



Machine and Robot in Harmony

RPS: Documentation summary

Trio Motion Technology

Trio Robotics Series

First Edition • 2020

Revision: 1.2

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.

Copyright (C) 2000-2018 Trio Motion Technology Ltd. All Rights Reserved

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	RPS documentation summary guide	1
1.1	Introduction.....	1
1.2	System requirements.....	1
1.3	Setting up the system	1
1.4	Setting up TPS (Teach Pendant).....	1
1.5	Run the system	1
1.6	RPS and TPS reference guides	2
2	How to work with RPS and TPS	3
2.1	Using controller simulator.....	3
2.2	Using real controller.....	3
3	RPS system logic.....	4
4	To Bear in Mind.....	6
A.	References.....	7

1 RPS documentation summary guide

1.1 Introduction

This document introduces a general guide of using the RPS documentation to setup and understand the system from the very beginning. It is necessary to pay attention at each step, going through all the documents in the correct order. Steps are linked to the specific sections of the documents. The user is encouraged to read all documents that are mentioned in the reference section.

1.2 System requirements

In order to have the system working, the following component versions has to be installed:

- TPS v1.5
- Firmware 2.0303
- Motion Perfect 5.0.1

1.3 Setting up the system

The following steps, described in *RPS: Robot Setup and Configuration*¹ document, have to be performed in order to have the system up and running.

1. Creating a project
2. Creating a robot
3. Would the Teach Programming System, TPS, be required?
 - a. If yes, then please go through step 4
 - b. If not, then the system is already set

1.4 Setting up TPS (Teach Pendant)

In order to have TPS in the system, the steps described in *TPS: Teach Pendant Setup and Configuration*² has to be performed:

1. Install and configure the latest TPS package
2. Creating Datuming program
3. Setting Virtual Teach Pendant

1.5 Run the system

Quick guide to run the already set up system and creation of small program, explained in document *RPS: Quick run*².

1. Setting up 3D Visualization
2. How to jog a robot
3. How to write a program
 - a. One of the available options is to draft the program using Motion Perfect

- b. Alternatively, the program can be drafted through the Teach Pendant either, virtual or real
- 4. Program state manipulation
- 5. Safety components

1.6 RPS and TPS reference guides

Full reference guides with lists of all commands, instructions and options available in RPS and TPS separated in two documents: *RPS: Reference guide*³ and *TPS: Reference guide*⁴.

2 How to work with RPS and TPS

In this section, different modes of operation are described. User is expected to work with any of the specified configurations only. This is a key requirement to utilise the system appropriately and with accordance to designed system architecture.

2.1 Using controller simulator

Virtual Teach Pendant can be launched in the case of existing TPS package in the system. Bear in mind that Teach Pendant has to connect with the controller through its IP address and, in the case of simulator, it will be the local 127.0.0.1.

You can then work either in virtual Pendant or Motion Perfect software. This is a test environment with no real connection to the robot. With 3D Visualisation tool being active as well, this is a good way to verify performance of the developed program.

2.2 Using real controller

If TPS package is in the system, a virtual or real Teach Pendant can be used. The 3D Visualization tool can be used as well in combination with the real robot.

There are some non-transferable encrypted files being always present in the controller; they represent the Intellectual Property that is protected and are vital for the system to operate.

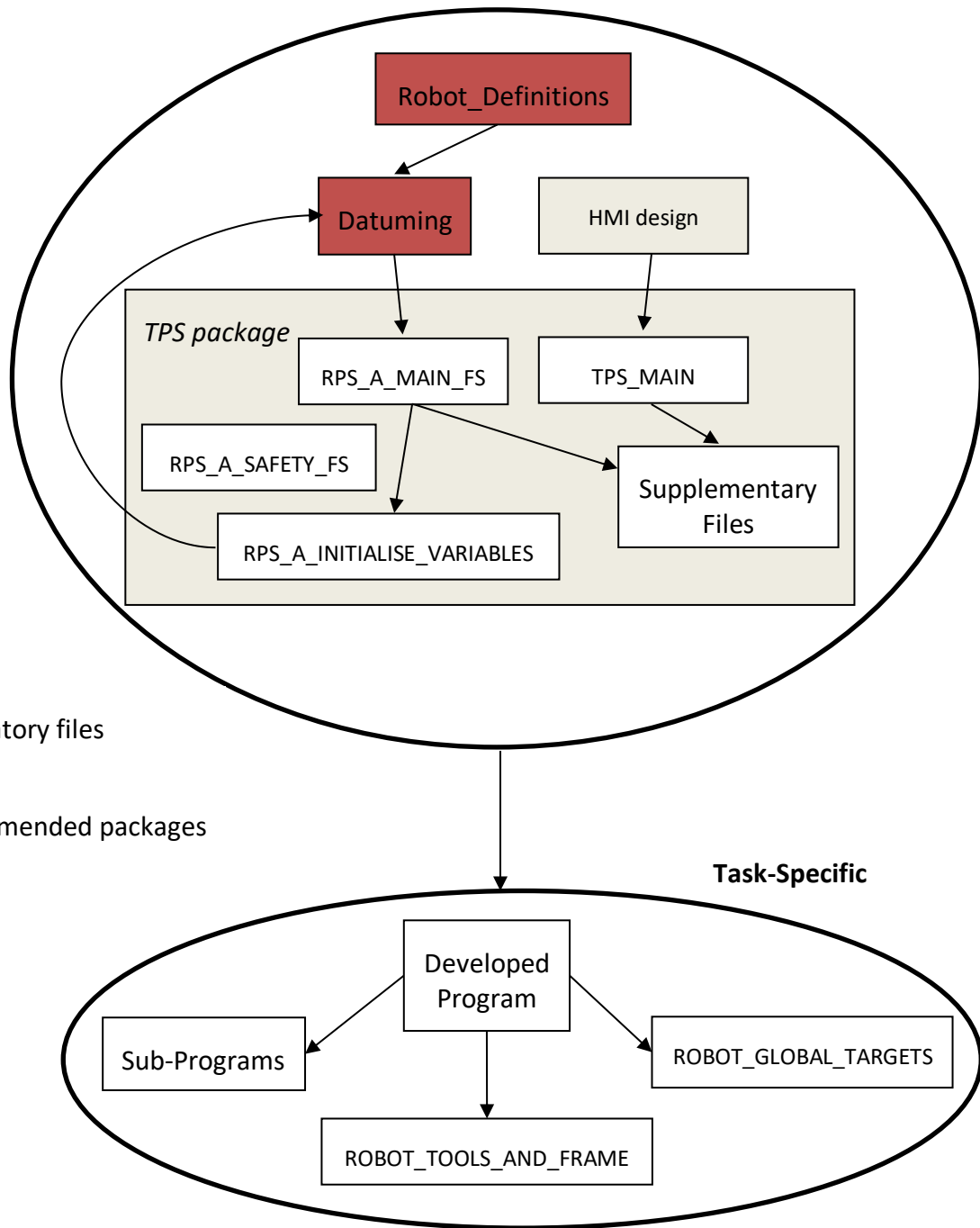
In the case of synchronising Motion Perfect with the controller with a different PC than the one that created the system, those encrypted files will not be synchronized.

However, it will be possible to synchronize the rest of the project files, create and edit new ones.

This means that the project will not be able to be transferable to another controller unless is being done with the PC that creates the original project.

When synchronized Motion Perfect with the controller from a different PC, the content of the controller can be overwritten by creating a new project. Please, pay attention during synchronization process.

3 RPS system logic



- ROBOT_DEFINITIONS represents a mandatory file where the properties of the selected robot are defined and limitations to the drives are applied.
- Datuming is a mandatory file that sets the home position (initial configuration of the robot) and triggers the whole system.
- HMI - Human Machine Interface design is an optional feature if TPS will be in the system. It is the link in between TPS package and the controller.
- TPS package stands for the Teach Programming System and is an optional module that allows programming the robot in a managed and safe environment, using real or virtual system. The following files are part of TPS package:
 - TPS_MAIN is the start-up page of the TPS system.
 - RPS_A_MAIN_FSM is the main functional state machine of the system.
 - RPS_A_SAFETY_FSM is a file that works together with RPS_A_MAIN_FSM and ensures that safety aspect of the system is addressed.
 - RPS_A_INITIALISE_VARIABLES is an input for the Datuming file which introduces the environment for further manipulations with variables.
- Supplementary Files store information that is a basement for the TPS and RPS.
- Developed Program contains the code that was produced by user.
- ROBOT_GLOBAL_TARGETS is a Global Targets Array file which contains an array of globally available target points. It is a backup for the information stored in flash.
- ROBOT_TOOLS_AND_FRAMES file stores tool and object frame definitions and is a backup for the information that is in flash memory.
- Sub-Programs represent the user-developed code that is hierarchically dependant on the main program.

4 To Bear in Mind

- Once the program is created and virtual simulation of the robotic system is used, and 3D visualisation is also switched on to assess the correctness of the program, then it is necessary to press “start simulation” in 3D visualisation windows before executing the program. This way the outcome of the program will be visualised. Otherwise, the program will just run without visualisation.
- If the system through the error “*Running of RPS programs inhibited (use DISABLE_ROBOT)*” that means TPS is in the system but is not running or, it has been running but then is stopped and a robot program has been attempted to run. Do one of the following options to fix this problem:
 - If TPS is still in the system, please make sure that auto run is set for DATUM program or TPS_MAIN_FSM and reset the controller.
 - Remove TPS package from the system and reset the controller.
- If the TPS package is embedded, it is mandatory to have it running meanwhile you execute any programs.
- Before executing any linear motion, make sure that the robot is not in a singular position (Trio Pendant will issue a warning about that). Jogging Tool can be used to set up the non-singular configuration of the robot.
- If in Trio Pendant properties servo is set as (-1), meaning that there is no physical cable connected, then please press drive enabled for the program to run.

A. References

- 1) *“RPS: Robot Setup and Configuration”*, 1st edition, revision 1.4, 2020.
- 2) *“TPS: Teach Pendant Setup and Configuration”*, 1st edition, revision 1.4, 2020.
- 3) *“RPS: Quick run”*, 1st edition, revision 1.0, 2019.
- 4) *“RPS: Reference guide”*, 1st edition, revision 1.4, 2020.
- 5) *“TPS: Reference guide”*, 1st edition, revision 1.4, 2020.