

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

|  |
| --- |
| **ELEKTROS IR ELEKTRONIKOS FAKULTETAS** |

**Einius Dumskis**

**Robotizuotos Sistemos Projektas**

(T125B153)

**IRC5 valdiklio sąsajos su PC platformą sudarymas naudojant WindowsForms aplikacija**

Projektas

|  |  |
| --- | --- |
|  | **Vadovas**  Renaldas Urniežius |

**KAUNAS, 2018**

Turinys

[1. Įvadas 2](#_Toc532943672)

[2. ABB IRB 120 manipuliatorius 2](#_Toc532943673)

[3. IRC 5 Valdiklis 3](#_Toc532943676)

[4. FlexPendant Valdymo Operatorius 4](#_Toc532943677)

[5. RobotStudio Aplinka 8](#_Toc532943678)

[6. RAPID Programavimas 11](#_Toc532943679)

[7. .NET sąsaja su IRC5 17](#_Toc532943680)

[8. Pagrindinės kodo funkcijos 29](#_Toc532943681)

[9. Duomenų modifikavimas 31](#_Toc532943682)

[10. Pavyzdinė užduotis 33](#_Toc532943683)

[11. Rezultatai 38](#_Toc532943684)

[12. Galimos problemos 39](#_Toc532943685)

[13. Priedas A 40](#_Toc532943686)

[14. Priedas B 41](#_Toc532943687)

# **Įvadas**

Pagrindinis darbo tikslas įgauti pagrindinių IRB 120 manipuliatoriaus valdymo žinių prgramuojant RAPID kalba RobotStudio Aplinkoje ir C# kalba, sudarant WindowsForm aplikaciją, VisualStudio aplinkoje.

Darbo užduotis susidės iš šių punktų:

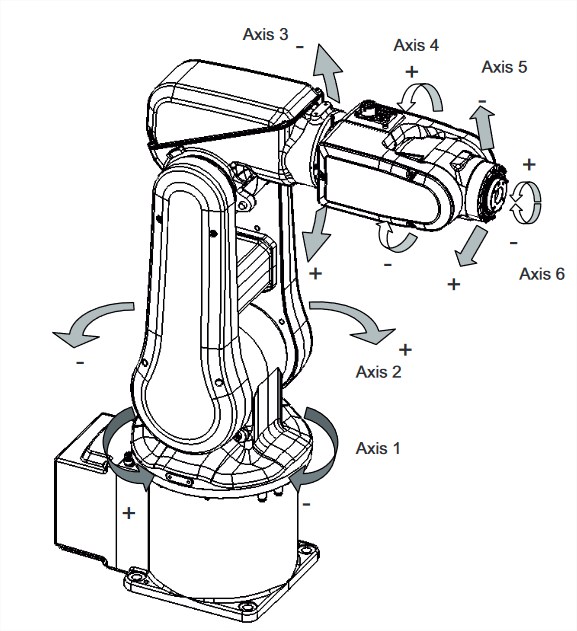
1. Susipažinti su ABB IRB 120 manipuliatoriumi ir ABB IRC 5 valdikliu.
2. Susipažinti su ABB FlexPendant rankiniu operatoriumi.
3. Susipažinti su ABB RobotStudio projektavimo aplinka.
4. Susipažinti su RAPID programavimu.
5. Sudaryti .NET platformos sąsają su IRC 5 valdikliu.
6. Parašyti C# ir RAPID programas, kurios nurodytų robotui nubraižyti trikampį.

Priedą A sudarys RAPID kodas iš pavyzdinės užduoties.

Priedą B sudarys kai kurių C# funkcijų gilesni paaiškinimai.

# ABB IRB 120 manipuliatorius

IRB 120 yra 6 ašių manipuliatorius. Jo maksimalus siekis yra 580mm, krovinio svoris gali būti iki 3 kg.



pav.



# IRC 5 Valdiklis

IRC 5 valdiklis vykdo komandas judindamas IRB 120 manipuliatorių. Jį galima valdyti FlexPendant rankiniu operatoriumi arba prisijungus ethernet kabelį į PC naudojantis RobotStudio programavimo ir projektavimo aplinka. Jis įjungiamas sukamo kontaktoriaus įjungimu. Esant pavojingai situacijai, valdiklis gali būti išjungiamas raudono mygtuko paspaudimu. Norint paleisti programą, turi būti įjungtas automatinis režimas pasukant raktą į kairiąją poziciją ir paspaustas baltas mygtukas, kad būtų įjungti varikliai. Norint valdyti manipuliatorių per FlexPendant operatorių turi būti įjungiamas rankinis režimas pasukant raktą į dešinę poziciją.



D

C

A

B

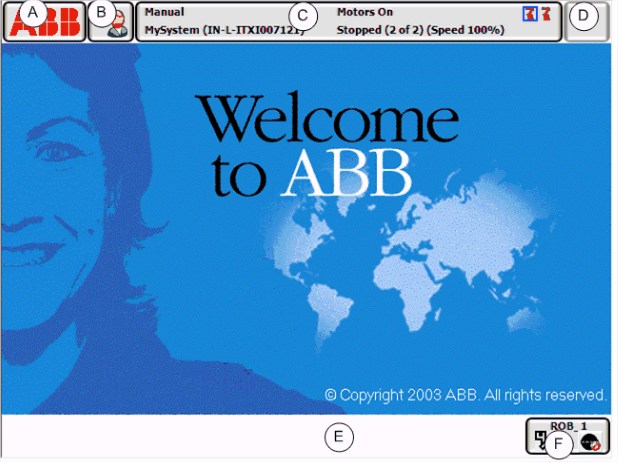
pav. IRC 5 Valdiklis

# FlexPendant Valdymo Operatorius

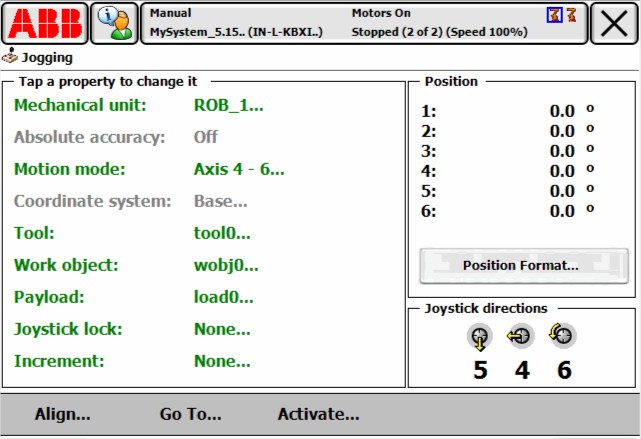
FlexPendant yra rankomis laikomas valdymo prietaisas naudojamas atlikti dauguma užduočių kontroliuojant robotų sistemą: pradėti programoms, valdyti manipuliatorių, modifikuoti RAPID programas, keisti parametrus.FlexPendant susideda iš techninės ir pragraminės įrangos ir pats yra kaip atskiras kompiuteris. Jis yra neatskiriama IRC5 valdiklio dalis ir yra sujungtas su valdikliu per kabelį.

Norint išbandyti operatorių reikia pirmiausiai įjungti IRC5 valdiklį(Pav.2 A) ir kai pasikraus valdiklio operacinė sistema įjungti rankinį rėžimą pasukus raktą pagal laikrodžio rodyklę.(Pav.2 D)

Tada reikia nueiti į ABB meniu (Pav.4 A), paspausti skiltį „Jogging“ ir nustatyti „Motion mode“ į „Linear“(Pav.5 A), o „Coordinate system“ į Base(Pav.5 B),. Ties laukeliu „Joystick directions“ galima pamatyti kaip yra valdomas manipuliatorius valdymo lazdele. Rodyklės nurodo manipuliatoriaus galinio įtaiso judėjimą teigiama ašies puse pasirinktoje koordinačių sistemoje.



pav. Pradinis langas



B

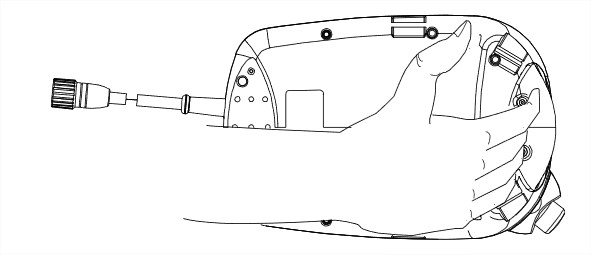
A

C

pav. Jogging parinktys

Norint atlikti judėjimo operacija reikia įjungti saugikli FlexPendant šone, kaip parodyta.(Pav 6)

**Pastaba: Paspaudus saugiklį per stipriai, jis neįsijungs.**



pav. Saugiklio įjungimas

Išgirsite kaip įsijungia varikliai ir judinant valdymo lazdelę.(Pav 7 A)

C



A

B

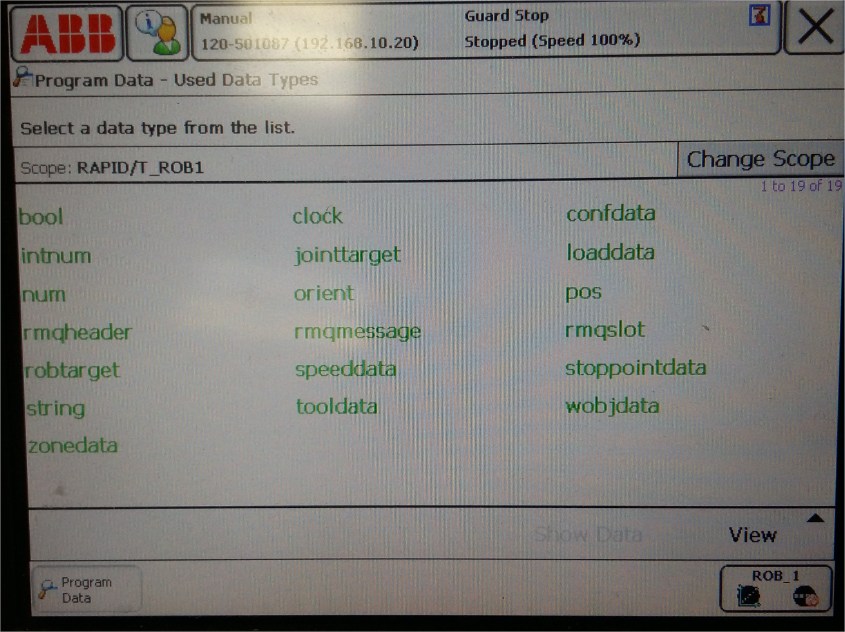
pav. FlexPendant

Roboto judėjimo greitis priklausys nuo to, kiek pasukta ar palenkta lazdelė.

**Ispėjimas: Patartina iš pradžių menkai pasukti lazdelę, kadangi manipuliatorius gali greitai pajudeti ir isirėžti, jeigu aplinkui yra objektų ir esate nepatyrę. Esant pavojingai situacijai išjungti valdiklį galima su „Emergency Stop“ mygtuku.(Pav 7 C)**

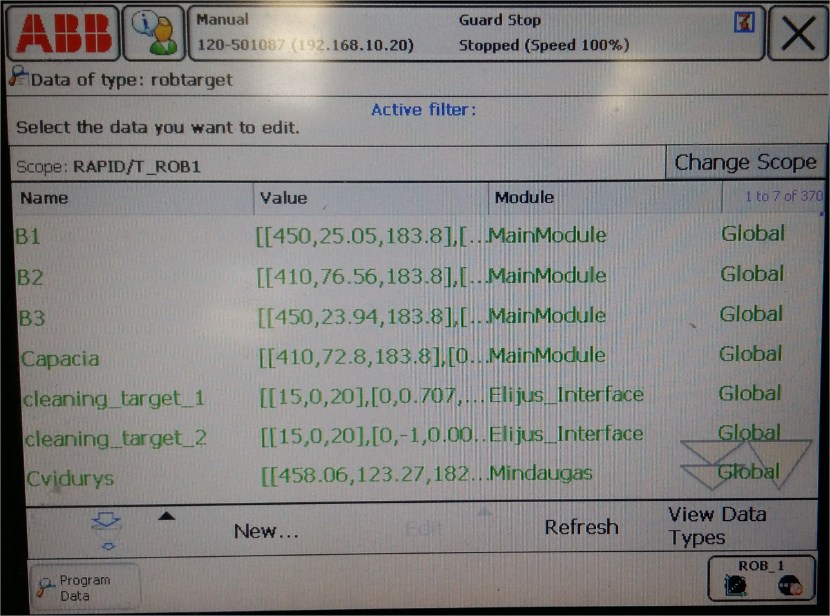
Ant roboto esantis darbo įrankis gali būti reorentuotas. Jeigu „Tools“ skiltyje yra parinktas „tool0“(Pav.5 C), kurio koordinatės yra ties 6-ja roboto ašimi, įrankis jį galima sukinėti įjungus reorentacinį režimą „Motion mode“ skiltyje(Pav.5 A) arba FlexPendant pulte paspaudus mygtuką, kuris keičia tarp „Linear“ ir „Reorient“ režimų.(Pav.7 B)

FlexPendant pulte galima keisti, įrašyti duomenis. Tai galima padaryti FlexPendant pulte paspaudus „ABB meniu“, pasirinkus „Program data“.



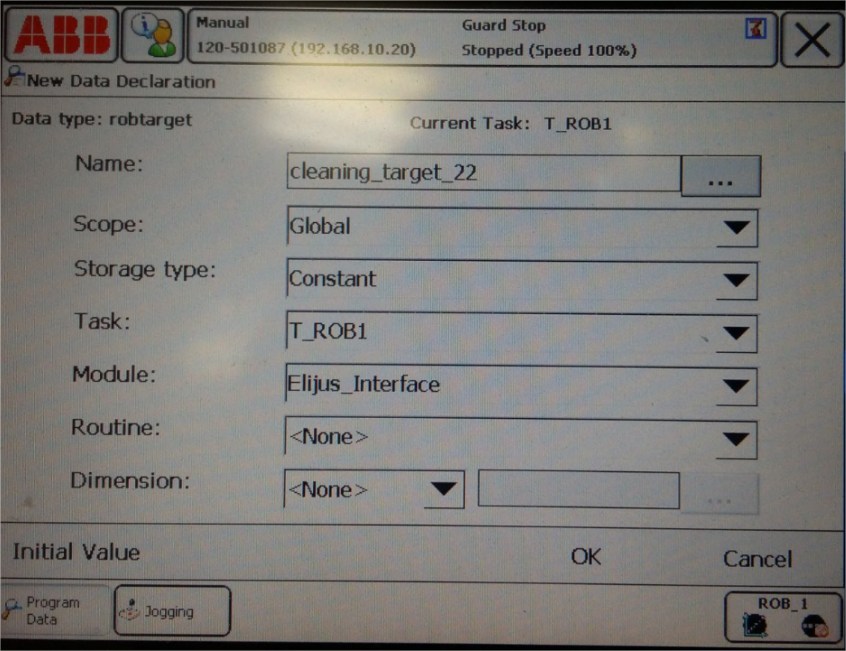
pav.

Čia yra surašyti visų tipų kintamieji, kurie yra naudojami programos. Susikurti nauja „robtarget“ tipo kintamajį reikia dukart paspausti ant „robtarget“.



pav.

Įėjus matomi visi „robtarget“ kintamieji. Norint susikurti naują, reikia spausti „New...“, parinkti „Name“ ir kitus parametrus, apie kuriuos daugiau bus pasakyta „RAPID Programavimo“ skiltyje.

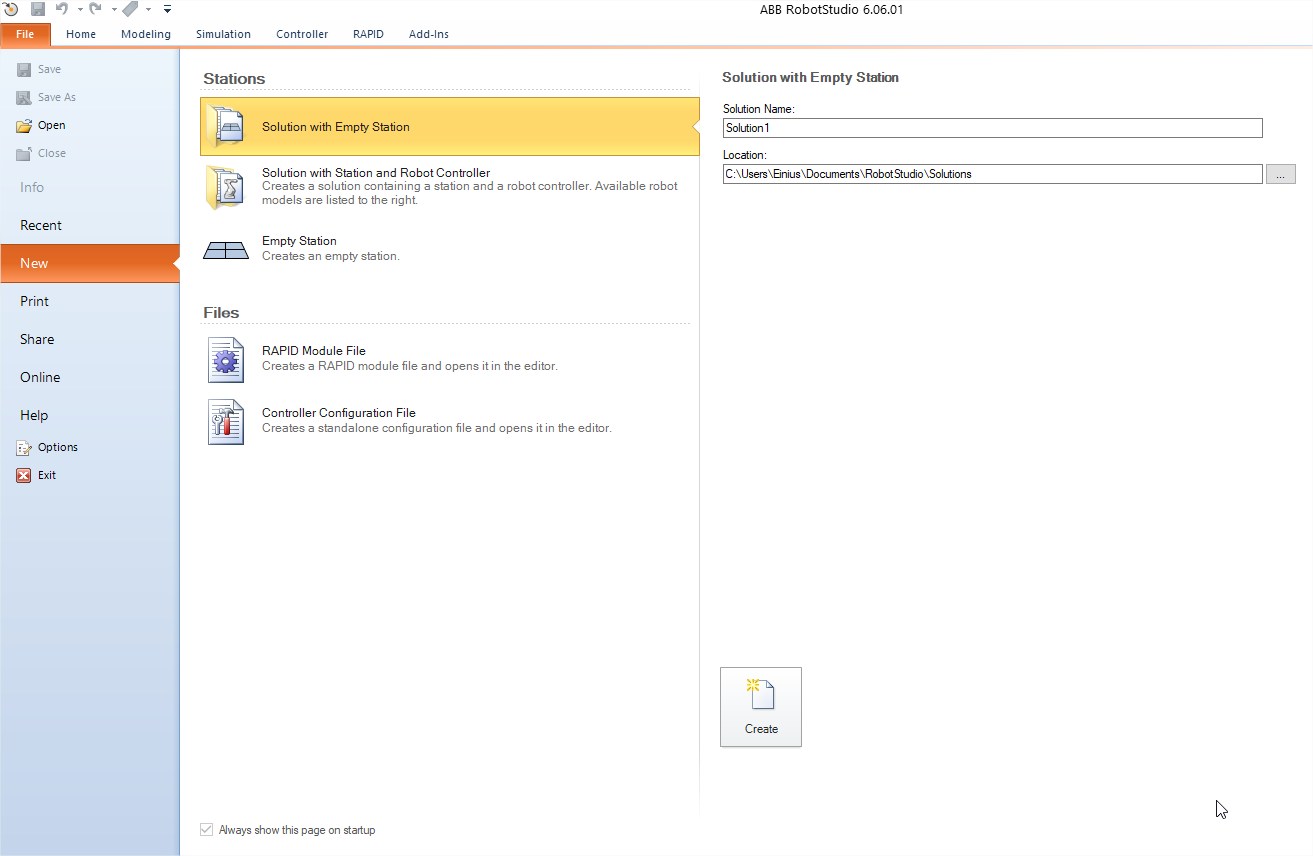


pav.

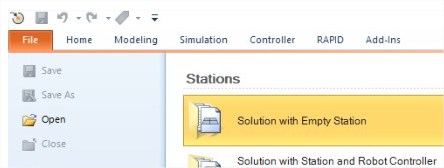
Paspaudus „OK“ bus sukurtas naujas kintamasis, kuris, šiuo atveju, apibūdina manipuliatoriaus poziciją ir orientaciją su tam tikru įrankiu, kurį galima pasikeisti „Jogging“ skiltyje.

# RobotStudio Aplinka

Norint matyti, keisti IRC5 valdiklyje esančius RAPID programų failus reikia naudotis RobotStudio projektavimo/programavimo aplinka. Pirmiausia reikia įsijungti RobotStudio aplinką. Ir eiti į „Controller“ skiltį.

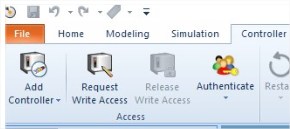


pav. ABB RobotStudio



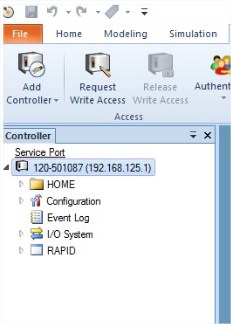
pav.

Prisijungus IRC5 valdiklio ethernet kabelį prie PC reikia paspausti ant „Add Controller“.



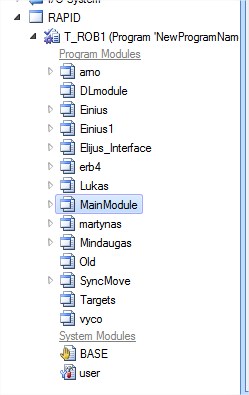
pav. RobotStudio meniu juosta

Prisijungus valdikliui pamatysite toki vaizda:



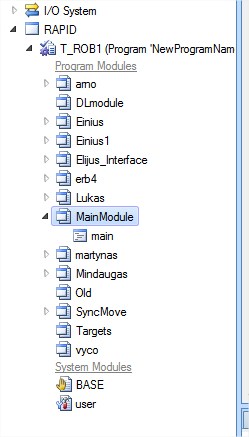
pav. Valdiklio skiltis

Paspaudus RAPID skiltį du kartus ir joje esančią T\_ROB1 skiltį du kartus, pasirodys T\_ROB1 užduoties moduliai:



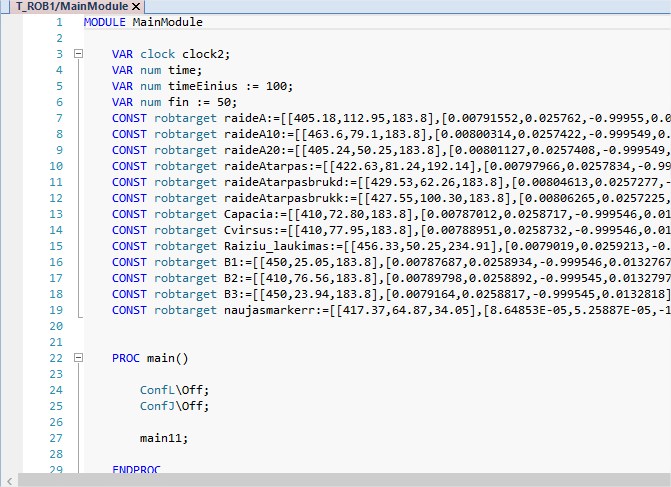
pav. Moduliai

Paspaudus ant bet kurio modulio galima matyti jame esančius procesus:



pav. Modulio procesai

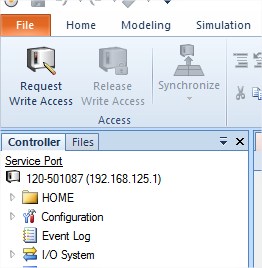
Pasirinkus modulį, galime matyti apibūdintus procesus bei kintamuosius:



pav. RAPID kodas

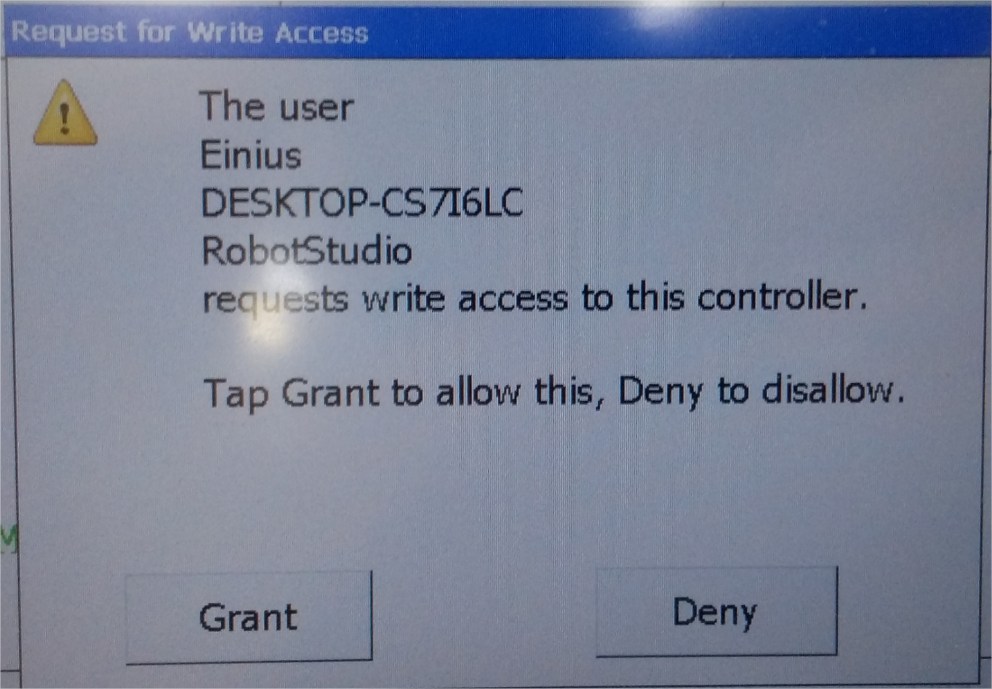
# RAPID Programavimas

Norint pradėti programuoti RAPID reikia susikurti nauja modulį valdiklyje. Tai padaroma pirmiausiai gaunant prieigą prie valdiklio, kai jis yra automatiniame režime, spaudžiant „Request Write Access“.



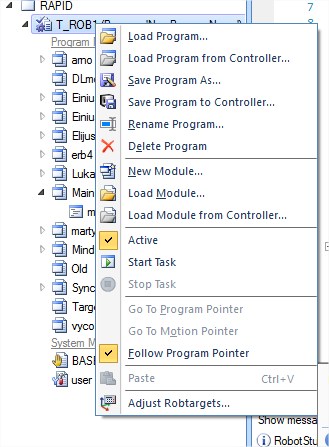
pav. Meniu juosta

Jeigu valdiklis yra rankiniame režime, paspaudus „Request Write Access“, FlexPendant pulte atsiras užklausa leisti prisijungti prie valdiklio :



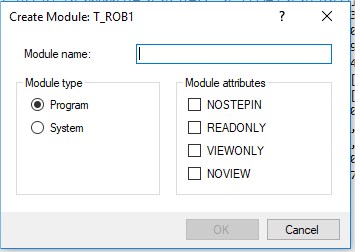
pav. Užklausa

Susikurti nauja modulį spaudžiama dešiniu ant T\_ROB1 užduoties ir tada „New Module...“:



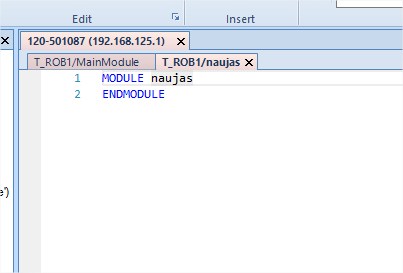
pav. Naujo Modulio sukūrimas

Išmetamas langas ir jame parašome modulio vardą ir spaudžiam „OK“:



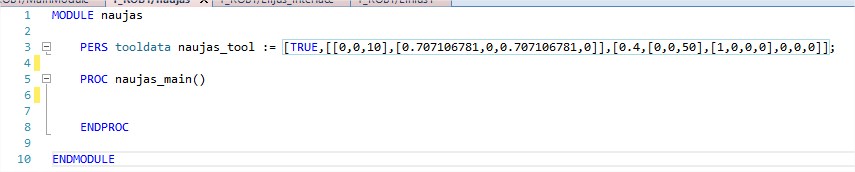
pav. Kūrimo parametrai

Kai modulis sukurtas, išmetama tuščia modulio skiltis:



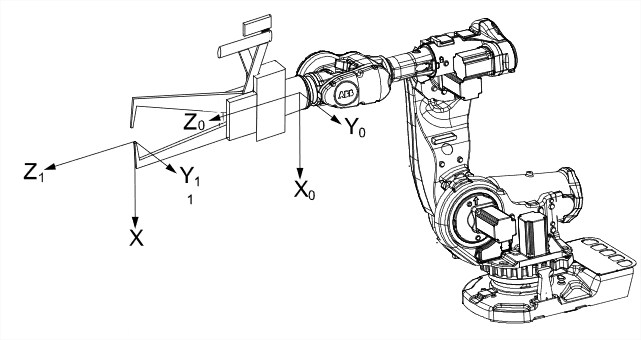
pav. Naujas modulis

Modulyje gali būti apibūdinami nauji procesai, funkcijos, kintamieji. Pirmiausia reikia apibūdinti kokį įrankį manipuliatorius laiko ir proces, kur bus vykdomos funkcijos:



pav. Naujas kintamasis

Naujas kintamas prasideda „PERS“, kas pasako, kad tai kintamasis, kurio paskutinė vertė išsisaugo po valdiklio išjungimo, „tooldata“ pasako, kad tai įrankio tipo kintamasis, „naujas\_tool“ yra vardas. Parametras „TRUE“ pasako, kad robotas laiko šį irankį. Parametrai „[0,0,10],[0.707106781,0,0.707106781,0]” nusako kiek nutolusi įrankio koordinatė ir kaip ji orientuota pagal „tool0“ koordinačių sistemą:



pav. Įrankio koordinčių sistema

Kiti parametrai nusako svorį, kiek nutolęs svorio centras nuo „tool0“ koordinačių sistemos, inercinius momentus.

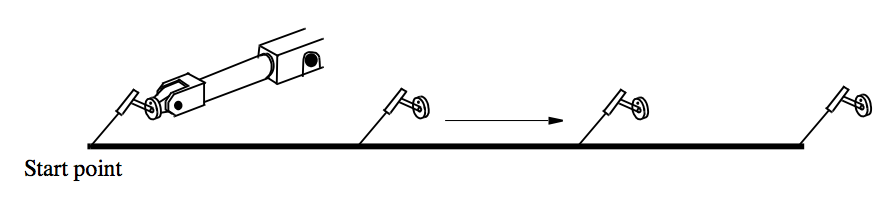
Norint išbandyti paprasta roboto judesio funkcija reikia apibundinti „robtarget“ tipo kintamajį ir priskirti jam esamą roboto poziciją ir orientaciją:



pav. Paprastas manipuliatoriaus judėsio procesas

“Naujas\_target” yra priskiriama esama roboto pozicija ir orientacija su irankiu “naujas\_tool”. Tai padaroma su funkcija “CRobT”, kuri gražina esamą roboto pozicija su įrankiu, kuris priskiriamas su parametru “\Tool:=”.

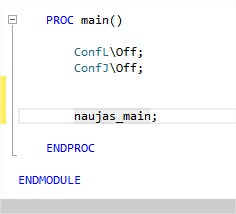
MoveL procesas nuveda manipuliatorių iš esamos į nurodytą poziciją. Šiuo atveju esama pozicija, skaičiuojama nuo įrankio koordinatės, yra pastumiama X ašimi 10mm pagal bazines koordinates. Panaudota „Offs“ funkcija, kuri pastumia „naujas\_target“ per 10mm X ašimi. „v50“ apibudina greitį, kurio judės taškas, šiuo atveju 50mm/s. „Fine“ parametras nusako, kad manipuliatorius judės tiesiai link taško be jokių apėjimų. „Naujas\_tool“ nusako, kad manipuliatorius nuveda pačio įrankio pradinį tašką, kuris šiuo atveju yra 10mm nutolęs nuo manipuliatoriaus paskutinės grandies:



pav. Manipiliatoriaus judėjimo su įrankiu

**Pastaba:** Funkcijoms reikia skliaustais apimti parametrus, o procesams tik kablelių pakanka.

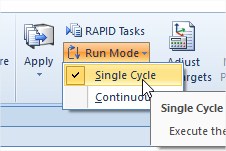
Norint pradėti programą, procesas „naujas\_main“ turi būti aprašytas „MainModule“ modulio „main“ procese:



pav. Proceso įtraukimas į pagrindinį modulį

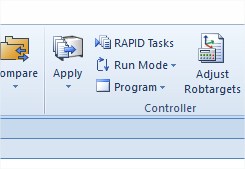
Patartina apsirašyti „ConfL\Off“ ir „ConfJ\Off“, antraip gali mesti klaidą.

Pasirenkame vieno ciklo režimą, kad manipuliatorius, kartodamas ciklą, neįsirėžtų:



pav. Vieno ciklo pasirinkimas

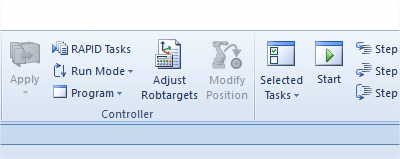
Spaužiame „Apply“:



pav. RAPID kodo įrašymas

Norint pradėti programą, valdiklis turi būti pejungtas raktu į automatinį režimą ir turi buti nuspaustas baltas mygtukas varikliu įjungimui.

Tada programa gali būti paleidžiama RobotStudio aplinkoje.



pav. Paleidimas RobotStudio aplinkoje

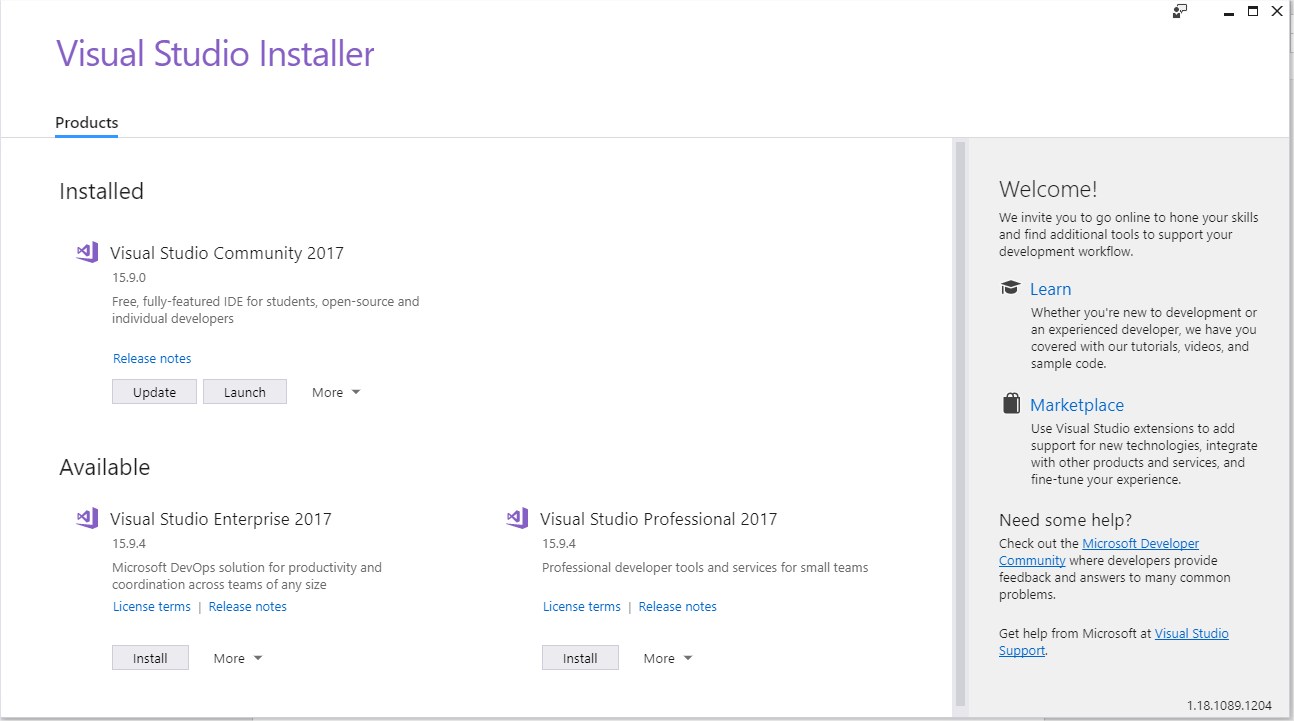
Arba per FlexPendant paspaudus “Play“ mygtuką:



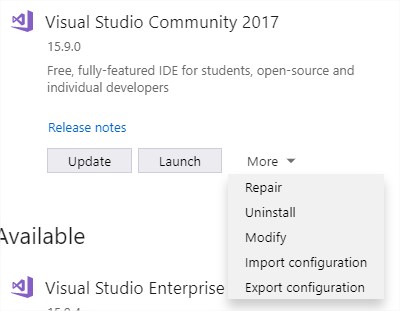
pav. FlexPendant

# .NET sąsaja su IRC5

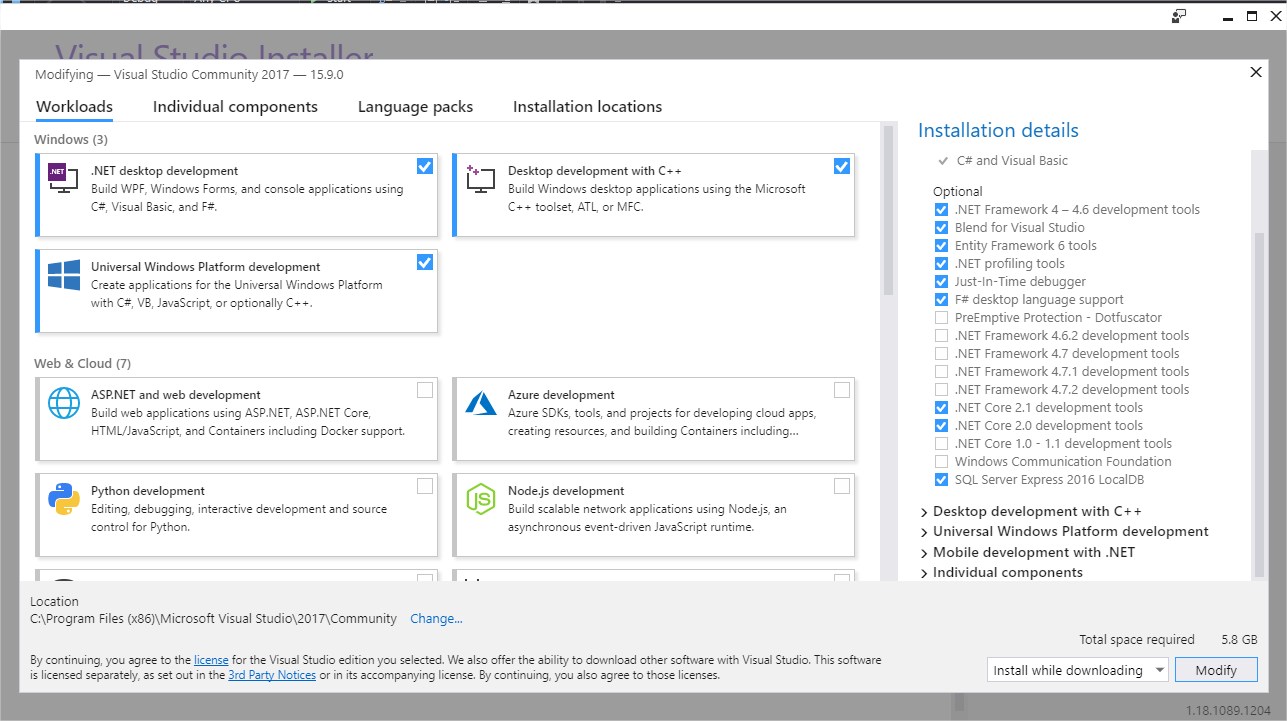
Norint susikurti .NET sąsaja su IRC5 valdikliu pirmiausiai reikia turėti VisualStudio su reikiamais paketais. Jei VisualStudio suinstaliuota ją galima modifikuoti:



pav. Visual Studio installer



pav. Spaudžiama modify



pav. VisualStudio paketų langas

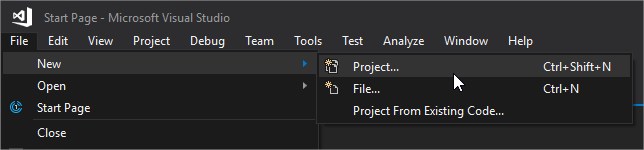
Butiniausias paketas:



pav. Paketas leidžiantis WindowsForms aplikacijas

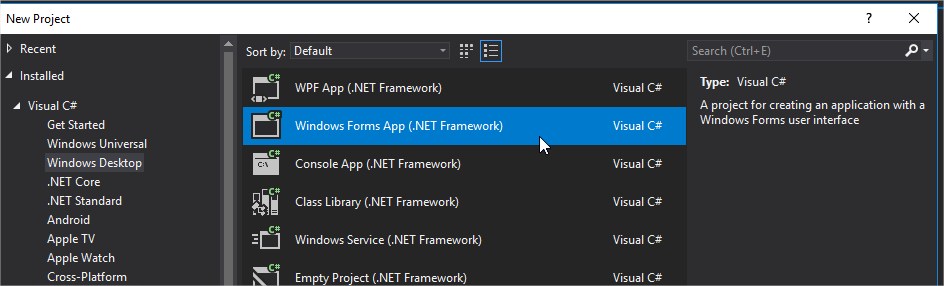
Tada spaudžiama „Modify“ ir laukiama kol suinstaliuos.

Suinstaliavus paketus reikia VisualStudio aplinkoje susikurti WindowsForms aplikaciją:



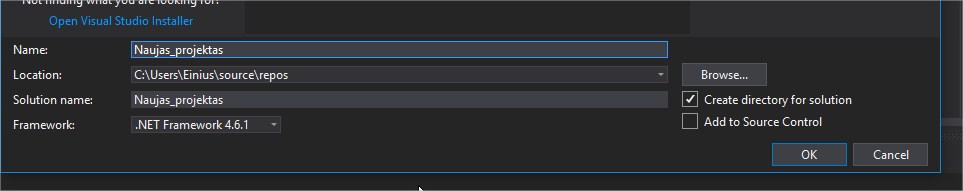
pav. Naujas projektas

Pasirenkame VisualC#/Windows Desktop/WindowsForm App:



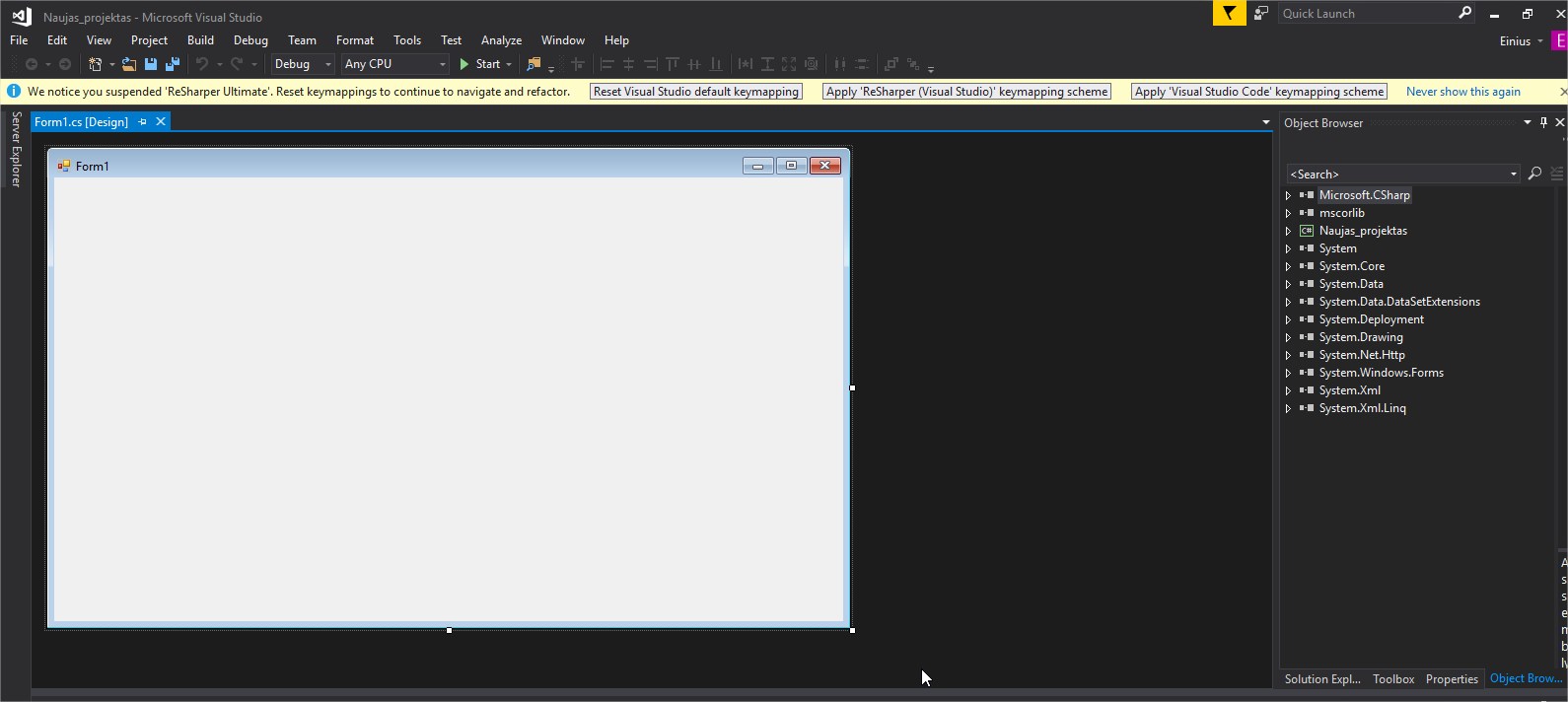
pav. WindowsForms aplikacija

Ir parašome pavadinimą:



pav. Parenkamas pavadinimas

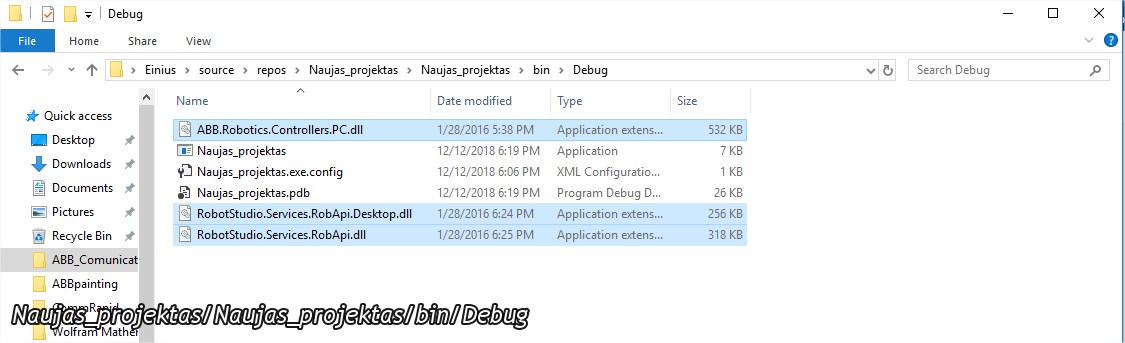
Paspaudus „OK“ atsidaro naujas projektas:



pav. Dizaino langas

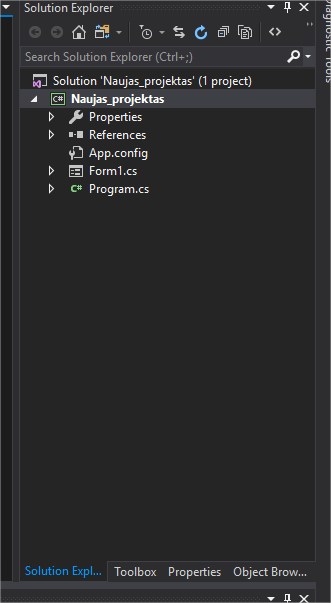
Debug režime subuildiname projektą paspausdami „Start“ ir išmes tuščia Form1 langą.

Toliau reikia susidėti ABB dinamines bibliotekas, kurios bus pateiktos pavyzdiniame projekte, į projekto Debug direktoriją(Pav 39 komentaras yra direktorijos pavyzdys):



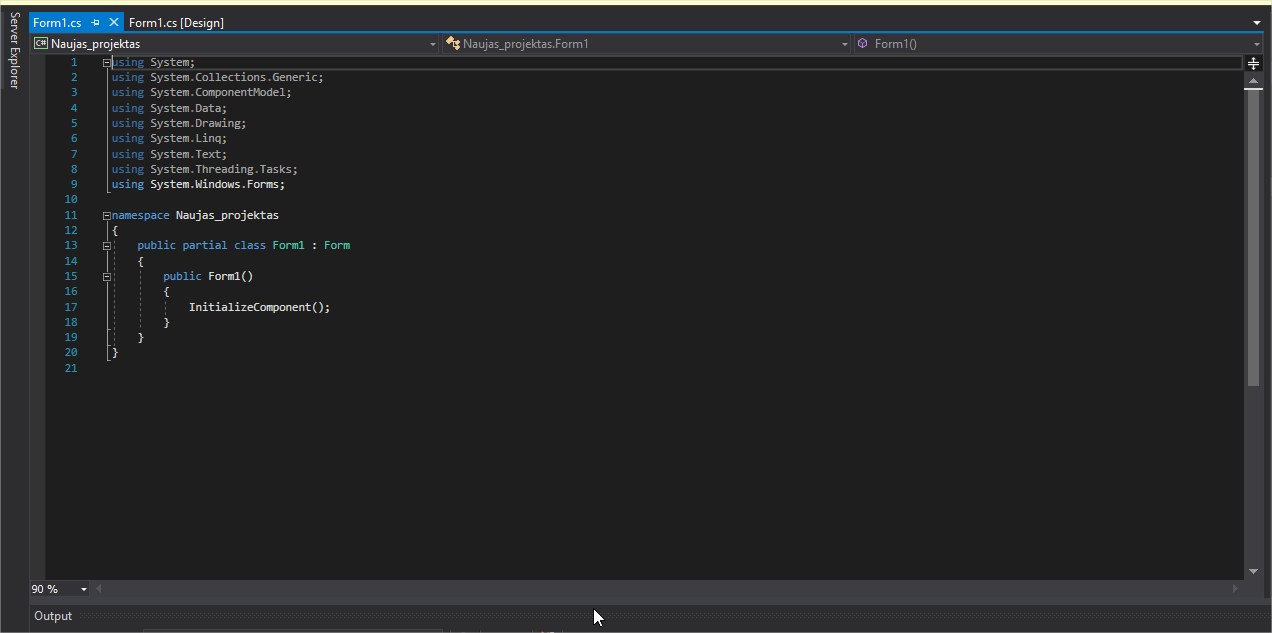
pav. Projekto direktorija

Susidėjus dinamines bibliotekas reikia jas įtraukti į kodą. Nuėjus į „Solution explorer“:



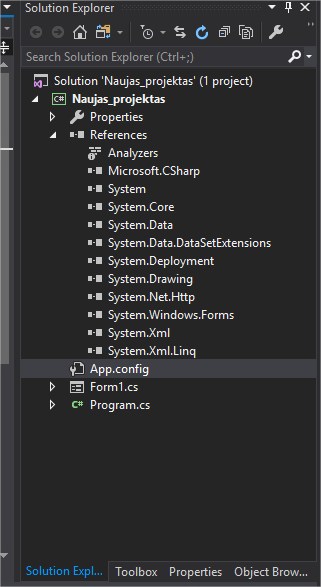
pav. Solution Explorer

Ir paspaudus kairiu pelės klavišu ant „Form1“ ir tada F7, atsiranda „Form1“ kodo skiltis:



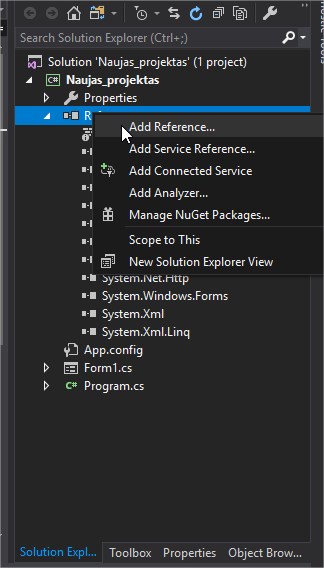
pav. Form1.cs kodas

Nueiname į „references“ skiltį:



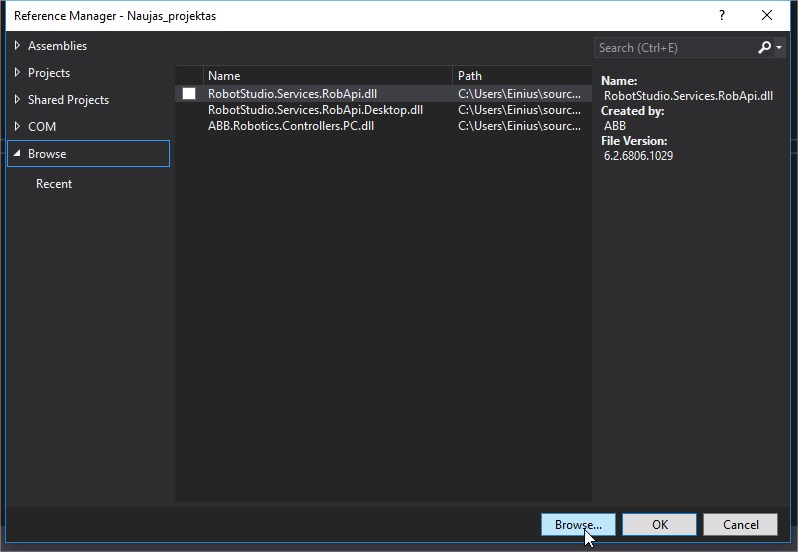
pav. Referances skiltis

Paspaudžiame dešiniu pelės klavišu ir „Add reference“:



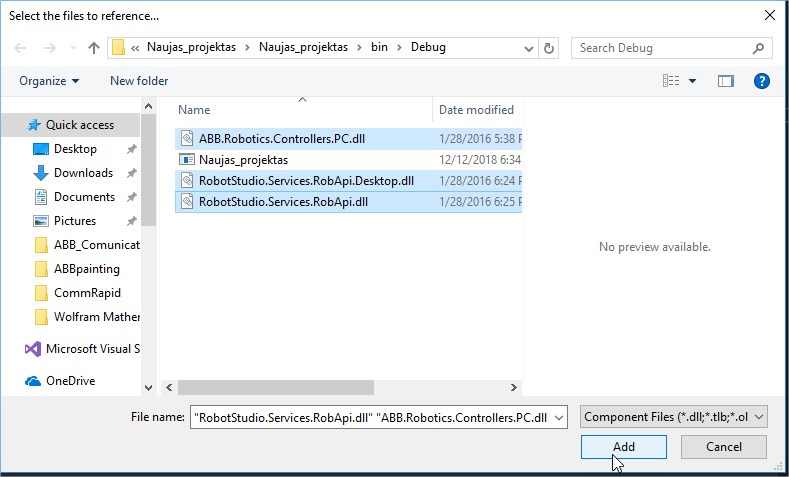
pav. Naujos nuorodos pridėjimas

Tada „Reference Manager“ lange pasirenkame „Browse...“:



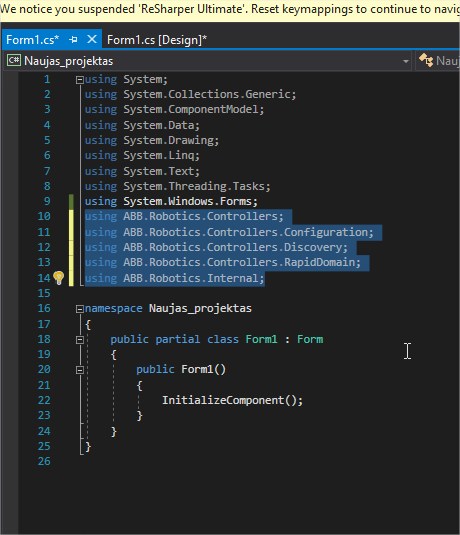
pav. Reference manager

Ir nuėję į direktorija, kurioje yra sukeltos dinaminės bibliotekos, jas pasižymim ir spaudžiame „ADD“ ir tada „OK“:



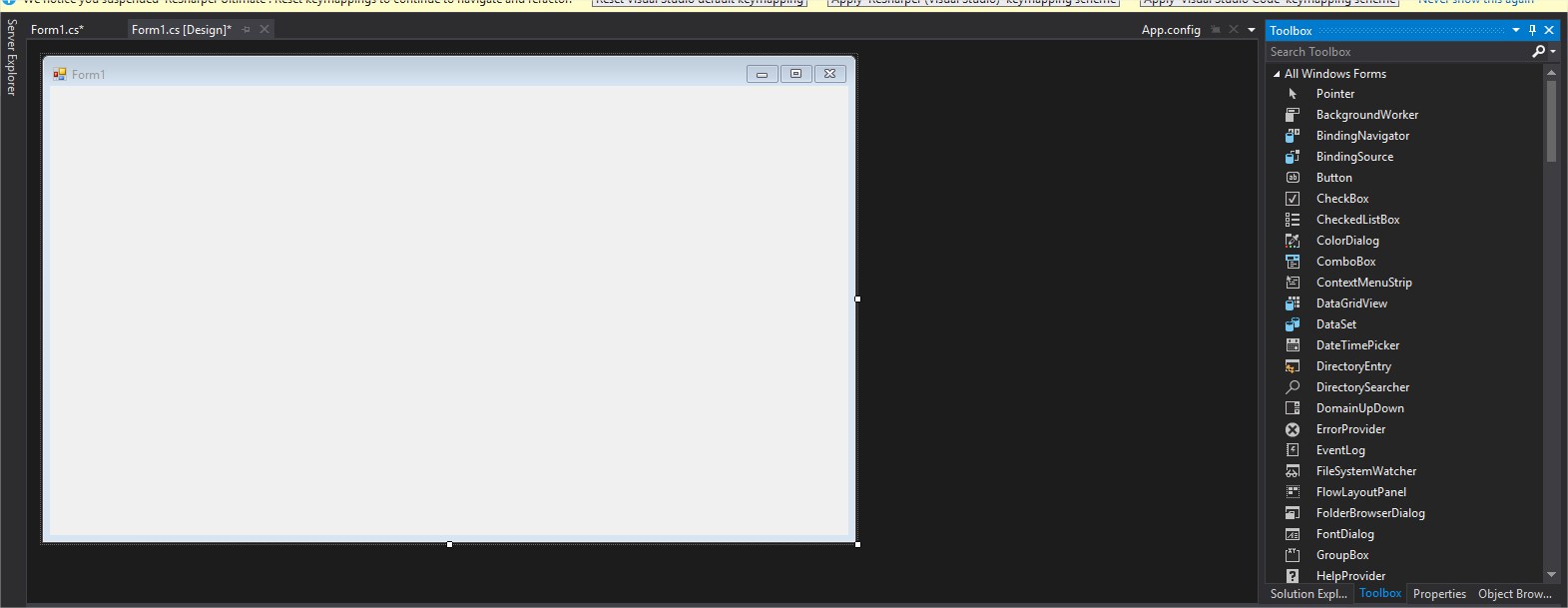
pav. Dll pasirinkimas

Tada „Form1“ kode naudojantis „using“ direktyva pridedame šias bibliotekas:

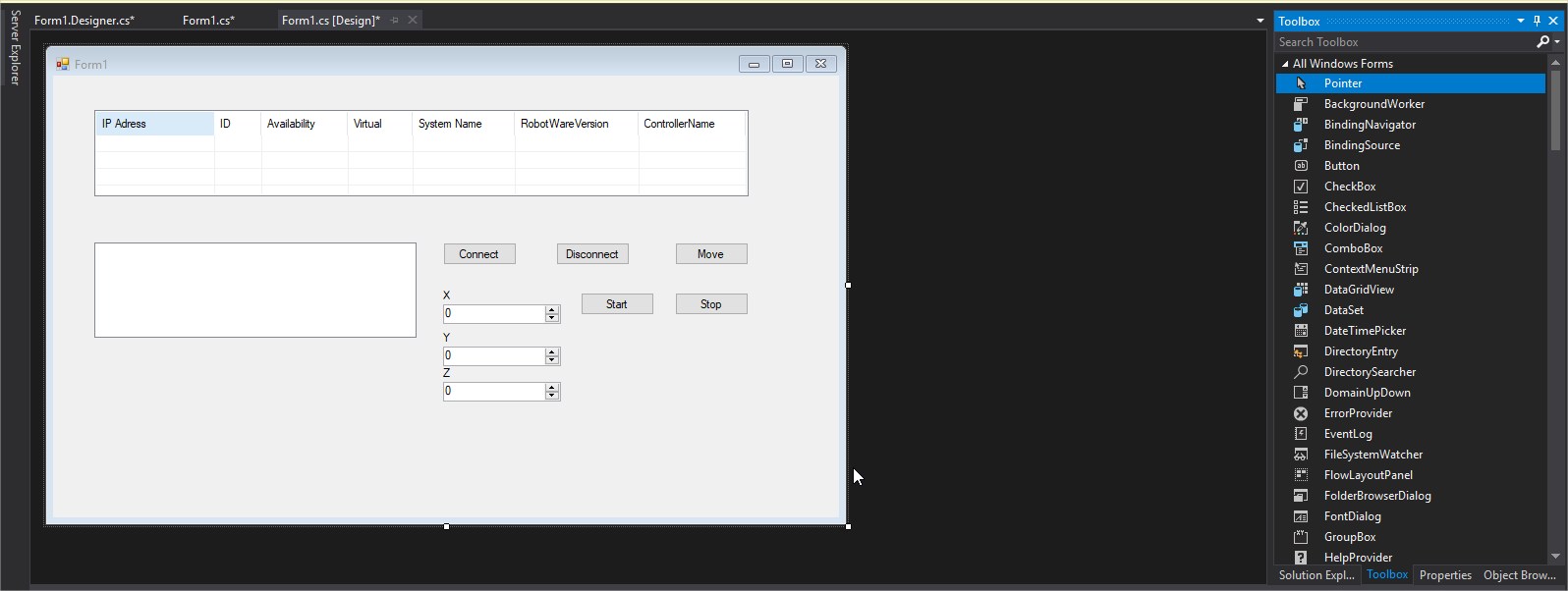


pav. Dll įtraukimas į Form1.cs kodą

Tada „Form1 [Design]“ skiltyje susikuriame „listview“ ir kelis „button“ bei kelis „numericUpDown“ objektus naudojantis „toolbox“ skiltimi. Tai padaroma nuėjus į pastarąją skiltį ir pažymejus „Form1“ langa su kairiu pelės klavišu, įjungiame „toolbox“ skilty ir per „search toolbox“ sudeliojame komponentus:

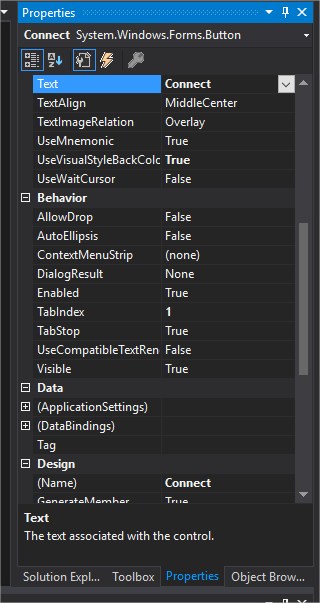


pav. Prieš keitimą



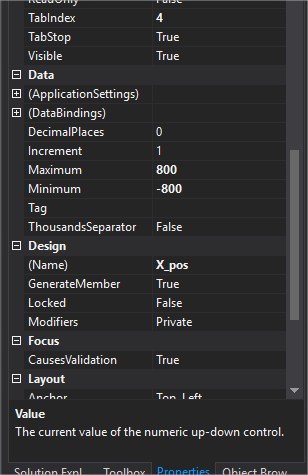
pav. Po keitimo

Kiekvieną „button“ galima modifikuoti paspaudus ant jo ir nuėjus į skiltį properties, kuriame reikia pakeisti „button“ objekto „Text“ ir „Design (Name)“ į tokius, kokie pavaizduoti 48 Pav:



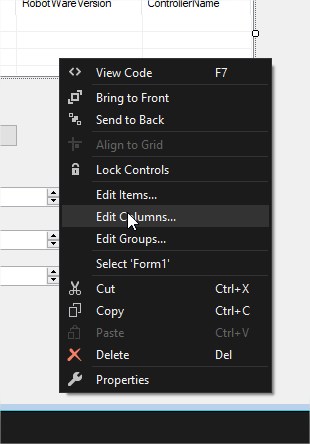
pav. Properties

O „NumericUpDown“ objektų „Design (Name)“ reikia keisti į X\_pos, Y\_pos, Z\_pos nuo viršaus žemyn, o „Data maximum minimum“ kaip pavaizduota pav 50:



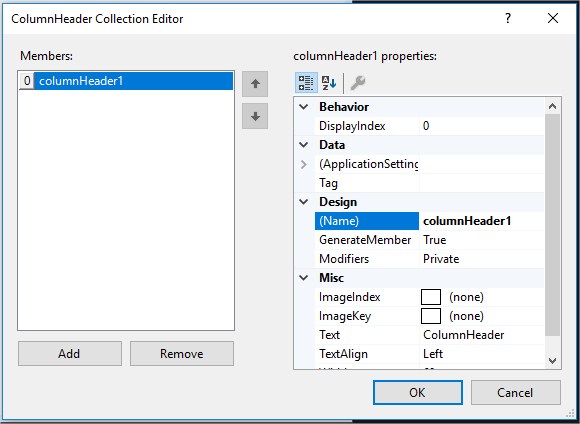
pav. NumericUpDown properties

„Listview“ objektus modifikuoti reikia paspaudus ant jo kairiu, o po to dešiniu pelės klavišu ir tada „Edit Columns..“:



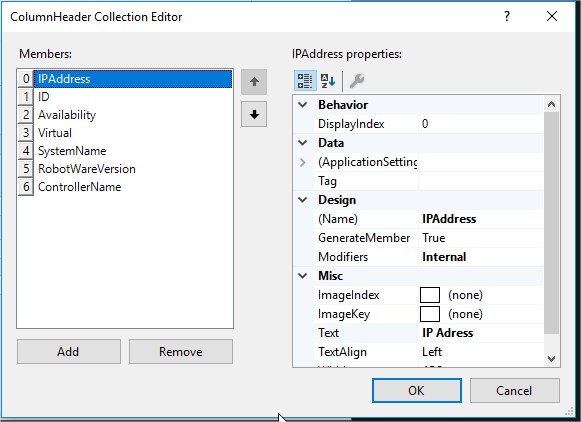
pav. ListView modifikavimas

Išmestame lange spaudžiame „Add“ ir gauname naują narį:



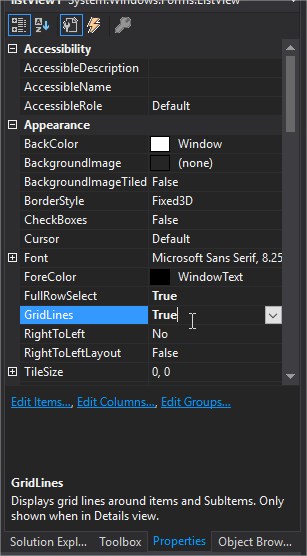
pav. Editor prieš keitimą

Prie kiekvieno (Name) ir Text parašome atitinkamus pavadinimus kol gausis:



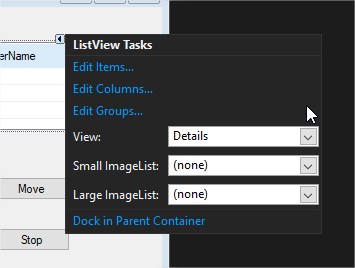
pav. Editor po keitimo

Tada „listView“ objekto properties pasirenkame „GridLines -> True“:



pav. ListView parametruose įjungiamos linijos

Ir paspaudus ant „listView“ objekto dešiniajame viršutiniame kampe atsiranda trikampis kvadrate simbolis, jį paspaudus pakeičiame View į Detais:



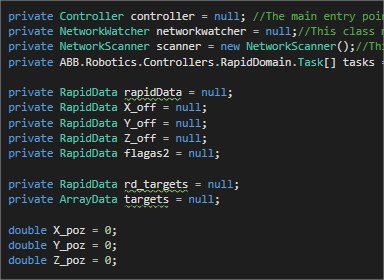
pav. Tasks

Padarius šiuos žingsnius inicializuojame „button“ ir „numericUpDown“ objektus dukart paspaudus ant jų ir galima tada nusikopijuoti pavyzdinį kodą iš ABB\_Comunication projekto Form1.cs ir pelkelti jį į savo Form1.cs ištrynus prieš tai viską. Taipogi galima tiesiog pasileisti pavyzdinį kodą ir tęsti nuo čia.

# Pagrindinės kodo funkcijos

C# programos kodas susideda IRC5 valdiklio ryšio užmezgimo ir nutraukimo, ir atskirų komandų davimo ir kintamųjų keitimo funkcijų.

Pagrindiniai objektai naudojami pavaizduoti pav 56:



pav. Kintamieji

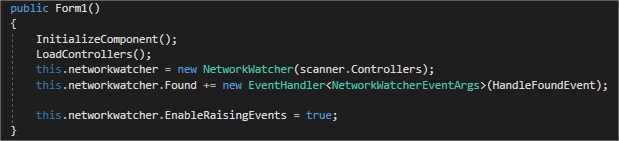
„Controller“ objektas yra surišamas su IRC5 valdikliu ir naudojantis šia klase galima išgauti ir pakeisti duomenis esančius valdiklyje.

„NetworkWatcher“ ir „NetworkScanner“ reikalingi surasti ir užmegzti ryšį su valdikliu.

„Task“ klasė apibūdina RAPID programą esančią IRC5 valdiklyje.

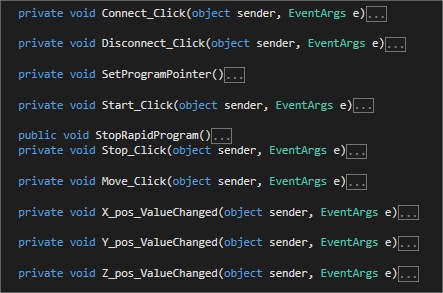
„RapidData“ apibūdina RAPID kodo duomenis, o „ArrayData“ duomenų masyvus.

Programa prasideda „Form1“ komponentų inicializavimu(InitializeComponent()) ir esamų valdiklių tinkle užkrovimu(LoadControlers()):



pav. Form1 konstruktorius

Paskui yra aprašomos „button“ inicializuotos funkcijos, kurios reguoja, kai programos veikimo metu yra paspaudžiami šie pastarieji objektai:



pav. Form1 funkcijos

„Connect\_Click“ funkcija sujungia „Form1“ aplikacija su pasirinktu valdikliu, šiuo atveju IRC5.

„Disconnect\_Click“ funkcija atjungia esamą valdiklį nuo „Form1“ aplikacijos.

„SetProgramPointer“ nustato RAPID programos pointerį į aprašytos užduoties(„Task“) modulio eilutę.

„Start\_Click“ pradeda valdiklio RAPID programą.

„Stop\_Click“ sustabdo valdiklio RAPID programą. „StopRapidProgram“ yra pastarojoje funkcijoje.

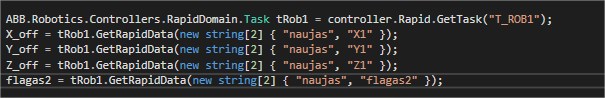
„Move\_Click“ siunčia (X,Y,Z) ir flagą valdikliui, kuris panaudoja flagą įjungti RAPID programos procesą, kuris duoda komandą manipuliatoriui pajudėti (X,Y,Z) milimetrų iš esamos pozicijos.

„X\_Pos\_ValueChanged“ ir visos likusios funkcijos siejamos su „NumericUpDown“ objektais „Form1“ lange ir pakeičia kintamuosius (X,Y,Z), kurie yra naudojami „Move\_Click“ funkcijoje.

# Duomenų modifikavimas

Kai .NET aplikacija yra įjungta, sujungta su IRC5 valdikliu, ir RAPID programa paleista naudojantis „Start\_Click“ funkcija, ir įjungti valdiklio motorai, galima naudotis „Move\_Click“ funkcija.

„Move\_Click“ funckcija prasideda:



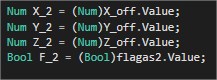
pav. Duomenų gavimas

„Task tRob1“ kintamajam yra priskiriamas „T\_ROB1“ RAPID programa esanti valdiklyje.

„RapidData X\_off“ kintamajam yra priskiriamas „X1“ kintamasis esantis „naujas“ modulyje, esančiame „T\_ROB1“ RAPID programoje.

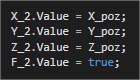
Likę kintamieji analogiški praeitam.

„RapidData“ kintamuosius reikia „castinti“ į tipus, kokiais jie yra RAPID programoje ir priskirti lokaliems ar globaliems kintamiesiems:



pav. Castinimas

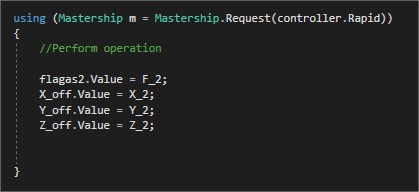
Šiuos naujus kintamuosius galima modifikuoti:



pav. Modifikavimas

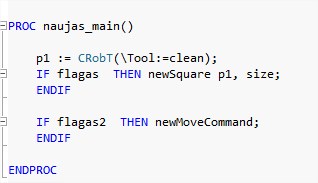
„X\_poz, Y\_poz, Z\_poz“ yra „NumericUpDown“ objekto vertės.

Tada atlikus norimas modifikacijas, galima kintamuosius persiųsti atgal į valdiklį:



pav. Persiuntimas į valdiklį

RAPID programos procesai yra vykdomi per if sąlygą:



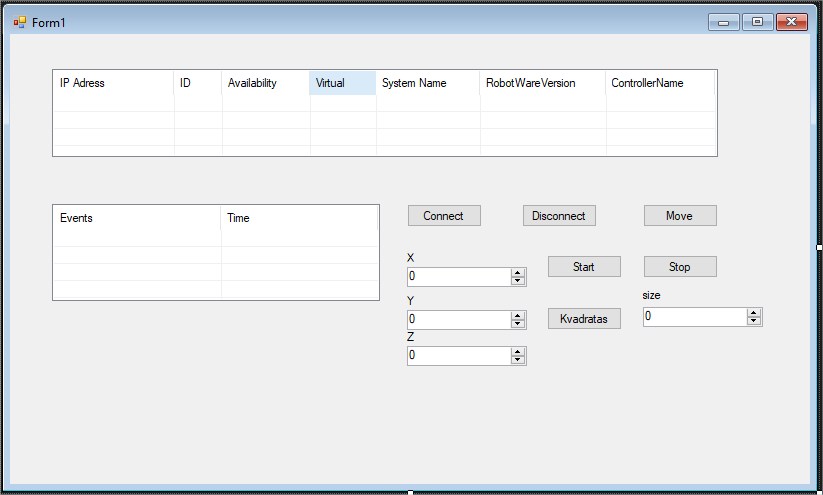
pav. RAPID procesas

Kai RAPID programa yra vykdoma, ji suka nuolatinį ciklą, jei „RunMode“ yra nustatytas „Continuos cycle“ ir „IF“ sąlyga yra nuolatos tikrinama.Kai „Form1“ lange yra paspaudžiamas „Move“ mygtukas yra pradedama „Move\_Click“ funkcija, kuri modifikuoja „flagas2“ kintamąjį į „true“, del ko sąlyga patampa teisinga ir yra vykdomas „newMoveCommand“ procesas.

# Pavyzdinė užduotis

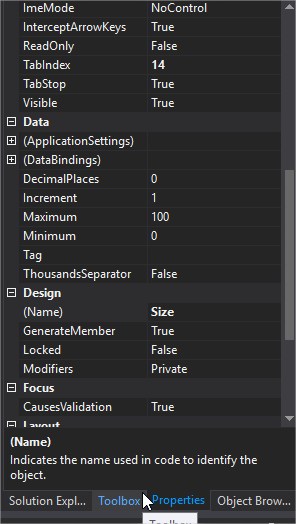
Reikia užprogramuoti valdiklį, kad manipuliatorius nusileistų 20 milimetrų, nubraižytų pasirinkto kraštinės dydžio kvadratą ir vėl pakiltų 20mm į pradinę poziciją.

.NET platformos projekte, „Form1“ lange susikuriame naują „button“ ir „NumericUpDown“ objektus:



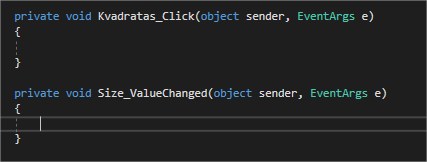
pav. Form1 Langas

Ir pakeičiame „Text“ ir „Design (Name)“ į „Kvadratas“ „button“ objekte ir „Design (Name)“ į „Size“ „NumericUpDown“ objekte.



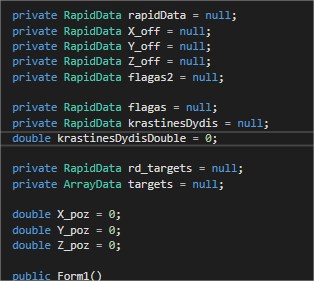
pav. „NumericUpDown“ parametrai

Inicializuojame objektus dukart nuspaudžiant ant jų su kairiu pelės klavišu:



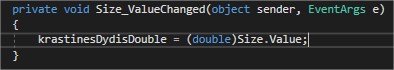
pav. Inicializuoti objektai Form1.cs

Apibūdiname naujus kintamuosius “flagas”, “krastinesDydis” ir “krastinesDydisDouble”:



pav. Naujų kintamųjų apibūdinimas

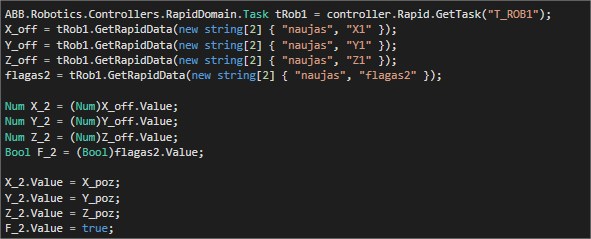
„Size\_ValueChanged“ funkcija apibūdiname:



pav. Size\_ValueChanged funkcija

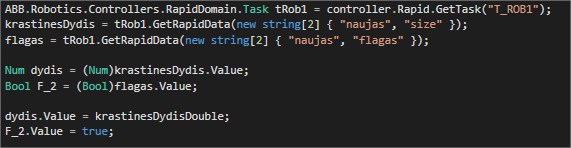
Kai „NumericUpDown“ objektas yra keičiamas, jo vertė bus priskirta „krastinesDydisDouble“ kintamajam. „Castingas“ reikalingas, kadangi „Size.Value“ yra int tipo kintamasis, o „krastinesDydisDouble“ yra double tipo.

Iš „Move\_Click“ funkcijos nukopijuojame viską į „Kvadaras\_Click“ funkciją,



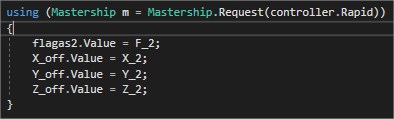
pav. Move\_Click funkcija

ir pakeičiame kodą atitinkamai kaip parodyta:



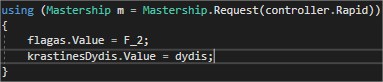
pav. Kvadratas\_Click funkcija

Taipogi pakeičiame using direktyvos kodo dalį:



pav. Using direktyva Move\_click funkcijoje

Keičiame į:

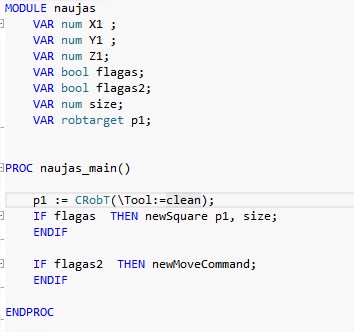


pav. Using direktyva Kvadratas\_click funkcijoje

Atlikus šiuos žingsnius, įsijungiame RobotStudio ir prisijungiam IRC5 Valdiklį.

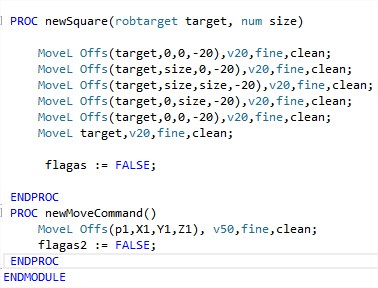
RAPID programos kodas yra toks pats kaip praeituose skyriuose, tačiau su keliais pakeitimais.

Nusirašom RAPID programą su pakeitimais, kurie pažymėti:



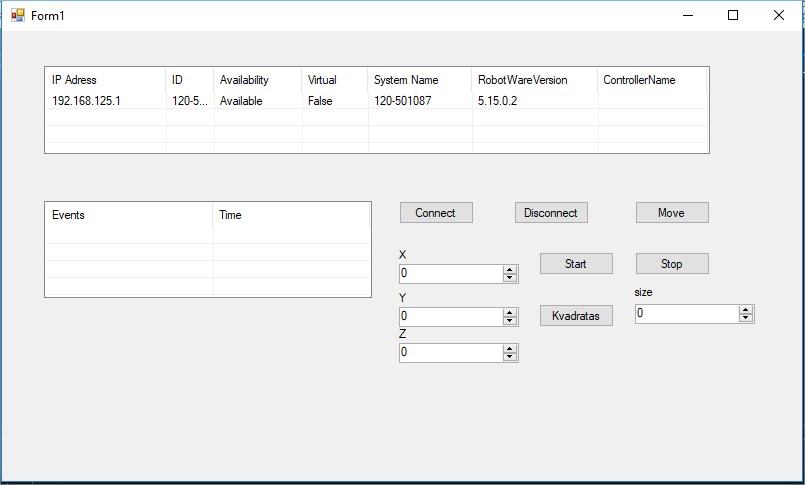
pav. Pirma dalis RAPID kodo

Antra dalis:



pav. Antra dalis RAPID kodo

Atlikus šiuos pakeitimus, ir paspaudus „Apply“ RobotStudio aplinkoje RAPID programa yra įrašoma. Norint paleisti programą, reikia paleisti Form1 aplikacija debug režime:



pav. Form1 langas

Paspaudžiame „Connect“, „Start“, parenkame atitinkamą kvadrato dydį pe „Size“ „NumericUpDown“ objektą ir spaudžiame „Kvadratas“. Manipuliatorius pradės braižyti kvadratą.

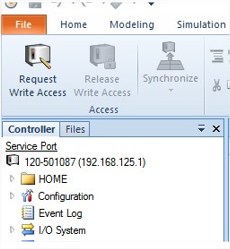
# Rezultatai

Atlikus šį projektą įgaunama žinių naudojantis ABB RobotStudio projektavimo, programavimo aplinka, susipažįstama su RAPID programavimu, C# programavimu bei WindowsForms aplikacijos kurimu.

Sukuriama sąsaja tarp PC ir IRC5 valdiklio, kas leidžia atlikti platesnius duomenų apdorojimus PC platformoje.

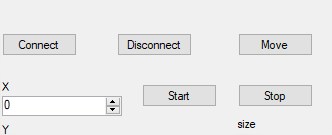
# Galimos problemos

**Pastaba: Dirbant su Windows Forms aplikacija, RobotStudio aplinkoje turi būti atleidžiama įrašymo prieiga prie IRC5 valdiklio, aplikacija galėtų prisijungti prie valdiklio ir apsikeisti duomenimis.**



pav. Meniu juosta

**Pastaba: Jeigu Robotstudio aplinkoje neleidžia įrašyti į IRC5 valdiklį patikrinkite ar Windows Forms aplikacija nera prisijungusi prie IRC5 valdiklio ir atsijunkite nuo jo.**



pav. Atsijungimas nuo valdiklio

Galimos klaidos, kurias gali išmesti Windows Forms aplikacija:

1. "Automatic mode is required to start execution from a remote client." - reikalauja įjungti valdiklį į automatinį režimą, kas padaroma rakto pasukimu pagal laikrodžio rodyklę.
2. "Mastership is held by another client." – reikalauja kitam prietaisui/aplikacijai atsijungti nuo valdiklio.
3. "Unexpected error occurred: " – nežinoma problema. Po dvitaškio parašomas problemos kodas, kurį galima išsianalizuoti internete.

# Priedas A

MODULE naujas

VAR num X1 ;

VAR num Y1 ;

VAR num Z1;

VAR bool flagas;

VAR bool flagas2;

VAR num size;

VAR robtarget p1;

PROC naujas\_main()

p1 := CRobT(\Tool:=clean);

IF flagas THEN newSquare p1, size;

ENDIF

IF flagas2 THEN newMoveCommand;

ENDIF

ENDPROC

PROC newSquare(robtarget target, num size)

MoveL Offs(target,0,0,-20),v20,fine,clean;

MoveL Offs(target,size,0,-20),v20,fine,clean;

MoveL Offs(target,size,size,-20),v20,fine,clean;

MoveL Offs(target,0,size,-20),v20,fine,clean;

MoveL Offs(target,0,0,-20),v20,fine,clean;

MoveL target,v20,fine,clean;

flagas := FALSE;

ENDPROC

PROC newMoveCommand()

MoveL Offs(p1,X1,Y1,Z1), v50,fine,clean;

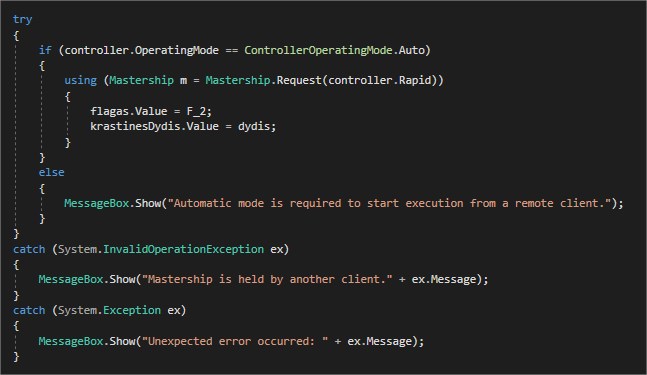
flagas2 := FALSE;

ENDPROC

ENDMODULE

# Priedas B

Visose funkcijose yra naudojamos „try catch“ direktyvos, jos naudojamos „gaudyti“ įvykius „event“ ir atlikti atitinkamus veiksmus, kai jie įvyksta, kaip šiuo atveju išmesti MessageBox langą su atitinkama klaida:



pav. 78 Try catch direktyva