

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

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

**Povilas Rūgys**

**Vilius Zobėla**

**Robotizuotų sistemų modeliavimas**

(T125B151)

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

Projektas

|  |  |
| --- | --- |
|  | **Vadovas**  Lekt. G. Narvydas |

**KAUNAS, 2019**

Turinys

[1. Įvadas 2](#_Toc5723068)

[2. ABB IRB 120 manipuliatorius 2](#_Toc5723069)

[3. IRC 5 Valdiklis 3](#_Toc5723072)

[4. FlexPendant Valdymo Operatorius 4](#_Toc5723073)

[5. .NET sąsaja su IRC5 8](#_Toc5723074)

[6. Pagrindinės kodo funkcijos 21](#_Toc5723075)

[7. Duomenų modifikavimas 23](#_Toc5723076)

[8. Pavyzdinė užduotis 25](#_Toc5723077)

[9. Rezultatai 31](#_Toc5723078)

[10. Galimos problemos 32](#_Toc5723079)

[11. Priedas A 33](#_Toc5723080)

[12. Priedas B 34](#_Toc5723081)

# **Įvadas**

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

Darbo užduotis:

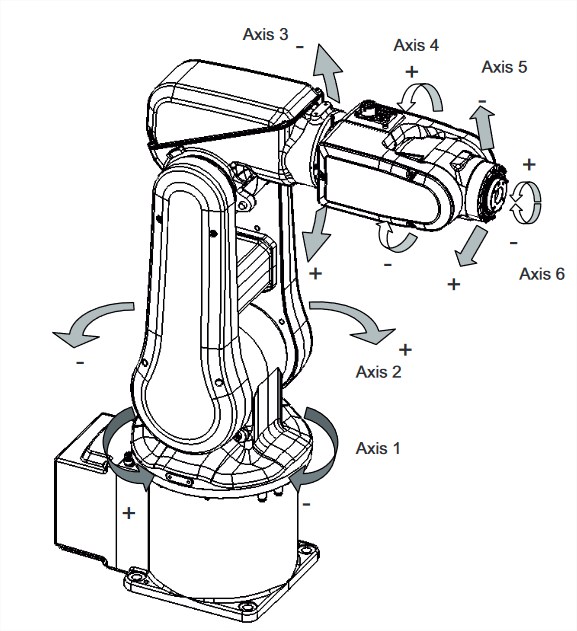
1. Susipažinti su ABB IRB 120 manipuliatoriumi ir ABB IRC 5 valdikliu.
2. Susipažinti su ABB FlexPendant rankiniu operatoriumi.
3. Susipažinti su ABB RobotStudio projektavimo aplinka.
4. Susipažinti su RAPID programavimu.
5. Sudaryti .NET platformos sąsają su IRC 5 valdikliu.
6. Parašyti C# ir RAPID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

Priedą A sudarys naudojamas RAPID kodas.

Priedą B sudarys kai kurių C# metodų 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. 1



# 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

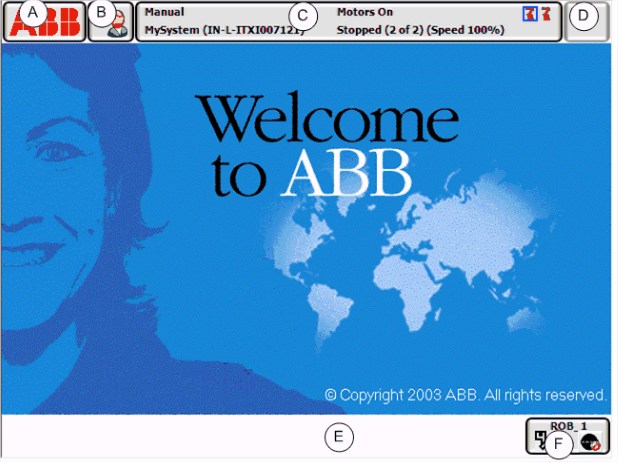
pav. 2 IRC 5 Valdiklis

# FlexPendant Valdymo Operatorius

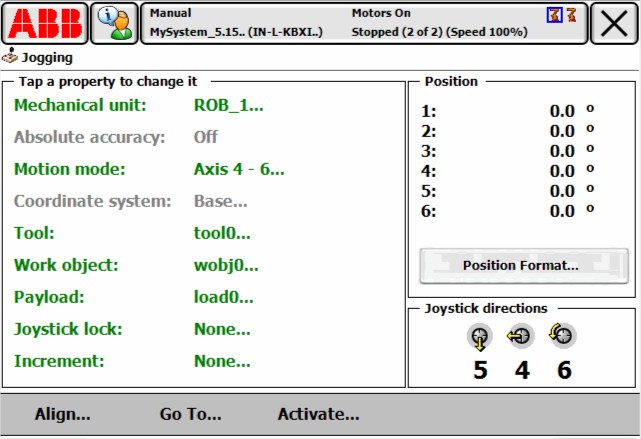
FlexPendant yra rankomis laikomas valdymo prietaisas naudojamas atlikti daugumą užduočių, kontroliuojant robotų sistemą: pradėti programoms, valdyti manipuliatorių, modifikuoti RAPID programas, keisti parametrus. FlexPendant susideda iš techninės ir programinės įrangos ir pats yra kaip atskiras kompiuteris. 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 režimą pasukus raktą pagal laikrodžio rodyklę (Pav.2 D).

Tuomet 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. 3 Pradinis langas



B

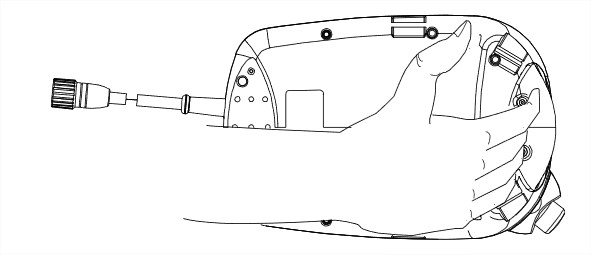
A

C

pav. 4 Jogging parinktys

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

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



pav. 5 Saugiklio įjungimas

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

C



A

B

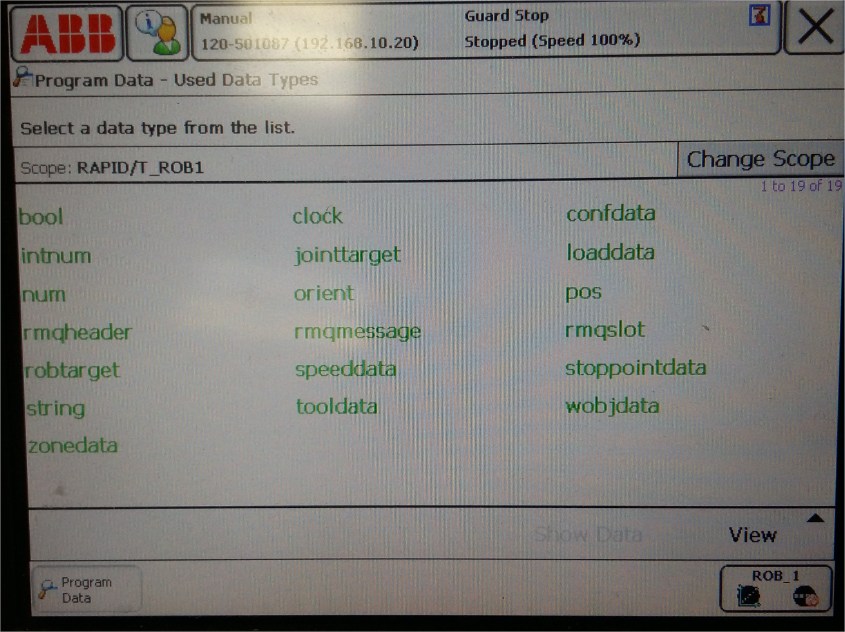
pav. 6 FlexPendant

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

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

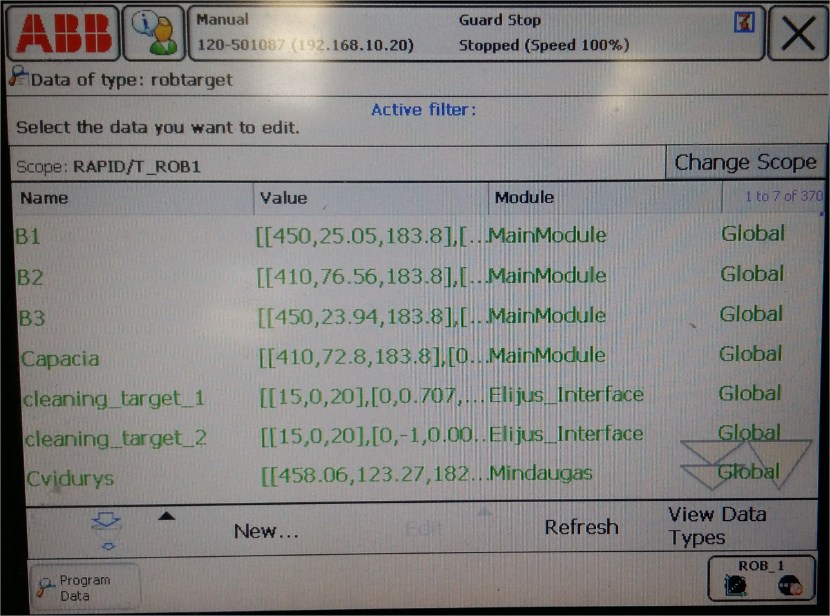
Ant roboto esantis darbo įrankis gali būti perorientuojamas. Jeigu „Tools“ skiltyje yra parinktas „tool0“ (Pav.5 C), kurio koordinatės yra ties 6-ja roboto ašimi, įrankis jį galima sukinėti įjungus reorentacinį režimą „Motion mode“ skiltyje (Pav.5 A) arba FlexPendant pulte paspaudus mygtuką, kuris keičia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



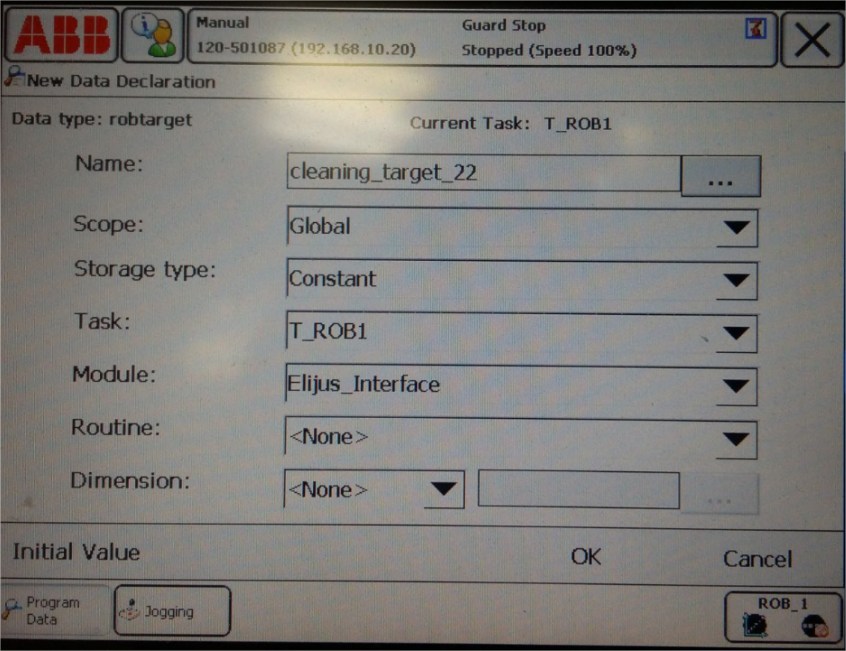
pav. 7

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



pav. 8

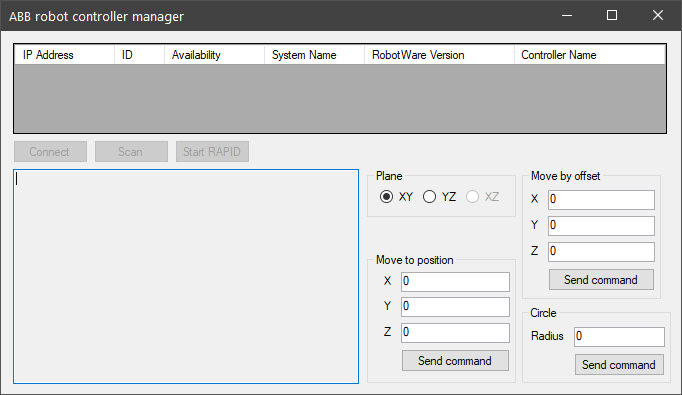
Įėjus matomi visi „robtarget“ kintamieji. Norint susikurti naują, reikia spausti „New...“, parinkti „Name“ ir kitus parametrus, apie kuriuos daugiau bus pasakyta „RAPID Programavimo“ skiltyje.



pav. 9

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

# WinForms vartotojo sąsaja su IRC5

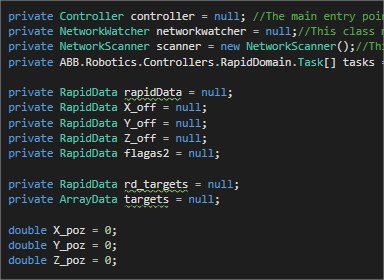


Naudokites lol

# 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. 56 Kintamieji

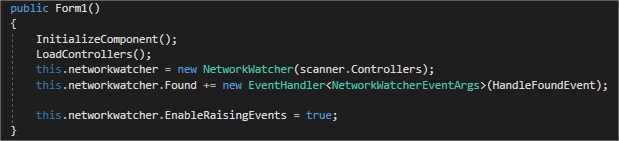
„Controller“ objektas yra surišamas su IRC5 valdikliu ir naudojantis šia klase galima išgauti ir pakeisti duomenis esančius valdiklyje.

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

„Task“ klasė apibūdina RAPID programą esančią IRC5 valdiklyje.

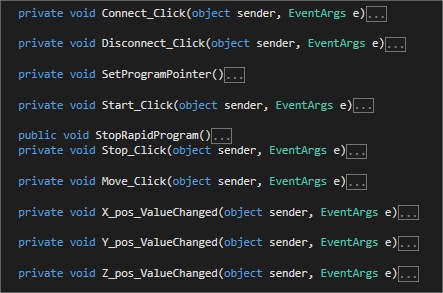
„RapidData“ apibūdina RAPID kodo duomenis, o „ArrayData“ duomenų masyvus.

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



pav. 57 Form1 konstruktorius

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



pav. 58 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.

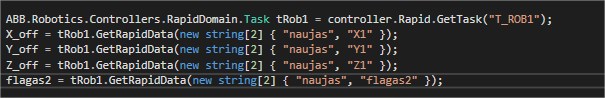
„Move\_Click“ siunčia (X,Y,Z) ir flagą valdikliui, kuris panaudoja flagą įjungti RAPID programos procesą, kuris duoda komandą manipuliatoriui pajudėti (X,Y,Z) milimetrų iš esamos pozicijos.

„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:



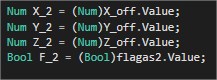
pav. 59 Duomenų gavimas

„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.

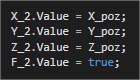
Likę kintamieji analogiški praeitam.

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



pav. 60 Castinimas

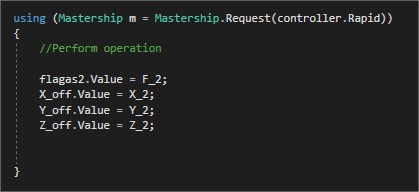
Šiuos naujus kintamuosius galima modifikuoti:



pav. 61 Modifikavimas

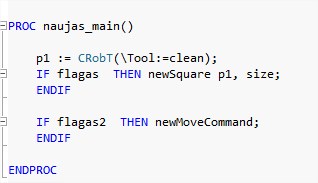
„X\_poz, Y\_poz, Z\_poz“ yra „NumericUpDown“ objekto vertės.

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



pav. 62 Persiuntimas į valdiklį

RAPID programos procesai yra vykdomi per if sąlygą:



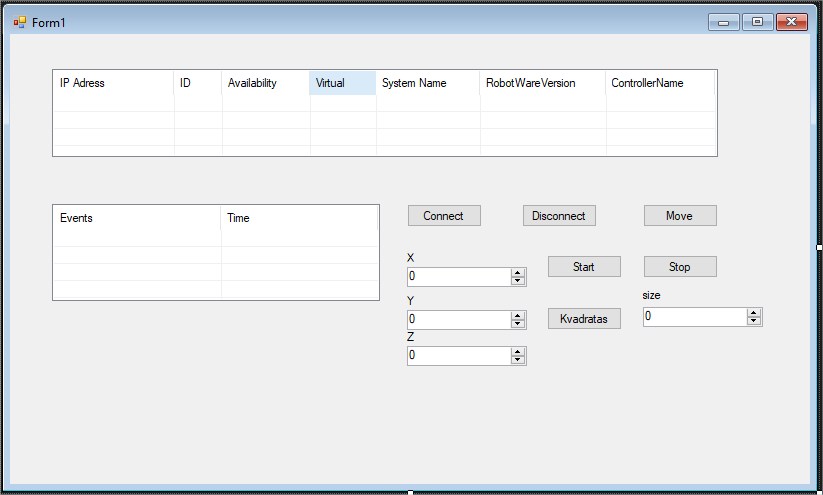
pav. 63 RAPID procesas

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

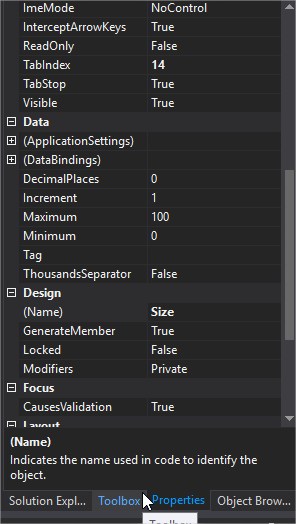
Reikia užprogramuoti valdiklį, kad manipuliatorius nusileistų 20 milimetrų, nubraižytų pasirinkto kraštinės dydžio kvadratą ir vėl pakiltų 20mm į pradinę poziciją.

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



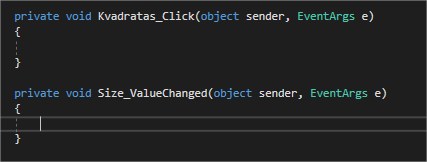
pav. 64 Form1 Langas

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



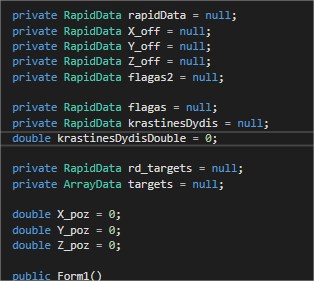
pav. 65 „NumericUpDown“ parametrai

Inicializuojame objektus dukart nuspaudžiant ant jų su kairiu pelės klavišu:



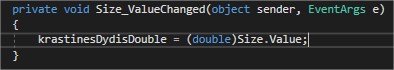
pav. 66 Inicializuoti objektai Form1.cs

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



pav. 67 Naujų kintamųjų apibūdinimas

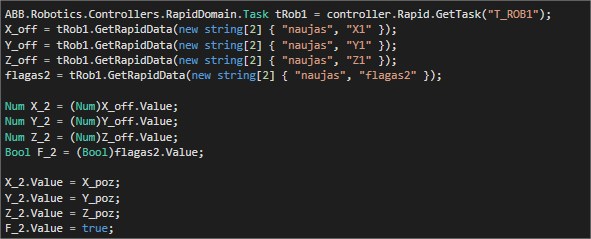
„Size\_ValueChanged“ funkcija apibūdiname:



pav. 68 Size\_ValueChanged funkcija

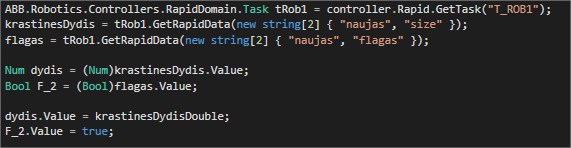
Kai „NumericUpDown“ objektas yra keičiamas, jo vertė bus priskirta „krastinesDydisDouble“ kintamajam. „Castingas“ reikalingas, kadangi „Size.Value“ yra int tipo kintamasis, o „krastinesDydisDouble“ yra double tipo.

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



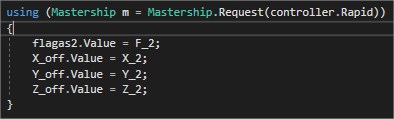
pav. 69 Move\_Click funkcija

ir pakeičiame kodą atitinkamai kaip parodyta:



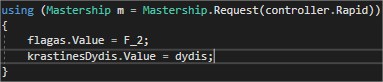
pav. 70 Kvadratas\_Click funkcija

Taipogi pakeičiame using direktyvos kodo dalį:



pav. 71 Using direktyva Move\_click funkcijoje

Keičiame į:

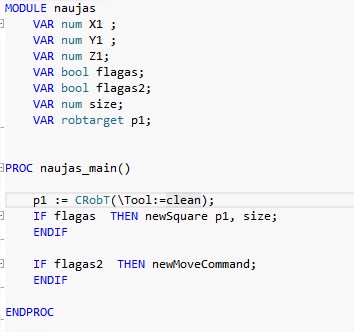


pav. 72 Using direktyva Kvadratas\_click funkcijoje

Atlikus šiuos žingsnius, įsijungiame RobotStudio ir prisijungiam IRC5 Valdiklį.

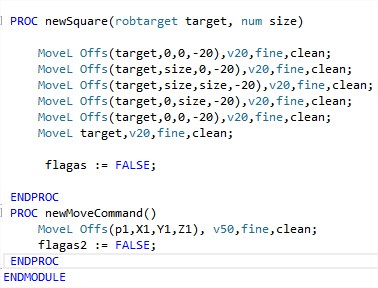
RAPID programos kodas yra toks pats kaip praeituose skyriuose, tačiau su keliais pakeitimais.

Nusirašom RAPID programą su pakeitimais, kurie pažymėti:



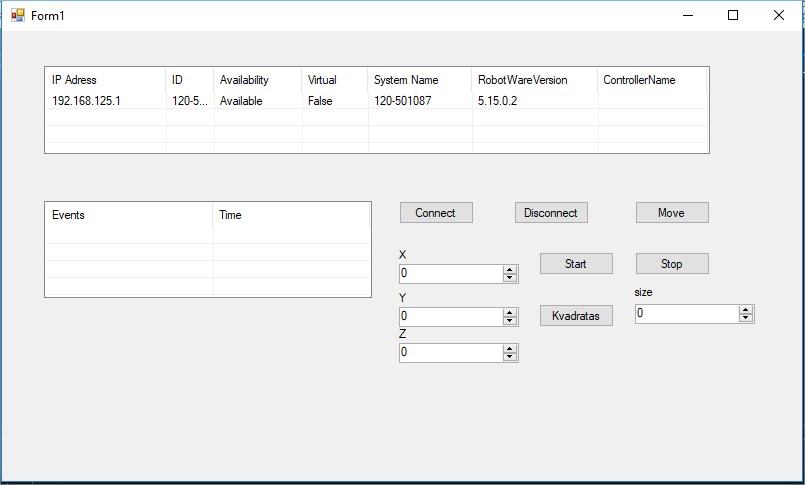
pav. 73 Pirma dalis RAPID kodo

Antra dalis:



pav. 74 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. 75 Form1 langas

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

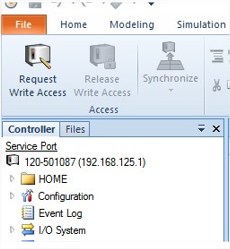
# Rezultatai

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.

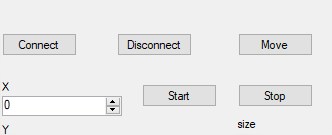
# Galimos problemos

**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. 76 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. 77 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;

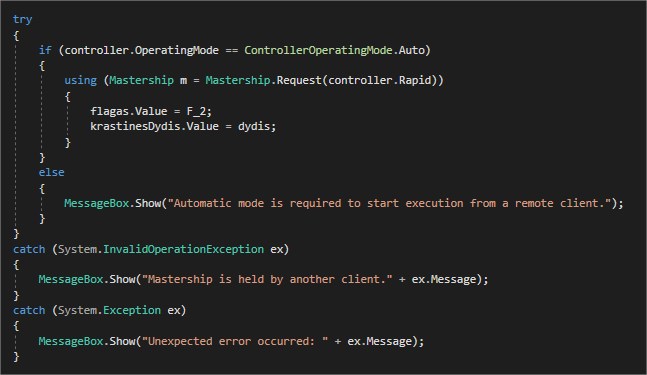
flagas2 := FALSE;

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