DS1006 Processor Board

Features

Release 2021-A – May 2021

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

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

Contents

About This Document

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 DS1006

Where to go from here

Information in this section

Block Diagram 7
The block diagram gives an overview of the functional units of the DS1006.

Feature Overview 9
Gives an overview of the features of the DS1006 Processor Board.

Information in other sections

DS1006 Data Sheet (up to Revision DS1006-03) (PHS Bus System Hardware Reference 🕮)
Summarizes the technical specifications of the DS1006.

DS1006 Data Sheet (as of Revision DS1006-06) (PHS Bus System Hardware Reference 🕮)
Summarizes the technical specifications of the DS1006 multicore processor board.

Block Diagram

Introduction

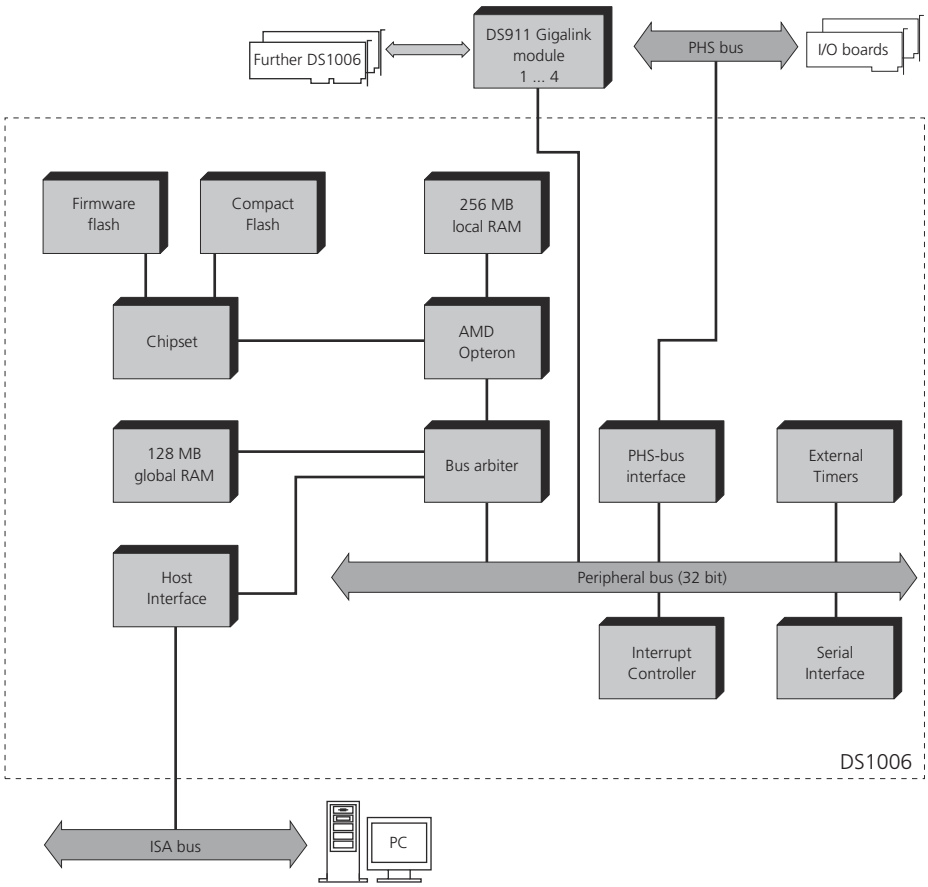
The DS1006 Processor Board is based on a single-core or a multicore AMD Opteron™ processor. This real-time processor (RTP) is the main processing unit. It can access modular I/O boards via its PHS bus. It is multiprocessing-capable via the (optional) DS911 Gigalink Module.

The block diagrams give an overview of the functional units of the DS1006.

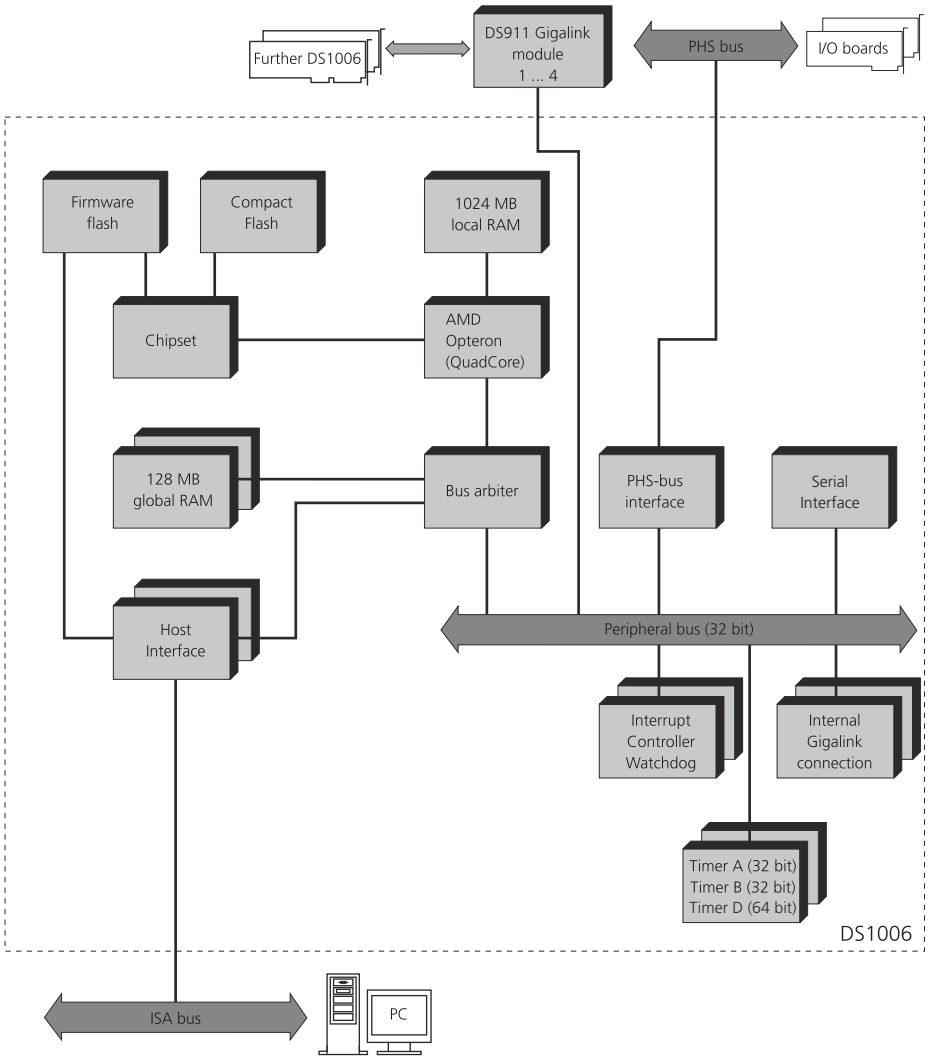
Block diagrams

The following illustrations give overviews of the functional units of the DS1006.

Board revisions DS1006-01 to DS1006-03 provide a single-core processor:



As of board revision DS1006-06, a multicore processor is provided:



Feature Overview

Introduction

The DS1006 Processor Board provides several features.

Systems

The DS1006 Processor Board can be used in two kind of systems.

Single-processor system Means that your real-time application is running on one single-core DS1006 board or one processor core of a multicore DS1006 board.

Multiprocessor systems Means that your real-time application is running on two or more single-core DS1006 boards, on two or more processor cores of a multicore DS1006 board, on two or more multicore DS1006 boards, or a combination of all these.

Features

The DS1006 provides the following features:

Processor Representing the computing power of the board. Refer to Processor on page 12.

Memory Comprising RAM, cache, and flash. Refer to Memory on page 13.

Timers and time base counters Timers and time base counters comprising:

- Timer A and Timer D: Sample rate timer with interrupt function
- Timer B: Interval timer with interrupt function
- Real-Time Processor Built-In Time-Stamp Counter: Time base for single-processor systems
- Synchronous Time Base Unit: Time base for multiprocessor systems

Refer to Timers and Time-Stamp Counters on page 14.

Interrupt control Providing various hardware interrupts. Refer to Interrupt Controller on page 18.

PHS bus Allowing access to up to 16 I/O boards. Refer to PHS-Bus Interface on page 21.

Gigalink module (optional) For setting up multiprocessor systems. Refer to DS911 Gigalink Module on page 36.

Synchronization Comprising:

- Timer A for multiprocessor systems. Refer to Synchronized Timer Tasks on page 43.
- Global Time Base in an MP System on page 44
- Tracing Signals in an MP System on page 45

Monitoring Allowing monitoring of program execution and host access:

- Watchdog to monitor program execution. Refer to Watchdog on page 27.
- Force reset to monitor host access. Refer to Force Reset on page 28.

Serial interface For setting up a user-specific communication or for debugging, etc. Refer to Serial Interface on page 24.

Host interface For setting up the DS1006, downloading programs and transferring run-time data to/from the host PC. Refer to Host Interface on page 25.

Features of the DS1006

Introduction

See the following topics for detailed information on the features of the DS1006 Processor Board.

Where to go from here

Information in this section

Processor and Memory

Where to go from here

Information in this section

Processor

Introduction

The DS1006 uses a single-core or multicore AMD Opteron™ processor as the real-time processor (RTP). The RTP calculates your real-time models and accesses the I/O boards via the PHS bus.

Characteristics

The processor has the following characteristics:

- Cache sizes:

Buffer	Single-core Processor	Multicore Processor
L1 data cache	64 KB	64 KB
L1 instruction cache	64 KB	64 KB
L2 cache	1024 KB	512 KB per core
L3 cache	-	6 MB

- Temperature sensor to prevent the processor from overheating:
  - A software mechanism terminates the real-time application and enters low power mode.
  - At an even higher temperature, a processor-internal mechanism (called AMD thermtrip) shuts down the CPU clock.
- HyperTransport™ technology connection for high bandwidth and low latency access to I/O devices
- Built-in memory controller for high-speed memory access

For detailed information on the AMD Opteron™, refer to the AMD Web site at http://www.amd.com.

Note

The processor specifications may be modified. dSPACE may change the processor used to a more powerful AMD Opteron™ processor without notice. This change will not affect the board revision number. If the processor is changed, dSPACE can upgrade older DS1006 Processor Board and their firmware. Inquire at dSPACE for further details.
For details on the available processors, refer to Data Sheets (PHS Bus System Hardware Reference 🕮).

Memory

Introduction

The single-core and multicore DS1006 boards have different memory features.

Characteristics

The following table shows the characteristics of the memory of a single-core processor and multicore processor.

Memory	Single-Core Processor	Multicore Processor
L2 cache (on-chip of the processor)	1024 KB	512 KB per core
L3 cache	–	6 MB
Flash memory for boot firmware (on-chip)	2 MB	2 MB
Global memory for host data exchange	128 MB SDRAM	128 MB DDR2-267 per core
Local memory for the application and dynamic application data	256 MB DDR-400	1 GB DDR2-800
Slot for a CompactFlash card	1	1

CompactFlash cards are necessary for autobooting applications. For more information on the usable CompactFlash cards, refer to Data Sheets (PHS Bus System Hardware Reference 🕮).

Timers and Time-Stamp Counters

Where to go from here

Information in this section

Overview of Timers and Time-Stamp Counters

Timers

The following timers and time-stamp counters are available with the DS1006:

- 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 Timers A, B, and D derive from the bus clock (BCLK). The clock of Time-Stamp Counter derives from the CPU clock (CPUCLK).

Each processor core of the multicore DS1006 board provides its own timers.

Tip

In the dSPACE experiment software, you can find information on the bus clock and the CPU clock in the **Properties** dialog of the DS1006 used.

Related topics

References

DS1006 Data Sheet (as of Revision DS1006-06) (PHS Bus System Hardware Reference 🕮)
DS1006 Data Sheet (up to Revision DS1006-03) (PHS Bus System Hardware Reference 🕮)
Timer Interrupt Block (RTI and RTI-MP Implementation Reference 🕮)

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 and Timer D are driven by BCLK/2.

Timers in multiprocessor system

In a multiprocessor system, the interrupts of Timer A can be forwarded to other processor cores of a multicore board and other DS1006 boards via Gigalink connection. The interrupts of Timer D cannot be forwarded.

Related topics

Basics

PHS-Bus Interface.....21

References

Timer A (DS1006 RTLib Reference 🕮)
Timer D (DS1006 RTLib Reference 🕮)

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/512 … BCLK/4. 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 multicore board and other DS1006 boards via Gigalink connection.

Related topics

Basics

PHS-Bus Interface 21

References

Timer B (DS1006 RTLib Reference 🕮)

Time-Stamp Counter

Introduction

The Time-Stamp Counter is a 64-bit up counter with a fast access time and is driven by the CPU clock (CPUCLK). It is used for time interval measurement and time-stamping. If the CPUCLK is 2.2 GHz, the counter has a hardware resolution of approximately 0.5 ns. However this resolution cannot be achieved because of limitations in the software. The effective resolution typically achieved by the measurement functions is about 20 ns.

The Time-Stamp Counter is used for time interval measurement and time-stamping.

Time interval measurement

The Time-Stamp Counter can be used for determining absolute points in time as well as for measuring intervals. Refer to Time Interval Measurement (DS1006 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.

In multiprocessor systems, time-stamping is used to produce a global time base for all the connected processor cores of a multicore board and all the connected processor boards. Since the Time-Stamp Counter is not suitable for multiprocessor systems, the DS1006 is equipped with the Synchronous Time Base Unit, which provides the time base for a multiprocessor system. Refer to Global Time Base in an MP System on page 44.

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. Refer to Time-Stamping (DS1006 RTLib Reference 🕮).

Interrupt Controller

Where to go from here

Information in this section

Information in other sections

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

Subinterrupt Handling (DS1006 RTLib Reference 🕮)

Basics of the Interrupt Controller

Introduction

The interrupt controller handles the DS1006's various interrupts (level or edge triggered).

Interrupt controller

The interrupt controller handles the DS1006's various interrupts (level-triggered or edge-triggered), for example, timer, PHS bus, Gigalink (for multiprocessor systems), Watchdog, host, and I/O error interrupt. Each processor core of a multicore DS1006 board has its own interrupt controller. The interrupts can be masked. A global interrupt enable/disable is also available. 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.

Interrupt Handling

Introduction

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

Related topics

Basics

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

References

Interrupt Handling (DS1006 RTLib Reference 🕮)
Subinterrupt Handling (DS1006 RTLib Reference 🕮)

Available Interrupts

Interrupts

The following table lists the interrupts that are available for handcoded models on the DS1006. If you create a handcoded model, you can use RTLib functions to handle these interrupts. They are sorted in descending priority (a smaller vector corresponds to a higher priority):

Vector	Description
0	PHSBUS I/O error
1	Watchdog
2	PHSBUS interrupt line 0
3	PHSBUS interrupt line 1
4	PHSBUS interrupt line 2
5	PHSBUS interrupt line 3
6	PHSBUS interrupt line 4
7	PHSBUS interrupt line 5
8	PHSBUS interrupt line 6
9	PHSBUS interrupt line 7
10	Timer A
11	Timer B
12	Gigalink 0
13	Gigalink 1
14	Gigalink 2
15	Gigalink 3

Vector	Description
16	UART
17	Host interrupt
18	STBU synchronization (Macrotick)
20	Timer A (forwarded by Gigalink)
21	Timer B (forwarded by Gigalink)
23	Timer D

Related topics

Basics

Basics of the Interrupt Controller..... 18

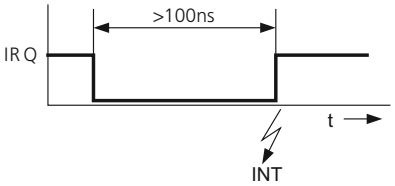
External Interrupt

Introduction

To generate an interrupt request (IRQ) via an I/O board.

External interrupt

To generate an interrupt request (IRQ) via a PHS-bus I/O board, the input signal to this I/O board should be low level for at least 100 ns. The following illustration shows the required signal shape (active edge triggered):



Related topics

Basics

Basics of the Interrupt Controller..... 18

Interfaces

Introduction

The DS1006 has interfaces for data communication.

Where to go from here

Information in this section

PHS-Bus Interface

PHS bus

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.

Using a multicore processor board, the PHS bus is assigned to the board and not to the processor cores. The available bandwidth must be therefore divided by the number of cores used.

IOERROR line

In addition to the standard control lines for reading and writing, the PHS bus provides the IOERROR control line. 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 can either hold its output voltage or set it to zero. The DS1006 can set or read this line.

Note

Whenever the DS1006 is reset, it always activates the IOERROR line.

Characteristics

The PHS bus interface provides:

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

Partitioning the PHS bus

Usually the processor board and the I/O boards of a PHS-bus-based system are installed in a single expansion box. With the DS802 PHS Link Board, you can spatially partition the PHS bus by arranging the I/O boards in several expansion boxes.

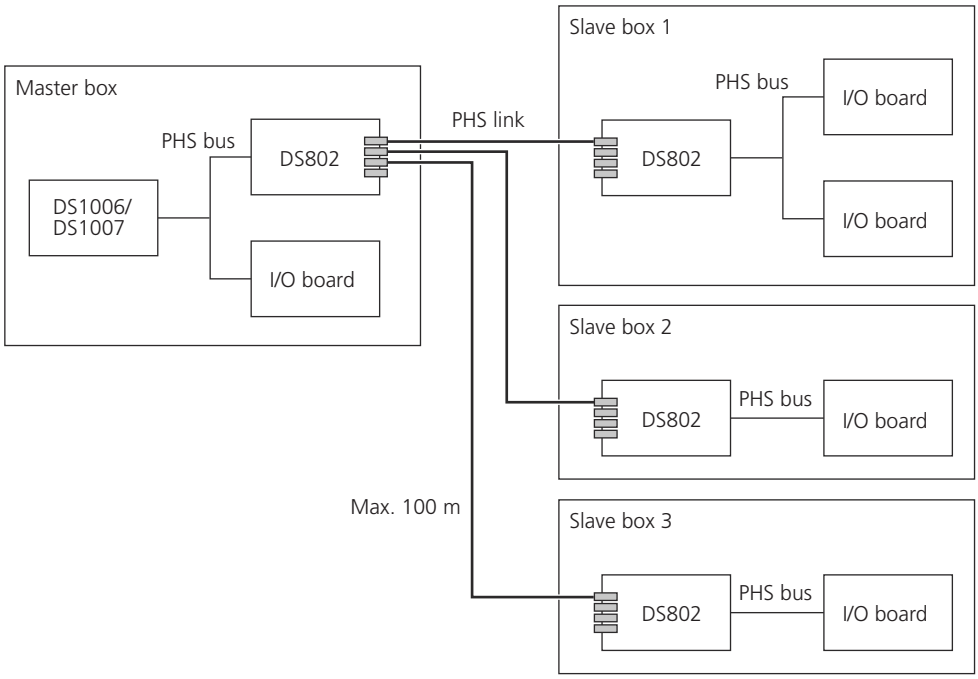
Use scenarios You can use the DS802, for example, in the following scenarios:

- According to your development stage, you can easily extend the PHS-bus-based system with new I/O boards by connecting a separate preconfigured expansion box to the existing expansion box.
- Components of the PHS-bus-based system (installed in a separate expansion box) can easily be replaced and reused, for example, in other projects, without additional configuration work.
- If the devices of a test bench are spread out over a large area, you can shorten the cabling between dSPACE hardware and external devices (sensors/actuators) by installing I/O boards in a separate expansion box which is near the external devices.
- In a few cases the components of your PHS-bus-based system might require more power than the maximum that a single expansion box supplies. In this case, you can distribute the I/O boards on several boxes via DS802. This avoids power limitations in your system caused by single expansion boxes.

Note

The DS802 does not increase the number of usable I/O boards in a PHS-bus-based system. Note that the PHS bus provides 16 PHS-bus base addresses. Each I/O board in a PHS-bus-based system and each DS802 installed in a master box require a unique PHS-bus address. So, up to 15 I/O boards can be used in a PHS-bus-based system with one DS802.

System overview The following illustration shows a typical system architecture of a partitioned PHS-bus-based system.



The box which contains the processor board is called the master box. Boxes which contain only I/O boards are called slave boxes. One DS802 must be installed in the master box and one in every slave box.

The DS802 boards provide the necessary link between the boxes via fiber-optic cable (up to 100 m). As shown above, the link must be a direct connection between the communication ports of a master box and a slave box. It does not matter which ports you use to connect a master box to a slave box. All ports provide the same functionality. DS802 boards installed in a slave box can use only one communication port. You cannot connect one slave box to several master boxes at the same time.

You can use the DS802 in multiprocessor systems, for example, where several processor boards are connected via Gigalink modules.

Behavior of real-time application The logical behavior of the real-time application (for example, building or executing) when used with a partitioned PHS-bus-based system is the same as when used with a non-partitioned PHS-bus-based system.

There is one difference. Partitioning results in additional latencies. This means that the execution times for C functions of I/O boards installed in a slave box increase with the number of PHS-bus accesses.

The execution times increase by a factor in a range of approx. 3 … 7. You can use this range, to estimate the function execution time for an I/O board installed in a slave box.

Tip

The function execution times for boards used in a non-partitioned PHS-bus-based system are documented in the *RTLib References* of the respective I/O board. The function execution times for I/O boards installed in a slave box are not documented. Use the above mentioned factor to get comparable values.

Compatibility with processor boards The DS802 can be used in PHS-bus-based systems. Newer versions of the processor boards are fully compatible with the DS802. There are older versions which are compatible after an update done by dSPACE.

For a complete list of compatible processor board versions, refer to DS802 Data Sheet (PHS Bus System Hardware Reference 🕮).

Supported I/O boards The DS802 can be used in combination with many types of available dSPACE I/O boards. However, some I/O boards and some functionalities of specific I/O boards are not supported.

The I/O board support depends on the dSPACE software release which you use. For a list of supported I/O boards, refer to DS802 Data Sheet (PHS Bus System Hardware Reference 🕮).

Further information For examples on the usage of the DS802, and notes on installation and configuration, refer to Partitioning a PHS-Bus-Based System with the DS802 PHS Link Board (DS1006 Hardware Installation and Configuration Guide 🕮).

Related topics

References

PHS-Bus Handling (DS1006 RTLib Reference 🕮)

Serial Interface

Introduction

The DS1006 is equipped with a serial interface (UART) to communicate with standard RS232 devices. The UART is driven by a 14.7456 MHz oscillator, which allows transfer rates of up to 115.2 kbaud.

Pinouts

For information on the pinouts, refer to UART RS232 Connector (P6) (PHS Bus System Hardware Reference 🕮).

Related topics

References

Serial Interface (DS1006 RTI Reference 🕮)
Serial Interface Communication (Low Level) (DS1006 RTLib Reference 🕮)

Specifying the Baud Rate of the Serial Interface

Oscillator frequency

The serial interface of the processor board is driven by an oscillator with a frequency f_{osc} = 14.7456 MHz.

Available baud rates

You can specify the baud rate for serial communication with the processor board in the range 5 … 115,200 baud. Using RTI or RTLib, you can specify any baud rate in this range. However, the baud rate actually used by the processor board is a fraction of the oscillator frequency f_{osc}. The available baud rates can be calculated according to

$f = f_{osc} / (16 \cdot n)$,

where n is a positive integer within the range 1 … 65535.

When you specify a baud rate in RTI or RTLib, the closest available baud rate is used for serial communication. For example, if you specify 70,000 baud as the baud rate, the baud rate used is 70,892 baud.

Host Interface

Introduction

The host interface of the single-core DS1006 board consists of eight 16-bit I/O ports. It serves the setup of the DS1006, program downloads and run-time data transfers to/from the host PC. Only 16-bit I/O instructions are valid.

The host interface also features a supervision circuit. Refer to Force Reset on page 28.

Each processor core of the multicore DS1006 board provides its own host interface. The specified board address is therefore internally enlarged with the addresses of the processor cores.

For example, if you have set 0x300 as the board address for a quadcore DS1006, Core 0 is set to 0x300, Core 1 is set to 0x310, Core 2 is set to 0x320, and Core 3 is set to 0x330.

Memory and I/O Access

Big endian and little endian The endian modes define the byte sequence within data types containing several bytes. In big endian mode, the most significant byte of a multibyte data type is stored at the lowest memory address. In little endian mode, the most significant byte is stored at the highest memory address.

DS1006 and host PC The DS1006's RTP always works in little endian mode. As the host PC also works in little endian mode, changing the byte sequence is not necessary.

Note

You have to use the correct data width for access to the variables. For example, short integers have to be read with 16-bit accesses. Be careful when using block transfers to a data structure with mixed data types.

Monitoring

Introduction

The DS1006 provides two monitoring features.

Where to go from here

Information in this section

Watchdog

Introduction

The Watchdog can be used to monitor program execution. It consists of a timer that has to be strobed before the timer period expires. If strobed, the timer period is restarted. If the timer period expires, the Watchdog carries out one of the actions listed below (depending on the Watchdog's operation mode):

Normal mode The Watchdog generates an interrupt and stops the timer. When strobed, it restarts the timer.

Reset mode The Watchdog generates an interrupt, reloads, and restarts the timer. If this period expires, it resets the RTP. If strobed, however, it restarts the timer.

If the RTP is reset via the Watchdog, it reboots. For details on booting, refer to DS1006 Application Start on page 51.

If you use a multicore processor board each core has its own watchdog, but all the cores are connected to the same HRESET signal. If one core has started his watchdog in *Reset mode* and doesn't service it, all the cores will be reset.

Related topics

Basics

Force Reset 28

References

Watchdog Handling (DS1006 RTLib Reference 🕮)

Force Reset

Introduction

If the DS1006 is installed in an expansion box which is connected to the host PC via Ethernet, a slot CPU is used. In some very rare cases the DS1006 might block the slot CPU. This situation can result from a too-long PHS-bus WAIT assertion while the slot CPU tries to access the DS1006's memory. Due to the arbitrated memory, the slot CPU has to wait until the RTP finishes the I/O access. To prevent this situation, the DS1006 measures the duration of the host access and resets the RTP if the access exceeds 15.2 µs.

Related topics

Basics

Watchdog.....27

DS1006 Multiprocessor Systems

Introduction

A single DS1006 board might not offer sufficient computing power for simulating large applications in real time. You might therefore want to implement such a large real-time application on a multiprocessor (MP) system or multicore processor.

Where to go from here

Information in this section

Information in other sections

Setting Up a Multiprocessor System (DS1006 Hardware Installation and Configuration Guide 🕮)
A multiprocessor system is based on two or more processor boards.

Basics of Handling Platforms (ControlDesk Platform Management 🕮)

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

Basics of DS1006 Multiprocessor Systems and DS1006 With Multicore Processor

Where to go from here

Information in this section

Basics of DS1006 Multiprocessor Systems

DS1006 in MP systems

The DS1006 board is well equipped for setting up multiprocessor systems that provide massive computing power:

- You can run up to twenty DS1006 boards as a multiprocessor system (software limitation).
- Arbitrary net topologies are possible.
- All the DS1006 boards provide full trace support.
- Time stamping provides a global time base for all the DS1006 boards.

Related topics

Basics

Using DS1006 With Multicore Processor

Introduction

If a DS1006 board has a multicore processor, you can use it as a single-processing system or as multi-processing system.

Using DS1006 boards with different core types

- The multicore DS1006 board is only compatible with a single-core DS1006 board if the firmware version used is 1.3. In this case, you can run an application on a single-core or a multicore DS1006 board without having to recompile the real-time application. The application is then only running on one core of a multicore DS1006 board.

  With firmware version 2.1 and the related FPGA code for the multicore support (both released with DS1006MC Service Pack for dSPACE Release 6.5 and dSPACE Release 6.6), applications can be executed on each of the cores. These applications are no more compatible with applications generated with older dSPACE releases.

  Tip

  If you have a multicore DS1006 board that supports the single-core usage only, you must update the firmware and the FPGA code of the board to activate the full multicore support.

- Applications compiled with the RTLib1006 (version 2.2) can be executed on a single-core DS1006 when firmware version 2.1 is installed. The new firmware and RTLib is intended for both, multicore DS1006 boards (Rev. 6) and single-core DS1006 boards (Rev. 3).

  Note the following supported combinations for boot firmware version and RTLib version.

RTLib Version	Firmware 1.x	Firmware 2.x
RTLib 1.x	OK	Not supported. Recompile with new RTLib or downgrade the firmware.
RTLib 2.x	Not supported. Recompile with old RTLib or upgrade the firmware.	OK

- The following RTLib functions have a different behavior on multicore DS1006 systems:
  - `ds1006_mp_init`

    If you use a multicore DS1006 board, a DS911 Gigalink module is not required if the specified topology can be realized using the internal virtual Gigalinks.

The internal Gigalinks are connected virtually and automatically configured according to the specified topology.

- `ds1006_gl_module_present`

  If you are using a multicore DS1006 board, this function always returns 1, because the internal virtual Gigalinks are always present.

- `dsgl_ptr_get`

  This function was deprecated since dSPACE Release 6.4 and must not be used for virtual Gigalinks.

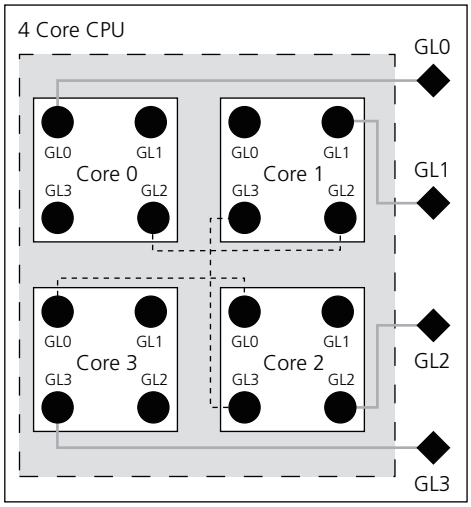
- If you have configured a multiprocessor system, you can now implement the computation nodes as single-core boards, one or more multicore boards, or a combination of the two.
- An I/O board that is installed in your system can be accessed by one processor core only. The available I/O boards in your system can only be assigned to the multicore board as a whole and not to one of its processor cores.
- The numbering of an I/O board depends on the number of processor boards used in the system and not the number of processor cores. For example, if you have two single-core boards in your system and two DS2001 boards, each processor board can access a *DS2001_1*. If the two single-core processor boards are replaced by two cores of a multicore processor board, the I/O boards are named *DS2001_1* and *DS2001_2*.

Network topology

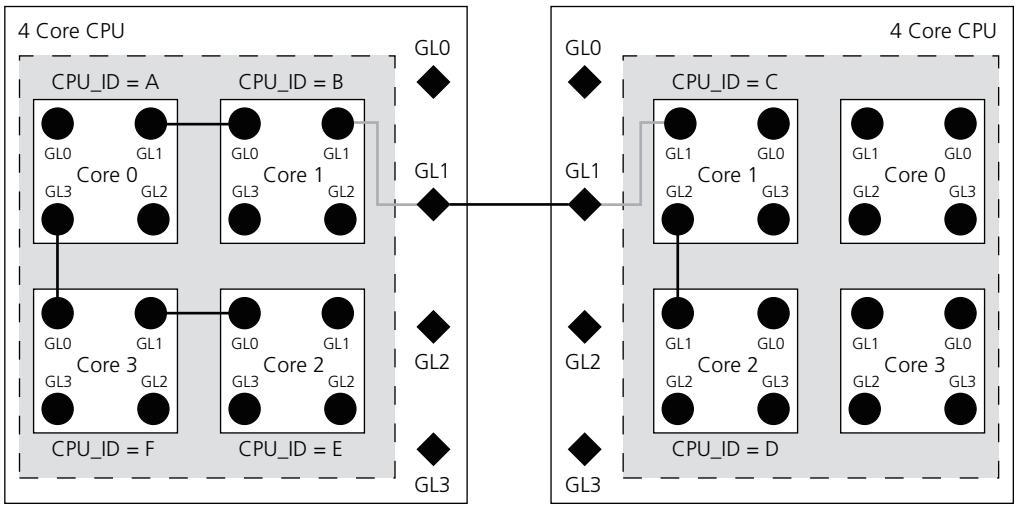
The network topology that is realized by connecting the Gigalinks of your processor boards is also relevant for a multicore processor board. The topology can consist of internal and external Gigalinks. Both Gigalink types are handled in a similar way via the IPC blocks of the RTI-MP Blockset. The mapping between physical and logical computation nodes is done during platform registration, for example, in ControlDesk.

If you use a multicore DS1006 board, a DS911 Gigalink module is not required if the specified topology can be realized using the internal virtual Gigalinks. The internal Gigalinks are automatically connected and configured according to the specified topology.

The illustration below shows the default topology for a quadcore processor board with no external connection as an example. The default topology is used unless another topology is specified.



The next illustration shows an example of combining internal and external Gigalinks. Six of the eight available cores are used in the MP system.



Characteristics of the internal virtual Gigalinks

The internal Gigalinks are using the local memory for data transfer.

The minimum transfer rate is about 900 MByte/s, if you transfer 8 KByte data blocks at the same time on 4 CPU cores.

If you use smaller data blocks or less CPU cores at the same time, the transfer rate is higher.

Reboot behavior

Because a single-core DS1006 board can execute only one application at a time, if a reboot is required (for example, after inserting a CompactFlash card), it is done automatically. With the multicore DS1006 board, applications can run in parallel on different processor cores. If a reboot is required, you must start it explicitly.

Related topics

Basics

DS911 Gigalink Module..... 36

References

ds1006_gl_module_present (DS1006 RTLib Reference 🕮)
ds1006_mp_init (DS1006 RTLib Reference 🕮)

DS911 Gigalink Module

Introduction

The DS1006 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 DS1006 boards to a multiprocessor system. It provides high-speed serial data transmission via optical fiber.

Note

If you order a multiprocessor system, the DS1006 boards are already equipped with the DS911 Gigalink Modules. If you want to make your single-processor DS1006 multiprocessing-capable, inquire at dSPACE for further details.

Characteristics

The DS911 Gigalink Module has the following characteristics:

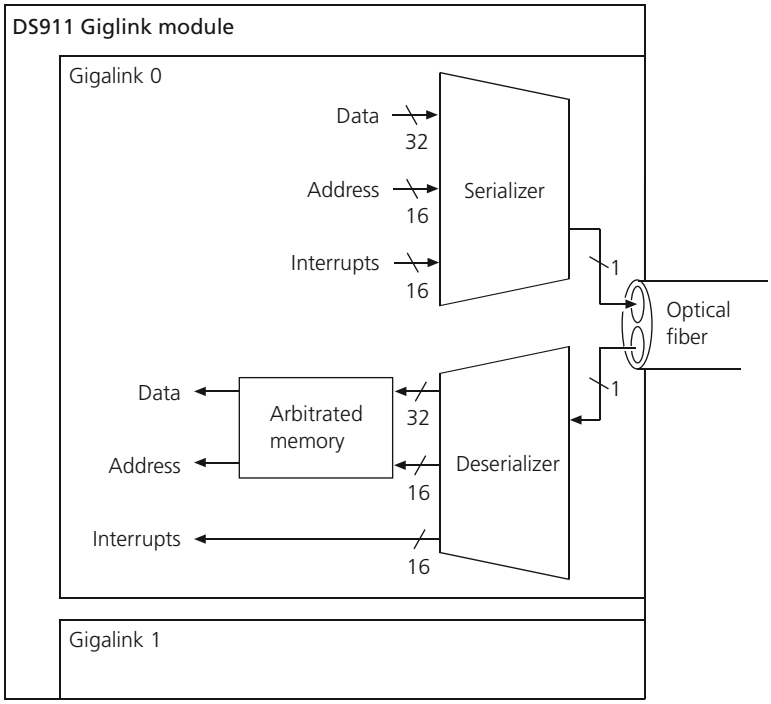
- Four optical high-speed communication ports (Gigalinks), each providing:
  - 16 bidirectional interrupt lines (including the PHS bus' IOERROR line); one interrupt line is used internally by the RTLib.

- 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 39 and Swinging Buffer Mode on page 40.

  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

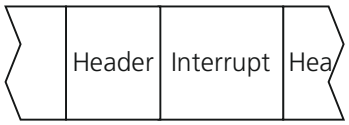
Functional units

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:



Gigalink access times

The maximum time (t_{max}) for data transfer from the main memory of the sender DS1006 to the main memory of the receiver DS1006 can be estimated via the formula

$$t_{max} = t_{write} + l_{optical\ fiber} \cdot 5\ ^{ns}/_{m} + 80\ ns + t_{switch} + t_{read}$$

Where

$l_{optical\ fiber}$	is the length of the optical fiber
t_{write}, t_{read}	are the write/read access times
t_{switch}	is the read buffer switch time (block read operations in the swinging buffer mode only).

The following table lists a few access times of the required Gigalink operations (t_{write}, t_{read}, t_{switch}) for a particular configuration with two DS1006 boards (board revision: 2.1.5, CPU clock: 2.2 GHz, transfer rate: 1 GBit/s).

Action	Size	Total Access Time	Transfer Rate
Write Operation (t_{write})			
Single 32-bit word	32 bit	85 ns	47 MB/s
Single 64-bit word	64 bit	175 ns	45 MB/s
Block	32 bit	85 ns	47 MB/s
	2048 · 32 bit	177000 ns	46 MB/s
Read Operation (t_{read})			
Single 32-bit word	32 bit	565 ns	7 MB/s
Single 64-bit word	64 bit	1080 ns	7,4 MB/s
Block	32 bit	565 ns	7 MB/s
	2048 · 32 bit	1070000 ns	7,6 MB/s
Read Buffer Switch Time (t_{switch})			
		550 ns	

Note

- The times in the table denote 32-bit aligned operations (meaning that the address is a multiple of 4). Other addresses lead to reduced performance.
- The DS911 Gigalink Module is organized in 2 groups: Gigalink 0 and Gigalink 1 are one group, Gigalink 2 and Gigalink 3 are another. If one or both Gigalinks of a group receive data during a read access, the read access times decrease by 15% … 20%.

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 40.
- Channels 8 … 15 operate in the virtual shared memory mode (channel 8 is reserved). Refer to Virtual Shared Memory Mode on page 39.

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 (DS1006 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

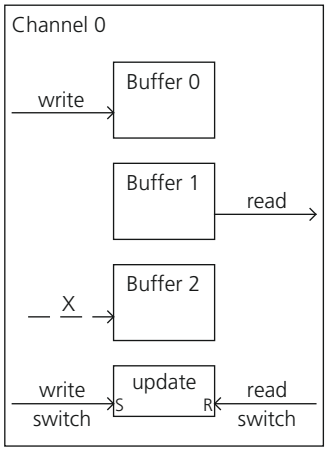
Basics

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

Swinging Buffer Mode

Introduction

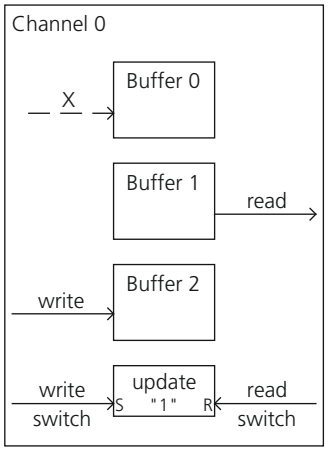
This mode is based on three receiver buffers for each channel, which all appear at the same memory location so that the sender and the receiver see only one buffer at a time. Consider the following illustration:



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:

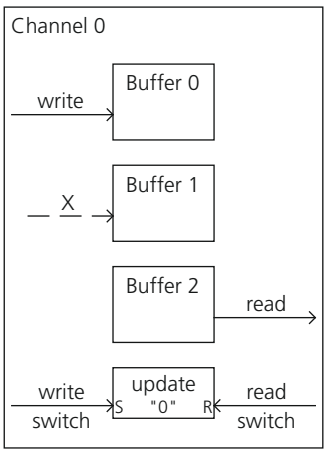


Reading data Before the receiver reads data, it sends a read switch signal. On receipt of this signal, the update flag is evaluated:

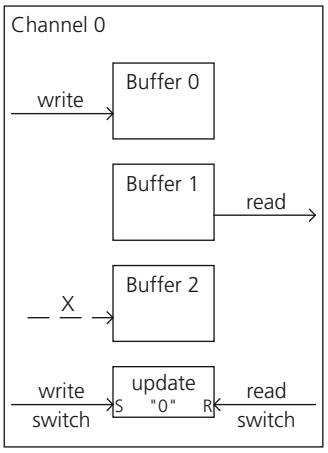
- If it is set (the X pointer marks new data), the read pointer and the X pointer are swapped.
- If it is not set (the X pointer marks old data), the read pointer remains in its old position.

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 a real-time application runs on a multiprocessor system, it is important to have the same time on all the CPUs and to trigger the timer-driven tasks of all the CPUs by means of the same timer interrupt.

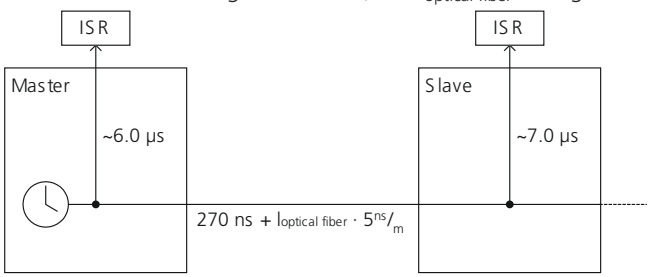
Where to go from here

Information in this section

Synchronized Timer Tasks

Introduction

The timer-driven tasks calculated on the different CPUs have to be driven by the same timer interrupt. This is done by forwarding the interrupt of the master's Timer A to all the slave boards. This forwarding causes a short delay, so that the start of the corresponding interrupt service routine (ISR) is slightly delayed, as shown in the following illustration (with $l_{optical\ fiber}$ = length of the optical fiber):



The times have been measured with a single-core board. A multicore board has a latency of ~1 µs, and the delay of the internal gigalinks is ~45 ns.

Example

If you have three single-core DS1006 connected in series (Master – Slave 1 – Slave 2) using 5 m Gigalink connections, the interrupt service routines are delayed as follows:

Board	Approx. ISR Delay
Master	6.0 µs
Slave 1	7.3 µs
Slave 2	7.6 µs

Related topics

Basics

Timer A and Timer D 15

References

Global Sample Rate Timer in an MP System (DS1006 RTLib Reference 🕮)

Global Time Base in an MP System

Introduction

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

Basics

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 the macrotick. When a macrotick occurs, the number of microticks is set to zero, and the number of macroticks is incremented at each processor. Starting from this point in time, the global time is calculated by multiplying the macrotick and microtick periods by their counter values and adding the products. The global time is used for time-stamping. At the recommended STBU synchronization period (macrotick period) of 10 ms, the global time has a maximal deviation of ±1 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 (DS1006 RTLib Reference 🕮).

Related topics

Basics

Timers and Time-Stamp Counters..... 14
Time-Stamping and Data Acquisition (RTI and RTI-MP Implementation Guide 🕮)

Tracing Signals in an MP System

Introduction

The Synchronous Time Base Unit (STBU) is used to provide a global time base in a multiprocessor system. This time base is used for time-stamping, which improves the tracing of signals in a multiprocessor system.

Distributed tracing

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

Related topics

Basics

Global Time Base in an MP System..... 44
Timers and Time-Stamp Counters..... 14
Time-Stamping and Data Acquisition (RTI and RTI-MP Implementation Guide 🕮)

Working with Subsets of a Multiprocessor Topology

Introduction

You can specify certain CPUs as optional and work with topological subsets of the currently registered MP system. This allows you to download a real-time application either to the entire MP system or to a topological subset of this system without having to rebuild the application.

Where to go from here

Information in this section

Basics of Working with Optional CPUs

Introduction

dSPACE's RTI-MP and ControlDesk provide two features that allow you to modify the topology of a dSPACE multiprocessor system. Without these features, you would have to rebuild the real-time application with RTI-MP and register the modified multiprocessor system with ControlDesk.

Specifying optional CPUs in RTI-MP

To avoid having to rebuild the real-time application, RTI-MP allows you to specify certain DS1006 processor boards as optional CPUs.

Each CPU in a multiprocessor system corresponds to a specific real-time application. Therefore, specifying optional CPUs implicitly specifies optional real-time applications. When you build the real-time application for the entire multiprocessor system, it is therefore possible to run it on different topological subsets of the system.

For details on the CPUs that can be specified as optional in an MP system, and instructions that apply to the corresponding RTI-MP model, see How to Specify Optional CPUs in RTI-MP on page 49.

Working with topological subsets of an MP system

To avoid having to re-register a multiprocessor system each time you modify it, 's the Platforms/Devices controlbar of ControlDesk allows you to handle topological subsets of a multiprocessor system containing one or more optional CPUs. This means that ControlDesk works properly even if one or more optional CPUs are currently disconnected or switched off.

For details, refer to Working with Multiprocessor Systems with Optional Processors (ControlDesk Platform Management 🕮).

Related topics

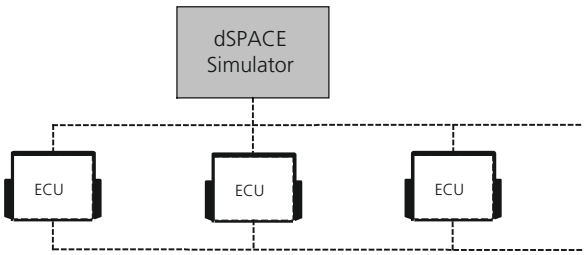
HowTos

How to Specify Optional CPUs in RTI-MP.....49

Basics of Topological Subsets

Introduction

In hardware-in-the-loop (HIL) simulations, you can test virtually any number of electronic control units (ECUs) under realistic conditions with HIL test benches based on dSPACE simulator.

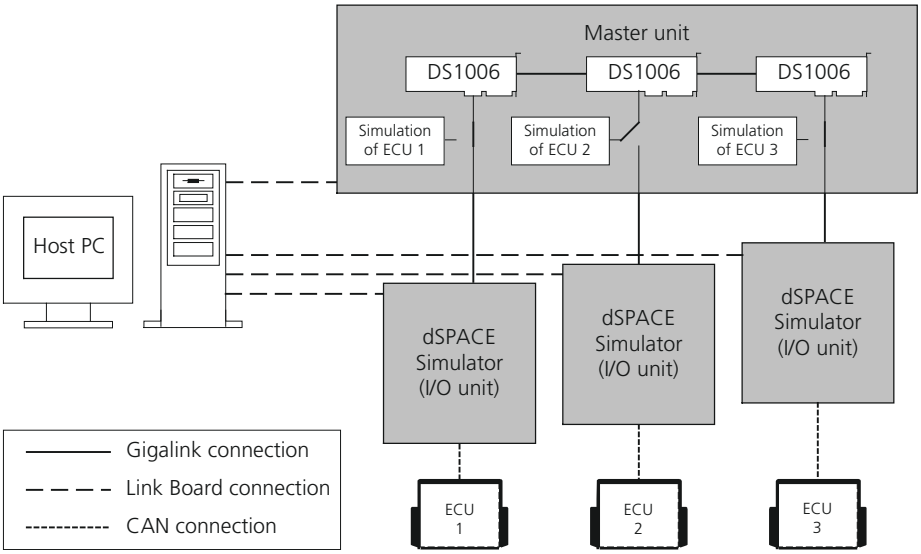


Testing networked ECUs

In many cases, it is desirable to first test each ECU separately with different dSPACE simulators. However, the overall function of many ECUs can be tested only when networked.

You can also setup a multiprocessor system from the different dSPACE simulators and perform integration tests of the networked ECUs. Each dSPACE simulator is connected to a master unit, which performs the simulation control and allows the different ECUs to be switched easily.

Possible multiprocessor system setup The following illustration shows a possible setup of this multiprocessor system. Since RTI-MP and ControlDesk allow optional CPUs and topological subsets, this setup can be used for putting ECUs connected to the dSPACE simulators into operation step-by-step.



Components of the multiprocessor system setup In this topology, the multiprocessor system consists of the following components:

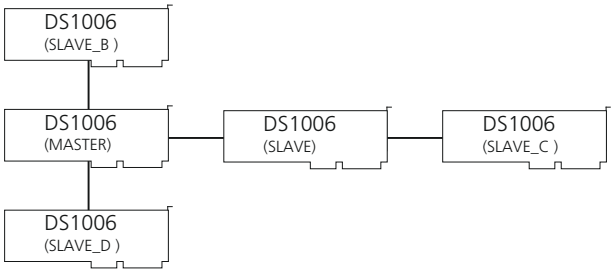
- Host PC, used to
  - build the real-time application and download it to the multiprocessor system. Since no dSPACE hardware needs to be connected to the host PC during the build process, you may also use another PC for the build: see How to Create an Application Without a Connected dSPACE System (RTI and RTI-MP Implementation Guide 🕮).
  - control and monitor ControlDesk experiments and scripts running on the connected multiprocessor system.
- Master unit, containing one or more DS1006 boards.
  - The master unit performs the communication to the connected dSPACE simulators (I/O units) via Gigalinks.
  - If one or more I/O units are currently disconnected, the master unit can perform the model calculation for the corresponding I/O submodel(s).

  During operation, the master unit must not be turned off.
- dSPACE simulators (I/O units), each containing one DS1006. They perform the communication between the connected ECUs and the master unit. Each of the I/O units may be turned on/off individually.
- ECU 1 … 3. These ECUs are connected to the dSPACE hardware via CAN.
- Simulation of ECU 1 … 3. If one or more I/O units are currently disconnected from the master unit, the signals coming from the connected ECU can be calculated in (simplified) subsystems running on the DS1006 boards of the master unit.

How to Specify Optional CPUs in RTI-MP

Objective

For DS1006-based multiprocessor systems, RTI-MP allows you to specify certain DS1006 boards as *optional*. This is possible for boards that do not forward simulation control information to further DS1006 boards. For example, only the CPUs "SLAVE_B", "SLAVE_C" and "SLAVE_D" can be specified as optional in the multiprocessor system displayed below. The CPU "SLAVE" cannot be specified as optional since it provides simulation control information to "SLAVE_C".



Tip

Since each DS1006 can be connected to up to four different DS1006 boards, one free Gigalink remains on the "MASTER" of the multiprocessor system displayed above. If you connect the "MASTER" directly to "SLAVE_C", the DS1006 named "SLAVE" can also be removed from this topology.

The following instructions apply to the corresponding RTI-MP model.

Method

To specify CPUs as optional

1 In the Multiprocessor Setup dialog, mark the Treat as optional CPU checkbox on the CPU's page of the corresponding DS1006. By default, all the DS1006 are specified as non-optional.

Note

Only end nodes can be specified as optional. The MASTER cannot be specified as optional.

Next steps

- After you specify all the Gigalink connections in the Multiprocessor Topology Setup dialog, you can build the real-time application for the entire multiprocessor system. Refer to Multiprocessor Topology Setup Dialog (RTI and RTI-MP Implementation Reference 🕮) and Building and Downloading the Model (RTI and RTI-MP Implementation Guide 🕮).

Note

The interprocessor communication between different CPUs is established via IPC blocks. During a running real-time application, all the inputs from unconnected CPUs are 0. In some cases, however, it is necessary to provide more meaningful values, for example, by simulating the application of optional CPUs. You should implement this simulation on a non-optional CPU.

- You can use ControlDesk to change the topological subset on which you want to download the real-time application. For instructions, refer to Working with Multiprocessor Systems with Optional Processors (ControlDesk Platform Management 🕮).

  ControlDesk's **Variables** controlbar displays only the variables that belong to enabled CPUs.

  You can recognize data connections between variables that belong to disabled CPUs and instruments by the no-value view.

Related topics

Basics

Basics of Topological Subsets.....47

DS1006 Application Start

Introduction

After power-up and reset, the DS1006 boots automatically and the RTP executes the boot firmware located in the on-board flash memory. There are two ways to run an application.

Where to go from here

Information in this section

Information in other sections

Boot Firmware

Introduction

After power-up and reset, the DS1006 boots from the on-board flash memory, which holds the preinstalled boot firmware.

Characteristics of the boot firmware

The boot firmware carries out the following steps:

- It determines the I/O boards connected to the DS1006's PHS bus.
- It starts the application:
  - The board is reset.
  - The firmware starts and receives the application from ControlDesk or the CompactFlash. An application loaded by ControlDesk takes precedence over an application in the CompactFlash card.
  - The application is started.
- If the CompactFlash card does not contain an application and no application is loaded by ControlDesk, the boot firmware collects information about the DS1006 boards connected via Gigalinks. This information is passed to the **Platforms/Devices** controlbar of ControlDesk, which displays the multiprocessor system topology.

Running an Application from the Memory

Introduction

In general, you download the application from the host PC to the DS1006's local memory.

Large application

If an application is larger than 10 MBytes, it cannot be downloaded to a DS1006 with a firmware version earlier than 1.0.2. In this case, update the firmware (see Updating the Firmware (DS100x, DS110x, MicroAutoBox II, MicroLabBox – Software Getting Started 🕮)). This firmware version supports application up to 32 MBytes.

Download time

Download times may differ considerably. It takes longer, for example, if the application can not be unloaded due to errors in the application or it is programmed with RTLib Version 1.0.

Running an application

After download, the application runs automatically.

Reloading an application

To stop and restart the application you have to reload it again, for example, via **Reload Application** command of ControlDesk.

Note

If you switch off the DS1006, the contents of the local and global memory are lost. If you download/reload an application, the previous contents of the global memory are overwritten.

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

Basics

Running an Application from the CompactFlash Card.....53

Running an Application from the CompactFlash Card

Introduction

If your DS1006 is equipped with a CompactFlash card, you can start an application automatically after power-up.

Loading an application

If you want an application to be started automatically after power-up, you have to load it to the CompactFlash card. This is possible via RTI, RTI-MP and ControlDesk. After the download has finished, the application is copied to the local memory, overwriting any existing application, and started automatically.

Whenever you switch on the DS1006, the boot firmware copies the application from the CompactFlash card to the local memory and starts it afterwards, regardless of whether the board is connected to the host PC or not.

Stopping and starting an application

You can stop and restart the application by resetting the DS1006, for example, by switching off the power and turning it on again. If you use ControlDesk's **Stop Real-Time Processor** and **Reload Application** commands instead, the application is first reloaded from the host PC to the CompactFlash card, and then from the CompactFlash card to the local memory. Then the application is started.

Clearing an application

To clear the application from the CompactFlash card, you have to use ControlDesk's **Clear Flash** command. Flash operations can be very time-consuming: clearing or reprogramming usually takes longer than 45 seconds.

Switching the DS1006

If you switch off the DS1006, the contents of the local memory are lost. If you switch on the power again, the contents of the CompactFlash card are copied to the local memory and started.

More information

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

For information on the flash memory's characteristics, refer to Memory on page 13.

Related topics

Basics

Running an Application from the Memory..... 52

Autobooting Real-Time Applications on dSPACE Hardware

Introduction

Some dSPACE real-time systems support autobooting a real-time application. Autobooting allows you to use such a system as a stand-alone system without a connection to the host PC.

Basics on autobooting dSPACE hardware

You can enable dSPACE hardware to start an application from flash memory or from a USB mass storage device, for example. On power-up or restart of the hardware, this application is automatically downloaded to the hardware and started on it.

After a real-time application autoboots on dSPACE hardware, you can connect ControlDesk to the hardware. ControlDesk then detects the running real-time application. ControlDesk therefore does not unload the application when online calibration is started.

Autobooting an application on DS1006

The board has a built-in CompactFlash card slot and thus supports autobooting. To prepare autobooting, you must load the application to the flash memory. For instructions, refer to How to Load an Application to the Flash Memory of dSPACE Real-Time Hardware (ControlDesk Platform Management 🕮).

Related topics

HowTos

How to Load an Application to the Flash Memory of dSPACE Real-Time Hardware (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