Machine and Robot in Harmony

TPS: Teach Pendant setup and configuration

Trio Motion Technology

Trio Robotics Series

First Edition • 2020

Revision: 1.4

Trio Programming Guides are designed to aid learning of the TrioBASIC language through description and examples. Each one will cover a particular topic and discuss which commands and parameters in the TrioBASIC are required to complete the task.

A general understanding of TrioBASIC is required and it is recommended to attend an introduction to TrioBASIC training course. The programming guides are not a replacement for the TrioBASIC help files which can be found in *Motion* Perfect as well as the manual which cover each command and parameter in more detail and should be referenced when required.

Any examples given in the programming guide will work and have been tested on an isolated controller. If you choose to use these examples on a machine please take care that it will not cause damage or injury and that they are correctly included in the project changing parameters and values where required.

All goods supplied by Trio are subject to Trio's standard terms and conditions of sale.

The material in this manual is subject to change without notice. Despite every effort, in a document of this scope errors and omissions may occur. Therefore Trio cannot be held responsible for any malfunctions or loss of data as a result.

UK | USA | CHINA | INDIA

www.triomotion.com

SAFETY WARNING

During the installation or use of a control system, users of Trio products must ensure there is no possibility of injury to any person, or damage to machinery.

Control systems, especially during installation, can malfunction or behave unexpectedly.

Bearing this in mind, users must ensure that even in the event of a malfunction or unexpected behaviour the safety of an operator or programmer is never compromised.

This document uses the following icons for your reference:

💣	📄	★	➡
Information that relates to safety issues and critical software information.	Information to highlight key features or methods.	Useful tips and techniques.	Example programs.

Contents

1 Teach Programming System

1.1 How to add TPS package to a project

To add the TPS to a project, right click on "*Programs*" in the program tree and select "*Add Package…*"

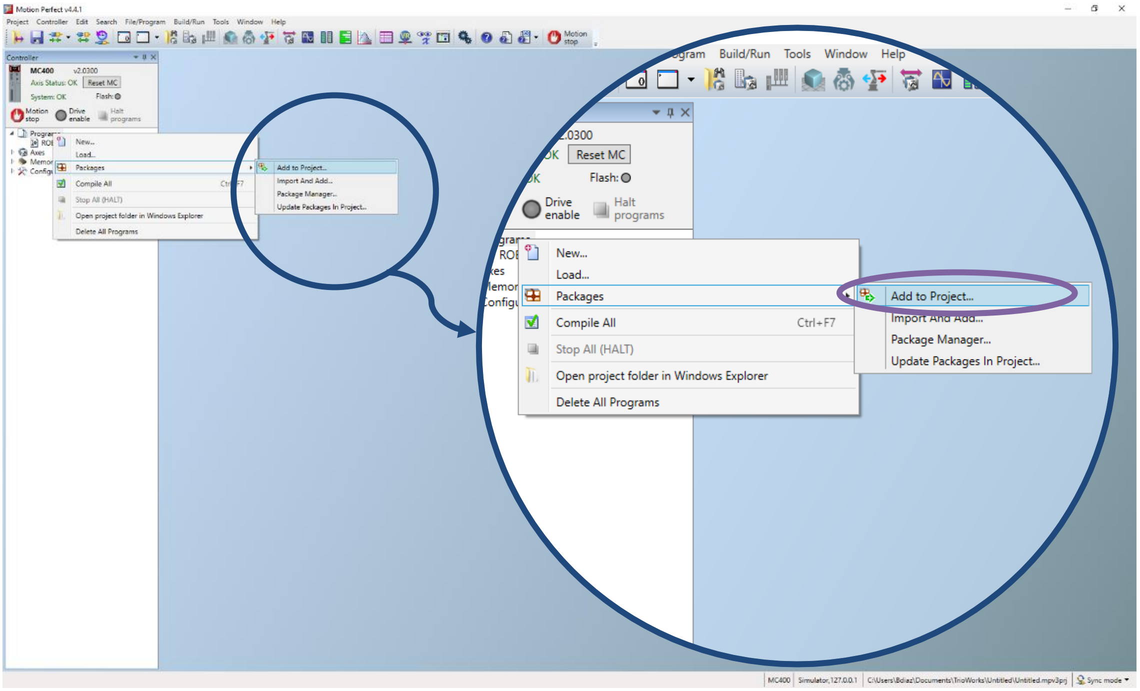
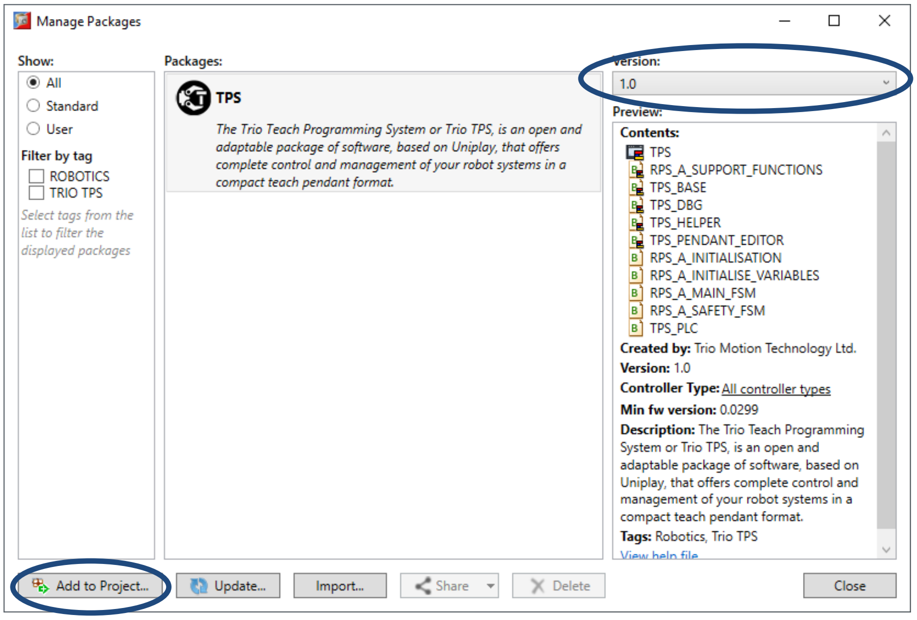


Figure 12. Adding a Package to a project

Select the TPS package and click on "*Add to Project…*".



Show:

● All

○ Standard

○ User

Filter by tag

□ ROBOTICS

□ TRIO TPS

Select tags from the list to filter the displayed packages

Packages:

TPS

The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.

Version: 1.0

Preview:

Contents:

- TPS
- RPS_A_SUPPORT_FUNCTIONS
- TPS_BASE
- TPS_DBG
- TPS_HELPER
- TPS_PENDANT_EDITOR
- RPS_A_INITIALISATION
- RPS_A_INITIALISE_VARIABLES
- RPS_A_MAIN_FSM
- RPS_A_SAFETY_FSM
- TPS_PLC

Created by: Trio Motion Technology Ltd.

Version: 1.0

Controller Type: All controller types

Min fw version: 0.0299

Description: The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.

Tags: Robotics, Trio TPS

View help file

Figure 13. Adding TPS package to a project

1.2 How to update TPS

It is possible to update the system packages in two different ways: automatically through internet or manually with the package file.

The package manager allows for package updates and management. It can be found over Projects in the top menu bar, Packages, Package Manager…

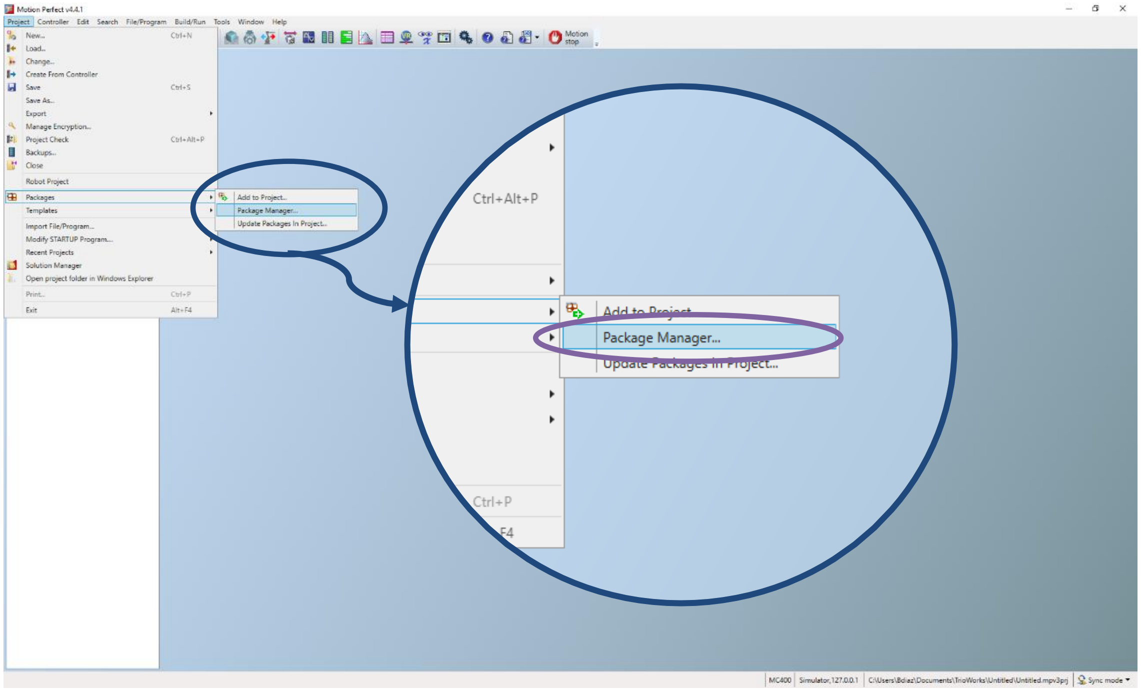
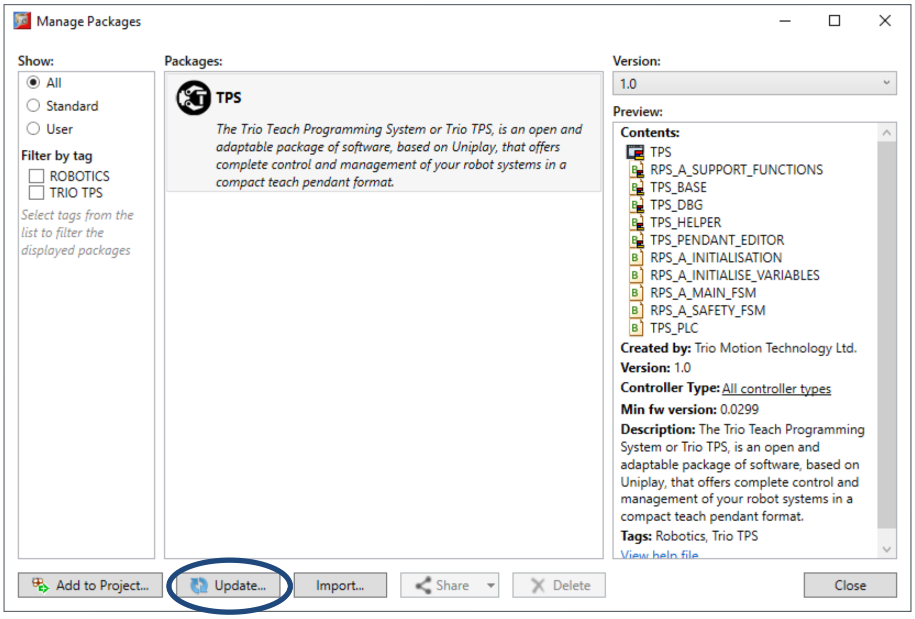


Figure 14. Package manager



TPS

The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.

Version: 1.0

Contents:

- TPS
- RPS_A_SUPPORT_FUNCTIONS
- TPS_BASE
- TPS_DBG
- TPS_HELPER
- TPS_PENDANT_EDITOR
- RPS_A_INITIALISATION
- RPS_A_INITIALISE_VARIABLES
- RPS_A_MAIN_FSM
- RPS_A_SAFETY_FSM
- TPS_PLC

Created by: Trio Motion Technology Ltd.

Version: 1.0

Controller Type: All controller types

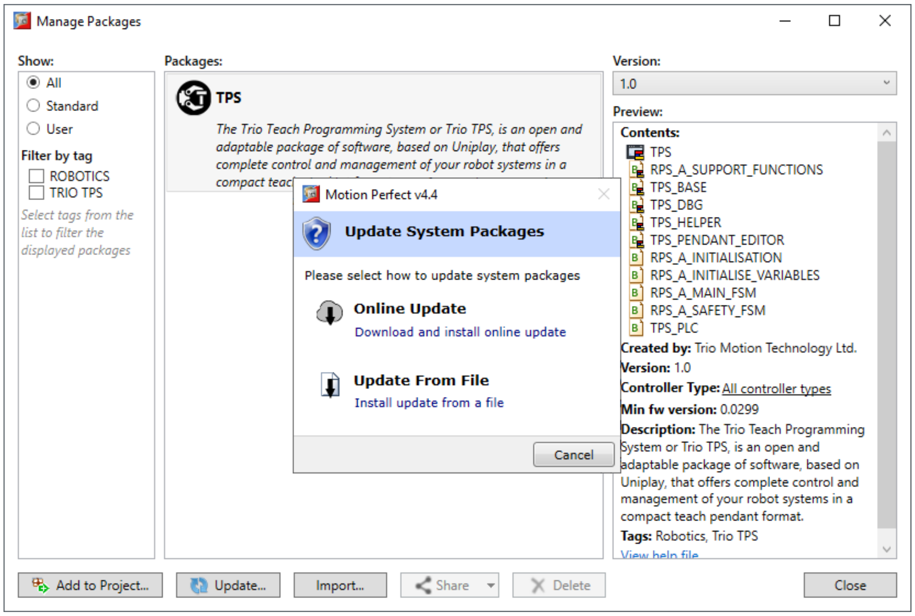
Min fw version: 0.0299

Description: The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.

Tags: Robotics, Trio TPS

View help file

Figure 15. Package manager



Select tags from the list to filter the displayed packages

TPS

The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teac

Please select how to update system packages

Version: 1.0

Contents:
TPS
RPS_A_SUPPORT_FUNCTIONS
TPS_BASE
TPS_DBG
TPS_HELPER
TPS_PENDANT_EDITOR
RPS_A_INITIALISATION
RPS_A_INITIALISE_VARIABLES
RPS_A_MAIN_FSM
RPS_A_SAFETY_FSM
TPS_PLC

Created by: Trio Motion Technology Ltd.
Version: 1.0
Controller Type: All controller types
Min fw version: 0.0299
Description: The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.
Tags: Robotics, Trio TPS
View help file

Figure 16. Package manager

Online update will check for new packages in Trio servers and update if any new ones are available.

To import new packages from files simply click over "*Update From File*" button, select the package and open it.

> *This process will update Motion Perfect with new packages. After Motion Perfect contains the new packages, the project has to be updated.*

To update project package just right click over the package in controller tree and select "*Update Package*".

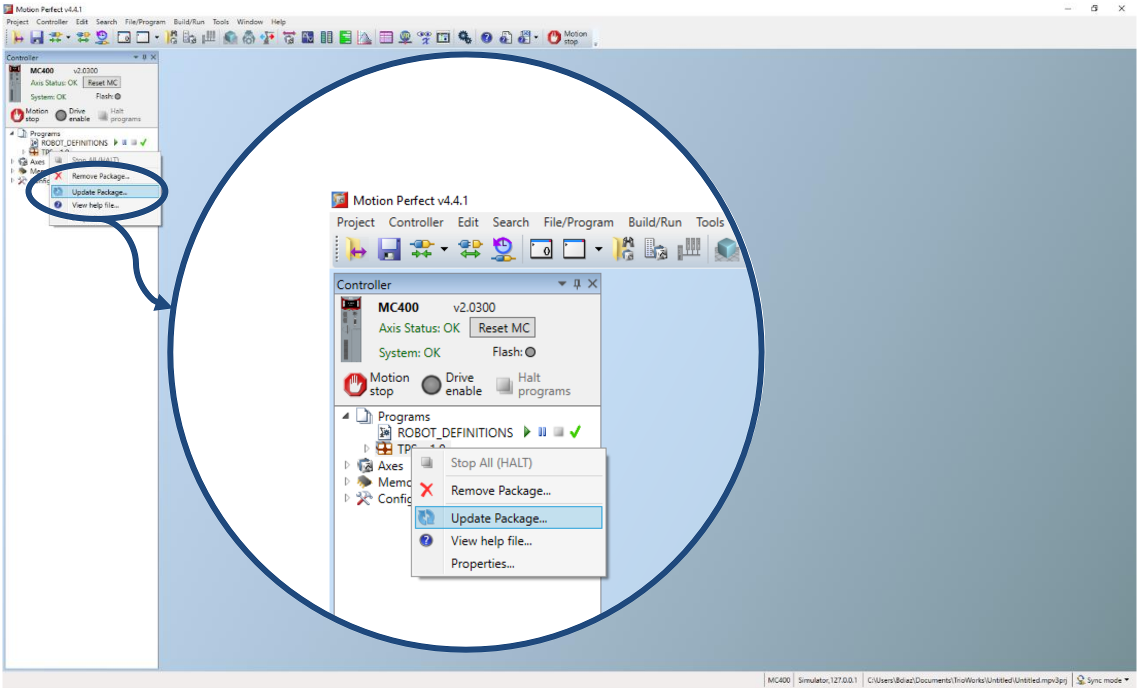
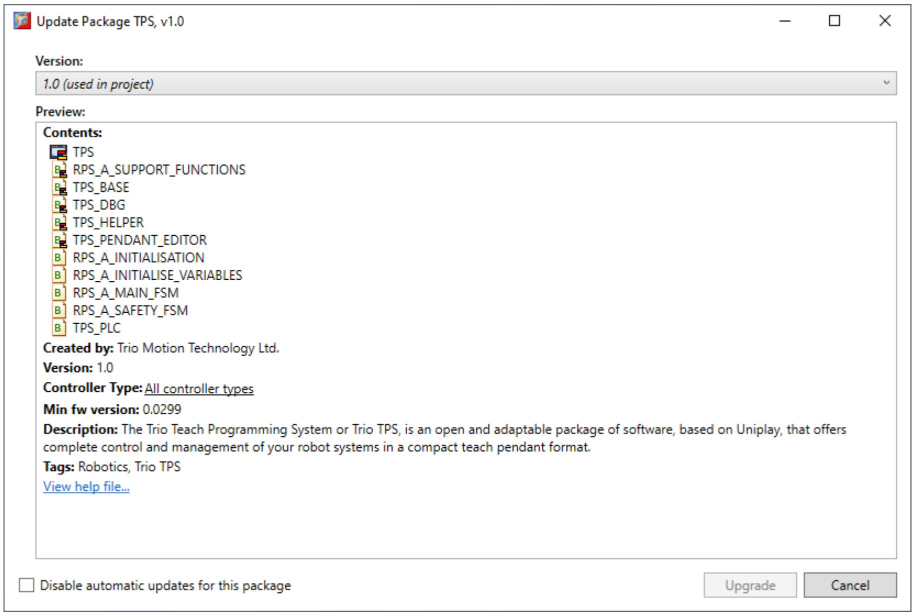


Figure 17. Update package

Then select the new package version and click “Upgrade”.



Update Package TPS, v1.0

Version:

1.0 (used in project)

Preview:

Contents:

- TPS
- RPS_A_SUPPORT_FUNCTIONS
- TPS_BASE
- TPS_DBG
- TPS_HELPER
- TPS_PENDANT_EDITOR
- RPS_A_INITIALISATION
- RPS_A_INITIALISE_VARIABLES
- RPS_A_MAIN_FSM
- RPS_A_SAFETY_FSM
- TPS_PLC

Created by: Trio Motion Technology Ltd.

Version: 1.0

Controller Type: All controller types

Min fw version: 0.0299

Description: The Trio Teach Programming System or Trio TPS, is an open and adaptable package of software, based on Uniplay, that offers complete control and management of your robot systems in a compact teach pendant format.

Tags: Robotics, Trio TPS

View help file...

□ Disable automatic updates for this package

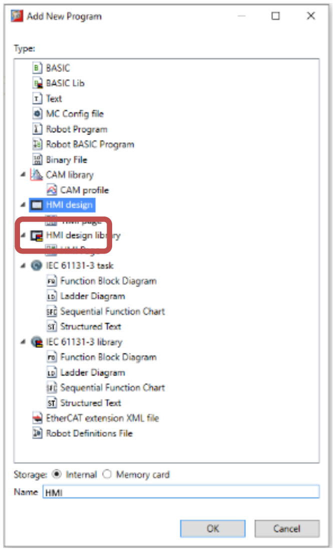
Figure 18. Update package

1.3 How to install and run TPS

1.3.1 Adding an HMI design

TPS requires an HMI design so the pages in the TPS library will be shown.

To create an HMI design, right click over *Programs* in controller tree and select *Add New Program* >> HMI design. Set a name and click Ok.

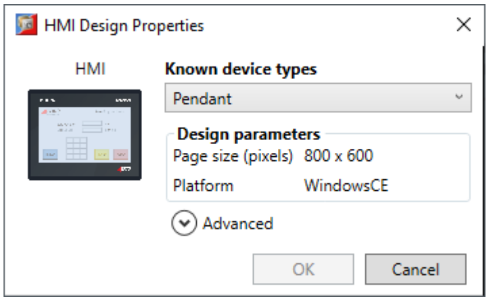


Storage: ● Internal ○ Memory card

Name HMI

Figure 19.a. Add an HMI page to project

HMI design property should be selected as *Pendant.* This window should prompt right after create an HMI design. However, it could be accessed through right click on HMI design in controller tree, properties.



Known device types

Pendant

Design parameters

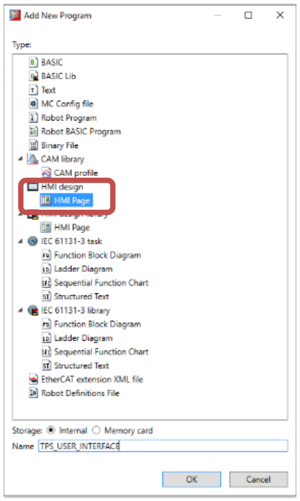
Page size (pixels) 800 x 600

Platform WindowsCE

Figure 19.b. HMI Design properties

1.3.2 Adding an HMI page

TPS requires a specific HMI page for users' interface. To create an HMI page, right click over *Programs* in controller tree and select *Add New Program* >> HMI Page. Name it as "TPS_USER_INTERFACE". Click on '*OK*' to add a page.



Storage: ● Internal ○ Memory card

Name TPS_USER_INTERFACE

Figure 20. Add an HMI page to project

The name of the page has to be specifically "TPS_USER_INTERFACE". TPS will point only to this page for custom user page.

Set the Start-up page to TPS_MAIN.

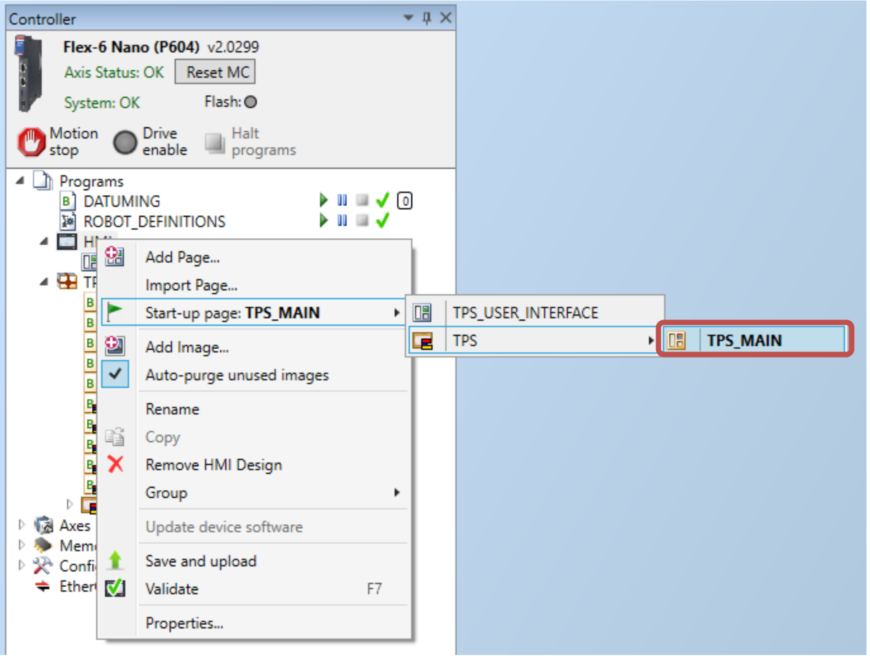


Figure 21. Setting HMI Start-up page

1.3.3 Setting Datuming

In a real system, home position of the robot should be known in encoder absolute values. These values should be written in specific VRs so RPS knows it.

The following code shows a Datuming sample program, setting absolute positions of the encodes for home position.

Create a program called DATUMING and use this code or a modified version of it.

It also initialises EtherCAT network and setup offpos of the real axis based on the read absolute position and home position specified.

Finally, it runs RPS_A_MAIN_FSM, which initialises TPS.

```
INCLUDE "RPS_A_INITIALISE_VARIABLES" ' get access to all the VRs

DIM initialised_ecat AS BOOLEAN
DIM i, j, virtual_axis AS INTEGER

' Move your robot to the home position. To see the position, please refer to
RPS documentation.
' Write the encoder value for each axis at home position into the following
VRs:
VR(absolute_home0) = 38440
VR(absolute_home1) = -18177
VR(absolute_home2) = 97170
VR(absolute_home3) = -3309730
VR(absolute_home4) = 0
VR(absolute_home5) = 0

DISABLE_ROBOT(TRUE) ' Stop all motion and inhibit running of ROBOT and
ROBOT_PROGRAMS

ETHERCAT($22, 0, ecat_status) 'check Ethercat status is not operational
IF VR(ecat_status) <> 3 THEN
    IF VR(debug)=1 THEN
        PRINT #5, "Initialising EtherCat Network"
    ENDIF
    WA(5000) ' wait for the drives to power on
    ETHERCAT(0, 0) ' Initialise Ethercat Network
    WA(200)
    initialised_ecat = TRUE
ENDIF
IF initialised_ecat=FALSE AND VR(debug) = 1 THEN
    PRINT #5, "EtherCat already initialised ... skipping Initialisation"
    PRINT #5, "                                ... skipping homing routine"
ENDIF

KINEMATIC_GROUP(0, -1)
WA(50)
virtual_axis = FALSE
FOR i = VR(real_ax_first) TO VR(real_ax_last)
```

```
    IF ATYPE AXIS(i) = 0 THEN
        IF VR(debug)=1 THEN
            PRINT #5,"[",i,"] Virtual axis detected on one of the axes that
were defined as real axes."
            PRINT #5, "Skipping homing routine"
        ENDIF
        virtual_axis = TRUE
    ENDIF
NEXT i

IF initialised_ecat = TRUE AND virtual_axis = FALSE THEN
    FOR i = VR(real_ax_first) TO VR(real_ax_last)
        j = (i-VR(real_ax_first))
        BASE(i)
        UNITS = 1
        OFFPOS = -VR(absolute_home0+j)
    NEXT i
ENDIF

RUN "RPS_A_MAIN_FSM" 'Initialise TPS
```

1.3.4 Setting Autorun

Set “DATUMING” to autorun on a process (process 0 is recommended).

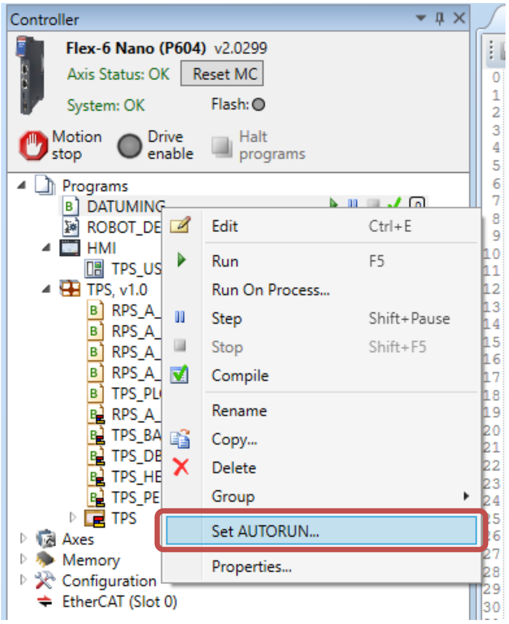
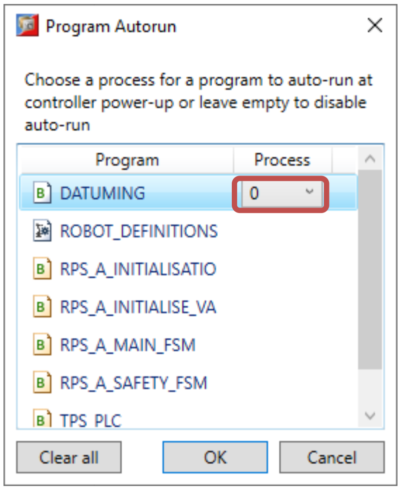


Figure 22. Setting Autorun



Program Autorun

Choose a process for a program to auto-run at controller power-up or leave empty to disable auto-run

Program	Process
DATUMING	0
ROBOT_DEFINITIONS	
RPS_A_INITIALISATIO	
RPS_A_INITIALISE_VA	
RPS_A_MAIN_FSM	
RPS_A_SAFETY_FSM	
TPS_PLC	

Figure 23. Setting RPS_A_MAIN_FSM to autorun on process 0

1.3.5 Teach Pendant communication configuration

The Teach Pendant has a set of safety and control components to operate the system, enabling or disabling different parts of the system.

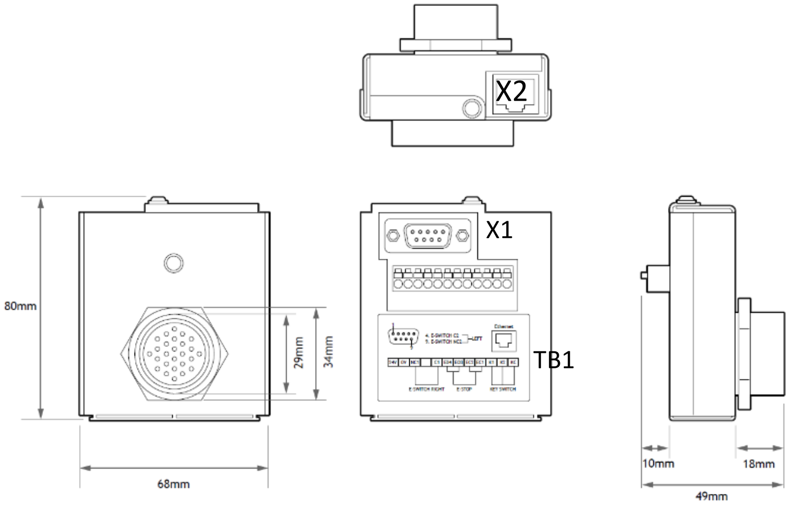
These components are: E-Stop, demand button and key switch.

These components have to be hardwired to the machine and the controller so the system will know their status.

The Teach Programming System could be used in a real or virtual Teach Pendant.

1.3.5.1 Real Teach Pendant

The real Teach Pendant comes with a junction box where all the components have their terminals. Those terminals have to be wired to the machine and to some controller inputs.



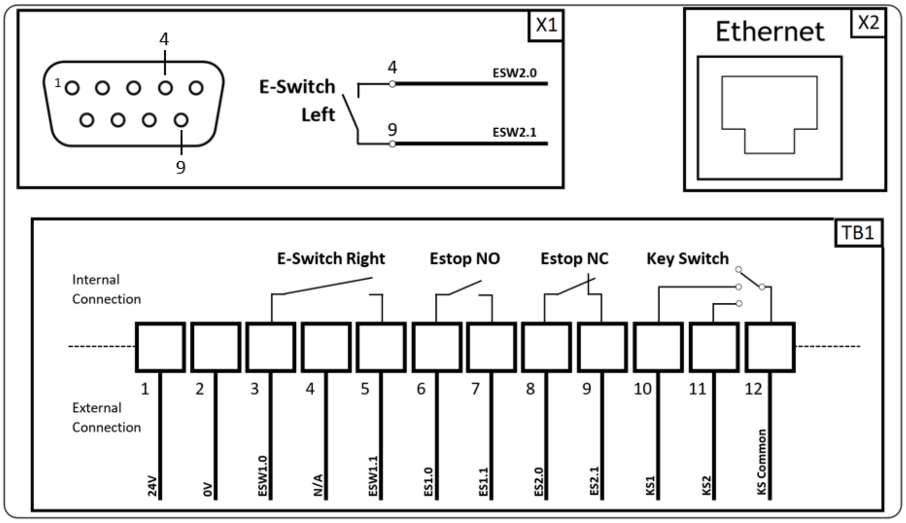


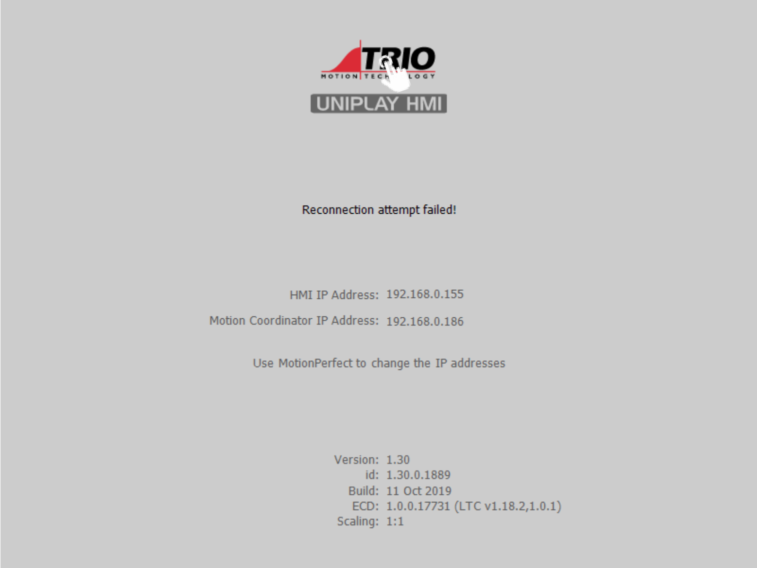
Figure 24. Teach Pendant Junction box

After the wiring, the system needs to know what controller inputs correspond with the different Teach Pendant component.

To do that, the Teach Pendant has to be powered ON and TPS has to be up and running.

To have TPS package running, the Teach Pendant needs to be connected with the controller by Ethernet cable and configured to the same IP address.

Once the Teach Pendant is powered ON and Uniplay is loaded, the Trio logo has to be touched to show a left upper corner menu:



Reconnection attempt failed!

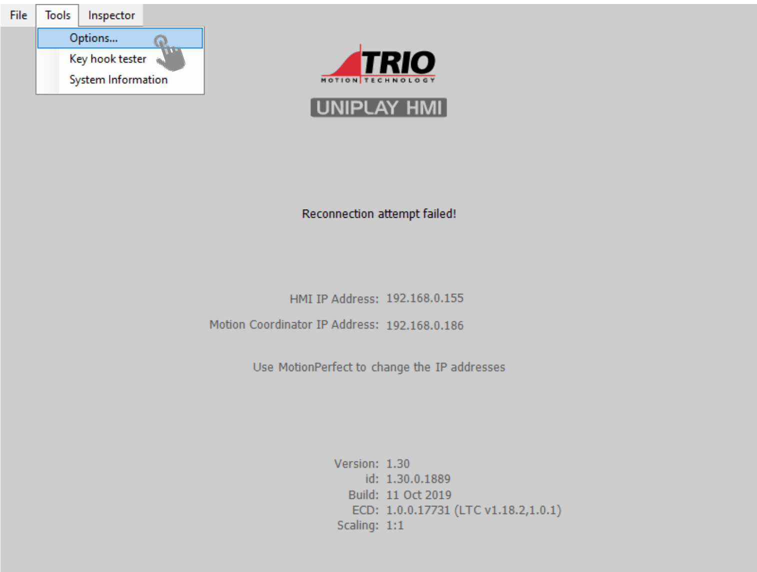
HMI IP Address: 192.168.0.155
Motion Coordinator IP Address: 192.168.0.186

Use MotionPerfect to change the IP addresses

Version: 1.30
id: 1.30.0.1889
Build: 11 Oct 2019
ECD: 1.0.0.17731 (LTC v1.18.2,1.0.1)
Scaling: 1:1

Figure 25. Uniplay

Click over Tools and then Options to display Options window.



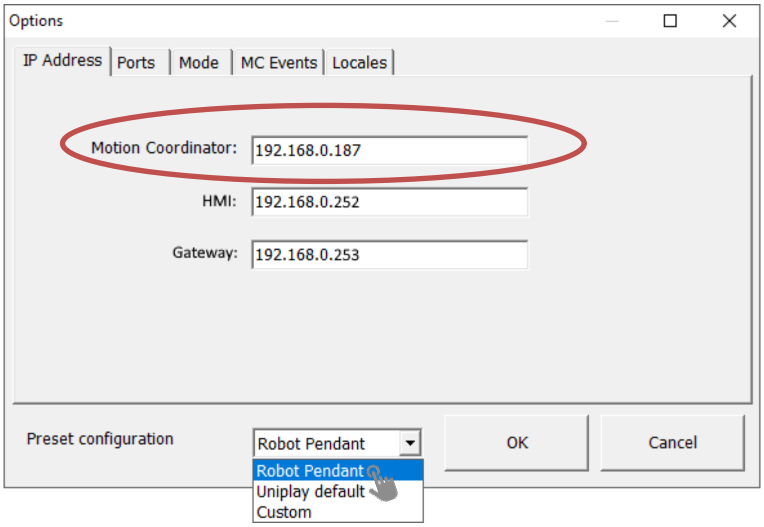
Reconnection attempt failed!

HMI IP Address: 192.168.0.155
Motion Coordinator IP Address: 192.168.0.186

Use MotionPerfect to change the IP addresses

Version: 1.30
id: 1.30.0.1889
Build: 11 Oct 2019
ECD: 1.0.0.17731 (LTC v1.18.2,1.0.1)
Scaling: 1:1

Figure 26. Uniplay options



Options

Motion Coordinator: 192.168.0.187

HMI: 192.168.0.252

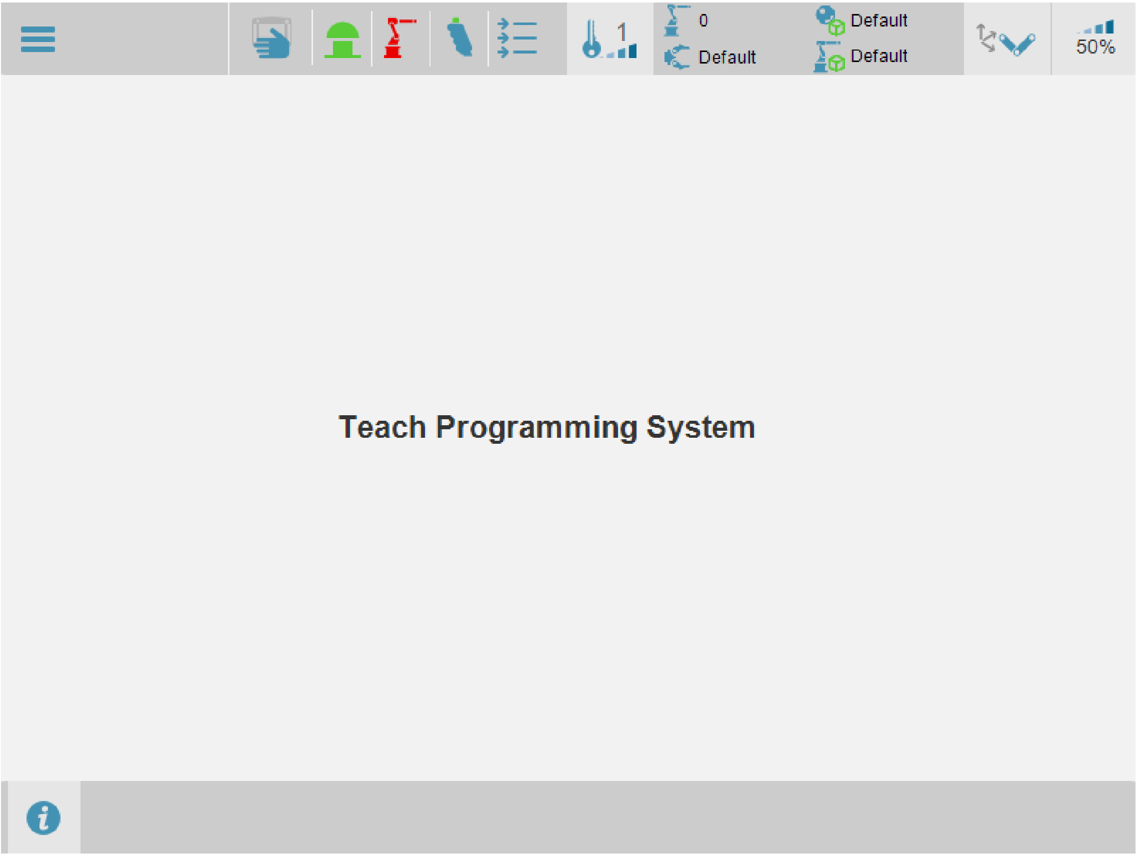
Gateway: 192.168.0.253

Preset configuration: Robot Pendant

Figure 27. Uniplay options window

Insert the IP address of the controller (Motion Coordinator) and select Robot Pendant as Preset configuration.

The OK button has to be pressed to store the changes and restart the virtual Teach Pendant. After a few seconds, if the configuration is correct, the TPS home page will be loaded.



Teach Programming System

Figure 28. TPS main page

1.3.5.2 Virtual Teach Pendant

To launch the virtual Teach Pendant, the following HMI button in Motion Perfect has to be pressed.

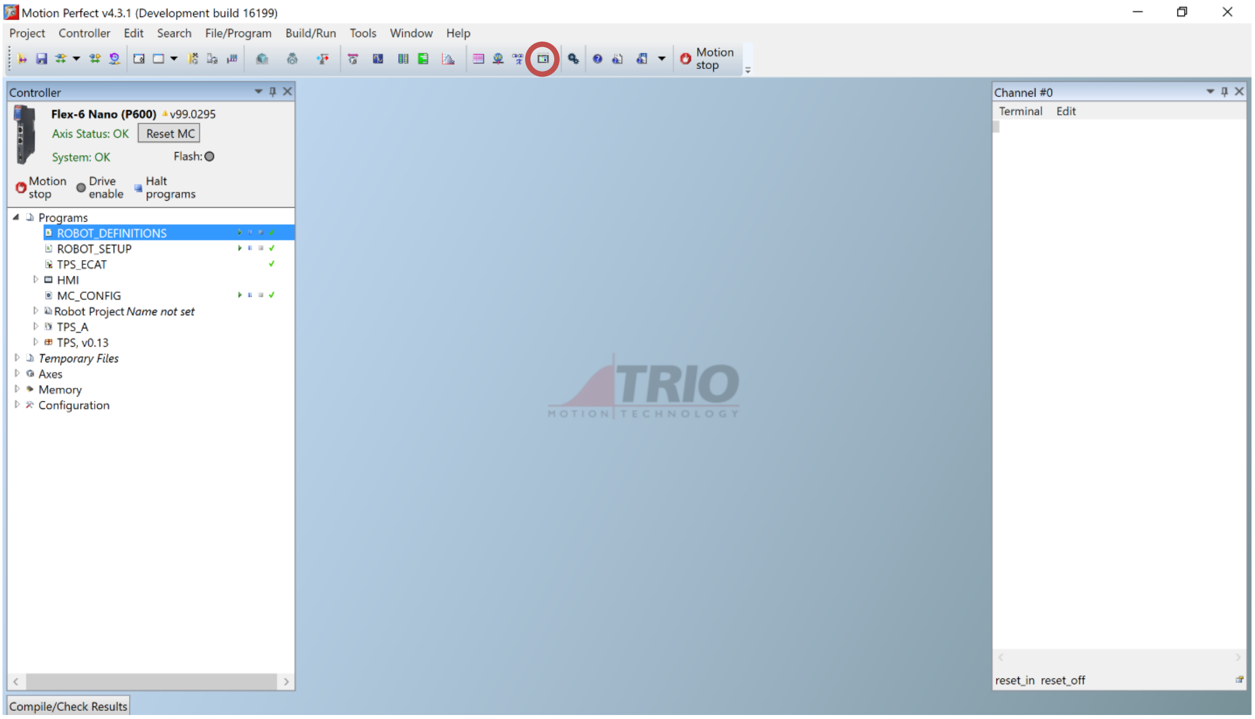
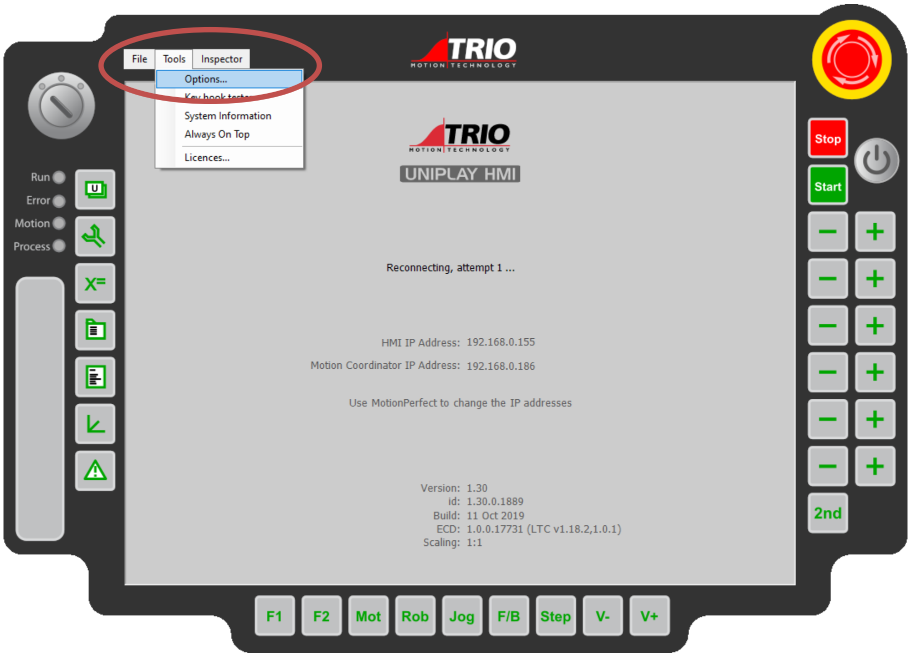


Figure 29. Virtual Teach Pendant link

The option menu could be found at the top left corner and it is always visible.



Reconnecting, attempt 1 ...

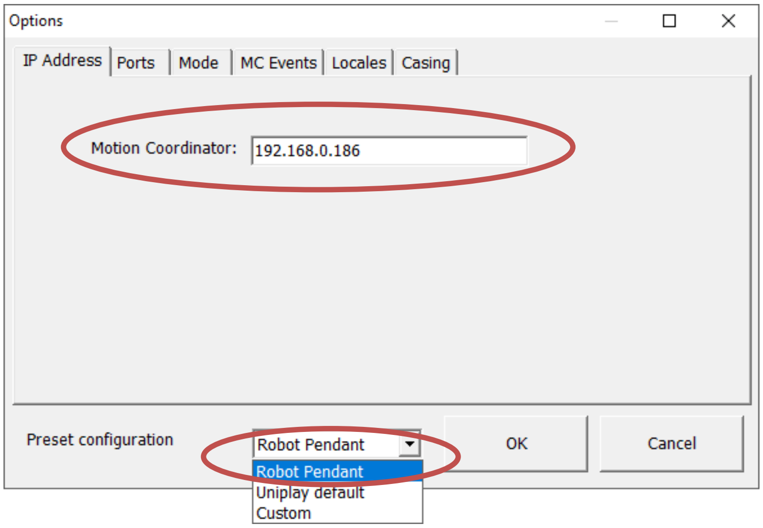
HMI IP Address: 192.168.0.155
Motion Coordinator IP Address: 192.168.0.186

Use MotionPerfect to change the IP addresses

Version: 1.30
id: 1.30.0.1889
Build: 11 Oct 2019
ECD: 1.0.0.17731 (LTC v1.18.2,1.0.1)
Scaling: 1:1

Figure 30. Uniplay options

Insert the IP address of the controller (Motion Coordinator) and select Robot Pendant as Preset configuration.



Motion Coordinator: 192.168.0.186

Preset configuration: Robot Pendant

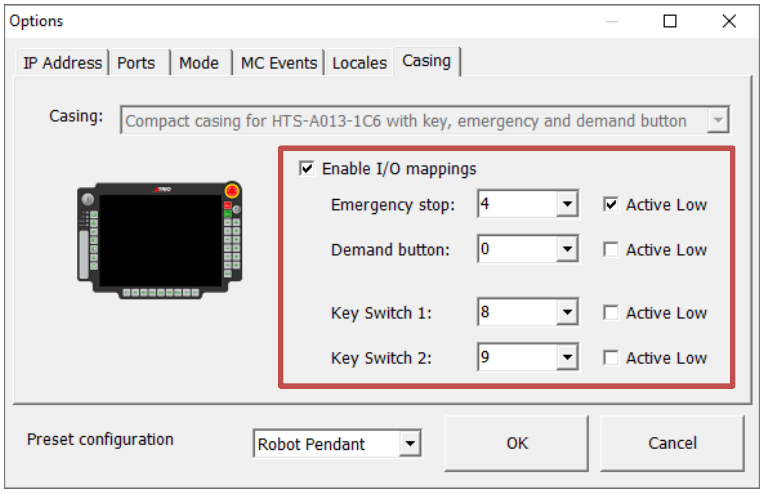
- Robot Pendant
- Uniplay default
- Custom

Figure 31. Uniplay options window

It is possible to use the controller simulator (virtual environment) instead of a real one. In that case, the Teach Pendant will have to connect with a local machine IP address, which is 127.0.0.1.

In order to have the E-Stop, demand buttons and Key Switch working, they have to be linked to some IO, that could be real or virtual and active high or low depending on the machine requirements.

To address this, there are specific settings under *Casing* tab of the virtual pendant options window.



Casing: Compact casing for HTS-A013-1C6 with key, emergency and demand button

☑ Enable I/O mappings

Emergency stop:	4	☑ Active Low
Demand button:	0	□ Active Low
Key Switch 1:	8	□ Active Low
Key Switch 2:	9	□ Active Low

Preset configuration: Robot Pendant

Figure 32. Uniplay casing options window

After configuration, OK button has to be pressed to store the changes and restart the virtual Teach Pendant. In a few seconds the system will start up and, if the configuration is correct, it will connect to the controller successfully.

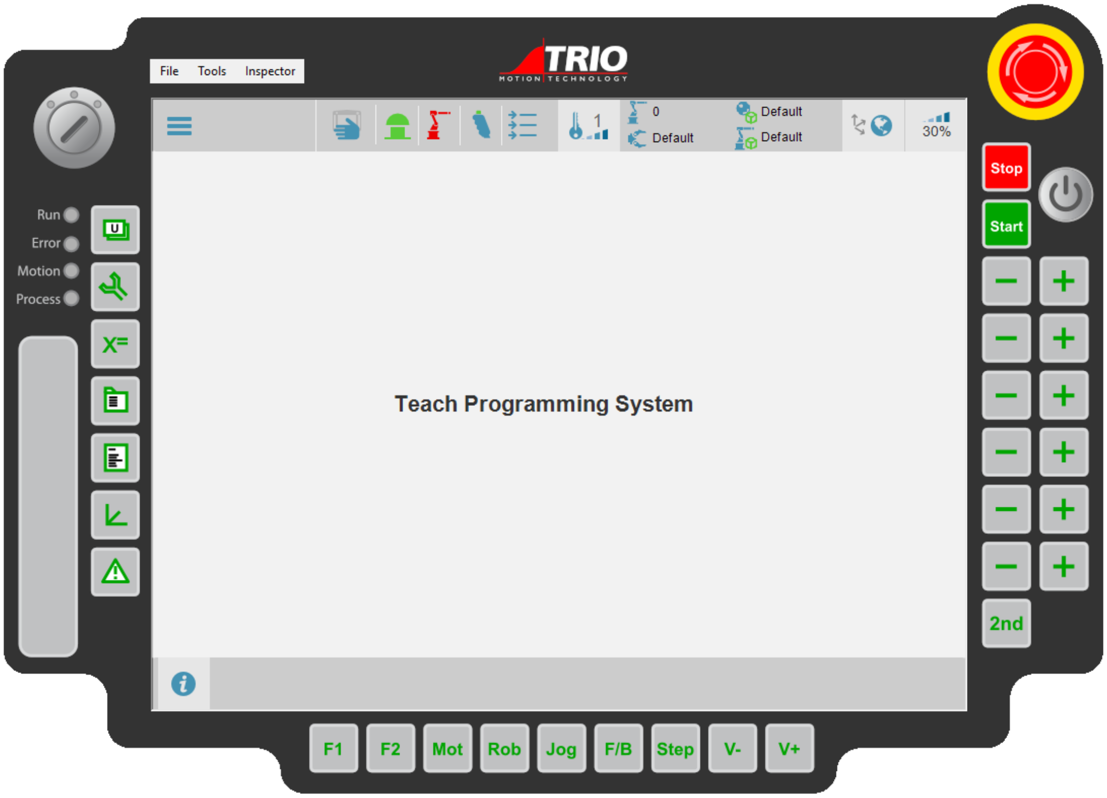
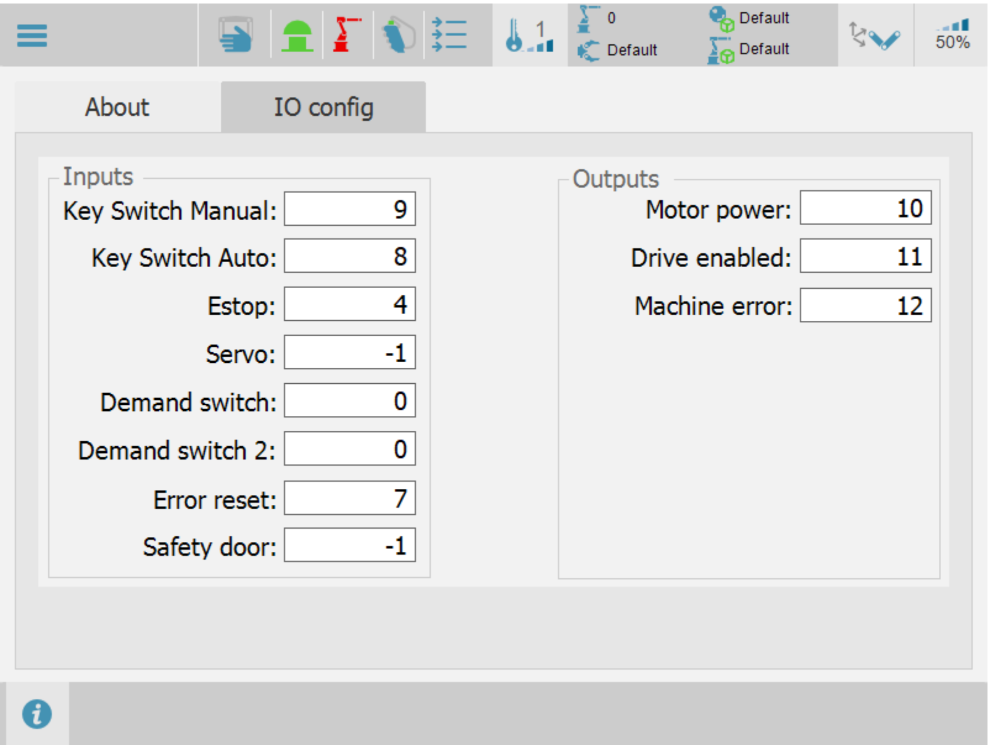


Figure 33. TPS home page

1.3.6 TPS IOs configuration

Every machine has different IO configuration. This could be done through the junction box of the real Teach Pendant or through the casing configuration of the virtual one. TPS need to know what IOs are the selected ones to behave correctly.

To setup the IOs, click on main menu ≡ located on the top left corner of the teach pendant. Next, click on *settings* >> *IO config* and enter the correct values for the machine.



Inputs

Key Switch Manual: 9
Key Switch Auto: 8
Estop: 4
Servo: -1
Demand switch: 0
Demand switch 2: 0
Error reset: 7
Safety door: -1

Outputs

Motor power: 10
Drive enabled: 11
Machine error: 12

Figure 34. TPS IO config

The inputs should be set as per physical wiring of the components.
It could be the possibility Servo, Error reset and Safety door do not exist in the real system. For those ones, a -1 value has to be set. This will tell to TPS no real input has been set for that particular component.

In the case of Error reset, if there is not real input and -1 is set, Refresh button ⟳ *has to be pressed to reset the errors in case they occur (refer to section 4.4 Warning Error window in TPS Reference guide document).*