DS1007 PPC Processor Board

Features

Release 2021-A – May 2021

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

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

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/patches for software updates and patches.

Important Notice

This publication and the contents hereof are subject to change without notice.

Contents

About This Document

Content

This document provides feature-oriented access to the information you need to implement your control models on a DS1007 modular system. A DS1007 modular system is a PHS-bus based system that is assembled from a DS1007 PPC Processor Board.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	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, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
?	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
📖	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%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.

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%\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 dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the 📄 icon in dSPACE Help. The PDF opens on the first page.

Introduction to the Features of the DS1007

Introduction

The DS1007 PPC Processor Board is based on the Freescale P5020 (DualCore) processor. This real-time processor (RTP) built on the Freescale Power Architecture® technology forms the main processing unit. This is used to execute the real-time application. Via the board's PHS bus it can access modular I/O boards. The connection to the host PC is realized via Ethernet interface.

Where to go from here

Information in this section

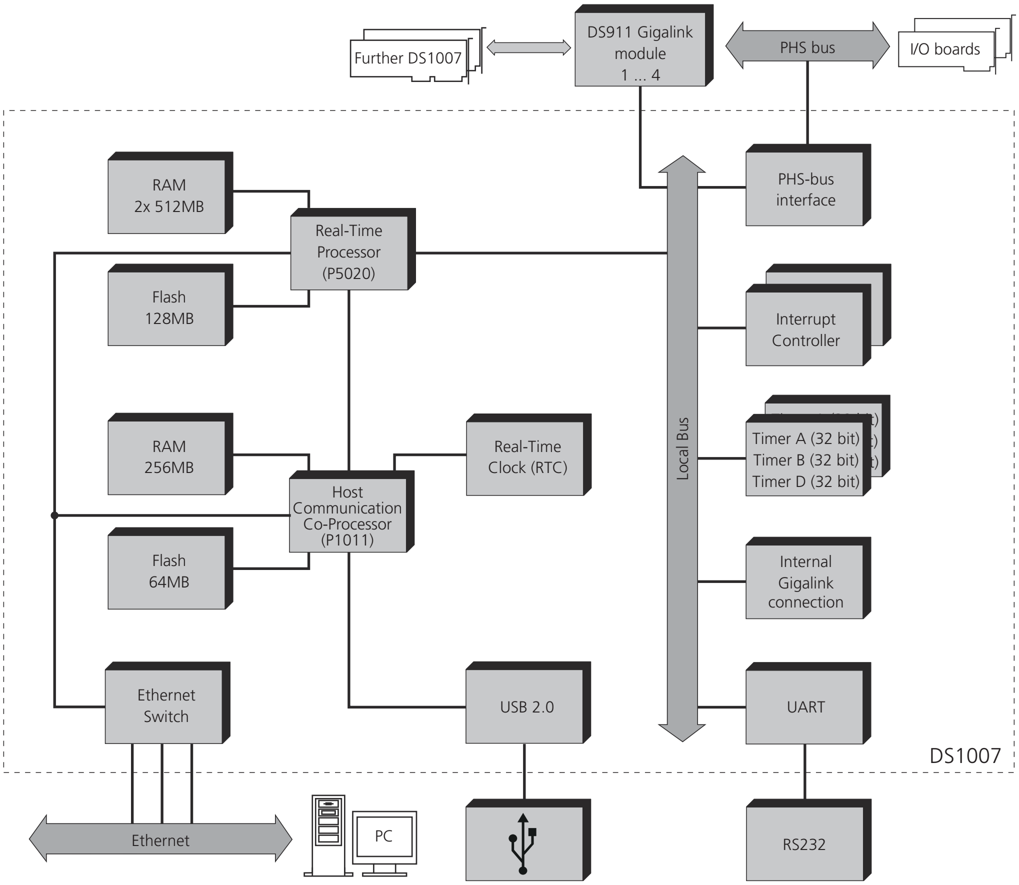
Information in other sections

DS1007 PPC Processor Board (PHS Bus System Hardware Reference 🕮)

Block Diagram

System overview

The following illustration gives an overview of the functional units of the DS1007 PPC Processor Board:



Feature Overview

Introduction

To give you an overview of the basic features of the board.

Processor board features

The DS1007 PPC Processor Board provides the following features:

Real-Time Processor The real-time processor (RTP) is used to run the real-time application, see Processor on page 13.

The dual-core processor allows you to run real-time applications implemented for a single-processor platform (running on one processor core) or for a multiprocessor platform (running on both processor cores).

With the optional DS911 Gigalink module, you can connect multiple DS1007 boards building a multiprocessor system, see DS1007 Multiprocessor Systems on page 35.

Memory Comprises RAM, flash, and cache memory, see Memory on page 14.

Timers and Time Base Counters Comprises:

- Timer A and Timer D on page 30: Sample rate timer with interrupt function
- Timer B on page 31: Interval timer with interrupt function
- Time-Stamp Counter on page 31: Time base for single-processor systems
- Synchronization on page 46: Synchronized time base for multiprocessor/multicore systems

See Timers and Time-Stamp Counters on page 30.

Interrupt control Provides various hardware and software interrupts, see Interrupt Controller on page 33.

PHS bus Allows access to up to 16 I/O boards, see PHS Bus Interface on page 15.

Gigalink module (optional) For setting up multiprocessor systems, see DS911 Gigalink Module on page 40.

Serial interface For setting up a user-specific communication, see Serial Interface of the DS1007 on page 16.

Host interface For setting up the DS1007, downloading programs and transferring run-time data to and from the host PC, see Host Interface on page 17.

USB interface For accessing a USB mass storage device to store flight recording data and/or boot an application, see USB Interface on page 19.

Ethernet I/O interface For setting up Ethernet-based communication to external devices using UDP/IP or TCP/IP protocol, see Ethernet I/O Interface on page 17. The host interface also uses this interface.

Features of the Processor Board

Introduction

The processor board provides features to fulfill the basic requirements of a real-time application.

Where to go from here

Information in this section

General Information

Introduction

To give you a short overview on the board's hardware.

Where to go from here

Information in this section

Information in other sections

DS1007 PPC Processor Board (PHS Bus System Hardware Reference 📖)

Physical Characteristics

Introduction

To give you information about the main physical characteristics. For detailed information, refer to DS1007 PPC Processor Board (PHS Bus System Hardware Reference 📖).

Installation and power supply

The DS1007 requires one slot in an expansion box.

The board has an active cooling via cooling fan.

The power supply has to provide 9 A on 5 V rail and 0.8 A on 12 V rail.

Interfaces available on the bracket

The board provides the following connectors on its bracket:

- Three Ethernet connectors (RJ45)

  All three connectors are identically connected to an internal Gigabit Ethernet switch.
- One USB slot to connect a USB mass storage device for flight recording and loading an application
- One USB eject connector, that allows you to connect the USB eject cable that is used to safely unmount a USB mass storage device before you disconnect it.

A status LED on the bracket gives you information on the board's status.

Interfaces available on the board

The PHS bus connector is used to connect the processor board with the I/O boards of your PHS-bus-based system.

Connection to the host PC

The board is connected to the host PC via Ethernet.

For further information, refer to Using the Web Interface (DS1007 Hardware Installation and Configuration Guide 📖).

Processor

Characteristics of the real-time processor

The DS1007 uses the Freescale P5020 processor as the real-time processor (RTP). It is also called computation node (CN). The RTP calculates your real-time models and accesses the I/O boards via the PHS bus. This processor has the following characteristics:

- Dual-core processor with 2 GHz CPU clock
- Cache sizes:

Cache Type	Cache Size
L1 data cache	32 KByte per core
L1 instruction cache	32 KByte per core
L2 cache	512 KByte per core
L3 cache	2 MByte total

- Temperature sensor to warn about possible CPU overheating:

  The CPU temperature is continuously monitored. If the temperature exceeds the predefined limit, a host message is generated.

For detailed information on the Freescale P5020 processor, refer to the Freescale web site at http://www.freescale.com and search for "P5020".

Characteristics of the host communication co-processor

The DS1007 uses the Freescale P1011 processor as host communication co-processor. It is also called host communication node (HCN). The co-processor handles the entire communication with the host PC as well as data streaming to the USB mass storage device.

For detailed information on the Freescale P1011 processor, refer to the Freescale web site at http://www.freescale.com and search for "P1011".

Memory

Characteristics

The DS1007 real-time processor has the following memory features.

The memory of the host communication co-processor is only internally used and not relevant to custom applications.

Memory Type	Available Memory Size
RAM	< 1 GB DRAM The available local memory is reduced by the memory that is allocated by the required real-time services.
FRAM	Available Memory Size: 64 KB The FRAM has a size of 128 KB, but only 64 KB can be used with the NVDATA functions because of the double-buffer update mechanism.
Flash	32 MB You can use 32 MB out of the 128 MB built-in flash memory to store real-time applications and customer-specific libraries. The rest of the memory is reserved, for example, for the boot firmware.
Memory for flight recording	Depends on the size of the connected USB mass storage device and whether the device is also used for booting an application. The recommended maximum size is 32 GB.

Interfaces

Introduction

The DS1007 has several interfaces for data communication with the connected I/O boards, the host PC and external devices.

Where to go from here

Information in this section

Information in other sections

DS1007 PPC Processor Board (PHS Bus System Hardware Reference 📖)

PHS Bus Interface

PHS bus characteristics

The processor controls the modular I/O boards via the PHS bus, which is a 32-bit I/O bus.

The PHS bus fully supports all PHS-bus I/O boards, including those that support the improved PHS++ bus standard.

The PHS bus is assigned to the board and not to the processor cores. The available bandwidth must be therefore shared between the CPU cores.

Technical characteristics are:

- Peak transfer rate: 20 MByte/s, even higher for the improved PHS++ bus standard
- Eight interrupt lines for I/O boards
- Up to eight hardware interrupts from an I/O board per interrupt line

Control lines

In addition to the standard control lines for reading and writing, the PHS bus provides the following special control lines:

IOERROR This line can be activated by any of the connected I/O boards to indicate that an error occurred. The I/O boards that support the IOERROR line can react individually to errors: For example, a DAC board could hold its output voltage or set it to zero. The DS1007 can set or read this line.

> **Note**
>
> Whenever the DS1007 is reset or an application has been unloaded, it activates the IOERROR line.

SYNCIN, SYNCOUT These lines allow a synchronized reading (SYNCIN) or writing (SYNCOUT) of I/O data across several I/O boards. It is also possible to trigger the lines individually via Timer A or Timer B. To use SYNCIN and SYNCOUT, the involved I/O boards need to be configured accordingly.

Partitioning the PHS bus

With the DS802 PHS Link Board, you can spatially partition the PHS bus by arranging the I/O boards in several expansion boxes. For further information, refer to Partitioning a PHS-Bus-Based System with the DS802 PHS Link Board (DS1007 Hardware Installation and Configuration Guide 🕮).

Related topics

References

PHS-Bus Handling (DS1007 RTLib Reference 🕮)

Serial Interface of the DS1007

UART characteristics

The DS1007 is equipped with a serial interface (UART) to communicate with standard RS232 devices. The UART allows data transfer rates between 5 Baud and 230.4 kBaud.

Pinout

For information on the connector pinout, refer to UART RS232 Connector (PHS Bus System Hardware Reference 🕮).

Specifying baud rates

You can specify the baud rate for serial communication with the processor board in the range 5 … 230.400 baud. Using RTI or RTLib, you can specify any baud rate in this range.

Related topics

References

Serial Interface (DS1007 RTI Reference 📖)
Serial Interface Communication (DS1007 RTLib Reference 📖)

Host Interface

Basics

The DS1007 uses a standard Ethernet connection with TCP/IP protocol as its host interface.

The host interface is used to:

- Connect the board to a host tool such as ControlDesk.
- Download a real-time application to the board.
- Transfer runtime data to and from the host PC.
- Display and configure the board via its web interface.

By entering `http://<IPAdress>` in a standard web browser, you can open the web interface for configuring and managing the board.

This tool provides the following functions:

- Specifying the board's network configuration.
- Configuring the Ethernet switch.
- Managing the onboard flash memory.
- Managing applications on a connected USB mass storage device.
- Generating a system status report to provide information to dSPACE Support.
- Displaying host messages generated by the board without ControlDesk connected.
- Rebooting the system in different modes.

For further information, refer to Basics on the Web Interface (DS1007 Hardware Installation and Configuration Guide 📖).

Ethernet I/O Interface

Basics

The DS1007 board provides three RJ45 connectors for Ethernet communication. The connectors are internally managed by an Ethernet switch. Each of the connectors is therefore equivalent. While one connector is always to be used for the host PC communication, the other two connectors can be individually configured for I/O communication or host PC communication.

Field of application

You can use the Ethernet I/O Interface to perform the following I/O activities:

- Bypassing an ECU, that means, that a subset of ECU functions is executed by the DS1007 modular system by using the RTI Bypass Blockset and the XCP on UDP/IP Protocol.
- Performing communication between the real-time application and any external device that also provides an Ethernet interface using TCP/IP or UDP/IP protocol, such as another dSPACE board or a calibration device. The RTI Ethernet Blockset lets you implement the communication in a Simulink® model. For further information, refer to RTI Ethernet Blockset Reference 📖 .

Data transfer

The Ethernet I/O Interface allows Gigabit connections and supports the TCP/IP and UDP/IP protocols. This interface is optimized to provide low data transfer latencies and high data transfer rates.

Supported features and limitations If you want to implement network-based communication, you should have a basic knowledge of IP-based networks. Here is a list of some basic features and limitations.

The following general features are supported:

- UDP/IP protocol and TCP/IP protocol
- 10/100/1000 Mb data rates
- Half duplex/Full duplex modes
- Up to 256 sockets
- Auto negotiation with Auto-MDIX
- DHCP
- Gateway configuration
- Local loopback
- Interrupt generation when data has been received
- IP fragmentation
- UDP-specific features:
  - Broadcast
  - Dynamic change of destination addresses
- TCP-specific features:
  - Client and server mode

  TCP-specific limitation:
  - A server socket cannot accept more than one client socket to be connected to.

XCP on UDP/IP In combination with the RTI Bypass Blockset, the Ethernet I/O Interface lets you access external systems via the XCP on UDP/IP protocol. This protocol allows you to access ECUs for bypassing, controlling, sensor emulation, etc. Additionally, it allows you to access modules providing an XCP on UDP/IP service interface, for example, to read values from modules which incorporate specific measurement technology.

Specifying the IP addresses

Because the IP protocol is used, you must specify the IP addresses of the Ethernet I/O Interface (source) and the target device (e.g., ECU).

With the RTI Bypass Blockset, the RTIBYPASS_SETUP_BLx block provides the relevant IP address parameters, refer to Options Page (RTIBYPASS_SETUP_BLx for XCP on UDP/IP) (RTI Bypass Blockset Reference 🕮).

Using the RTI Ethernet Blockset, the ETHERNET_SETUP_BLx block provides the relevant IP address parameters, refer to ETHERNET_SETUP_BLx (RTI Ethernet Blockset Reference 🕮).

Related topics

Basics

Features of the RTI Bypass Blockset (RTI Bypass Blockset Reference 🕮)

References

RTI Ethernet Blockset Reference

USB Interface

Basics

The DS1007 board provides one USB 2.0 connector. It can be used for flight recording and booting an application. For detailed information, refer to USB Flight Recorder on page 24 and Running an Application from a USB Mass Storage Device on page 53.

Nonvolatile Data Handling (NVDATA)

Where to go from here

Information in this section

General Information on Handling Nonvolatile Data

Introduction

Gives you details about the features and information on how to access the board's nonvolatile memory.

Characteristics

The nonvolatile data handling feature lets you access the board's nonvolatile memory via a real-time application. The values that you want to write to the memory or that you want to read from it must be provided by a data set.

- Up to 64 data sets can be handled.
- Up to 64 elements can be contained in one data set.

  Only elements of the same data type can be used in a data set.
- The maximum memory size available for nonvolatile data is 64 kB.
- The data transfer is protected by a double buffer mechanism.
- The data rate for writing a data set is limited to 10 MB/s.

Note

> The board's nonvolatile memory is a global memory. The data sets can be therefore accessed by several applications running on the hardware. This might cause conflicts, e.g., if an application tries to create a data set with a name that already exists but with a different data type.

To manage the nonvolatile data without running a real-time application, you can use the board's web interface, refer to Using the Web Interface for Nonvolatile Data Handling on page 21.

Supported data types

The elements of a data set must be of the same data type. The following data types can be specified:

Data Type	Meaning
Int8	8-bit integer values Allocates 1 byte
UInt8	8-bit integer values (unsigned) Allocates 1 byte
Int16	16-bit integer values Allocates 2 bytes
UInt16	16-bit integer values (unsigned) Allocates 2 bytes
Int32	32-bit integer values Allocates 4 bytes
UInt32	32-bit integer values (unsigned) Allocates 4 bytes
Single (Float32)	32-bit float values Allocates 4 bytes
Double (Float64)	64-bit float values Allocates 8 bytes

RTI/RTLib support

You can access the board's nonvolatile memory via RTI and RTLib.

For details, refer to:

- RTI: Nonvolatile Data Handling (NVDATA) (DS1007 RTI Reference 🕮)
  - NVDATA_READ_BLx
  - NVDATA_WRITE_BLx
- RTLib: Nonvolatile Data Handling (NVDATA) (DS1007 RTLib Reference 🕮)

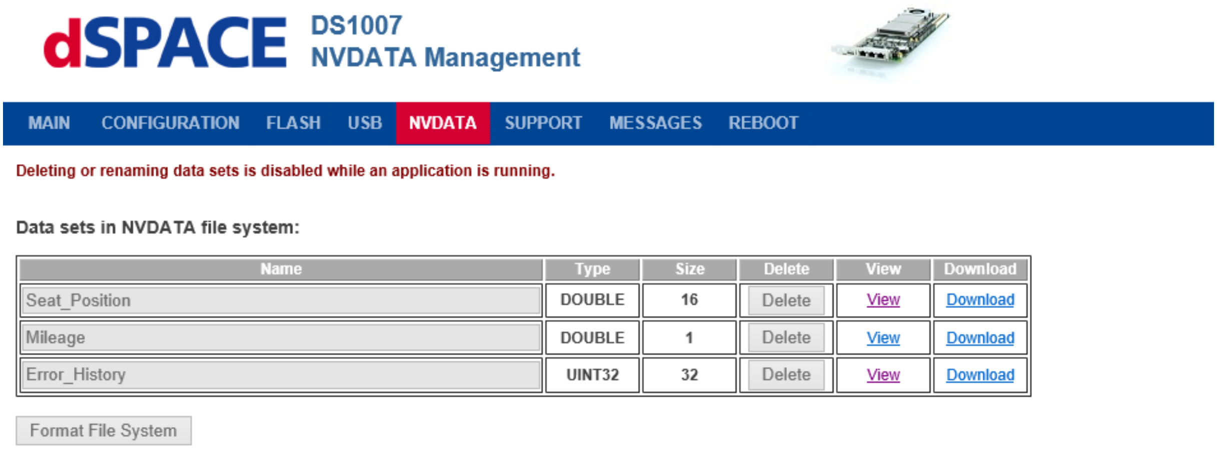
Using the Web Interface for Nonvolatile Data Handling

Introduction

The web interface of your DS1007 provides a configuration page that lets you manage the data sets stored in the board's nonvolatile memory. You can view, export, rename, or delete the data sets. You can also format the entire file system to restore its initial state.

Data set overview

If you click NVDATA in the main menu, the NVDATA Management page opens and the currently stored data sets are displayed with their names, data types, and the number of contained elements. You can delete, view the details of and download each data set to your computer via the relevant commands.



Deleting or renaming data sets is disabled while an application is running.

Data sets in NVDATA file system:

Name	Type	Size	Delete	View	Download
Seat_Position	DOUBLE	16	Delete	View	Download
Mileage	DOUBLE	1	Delete	View	Download
Error_History	UINT32	32	Delete	View	Download

Note

- If a real-time application is currently running on the board, deleting a data set, renaming a data set, and formatting the file system is not possible. Before you can execute one of these functions, the real-time application must be stopped and unloaded.
- The current view on the board's nonvolatile memory is only a snapshot. If a real-time application is currently running, the data might be modified while you are looking at it. Refresh the page to update its contents.

Renaming data sets

You can edit the data set name by clicking the displayed name. To save the modified name, press Enter.

The supported characters for a data set name are [A-Z], [a-z], [0-9], and the underscore '_'. Whitespaces or special characters are not supported.

Deleting data sets

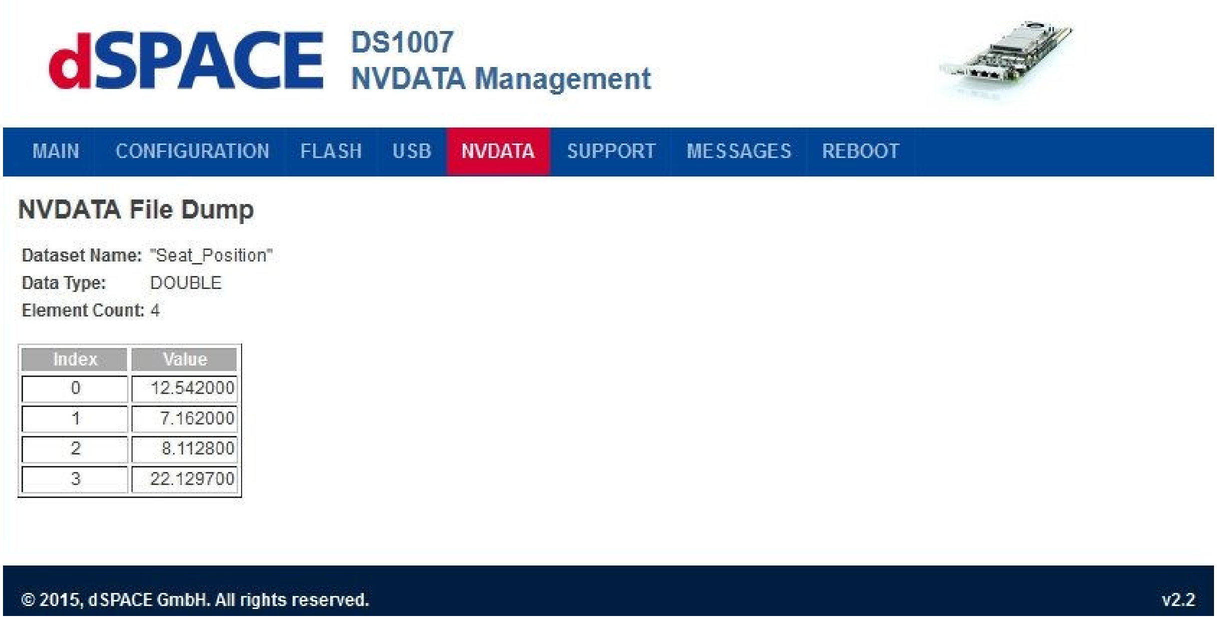
You can delete a data set by clicking its related Delete button.

The data set is deleted without further confirmation. It is not possible to recover the data.

Viewing data set details

You can view the elements contained in a data set by clicking its related View link. The contents of the data set are then displayed.

When you view a data set while an application is running, a temporary snapshot of the data set is displayed.



NVDATA File Dump

Dataset Name: "Seat_Position"
Data Type: DOUBLE
Element Count: 4

Index	Value
0	12.542000
1	7.162000
2	8.112800
3	22.129700

© 2015, dSPACE GmbH. All rights reserved. v2.2

Exporting data sets

You can export a data set by using its related Download link. Right-click the link to open the context menu and select the **`Save target as`** command.

The data set is downloaded and saved as a text file in the comma-separated CSV format.

Note

Floating-point values are represented with the decimal point. If you want to import the CSV file into a spreadsheet application, another character might be expected, such as the decimal comma. You can replace the characters using any standard text editor.

Formatting the NVDATA file system

You can format the board's nonvolatile memory to its initial state by clicking the Format File System button. All data sets are deleted.

USB Flight Recorder

Purpose

With the USB Flight Recorder, you can perform long-term data acquisition. The values of selectable variables are written to the connected USB mass storage device during simulation.

Where to go from here

Information in this section

Information in other sections

Running an Application from a USB Mass Storage Device.....53

RTI USB Flight Recorder Blockset Reference
Provides concise information on the RTI USB Flight Recorder Blockset.

USB Flight Recorder RTLib Reference
Provides detailed descriptions of the C functions needed to program RTI-specific Simulink S-functions or implement your real-time models manually via C programs (handcoding).

Basics on USB Flight Recorder

General information

The USB Flight Recorder is used to store time histories of real-time variables. The values of real-time variables are written to an externally connected USB mass storage device during real-time simulation.

A maximum of 250 different real-time variables can be recorded.

The recorded data is written to a file in the root directory of the USB mass storage device. The file name is automatically generated and contains the name of the real-time application, the creation date and the creation time.

On multicore platforms such as the DS1007 PPC Processor Board, the USB Flight Recorder is separately configured for each real-time application. A separate sequence of output files will be generated for each application.

The default maximum size of a single file is 32 MB. With RTLib, you can specify a maximum file size of up to 256 MB. If more than the specified maximum file size is recorded during one simulation, the data is split into several files. Consecutive files are created until the storage capacity of the USB device has been reached.

Then the captured data is discarded or older files are overwritten, according to your setting (refer to Memory overwrite mode on page 25).

After the simulation has finished, the recorded data can be read out by the host PC, refer to Handling the Data of the USB Flight Recorder on page 28.

Before you power down the board, you have to stop flight recording. For information how to terminate a flight recorder session, refer to Avoiding data loss on page 25.

Memory overwrite mode

You can use RTI or RTLib functions to define how to handle data when the USB mass storage device for flight recording is full:

Discard new data (blocked mode) When the USB mass storage device for flight recording is full, no further data is recorded.

The flight recording session is stopped, but the real-time application continues to run.

Replace old data (overwrite mode) When the USB mass storage device for flight recording is full, the oldest files are replaced.

Note

On multicore platforms such as the DS1007 PPC Processor Board, all real-time applications must use the same memory overwrite mode.

Requirements on the USB mass storage device

Any standard USB 2.0 mass storage device can be used, such as a USB memory stick or an external USB hard drive with or without separate power supply. The USB device must be formatted with the Microsoft FAT32 file system and must be directly connected to your hardware.

Note

A connection via a USB hub is not supported.

USB mass storage devices differ according to their rates for writing data. It is recommended to use a fast device for good performance.

The maximum supported file system size is 32 GB. Using a file system with a size greater than 32 GB might work but is neither recommended nor supported.

If you use an external USB hard drive with more than one partition, the flight recorder data is stored only in the first partition.

Avoiding data loss

The Windows FAT32 file system is not designed to operate in a fail-safe manner. For example, removing a USB memory stick while data is written to it can result in corrupted data.

Note

Risk of data loss

While a USB Flight Recorder session is active:

- Do not unplug the USB device from your hardware.
- Do not switch off your hardware.

You can recognize an active USB Flight Recorder session by the green flashing USB status LED.

To safely remove the USB device while an application is running, follow the instructions in this section.

To avoid partial data loss or corruption of the recorded data, you have to take the following precautions.

Removing the USB device while an application is running To safely remove the USB device while an application is running, apply one of the following methods:

- Stop the real-time application.

  The USB device can be safely removed as soon as the real-time application is stopped, for example, by using ControlDesk.
- Eject by button.

  The board bracket of the DS1007 PPC Processor Board provides an USB eject connector that you can use for unmounting the USB device. The unmount procedure is started when you press the button of the USB eject cable. You can remove the USB device when the USB status LED is off. A host message is also generated.

  For further information on the USB eject connector, refer to Board Overview (PHS Bus System Hardware Reference 🕮).
- Eject by user-defined signal.

  You can use the USB_FLIGHT_REC_EJECT block in your Simulink model or the **`dsflrec_usb_eject`** command in your handcoded application to unmount the USB device as a reaction to a signal, for example, a specific model variable.

To restart the USB Flight Recorder, you have to remove the USB device and reconnect it to your hardware.

Accessing USB Flight Recorder files via FTP

You can use any standard FTP client to retrieve USB Flight Recorder files without disconnecting the USB device from your hardware. The USB Flight Recorder files are stored in `<FTP_Root>\usb`. When the real-time application is not running, reading USB Flight Recorder files via FTP is safe, otherwise the following limitations apply:

- An FTP connection generates a CPU load that might result in partial data loss if the USB flight recording load is high.
- If a file is read via FTP while a flight recording session is running, incomplete data will be retrieved if the flight recorder data is written in overwrite mode.
- Any data capture session running via Ethernet might be disturbed by an FTP connection because the network bandwidth is shared.

Note

Do not download flight recorder data while the real-time application is running.

USB status LED

The status LED of the USB connector displays the current status of the USB device and the flight recorder.

LED Status	Meaning
Off	No USB device is connected.
Green	USB device is connected and flight recorder is not running.
Green blinking	USB device is connected and flight recorder is running.
Orange	USB device is full and the active flight recorder is specified not to overwrite old files.
Red	Write error when accessing the USB device, for example, if the device was removed while the flight recorder was running.

Time base

In the flight recorder, data captures are stored together with time stamps. Time stamps are measured in seconds relative to the time base 01/01/1970. Time stamps are interpreted appropriately by MATLAB or dSPACE experiment software. You can change the time base using M-program code. For an example, refer to MAT File Format for the USB Flight Recorder (RTI USB Flight Recorder Blockset Reference 🕮).

Each entry is stored together with a time stamp indicating an absolute date and time value with a resolution of 10.24 µs.

Startup behavior and maximum data rate

The maximum data rate per application depends on the real-time platform, the USB mass storage device and the number of running applications.

Using DS1007 PPC Processor Board A maximum data rate of 8 MB/s is possible without data loss. If separate real-time applications are running on the multicore board, the maximum data rate is totally 8 MB/s.

Limitations using the USB Flight Recorder

Stopping the simulation Do not stop the simulation during recording, for example, by switching the simulation state from *Run* to *Stop*, and then to *Run* again. If the data is not continuously recorded, time-stamping might be corrupted.

Using a USB mass storage device There are some limitations when working with a USB mass storage device:

- It is recommended to use a separate USB mass storage device for flight recording. Other files in the root folder of the device will be deleted by the USB Flight Recorder.

- Do not use a USB hub. The device must be directly connected to your hardware.
- If you remove the USB device while data is written, data loss might not be the only problem. In rare cases, the USB driver of the board fails to detect a reconnected USB device. To solve the problem, you have to restart the board.

RTI/RTLib support

Using RTI You can use the RTI blocks from the RTI USB Flight Recorder blockset to write flight recorder data to the USB mass storage device, refer to Components of the RTI USB Flight Recorder Blockset (RTI USB Flight Recorder Blockset Reference 📖).

Using RTLib You can use the `dsflrec_usb` RTLib functions to write flight recorder data to the USB mass storage device, refer to USB Flight Recorder (USB Flight Recorder RTLib Reference 📖).

Handling the Data of the USB Flight Recorder

Introduction

The data recorded by the USB Flight Recorder can be handled via ControlDesk.

Loading data to the host PC

After the simulation has finished, the recorded data can be downloaded to the host PC.

If the USB device is connected to your hardware, you can use ControlDesk and its specific functions for USB Flight Recorder handling to access the recorded data. You can select several binary files to download and convert them to CSV or MAT file format and to delete the binary files.

For further information, refer to How to Upload Flight Recorder Data Written to a USB Mass Storage Device (ControlDesk Measurement and Recording 📖).

Alternatively, you can use an FTP client or the File Explorer to download data from the USB mass storage device. Use `ftp://<IP_Address>` to connect to your real-time hardware.

Note

Do not delete any files on the USB mass storage device while a flight recorder session is still running.
For further information, refer to *Accessing USB Flight Recorder files via FTP* in Basics on USB Flight Recorder on page 24.

If the USB device is directly connected to your PC, you can use ControlDesk's functions to load and convert a single binary file. You can also use a standard file manager, for example, the File Explorer, to copy the recorded binary files to a local drive or to delete them from the USB device.

Note

See the section *Avoiding data loss* in Basics on USB Flight Recorder on page 24 for information on how to safely remove the USB device from your hardware.

For the handling of a great amount of binary files on the USB mass storage device, or if there is no ControlDesk installed on the PC used for postprocessing the flight recorder data, you can use a command line tool for merging, extracting and converting several binary files, refer to Merging, Extracting and Converting BIN Files of a Flight Recorder (ControlDesk Measurement and Recording 📖).

Related topics

Basics

Basics on USB Flight Recorder..24

Timers and Time-Stamp Counters

Timers

The following timers and time-stamp counters are available with the DS1007:

- Timer A and Timer D, which you can use for periodic timer events
- Timer B, which you can use for periodic or asynchronous events
- Time-Stamp Counter, which you can use for time interval measurement and time-stamping in single-processor systems
- Synchronous Time Base Unit for time-stamping in multiprocessor systems

The clocks of the time-stamp counter and the timers A, B, and D derive from the bus clock (BCLK) of 100 MHz that equates a timer period of 10 ns.

Each processor core of the multicore DS1007 board provides its own timers.

Where to go from here

Information in this section

Timer A and Timer D

Basics

Timer A and Timer D are used in the same way. They are usually used for periodic timer events such as the timer-driven tasks of an application. The timers are 32-bit down counters that generate an interrupt whenever they reach zero. After generating an interrupt, the timers are reloaded automatically.

Timer A

Timer A is driven by BCLK/2, which equates to a timer period of 20 ns.

Timer D

Timer D is driven by BCLK/2, which equates to a timer period of 20 ns.

Timers in multiprocessor system

In a multiprocessor system, the interrupts of Timer A can be forwarded to other processor cores of the DS1007. The interrupts of Timer D cannot be forwarded.

Related topics

Basics

PHS Bus Interface.. 15

Timer B

Basics

Timer B can be used for periodic or asynchronous events. It is a 32-bit up counter with a prescaler and programmable compare value. Timer B generates an interrupt when it reaches the compare value. After generating an interrupt, the counter continues counting. To generate the next interrupt, the compare value has to be set to the next desired time. The prescaler is programmable in power-of-two steps (4 … 512) so that Timer B can be driven by BCLK/4 … BCLK/512 (40 ns … 5120 ns). If you use the interrupts via RTI blocks, the prescaler is always set to the highest resolution (BCLK/4).

Timer B in multiprocessor system

In a multiprocessor system, the interrupts of Timer B can be forwarded to other processor cores of a DS1007.

Related topics

Basics

PHS Bus Interface.. 15

Time-Stamp Counter

Basics

The Time-Stamp Counter is a 64-bit counter based on the CPU-internal time-base register. The counter is driven by a 25 MHz clock and has therefore a resolution of 40 ns.

Each CPU core has its own 64-bit time-base register. Both registers are synchronized.

You can either read the entire 64 bits of the register or the lower 32 bits only.

Time interval measurement

The Time-Stamp Counter can be used for determining absolute points in time as well as for measuring intervals. For details, refer to Time Interval Measurement (DS1007 RTLib Reference 🕮).

Time stamping

In single-processor systems, the Time-Stamp Counter provides the time base for time-stamping. Time-stamping supplements data points. This means that the plots are not distorted even if data points are sampled at irregular intervals, for example, when asynchronous tasks are simulated.

Tip

You can always use the RTLib's Time Stamping module to read the current system time. The Time Stamping module will automatically access the correct time base. For details on the Time Stamping module, refer to Time-Stamping (DS1007 RTLib Reference 📖).

In multiprocessor/multicore systems, time-stamping is used to produce a global time base for all the connected processor cores of a DS1007. For more information, refer to Synchronization on page 46.

Interrupt Controller

Introduction

The interrupt controller handles the DS1007's various interrupts (level- or edge-triggered), for example, timer, PHS bus, and I/O error interrupt. Each processor core has its own interrupt controller. The interrupts can be masked. You can globally enable and disable interrupt generation. The interrupts are prioritized. The I/O error line and the PHS-bus interrupt lines are filtered by a digital noise filter, which suppresses short spikes.

Where to go from here

Information in this section

Interrupt Handling

Basics

With RTI you can easily implement interrupt-driven subsystems by means of specific interrupt blocks provided in the RTI library. You can use these blocks to receive interrupts from I/O boards.

If you create a handcoded model, you can use RTLib functions to handle interrupts.

Related topics

Basics

Handling Tasks (RTI and RTI-MP Implementation Guide 📖)

References

Interrupt Handling (DS1007 RTLib Reference 📖)
Subinterrupt Handling (DS1007 RTLib Reference 📖)

Available Interrupts

List of available interrupts

The following table lists the interrupts that are available on the DS1007. The predefined symbols are to be used when you handcode an application with RTLib.

Predefined Symbol	Meaning
SRTK_INT_PHS_0	PHS-bus interrupt line 0
...	...
SRTK_INT_PHS_7	PHS-bus interrupt line 7
SRTK_INT_TIMER_A	Timer A interrupt
SRTK_INT_TIMER_B	Timer B interrupt
SRTK_INT_TIMER_D	Timer D interrupt
SRTK_INT_SERIAL_UART	Serial UART interrupt
SRTK_INT_MACROTICK	Macrotick interrupt
SRTK_INT_FWD_TIMER_A	Forwarded Timer A interrupt
SRTK_INT_FWD_TIMER_B	Forwarded Timer B interrupt
SRTK_INT_GL_0	Gigalink 0 interrupt
...	...
SRTK_INT_GL_3	Gigalink 3 interrupt
SRTK_INT_IO_ETH	I/O Ethernet interrupt

DS1007 Multiprocessor Systems

Introduction

If the computing power of a single core is not sufficient for simulating rather large real-time models in real time, you can use the RTI-MP feature to separate the application into smaller subapplications which you can distribute to the available computation nodes. With the DS1007, a multiprocessor application can be distributed on multiple processor cores of multiple DS1007 boards. The entire application is automatically synchronized via Gigalink connections which can either be established physically via Gigalink modules between different boards or virtually between different cores of one board.

DS1007 in MP systems

You can run up to ten DS1007 boards as a multiprocessor system providing twenty cores (limited by the current device driver).

Where to go from here

Information in this section

Information in other sections

Basics of Handling Platforms (ControlDesk Platform Management 🕮)

Distributing the Model for MP Systems (RTI and RTI-MP Implementation Guide 🕮)

DS1007 With Dual-Core Processor

Introduction

Because the DS1007 board has a dual-core processor, you can use it as a multiprocessor system.

Where to go from here

Information in this section

Using DS1007 as Multicore Platform

Introduction

To give you important information on using a dual-core DS1007 board.

Using DS1007 boards with dual-core processor

- The following RTLib functions have a different behavior when DS1007 modular systems are used as multicore platform:
  - **`dsgl_mp_init`**

    The internal Gigalinks are connected virtually and automatically configured according to the specified topology.
  - **`dsgl_module_present`**

    This function always returns 1, because the internal virtual Gigalinks are always present.
- An I/O board is assigned to the DS1007 board as a whole, but you can access it by one processor core only at the same time.

Network topology for dual-core systems

If your DS1007 does not provide a DS911 Gigalink Module, you can anyway use a multiprocessor application, if it consists of only two subapplications. Each core provides four internal virtual Gigalinks, which can handle the IPC blocks of the RTI-MP Blockset.

If you load the real-time application (RTA file) to the platform, its application processes are automatically assigned to the board's cores. The internal network topology will be dynamically generated.

Using DS1007 as Multiprocessor Platform

Introduction

To give you important information on using a DS1007 as a multiprocessor platform.

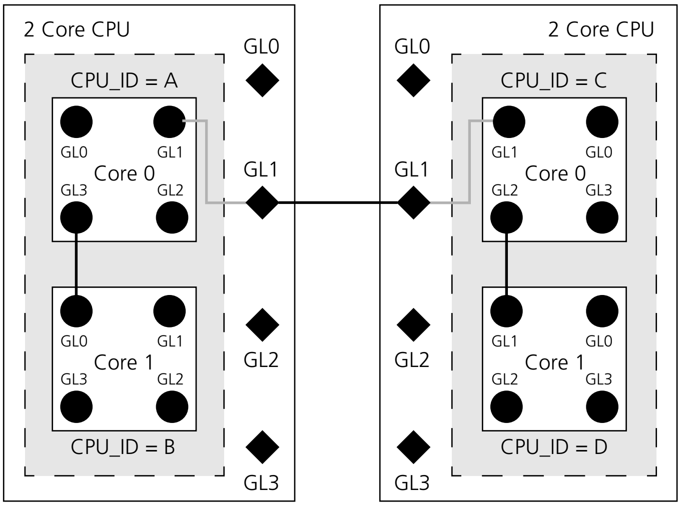
Using DS1007 boards with the optional DS911 Gigalink Module

If your DS1007 board is equipped with a DS911 Gigalink Module, you can connect your board with other PHS-bus-based systems or SCALEXIO. The DS1005, DS1006, and DS1007 processor boards you want to connect to must also be equipped with Gigalink Modules.

Network topology for multiprocessor systems

The network topology that is realized by connecting the Gigalinks of your processor boards is also relevant for a multicore processor board. The topology can consist of internal and external Gigalinks. Both Gigalink types are handled in a similar way via the IPC blocks of the RTI-MP Blockset.

The illustration shows an example of combining internal and external Gigalinks.



The network topology of this example can be represented in a mapping table.

CPU_ID	GL0	GL1	GL2	GL3
A	–	C: GL1	–	B: GL0
B	A: GL3	–	–	–
C	–	A: GL1	D: GL1	–
D	–	C: GL2	–	–

If you use RTLib, you specify the network topology as an array structure.

```
{
  { {CPU_NONE, GL_NONE}, {CPU_C, GL_1}, {CPU_NONE, GL_None}, {CPU_B, GL_0} }
  { {CPU_A, GL_3}, {CPU_NONE, GL_NONE}, {CPU_NONE, GL_None}, {CPU_NONE, GL_None} }
  { {CPU_NONE, GL_NONE}, {CPU_A, GL_1}, {CPU_D, GL_1}, {CPU_NONE, GL_NONE} }
  { {CPU_NONE, GL_NONE}, {CPU_C, GL_2}, {CPU_NONE, GL_NONE}, {CPU_NONE, GL_NONE} }
}
```

Related topics

Basics

DS911 Gigalink Module.. 40

DS911 Gigalink Module

Introduction

The DS1007 board which you want to connect to the multiprocessor system must be equipped with a Gigalink module.

Where to go from here

Information in this section

Basics of DS911 Gigalink Modules

Introduction

The optional DS911 Gigalink Module is the interface for connecting several DS1007 boards to a multiprocessor system. It provides high-speed serial data transmission via optical fiber.

Note

If you order a multiprocessor system, the DS1007 boards are already equipped with the DS911 Gigalink Modules. If you want to make your single-processor DS1007 multiprocessing-capable, inquire at dSPACE for further details.

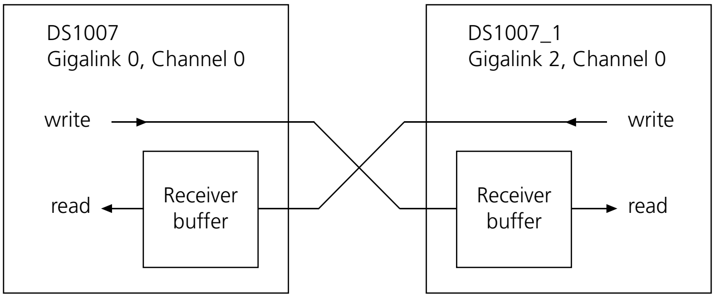
Characteristics

The DS911 Gigalink Module has the following characteristics:

- Four optical high-speed communication ports (Gigalinks), each providing:
  - 16 bidirectional interrupt lines (including the PHS bus' IOERROR line); one interrupt line is used internally by the RTLib.

- 16 bidirectional data channels (8 channels in the virtual shared memory mode, 8 channels in the swinging buffer mode; width of each channel: 8 kByte.). Refer to Virtual Shared Memory Mode on page 43 and Swinging Buffer Mode on page 43.

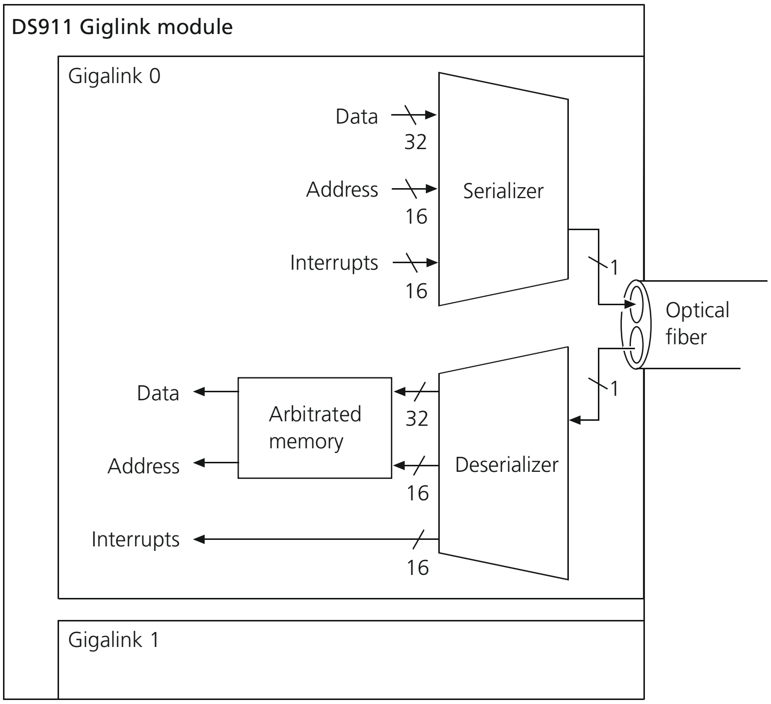
  Three interrupt lines and one shared memory channel are used by the simulation environment.
- I/O access speed of send operations adapted to the Gigalinks' capabilities: data is sent directly without a FIFO buffer, see the following illustration.



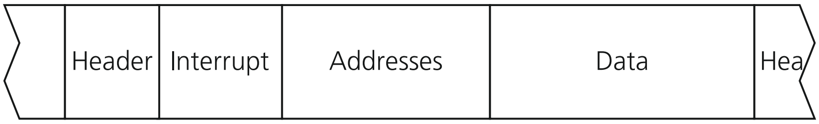
- Total transfer rate: 1.25 Gbit/s. See below for an example of how much time certain Gigalink operations might take.
- 32-bit and 64-bit read/write operations allowed
- Possible distance: up to 100 m

Functional units

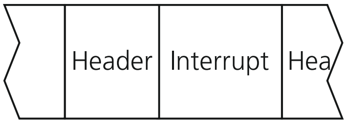
Each Gigalink consists of a full-duplex bidirectional interface. Before transmission, the sender serializes the number, data, and addresses of the pending interrupt with the highest priority. The receiver stores this data in an arbitrated memory. Incoming interrupt requests are handed over to the interrupt controller. The following illustration shows the functional units of the Gigalinks:



Since there is only one optical fiber per Gigalink and direction, the number, addresses and data of the highest-priority pending interrupt. This is done via the protocol shown in the following illustration. A single frame of this protocol can hold one interrupt request and the address and data of one data word.



If only interrupts need to be transmitted, the Gigalink module uses a simplified protocol as shown below:



Data transmission

The DS911 Gigalink Module provides two operation modes for data transmission:

- Channels 0 … 7 operate in the swinging buffer mode. Refer to Swinging Buffer Mode on page 43.
- Channels 8 … 15 operate in the virtual shared memory mode (channel 8 is reserved). Refer to Virtual Shared Memory Mode on page 43.

With RTI-MP, you simply have to select the desired protocol in the Communication Channel Setup dialog (refer to Interprocessor Communication Using IPC Blocks (RTI and RTI-MP Implementation Guide 📖)). If you use RTLib, you can use identical read and write operations because both modes are implemented in hardware and their operation is fully transparent to the application (refer to Gigalink Communication (DS1007 RTLib Reference 🕮) for details).

Virtual Shared Memory Mode

Introduction

This mode is based on a single receiver buffer for each channel, which the sender and the receiver access simultaneously. The receiver buffer is arbitrated between the sender and the receiver, and can be accessed with 32-bit and 64-bit operations.

> **Note**
>
> If you require block consistency (meaning that all the elements of a vector/array are consistent), you have to use the swinging buffer mode.

Writing data The sender writes its data to the buffer regardless of whether the receiver is currently reading the buffer or not.

Reading data The receiver reads the content of the buffer regardless of whether the sender is currently writing the buffer.

Related topics

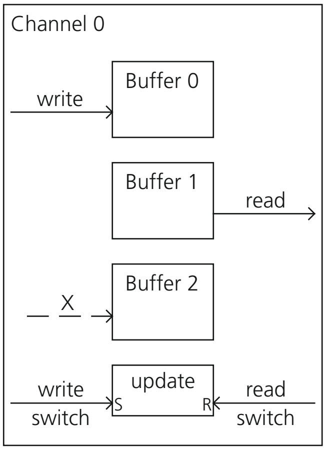
Basics

Virtual Shared Memory Protocol (RTI and RTI-MP Implementation Guide 🕮)

Swinging Buffer Mode

Introduction

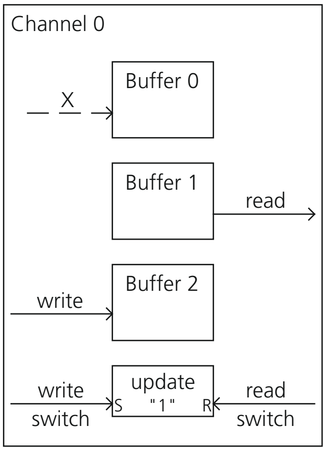
This mode is based on three receiver buffers for each channel, which all appear at the same memory location so that the sender and the receiver see only one buffer at a time. Consider the following illustration:



Description

There are three pointers, each pointing to one buffer. The write pointer marks the current write buffer, the read pointer marks the current read buffer, and the X pointer marks the buffer that is currently neither a write nor a read buffer. The sender and the receiver access these buffers according to the following rules:

Writing data After the sender has written data to the current write buffer, it sends a write buffer switch command. On receipt of this signal, the write pointer and the X pointer are swapped, and the update flag ("new data"). The X buffer therefore now contains a complete block of consistent data. The illustration below shows the resulting configuration:

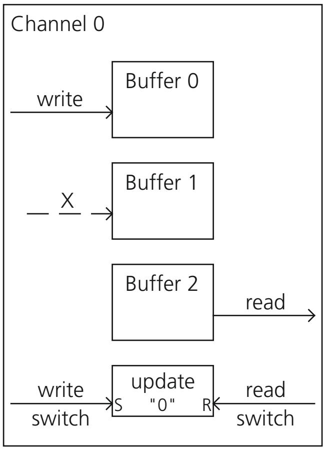


Reading data Before the receiver reads data, it sends a read switch signal. On receipt of this signal, the update flag is evaluated:

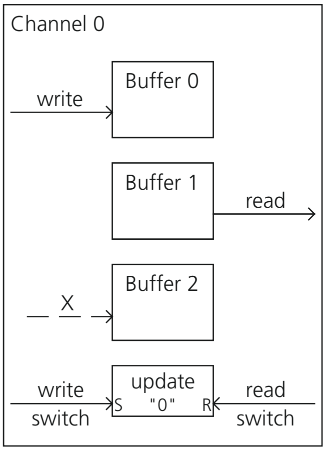
- If it is set (the X pointer marks new data), the read pointer and the X pointer are swapped.
- If it is not set (the X pointer marks old data), the read pointer remains in its old position.

Then the update flag is reset and the receiver gets the data from the current read buffer. Starting from the first configuration, one of the following two configurations can result next.

- If the update flag was "1" ("new data") beforehand, the following configuration results:



- If the update flag was "0" ("no new data") beforehand, the following configuration results:



This behavior corresponds to RTI-MP's unsynchronized swinging buffer protocol. As an alternative, the software can make the receiver wait for new data, which corresponds to RTI-MP's synchronized swinging buffer protocol.

Related topics

Basics

Swinging Buffer Protocol (RTI and RTI-MP Implementation Guide 🕮)

Synchronization

Introduction

When running a real-time application on a multiprocessor system it is important to have an absolute time on all CPUs and to trigger the timer-driven tasks of all CPUs by means of the same timer interrupt. These synchronization aspects are described in the following sections.

Synchronization Aspects

Synchronized timer tasks

Each core of the DS1007 has its own timers. Timer-driven tasks running on the different cores should be executed synchronously. To achieve this, the timers are automatically synchronized.

Global time base in an MP system

The global time base in a multiprocessor system is set by the Synchronous Time Base Unit (STBU). It consists of two 32-bit counters and is driven by BCLK/2.

Due to manufacturing tolerances, which lead to clock drifts, the local clocks in a multiprocessor system have to be synchronized periodically. To keep the communication effort low, synchronization does not take place at every tick of the local clocks (microtick), but at a selected tick of a timing master. This selected tick is called macrotick. Upon the occurrence of a macrotick, the number of microticks is set to zero, and the number of macroticks is increased by 1 at each processor. Starting from this point in time, the absolute time is calculated by multiplying the macrotick and microtick period with its counter values and adding the products. The absolute time is used for time-stamping. At the recommended STBU synchronization period (macrotick period) of 10 ms, the absolute time has a maximal deviation of ±0.5 µs.

Tip

You can always use the RTLib's Time Stamping module to read the current system time, regardless of whether you run a single-processor or a multiprocessor system. The Time Stamping module will automatically access the correct time base. For details on the Time Stamping module, refer to Time-Stamping (DS1007 RTLib Reference 🕮).

Tracing signals in an MP system

The global time base is also used for time-stamping that improves the tracing of signals in a multiprocessor or multicore system.

Master-slave tracing In the master-slave tracing method, data is copied from all processor cores to a master processor core, which writes it in the master's trace buffer. The copy operation causes a time overhead.

Distributed tracing The distributed tracing method traces the signals locally at each processor core. The host PC accesses them separately and puts the plots into relation based on their time stamps. This means that ControlDesk can display the plot as if it originated from one processor core. In a multiprocessor system, only the distributed tracing method is used.

Related topics

References

Global Sample Rate Timer in an MP System (DS1007 RTLib Reference 📖)

DS1007 Application Start

Introduction

After power-up, the DS1007 boots automatically and executes the firmware located in the onboard flash memory.

Where to go from here

Information in this section

Firmware

Introduction

After power-up and reset, the DS1007 boots from the on-board flash memory, which holds the preinstalled firmware.

Characteristics of the firmware

The firmware carries out the following steps:

- It determines the I/O boards connected to the DS1007's PHS bus.
- It determines whether a user-specific application exists in the on-board flash memory.

  If the on-board flash memory contains an application, it is started automatically.
- If the on-board flash memory does not contain an application, the firmware determines whether an application is available in the autostart folder of the connected USB mass storage device. If so, the application is started automatically.

Note

Use the latest firmware

A real-time application that you built with the latest dSPACE software version usually requires the latest firmware version on your platform for execution.

The firmware is available on the dSPACE Release DVD. You should regularly check the availability of new firmware at http://www.dspace.com/go/firmware. To update the firmware, use the dSPACE Firmware Manager, refer to Firmware Manager Manual 🕮.

Running the board in secured mode

If the DS1007 cannot correctly boot its firmware, it automatically starts in *secured mode*. In secured mode, the user can restore the firmware by using the dSPACE Firmware Manager. Loading and running real-time applications is disabled in secured mode.

If you have problems with the board that might be caused by corrupted firmware, but the board does not automatically boot in secured mode, you can force secured mode.

- The board's web interface provides a function to reboot the board in secured mode.
- Change the on-board switch to run the board in secured mode.

For further information, refer to How to Start the DS1007 to Secured Mode (DS1007 Hardware Installation and Configuration Guide 🕮).

⚠ CAUTION

Risk of injury and/or material damage. Updating the firmware can cause uncontrolled movements of connected devices.

Disconnect actuators and sensors from the hardware, before you start the DS1007 in secured mode, or before you start a firmware update.

Related topics

Basics

Firmware Manager Manual

HowTos

How to Start the DS1007 to Secured Mode (DS1007 Hardware Installation and Configuration Guide 🕮)

References

General Settings Properties (ControlDesk Platform Management 🕮)

Running an Application from Local Memory

Introduction

Applications in the local memory are cleared if you turn off the hardware. You have to download an application from the host PC to the local memory of the board before you can run it.

Download time

Download times can differ considerably. It takes longer, for example, if a formerly loaded application cannot be unloaded due to errors in the application.

Running an application

After download, you have to explicitly start the application, for example via the **Platforms/Devices** controlbar in ControlDesk. To start the application automatically after download, use the `Real-Time Application / Offline Simulation Application - Load and Start` command.

Reloading an application

Reloading an application is only necessary if you want to overwrite the currently loaded application. If you stop an application, it is still loaded and can be restarted by using the **Start** command, for example, in ControlDesk.

For details of handling real-time applications, refer to Handling Real-Time Applications with ControlDesk (DS100x, DS110x, MicroAutoBox II, MicroLabBox – Software Getting Started 📖).

Related topics

References

Real-Time Application / Offline Simulation Application - Load (ControlDesk Platform Management 📖)
Real-Time Application / Offline Simulation Application - Load and Start (ControlDesk Platform Management 📖)
Reload (ControlDesk Platform Management 📖)
Reload and Start (ControlDesk Platform Management 📖)
Start (ControlDesk Platform Management 📖)
Stop (ControlDesk Platform Management 📖)

Running an Application from Flash Memory

Introduction

If you want an application start automatically after power-up, you have to load it to the board's flash memory.

You can also start an application from a connected USB mass storage device, refer to Running an Application from a USB Mass Storage Device on page 53.

Basics

The DS1007 provides an onboard flash memory. You can use 32 MB of the flash memory for real-time applications and custom libraries. To load an application to the flash memory, you can use the **Load to Flash** command, for example, in ControlDesk. After an application has been written to the flash memory, it is also copied to the local memory. To start the application, use the **Start** command or the **Load to Flash and Start** command.

Upon every power-up of the DS1007, the firmware copies the application from the flash memory to the local memory and starts it, regardless of whether the board is connected to the host PC or not.

If the DS1007 is not connected to the host PC, you can stop and restart the application by resetting the DS1007.

If the DS1007 is connected to the host PC, you can use the dSPACE experiment software, for example, ControlDesk, to stop and restart the application without reloading it.

You can clear an application from the flash memory via the **Clear Flash** command, for example, in ControlDesk. Further functions for managing the flash memory are available in the board's web interface. For further information, refer to Basics on the Web Interface (DS1007 Hardware Installation and Configuration Guide 🕮).

Note

When you switch off the DS1007, the contents of the local memory are lost. When you switch on the power again, the contents of the flash memory are copied to the local memory and the application starts.

For information on the flash memory’s characteristics, refer to Memory on page 14.

For information on handling real-time applications using ControlDesk, refer to Handling Real-Time Applications with ControlDesk (DS100x, DS110x, MicroAutoBox II, MicroLabBox – Software Getting Started 🕮).

Note

Some platform management activities require exclusive access to the platform. For further information, refer to Synchronized Platform Management with Several dSPACE Products (ControlDesk Platform Management 🕮).

Related topics

References

Clear Flash (ControlDesk Platform Management 🕮)
Real-Time Application - Load to Flash (DS1007/MicroLabBox/MicroAutoBox III/SCALEXIO) (ControlDesk Platform Management 🕮)
Real-Time Application - Load to Flash and Start (DS1007/MicroLabBox/MicroAutoBox III/SCALEXIO) (ControlDesk Platform Management 🕮)

Running an Application from a USB Mass Storage Device

Introduction

If you want an application to start automatically after power-up, you can use a connected USB mass storage device for it.

Basics

If no application is stored in the board's flash memory, the boot firmware can load an application from a connected USB mass storage device to the local memory of the board and start the application.

Requirements on the USB mass storage device Any standard USB 2.0 or 3.0 mass storage device can be used at the USB 2.0 port, such as a USB memory stick or an external USB hard drive with or without separate power supply. The USB device must be formatted with the Microsoft FAT32 file system and must be connected directly to your hardware.

Note

A connection via a USB hub is not supported.

USB mass storage devices differ according to their rates for writing data. It is recommended to use a fast device for good performance.

The maximum supported file system size is 32 GB. Using a file system with a size greater than 32 GB might work but is neither recommended nor supported.

If you also want to use flight recording via a USB device, you have to use the same USB mass storage device from which you loaded the application.

Preparing the USB mass storage device Before you can run an application from a USB mass storage device, you have to prepare the device by using your host PC.

To run an application from a USB mass storage device, you have to create the following two folders on the device:

- `autostart`

  You have to copy the application that you want to start after power-up to this folder.

  > **Note**
  >
  > - Only one application file is allowed in this folder. If multiple applications are stored in this folder, the board will not load any of them.
  > - If an application is stored in the board's flash memory, the application on the USB device will be ignored. You have to clear the flash memory so you can use the USB device for loading an application.

- `applications`

  You can copy the applications that you want to start manually to this folder.

  If the USB mass storage device is connected to the board, you can create an image of the currently running application. The image file will then be stored in this folder as well.

After you prepared the USB mass storage device, you can plug the USB device into your board. Then, you can use the board's web interface to manage the applications.

Flight recording data will be stored directly in the root folder of your USB mass storage device.

Managing applications on the USB device Enter the board's IP address in any web browser to open the board's web interface, which provides the USB Management page. You can use the commands on this page to manage the applications on the USB device without disconnecting it from the board.

For general information on the web interface, refer to Basics on the Web Interface (DS1007 Hardware Installation and Configuration Guide 🕮).

The USB Management page provides the following commands:

- Create Application Image

  With this command, you can create an application image from the currently running application and store it to the `applications` folder. The file name consists of the application name, the current date and time and the suffix *img*.

  In contrast to applications that were generated by using MATLAB/Simulink (with the suffix *rta*), an image file does not contain any dynamic state information that might have existed at the time the image file was captured.

- Manage USB Applications

  You can use this command to display an overview of the application files stored in the `autostart` and `applications` folders on the USB device. Each entry provides the **`Delete`** command to delete the selected application from the USB device and the **`Start`** command to start it manually.

Note

- Files are deleted immediately, without confirmation.
- Before you can load and start an application, any running application must be unloaded from the RAM. You can use the **`Unload Application`** command for this. You can start applications only in the running state, not in the stopped state.

You can use the **`Copy to Autostart`** command to copy the selected application file to the `autostart` folder. Any existing application files in the `autostart` folder are renamed, so that they will no longer be recognized as executable applications. The copied application can be started manually by using the **`Start`** command or after a reboot of the board.

- List USB Applications

  You can use this command to display an overview of the application files stored on the USB device.
- Unload Application

  You can use this command to stop and unload the running application.
- Delete USB Autostart

  You can use this command to delete any files stored in the `autostart` folder of the USB device. This is primarily required to delete the copies of application files that were created when copying an application file from the `applications` folder to the `autostart` folder.

Note

The executable application will also be deleted.

Related topics

References

Clear Flash (ControlDesk Platform Management 🕮)

I/O Board Overview

Introduction

To meet the various demands in rapid control prototyping and hardware-in-the-loop simulation, dSPACE offers a wide range of I/O boards for any purpose. You can find your I/O board via board name or I/O functions.

Where to go from here

Information in this section

Information in other sections

Defining I/O Interfaces (RTI and RTI-MP Implementation Guide 🕮)

How to Add I/O Blocks to Models (RTI and RTI-MP Implementation Guide 🕮)

I/O Board Overview by Board Name

Introduction

In the following, the various I/O boards that can be connected to the dSPACE processor boards are listed in alphabetical order.

DS2001 High-Speed A/D Board

The DS2001 High-Speed A/D Board features 5 parallel A/D converter channels.

DS2002 Multi-Channel A/D Board

The DS2002 Multi-Channel A/D Board is equipped with two multiplexed A/D converters offering 32 parallel A/D converter channels.

DS2003 Multi-Channel A/D Board

The DS2003 Multi-Channel A/D Board is equipped with two multiplexed A/D converters offering 32 parallel A/D converter channels.

DS2004 High-Speed A/D Board

The DS2004 High-Speed A/D Board features 16 parallel A/D converter channels for differential input signals.

DS2101 D/A Board

The DS2101 D/A Board features 5 parallel D/A channels with 12-bit resolution.

DS2102 High-Resolution D/A Board

The DS2102 High-Resolution D/A Board features 6 parallel D/A channels with 16-bit resolution.

DS2103 Multi-Channel D/A Board

The DS2103 Multi-Channel D/A Board features 32 parallel D/A channels with 14-bit resolution.

DS2201 Multi-I/O Board

The DS2201 Multi-I/O Board is equipped with I/O units for ADC, DAC, digital I/O, and PWM generation.

DS2202 HIL I/O Board

The DS2202 HIL I/O Board lets you simulate and measure automotive signals. The board contains signal conditioning for typical signal levels of 12 V and 42 V automotive systems.

DS2210 HIL I/O Board

The DS2210 HIL I/O Board is tailored to simulate and measure automotive signals. It combines a variety of typical HIL I/O functions on one board. The board also contains signal conditioning for typical signal levels of 12 V automotive systems.

DS2211 HIL I/O Board

The DS2211 HIL I/O Board is tailored to simulate and measure automotive signals. It combines a variety of typical HIL I/O functions on one board. The board also contains signal conditioning for typical signal levels of 12 V and 42 V automotive systems.

DS2301 Direct Digital Synthesis Board

The DS2301 Direct Digital Synthesis (DDS) Board is equipped with 6 DSPs and designed for fast and flexible waveform generation. It computes each signal sample just-in-time and outputs it immediately. The DS2301 C functions allow to exchange data between your main application and an application running on one of the DSPs. The communication is established via the dual-ported memories of the DDS board.

DS2302 Direct Digital Synthesis Board

The DS2302 Direct Digital Synthesis (DDS) Board is equipped with six DSPs and designed for fast and flexible waveform generation. It computes each signal sample just-in-time and outputs it immediately. The DS2302 C functions allow to exchange data between your main application and an application running on one of the six DSPs. The communication is established via the dual-ported memories of the DDS board.

DS2401 Resistive Sensor Simulation Board

The DS2401 Resistive Sensor Simulation Board features 4 resistor output channels.

DS3001 Incremental Encoder Interface Board

The DS3001 Incremental Encoder Interface Board features 5 independent incremental encoder interface channels and captures digital position signals.

DS3002 Incremental Encoder Interface Board

The DS3002 Incremental Encoder Interface Board features 6 independent incremental encoder interface channels and captures digital position signals and sinusoidal position signals.

DS4001 Timing and Digital I/O Board

The DS4001 Timing and Digital I/O Board provides 32 bidirectional TTL digital I/O lines.

DS4002 Timing and Digital I/O Board

The DS4002 Timing and Digital I/O Board provides 24 bidirectional plus four input and four output TTL digital I/O lines. It can also be used for the analysis and the generation of square waves or pulse-width modulated signals.

DS4003 Digital I/O Board

The DS4003 Timing and Digital I/O Board provides 96 bidirectional TTL digital I/O lines, divided into three 32-bit ports.

DS4004 HIL Digital I/O Board

The DS4004 HIL Digital I/O Board provides 96 bidirectional digital I/O lines with signal conditioning, divided into three 32-bit ports.

DS4121 ECU Interface Board

The DS4121 ECU Interface Board lets you establish communication between an electronic control unit (ECU) and a dSPACE real-time system in combination with a custom-designed plug-on device (POD) that adapts the ECU signals to the DS4121 interface.

DS4201 Prototyping Board

The DS4201 Prototyping Board is designed to integrate customized user hardware. It is the interface between custom I/O devices that can be mounted directly on the board and the dSPACE PHS bus, thus enabling the connection to the dSPACE system.

Due to the DS4201's capability to offer an access for a wide range of custom I/O devices to the dSPACE system, the support for the board provided by the Real-Time Interface (RTI) is restricted to some basic features.

DS4201-S Serial Interface Board

The DS4201-S Serial Interface Board provides 4 serial communication channels with selectable line transceivers (RS232, RS422 or RS485).

DS4302 CAN Interface Board

The DS4302 CAN Interface Board allows data transfer between dSPACE real-time systems and various other control units via 4 CAN bus interfaces.

DS4330 LIN Interface Board

The DS4330 LIN Interface Board allows data transfer between dSPACE real-time systems and various other control units via 16 LIN bus interfaces.

DS4501 IP Carrier Board

The DS4501 IP Carrier Board allows to use IP modules in a PHS-bus-based system.

DS4505 Interface Board

The DS4505 Interface Board together with different interface modules allows data transfer between dSPACE real-time systems and various other control units:

- Via FlexRay bus by means of DS4340 FlexRay Interface Modules
- Via CAN bus by means of DS4342 CAN FD Interface Modules

DS5001 Digital Waveform Capture Board

The DS5001 Digital Waveform Capture Board is used for the analysis of input signals such as square waves, pulse trains, or pulse-width modulated signals. These waveforms can be seen as a series of rising and falling edges and the corresponding times (timestamps).

16 input channels allow to capture digital pulses with a resolution of 25 ns. Up to 512 events per input channel (edge direction and timestamp) are stored for further calculations.

DS5101 Digital Waveform Output Board

The DS5101 Digital Waveform Output Board autonomously generates various TTL pulse patterns on up to 16 channels with a time resolution of 25 ns. You can also program your own pulse patterns in an intuitive high-level language.

DS5202 FPGA Base Board

The DS5202 FPGA Base Board provides a field programmable gate array (FPGA) and connectors for a customization module (piggy-back module) for implementing customer-specific I/O adaptations. Algorithms for real-time applications with high sampling rates can be shifted to the FPGA.

DS5203 FPGA Board

The DS5203 FPGA Board provides a Xilinx© FPGA to implement custom FPGA applications. With the DS5203M1 Multi-I/O Module (piggy-back module) you can extend its I/O capability. Algorithms for real-time applications with high sampling rates can be shifted to the FPGA.

I/O Board Overview by I/O Function

Introduction

In the following, the various I/O boards that can be connected to the dSPACE processor boards are listed according to their specific I/O capabilities.

A/D conversion

The following dSPACE boards can be controlled by the dSPACE processor boards to perform A/D conversion:

- DS2001 High-Speed A/D Board
- DS2002 Multi-Channel A/D Board
- DS2003 Multi-Channel A/D Board
- DS2004 High-Speed A/D Board
- DS2201 Multi-I/O Board

D/A conversion

The following dSPACE boards can be controlled by the dSPACE processor boards to perform D/A conversion:

- DS2101 D/A Board
- DS2102 High-Resolution D/A Board
- DS2103 Multi-Channel D/A Board
- DS2201 Multi-I/O Board

Automotive signal generation and measurement

The following dSPACE boards can be controlled by the dSPACE processor boards to generate and measure automotive signals:

- DS2202 HIL I/O Board
- DS2210 HIL I/O Board
- DS2211 HIL I/O Board
- DS2302 Direct Digital Synthesis Board

Digital I/O

The following dSPACE boards can be controlled by the dSPACE processor boards to perform bit I/O:

- DS2201 Multi-I/O Board
- DS2301 Direct Digital Synthesis Board
- DS2302 Direct Digital Synthesis Board
- DS4001 Timing and Digital I/O Board
- DS4002 Timing and Digital I/O Board
- DS4003 Digital I/O Board
- DS4004 HIL Digital I/O Board

Timing I/O

The following dSPACE boards can be controlled by the dSPACE processor boards to perform timing I/O, such as the generation of various pulse patterns including PWM or the capture of digital frequency signals:

- DS2201 Multi-I/O Board
- DS2301 Direct Digital Synthesis Board
- DS2302 Direct Digital Synthesis Board
- DS4001 Timing and Digital I/O Board
- DS4002 Timing and Digital I/O Board
- DS4004 HIL Digital I/O Board
- DS5001 Digital Waveform Capture Board
- DS5101 Digital Waveform Output Board

Interface boards

The following dSPACE boards can be controlled by the dSPACE processor boards to integrate more specialized custom devices I/O into the dSPACE real-time system:

- DS3001 Incremental Encoder Interface Board
- DS3002 Incremental Encoder Interface Board
- DS4201 Prototyping Board
- DS4201-S Serial Interface Board
- DS4302 CAN Interface Board
- DS4330 LIN Interface Board
- DS4505 Interface Board with DS4340 FlexRay Interface Modules or DS4342 CAN FD Interface Modules

Special I/O

The following dSPACE boards can be controlled by the dSPACE processor boards to perform more specialized I/O:

- DS2210 HIL I/O Board
- DS2211 HIL I/O Board
- DS2301 Direct Digital Synthesis Board
- DS2302 Direct Digital Synthesis Board
- DS2401 Resistive Sensor Simulation Board

Integrating FPGA applications

The following dSPACE boards can be controlled by the dSPACE processor boards to integrate FPGA applications:

- DS5202 FPGA Base Board
- DS5203 FPGA Board

ECU interfacing

The following dSPACE boards have been designed to provide the link between the customer's prototype or a production-type electronic control unit (ECU), and a dSPACE modular system:

- DS4120 ECU Interface Board
- DS4121 ECU Interface Board

Related topics

Basics

I/O Board Overview by Board Name... 57