

# Izrada korisničkog sučelja za robot ABB IRB120

---

Rabić, Leon

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:122:493083>

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

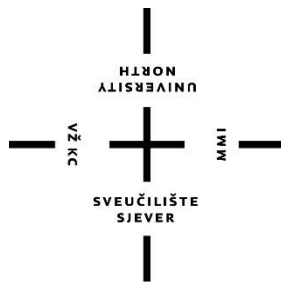
Download date / Datum preuzimanja: **2024-12-25**



Repository / Repozitorij:

[University North Digital Repository](#)





# Sveučilište Sjever

**Završni rad br. 036/MEH/2024**

## **Izrada korisničkog sučelja za robot ABB IRB120**

**Leon Rabić, 0336053374**

Varaždin, rujan 2024. godine



# Sveučilište Sjever

**Odjel za Mehatroniku**

**Završni rad br. 036/MEH/2024**

## **Izrada korisničkog sučelja za robot ABB IRB120**

### **Student**

Leon Rabić, 0336053374

### **Mentor**

Zoran Busija, dipl.ing.stroj.

Varaždin, rujan 2024. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za mehatroniku		
STUDIJ	prediplomski stručni studij Mehatronika		
PRISTUPNIK	Leon Rabić	JMBAG	0336053374
DATUM	09.09.2024.	KOLEGIJ	Robotika
NASLOV RADA	Izrada korisničkog sučelja za robot ABB IRB120		
NASLOV RADA NA ENGL. JEZIKU	Creation of a user interface for the ABB IRB120 robot		

MENTOR	Zoran Busija, dipl. ing. stroj.	ZVANJE	predavač
ČLANOVI POVJERENSTVA	1. Siniša Švogor, 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

BROJ	036/MEH/2024
------	--------------

OPIS

U završnom radu potrebno je:

- opisati ABB privjesak za učenje (FrexPendant).
- prikazati rad sa programom ScreenMaker.
- izraditi sučelje za igranje igre križić/kružić.
- postaviti novo sučelje na virtualni privjesak za učenje unutar softvera RobotStudio.

ZADATAK URUČEN

12.09.2024.



*Zoran Busija*

## **Predgovor**

Rad pod nazivom Izrada korisničkog sučelja za robot ABB IRB120 odabrao sam prema vlastitim interesima i zahvalan sam prof. Zoranu Busiji, dipl.ing.stroj., što mi je svojom podrškom i stručnim znanjem, omogućio bavljenje ovom temom. Početak interesa javio se tijekom seminarskoga rada iz robotike, a nadogradnjom zadane teme i dodatnim istraživanjem, želio sam proširiti svoje znanje te se ostvariti u navedenom području. Nadam se da će ovaj rad pomoći učenicima, studentima i drugima koje zanima ova tema, brže i lakše upoznati sustav prilikom rada sa softverom *ScreenMaker*.

Nadam se kako će moj rad biti novi izvor znanja kao i motivacija budućim studentima za upuštanje u slična istraživačka putovanja.

Također, želim izraziti zahvalnost svojoj obitelji na podršci tijekom cijelog akademskog obrazovanja. Njihovo povjerenje bilo mi je poticaj za uspješno svladavanje svih izazova na koje sam nailazio.

## Sažetak

U ovom završnom radu koncipiran je i sažet postupak korištenja i upravljanja uređajem *FlexPendant* unutar programa *RobotStudio*, kako bi rad mogao poslužiti kao priručnik za buduće korisnike. Rad obuhvaća pojednostavljeni postupak manevriranja kroz sustav, najvažnije značajke, funkcije i mogućnosti, kao i postupak povezivanja i prebacivanja *ScreenMaker-a* na *FlexPendant*. Također, objašnjen je postupak instalacije *ScreenMaker-a* za uređivanje zaslona te korištenje njegovih alata radi lakšeg upravljanja robotom putem *FlexPendant-a*, ovisno o zadatku. Korišteni su 3D modeli iz drugog završnog rada. Rezultati pokazuju uspješnu implementaciju *ScreenMaker-a* na *FlexPendant* i simulaciju upravljanja robotom u radnoj okolini.

Ključne riječi: *FlexPendant*, *ScreenMaker*, IRB120, IRC5 Compact

## **Summary**

In this final thesis, a concise procedure for using and managing the FlexPendant device within the RobotStudio software has been designed, making the work a potential manual for future users. The thesis covers a simplified process of navigating through the system, the most important features, functions, and capabilities, as well as the procedure for connecting and transferring ScreenMaker to FlexPendant. Additionally, the installation process of ScreenMaker for screen editing is explained, along with the use of its tools to facilitate easier robot control via FlexPendant, depending on the task. 3D models from another final thesis were used. The results demonstrate the successful implementation of ScreenMaker on FlexPendant and the simulation of robot control in a working environment.

Key words: FlexPendant, ScreenMaker, IRB120, IRC5 Compact

## Popis korištenih kratica

<b>GUI</b>	Graphical User Interface Grafičko korisničko sučelje
<b>ABB</b>	ASEA Brown Boveri Švicarsko-Švedska kompanija sa sjedištem u Zurichu
<b>IRC5</b>	Industrial Robot Controller Robotski kontroler za industriju pete generacije
<b>SDK</b>	Software Development Kit Skup alata za razvijanje softvera
<b>HMI</b>	HumanMachineInterface Komunikacija između čovjeka i stroja
<b>.NET</b>	Free, open-source and cross-platform developer platform Besplatna platforma za razvojne programere otvorenog koda i više platformi



# Sadržaj

1.	Uvod.....	1
2.	RobotStudio .....	2
2.1.	Povijest .....	2
2.2.	Ključne značajke IRC5 kontrolera .....	2
2.3.	FlexPendant SDK.....	3
3.	ScreenMaker .....	4
3.1.	Instalacija ScreenMaker-a .....	4
3.2.	Pokretanje ScreenMakera.....	6
3.3.	Razvojno okruženje.....	7
3.3.1.	<i>Dijelovi razvojnog sučelja.....</i>	8
3.3.2.	<i>Controller.....</i>	9
3.4.	Dizajniranje korisničkog sučelja .....	10
3.4.1.	<i>ActionTrigger .....</i>	10
3.4.2.	<i>Button.....</i>	10
3.4.3.	<i>NumericUpDown.....</i>	11
3.4.4.	<i>RunRoutineButton .....</i>	12
4.	Izrada sučelja i programa za Križić/kružić .....	13
4.1.	Opis problema i zahtjevi .....	13
4.2.	Korišteni 3D modeli i alat .....	13
4.3.	Workobjects & Targets .....	14
4.4.	Putanje i procedure (engl. <i>Paths &amp; Procedures</i> ).....	15
4.5.	Pametne komponente (engl. <i>SmartComponent</i> ).....	16
4.5.1.	<i>Korištene pametne komponente SmartComponenti.....</i>	16
4.5.2.	<i>Oblikovanje pametene komponente Desing.....</i>	17
4.6.	Digitalni signali .....	18
4.7.	Virtualno povezivanje .....	19
4.8.	RAPID .....	20
4.9.	Primjena ScreenMaker-a .....	21
4.9.1.	<i>Omogućavanje RunRoutineButton alata .....</i>	21
4.9.2.	<i>Dizajniranje sučelja za igru .....</i>	22
4.9.3.	<i>Podšavanje korištenih alata .....</i>	23
4.10.	Učitavanje programa na FlexPendant-u .....	24
5.	Zaključak.....	25
6.	Literatura.....	26
7.	Popis slika.....	27

# 1. Uvod

RobotStudio je najpopularniji svjetski offline programski i simulacijski alat za robotske aplikacije. To je softver koji omogućava programiranje putanje rada robota, za ABB robote.

Računalna verzija RobotStudio-a omogućuje izvođenje programiranja i simulacija bez ometanja tekuće proizvodnje. RobotStudio Cloud je zasebni program koji omogućuje pojedincima i timovima suradnju u stvarnom vremenu na dizajniranju robotske ćelije s bilo kojeg mjesta u svijetu, na bilo kojem uređaju. RobotStudio Augmented Reality Viewer zasebni je program koji omogućuje besplatnu vizualizaciju robota i rješenja u stvarnom okruženju ili u virtualnoj sobi na bilo kojem mobilnom uređaju. Računalne i mobilne aplikacije omogućuju timovima suradnju i brže donošenje odluka, ubrzavajući fazu planiranja i dizajna.

*ScreenMaker* je alat u RobotStudio koji se koristi za razvoj prilagođenih zaslona na privjesku za učenje kojeg ABB zove *FlexPendant*. Riječ *FlexPendant* biti će korištena umjesto riječi privjesak za učenje. *ScreenMaker* omogućuje stvaranje prilagođenog *FlexPendant*-a GUI-ja bez potrebe za učenjem Visual Studio razvojnog okruženja i .NET programiranja. Prilagođeno operatersko sučelje u tvornici ključ je jednostavnog robotskog sustava. Dobro dizajnirano prilagođeno operatersko sučelje predstavlja pravu količinu informacija u pravo vrijeme i u pravom formatu za korisnika, a vrijeme zastoja (zbog pogrešaka u radu) je minimalno. Međutim, prilagođena korisnička sučelja su skupa i potreban im je vrlo dugotrajan razvoj. Za sada je potrebno razumijevanje nekih objektno orijentiranih programskih jezika (kao što su C, C++; VB i C#) i razvojni okvir (.NET, Visual Studio) za razvoj zaslona. To je posao za IT profesionalce, a ne za industriju robotike čiji su radnici navikli na jednostavne programske jezike kao što je RAPID; koji se koristi unutar *ScreenMaker*-a.

Unutar rada je prikazana i objašnjena instalacija programa *ScreenMaker*, korištenje i povezivanje sa uređajem *FlexPendant*. Prikazan je rad s programom *ScreenMaker* i izrada sučelja za igranje igrice Križić/kružić.

## 2. RobotStudio

RobotStudio je najpopularniji svjetski offline programski i simulacijski alat za robotske aplikacije. To je softver koji omogućava programiranje putanje rada ABB robota. Ovaj alat omogućava projektiranje grafike i simulaciju kretanja robota, što ga čini idealnim za planiranje i izradu robotskih sustava. Temeljen na tehnologiji virtualnog upravljača, paket RobotStudio daje mogućnost da ono što je vidljivo na ekranu, odgovara načinu na koji će se robot kretati u stvarnom životu. Omogućava izgradnju, testiranje i poboljšanje svoje instalacije robota u virtualnom okruženju. Ova tehnologija značajno ubrzava vrijeme puštanja u pogon i povećava produktivnost u radu.



*Slika 2.1-1 Logo RobotStudio programa*

### 2.1. Povijest

Godine 1990. ABB je ušao na američko tržište robota kupovinom tvrtke Cincinnati Milacron. To je omogućilo ABB-u proširivanje svojega poslovanja u automobilsku industriju sa svojim robotima za točkasto zavarivanje. Godine 1991. razvijen je IRB 6000, koji je postao najprecizniji i najbrži industrijski robot za točkasto zavarivanje. Tijekom 1994. ABB je predstavio nasljednika svog robotskog kontrolera S2, S4. Nekoliko godina kasnije, 1998., razvili su brzi delta robot *FlexPicker* za automatizirano uzimanje i robotsko pakiranje. Iste godine su napravili revoluciju u robotskom programiranju sa svojim softverom RobotStudio koji je omogućio simulaciju izvan mreže. Tijekom 2000-ih ABB je značajno proširio robotsku automatizaciju izdavanjem nekoliko svojih najuspješnijih modela. Godine 2004. uveden je IRC5 kontroler s mogućnošću upravljanja s četiri ABB robota.

### 2.2. Ključne značajke IRC5 kontrolera

Najvažnije značajke su maksimizacija produktivnosti, a samim tim i programiranje i testiranje unutar 3D okruženja koje odgovara stvarnome, bez ometanja proizvodnje. Također, naglašena je fleksibilnost proizvodnje koja omogućuje brzo planiranje novih robotskih rješenja. Time se ubrzava radni proces i prilagođava postojeća proizvodnja za obavljanje novih zadataka.

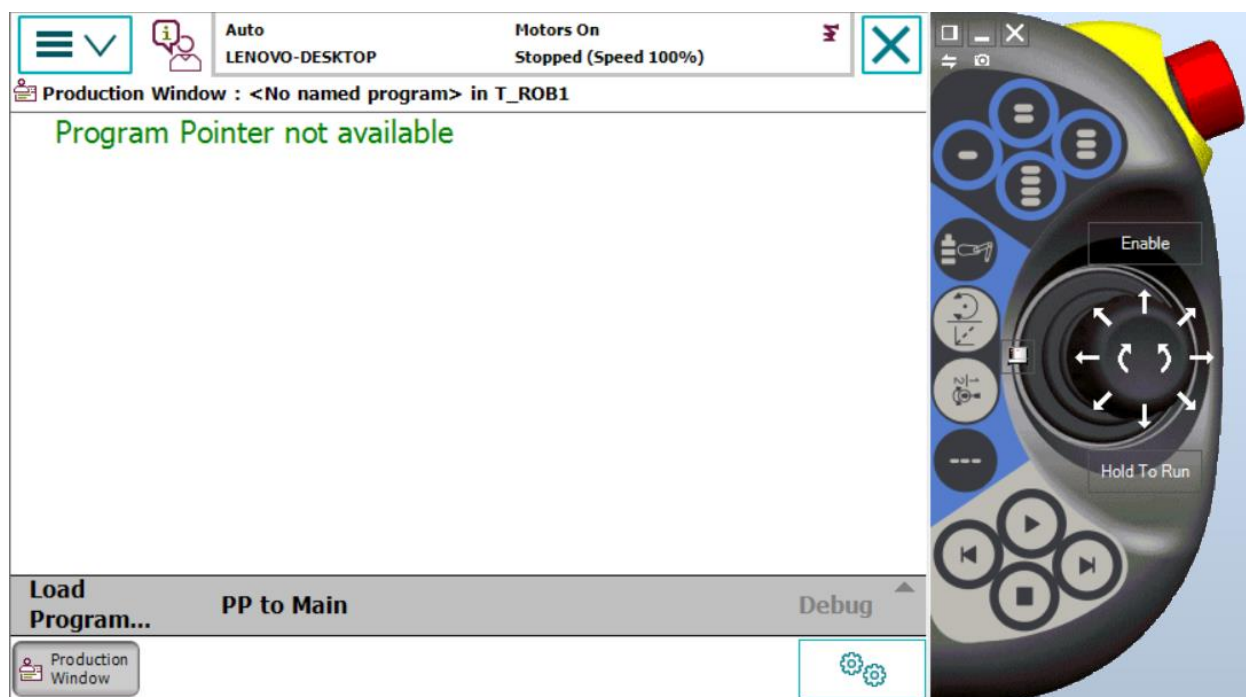
Jedan od najvažnijih ciljeva je potpuno eliminiranje zastoja, uz značajno smanjenje vremena potrebnog za pokretanje sustava, koje se skraćuje s nekoliko dana na samo nekoliko sati.

Simulacijom procesa omogućuje smanjenje potrošnje energije i minimiziranje gubitaka već od samog početka održavanja sustava.

### 2.3. FlexPendant SDK

Roboti se obično isporučuju s općim operaterskim sučeljem. Međutim, drugačiji procesi zahtijevaju drugačije rukovanje operatera, a klijenti trebaju fleksibilna rješenja gdje je korisničko sučelje prilagođeno specifičnim potrebama korisnika. *FlexPendant SDK* omogućuje integratorima sustava i korisnicima dodavanje svojih vlastitih prilagođenih operaterskih sučelja za IRC5 kontroler. Takve prilagođene aplikacije mogu se dodati standardnim prikazima *FlexPendant-a*.

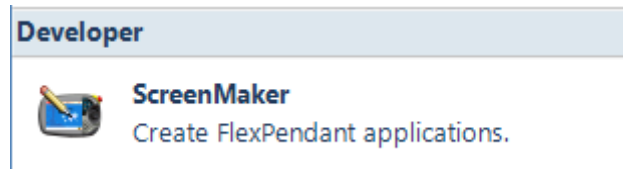
Unutar programa RobotStudio nalazi se virtualno sučelje kao simulacija *FlexPendant-a* koja služi kao komunikacija između čovjeka i stroja. Korištenjem kontrolera korisnik može brže i jednostavnije upravljati i nadzirat rad robota ili stroja, ali može nadzirati i rad cijelog pogona. Najbitnije je to što korisnici mogu upravljati i nadzirati rad robota neovisno o kontroleru, što pruža veliku fleksibilnost u radu.



Slika 2.3-1 Simulacija FlexPendant unutar programa RobotStudio

### 3. ScreenMaker

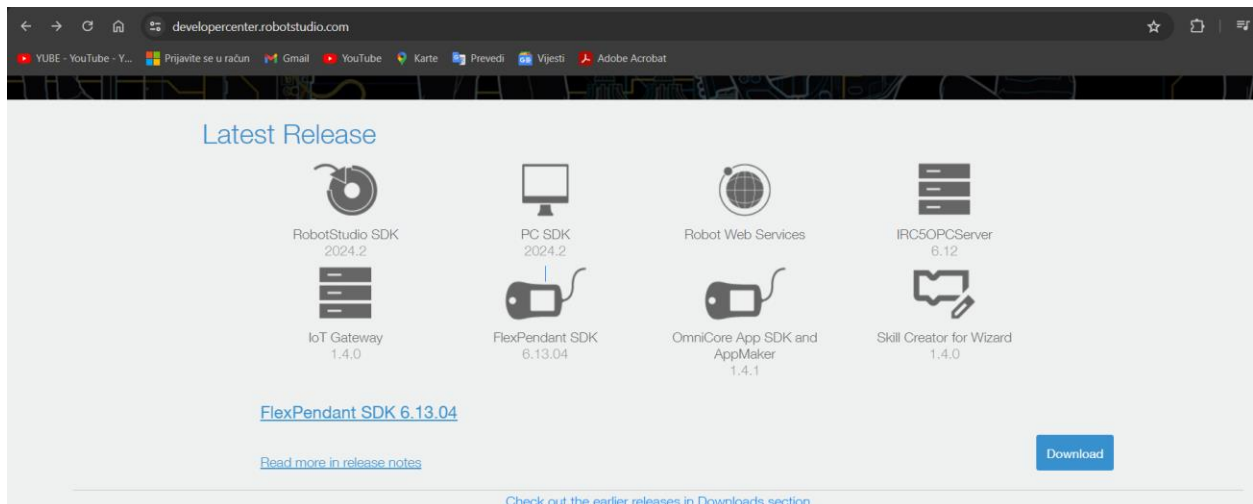
*ScreenMaker* unutar RobotStudija omogućuje kreiranje prilagođenih GUI zaslona za *FlexPendant* bez potrebe za učenjem Visual Studija i .NET programiranja. Prilagođeno sučelje pojednostavljuje rad s robotskim sustavima pružajući točne informacije, čime se smanjuje vrijeme obuke i zastoja. Dok tradicionalni razvoj sučelja zahtijeva poznavanje jezika poput C, C++, i C#, *ScreenMaker* to pojednostavljuje i idealan je za industriju robotike koja koristi jednostavnije jezike poput RAPID jezika.



Slika 2.3-1 Aplikacija ScreenMaker softvera unutar programa RobotStudio

#### 3.1. Instalacija ScreenMaker-a

Za instalaciju je potrebno zadovoljiti softverske, hardverske zahtjeve i imati odgovarajući operacijski sustav. Prije samog instaliranja i korištenja potrebno ja provjeriti je li instalirana verzija RobotStudija sa Premium licencom i testirati Virtualni kontroler. *ScreenMaker* se instalira zasebno i ne instalira se zajedno sa RobotStudijem. Instalacijski paket koji je korišten skinut je sa stranice [ABB-developercenter.robotstudio.com](http://ABB-developercenter.robotstudio.com).



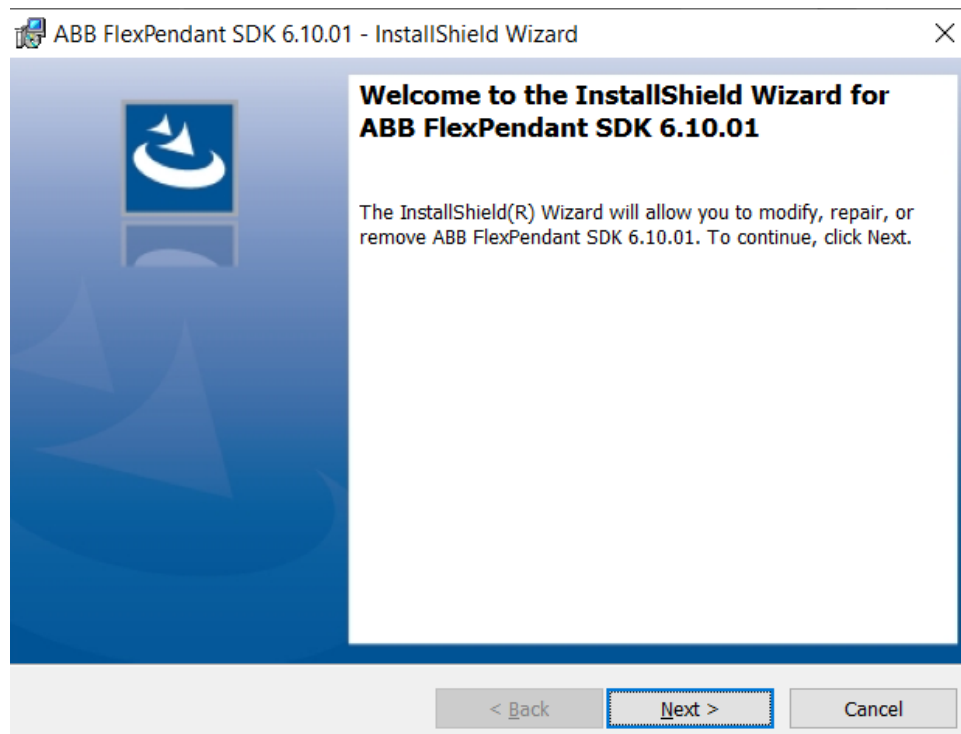
Slika 3.1-1 Stranica za preuzimanje instalacijskog paketa

Na izborniku *Latest Release* (Slika 3.1-1) odabrati *FlexPendant SDK*, pogledati ranija izdanja u odjeljku preuzimanja i odabrati onu verziju koja odgovara verziji RobotStudija. Instalirani paket raspakirati (Slika 3.1-2) i od Vrsta datoteke instalirati Paket *Windows Installer* i Aplikaciju.

Naziv	Datum izmjene	Vrsta	Veličina
0x0409	21.03.2016. 12:04	Postavke konfiguracije	22 KB
ABB FlexPendant SDK 6.10.01	06.12.2019. 23:58	Paket Windows Installer	896 KB
Data1	06.12.2019. 23:58	WinRAR archive	4.447 KB
setup	06.12.2019. 23:58	Aplikacija	1.067 KB
Setup	06.12.2019. 23:58	Postavke konfiguracije	6 KB

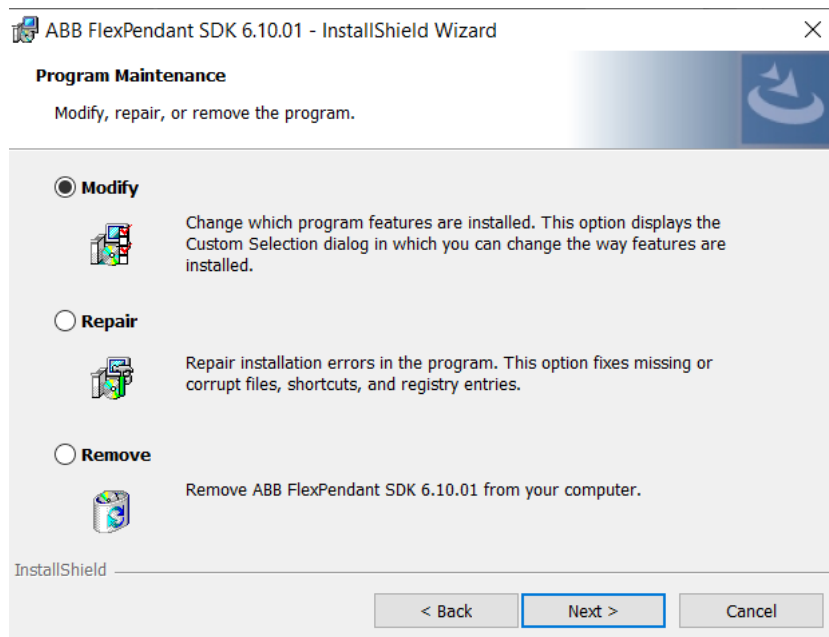
Slika 3.1-2 Datoteke unutar instalacijskog programa

Pokretanjem setup (Slika 3.1-2) otvara se instalacijski prozor (Slika 3.1-3) na kojemu odabiremo opciju *Next >*.



Slika 3.1-3 Instalacijski prozor

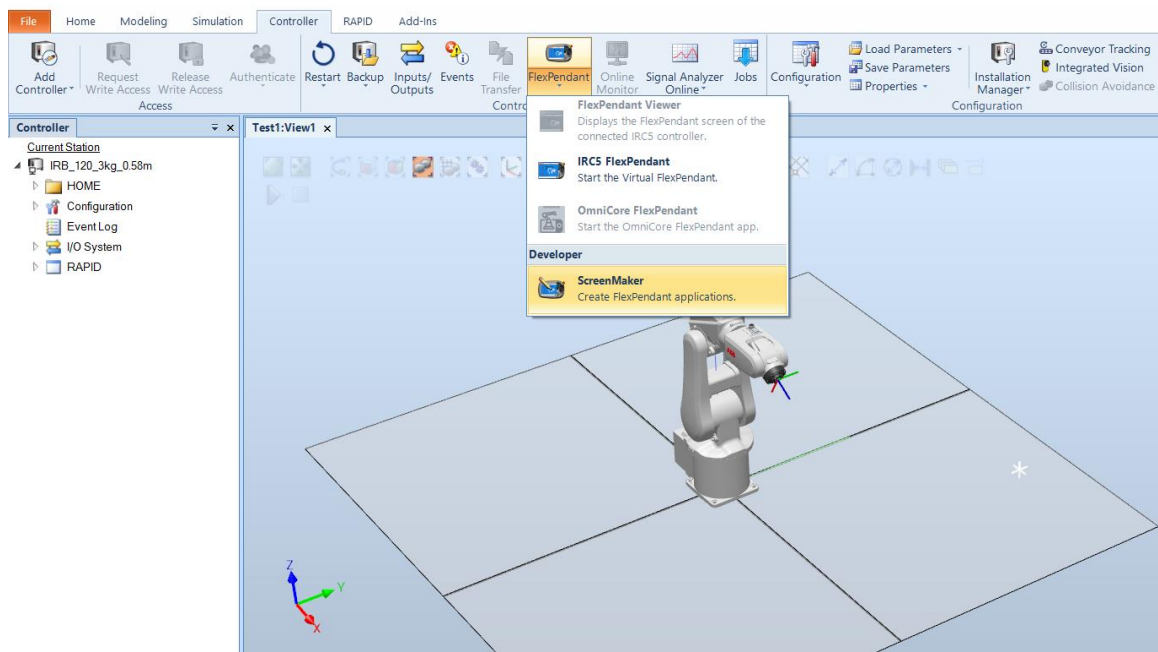
Otvora se sljedeći prozor (Slika 3.1-4) unutar kojega odabiremo Održavanje programa, gdje postoje opcije ovisno o tome što nam je potrebno; modificiranje, popravak ili deinstalacija programa. Prvo je potrebno odabrati *Modify*, tako da se modificiraju postojeće programske značajke, ovisno o tome što je instalirano, a ukoliko dođe do poteškoća tj. nedostatka neke datoteke, koristimo *Repair*. Opcija *Repair* popravlja instalacijske pogreške koje se mogu pojaviti poput, nedostajećih ili oštećenih datoteka. Odabirom alata završimo radnju i možemo pokrenuti RobotStudio.



Slika 3.1-4 Održavanje programa

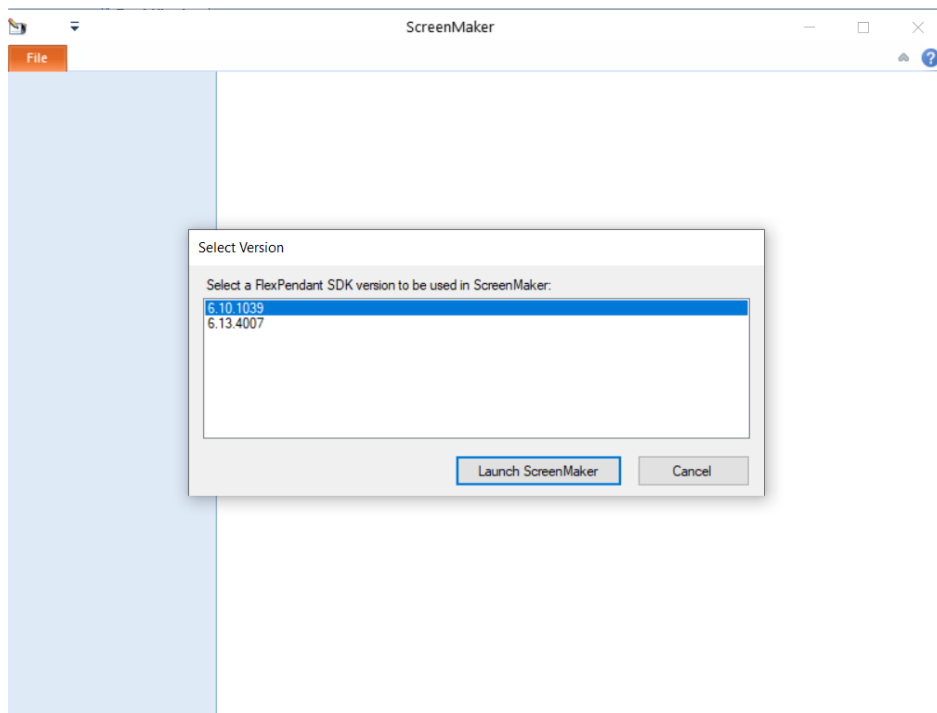
## 3.2. Pokretanje ScreenMaker-a

Otvaranjem programa RobotStudio, odabiremo karticu *Controller* i vrpcu *FlexPendant* unutar same vrpce pod odjelom *Developer* potrebno je odabrati *ScreenMaker* (Slika 3.2-1).



Slika 3.2-1 Pokretanje ScreenMaker-a

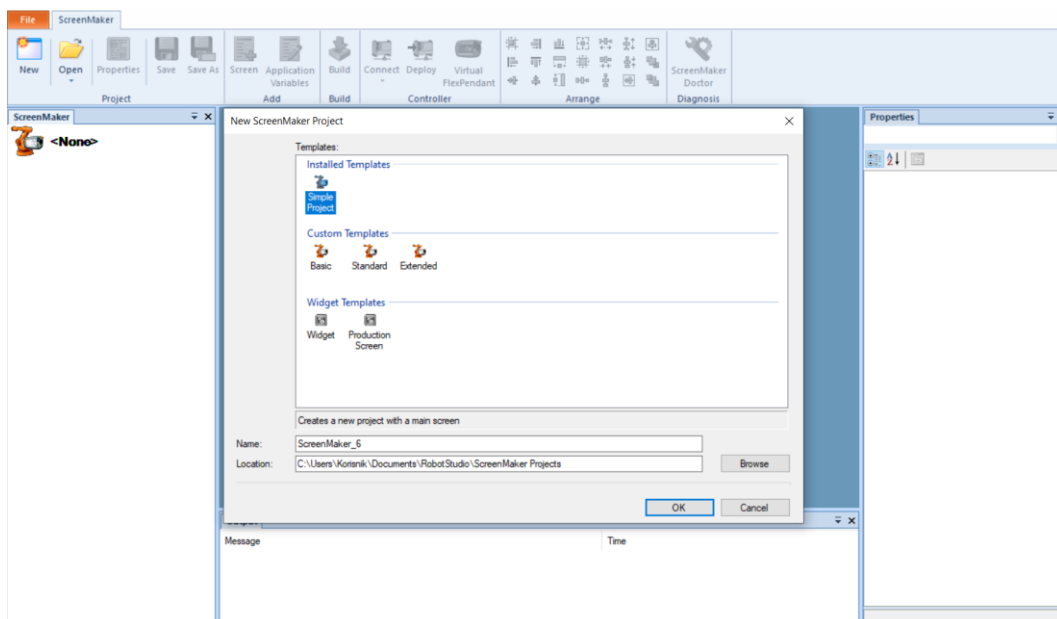
Prije samog pokretanja *ScreenMaker-a* potrebno je odabrati odgovarajuću verziju *FlexPendant-a SDK* koja odgovara verziji RobotStudija za korištenje *ScreenMaker-a* (Slika 3.2-2).



Slika 3.2-2 Odabiranje odgovarajuće verzije ScreenMaker-a

### 3.3. Razvojno okruženje

Pokretanjem razvojnog okruženja *ScreenMaker*, otvara se radni prostor unutar kojega je moguće odabrati novo ili postojeće sučelje. Otvaranjem novog projekta tj. sučelja možemo odabrati ona koja su već instalirana ili ako želimo svoje vlastito, potrebno je odabrati *Simple Project* (Slika 3.3-1) unutar kojega se dizajnira novo sučelje.

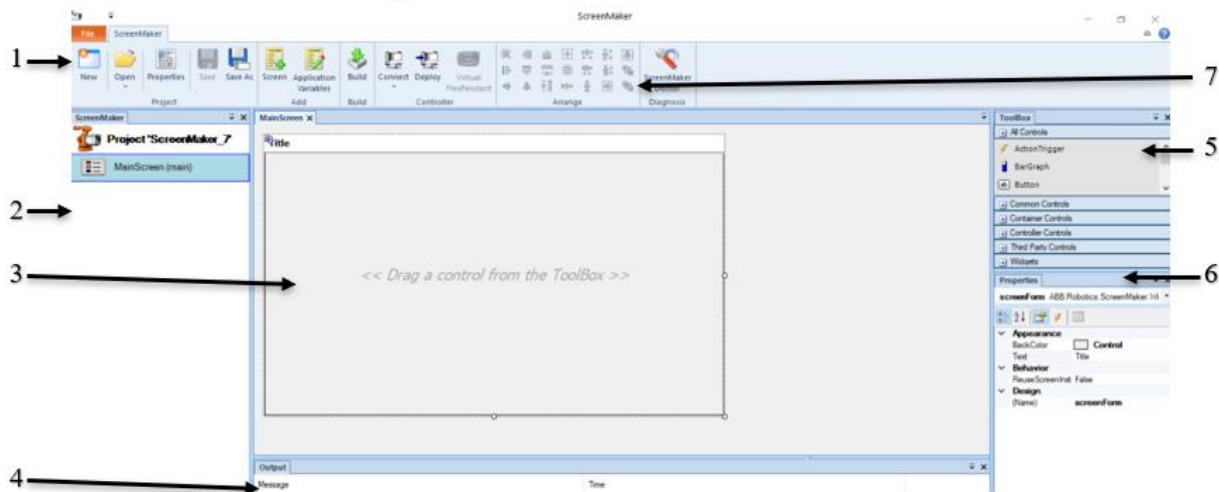


Slika 3.3-1 Razvojno okruženje unutar ScreenMaker-a



### 3.3.1. Dijelovi razvojnog sučelja

Prikaz svih 7 dijelova od kojih se sučelje sastoji (Slika 3.3-2) i opis istih (Tablica 3.1) kako bismo olakšali samo korištenje.



Slika 3.3-2 Dijelovi sučelja ScreenMaker

DIO	OPIS
1	Vrpca Prikazuje grupu ikona organiziranih u logičnom nizu Funkcije.
2	Prikaz projekta Prikazuje projekt aktivnog zaslona i navodi zaslone koji definirani su projektom.
3	Sučelje za dizajniranje Raspored za dizajn zaslona s dostupnim kontrolama.
4	Izlazni prozor Prikazuje informacije o događajima koji se događaju tijekom razvoja <i>ScreenMaker-a</i> .
5	Alati Prikazuje popis dostupnih kontrola.
6	Svojstva Sadrži dostupna svojstva i događaje odabrane kontrole. Vrijednost svojstava može biti fiksna vrijednost ili poveznica na IRC5 podatke ili aplikaciju varijabli.
7	Uređivanje Prikazuje ikone za promjenu veličine i pozicioniranje kontrola na dizajn površine.

Tablica 3.1 Popis dijelova ScreenMaker

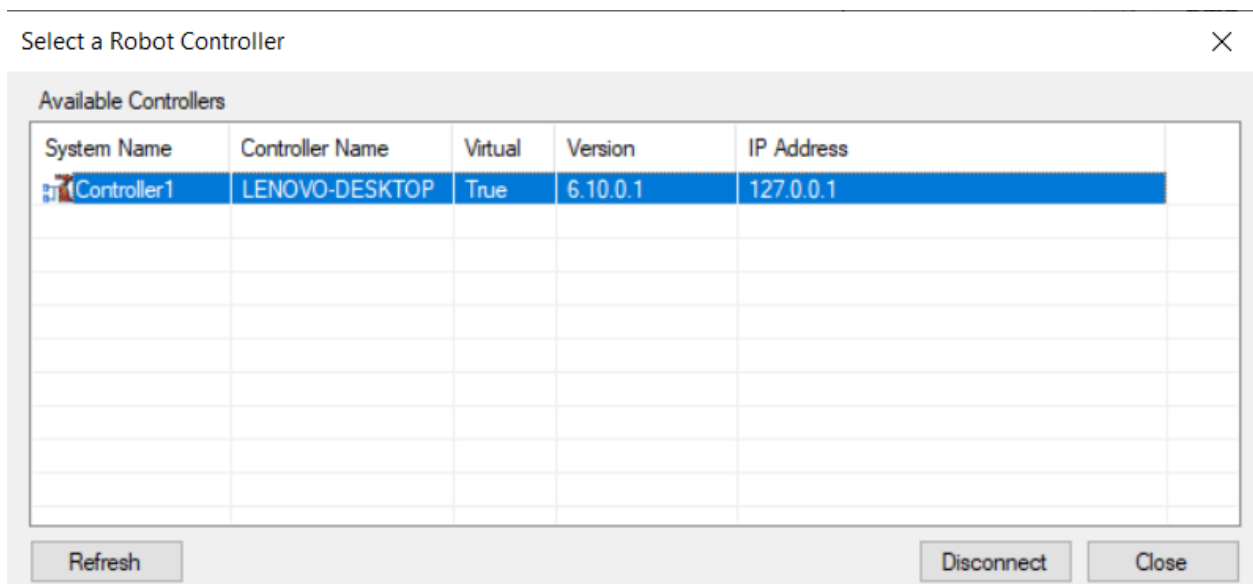
### 3.3.2. Controller

Prije samog uređivanja *ScreenMaker-a* potrebno se spojiti sa kontrolerom zbog same funkcionalnosti alata u programu. Unutar kartice *Controller* (Slika 3.3-3) nalazi se aplikacija *Connect* koja omogućava povezivanje GUI-a s kontrolerom robota, što omogućuje slanje naredbi, primanje podataka te testiranje i upravljanje robotom u stvarnom vremenu.



Slika 3.3-3 Kartica Controller

Pri povezivanju nam se otvara izbornik (Slika 3.3-4) za odabir odgovarajućeg kontrolera koji želimo povezati. Unutar tablice su nam prikazani svi dostupni kontroleri za povezivanje.



Slika 3.3-4 Odabir Robot Controller-a

## 3.4. Dizajniranje korisničkog sučelja

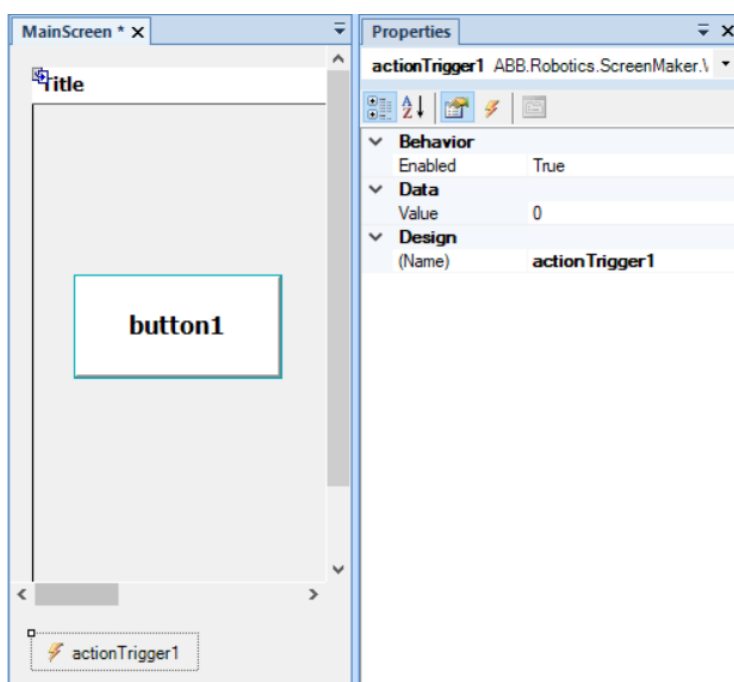
Ovaj odjeljak opisuje izgradnju GUI-ja korištenjem postojećih alata iz *ToolBox-a*.

### 3.4.1. ActionTrigger

Okidač radnje pokreće događaj, primjerice kada izvedena radnja putem kontrole učini skriveni objekt vidljivim. Omogućuje pokretanje popisa radnji kada se promijeni vrijednost svojstva.

Vrijednost svojstva može se vezati uz signal, brze podatke ili varijablu aplikacije.

Kontrola *ActionTrigger* također se može koristiti za pozivanje aplikacije iz RAPID-a.

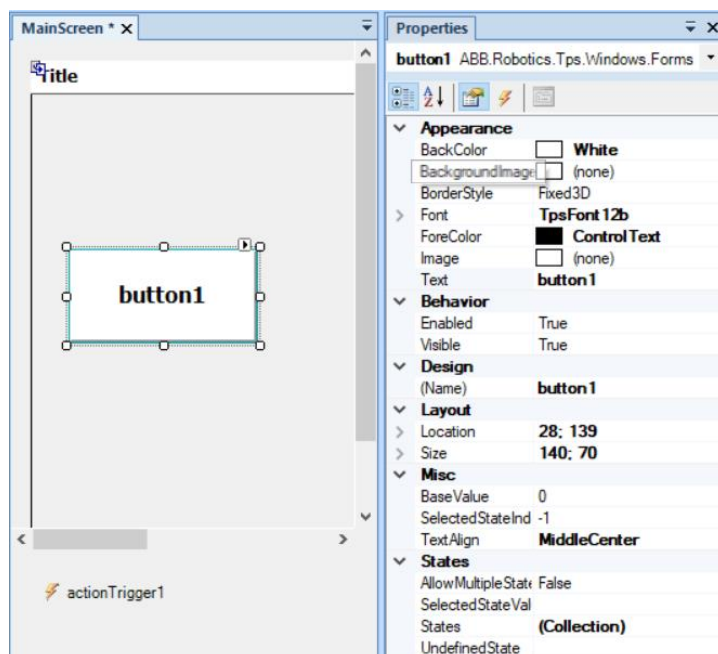


Slika 3.4-1 Prikaz alata *ActionTrigger*

### 3.4.2. Button

Alat *Button* koristi se za direktno izvršavanje naredbi prilikom aktiviranja ili deaktiviranja, ovisno o tome kako smo ga podesili. Aktiviranje tipke se postiže pritiskom na samu tipku, izlazni impuls prelazi iz nule u jedinicu i najčešće se izvršava naredba RAPID programa koja je povezana s tom tipkom. Uređivanje tipke se izvršava unutar trake *Properties* (Slika 3.4-2) gdje je moguće mijenjati njenu veličinu, položaj, tekst unutar tipke, boju, ponašanje prilikom aktiviranja.

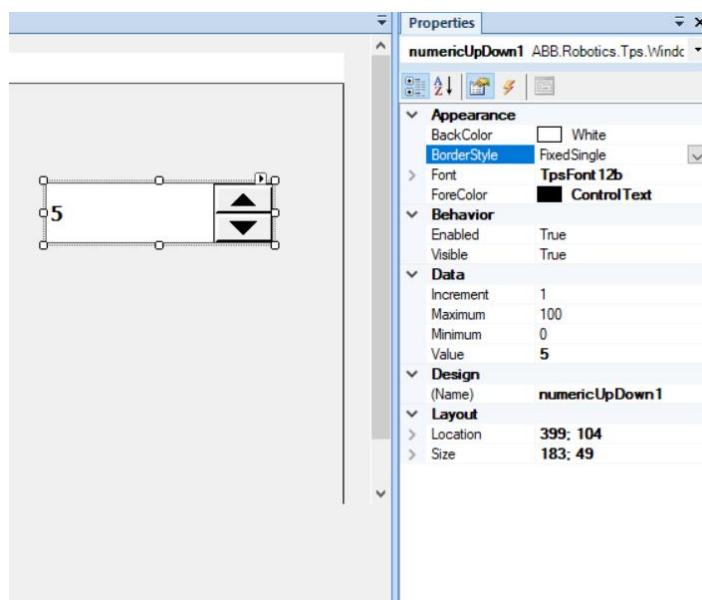
Pomoću tipke možemo se prebacivati sa zaslona na zaslon, a korištenjem više zaslona na *FlexPendant-u* olakšano nam je korištenje aplikacije zbog preglednosti zaslona.



Slika 3.4-2 Prikaz Button-Properties alata

### 3.4.3. NumericUpDown

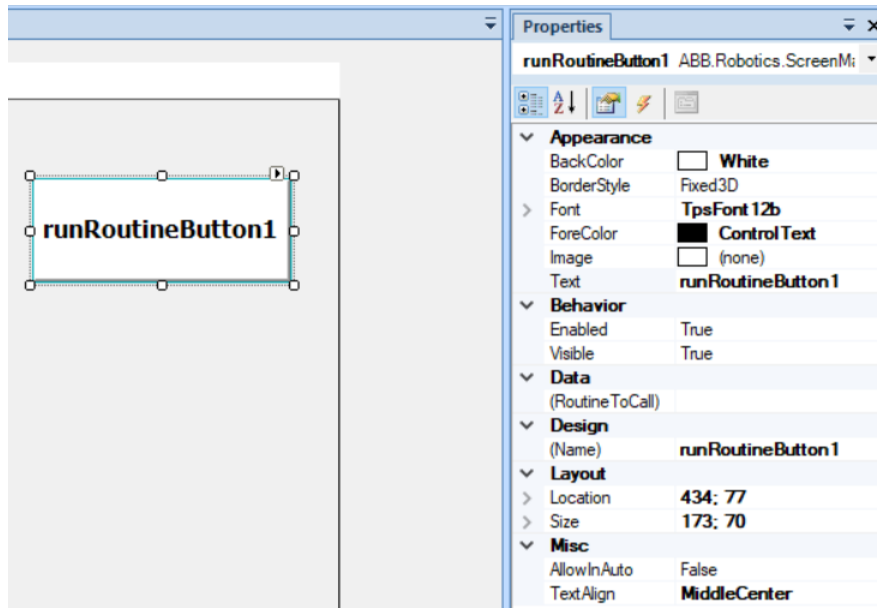
*NumericUpDown* se koristi kao ulazna funkcija pomoću koje određujemo broj ponavljanja koje želimo da robot izvrši prilikom pokretanja programa. Zadavanje broja ponavljanja nije potrebno u *ScreenMaker-u* jer je moguće pomoću strjelica mijenjati vrijednosti na *FlexPendant-u*.



Slika 3.4-3 Prikaz NumericUpDown alata

### 3.4.4. RunRoutineButton

*RunRoutineButton* je alat koji ima slična svojstva kao alat *Button*, ali glavna razlika je izvršavanje rutine. *RunRoutineButton* nije potrebno povezivati unutar RAPID koda kako bi program funkcionirao poput *Button* alata. Rutina može biti jednostavan pokret poput pomicanja robotske ruke gore-dolje u dvije točke, može biti složenija i sadržavati više točaka kretanja i signala poput; uključivanja trake, senzora, otvaranja/zatvaranja prihvatnice...



Slika 3.4-4 Prikaz *RunRoutineButton* alata

## 4. Izrada sučelja i programa za Križić/kružić

### 4.1. Opis problema i zahtjevi

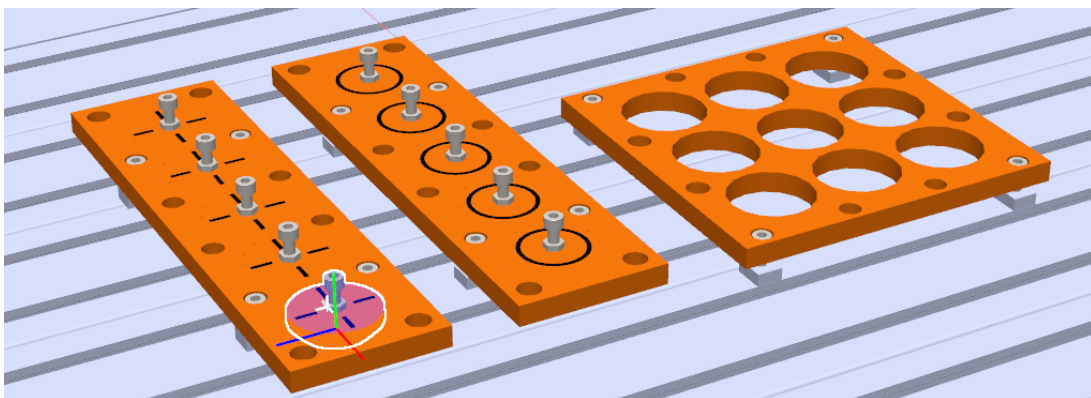
Cilj praktičnog dijela završnog rada je razviti računalnu verziju igre Križić/kružić s jednostavnim grafičkim sučeljem. Igra Križić/kružić je klasična igra za dva igrača, gdje svaki igrač naizmjenično postavlja svoj simbol (X ili O) na mrežu od 3x3 polja, s ciljem da prvi spoji tri simbola u nizu: horizontalno, vertikalno ili dijagonalno.

Osnovni zahtjevi za ovu implementaciju su:

- Grafičko sučelje: Treba omogućiti vizualnu reprezentaciju igre putem jednostavnog i preglednog korisničkog sučelja na uređaju *FlexPendant* preko kojega igrači unose potez.
- Interakcija korisnika: Igrači trebaju moći unositi svoje poteze klikom na polje u kojem se nalazi figura i odabrati željeno polje u mreži. Sučelje treba omogućiti prikaz simbola X i O tj. polja u kojima se figure nalaze, ovisno o igraču koji je na potezu.
- Ponovno pokretanje igre: Nakon završetka svake partije, korisnici trebaju posložiti se figure nazad na svoje mjesto, te nakon toga mogu započeti s novom partijom.

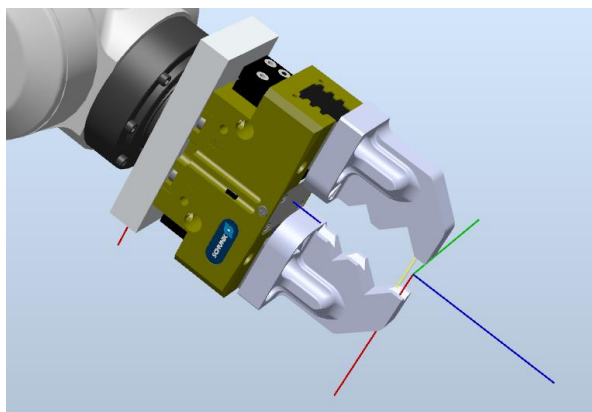
### 4.2. Korišteni 3D modeli i alat

Za razvojno okruženje unutar programa RobotStudio korišteni su posebno oblikovani 3D modeli objekata. Ovdje korišteni modeli su napravljeni u sklopu drugog završnog rada (Slika 4.2-1): figure X i O, postolje za figure X i O i postolje za igru.



Slika 4.2-1 Korišteni 3D modeli

Prezeti su i radni stol (RSTableIRB120UNIN) na koji su postavljeni svi modeli i korišteni robot IRB120 te je dodana prihvatnica (PrihvatnicaHanoi) (Slika 4.2-2) za alat. Pomoću prihvatnice moguće je hvatanje figura i njihova manipulacija u radnom prostoru.



Slika 4.2-2 Korišteni alat - prihvatnica

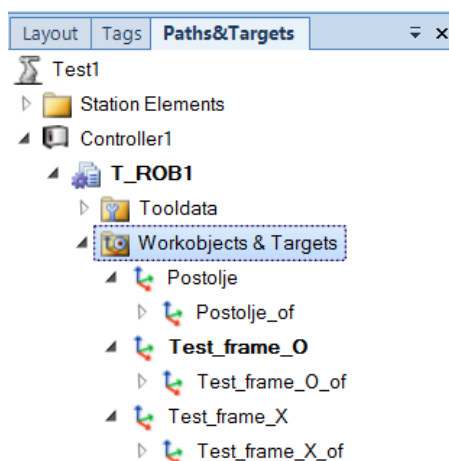
### 4.3. Workobjects & Targets

Koordinatni sustav (engl. *Workobject*) je pojam unutar jezika RAPID koji opisuje prostoru kojem robot izvodi zadatke. Koristi se za definiranje položaja i orijentacije radne površine ili dijela objekta na kojem robot radi. To omogućuje robotu da točno zna gdje se nalaze objekti u prostoru i kako se orijentirati u odnosu na njih.

Točka (engl. *Target*) je položaj u prostoru koji robot mora dosegnuti. Svaka točka opisana je pomoću koordinata (X, Y, Z) i orijentacije te se koristi za programiranje preciznih pokreta robota. Točke se nalaze unutar određenog koordinatnog sustava (*workobject*), što omogućuje robotu točno dohvaćanje zadane točke unutar radnog prostora.

Zadana su tri koordinatna sustava (Slika 4.3-1):

1. Postolje – sadrži 18 točaka koje su postavljene na postolje za igranje
2. Test\_frame\_O – sadrži 10 točaka koje su postavljene ovisno o postolju za O figure
3. Test\_frame\_X – sadrži 11 točaka koje su postavljene ovisno o postolju za X figure i home točku koja vraća robota u početni položaj



Slika 4.3-1 Korišteni koordinatni sustavi

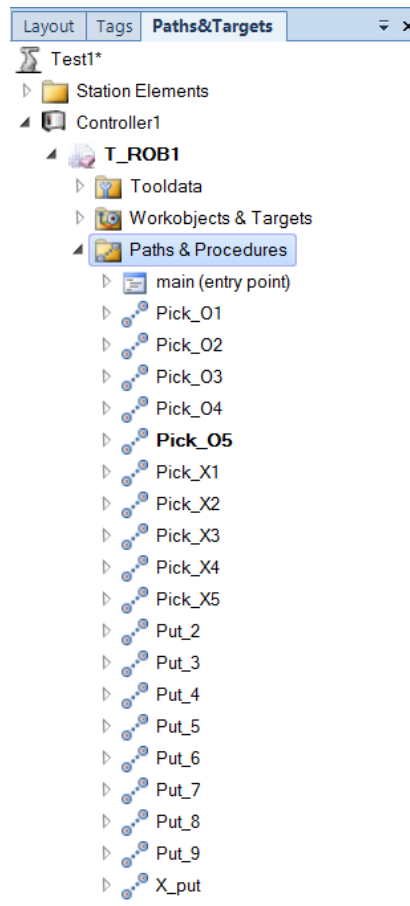
#### 4.4. Putanje i procedure (engl. *Paths & Procedures*)

Putanje (*paths*) predstavljaju niz točaka koje robot prati tijekom kretanja. Putanja definira kako i gdje će se robot kretati u prostoru, bilo linearno ili zakrivljenim putanjom, kako bi izvršio zadatak.

Procedure (*procedures*) su skupovi radnji koje robot izvodi. One uključuju niz putanja i operacija koje robot obavlja u određenom redosljedju, omogućujući fleksibilno programiranje različitih zadataka.

Zajedno, *paths* i *procedures* definiraju kretanje i zadatke robota, omogućujući mu da precizno i učinkovito izvršava svoje funkcije.

Korišteni *paths & procedures* raspodijeljeni su u *Pick* i *Put* (Slika 4.4-1). Unutar *Pick* putanje robot dolazi u položaj iznad figure X ili O otvara prihvatincu, spušta se do figure, zatvara prihvatincu i podiže figuru. Unutar *Put* putanje po kojima robot dolazi u točan položaj iznad podloge za igranje, spušta figuru u željenu točku, otvara prihvatnicu i ispušta figuru.

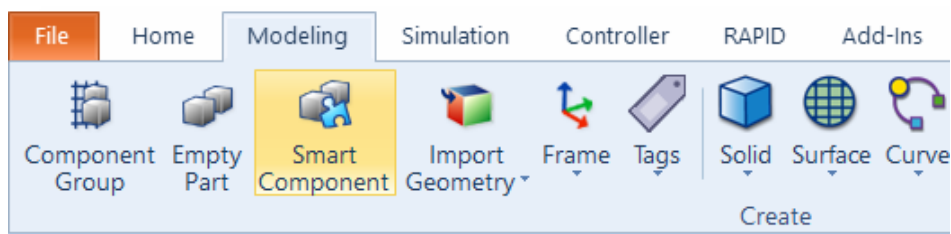


Slika 4.4-1 Paths & Procedures



## 4.5. Pametne komponente (engl. *SmartComponent*)

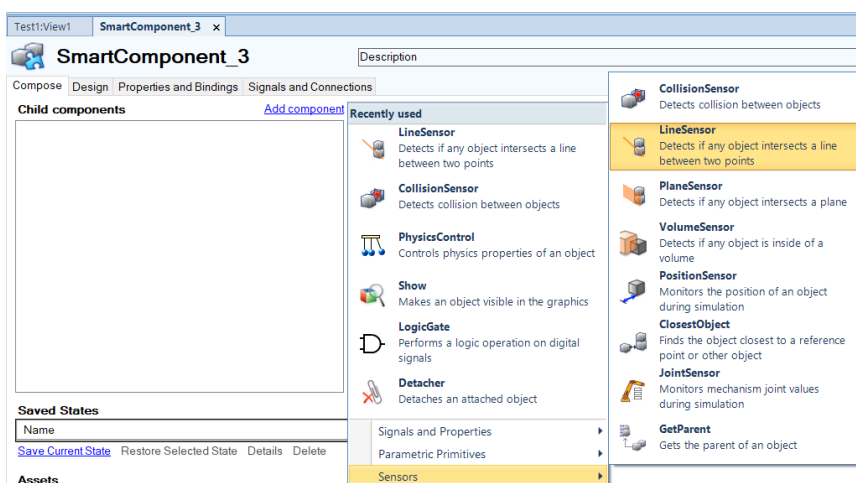
*SmartComponent* (Slika 4.5-1) unutar RobotStudija je komponenta koja omogućava simulaciju različitih funkcionalnosti unutar simulacijskog okruženja bez potrebe za dodatnim programiranjem. Koristi se za simulaciju uređaja, senzora ili bilo kojeg drugog elementa koji može biti povezan s robotskim sustavom. Pametne komponente određuju ponašanje, logiku i interakciju između robota i drugih komponenti u simulaciji. To uključuje senzore, aktuatorne, signale i ostale uređaje. Na taj način se omogućava realističnija simulacija proizvodnih procesa i testiranje interakcije između robota i elemenata.



Slika 4.5-1 Aplikacija *SmartComponent*

### 4.5.1. Korištene pametne komponente (*SmartComponents*)

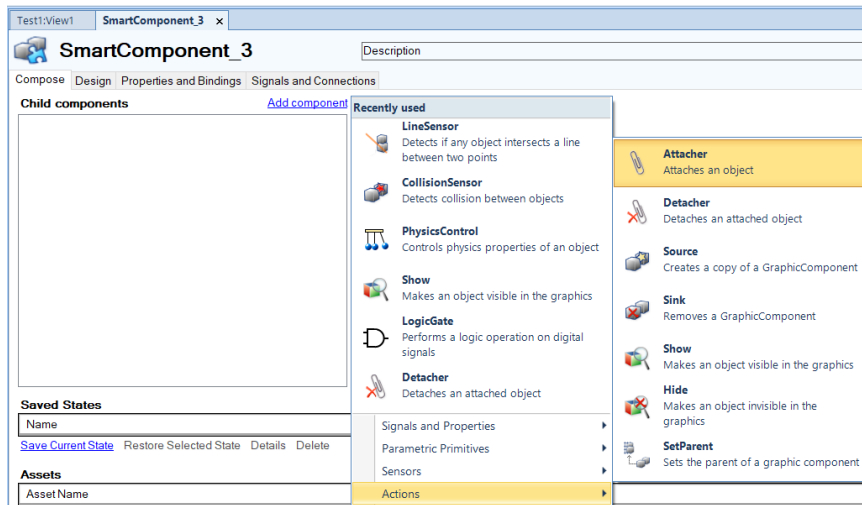
*LineSensor* (Slika 4.5-2) je linijski senzor koji služi za detektiranje predmeta. Prvo je potrebno aktivirati senzor, a kada predmet presiječe liniju senzora, dolazi do detekcije predmeta i senzor u simulaciji daje logičku jedinicu. Pozicioniranje i dimenzioniranje odrađuje se u *Properties* senzora. Senzor pričvršćujemo na prihvatnicu robota kako bi se kretao skupa s prihvatnicom i zbog lakšeg detektiranja predmeta.



Slika 4.5-2 *LineSensor* unutar *SmartComponent*

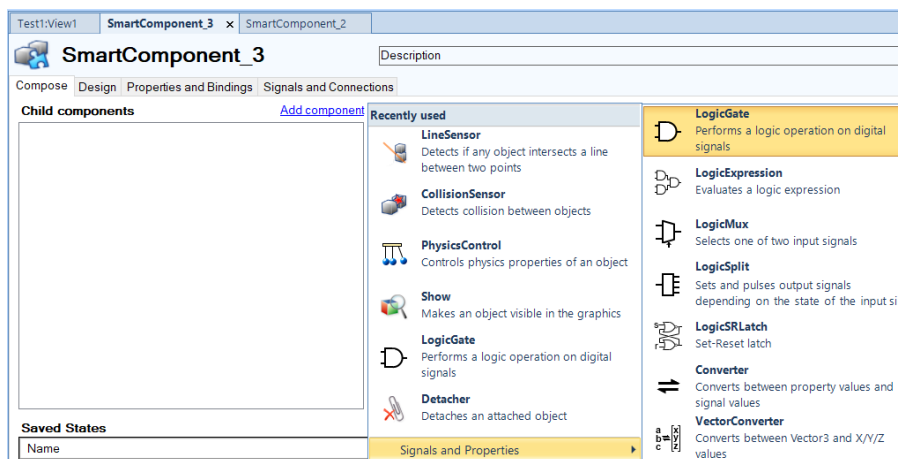
**Attacher** (Slika 4.5-3) se koristi za pričvršćivanje predmeta za druge predmete ili alate koji se koriste.

**Detacher** (Slika 4.5-3) služi za razdvajanje pričvršćenih predmeta i alata ponovno kao zasebnu cjelinu.



Slika 4.5-3 Attacher i Detacher

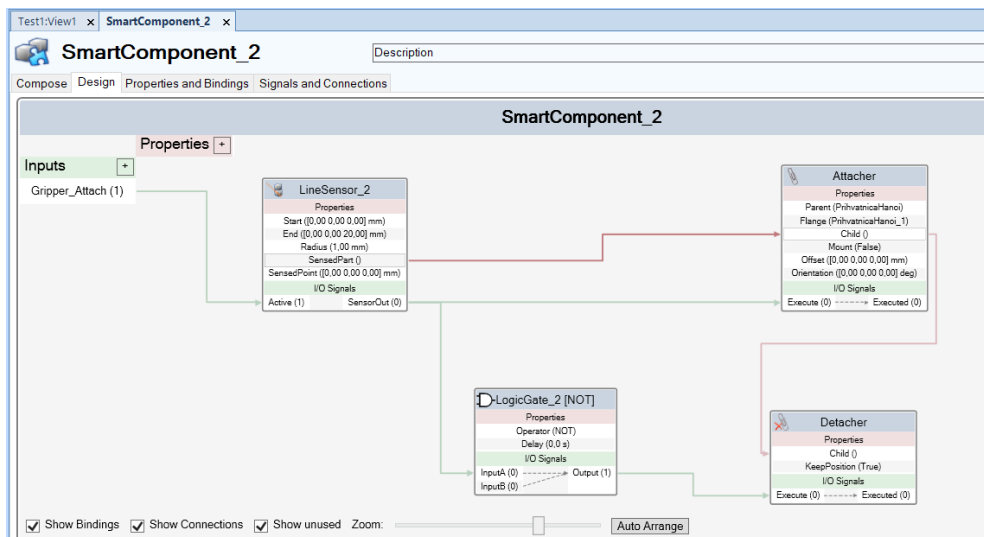
**LogicGate [NOT]** (Slika 4.5-4) je logička funkcija, ne pretvara pozitivan signal na ulazu u negativan na izlazu i obrnuto.



Slika 4.5-4 Logička funkcija NOT

## 4.5.2. Oblikovanje pametne komponente

Spajanje komponenti odrađuje se u kartici *Desing* unutar koje su sve komponente postavljene kao blokovi i potrebno ih je međusobno logički povezati kako bi tvorile uvjete, kontrole i veze koje definiraju ponašanje i interakciju između različitih elemenata simulacije.

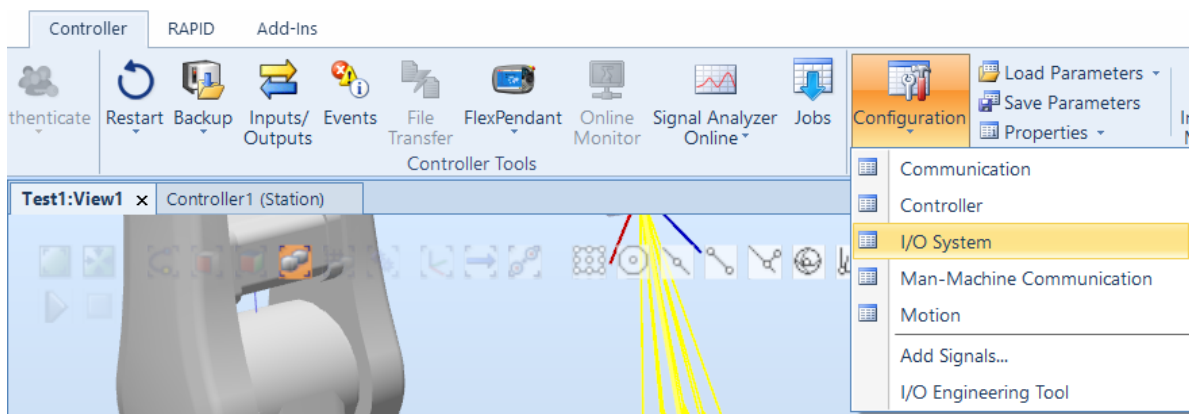


Slika 4.5-5 Desing SmartComponenta

Unutar *Desing* kartice *SmartComponent-a* potrebno je dodati *Input* koji će se koristiti kao ulazni signal na kontroleru i koristi se za aktiviranje senzora. Zelena linija pokazuje ulaze i izlaze koji su međusobno povezani unutar blokova. Spajanjem Ulaza na *Active* senzora aktiviramo senzor, a prilikom detektiranja predmeta, senzor na izlazu daje logičku jedinicu. Izlaz senzora prelazi u logičku jedinicu i aktivira *Attacher* koji pričvršćuje predmet na prihvatincu. Ulaz logičke funkcije *NOT* spojen je se izlazom senzora i služi za aktiviranje i deaktiviranje *Detacher-a*. Predmet s kojim su *Attacher* i *Detacher* povezani, ovisi o predmetu koji senzor detektira, povezano crvenom linijom (Slika 4.5-5).

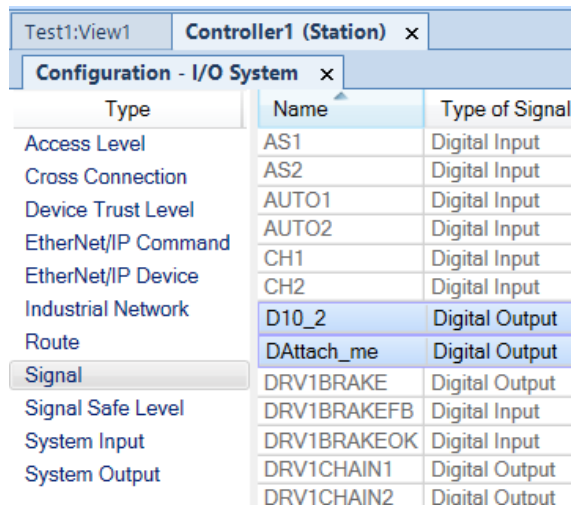
## 4.6. Digitalni signali

Konfiguriranje digitalnih signala za otvaranje i zatvaranje prihvatnice i aktiviranje senzora. Konfiguracija signala nalazi se u tabu *Controller* (Slika 4.6-1) pod izbornikom *Configuration – I/O System – Type – Signal* (Slika 4.6-2)



Slika 4.6-1 Konfiguracija I/O signala

Kreiranje novih digitalnih signala pod nazivom *D10\_2* za otvaranje/zatvaranje prihvatnice i *DAttach\_Me* za aktiviranje i deaktiviranje senzora.

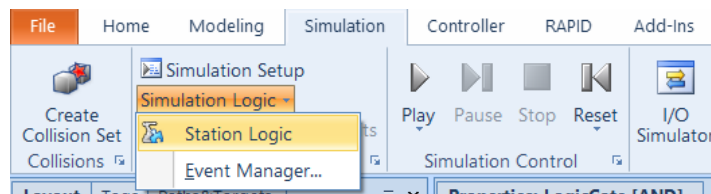


Type	Name	Type of Signal
Access Level	AS1	Digital Input
Cross Connection	AS2	Digital Input
Device Trust Level	AUTO1	Digital Input
EtherNet/IP Command	AUTO2	Digital Input
EtherNet/IP Device	CH1	Digital Input
Industrial Network	CH2	Digital Input
Route	D10_2	Digital Output
Signal	DAttach_me	Digital Output
Signal Safe Level	DRV1BRAKE	Digital Output
System Input	DRV1BRAKEFB	Digital Input
System Output	DRV1BRAKEOK	Digital Input
	DRV1CHAIN1	Digital Output
	DRV1CHAIN2	Digital Output

Slika 4.6-2 Signali prihvatnice i senzora

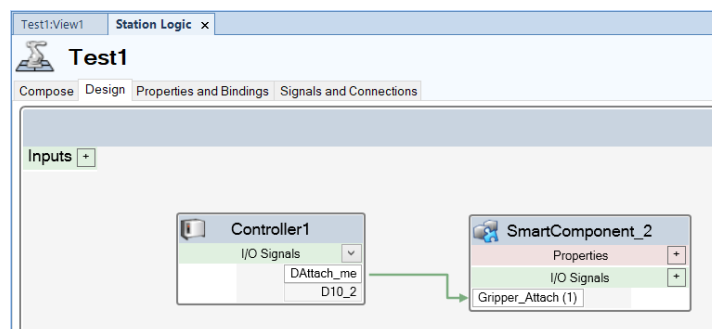
## 4.7. Virtualno povezivanje

*SmartComponent-a* smještena je u simulatoru i nije dio robota i potrebno ju je povezati sa robotom. Povezivanje *SmartComponent-e* s robotom odrađuje se u tabu *Simulation* pod izbornikom *Station Logic* (Slika 4.7-1).



Slika 4.7-1 Station Logic ikona

Unutar *Station Logic* (Slika 4.7-2) naredbe nalazi se *Controller1* koji predstavlja robota. Za ulazni signal robota odabiremo *DAttach\_me* signal koji predstavlja signal za otvaranje/zatvaranje prihvatnice i *SmartComponent\_2* za prihvaćanje predmeta za prihvatincu.

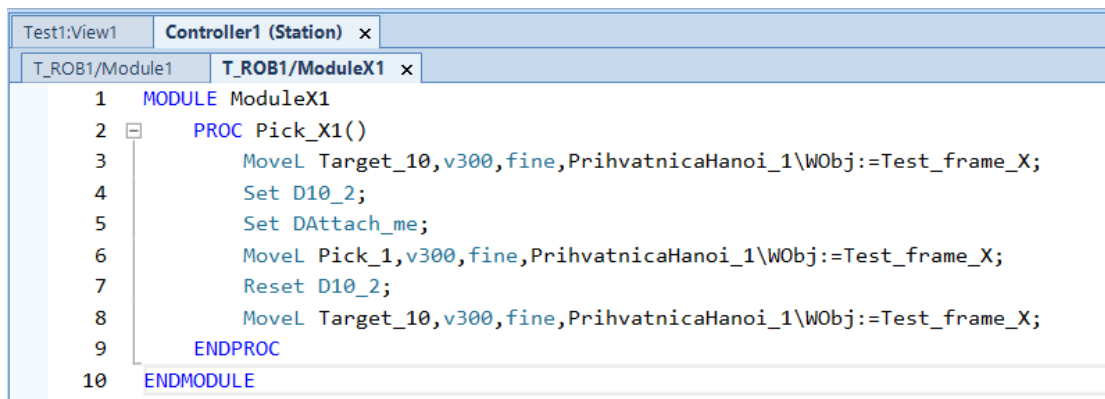


Slika 4.7-2 Station Logic

## 4.8. RAPID

RAPID je programski jezik razvijen od strane ABB-a za programiranje industrijskih robota. Koristi se za upravljanje kretanjem robota, rukovanje alatima, obradom podataka te interakcijom s različitim uređajima poput senzora. U RobotStudiju, RAPID kod omogućuje simulaciju i programiranje robota u virtualnom okruženju.

Stvorene putanje sinkroniziramo u RAPID kod, u kojem su definirane brzine gibanja i preciznost pomicanja robota. Unutar koda potrebno je dodati signale za uključivanje/isključivanje senzora i postavljanja prihvatnice u otvoreni/zatvoreni položaj (Slika 4.8-1 red 4,5,7).



```
Test1:View1 Controller1 (Station) x
T_ROB1/Module1 T_ROB1/ModuleX1 x
1 MODULE ModuleX1
2   PROC Pick_X1()
3     MoveL Target_10,v300,fine,PrihvatnicaHanoi_1\WObj:=Test_frame_X;
4     Set D10_2;
5     Set DAttach_me;
6     MoveL Pick_1,v300,fine,PrihvatnicaHanoi_1\WObj:=Test_frame_X;
7     Reset D10_2;
8     MoveL Target_10,v300,fine,PrihvatnicaHanoi_1\WObj:=Test_frame_X;
9   ENDPROC
10 ENDMODULE
```

Slika 4.8-1 RAPID kod uzimanja predmeta

Naredba **Set D10\_2;** postavlja logičku jedinicu na signalu za otvaranje prihvatnice; dok naredba **Reset D10\_2;** postavlja logičku nulu i resetira signal za zatvaranje prihvatnice.

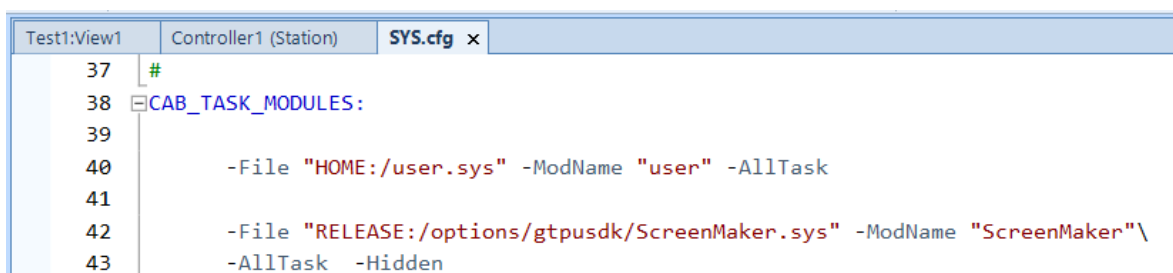
**Set DAttach\_me;** postavlja logičku jedinicu za aktiviranje senzora, a **Reset DAttach\_me;** resetira signal i deaktivira senzor.

## 4.9. Primjena ScreenMaker-a

Za dizajniranje sučelja korištena je aplikacija *ScreenMaker*. Unutar *ScreenMaker*-a korišteni su alati *RunRoutineButton* za pozivanje rutina koje će robot izvršiti i *Button* za otvaranje željenog zaslona.

### 4.9.1. Omogućavanje RunRoutineButton alata

Prije povezivanja *RunRoutineButton* alata sa željenom rutinom, potrebno je omogućiti izvršavanje rutine. Za omogućavanje *RunRoutineButton* alata korišten je priručnik *ABB Operatin manual* za RobotStudio. Potrebno je ubaciti liniju koda u *SYS.cfg* datoteku koja se nalazi na računalu kako bi se riješio problem. Pretraživanje datoteke je moguće putem Windows pretraživača. Upisivanjem *SYS.cfg* datoteke u tražilicu.



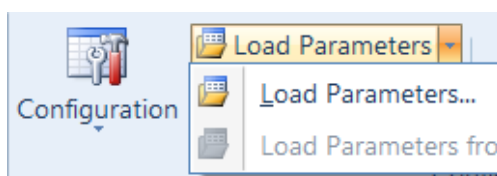
```
Test1:View1 | Controller1 (Station) | SYS.cfg x
37 | #
38 | CAB_TASK_MODULES:
39 |
40 |     -File "HOME:/user.sys" -ModName "user" -AllTask
41 |
42 |     -File "RELEASE:/options/gtpusdk/ScreenMaker.sys" -ModName "ScreenMaker"\
43 |     -AllTask -Hidden
```

Slika 4.9-1 Ubacivanje potrebnog koda

Liniju koda ubaciti ispod *CAB\_TASKS\_MODULES*:

```
File "RELEASE:/options/gtpusdk/ScreenMaker.sys" -ModName "ScreenMaker"\  
-AllTask -Hidden
```

Nakon ubacivanja koda potrebno je spremiti *SYS.cfg* datoteku, učitati u tabu *Controller>Cofiguration* prihvatiti promjene i novu datoteku zamijeniti s postojećom, resetirati sustav robota.



Slika 4.9-2 Učitavanje parametara i resetiranje kontrolera

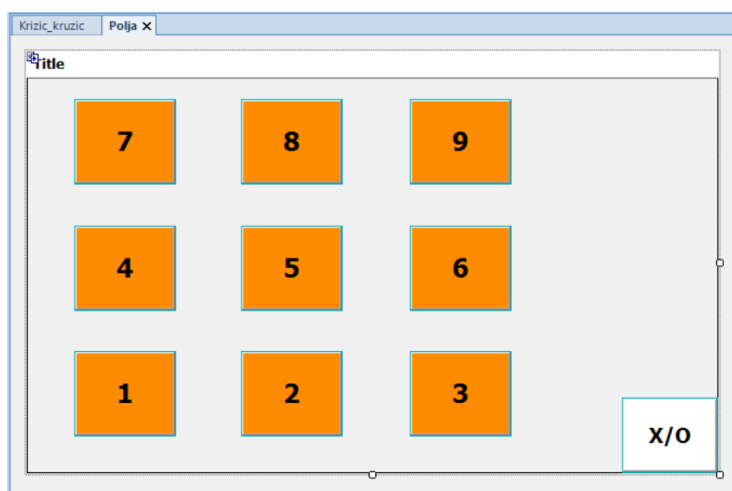
## 4.9.2. Dizajniranje sučelja za igru

Sučelje je raspodijeljeno na dva zaslona. Prvi zaslon *Krizic\_kruzic* je ujedno i Main screen koji će se prvi prikazati prilikom otvaranja sučelja. Na njemu su raspoređene dvije grupe X i O u dva stupca. Svaki stupac sadrži pet tipki tj. *RunRoutineButton*-a preko kojih robot izvršava radnju uzimanja figure iz odabrane grupe i određene pozicije. *Button* Polja otvara zaslon *Polja*.



Slika 4.9-3 Zaslon *Krizic\_kruzic*

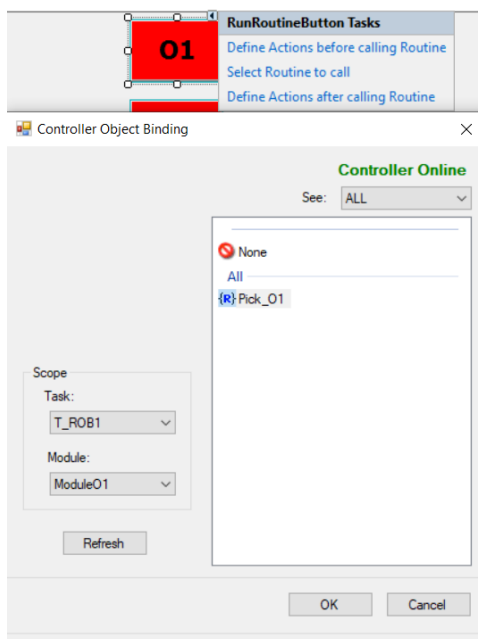
Drugi zaslon sadrži devet *RunRoutineButton* alata postavljenih u polje 3x3. Svaka tipka definira gibanje robota u tu točku, što obuhvaća pozicioniranje iznad te točke spuštanje figure malo iznad postolja, ispuštanje figure u odabrano polje i vraćanje robota u *Home* poziciju. *Button* tipka služi za otvaranje zaslona *Krizic\_kruzic*.



Slika 4.9-4 Zaslon *Polja*

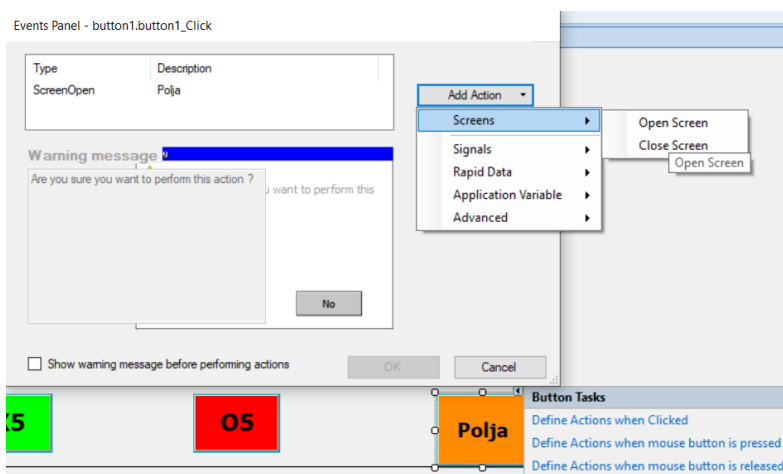
### 4.9.3. Podešavanje korištenih alata

Za povezivanje *RunRoutineButton* alata s rutinom potrebno je otvoriti izbornik *RunRoutineButton Task* koji se nalazi iznad alata i odabrati *Select Routine to call*. Odabirom zadatka koji želimo da alat odradi otvara se izbornik za povezivanje rutine (Slika 4.9-5). Unutar *Scope* izbornika odabiremo robotski sustav na kontroleru i željeni modul koji se nalazi unutar tog robotskog sistema.



Slika 4.9-5 Definiranje zadatka *RunRoutineButton* alata

Otvaranje drugog zaslona izvedeno je pomoću *Button* alata. Definiranjem radnje otvaranja drugog zaslona pritiskom tipke, odabirom zadatka *Define Action when Clicked*, dodaje se novi zadatak *Add Action>Screens>Open Screen* i odabire jedan od ponuđenih, u ovom slučaju zaslon Polja.

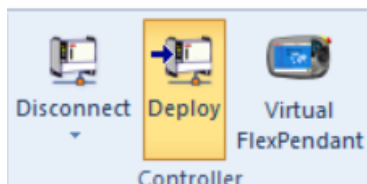


Slika 4.9-6 Definiranje zadatka *Button* alata



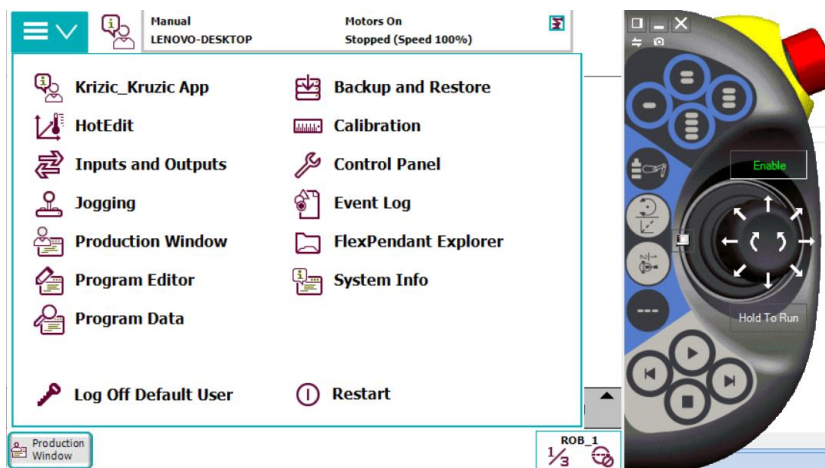
## 4.10. Učitavanje programa na FlexPendant-u

Nakon dizajniranja sučelja, sučelje prebacujemo u virtualni kontroler robota. *Deploy* aplikacija prebacuje kompletno sučelje iz *ScreenMaker-a* na *FlexPendant*.



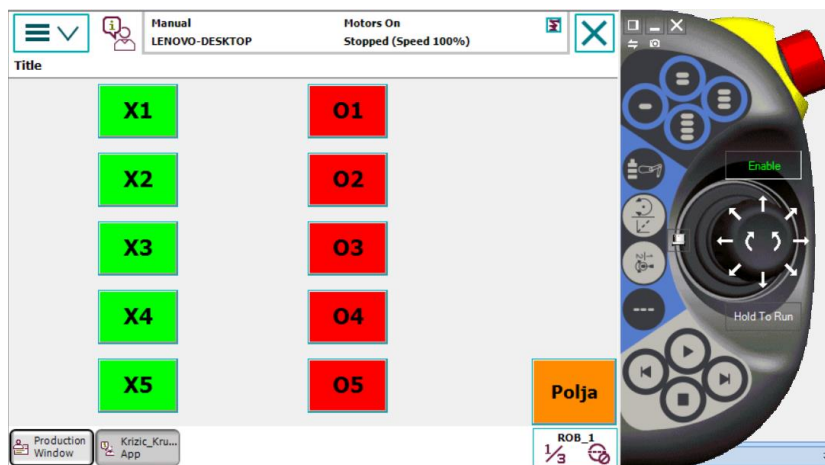
Slika 4.10-1 Deploy aplikacija

Završetkom prebacivanja unutar izbornika *FlexPendant* treba se nalaziti *ScreenMaker* pod spremljenim nazivom *Kirzic\_Kruzic App*.



Slika 4.10-2 Izbornik FlexPendant-a

Otvaranjem aplikacije (Slika 4.10-3) otvara se *Main Screen* *Krizic\_kruzic*.



Slika 4.10-3 Krizic\_Kruzic App

## 5. Zaključak

Ovaj rad služi kao priručnik za izradu i korištenje *ScreenMaker-a*. U radu je opisan postupak instalacije, izrade korisničkog sučelja za ABB IRB120 robot i simulaciju igre *Krizic\_kruzic*. Korištenje uređaja *FlexPendant* omogućuje korisnicima upravljanje robotom kroz radni prostor što uključuje uzimanje figura iz postolja u kojem se nalaze i postavljanjem istih nasumičnim izborom u postolje za igru.

Simuliranjem programa utvrđena je funkcionalnost korisničkog sučelja. Tijekom izrade rada naišli smo na problem virtualnog senzora. Senzor ne detektira predmet prilikom simulacije, provjereni su ključni parametri: aktiviranje senzora, vidljivost figura senzoru.

Rad predstavlja sažet i strukturiran pregled ključnih informacija u području koje je slabo pokriveno na internetu. S obzirom na ograničen broj dostupnih resursa i privjesaka ovog tipa, ovaj rad ispunjava važnu prazninu pružajući jasne smjernice i praktične alate za učenje, čime doprinosi poboljšanju dostupnosti i kvalitete edukativnih materijala.

Moguća nadogradnja u radu je napraviti dodatno sučelje za resetiranje programa unutar kojeg robot vraća se korištene figure nazad u postelje iz kojeg su korištene. Povećala bi se automatizacija rada robota i smanjila potreba za fizičkim vraćanjem figura pomoću korisnika.

## 6. Literatura

- [1] ABB Operating manual RobotStudio, Švedska, 6.08, 2008-2018  
<https://robotum.cz/wp-content/uploads/2019/12/3HAC032104-OM-RobotStudio-en.pdf>
  
- [2] Application manual ScreenMaker, Švedska, 5.12, 2009  
[https://library.e.abb.com/public/b62731a5b2d5528bc125766d003a6228/Application\\_Manual\\_ScreenMaker.pdf](https://library.e.abb.com/public/b62731a5b2d5528bc125766d003a6228/Application_Manual_ScreenMaker.pdf)
  
- [3] YouTube: kanal; CraigMaynard7  
<https://www.youtube.com/@CraigMaynard7/playlists>
  
- [4] Link ABB-ove stranice za instalaciju ScreenMaker-a  
<https://developercenter.robotstudio.com/>

## 7. Popis slika

Slika 2.1-1 Logo RobotStudio programa.....	2
Slika 2.3-1 Simulacija FlexPendant unutar programa RobotStudio.....	3
Slika 2.3-1 Aplikacija ScreenMaker softvera unutar programa RobotStudio .....	4
Slika 3.1-1 Stranica za preuzimanje instalacijskog paketa.....	4
Slika 3.1-2 Datoteke unutar instalacijskog programa.....	5
Slika 3.1-3 Instalacijski prozor .....	5
Slika 3.1-4 Održavanje programa .....	6
Slika 3.2-1 Pokretanje ScreenMaker-a .....	6
Slika 3.2-2 Odabiranje odgovarajuće verzije ScreenMaker-a .....	7
Slika 3.3-1 Razvojno okruženje unutar ScreenMaker-a .....	7
Slika 3.3-2 Dijelovi sučelja ScreenMaker .....	8
Slika 3.3-3 Kartica Controller .....	9
Slika 3.3-4 Odabir Robot Controller-a .....	9
Slika 3.4-1 Prikaz alata ActionTrigger .....	10
Slika 3.4-2 Prikaz Button-Properties alata .....	11
Slika 3.4-3 Prikaz NumericUpDown alata .....	11
Slika 3.4-4 Prikaz RunRoutineButton alata.....	12
Slika 4.2-1 Korišteni 3D modeli.....	13
Slika 4.2-2 Korišteni alat - prihvatnica.....	14
Slika 4.3-1 Korišteni Workobject-i .....	14
Slika 4.4-1 Paths & Procedures .....	15
Slika 4.5-1 Aplikacija SmartComponent.....	16
Slika 4.5-2 LineSensor unutar SmartComponent .....	16
Slika 4.5-3 Attacher i Detacher .....	17
Slika 4.5-4 Logička funkcija NOT .....	17
Slika 4.5-5 Desing SmartComponenta .....	18
Slika 4.6-1 Konfiguracija I/O signala.....	18
Slika 4.6-2 Signali prihvatnice i senzora .....	19
Slika 4.7-1 Station Logic ikona .....	19
Slika 4.7-2 Station Logic .....	19
Slika 4.8-1 RAPID kod uzimanja predmeta .....	20
Slika 4.9-1 Ubacivanje potrebnog koda .....	21
Slika 4.9-2 Učitavanje parametara i resetiranje kontrolera .....	21

Slika 4.9-3 Zaslona Krizic_kruzic .....	22
Slika 4.9-4 Zaslona Polja.....	22
Slika 4.9-5 Definicija zadatka RunRoutineButton alata .....	23
Slika 4.9-6 Definicija zadatka Button alata.....	23
Slika 4.10-1 Deploy aplikacija .....	24
Slika 4.10-2 Izbornik FlexPendant-a .....	24
Slika 4.10-3 Krizic_Kruzic App .....	24



## IZJAVA O AUTORSTVU

Završni 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, Leon Rabić pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog rada pod naslovom Izrada korisničkog sučelja za ABB IRB120 te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student:

*Leon Rabić*

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.