniranja. Trenutno je slika QR koda „paket\_vz“. Na slici se vidi da je prepoznat paket koji treba ići u prostor za odlaganje vz.



Slika 6.19 Definirani zadaci kamere

Kada su svi parametri definirani, potrebno je odabrati karticu „Simulation“ te opciju „Station Logic“. Potrebno je kreirati komponente kao što su bile kreirane prethodno u „SmartComponent“. Nakon što su sve komponente kreirane, potrebno ih je spojiti prema slici 6.20. Kada se spoje komponente prema slici može se krenuti sa simulacijom sustava koji je kreiran.





Slika 6.20 Definiranje veza unutar Station Logich

Prethodno kreirane signale je potrebno urediti kao što je prikazano na slici 6.21. Na isti način je potrebno formirati putanje i za ostale pakete.

- ⚡ WaitDI start,1
- ➡➡ MoveJ Target\_10
- ➡➡ MoveJ Target\_30
- ➡➡ MoveJ Target\_40
- ⚡ WaitTime 1
- ⚡ Set prihvatVZ
- ⚡ WaitTime 2
- ➡➡ MoveJ Target\_30
- ➡➡ MoveJ Target\_10
- ➡➡ MoveJ Target\_20\_k
- ⚡ WaitDI scan\_vz,1
- ➡➡ MoveJ Target\_10
- ➡➡ MoveJ Target\_50
- ➡➡ MoveJ Target\_60
- ⚡ WaitDI otpustanje,1
- ⚡ Reset prihvatVZ
- ➡➡ MoveJ Target\_50
- ➡➡ MoveJ Target\_10

Slika 6.21 Signali unutar putanje paketa vz

## 7. Zaključak

Izradom ove simulacije sustava sortiranja paketa pomoću čitanja QR koda s kamerom robotskog sustava proučen je veliki spektar alata unutar RobotStudio-a. U ovome radu je opisan svaki korak od kreiranja radnog okruženja, impelmetiranje pojedinih objekata unutar i izvan softvera RoboStudio. Pokazano je kako kreirati željene točke putanje te zatim kako od zadanih točaka kreirati putanje robota. Najbitniji dio koji se prikazuje u radu je kreiranje „*SmartComponent*“ objekata, povezivanje vanjskog softvera za simuliranje kamere. Najkompleksniji dio je logika iza kreiranja signala koji su potrebni za upravljanje sustavom te izvođenje signala unutar „*Station Logic*“ i „*SmartComponent*“ objekata. Zbog toga je bilo potrebno svaki korak detaljno opisati kako bi bilo što jasnije koji se alat koristi u određenom trenutku i koja je njegova funkcija. Također je pokazano kako se mogu skicirati te izraditi 3D modeli unutar softvera *SolidWorks*. Potrebno je puno vremena za izradu ovakvog jednostavnog sustava unutar simulacije. Zato je potrebno da se sustavi kreiraju softverski kako bi se stvarna izrada sustava odradila što brže i efikasnije. Upravo ovakvim kreiranjem se uveliko štedi na vremenu te financijama kupcima sustava. Ova simulacija nije odrađena savršeno, a to nije bio ni cilj. Cilj ove simulacije je bio da se pokaže kako se u današnje vrijeme izrađuju sustavi industrije 4.0.

Može se reći da je i ovo sustav industrije 4.0 jer se unutar ove simulacije nalazi praktički gotov sustav za sortiranje. Upravo ovom simulaciju dobivene su putanje, vremena, signali, upravljanje signalima, gotovo sve komponente sustava. Naravno uvijek ima mjesta za optimizaciju sustava.

## 8. Literatura

Internet izvori:

[1] Product specification - IRB 120, dostupno 20.04.2024

[2] Product manual – IRB 120, dostupno 20.04.2024

[3] <http://www.unin.hr>, dostupno 09.06.2015.

[4] In-Sight® 7000 Series Vision System Installation Manual - Vision System Specifications - Documentation | Cognex dostupno 05.09.2024

## Popis slika

Slika 2.1 Prikaz robota sa označenim osima (stupnjevi slobode gibanja).[1] .....	2
Slika 2.3 IRC 5 upravljačka jedinica .....	3
Slika 3.1. Kreiranje novog dijela (Part) .....	6
Slika 3.2 Odabir kartice „Sketch“ i odabir alata „Sketch“ u gornjem lijevom kutu.....	7
Slika 3.3 Odabir ravnine za kreiranje skice „Top Plane“ .....	7
Slika 3.4 Odabir opcije za skiciranje pravokutnika te njegovo definiranje.....	8
Slika 3.5 Odabir opcije „Extruded Boss/Base“ za kreiranje tijela pravokutnika .....	8
Slika 3.6 Odabir opcije „appearance“ te dodavanje izvora za uzorke.....	9
Slika 3.7 Odabir plohe tijela koje se želi implementirati - slika iz izbornika.....	9
Slika 3.8 Objekt „Paket s greškom“ .....	10
Slika 3.9 Spremnik za odlaganje paketa .....	10
Slika 3.10 Klizna staza .....	10
Slika 4.1 Početni izbornik RobotStudio-a .....	11
Slika 4.2 Implementacija komponenata.....	12
Slika 4.3 Pomicanje komponente IRC5.....	12
Slika 4.4 Implementacija objekata izvan RobotStudio-a.....	13
Slika 4.5 Pomicanje objekta pomoću opcije Position.....	13
Slika 4.6 Koordinatni sustav za pomicanje objekta.....	14
Slika 4.7 Dodavanje robotske prihvatnice .....	14
Slika 4.8 Spajanje robotske prihvatnice na robot .....	15
Slika 4.9 Sklop robot – alat (robotska prihvatnica) .....	15
Slika 4.10 Radni prostor simulacije.....	16
Slika 4.11 Sakrivanje objekta zbog preglednosti.....	16
Slika 4.12 Radni prostor nakon sakrivanja držača kamere.....	17
Slika 4.13 Odabir kretnje robota.....	18
Slika 4.14 Kreiranje prazne putanje .....	18
Slika 4.15 Freehand sekcija alata za pomicanje robota .....	18
Slika 4.16 Kreirane točke putanje.....	19
Slika 4.17 Putanje pojedinog paketa.....	19
Slika 4.18 Kreiranje kontrolera za kameru .....	20
Slika 4.19 Izbornik za kreiranje virtualnog kontrolera za kameru .....	21
Slika 4.20 Upisivanje imena novog virtualnog kontrolera .....	21
Slika 4.21 Odabir alata kojeg će novi virtualni kontroler upravljati .....	22

Slika 4.22 Odabir dodatne opcije „Integrated Vision Interface“ .....	22
Slika 4.23 Pokretanje virtualnog kontrolera za kameru .....	23
Slika 4.24 Odabir virtualnog kontrolera .....	23
Slika 4.25 Pokretanje kamere .....	24
Slika 4.26 Dodatni softver za simuliranje kamere „In-Sight Explorer“ .....	24
Slika 4.27 Početni izbornik unutar „In-Sight Explorer-a“ .....	25
Slika 4.28 Opcija „refresh Cameras“ .....	25
Slika 4.29 Virtualna kamera .....	25
Slika 4.30 Virtualna kamera unutar RobotStudio-a.....	26
Slika 6.1 Stvaranje „Smart Component“ objekta .....	28
Slika 6.2 Stvaranje ulaznog signala koji će dopremiti paket na kliznu stazu .....	28
Slika 6.3 Izbornik za definiranje parametara ulaznog signala.....	29
Slika 6.4 Stvaranje izvora paketa .....	29
Slika 6.5 Definiranje parametara „Source“ .....	30
Slika 6.6 Definiranje parametara „Attacher“ .....	30
Slika 6.7 Otvaranje ulazno izlaznih signala kontrolera .....	31
Slika 6.8 Dodavanje novog signala u kontroleru.....	31
Slika 6.9 Definiranje signala koji će pokrenuti gibanje robota .....	32
Slika 6.10 Definiranje signala za prihvata paketa .....	33
Slika 6.11 Definiranje signala za otpuštanje paketa .....	33
Slika 6.12 Opcija „Insert Action Instruction“ .....	34
Slika 6.13 Definiranje instrukcije unutar putanje robota.....	35
Slika 6.14 Definiranje instrukcije čekanja vremena u trajanju od 1 sekunde.....	35
Slika 6.15 Dopuna SmartComponenta_1 .....	36
Slika 6.16 Definiranje izlaznog signala SmartComponenta_1 .....	36
Slika 6.17 Odabir alata za prepoznavanje slike QR koda.....	37
Slika 6.18 Definiranje zadatka kamere .....	38
Slika 6.19 Definirani zadaci kamere.....	38
Slika 6.20 Definiranje unutar Station Logich .....	39
Slika 6.21 Signali unutar putanje paketa vz .....	39

—  
HARON  
ALISWARINO  
—

Sveučilište  
Sjever

+

—  
SVEUČILIŠTE  
SJEVER  
—

#### IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, TOMISLAV BABIĆ (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/~~ko~~ završnog/~~završnog~~ (obrisati nepotrebno) rada pod naslovom ČITANJE OS KUDA POMOĆU KLAMBE ROBOTIKOM GUSJAV (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica: TOMISLAV BABIĆ  
(upisati ime i prezime)

Tomislav Babić  
(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.