dSPACE System

First Work Steps

Release 2016-B – November 2016

Embedded Success dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

To contact dSPACE if you have problems and questions, fill out the support request form provided on the website at http://www.dspace.com/go/supportrequest.

The request form helps the support team handle your difficulties quickly and efficiently.

In urgent cases contact dSPACE via phone: +49 5251 1638-941 (General Technical Support)

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/support for software updates and patches.

Important Notice

Contents

About This Document

Contents

This document shows how to get started with the following dSPACE systems in connection with ControlDesk:

- MicroAutoBox
- MicroLabBox
- DS1103
- DS1104
- DS1005
- DS1006
- DS1007

Note

- This document does not show the hardware installation of dSPACE systems: Your specific dSPACE system must already be installed before you can perform the steps described in this document. For more information, refer to the installation and configuration documentation of your system.
- This document does not show the initial working steps if you have a SCALEXIO system. For more information, refer to 🕮 *ConfigurationDesk Getting Started With SCALEXIO* and 🕮 *SCALEXIO System Overview*.

Target audience

This document is intended for users who have no experience with dSPACE systems. Knowledge in handling computer hardware and Microsoft Windows operating systems is assumed.

<xxxx> wildcard

As the names of the demo model and the library are different for different platforms, a wildcard (<xxxx>) is used. Replace the wildcard by the number of your platform, for example, 1005 for a DS1005 board or 1401 for MicroAutoBox.

Platform	Number
MicroAutoBox	1401
MicroLabBox	1202
DS1103	1103
DS1104	1104
DS1005	1005
DS1006	1006
DS1007	1007

Files/elements used in this document

This document demonstrates some work steps using the following files/elements:

File/Element[1)]	Description	Installed by	Location[2)]
Down<xxxx>.exe utility	Command-line utility to compile and link an application for a particular processor board. The utility downloads, starts and checks the application as well.	RTLib	\Exe
smd_<xxxx>_ch.sdf	System check application to check whether your dSPACE system works correctly.	RTLib	\Demos\DS<xxxx>\Check\
smd_<xxxx>_hc.c C code	Hand-coded C-code example of a particular processor board.	RTI	\Demos\DS<xxxx>\GettingStarted\HandCode\
smd_<xxxx>_hc.zip	Demo project/experiment to open with ControlDesk		
smd_<xxxx>_sl.slx	MATLAB/Simulink-based model for a particular processor board.	RTI	\Demos\DS<xxxx>\GettingStarted\Simulink
smd_<xxxx>_sl.zip	Demo project/experiment to open with ControlDesk		
DS<xxxx> Demo	Demo project/experiment to open with ControlDesk	ControlDesk	The location depends on your operating system. For users of Windows 7, it is in C:\Users\<User>\Documents\dSPACE\ControlDeskNG\x.y.

1) <xxxx> stands for the platform number
2) Folder relative to the folder of the RCP & HIL installation.

Where to go from here

Information in this section

Conventions Used in the Documentation

Admonitions

The following admonitions may be used in this document.

Admonition	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that may cause property damage if you do not avoid it by following the instructions given.
Note	Indicates important information that should be kept in mind, for example, to avoid malfunctions.
Tip	Indicates tips containing useful information to make your work easier.

Naming conventions

The following abbreviations and formats are used in this document:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

📖 Precedes the document title in a link that refers to another document.

 Indicates that a link refers to another document, which is available in dSPACE HelpDesk.

 Indicates that a link refers to a glossary entry.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\My Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing Online Help and PDF Files

Objective

After you install your dSPACE software, the documentation for the installed products is available as online help and Adobe® PDF files.

Online help

You can access the online help, dSPACE HelpDesk, as follows:

Windows Start menu Select Start – (All) Programs – <ProductName> – dSPACE HelpDesk (<ProductName>) to open dSPACE HelpDesk with the start page of the selected product displayed. You can also navigate and search in the user documentation of any other installed software product and its supported hardware.

Context-sensitive Press the **F1** key or click the Help button in the dSPACE software to get help on the currently active context.

Note

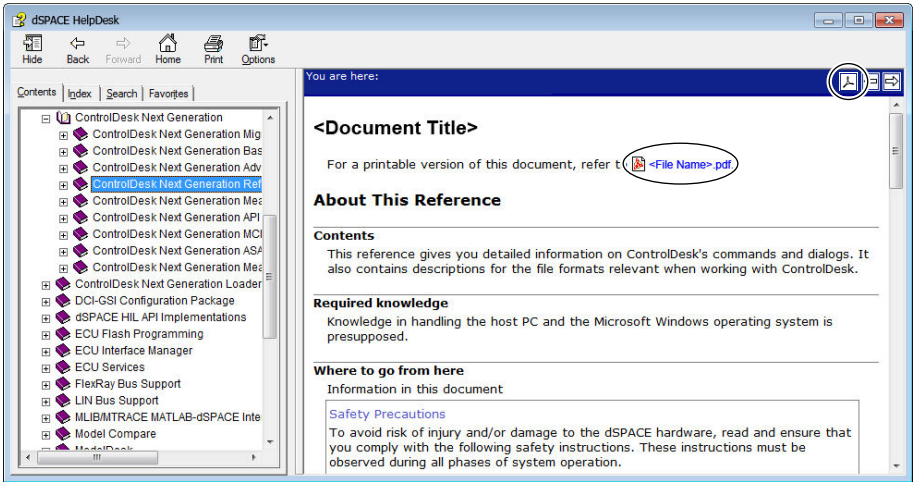
In some software products, context-sensitive help is not available.

Help menu in the dSPACE software On the menu bar, select Help – Contents or Help – Search (not available in all software products) to open dSPACE HelpDesk. It opens at the start page of the currently active product. You can also navigate and search in the user documentation of any other installed software product and its supported hardware.

PDF files

You can access the PDF files as follows:

dSPACE HelpDesk Click the PDF link at the beginning of a document or on a topic pane's header:



Related Documents

Objective

Below is a list of documents that you are recommended to read when working with a dSPACE board:

Information in other documents

Software installation and management *Software Installation and Management Guide* Provides detailed instructions on installing and handling the dSPACE software. It also shows you how to manage dSPACE licenses.
Hardware installation and configuration *DS1005 Hardware Installation and Configuration Guide*

🕮 *DS1006 Hardware Installation and Configuration Guide* 🕮 *DS1007 Hardware Installation and Configuration Guide* 🕮 *PHS Bus System Hardware Reference* Provides hardware-related reference information on your PHS-bus-based system and all the relevant information on signal connections to external devices. 🕮 *dSPACE Simulator Mid-Size Based on DS2211 Hardware Installation and Configuration* 🕮 *DS1103 Hardware Installation and Configuration* 🕮 *DS1104 Hardware Installation and Configuration* 🕮 *MicroAutoBox II Hardware Installation and Configuration* 🕮 *MicroLabBox Hardware Installation and Configuration*
RTI and RTI-MP 🕮 *RTI and RTI-MP Implementation Guide* Gives detailed information and instructions on using Real-Time Interface (RTI and RTI-MP) to implement your real-time models.
ControlDesk 🕮 *ControlDesk Introduction and Overview* Introduces ControlDesk.

Quick Start for Working with a dSPACE System

Objective

After checking the preconditions, you can start work on typical use-cases.

Where to go from here

Information in this section

Information in other sections

Required Software and Hardware

The dSPACE system must be installed before you can perform the steps described in this document.

Required software

To work with this document, the following software is required:

- The MATLAB software must be installed with at least the following components:
  - Simulink
  - Simulink® Coder™
  - Necessary toolboxes, refer to

    http://www.dspace.com/goto?toolboxes

  Note

  The MATLAB version must match the available dSPACE Release. For compatibility information, refer to http://www.dspace.com/goto?compatibility.
- The dSPACE software must be installed.

  The license must comprise at least:
  - RTI (Real-Time Interface), refer to

    http://www.dspace.com/go/qsrti
  - A C code cross compiler, refer to

    http://www.dspace.com/go/qscompiler
  - ControlDesk, refer to

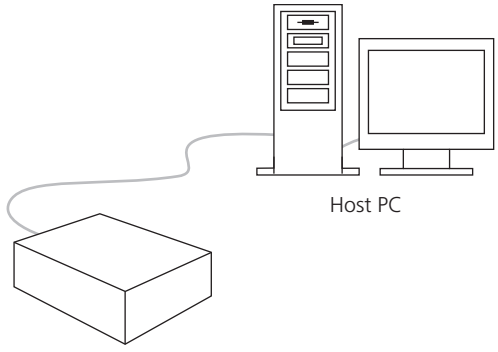
    http://www.dspace.com/go/controldesk

  For information on how to install dSPACE software, refer to the 🕮 *Software Installation and Management Guide*.

Required hardware

To work with this document, dSPACE hardware must be installed:

- dSPACE hardware is typically installed in an expansion box outside the PC. The expansion box must be connected to the host PC either via bus interface (using dSPACE's link boards) or via Ethernet.



- A DS1104 Controller Board is installed directly in the PC (PCI slot).
- MicroAutoBox and MicroLabBox are connected to the host PC via Ethernet.

dSPACE boards and their installations The table below lists the different dSPACE boards and their installations:

dSPACE Board	Installation/Connection
DS1104	The DS1104 is installed in the host PC.
DS1103	The DS1103 can be installed in the host PC or an expansion box[1)].
MicroAutoBox	MicroAutoBox is connected to the host PC via Ethernet.
MicroLabBox	MicroLabBox is connected to the host PC via Ethernet.
DS1005	The DS1005 can be installed in an expansion box[1)].
DS1006	The DS1006 is installed in an expansion box[1)].
DS1007	The DS1007 is installed in an expansion box connected to the host PC via Ethernet.

[1)] An expansion box can be connected to the host PC via bus interface or Ethernet.

For information on how to install dSPACE real-time hardware and set up the connection to the host PC, refer to the *Hardware Installation and Configuration* documents.

How to Generate a Real-Time Application

Objective	First you generate a real-time application for your dSPACE system.
Workflow	To generate a real-time application, you first have to create a Simulink model, and then add the I/O interface to the model using dSPACE RTI blocks. After C code generation, the code is compiled and linked to a real-time application. **Creating a Simulink model** You have to create a Simulink model using MATLAB and Simulink. Instead of programming C code manually, you implement the algorithm graphically using Simulink blocks. The models are saved as SLX files. **Specifying RTI I/O interfaces** To connect the simulation model to the physical world, you need to introduce I/O interfaces into the model. These allow you to replace parts of your simulated model with real hardware. dSPACE's RTI (Real-Time Interface) blocks provide I/O interfaces for accessing dSPACE hardware. **Generating C code** You can build the model created with Simulink and RTI blocks using the Simulink® Coder™. The Simulink® Coder™ generates C code from the model automatically. **Compiling and linking the real-time application** The cross compiler environment compiles the generated C code and links the object files and libraries into an executable application for the real-time processor.
Precondition	Before you can start the build and download procedure, you have to ensure that your dSPACE hardware is registered correctly in ControlDesk. Refer to *How to Register a dSPACE Board* on page 31.

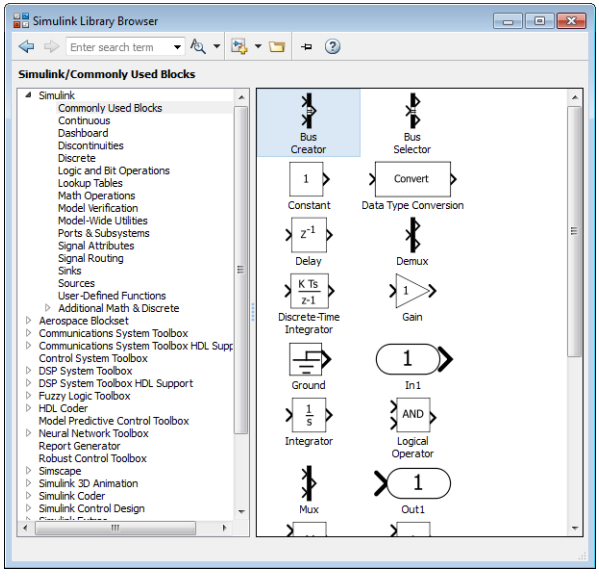
Method

To generate a real-time application

1 Start MATLAB.

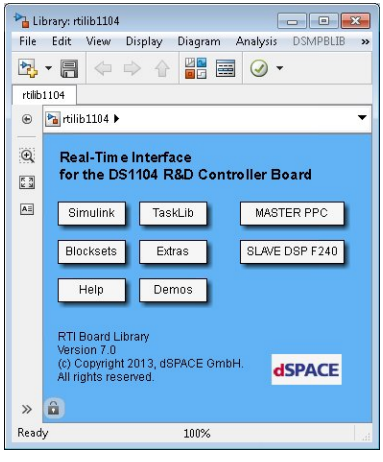
2 Type **`simulink`** in the MATLAB Command Window.

The Simulink Library Browser opens.



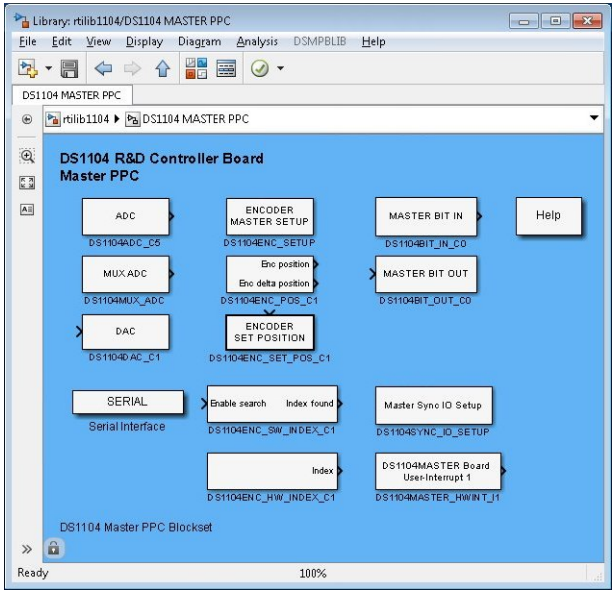
3 In the MATLAB Command Window, enter **`rti<Name of the Library>`**, for example, **`rti1104`**.

The RTI library opens.



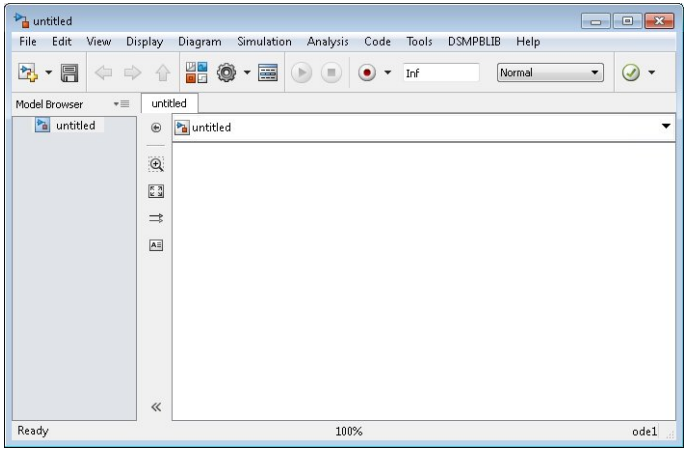
4 In the library, double-click Master PPC.

The DS1104 Master PPC library opens.

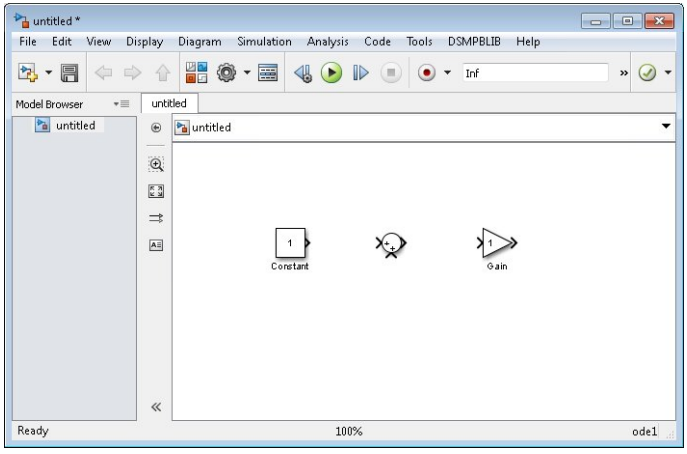


5 From the menu bar of the RTI library, select File – New – Model.

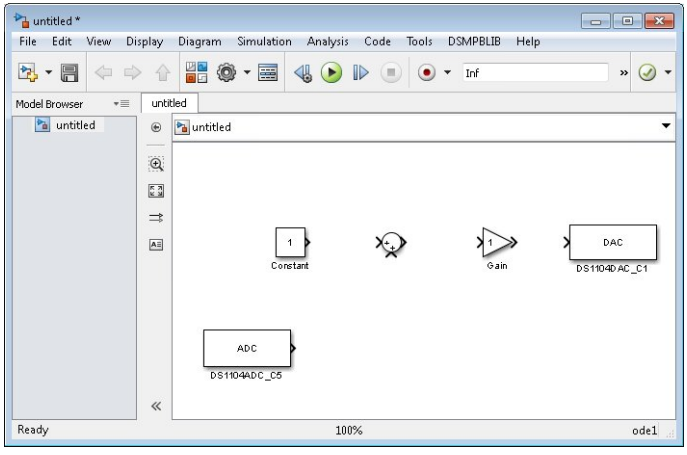
A new model opens.



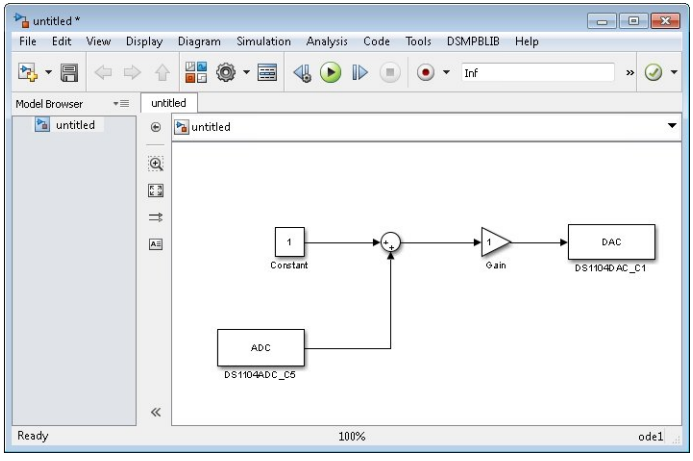
6 Add Simulink blocks from the Simulink Library Browser to the model via drag & drop.



7 Add RTI blocks from the DS1104 Master PPC library to the model via drag & drop.



8 Add the connections between the blocks. A simple model could look like this:



9 Save the model as an SLX file.

10 From the menu bar of the model, select Code – C/C++ Code– Build Model, or press **`Ctrl + B`**.

The Simulink® Coder™ generates C code. The RTI build process compiles the generated C code, links the object files and libraries into an executable application and downloads the application to the real-time hardware directly after the compilation (build). The build status is displayed in the MATLAB Command Window.

Result

You have generated a real-time application consisting of the following files:

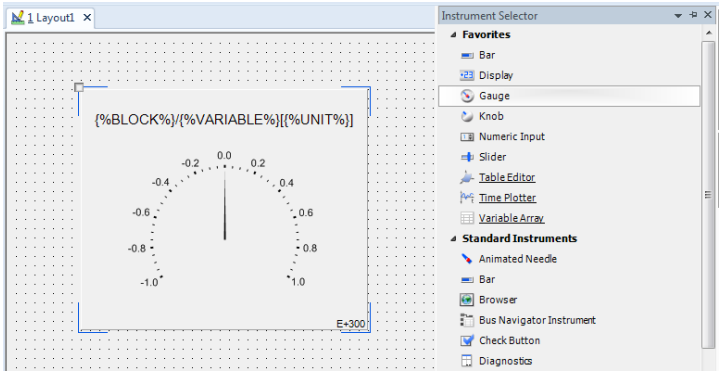
File	Description
MAP	Map file with address information of variables
PPC, x86 or RTA	Real-time application to be downloaded to the dSPACE real-time hardware
SDF	System description file to be used by ControlDesk. It contains references to the PPC/x86/RTA, MAP, and TRC file.
TRC	Variable description file
TRZ	If real-time testing is enabled for the generated real-time application, the build process generates a TRZ file containing the MAP file and the TRC file.

Next step

You can now experiment with the real-time application with ControlDesk. Refer to *How to Experiment with a Real-Time Application in ControlDesk* on page 19.

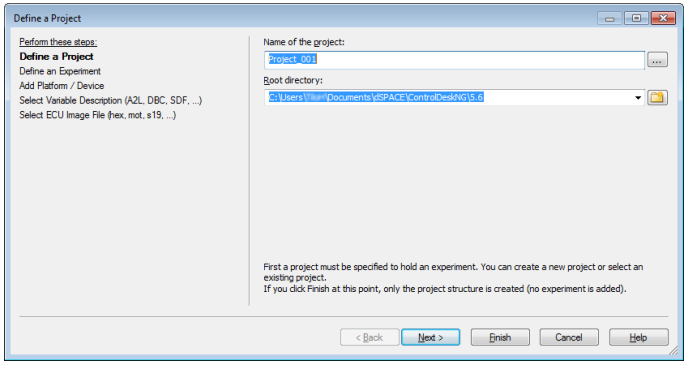
How to Experiment with a Real-Time Application in ControlDesk

Objective	You can download a real-time application to the hardware and perform experiments with ControlDesk.
Experimenting with a real-time application	You can download the generated real-time application to the hardware with ControlDesk. Using the information from the SDF file, ControlDesk can read and write the variables of the real-time application. You can experiment with real-time applications in ControlDesk, which provides numerous instruments for you to measure signals and change parameter values.



Method	**To experiment with a real-time application in ControlDesk** **1** Start ControlDesk. **2** On the File ribbon, click New – Project + Experiment. **3** In the Define a Project dialog, enter a name for the new project. Click Next >.

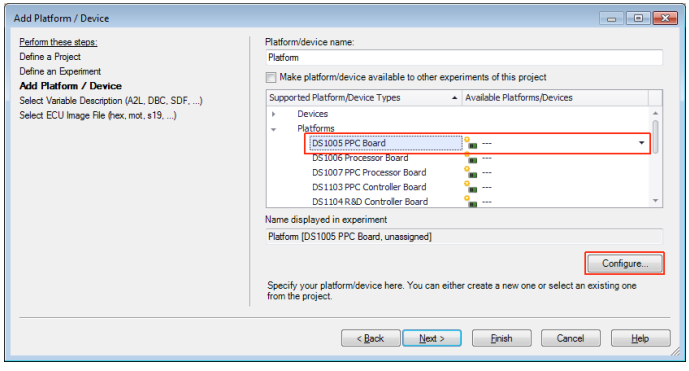
The Define an Experiment dialog opens.



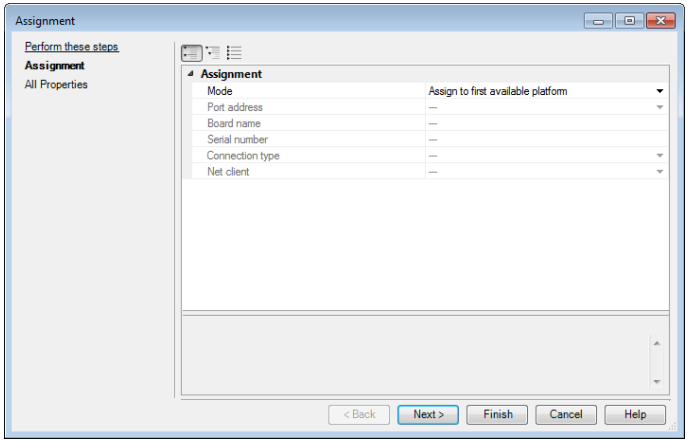
4 In the Define an Experiment dialog, enter a name for the new experiment and click Next >.

The Add Platform / Device dialog opens.

5 In the Add Platform / Device dialog, select your dSPACE board, for example the DS1005 PPC Board and click Configure.



The Assignment dialog opens.



6 In the Assignment dialog, select the connection type.

If you select the NET connection type, you must specify the Net client.

The net client is identified by its IP address or alias specified in the host table (refer to *Setting Up an Ethernet Connection Between PC and Expansion Box* in your *Hardware Installation and Configuration* document).

7 Select the Port address and click Finish to close the Assignment dialog.

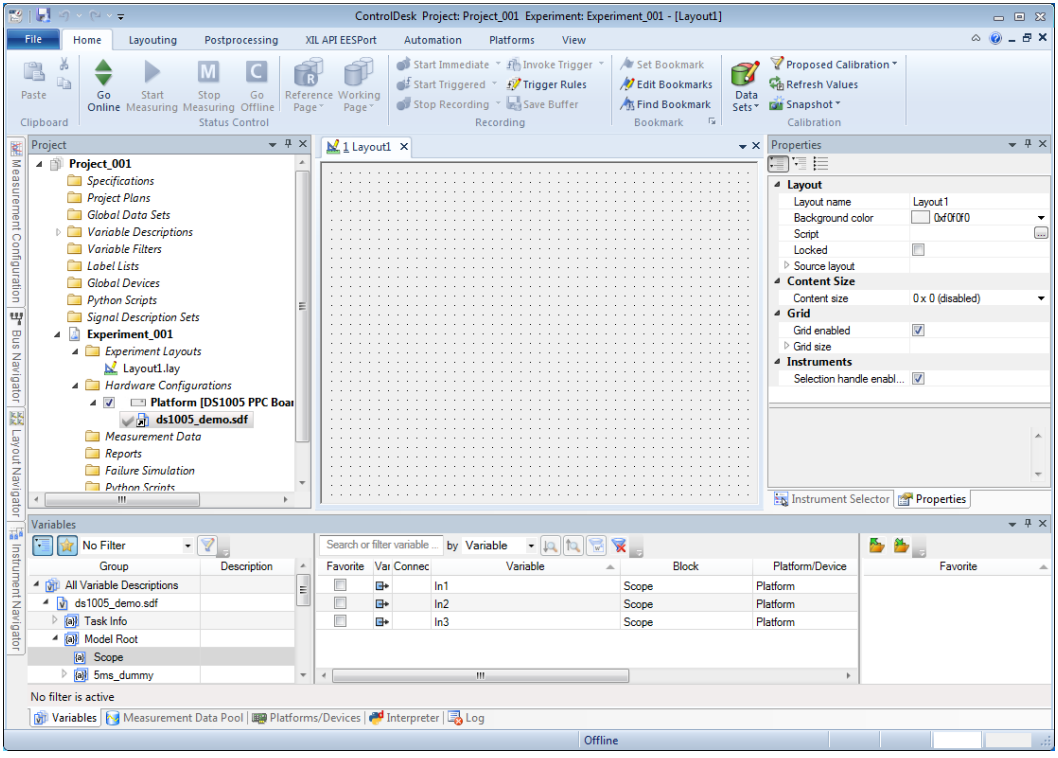
8 In the Add Platform / Device dialog, click Next >.

The select Select Variable Description dialog opens.

9 In the Select Variable Description dialog, click Import from file to navigate to the installation folder of your ControlDesk installation, select the `<platform>_demo.sdf` variable description file for your real-time hardware from the `\Demos\RTApplications\<platform>` folder and click Open. For example, select `../Demos/RTApplications/DS1005/ds1005_demo.sdf` for a DS1005.

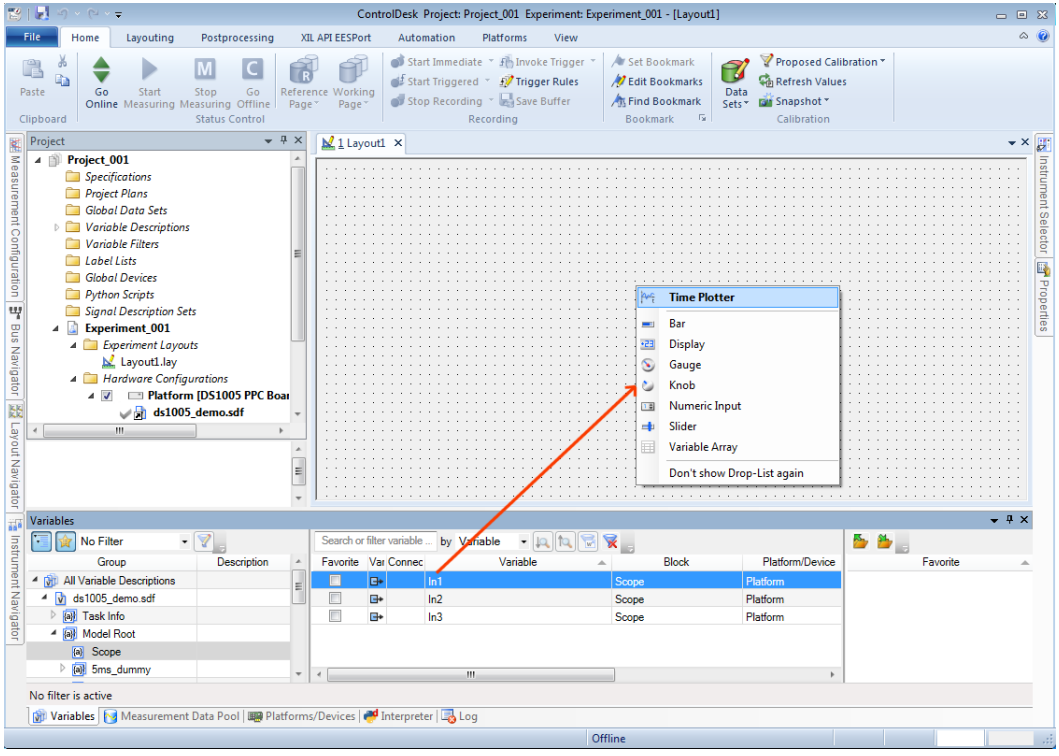
10 Click Finish.

You have created a new project/experiment.

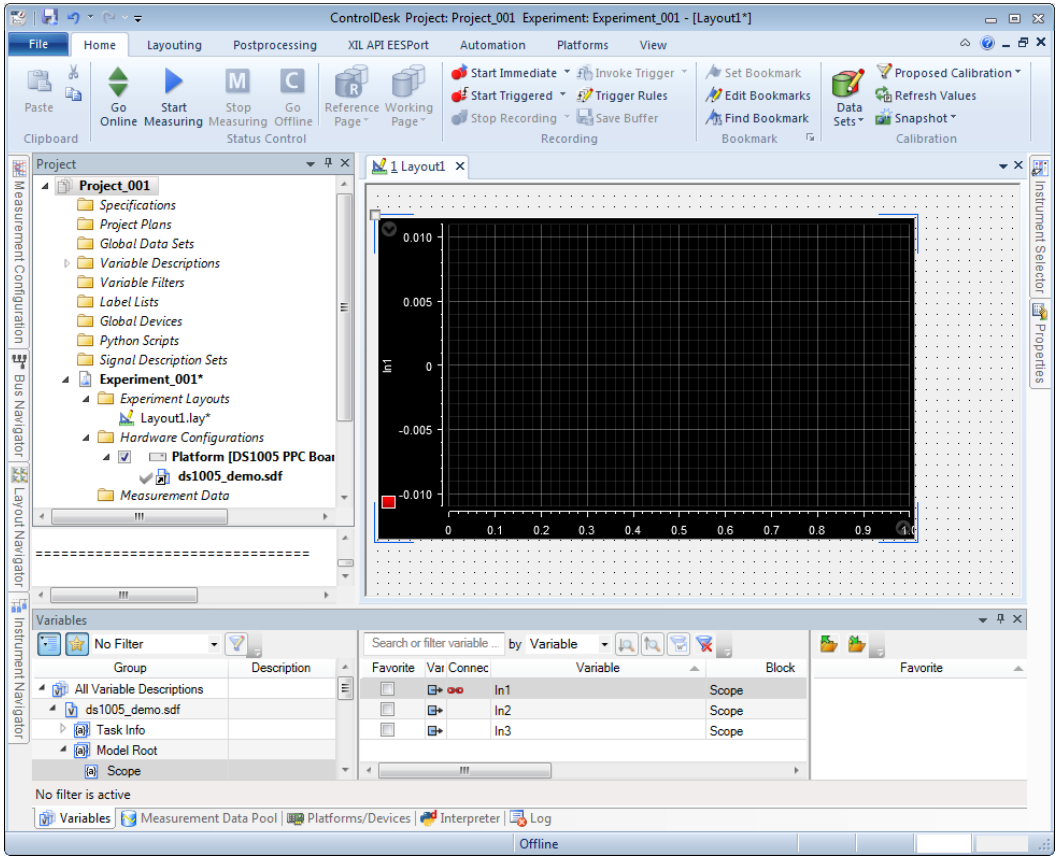


11 In the Variable Browser, select a measurement variable and drag it to the new layout.

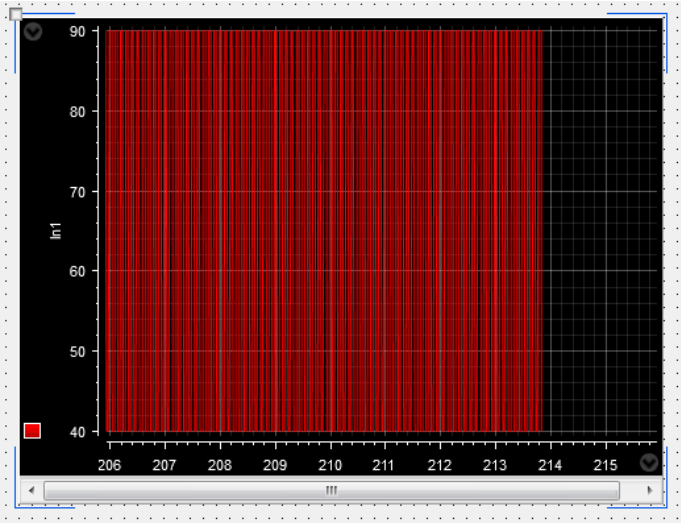
12 In the Instrument Type list, click Time Plotter.



The Time Plotter opens.



13 On the Home ribbon, click Status Control – Start Measuring, or press **F5**.



The plotter measures the values of the connected variable.

Result

You experimented with the real-time application in ControlDesk and finished the quick start using dSPACE hardware. If you need a detailed description of how to experiment with ControlDesk, refer to *Experimenting with ControlDesk* on page 69.

Detailed Description for Working with a dSPACE System

Objective

If the quick start instructions do not match your requirements, this document also provides a detailed description for working with your dSPACE system.

Where to go from here

Information in this section

Accessing the Hardware with ControlDesk

Objective

You can manage dSPACE real-time hardware with ControlDesk's Platform Manager.

Where to go from here

Information in this section

Basic Steps When Accessing the Hardware

Objective

An overview of the workflow and an introduction to the graphical user interface of ControlDesk help you to access the hardware.

Where to go from here

Information in this section

Workflow for Accessing the Hardware

Objective

To access dSPACE hardware, you can use ControlDesk, which lets you experiment with dSPACE systems. For example, you can use it to register dSPACE boards, download applications and manage experiments.

The first time you access dSPACE real-time hardware, you have to perform the following workflow:

1. Starting ControlDesk

   ControlDesk is installed on the host PC during software installation. It provides the Platform/Device Manager, which is the central component for configuring dSPACE real-time hardware. Refer to *How to Start ControlDesk* on page 30.

2. Registering dSPACE boards

   If a dSPACE board supports plug & play, it is registered automatically, otherwise you have to register it manually. You must also specify the connection type, which defines how the dSPACE board is connected to the host PC. Refer to *How to Register a dSPACE Board* on page 31.

3. Checking the configuration of dSPACE boards

   When the dSPACE board is registered you can check the board configuration. Refer to *Checking the Configuration of the dSPACE Boards* on page 37.

4. Updating the firmware

   Your board usually contains the latest firmware. If you install a new dSPACE Release it may contain newer firmware. In this case the firmware should be updated. Refer to *Updating the Firmware* on page 39.

How to Start ControlDesk

Objective

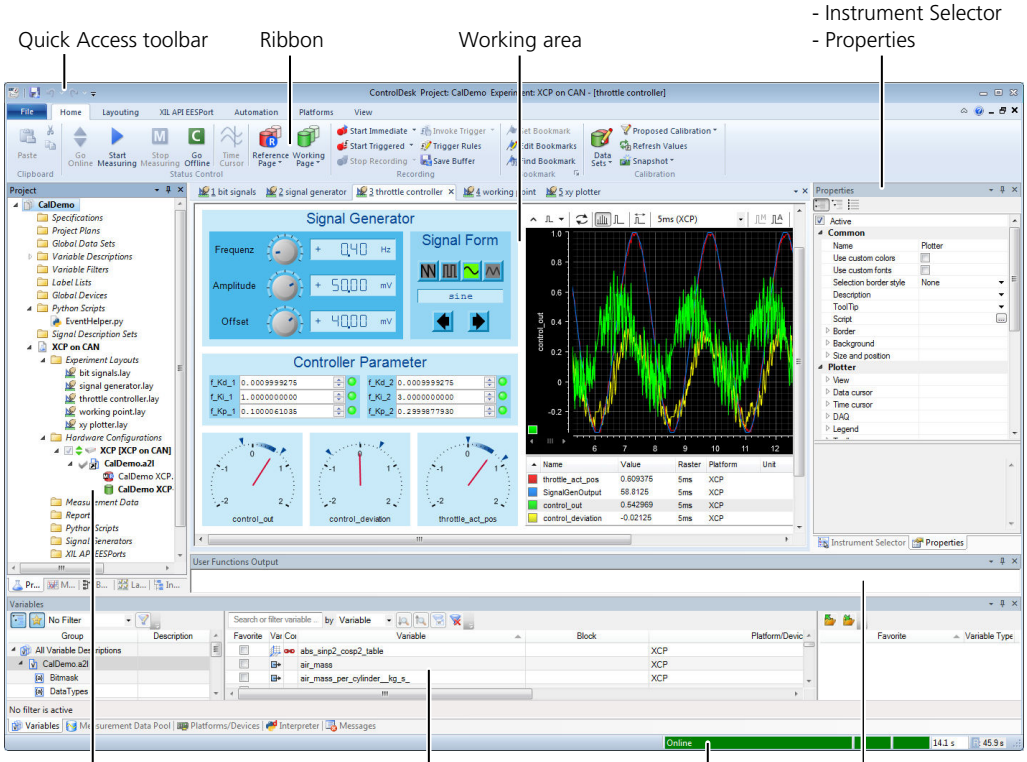
After installation of the dSPACE software, ControlDesk is available in the Windows Start menu and as an icon on the desktop.

Tip

For more information on ControlDesk and its features, refer to the 🕮 *ControlDesk Introduction and Overview*.

User interface

The following illustration shows the user interface of ControlDesk.

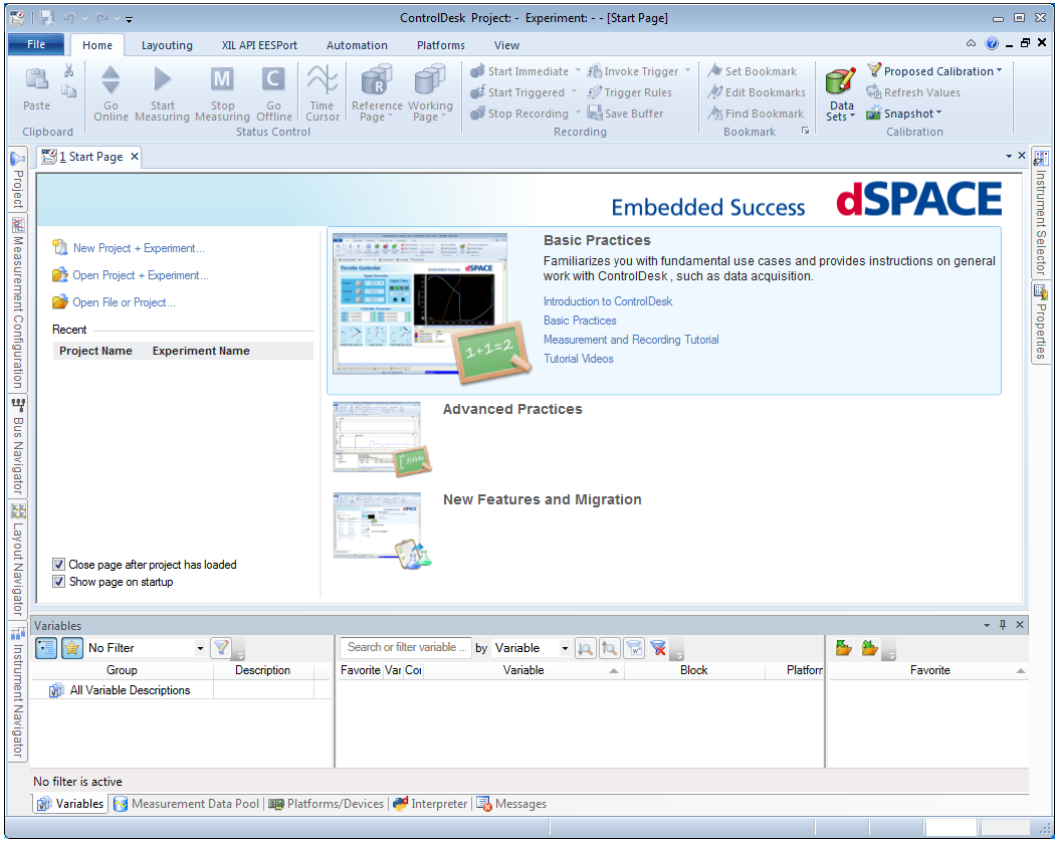


Method

To start ControlDesk

1 In the Windows Start menu, select Programs – dSPACE ControlDesk 6.0.

ControlDesk starts, see the following illustration.



Result

ControlDesk is ready to register your dSPACE board.

How to Register a dSPACE Board

Objective

After installing a single dSPACE processor or controller board, a multiprocessor system, MicroAutoBox, or MicroLabBox you have to register it to make it known to ControlDesk.

Tip

You do not need to register boards that support the plug & play feature. The Platform/Device Manager automatically registers them. This applies to:

- DS1104

Different registration process of DS1005 and DS1007

The working concepts of DS1005 and DS1007 in ControlDesk differ. One particular difference is in the registration process. For details, refer to *Working Concept Differences Between DS1005 and DS1007* (📖 *ControlDesk Platform Management*).

Preconditions

- To register a platform to access dSPACE real-time hardware, the hardware must be connected to the host PC.
- MicroAutoBox II: Before you register MicroAutoBox II, you should configure it using the `DS1401ConfigGUI.exe` utility, for example, to change MicroAutoBox II's default IP address. The utility is located in `<RCP_HIL_InstallationPath>\Exe`. For configuration details, refer to *Connecting MicroAutoBox to the Host PC via Ethernet* (📖 *MicroAutoBox II Hardware Installation and Configuration*).

  Tip

  You can add the `DS1401ConfigGUI.exe` tool as a user function. Refer to *How to Add External Programs or Scripts as User Functions to ControlDesk* (📖 *ControlDesk Customization*).

Method

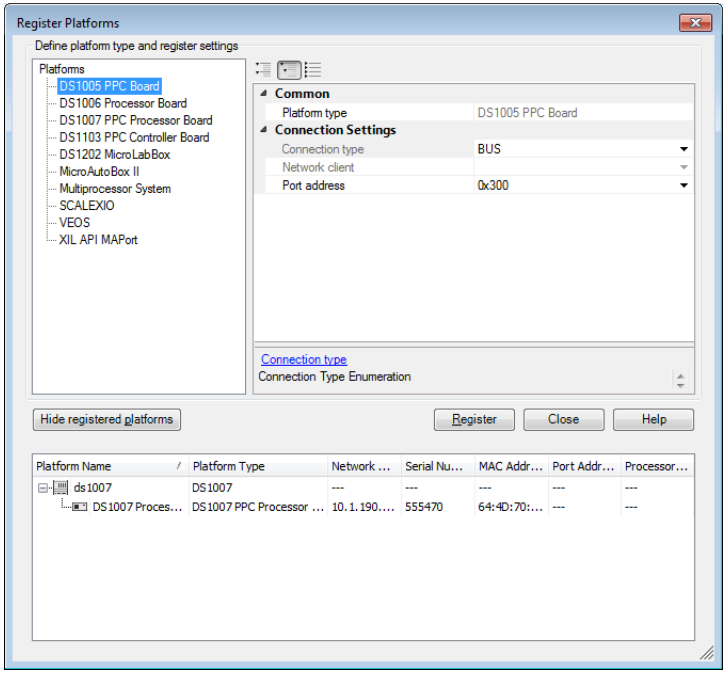
To register a dSPACE board

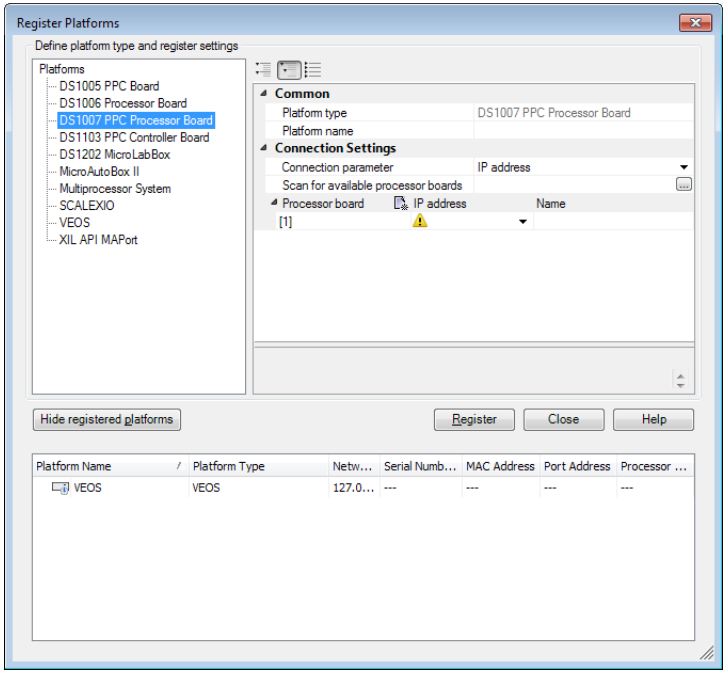
1. On the Platforms ribbon, click Platform Management - Register Platforms.

   The Register Platforms dialog opens.

2. From the Platforms list, select the type of the platform you want to register.

   The registration settings vary according to the selected platform type. The following illustrations show examples for the DS1005 and the DS1007 platform types.


Register Platforms
Define platform type and register settings
Platforms
DS1005 PPC Board
DS1006 Processor Board
DS1007 PPC Processor Board
DS1103 PPC Controller Board
DS1202 MicroLabBox
MicroAutoBox II
Multiprocessor System
SCALEXIO
VEOS
XIL API MAPort
Common
Platform type
DS1005 PPC Board
Connection Settings
Connection type
BUS
Network client
Port address
0x300
Connection type
Connection Type Enumeration
Hide registered platforms
Register
Close
Help
Platform Name
Platform Type
Network ...
Serial Nu...
MAC Addr...
Port Addr...
Processor...
ds1007
DS1007
DS1007 Proces...
DS1007 PPC Processor ...
10.1.190....
555470
64:4D:70:...


Register Platforms
Define platform type and register settings
Platforms
DS1005 PPC Board
DS1006 Processor Board
DS1007 PPC Processor Board
DS1103 PPC Controller Board
DS1202 MicroLabBox
MicroAutoBox II
Multiprocessor System
SCALEXIO
VEOS
XIL API MAPort
Common
Platform type
DS1007 PPC Processor Board
Platform name
Connection Settings
Connection parameter
IP address
Scan for available processor boards
Processor board
IP address
Name
[1]
Hide registered platforms
Register
Close
Help
Platform Name
Platform Type
Netw...
Serial Numb...
MAC Address
Port Address
Processor ...
VEOS
VEOS
127.0...

3 Specify the registration settings for the dSPACE hardware you want to register. The number and kinds of properties depend on the selected platform type.

Property	Description / Refer to
Common Properties	
Multiprocessor type	*Common Properties* (🕮 *ControlDesk Platform Management*)
Platform name	Lets you specify a unique name for the selected platform. After registration, the name is displayed in ControlDesk's Platform/Device Manager. The valid characters are 'a ... z', 'A ... Z', '0 ... 9', '_', '-' and ' '. The name must not start or end with an underline, hyphen or blank. If you do not specify a platform name, ControlDesk displays a default name in the Platform/Device Manager.
Platform type	*Common Properties* (🕮 *ControlDesk Platform Management*)
Topology check	*General Settings Properties* (🕮 *ControlDesk Platform Management*)
Connection Settings Properties	
Alias name	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
Board name	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
Connection parameter	Lets you select one of the following connection parameters for registration: ■ Alias name ■ Board name ■ IP address ■ MAC address The selected parameter is used to register the member processing units (SCALEXIO), processor boards (DS1007), or platforms (DS1202 MicroLabBox).
Connection type	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
IP address	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
MAC address	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
Network client	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
Platform	Lets you specify the platform belonging to the DS1202 MicroLabBox platform. The platform has an edit field to specify its connection parameter value.
Port address	*Assignment Properties* (🕮 *ControlDesk Platform Management*)

Property	Description / Refer to
Processor board	Lets you specify the processor boards belonging to the DS1007 platform. Click  to add a processor board, or click  to delete the selected processor board. **Note** You cannot subsequently add processor boards to a DS1007 system that is already registered. Each processor board has an edit field to specify its connection parameter value and an edit field to specify a unique name for it. After registration, the name is displayed in ControlDesk's Platform/Device Manager. The valid characters are 'a ... z', 'A ... Z', '0 ... 9', '_', '-' and ' '. The name must not start or end with an underline, hyphen or blank. If you do not specify a custom name for a processor board, ControlDesk displays a default name for it in the Platform/Device Manager.
Processing units	Lets you specify the processing units belonging to the SCALEXIO platform. Click  to add a processing unit, or click  to delete the selected processing unit. **Note** You cannot subsequently add processing units to a SCALEXIO system that is already registered. Each processing unit has an edit field to specify its connection parameter value and an edit field to specify a unique name for it. After registration, the name is displayed in ControlDesk's Platform/Device Manager. The valid characters are 'a ... z', 'A ... Z', '0 ... 9', '_', '-' and ' '. The name must not start or end with an underline, hyphen or blank. If you do not specify a custom name for a processing unit, ControlDesk displays a default name for it in the Platform/Device Manager.
Scan for available processor boards/ processing units/ platforms	Lets you scan the local network for connected platform hardware. Depending on the platform type to be registered, ControlDesk opens the Scan Local Network for Processor Boards, the Scan Local Network for Processing Units or the Scan Local Network for Platforms dialog and displays all the platform hardware found in the network. Refer to *Scan Local Network for Processor Boards/ Processing Units /Platforms dialog* (🕮 *ControlDesk Platform Management*).
MAPort Implementation Properties	
MAPort Implementation	Lets you select the MAPort implementation for the XIL API MAPort platform. The list displays all the supported XIL API MAPort implementations. The XIL API MAPort platform analyzes the IMF files in the standard XIL API folder (`ProgramData\ASAM\XIL\Implementation`) to determine the installed XIL API MAPort implementations.
Product name	*MAPort Implementation Properties* (🕮 *ControlDesk Platform Management*)
Product version	*MAPort Implementation Properties* (🕮 *ControlDesk Platform Management*)

Property	Description / Refer to
Vendor name	*MAPort Implementation Properties* (🕮 *ControlDesk Platform Management*)
XIL API version	*MAPort Implementation Properties* (🕮 *ControlDesk Platform Management*)
Multiprocessor Configuration Properties	
Network client	*Assignment Properties* (🕮 *ControlDesk Platform Management*)
Processors	Lets you specify the number of processors belonging to the multiprocessor system. Click  to add a processor, or click  to delete the selected processor. The type of the board to be added (DS1005 or DS1006) depends on the Multiprocessor type property. **Note** You cannot subsequently add members to a multiprocessor system that is already registered.
Processor name	*Common Properties* (🕮 *ControlDesk Platform Management*)
Port address	*Assignment Properties* (🕮 *ControlDesk Platform Management*)

4 Click Register to complete the registration.

The registered platform is displayed with its registration settings in the Registered platforms list.

Tip

If you register a platform with corrupted boot firmware or with hardware components containing different firmware versions, a message box containing warning messages about the detected firmware problem is displayed. The warning messages are also displayed in the Message Viewer. In the Platform/Device Manager, the affected hardware components are marked with the ! symbol. Its tool tip also provides information on the detected firmware problem. You should check the entries and perform firmware updates, if necessary. Refer to *Update Firmware* (🕮 *ControlDesk Platform Management*).

5 Click Close to close the Register Platforms dialog.

Result

You have registered a dSPACE board.

Checking the Configuration of the dSPACE Boards

Objective

After registration, you can read out information on your dSPACE board using ControlDesk.

Where to go from here

Information in this section

How to Check the Configuration

Objective

To check whether the system is ready to experiment with real-time applications.

Check applications

To check whether your dSPACE system works correctly, it is recommended that you load the system check application.

System	File	Location
DS1103	smd_1103_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1103\Check\
DS1104	smd_1104_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1104\Check\
MicroAutoBox	smd_1401_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1401\Check\
MicroLabBox	smd_1202_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1202\Check\
One DS1005	smd_1005_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1005\Check\
Two DS1005	pipt1_dual1005_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1005mp\Dual1005\Check\
One DS1006	smd_1006_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1006\Check\
Two DS1006	pipt1_dual1006_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1006mp\Dual1006\Check\
DS1007	smd_1007_ch.sdf	<RCP_HIL_InstallationPath>\Demos\DS1007\Check\

ControlDesk experiment For each check application, there is a ZIP file with a backup of an experiment referencing the check application. To check the configuration of your specific system, you open the system-specific ZIP file in ControlDesk and load the contained application to the system.

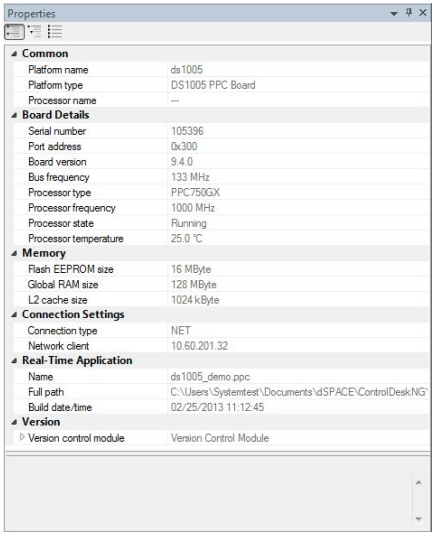
Method	**To check the configuration** **1** Start ControlDesk. **2** On the File ribbon, click Open - Project + Experiment from Backup. ControlDesk opens a standard Open dialog. **3** In the Open dialog, specify the ZIP file that matches your system type and click Open. For the location of the ZIP file, see the table above (refer to Check applications). ControlDesk opens the project and experiment that belong to the check application. **4** From the context menu of the experiment's platform, select Configure Platform/Device to configure the platform according to the dSPACE real-time hardware connected to the host PC. For example, you have to specify the IP address of the hardware. **5** Press F5 to start a measurement. **6** If no error message is displayed in the Log Viewer, the board is installed correctly.
Result	You have checked the configuration.

How to View the Properties of a dSPACE System

Objective	After registration, you can view the properties of the dSPACE system.
Preconditions	The dSPACE board is registered.
Method	**To view the properties of a dSPACE system** **1** On the View ribbon, click Controlbar - Switch Controlbars - Properties to open the Properties controlbar.

2 In the Project Manager, select the corresponding board.

The platform information is displayed in the Properties controlbar. The illustration below shows the properties of a DS1005 board as an example.



Result

You have viewed the properties of your dSPACE system. For a detailed description of the properties, refer to *Platform/Device-Related Properties* (🕮 *ControlDesk Platform Management*).

Updating the Firmware

Objective

Usually your board contains the latest firmware. If you install a new dSPACE Release it may contain newer firmware. In this case the firmware can be updated.

Where to go from here

Information in this section

Basics on Firmware Gives you information on the different kinds of firmware.	40
How to Prepare the Firmware Update Before you start an update or repair process, some preparations have to be made.	48

How to Update Firmware 50
Gives you the instructions for the firmware *update* mode.

Basics on Firmware

Objective

You can execute a real-time application on dSPACE real-time hardware only if the different kinds of firmware are available. The loaded firmware version has to provide the functionality implemented in the real-time application.

Firmware features

The firmware for a hardware component provides basic functionality that is stored in a nonvolatile memory. For example, it includes functions for the communication between the host PC and the hardware, and can also provide I/O functions such as CAN or LIN protocol support, or complex I/O functions for an FPGA component.

Firmware components The following firmware components are used with dSPACE real-time hardware.

Firmware	Meaning
Firmware for the main hardware components	
Boot firmware	Provides essential functions to access the hardware system connected to the host PC. It also contains functions to provide a hardware inventory and to load and start a real-time application. Relevant for any hardware.
CN boot firmware	Provides essential functions for the computation node (CN) to configure the initial hardware settings and to load the CN firmware. Relevant for DS1007 and MicroLabBox.
CN firmware	Provides the functionality for a real-time processor (computation node). Relevant for DS1007 and MicroLabBox.
CN CPU configuration	Provides the hardware configuration of a real-time processor. Relevant for DS1007 and MicroLabBox.
System FPGA firmware	Provides the system functions for the programmable logic device of the processor board. Relevant for DS1007 and MicroLabBox.

Firmware	Meaning
HCN boot firmware	Provides essential functions for the host communication co-processor (host communication node) to configure the initial hardware settings for communication and to load the HCN firmware. Relevant for DS1007 and MicroLabBox.
HCN firmware	Provides the host communication functionality for the host communication node and other services such as USB flight recording. Relevant for DS1007 and MicroLabBox.
Host IF firmware	Provides functions that allow the communication between the host PC and dSPACE real-time hardware that has an Ethernet-based host interface. Relevant for MicroAutoBox.
Host IF PLD firmware	Provides functions for the programmable logic device that controls the host interface communication. Relevant for MicroAutoBox.
System PLD firmware	Provides functions for the programmable logic device that offers system functions, for example, watchdog handling or accessing onboard sensors. Relevant for MicroAutoBox.
DSx86_32 UserFirmware	Provides essential functions to access the hardware system connected to the host PC. It also contains functions to provide a hardware inventory and to load and start a real-time application. Relevant for SCALEXIO.
Firmware for additional hardware components	
AIO TYPE 1 PLD firmware	Provides functions for accessing analog input and output signals via the programmable logic device of the AIO Type 1 module. Relevant for the DS1513 I/O board of MicroAutoBox.
ADC TYPE 4 PLD firmware	Provides functions for accessing analog input signals via the programmable logic device of the ADC Type 4 module. Relevant for the DS1511 and DS1513 I/O boards of MicroAutoBox.
CAN slave firmware	Provides functions for CAN communication via a slave processor or a connected I/O board. Relevant for DS1103, DS2202, DS2210, DS2211, and DS4302 boards.
CAN TYPE 1 firmware	Provides functions for CAN communication via the CAN Type 1 module. Relevant for MicroAutoBox and MicroLabBox.

Firmware	Meaning
DIO TYPE 3 PLD firmware	Provides functions for accessing digital input and output signals via the programmable logic device of the DIO Type 3 module. Relevant for the DS1511 I/O board of MicroAutoBox.
DIO TYPE 4 PLD firmware	Provides functions for accessing digital input and output signals via the programmable logic device of the DIO Type 4 module. Relevant for the DS1513 I/O board of MicroAutoBox.
DS4342 firmware	Provides functions for CAN communication via the DS4342 CAN FD Interface Module mounted on a DS4505 Interface Board or MicroAutoBox.
DS<xxxx> UserFirmware	Provides the basic functions for the related SCALEXIO board, for example, for accessing digital input and output signals or the bus interface. Relevant for SCALEXIO boards.
DS<xxxx> UserIplFirmware	Provides the bootloader for the related SCALEXIO board. Relevant for the SCALEXIO DS6101 Multi-I/O Board.
DS<xxxx> UserFpga	Provides the functions realized on an FPGA module. Relevant for SCALEXIO boards.
DS<xxxx> IoFpga	Provides the I/O functions realized on an FPGA module. Relevant for the SCALEXIO DS2680 I/O Unit.
FPGA Code	Provides the system and I/O functions realized on an FPGA module. Relevant for DS1006.
FPGA TYPE 1 PLD firmware	Provides basic functions for initializing and accessing the FPGA Type 1 module. Relevant for the DS1512 and DS1514 I/O boards of MicroAutoBox.
I/O Clock Buffer Configuration	Provides functions for processing the clock signal on the local bus. Relevant for MicroLabBox.
I/O CPLD firmware	Provides functions for controlling and accessing the I/O modules. Relevant for MicroLabBox.
I/O FPGA firmware	Provides the I/O functions realized on an FPGA module. Relevant for MicroLabBox.
LIN slave firmware	Provides functions for LIN communication via a connected I/O board. Relevant for DS4330.
Slave DSP firmware	Provides system and I/O functions for the slave DSP units. Relevant for DS1103 and DS1104.

Details on the firmware archives

The following firmware archives are available.

DS1005FwArchive.arc The firmware archive for a modular system based on a DS1005 PPC Board contains the following firmware components:

- DS1005 boot firmware
- DS2202 CAN slave firmware
- DS2210 CAN slave firmware
- DS2211 CAN slave firmware
- DS4302 CAN slave firmware
- DS4330 LIN slave firmware
- DS4342 firmware

Note

Using DS802 PHS Link Board

You can update the firmware of I/O boards only if they are supported by the DS802 PHS Link Board. For example, you cannot update the above mentioned CAN slave firmware components, if the boards are connected via DS802 PHS Link Board. For an overview of the supported I/O boards, refer to *DS802 Data Sheet* (📖 *PHS Bus System Hardware Reference*).

DS1006FwArchive.arc The firmware archive for a modular system based on a DS1006 Processor Board contains the following firmware components:

- DS1006 boot firmware
- DS1006 FPGA Code
- DS2202 CAN slave firmware
- DS2210 CAN slave firmware
- DS2211 CAN slave firmware
- DS4302 CAN slave firmware
- DS4330 LIN slave firmware

  (see the note above)
- DS4342 firmware

DS1007FwArchive.arc The firmware archive for a modular system based on a DS1007 PPC Processor Board contains the following firmware components:

- HCN Boot Firmware
- HCN Firmware
- CN Boot Firmware
- CN Firmware
- CN CPU Configuration
- System FPGA Firmware
- DS2202 CAN slave firmware
- DS2210 CAN slave firmware
- DS2211 CAN slave firmware
- DS4302 CAN slave firmware
- DS4330 LIN slave firmware

  (see the note above)
- DS4342 firmware

DS1103FwArchive.arc The firmware archive for a modular system based on a DS1103 PPC Controller Board contains the following firmware components:

- DS1103 Slave DSP firmware

  The slave DSP firmware can be enlarged by user functions, see User firmware. This firmware component must also be used for restoring the boot firmware.
- DS1103 CAN slave firmware

DS1104FwArchive.arc The firmware archive for a modular system based on a DS1104 R&D Controller Board contains the following firmware components:

- DS1104 boot firmware
- DS1104 Slave DSP firmware

MABXFwArchive.arc (for MicroAutoBox) The firmware archive for MicroAutoBox contains the following firmware components:

- DS1401 boot firmware
- DS1401 System PLD firmware
- DS1401 Host IF PLD firmware
- DS1401 Host IF firmware
- ADC TYPE 4 PLD firmware

- DIO TYPE 3 PLD firmware
- DIO TYPE 4 PLD firmware
- FPGA TYPE 1 PLD firmware
- AIO TYPE 1 PLD firmware
- CAN TYPE 1 firmware
- DS4342 firmware

Note

To program the firmware that supports the RTI DS1552 I/O Extension blockset for the DS1552 Multi-I/O module, you have to use the **`DS1401UpdateExtIO`** command, which is described in the *MicroAutoBox RTLib Reference*.

DS1202FwArchive.arc (for MicroLabBox) The firmware archive for MicroLabBox contains the following firmware components:

- HCN Boot Firmware
- HCN Firmware
- CN Boot Firmware
- CN Firmware
- CN CPU Configuration
- System FPGA Firmware
- CAN Type 1 firmware
- I/O clock buffer configuration
- I/O CPLD firmware
- I/O FPGA firmware

SCALEXIOFwArchive.arc The firmware archive for SCALEXIO systems contains the following firmware components:

- DSx86_32 UserFirmware
- DS2502 UserFpga
- DS2551 UserFpga
- DS2601 UserFirmware and DS2601 UserFpga
- DS2621 UserFirmware and DS2621 UserFpga
- DS2642 UserFirmware and DS2642 UserFpga
- DS2655 UserFirmware and DS2655 UserFpga
- DS2655M1 UserFpga
- DS2655M2 UserFpga

- DS2656 UserFirmware and DS2656 UserFpga
- DS2671 UserFirmware and DS2671 UserFpga
- DS2672 UserFirmware and DS2672 UserFpga
- DS2680 UserFirmware, DS2680 UserFpga, and DS2680 IoFpga1 … 3
- DS2690 UserFirmware and DS2690 UserFpga
- DS2907 UserFirmware and DS2907 UserFpga
- DS6051 UserFpga
- DS6101 UserFirmware, DS6101 UserIplFirmware, and DS6101 UserFpga
- DS6201 UserFpga
- DS6301 UserFpga

Note

- The archive format for DS1007 and MicroLabBox changed with Firmware Archives 2.0 contained in dSPACE Release 2015-B. To open an archive in the new format, you must use Firmware Manager 2.0 or later.
- The archive format for SCALEXIO changed with Firmware Archives 2.1 contained in dSPACE Release 2016-A. To open an archive in the new format, you must use Firmware Manager 2.1.

Special firmware

The firmware archives installed with your dSPACE software provide the standard firmware type. There might be other firmware types to be managed with the Firmware Manager.

Custom firmware The Firmware Manager allows you to install custom firmware that dSPACE provides for solutions or engineering projects.

User firmware

Note

When you use a SCALEXIO system, this term is used for the standard firmware. The following restrictions refer only to customized firmware.

User firmware is a firmware that is based on dSPACE firmware but extended with your own functionality. The Firmware Manager does not support loading user firmware.

NOTICE

dSPACE accepts no liability for incorrect operation or property damage when using user firmware with dSPACE hardware.

If you want to add user functions to the slave DSP firmware of a DS1103 PPC Controller Board, follow the descriptions in Ds1103_Slv_Usr_Prog.pdf.

You find this application note, in `C:\Program Files <(x86)>\Common Files\dSPACE\HelpDesk <ReleaseVersion>\Print`.

Default factory firmware MicroAutoBox, MicroLabBox, DS1007, and SCALEXIO are providing a secured mode for using the default factory firmware. If firmware is corrupted, the hardware, automatically reboots, if necessary, and loads the default factory firmware that lets you access the board and retry the firmware update.

For further information, refer to:

- Using MicroAutoBox: *How to Start MicroAutoBox to Secured Mode* (🕮 *MicroAutoBox II Hardware Installation and Configuration*)
- Using MicroLabBox: *How to Force a Restart with Factory Firmware* (🕮 *MicroLabBox Hardware Installation and Configuration*)
- Using modular system based on DS1007: *How to Start the DS1007 to Secured Mode* (🕮 *DS1007 Hardware Installation and Configuration Guide*)

Note

Note the following restriction if you use DS1007, MicroAutoBox, or MicroLabBox.

In secured mode, for example, caused by an interrupted firmware repair or update process, you cannot repair or update the firmware components of the I/O components of the board (connected I/O boards, internal I/O modules, or the I/O FPGA). You firstly have to repair or update the firmware components of the base board and then reboot the board to leave the secured mode. Error messages regarding to the repair or update process of the I/O components can be ignored. After reboot, you can continue the repair or update process for the firmware of the I/O components.

If your real-time hardware does not provide a secured mode for using the default factory firmware, you have to repair the board's boot firmware. You can do this via the command line interface of the firmware management or via the Platform Manager. To repair the corrupted boot firmware via the command line interface, refer to *Examples of Script-Based Firmware Management* (🕮 *Firmware Manager Document*).

How to Prepare the Firmware Update

Objective

The preparation of a firmware update consists of specifying some general firmware settings.

Preconditions

The following preconditions must be fulfilled for configuring the general firmware settings:

- The real-time hardware must be connected to the host PC.
- The real-time hardware must be switched on.
- The required firmware archive must be available.

  You can find the latest firmware archives on the dSPACE website at http://www.dspace.com/go/firmware.
- If a real-time application is loaded to the board's flash memory, it is recommended to clear the flash before starting the update process to avoid unpredictable output signals.

  If a real-time application is running, it is stopped by the firmware management.

Note

- The archive format for DS1007 and MicroLabBox changed with Firmware Archives 2.0 contained in dSPACE Release 2015-B. To open an archive in the new format, you must use Firmware Manager 2.0 or later.
- The archive format for SCALEXIO changed with Firmware Archives 2.1 contained in dSPACE Release 2016-A. To open an archive in the new format, you must use Firmware Manager 2.1.

Method

To prepare the firmware update

1. Open the Platform Manager.
2. If no real-time hardware is displayed in the Platform Manager, register the real-time hardware that you want to update.
3. Choose Update Firmware in the platform's context menu to open the Update Firmware Wizard.

   The wizard starts with the Select Mode dialog.
4. Select the firmware update mode.

   By default, the Update mode is set to update all firmware components of your real-time hardware with later firmware. With the Repair mode enabled, you can select the firmware components to be repaired.

   To switch to the repair mode, select Firmware repair mode in the Select Mode dialog.
5. Click Next to continue with the Select Firmware Archive dialog.

   The latest firmware archive for the selected platform is automatically set. Optionally, browse for another firmware archive. This might be useful if you want to update to a firmware version other than the latest or repair user firmware, for example.
6. Click Next to continue with the Select Firmware Components dialog.

Result

You have configured the settings which are required for a firmware update process in update or repair mode.

If you use a multiprocessor or a multicore system, note the following points:

- If you have registered a multiprocessor system, you can update only one processor at a time.
- If you have registered a multicore system with additional I/O boards, you have to select the core to which the I/O boards are connected for the update of the entire system. The other cores will be updated, too.

Related topics

HowTos

- *How to Repair Firmware* (📖 *Firmware Manager Document*)
- *How to Update Firmware* on page 50

How to Update Firmware

Objective	Gives you the instructions for the firmware *update* mode.
Preconditions	The firmware update process has to be prepared with the Update Mode specified as described in *How to Prepare the Firmware Update* on page 48.
Method	**To update firmware** **1** In the Select Firmware Components dialog, click Update to start the firmware update process. In the Update column, the firmware components to be updated are marked and red. The components are not marked for update if the version of the currently installed firmware is identical to or later than the firmware available in the specified firmware archive.
Result	If there are updatable firmware components, the update process starts. You can see the progress in the Status column. The initial '--' entry is replaced by a percentage. If the progress information cannot be detected continuously, only the states 50% and 100% are displayed. If the process successfully finished, an OK is shown, otherwise an error message is displayed. If the firmware update will require more than 40 minutes, an estimate of the time is displayed. Then you can decide whether to start the process. Interrupting a running firmware update process is not possible. **Note** You must not switch off the hardware during the firmware update process. This will cause a corrupted firmware. Follow the given instructions to complete the firmware update. For example, some firmware components require a hardware reboot.
Related topics	HowTos • *How to Prepare the Firmware Update* on page 48 • *How to Repair Firmware* (📖 *Firmware Manager Document*)

Implementing Models via Simulink/RTI or via Handcoding

Objective

The first step is to implement your model. You can either embed the blocks provided by dSPACE's Real-Time Interface (RTI) in a Simulink model or use RTLib's functions to handcode your application directly in C.

Where to go from here

Information in this section

How to Start RTI

Objective

Real-Time Interface (RTI) is the interface between Simulink and the dSPACE platforms. If you use MATLAB/Simulink to construct real-time models, the C code is automatically generated by the Simulink® Coder™ in conjunction with dSPACE's Real-Time Interface (RTI).

RTI libraries

The RTI libraries contain information, demos, and the icons available with the dSPACE platform. Each dSPACE platform has its own library.

Preconditions

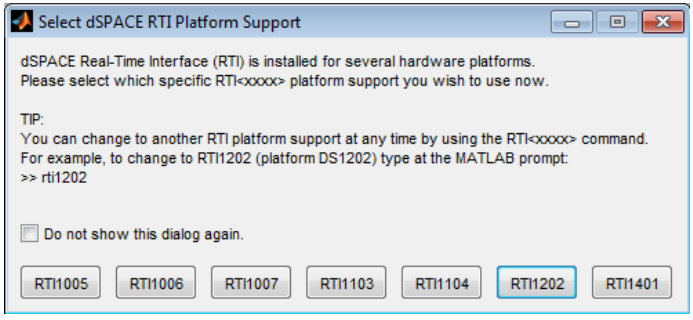
- Ensure that MATLAB, Simulink, Simulink® Coder™, and Real-Time Interface (RTI) for your dSPACE system are properly installed.
- The required licenses must be available and activated. For information on handling the license mechanism, refer to *Basics on dSPACE License Types* (🕮 *Software Installation and Management Guide*).

Method

To start RTI

1 Start MATLAB.

RTI displays a dialog for you to select the desired platform.



Tip

If you select the Do not show this dialog again checkbox, MATLAB always starts with the platform that was active last.

2 In the MATLAB Command Window, enter **`rti`**. The corresponding library is automatically opened according to your platform.

3 To switch between platforms, enter **`rti<name of the platform>`** as listed in the table below:

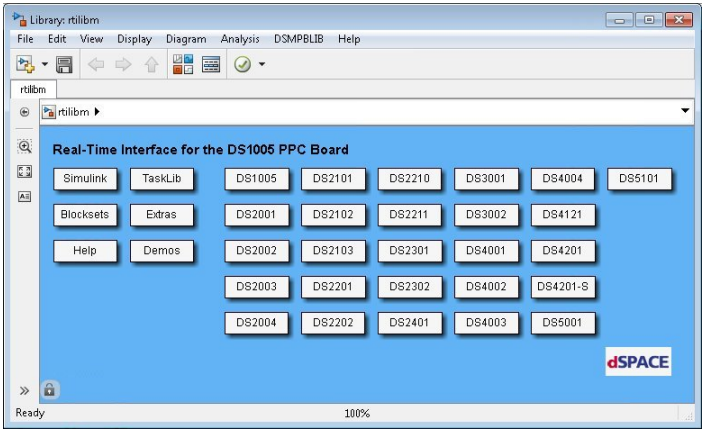
Platform	Command
DS1103	**`rti1103`**
DS1104	**`rti1104`**
MicroAutoBox	**`rti1401`**
MicroLabBox	**`rti1202`**
DS1005	**`rti1005`**
DS1006	**`rti1006`**
DS1007	**`rti1007`**
Multiprocessor system	**`rtimp`**

Result

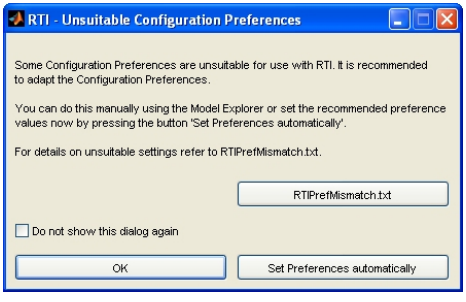
You started RTI and opened a dSPACE board library.

Example

You have chosen RTI1005 as your platform. Enter `rti` in the MATLAB Command Window to display the rtilibm RTI library for modular systems.



If you want to switch to another platform, for example, RTI1104, enter **`rti1104`**. A dialog opens, asking you whether you want to adapt the build options for the new platform automatically.



Click OK to open the RTI1104 library.

Next step

When the library is opened, you can work with a Simulink model. Refer to *How to Work with a Simulink Model* on page 54.

Related topics

HowTos

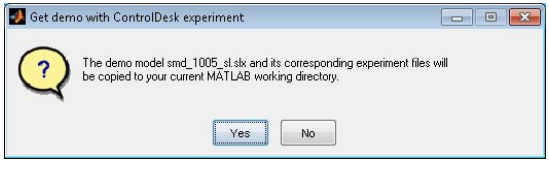
- *How to Work with a Simulink Model* on page 54

How to Work with a Simulink Model

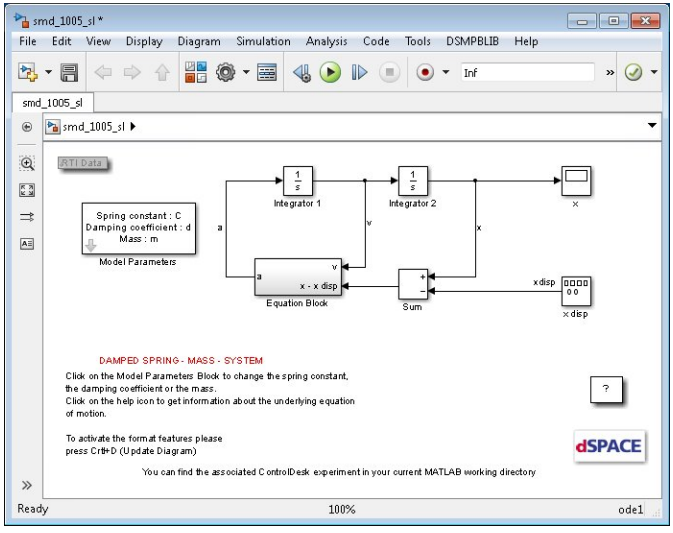
Objective	A demo shows you how to work with a Simulink model.
Demo model	The `smd_<xxxx>_sl` demo model is an example of how to use the RTI library. It simulates a damped spring-mass system stimulated by a square-wave signal. It is a ready-to-use example that does not require any I/O hardware. The related model files are located in the `<RCP_HIL_InstallationPath>\Demos\DS<xxxx>\GettingStarted\Simulink` folder.
Precondition	Before you can start the build and download procedure, you have to ensure that your dSPACE hardware is registered correctly in ControlDesk. Refer to *How to Register a dSPACE Board* on page 31.
Method	**To work with a Simulink model** **1** In the Library: rtilib<xxxx> window, double-click Demos. **2** Double-click the box titled Spring - Mass - Damper in the Library: rti<xxxx>demolib window. The Get demo with ControlDesk experiment dialog opens. **3** In the Get demo with ControlDesk experiment dialog, click Yes to copy the demo model and the experiment files to your current MATLAB working directory.

Tip

For multiprocessor systems, there are equivalent examples in the subfolders of `<RCP_HIL_InstallationPath>\Demos\DS<xxxx>MP\Dual<xxxx>\GettingStarted\Simulink`.



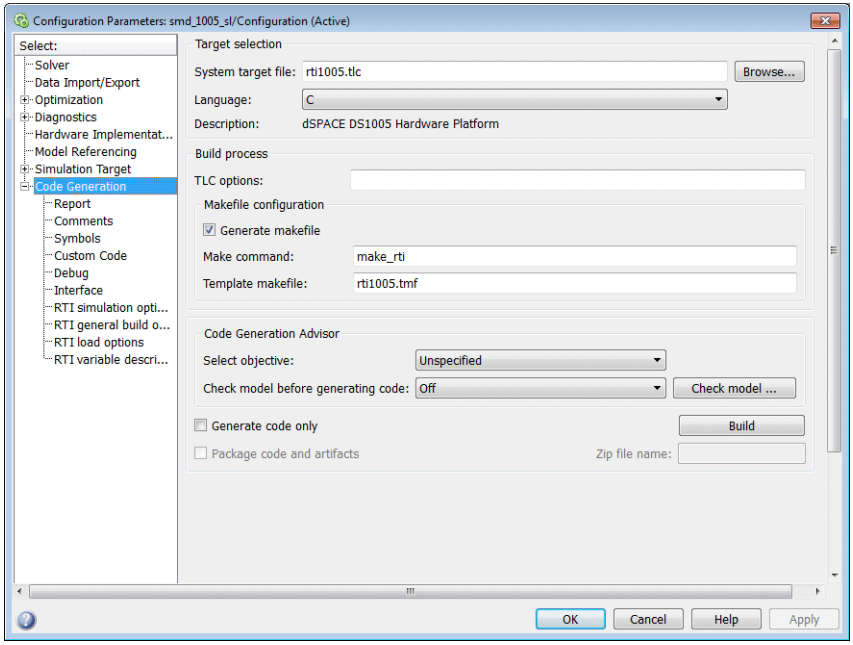
The Simulink model is then displayed in a separate window.



4 To examine the model, double-click its blocks.

5 From the menu bar of the model window, select Code – C/C++ Code – Code Generation Options to view the configuration parameters for Simulink® Coder™.

The Configuration Parameters dialog opens.



6 On the Code Generation page, check that the system target file (`rti<xxxx>.tlc`) and the template makefile (`rti<xxxx>.tmf`) match the processor board, and that the make command `make_rti` has been specified. If necessary, change the entries (according to the illustration above).

7 Click Build to build the real-time application and download it to the dSPACE board.

The system description file (`smd_<xxxx>_sl.sdf`) and the real-time application are created. The latter is downloaded to your dSPACE board and started automatically.

If the real-time application — stored in the current MATLAB working folder — is stopped, it can be downloaded to the dSPACE board again using the SDF file in ControlDesk. No new build process is needed if you do not change the Simulink model in the meantime.

Result

The real-time application is built and downloaded to the dSPACE board.

Next steps	Use dSPACE experiment software to stop and restart the real-time application, refer to *Handling Real-Time Applications with ControlDesk* on page 59.
Related topics	HowTos • *How to Implement a Model via Handcoding* on page 57 • *How to Start RTI* on page 51

How to Implement a Model via Handcoding

Objective	A demo model shows you how to work with a C-coded model.
C-coded demo model	The C-coded example `smd_<xxxx>_hc.c` demonstrates how to proceed without MATLAB and Simulink. This model does not require any I/O hardware. It simulates a damped spring-mass system stimulated by a square-wave signal. The parameters are spring, mass and damper.
down<xxxx> utility	To compile and link the real-time application, you can use the `down<xxxx>` utility in a Command Prompt window. After the real-time application is downloaded and started, the RTP error check utility is called automatically.
Method	**To implement a model via handcoding** **1** Extract the `<RCP_HIL_InstallationPath>\Demos\DS<xxxx>\GettingStarted\HandCode\smd_<xxxx>_hc.zip` file to a folder for which you have write permission. **2** Open the file `.\smd_<xxxx>_hc\Source Files\smd_<xxxx>_hc.c` in a text editor of your choice to inspect the code. **3** Open the `Command Prompt for dSPACE RCP and HIL` in the `dSPACE RCP and HIL` start menu folder. **4** In the `Command Prompt for dSPACE RCP and HIL`, change to the folder you extracted the ZIP file to. **5** Enter the following command: `down<xxxx> smd_<xxxx>_hc`
Result	The `smd_<xxxx>_hc` real-time application is generated, downloaded and started.

Example

For example, if you work with DS100x boards:

- Enter **down1005 smd_1005_hc** for the DS1005.

  This generates the `smd_1005_hc.ppc` real-time application file.
- Enter **down1006 smd_1006_hc** for the DS1006.

  This generates the `smd_1006_hc.x86` real-time application file.
- Enter **down1007 smd_1007_hc** for the DS1007.

  This generates the `smd_1007_hc.rta` real-time application file.

Next step

Use dSPACE experiment software to stop and restart the real-time application, refer to *Handling Real-Time Applications with ControlDesk* on page 59.

Related topics

HowTos

- *How to Start RTI* on page 51
- *How to Work with a Simulink Model* on page 54

Handling Real-Time Applications with ControlDesk

Objective

This section describes how to handle applications on real-time processors (RTP).

Where to go from here

Information in this section

Information in other sections

How to Download an Application to the Program Memory and Start the Real-Time Processor

Objective

You can load real-time applications to the program memory and start the real-time processor in ControlDesk's Platform/Device Manager.

Method

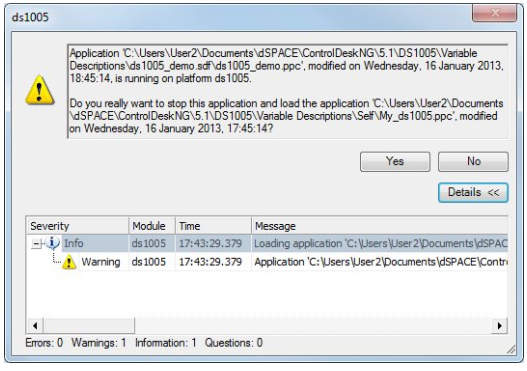
To download an application to the program memory and start the Real-Time Processor

1 In the Platform/Device Manager, open the context menu of your platform icon (or multiprocessor icon) and select Real Time Application - Load.

The Select Real-Time Application dialog opens.

2 In the Select Real-Time Application dialog, select a PPC file (x86 file for a DS1006, RTA file for a DS1007 or a MicroLabBox).

3 If a previously loaded application is still running on your system, a dialog similar to the following opens:



If you click Yes, the running application is stopped before the new application is downloaded and started.

Result

You have downloaded an application to the program memory and started the real-time processor.

Related topics

HowTos

- *How to Download an Application to the Flash Memory and Start the Real-Time Processor* on page 62

How to Stop a Real-Time Application

Objective

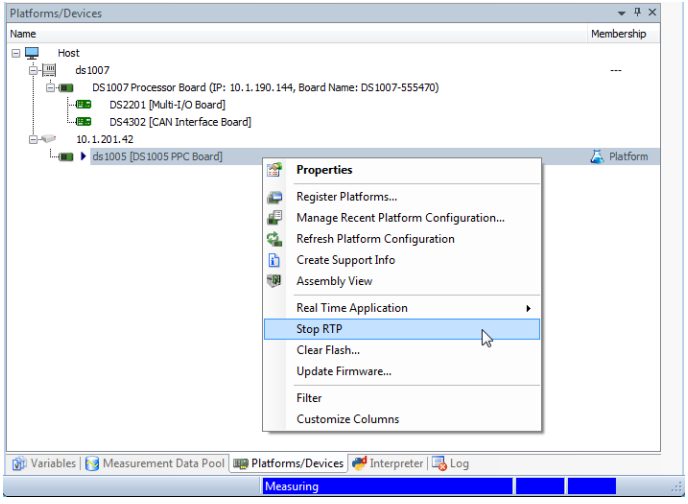
You can use ControlDesk's Platform/Device Manager to stop an application running on a real-time processor.

Method

To stop a real-time application

1 In the Platform/Device Manager, open the context menu of your platform. Depending on your platform, select Stop or Stop RTP.

The illustration below shows the context menu of the DS1005 as an example.



Result

You have stopped the real-time application.

Related topics

HowTos

- *How to Reload a Real-Time Application* on page 61

How to Reload a Real-Time Application

Objective

You can use ControlDesk's Platform/Device Manager to reload an application running on a real-time processor.

Restrictions

- You cannot reload a multiprocessor application. You have to download it again. Refer to *How to Download an Application to the Program Memory and Start the Real-Time Processor* on page 60.
- You cannot reload an application for a modular system containing DS230x boards. You have to download it again. Refer to *How to Download an Application to the Program Memory and Start the Real-Time Processor* on page 60.

Method

To reload a real-time application

1 In the Platform/Device Manager, open the context menu of your board icon and select Real Time Application - Reload.

Tip

For some platforms, the command is available in the context menu of the application to be reloaded.

Result

You have reloaded a real-time application.

Related topics

HowTos

- *How to Stop a Real-Time Application* on page 61

How to Download an Application to the Flash Memory and Start the Real-Time Processor

Objective

A flash memory is used to load a real-time application automatically after power-up. You must first download the real-time application to the flash memory.

Flash memory

Flash memory of dSPACE boards Several dSPACE boards are equipped with a flash memory for real-time applications. The flash memory can be used to load a real-time application automatically after power-up of the board.

The following table shows which dSPACE boards have a flash memory for autobooting a real-time application.

dSPACE Board	Description
DS1103	No flash memory
DS1104[1)]	Flash memory
MicroAutoBox II	Flash memory
MicroLabBox	Flash memory
DS1005	Flash memory
DS1006	Flash memory on the CompactFlash card
DS1007	Flash memory

1) The DS1104 cannot detect a reboot of the host PC. As a consequence, a real-time application in the DS1104's flash memory is not started automatically when the host PC is rebooted. To start a real-time application from the flash memory, the host PC must be disconnected from the mains for a short time. To do so, switch off the power supply or disconnect the power connector.

Loading the application on power up On power-up, the dSPACE board always starts executing the bootstrap loader contained in the flash memory. The loader checks for an application program currently stored in the flash memory. If it finds one, the application is started. If it does not detect an application in the flash memory, the loader enters the idle state and waits for commands from the connected host PC.

Method

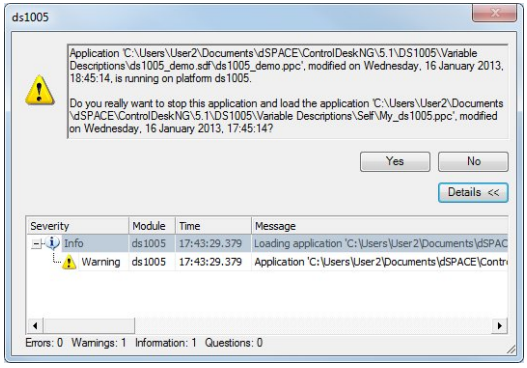
To download a real-time application to the flash memory and start the real-time processor

1. In the Platform/Device Manager, open the context menu of your platform and select Real Time Application - Load to Flash.

   The Select Real-Time Application dialog opens.

2. In the Select Real-Time Application dialog, select a PPC file (x86 file for a DS1006, RTA file for a DS1007 or a MicroLabBox) and click Open.

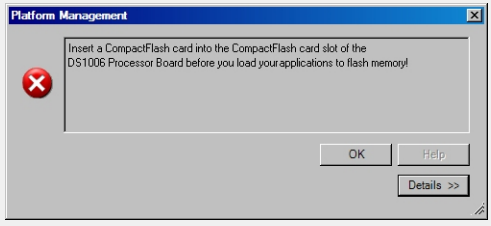
3 If a previously loaded application is still running on your system, a dialog similar to the following opens:



If you click Yes, the running application is stopped before the new application is downloaded and started.

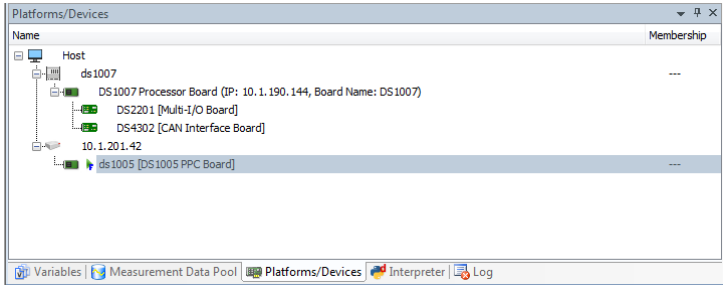
Note

If you try to load an application when your CompactFlash card is not inserted, a dialog similar to the following opens:



Result

You have downloaded an application to the flash memory and started the real-time processor. The F next to the board icon in the Platform/Device Manager indicates that the running application is loaded from the flash memory.



Related topics

HowTos

- *How to Download an Application to the Program Memory and Start the Real-Time Processor* on page 60

How to Clear an Application from the Flash Memory

Objective

If you want to prevent the system from booting a flash application, you have to clear the application from the flash memory.

DS1006 CompactFlash card removal

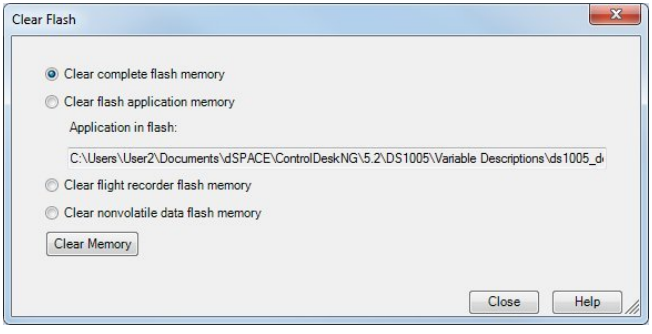
DS1006: A DS1006 uses the flash memory of a CompactFlash card, which is removable. You can remove the CompactFlash card to avoid an automatic start. Remove it only while the expansion box is switched off.

Method

To clear an application from the flash memory

1 In the Platform/Device Manager, open the context menu of your platform icon and select Clear Flash.

The Clear Flash dialog opens.



2 In the Clear Flash dialog, select one of the following options to clear the memory (the available options depend on the selected platform type):

- Clear complete flash memory to clear the whole flash memory.
- Clear application flash memory to clear only the loaded application from the application flash memory. The currently loaded application is displayed in the Application in flash field.

- Clear nonvolatile data flash memory to clear only data that is stored as nonvolatile. Flight recorder data is stored in the nonvolatile memory and therefore also cleared.

3 Click Clear Memory to clear the flash memory according to your selection.

Result

You have cleared an application from the flash memory.

Related topics

HowTos

- *How to Download an Application to the Flash Memory and Start the Real-Time Processor* on page 62

How to Download and Start an Application via Command Line

Objective

To download and start applications on the real-time hardware, you can use the `cmdloader` tool.

cmdloader A command line tool for handling applications without using the GUI of ControlDesk.

Using the cmdloader tool

The `cmdloader` tool is located in the `.\Main\bin` folder of your ControlDesk installation and in the `<RCP_HIL_InstallationPath>\Exe\CmdLoader\bin` folder of your RCP & HIL installation. When you use the tool, you have to specify the entire path to it. If the path contains blanks, you have to enclose the path in quotation marks.

Example For example, to display the `cmdloader` help, enter

```
"C:\Program Files (x86)\dSPACE ControlDesk 6.0\Main\bin\cmdloader" -?
```

in a DOS window.

Tip

To use the `cmdloader` tool in a DOS window without having to specify the path to the tool, select All Programs - dSPACE ControlDesk 6.0 - Utilities - Platform Management Loader from the Windows Start menu.

This opens a DOS window. You do not have to specify the path as long as this DOS window is open.

For details on `cmdloader` and its parameters, refer to *Handling Applications via Command Line* (🕮 *ControlDesk Platform Management*).

Preconditions

You must have registered the dSPACE system(s) since the `cmdloader` tool requires registration information.

Method

To download and start an application via command line

1 Select All Programs - dSPACE ControlDesk 6.0 - Utilities - Platform Management Loader from the Start menu.

This opens a DOS window. You do not have to specify the path as long as this DOS window is open.

2 In the DOS window, enter `cmdloader` with valid parameters as shown below:

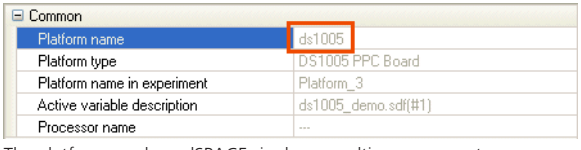
```
cmdloader <{application [-suppress_start]} | -start | -stop>
                <-p platform_name | {-rnc net_client -t platform_type}>
                [-go seconds] [-fl] [-ol output_level] [-q] [-ra] [-?]
```

The table below shows the most important command parameters:

Parameter	Description
`application`	The application to download to the selected platform. You must specify the file extension (SDF, PPC, RTA or x86). You also have to specify the relative or absolute path to the application. If the path contains blanks, you have to enclose the path in quotation marks, e.g.,: `"C:\Program Files\dSPACE ControlDesk 6.0\Demos\RTApplications\DS1005\ds1005_demo.sdf"`
`-p platform_name`[1), 2)]	To scan the recent hardware configuration for the platform (specified via `platform_name`), to try to register the platform, and to load the application (specified via the `application` parameter) to the platform.
`-fl`	To load the application (specified via the `application` parameter) to the flash memory of the platform (specified via the `-p platform_name` parameter).
`-ra`	To scan the recent hardware configuration for platforms that are not yet registered and to try to register them.

1) Mandatory parameter

2) The platform name as displayed in the Properties controlbar:



The platform can be a dSPACE single- or multiprocessor system.

Result

You have downloaded and started an application via command line.

Examples

The following examples show how to use the `cmdloader` tool for batch operations.

- **`cmdloader -p ds1104 ds1104_demo.sdf`**

  The `cmdloader` tool loads the `ds1104_demo.sdf` application to a platform named `ds1104` and starts the application.

- **`cmdloader -p ds1401 -fl ds1401_demo.sdf`**

  The `cmdloader` tool loads the `ds1401_demo.sdf` application to the flash memory of a platform named `ds1401` and starts the application.

- **`cmdloader -ra`**

  The `cmdloader` tool scans the recent hardware configuration for platforms that are not yet registered and tries to register them.

- **`cmdloader -ra -p ds1005 ds1005_demo.sdf`**

  This example combines the use of the following parameters:

  - `-ra`
  - `-p platform_name`
  - `application`

Related topics

Basics

- *Handling Applications via Command Line* (📖 *ControlDesk Platform Management*)

HowTos

- *How to Download an Application to the Flash Memory and Start the Real-Time Processor* on page 62
- *How to Download an Application to the Program Memory and Start the Real-Time Processor* on page 60

Experimenting with ControlDesk

Objective	To experiment with real-time applications, use ControlDesk.
Where to go from here	Information in this section

Demo Projects/Experiments and Related Files

Objective	You can use demo projects/experiments for your platform to experiment with ControlDesk features. The RT application demo projects allow you to work with ControlDesk and use all of its features with dSPACE real-time hardware connected to the host PC.
Accessing the demo projects/experiments	There is a demo project/experiment and related files for each platform in your ControlDesk installation folder. For example, the DS1005 demo is located in `../Demos/RTApplications/DS1005/`.
Related files	Demo projects/experiments contain the following files: **PPC/x86/RTA file** The real-time application file to be downloaded to the hardware. **TRC file** Provides information on available variables and how they are grouped. Trace files are either generated by RTI/Simulink or hand-coded. **MAP file** Maps names of variables to addresses of the physical memory. It is generated by the C compiler. **SDF file** The system description (SDF) file specifies which executable is downloaded to which processor.

How to Measure Variable Values

Objective

To measure the variable values of a running real-time application, you have to connect an instrument to the variables.

Tip

A successful measurement also confirms that your ControlDesk installation is working correctly.

Working without the demo project/experiment

To show you how to set up a project/experiment and perform a measurement, the instructions below show all the steps from defining a project/experiment to starting a measurement. For this reason, the instructions do *not* use the prepared demo project/experiment, but only the variable description file contained in the demo.

Preconditions

The board must be registered. Refer to *How to Register a dSPACE Board* on page 31.

Method

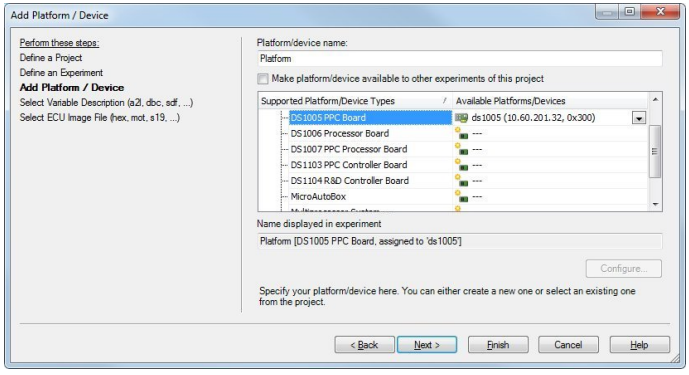
To measure variable values

1. Start ControlDesk.
2. On the File ribbon, click New - Project + Experiment to define a new project/experiment.

   The Define a Project dialog opens.
3. In the Define a Project dialog, enter a name for the new project.

   Click Next > to open the Define an Experiment dialog.
4. In the Define an Experiment dialog, enter a name for the new experiment and click Next > to open the Add Platform / Device dialog.

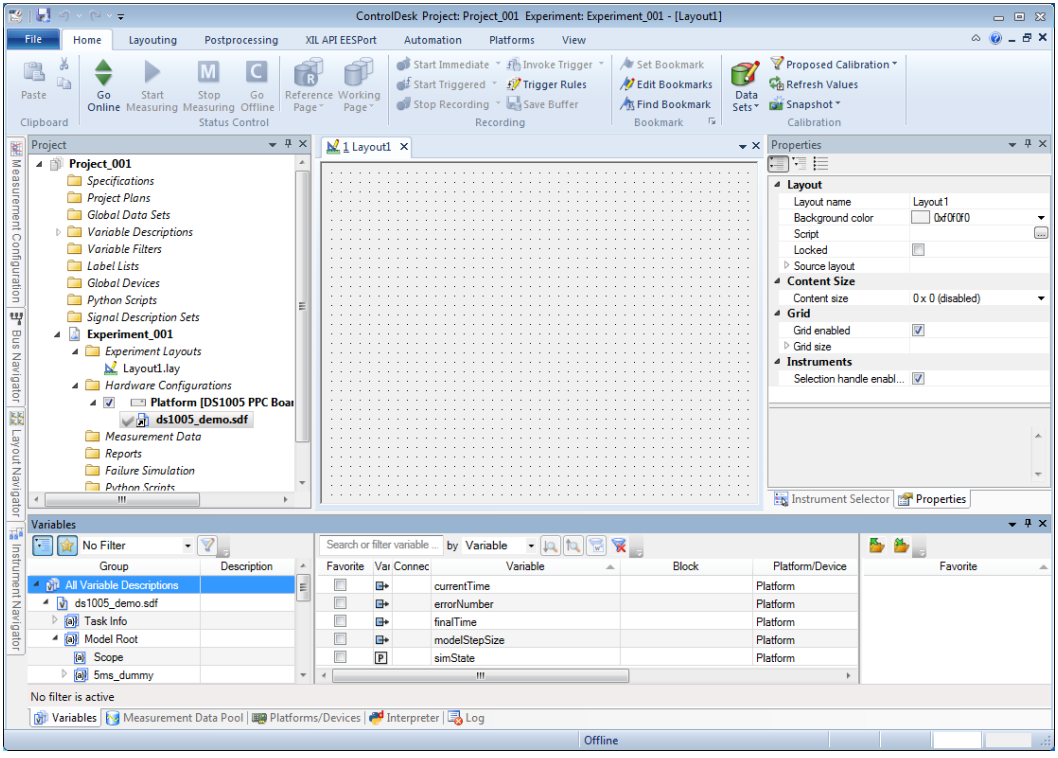
5 In the Add Platform / Device dialog, add the desired platform or device and click Next > to open the Select Variable Description dialog.



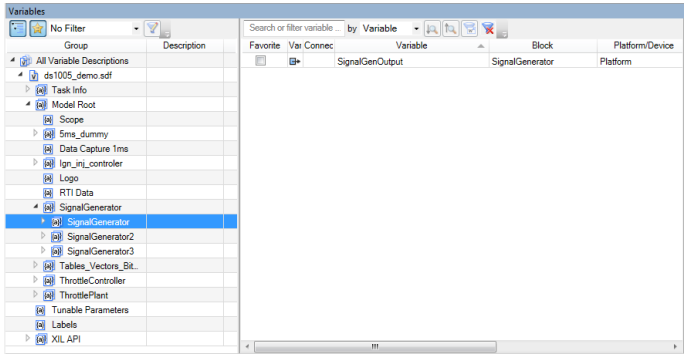
6 In the Select Variable Description dialog, click Import from file to navigate to the installation folder of your ControlDesk installation, select the `<platform>_demo.sdf` variable description file for your real-time hardware from the `\Demos\RTApplications\<platform>` folder and click Open. For example, select `../Demos/RTApplications/DS1005/ds1005_demo.sdf` for a DS1005.

7 Click Finish.

The new project/experiment opens.

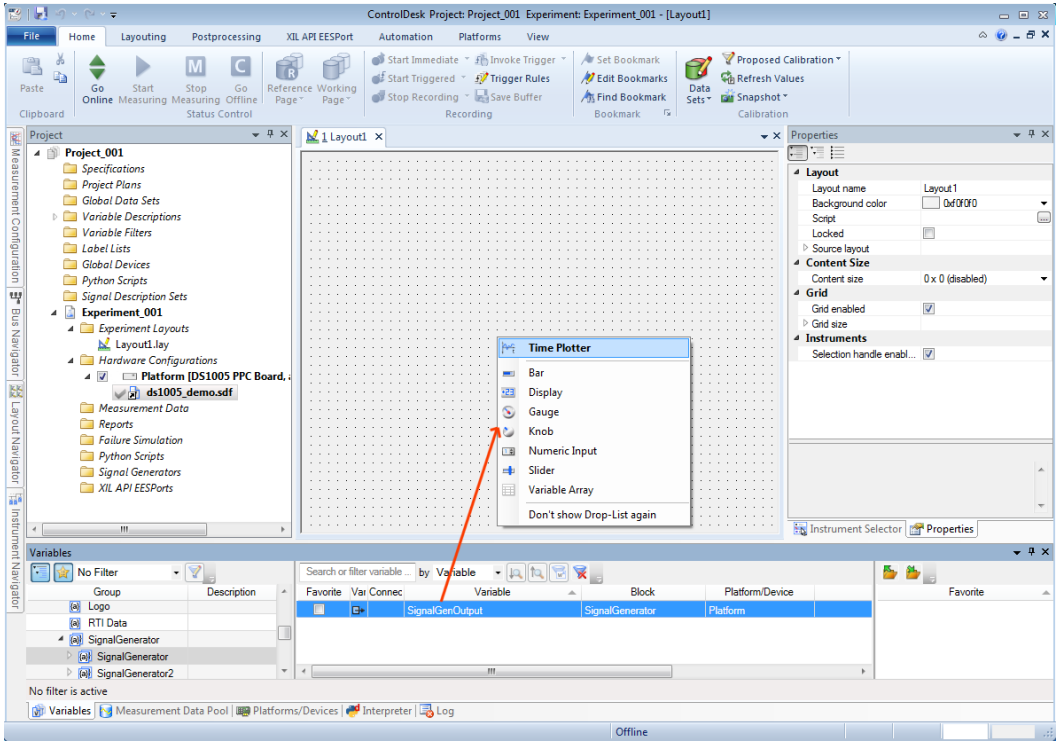


8 In the tree view of the Variable Browser, navigate to `DS1005_demo.sdf/Model Root/SignalGenerator/SignalGenerator`.



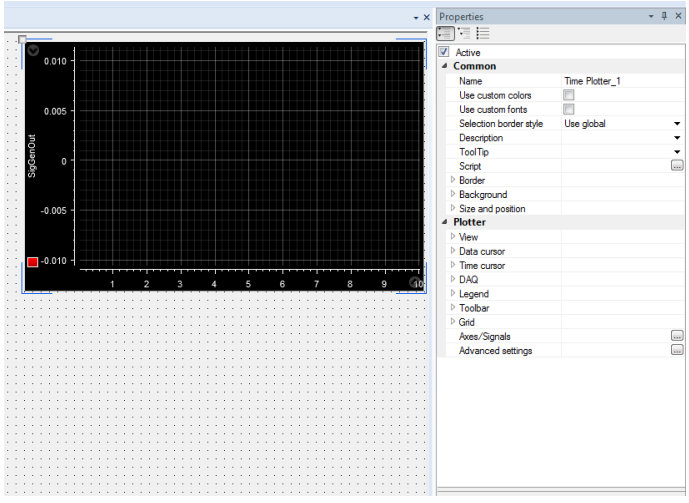
9 In the Variable list, select the SignalGenOutput variable and drag it to the new layout.

10 In the Instrument Type list, click Time Plotter.



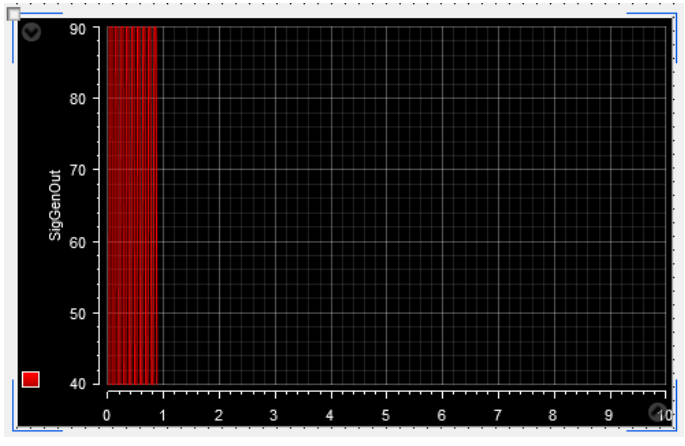
The Time Plotter opens. You have connected the SignalGenOutput variable to the Time Plotter.

11 To display or change the properties of the Time Plotter, click the instrument. The Properties controlbar shows the plotter properties.



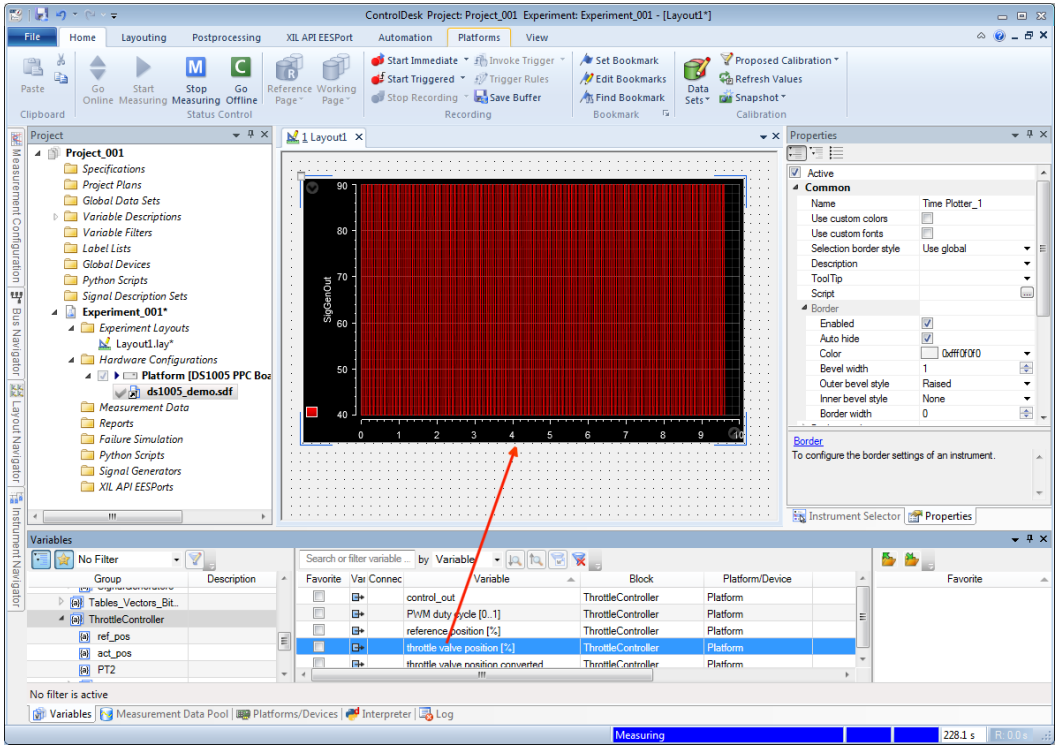
12 Press F5 to measure the variable values.

The illustration below shows a running measurement.

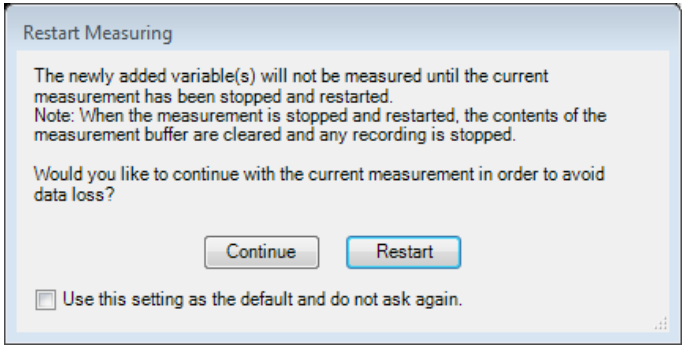


13 To add another variable to your measurement, change to the Throttle Controller variable group in the tree view of the Variable Browser and select the throttle valve position [%] variable in the Variable list.

14 Drag it to the Time Plotter.

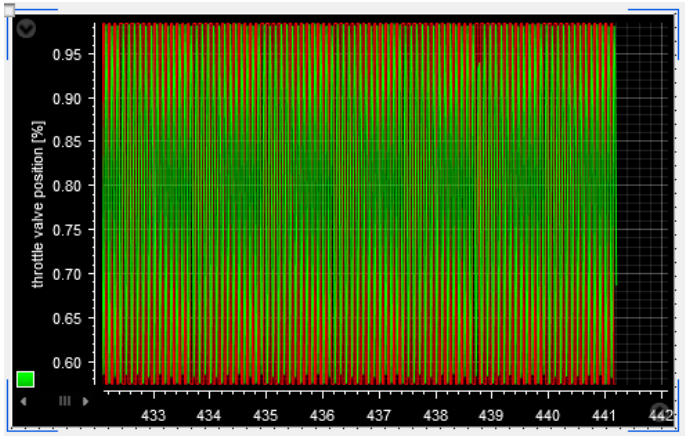


15 The Restart Measuring dialog opens and asks you either to keep the current measurement or to stop it and start a new one, because the added variable cannot be measured until a restart is performed.



Click Restart to perform the measurement again.

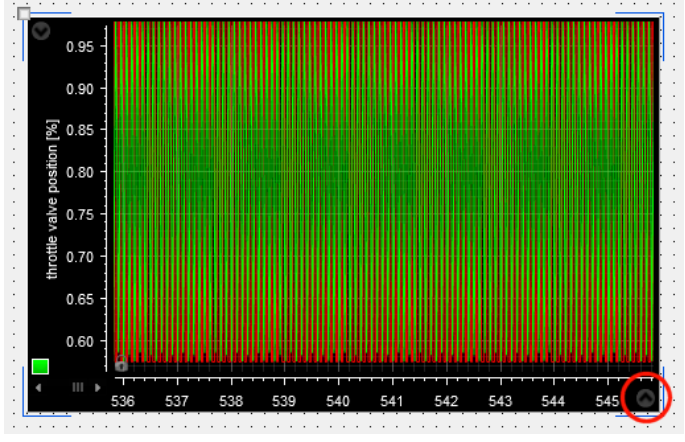
You have added another variable to your measurement.



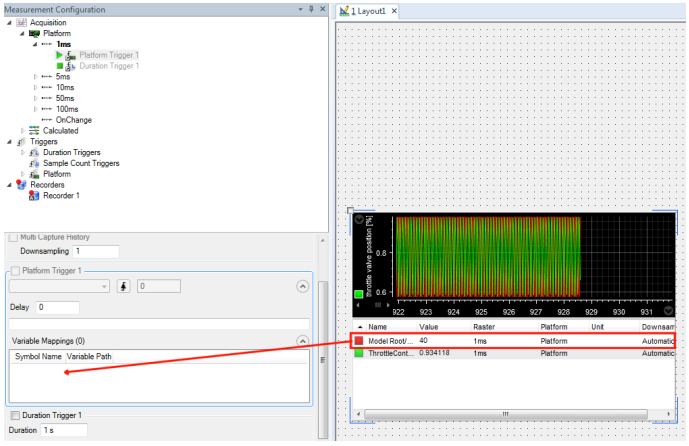
With ControlDesk, you can perform triggered measurements on dSPACE real-time hardware. Triggered measurement means that data capture with a measurement raster is not continuous, but started and stopped by triggers. In the following steps, you will specify a start trigger (indicated by a ▶ symbol), and a stop trigger (indicated by a ■ symbol) for the measurement raster.

16 In the Measurement Configuration controlbar, click Platform - 1ms - Platform Trigger 1. The trigger currently is grayed.

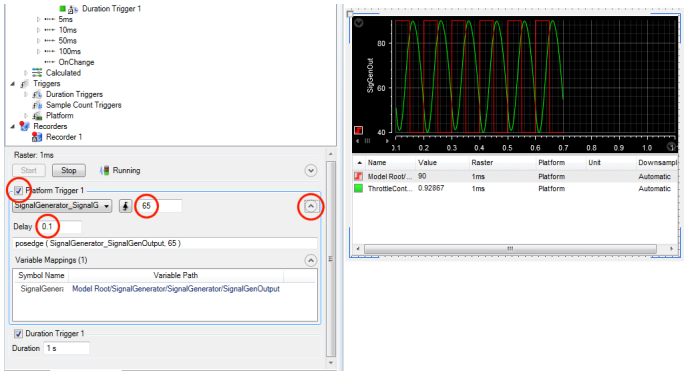
17 Expand the Time Plotter to get a list of all plotted variables by clicking the expand button in the lower right corner.



18 Drag the SignalGenOutput variable to the expanded Variable mappings field as shown below.



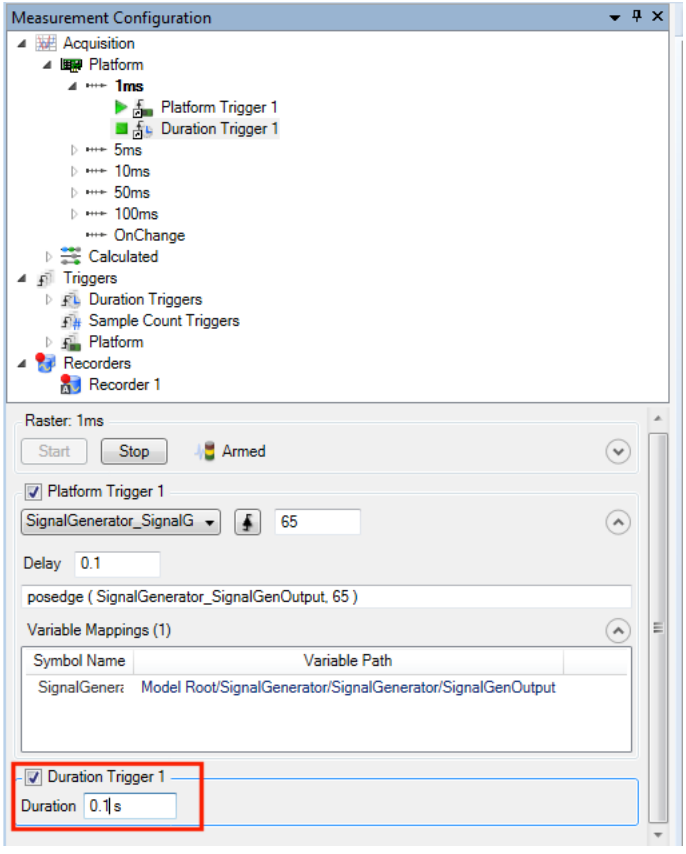
19 Make the specifications as shown in the following illustration:



Setting	Value
Platform Trigger 1 checkbox	Selected
Reference Value edit field	**65**
Delay [s] edit field (click the expand button if hidden)	**0.1**

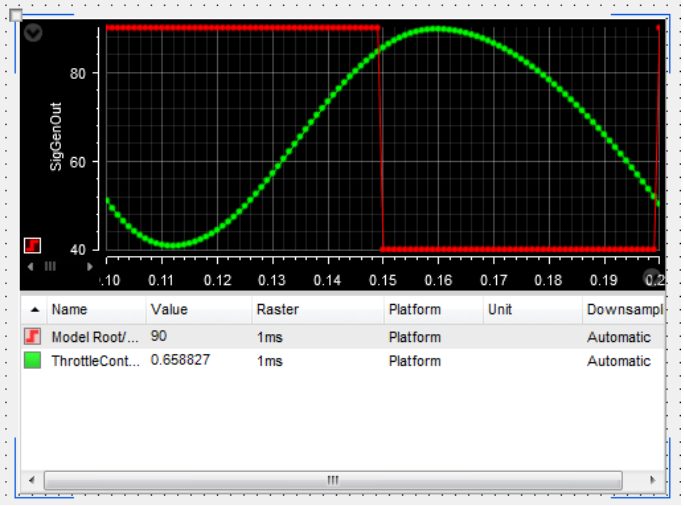
20 In the Measurement Configuration controlbar, click Platform - 1ms - Duration Trigger 1.

21 Make the specifications as shown in the following illustration:



Setting	Value
Duration Trigger 1 checkbox	Selected
Duration edit field	`0.1 s`

The plotter now displays the data stream of the variable values when triggered.



22 On the File ribbon, click Save Project or press **`Ctrl + Shift + S`** to save the project/experiment.

Result	You have measured the values of the variables.
Next steps	For instructions on changing the parameters of a running application with ControlDesk, refer to *How to Change Parameter Values of a Running Application* on page 79.
Related topics	Basics • *Demo Projects/Experiments and Related Files* on page 69 HowTos • *How to Change Parameter Values of a Running Application* on page 79

How to Change Parameter Values of a Running Application

Objective	You can change the parameter values of a running application with ControlDesk.
Demo project/experiment	For a short description of the demo project/experiment, refer to *Demo Projects/Experiments and Related Files* on page 69.

Preconditions

The board must be registered. Refer to *How to Register a dSPACE Board* on page 31.

Method

To change parameter values of a running application

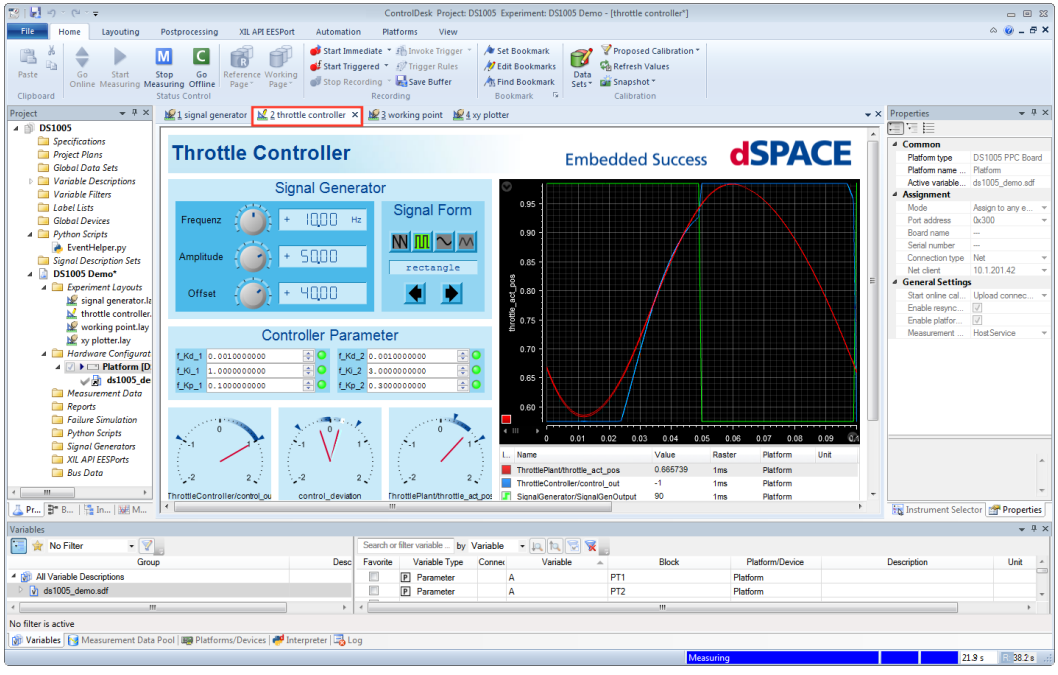
1 Start ControlDesk.

2 On the File ribbon, click Open – Open Project + Experiment to open the DS1005 Demo project/experiment.

ControlDesk opens the project/experiment together with all the related files including the SDF file.

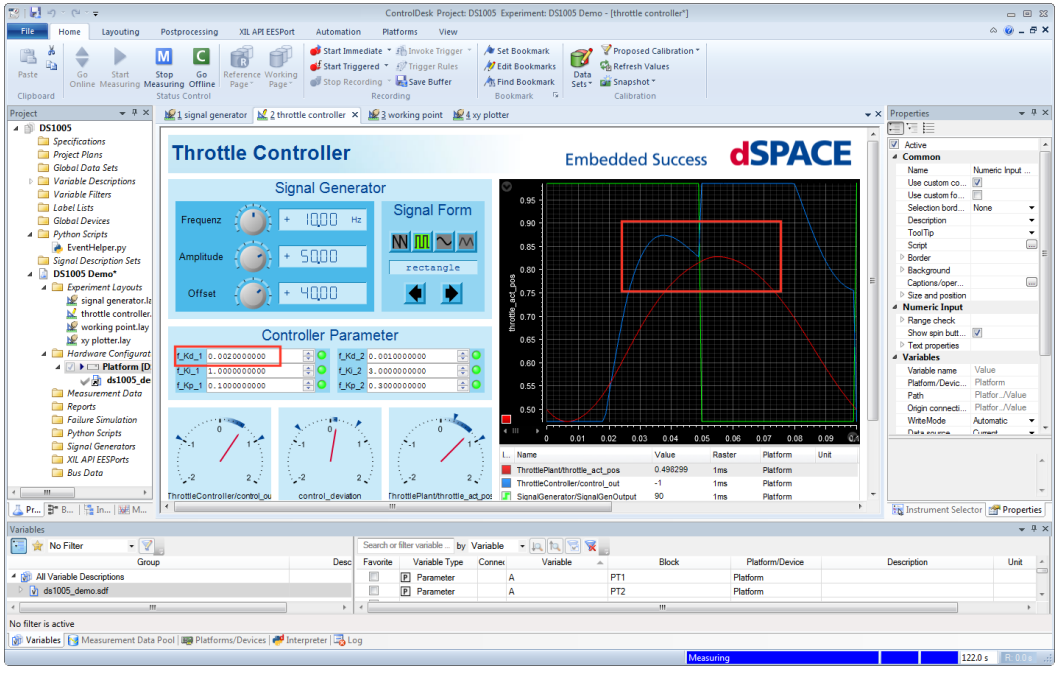
3 On the Home ribbon, click Status Control – Start Measuring.

4 Change to the throttle controller layout.



5 To change a parameter value while measuring, enter a new variable value: In the value field of the f_Kd_1 variable, change the value from `0.001` to `0.002`.

You can see that the plotter display changes slightly.



You have changed the parameter values during the measurement.

Result

You have changed parameter values of a running application.

Related topics

Basics

- *Demo Projects/Experiments and Related Files* on page 69

HowTos

- *How to Measure Variable Values* on page 70

First Work Steps Glossary

A

Acquisition An object in the *Measurement Configuration* controlbar that specifies the variables to be measured and their measurement configuration.

Automatic Reconnect Feature for automatically reconnecting to platform/device hardware, for example, when the ignition is turned off and on, or when the physical connection between the ControlDesk PC and the ECU is temporarily interrupted.

If the feature is enabled for a platform/device and if the platform/device is in the '*unplugged*' state, ControlDesk tries to reestablish the logical connection to the platform/device hardware. After the logical connection is reestablished, the platform/device has the same state as before the unplugged state was detected. A measurement started before the unplugged state was detected is resumed.

B

Bus connection A mode for connecting dSPACE real-time hardware to the host PC via bus. The list below shows the possible bus connections:

- dSPACE real-time hardware installed directly in the host PC
- dSPACE real-time hardware installed in an expansion box connected to the host PC via dSPACE link board

C

Calibration Changing the *parameter* values of *real-time applications*, *ECU application* (*ControlDesk Introduction and Overview*)s, or *VPU* (*ControlDesk Introduction and Overview*)s.

cmdloader A command line tool for handling applications without using the GUI of ControlDesk.

Connected A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware or VPU.
- A platform/device must be in the 'connected' state before it can change to the 'measuring/recording' or 'online calibration started' state.
- Online calibration is impossible. ControlDesk did not yet adjust the memory segments containing calibration data in the platform/device and on the corresponding hardware or VPU. Offline calibration is possible.
- Platform/device configuration is not possible. However, you can invoke platform/device configuration for a platform/device that is in the connected state. ControlDesk temporarily sets the platform/device to the disconnected state.

The 'connected' platform/device state is indicated by the icon.

Connection mode dSPACE real-time systems can be installed within the host PC or connected to the host via a bus interface and/or via Ethernet. When the Ethernet is being used, different network clients might exist. The connection type being used and, in the case of Ethernet, the network client being used, determine the dSPACE systems that can be accessed.

Controlbar A window or pane outside the working area. Can be docked to an edge of the main window or float in front of it. A controlbar can contain a document, such as a layout, or a tool, such as the Message Viewer. It can be grouped with other controlbars in a window with tabbed pages.

Controller board Single-board hardware computing the real-time application. Contains a real-time processor for fast calculation of the model and I/O interfaces for carrying out the control developments.

D

Disabled A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware or VPU.
- When a platform/device is disabled, ControlDesk does not try to establish the logical connection for that platform/device. Any communication between the platform/device hardware or VPU and ControlDesk is rejected.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disabled' platform/device state is indicated by the  icon.

Disconnected A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware or VPU.
- When a platform/device is in the disconnected state, ControlDesk does not try to reestablish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disconnected' platform/device state is indicated by the  icon.

Display An instrument (or a value cell type of the *Variable Array*) for displaying the value of a scalar variable or the text content of an ASCII variable.

dSPACE system A hardware system such as MicroAutoBox or SCALEXIO on which the *real-time application* runs.

Duration trigger A *trigger* that defines a duration. Using a duration trigger, you can, for example, specify the duration of data acquisition for a *measurement raster*. A duration trigger can be used as a *stop trigger* only.

E

Environment model A model that represents a part or all of the ECU's environment in a simulation scenario.

Ethernet connection A mode for connecting dSPACE real-time hardware to the host PC via Ethernet. The list below shows the possible Ethernet connections:

- dSPACE real-time hardware installed in an expansion box connected to the host PC via Ethernet.
- MicroAutoBox and MicroLabBox connected via Ethernet.

Expansion box A box that hosts dSPACE boards. It can be connected to the host PC via bus connection or via network.

Experiment A container for collecting and managing information and files required for a parameter calibration and/or measurement task. A number of experiments can be collected in a project but only one of them can be active.

F

Firmware update An update for the firmware installed in the board's flash memory. Firmware should be updated if it is older than required by the real-time application to be downloaded.

Flight recording The recording of data on a hardware device that does not require a physical connection between the host PC and the hardware device.

G

Gigalink module A dSPACE board for connecting several processor boards in a multiprocessor system. The board allows high-speed serial data transmission via fiber-optic cable.

I

Instrument An on-screen representation that is designed to monitor and/or control simulator variables interactively and to display data captures. Instruments can be arranged freely on *layout*s.

Instrument Navigator A *controlbar* that displays a tree with all the *instrument*s of the active *layout* and all the variables that are connected to them. The Instrument Navigator's main function is easy selection of instruments in complex layouts.

Instrument Selector A *controlbar* that provides access to ControlDesk's *instrument*s. The instruments can be placed on a *layout* via double-click or drag and drop.

Instrumentation The entirety of functions, menus, commands, and dialogs for working with instruments.

Interpreter controlbar A controlbar that can be used to execute line-based commands. It is used by the Interpreter to print out Python standard error messages and standard output during the execution or import of Python scripts.

L

Layout A window with *instrument*s connected to variables of one or more simulation models.

Layout Navigator A *controlbar* that displays all opened *layout*s. It can be used for switching between layouts.

M

Measurement Viewing and analyzing the time traces of *variable*s, for example, to observe the effects of ECU parameter changes.

ControlDesk provides various *instrument*s for measuring variables.

Measurement Configuration A *controlbar* that allows you to configure measurement and recording.

Measurement raster Specification of how often a value of a *variable* is updated during a *measurement*. A measurement raster is derived from a *measurement service* (📖 *ControlDesk Introduction and Overview*).

Message Viewer A *controlbar* displaying a history of all error and warning messages that occur during work with ControlDesk.

MicroAutoBox platform A platform that provides access to a MicroAutoBox connected to the host PC for function prototyping purposes such as bypassing.

Modular system A dSPACE processor board and one or more I/O boards connected to it.

O

Offline State in which the parameter values of hardware or a VPU in the current experiment cannot be changed. This applies regardless of whether or not the host PC is physically connected to the hardware.

The *mirrored memory* (📖 *ControlDesk Introduction and Overview*) allows parameter values to be changed even offline.

P

Parameter Any variable type that can be calibrated.

Platform A software component representing a simulator where a simulation application is computed in real-time (on dSPACE real-time hardware) or in non-real-time (on VEOS).

Platform trigger A *trigger* that is available for a *platform* and that is evaluated on the related dSPACE real-time hardware or VEOS.

Platform/Device Manager A software component represented by a *controlbar*. It provides functions to handle devices, platforms, and the applications assigned to the platforms.

Processor board A board that computes real-time applications. It has an operating system that controls all calculations and communication to other boards.

Project A container for collecting and managing the information and files required for experiment/calibration/modification tasks in a number of experiments. A project collects the experiments and manages their common data.

Project Manager A software component represented by a *controlbar*. It provides access to projects and experiments and all the files they contain.

Properties controlbar A *controlbar* providing access to the properties of platforms/devices, layouts/instruments, and measurement/recording configurations.

R

Real-time application An application that can be executed in real time on dSPACE real-time hardware. A real-time application can be built from a Simulink model containing RTI blocks, for example.

Recorder An object in the *Measurement Configuration* controlbar that specifies and executes the *recording* of variables according to a specific measurement configuration.

Recording Saving the time traces of variables to a file. Both measurement variables and parameters can be recorded. Recorded data can be *postprocessed* (📖 *ControlDesk Introduction and Overview*) directly in ControlDesk.

A recording can be started and stopped immediately or via a trigger:

- Immediate recording

  The recording is started and stopped without delay, without having to meet a trigger condition.

- Triggered recording

  The recording is not started or stopped until certain trigger conditions are met. These conditions can be defined and edited in ControlDesk.

S

SDF file The variable description file that describes the files to be loaded to the individual processing units of a simulation platform.

Single-processor system A system that is based on one dSPACE processor or controller board. The other system type is a *Multiprocessor System platform* (📖 *ControlDesk Introduction and Overview*).

Slave application An application assigned to the *slave DSP* of a controller or I/O board. It is usually loaded and started together with the *real-time application* running on the corresponding main board.

Slave DSP A DSP subsystem installed on a controller or I/O board. Its *slave application* can be loaded together with the *real-time application* or separately.

Stop trigger A *trigger* that is used, for example, to stop a *measurement raster*.

T

Time Plotter A *plotter instrument* (📖 *ControlDesk Introduction and Overview*) for displaying signals that are measured in a time-based raster (time plots).

Topology A description of the processor boards belonging to a multiprocessor system and their interconnections via Gigalinks. The topology also contains information on which Gigalink port of each processor board is connected to the Gigalink ports of other processor boards in the multiprocessor system.

Topology information is contained in the real-time application (PPC/x86/RTA) files of the multiprocessor system's processor boards.

TRC file A variable description file with information on the variables available in an *environment model* running on a dSPACE *platform*.

Trigger A condition for executing an action such as starting and stopping a *measurement raster* or a *recorder*.

The generic term for the following services:

- *Duration trigger*
- *Platform trigger*

U

Unplugged A platform/device state defined by the following characteristics:

- The logical connection between ControlDesk and the hardware was interrupted, for example, because the ignition was turned off or the ControlDesk PC and the hardware were disconnected.
- Before the state of a platform/device changes to 'unplugged', the platform/device was in one of the following states:
  - 'Connected'
  - 'Online calibration started'
  - 'Measuring' / 'Recording'

  Tip

  A device for which the connection between ControlDesk and the device hardware currently is interrupted is also set to the "unplugged" state when you start online calibration if both the following conditions are fulfilled:

  - The device's Start unplugged property is enabled.
  - The Start online calibration behavior property is set to "Ignore differences".

  This is possible for CCP and XCP devices. For details on the two properties listed above, refer to *General Settings Properties* (🕮 *ControlDesk Platform Management*).
- If the automatic reconnect feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk periodically tries to reestablish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'unplugged' platform/device state is indicated by the ⚠ icon.

V

Variable Any *parameter* or *measurement variable* (*ControlDesk Introduction and Overview*) defined in a *variable description*. ControlDesk provides various *instrument*s to visualize variables.

Variable Array An instrument for calibrating parameters and displaying measurement variable values.

The Variable Array can be used for the following variable types:

- Measurement ( )
- Measurement array ( )
- String ( )
- Struct ( )
- Struct array ( )
- Value ( )
- Value block ( )

Variable Browser The Variable Browser is a controlbar that provides access to the variables of the currently open experiment.

Variable description A file describing the variables in a simulation application, which are available for measurement, calibration, and stimulation.

Visualization The representation of *variable*s in *instrument*s:

- *Measurement variable* (*ControlDesk Introduction and Overview*)s are visualized in instruments to view and analyze their time traces.
- *Calibration parameters* are visualized in instruments to change their values.

Index