Generic Serial Interface (DCI-GSI2)

DCI-GSI2 Feature Reference

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

Contents

About This Reference

Contents

This document provides all the feature-by-feature information that you need to perform

- Calibration
- Measurement
- Bypassing
- ECU flash programming

with the DCI-GSI2 on your ECU.

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.

Safety Precautions

Introduction

To avoid risk of injury and/or damage to the dSPACE hardware, read and ensure that you comply with the following safety precautions. These precautions must be observed during all phases of system operation.

Warning About Using a DCI-GSI2

Introduction

Note the following warning and safety precautions when using a DCI-GSI2.

Danger potential

Connecting a DCI-GSI2 to an electronic control unit can affect system behavior. This can lead to unexpected or critical situations, or even constitute a risk of death. Therefore, only persons who are qualified to use a DCI-GSI2, and who have been informed of the dangers and possible consequences, are permitted to use the DCI-GSI2.

Before integrating the DCI-GSI2 and starting operation, read the warnings in this document carefully.

⚠ WARNING

Risk of serious injury or death due to electrical shock

The DCI-GSI2 is designed to be connected to devices that do not transmit hazardous voltages. According to the EN 61010 standard, a voltage higher than 33 V_{RMS}/ 46.7 V_{PEAK} AC and 70 V DC is classified as hazardous. It constitutes a risk of serious injury or even death.

Make sure that your system provides safety provisions so that no hazardous voltages are applied to the DCI-GSI2, even in the event of electrical faults.

If there is a risk of hazardous voltages being applied to a DCI-GSI2, for example, when it is connected to an engine ECU which typically generates transient hazardous voltages for ignition, one of the following measures must be taken **to avoid the risk of serious injury or death due to electrical shock**:

- The DCI-GSI2 and all devices connected to it must be within a separate test area according to the locally valid safety standards for the installation and operation of electrical test equipment.
- dSPACE provides dedicated interface cables to ensure an electrically safe connection to the host PC for systems featuring voltages up to 300 V DC/AC_{RMS} , or 600 V_{peak}: The ETH_CAB2 Ethernet Connection Cable must be used for connecting the DCI-GSI2 to the host PC.
  The DCI-GSI2 and the devices connected to it must be within a separate test area. When the above-mentioned cable is used, the host PC can be located outside the test area.

Using the DCI-GSI2 on wet locations

According to IEC 61010-1 (product safety), the DCI-GSI2 is not intended to be used on wet locations.

Unless the DCI-GSI2 is protected by a waterproof enclosure complying with the IP66 protection classification, using it in wet conditions might result in electric shock due to hazardous voltages or might damage the DCI-GSI2 and any connected ECU.

Electromagnetic compatibility

The DCI-GSI2 is a CE class A device. This equipment may cause interference in a residential installation. In this case the user is encouraged to perform appropriate measures to correct the interference. For details on CE compliance, refer to Certifications of the DCI-GSI2 on page 71.

Liability

It is your responsibility to adhere to instructions and warnings. Any unskilled operation or other improper use of this product in violation of the respective safety instructions, warnings, or other instructions contained in the user documentation constitutes contributory negligence, which may lead to a limitation of liability by dSPACE GmbH, its representatives, agents and regional dSPACE companies, to the point of total exclusion, as the case may be. Any exclusion or limitation of liability according to other applicable regulations, individual agreements, and applicable general terms and conditions remain unaffected.

Features of the DCI-GSI2

Introduction

For modern automotive microcontrollers featuring an on-chip debug interface, dSPACE provides the DCI-GSI2. You can use the DCI-GSI2 as the interface between the ECU and the host PC and/or the bypass system.

Where to go from here

Information in this section

General Features of the DCI-GSI2

Introduction

The number of electronic control units (ECUs) is increasing in modern vehicles. ECUs are used to minimize exhaust gas emissions, optimize driveability, minimize fuel consumption, and perform many other functions. The functioning of an ECU can be calibrated, in other words, fine-tuned. The results of calibration can be measured. ECU functions can be calculated outside the ECU on an external bypass system, which sends its results back to the ECU. The ECU code contains parameters for calibration and variables for measurement.

With the DCI-GSI2, you have one single tool to perform calibration, measurement, bypassing and flash programming tasks on an ECU.

The DCI-GSI2 supports various types of on-chip debug ports for ECU calibration, measurement, ECU flash programming and function bypassing, for example, JTAG/Nexus and JTAG/OCDS.

Illustration

The following illustration shows the DCI-GSI2 in its opened enclosure and a connector adapter:

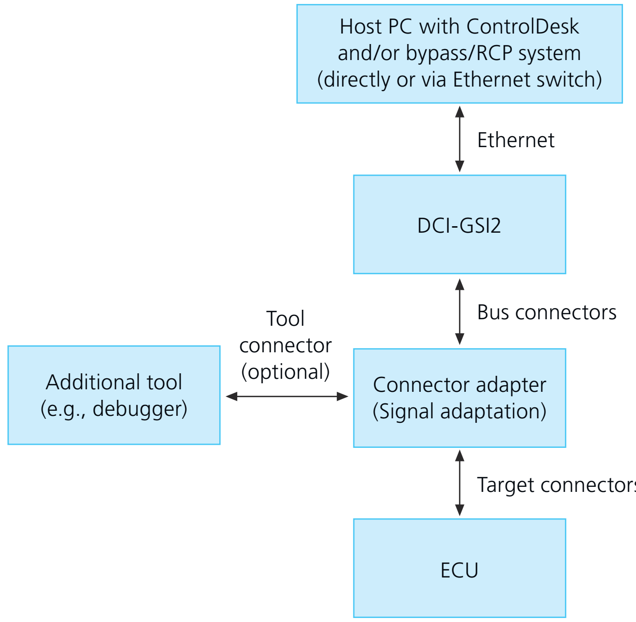
Generic design for multiple use

The DCI-GSI2 is of generic design, that is, it can be used with a wide range of different ECUs. It can be adapted to a specific ECU, for example, the ECU interface type, by a firmware and/or configuration update.

The DCI-GSI2 is equipped with two bus connectors to interface your individual ECU interface via an appropriate connector adapter. dSPACE provides various ECU-specific connector adapters.

System overview

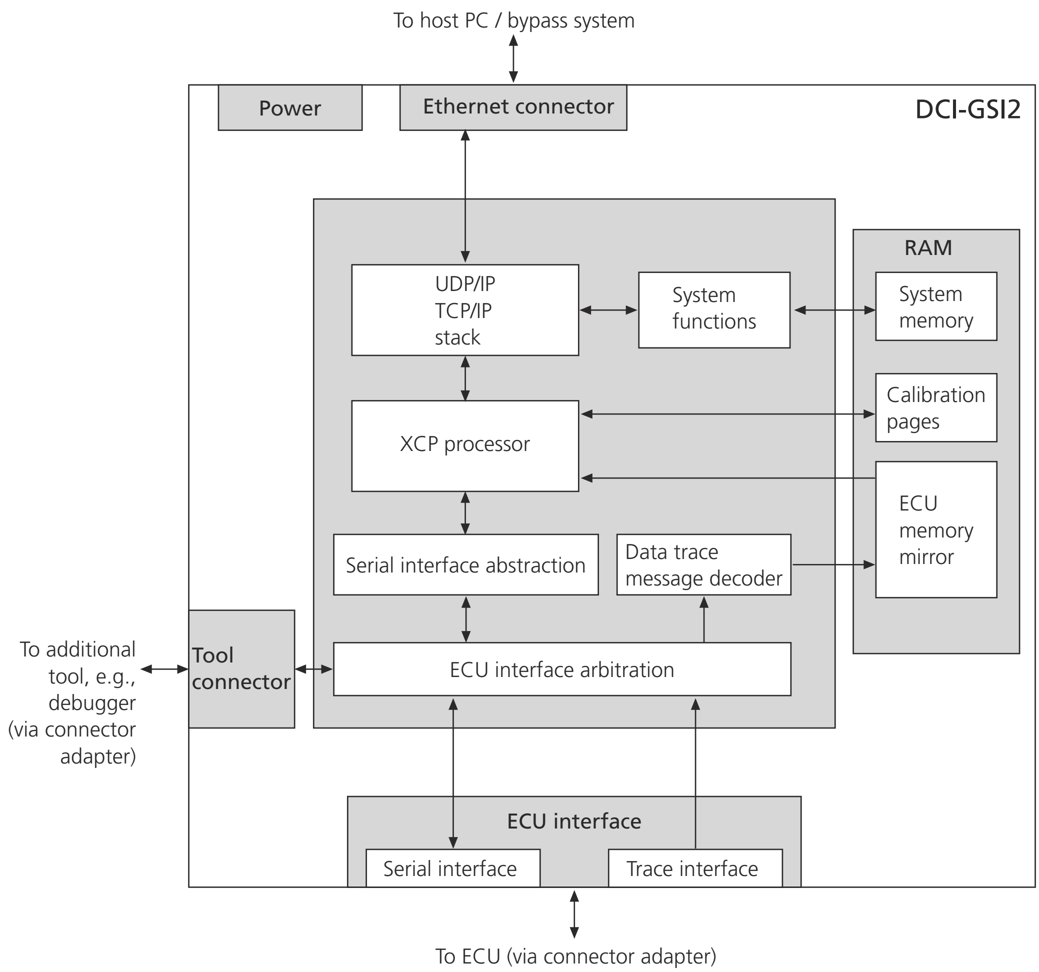
The following illustration shows the DCI-GSI2's hardware connections to the host PC, the ECU, the bypass system and an additional tool, for example, a debugger. Connection to an additional tool is optional, because it depends on the serial interface and connector adapter used.



Simultaneous calibration and ECU interfacing If you want to perform measurement, ECU calibration, and ECU interfacing in parallel, the host PC (used to access the ECU for ECU calibration and measurement) and the dSPACE real-time system (used to access the ECU for ECU interfacing) must be connected to the DCI-GSI2 via an Ethernet switch. Refer to Connecting an ECU with DCI-GSI2 for Simultaneous Calibration and ECU Interfacing (ECU Interfaces Hardware Installation and Configuration 📖).

Principle architecture

The following illustration shows the principle architecture and the most important functional units of the DCI-GSI2:



The functional units in the illustration above perform the following functions:

Functional Unit	Function
Ethernet interface	Ethernet connection between the DCI-GSI2 and the host PC and/or the bypass system.
ECU interface	Connects the DCI-GSI2 to the ECU via connector adapter. Refer to Adaptation to the ECU on page 61.
Tool interface	Connects an additional tool, for example, a debugger, to the DCI-GSI2. Refer to Debugging the ECU on page 42.
Power supply	Meets the requirements of the conditions in a vehicle and on the test bench. Refer to Power Management on page 44.

Open and standardized interfaces

The DCI-GSI2 is connected to the host PC and the bypass system via XCP on Ethernet, that is, the open and standardized XCP interface is used to interface the host PC for calibration and measurement and the prototyping system for function bypassing. This also means compatibility to third-party tools via XCP.

Using multiple service instances The DCI-GSI2 supports three separate XCP service instances. Each service instance behaves as an XCP slave, providing independent access to the ECU resources. You can therefore simultaneously connect the ECU with DCI-GSI2 with up to three tools. For example, you can connect your ECU with DCI-GSI2 with ControlDesk and a dSPACE prototyping system for simultaneous calibration, measurement and bypassing. The assignment of the XCP service instances to the tools is configurable and depends on the Ethernet port number.

Note

However, if you work with multiple service instances, they are not completely independent of each other since they all access one ECU microcontroller. For example, you cannot bypass an ECU and program the ECU's flash memory at the same time.

Access to ECU via ControlDesk's devices

For calibration, measurement and ECU flash programming purposes, you can access an ECU with DCI-GSI2 from within ControlDesk. ControlDesk provides the DCI-GSI2 device for this. Alternatively, you can use the XCP on Ethernet device.

Note

It is recommended to use the DCI-GSI2 device in ControlDesk since it provides some DCI-GSI2-specific features. For example, ControlDesk automatically detects all the connected DCI-GSI2s when you open or create an experiment in ControlDesk, so the IP address and port number for communication do not have to be specified in the A2L file.

Configuring the DCI-GSI2

dSPACE provides the DCI Configuration Tool, which allows you to set and modify the configuration settings of your DCI-GSI2.

> **Note**
>
> - A DCI-GSI2 shipped by dSPACE comes with a default configuration for the target processor and interface type it was ordered for. The DCI-GSI2 must not be connected to your ECU unless it is configured correctly, so you must check the device configuration yourself. You might need to optimize the device configuration.
> - Before changing the configuration of a DCI-GSI2, create a backup of the current configuration. For each available target processor family, a default configuration file is provided with the DCI Configuration Tool. Since dSPACE archives the configuration of a DCI-GSI2 before it is shipped, you can also get a backup of the initial configuration of your DCI-GSI2 from dSPACE, if necessary.

For further information on the DCI Configuration Tool and how to work with it, refer to the DCI Configuration 🕮 document.

Related topics

Basics

Configuring a DCI-GSI2 (DCI Configuration 🕮)

Calibration Features

Introduction

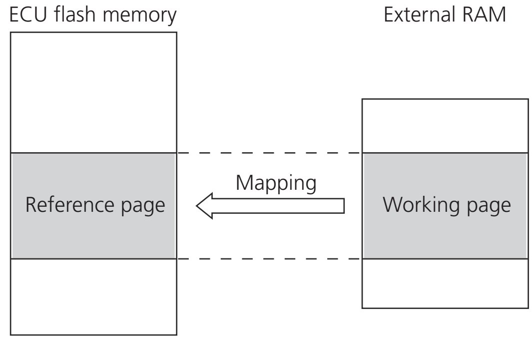
ECUs are calibrated by tuning the ECU parameters, which are located in the ECU's flash memory. The DCI-GSI2 can be used to access and calibrate these parameters.

The DCI-GSI2 provides two calibration methods:

- ECU calibration in external ECU RAM
- ECU calibration via overlay units of the ECU processor

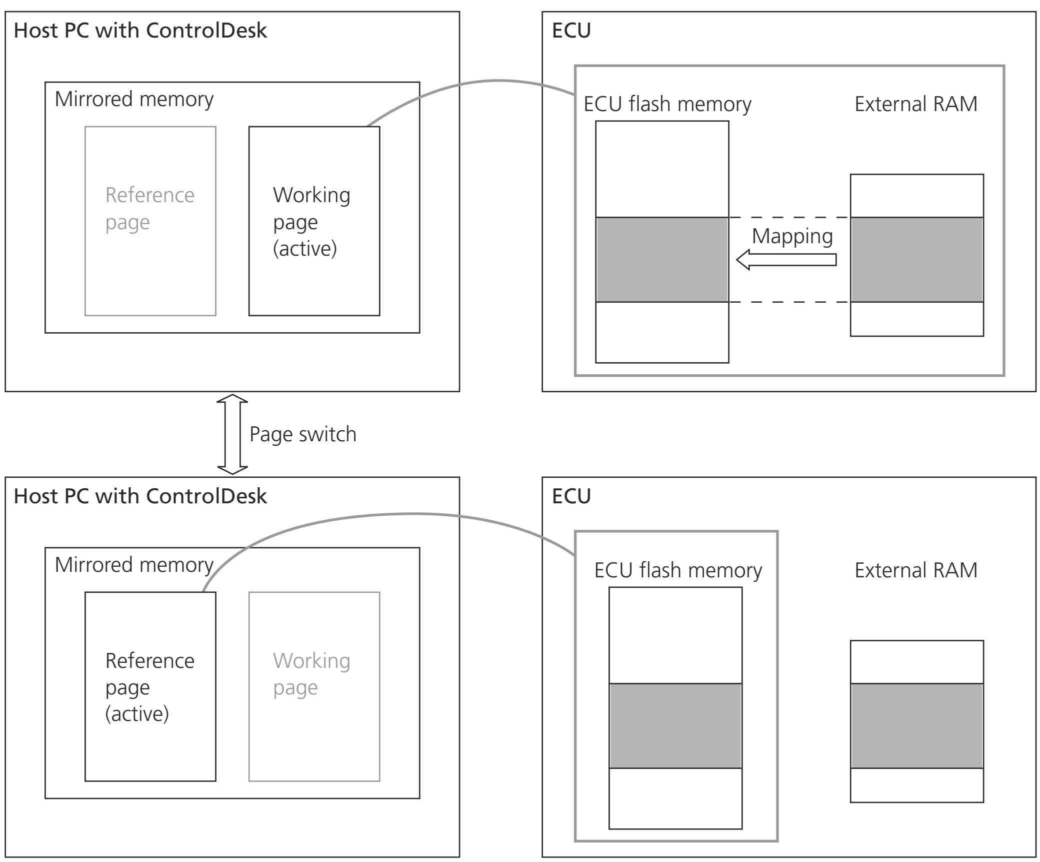
Calibration in external ECU RAM

The DCI-GSI2 provides ECU calibration on an external RAM located in the ECU. The reference page is stored in the ECU flash memory, and the working page is completely stored in the external RAM. When required, the external RAM is mapped to the ECU's flash memory.



Addresses are translated from the reference page to the working page by adding an offset to each address in the flash memory. The offset is calculated by the DCI-GSI2. You must specify the start address of the working page in the external ECU RAM in the DCI-GSI2 configuration for this purpose.

Switching memory pages Switching memory pages means enabling or disabling the memory mapping. If you switch from the reference page to the working page via ControlDesk, the RAM is mapped to the flash memory, and the ECU application works with the modified ECU parameters. If you switch from the working page to the reference page, the memory mapping is disabled.



The DCI-GSI2 provides page switching via ECU service. Page switching is performed by the dSPACE Calibration and Bypassing Service, which means that the ECU code must be instrumented. The `dsecu_custom_cal_page_switch` function is required for page switching and must therefore be integrated in the ECU code.

Page switching via ECU service is available for processors which provide a mapping mechanism by means of hardware, where page switching is done by the ECU only (for example, a memory management unit (MMU)).

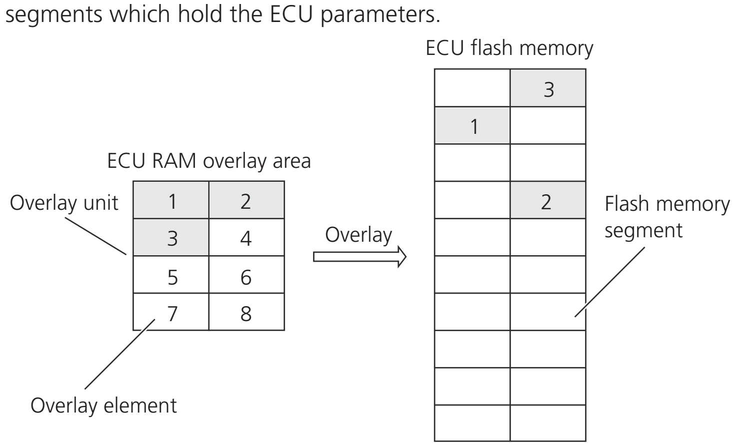
Configuration settings for calibration in external RAM Before ECU calibration in external RAM can be performed, the calibration method must be configured. The DCI-GSI2 must know the location of the mapped memory range (working page) and the page switching method. To specify the configuration settings for calibration in external RAM, you must adapt the DCI-GSI2 configuration accordingly.

Calibration via overlay units

Calibration of ECU parameters via overlay units requires a memory overlay feature in the ECU. Many ECU microcontrollers provide this feature. The most common implementation is to overlay internal flash memory with elements of microcontroller-internal RAM (overlay RAM).

An ECU parameter in the ECU's flash memory can be changed by writing the new parameter value to the overlay RAM and mapping the overlay element to the address location of the parameter. You can make changes to further ECU parameters as long as free overlay elements are available.

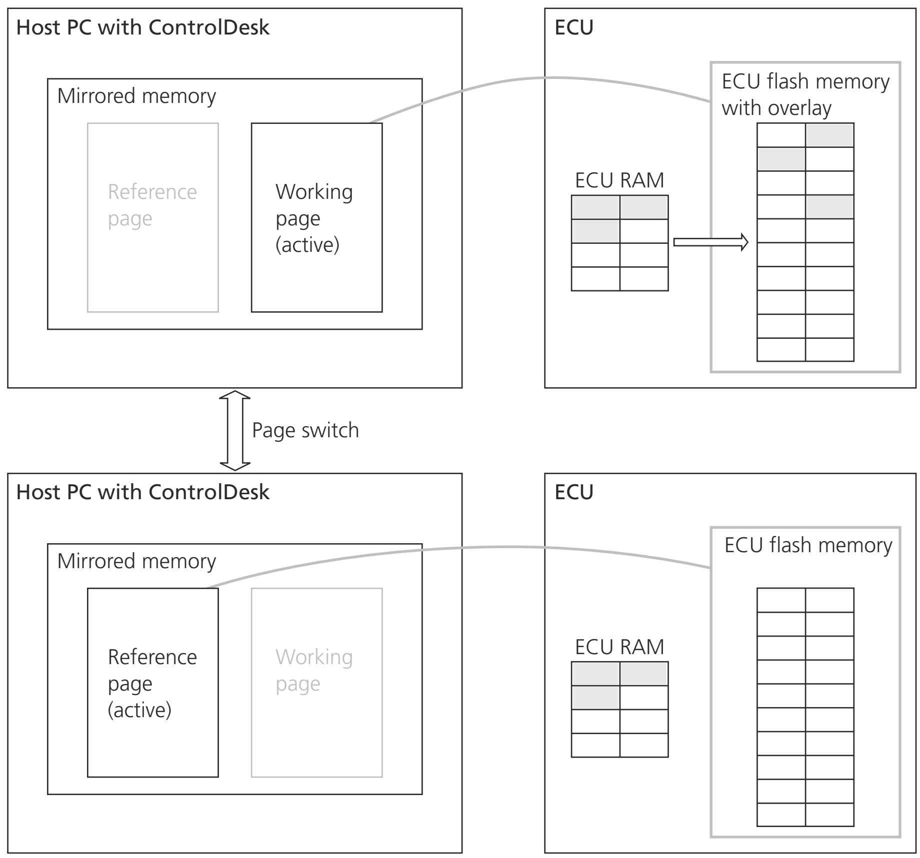
Example The following illustration shows the relation between the ECU RAM overlay area and the ECU's flash memory. Overlay is shown for an ECU's RAM with one overlay unit consisting of 8 overlay elements. Three overlay elements are used for ECU parameter changes. These elements overlay flash memory segments which hold the ECU parameters.



Dependencies of the overlay method Calibration with the DCI-GSI2 greatly depends on the extent to which the ECU's microcontroller supports the overlay method. The ECU's microcontroller determines the total size of the overlay area and the memory positions where overlay is possible. The available number of overlay elements and their sizes also depend on the ECU's microcontroller. Thus, the maximum number of ECU parameters that can be calibrated within a calibration task is determined by the size of the ECU's overlay RAM.

Switching memory pages In most calibration tasks, it is necessary to change not only a single parameter, but a whole parameter set. A parameter set switch must be performed concurrently, that is, all the parameters of the data set must be changed at the same time.

If you switch from the reference page to the working page via ControlDesk, the ECU's microcontroller-internal overlay units are activated. The affected segments of the ECU's flash memory are overlaid by the corresponding overlay elements of the ECU's RAM. If you switch from the working page to the reference page via ControlDesk, the overlay units in the ECU's flash memory become inactive.



The DCI-GSI2 allows you to switch between the pages during ECU operation. When you change parameters on one page, you can make the changes available to the ECU via a single page switch. If you request a page switch, it is either executed regardless of the ECU's status or related to an ECU event.

Mapping between logical and physical memory addresses

Addresses specified in an A2L file usually apply to logical memory addresses. The ECU-internal mapping between the logical and physical memory addresses is performed by the target processor's memory management unit (MMU). However, memory accesses by the DCI-GSI2 via the debug interface might not be handled by the MMU, so that the DCI-GSI2 needs to use physical addresses.

To let the DCI-GSI2 know the mapping between the logical and the physical memory addresses, the memory mappings must be specified and activated in the DCI-GSI2 configuration. If the logical addresses specified in the A2L file match the physical addresses, you do not have to specify memory mappings.

Comparison of pages via checksums

On the host PC, ControlDesk has the mirrored memory consisting of the reference page and the working page. The ECU with DCI-GSI2 has memory pages. If the connection between ControlDesk and the ECU is offline, the pages in ControlDesk can differ from the memory pages of the ECU. When you start online calibration, ControlDesk automatically compares the ECU's memory contents with the corresponding pages in ControlDesk. To speed up the comparison, checksums are used. ControlDesk calculates checksums for the reference page and the working page. The DCI-GSI2 calculates checksums for the memory pages of the ECU. If the pages differ, you have to equalize them to ensure data integrity on the host PC and on the ECU. You can either upload the ECU's memory contents to ControlDesk, or download the pages from ControlDesk to the ECU.

Note

Since the ECU's flash memory is read-only, you usually cannot download the reference page from ControlDesk to the ECU. You must flash the ECU's flash memory with the reference page via an external flash tool such as the dSPACE ECU Flash Programming Tool.
However, there are cases where the DCI-GSI2 cannot initialize the reference page by reading the ECU flash memory contents (for example, if the ECU's flash memory is read-protected). In these cases, the reference page data can be downloaded to the DCI-GSI2. You must explicitly enable the download feature in the DCI-GSI2 configuration.

ControlDesk allows you to start online calibration without equalizing the contents of the memory pages and the mirrored memory. Differences in the memory contents are ignored. You can calibrate parameters even if the parameter values on the connected hardware and on the host PC differ. In this case, you must ensure data integrity yourself. You can specify ControlDesk's default behavior when online calibration is started during device configuration via the **Properties** controlbar in ControlDesk. Refer to General Settings Properties (ControlDesk Platform Management 🕮).

Page switch related to an ECU event

If you request a page switch via ControlDesk, it can be switched immediately or be delayed until an ECU event occurs. The DCI-GSI2 offers hardware-support for detecting ECU events.

Related topics

References

dsecu_custom_cal_page_switch (dSPACE Calibration and Bypassing Service Implementation 📖)

Measurement Features

Introduction

The DCI-GSI2 supports several measurement strategies. Some measurement features are independent of the applied strategy.

Measuring variables

You can measure variables to visualize time traces of ECU variables, evaluate the effects of parameter calibration, and capture ECU variable values for later analysis.

Options for data acquisition with the DCI-GSI2

The DCI-GSI2 supports several measurement strategies. You can perform data acquisition (DAQ) with or without an ECU software service, depending on your requirements for data consistency. Data acquisition can be triggered by the ECU for synchronous measurements, or asynchronously by using DCI-GSI2-internal timers. Refer to Overview of the DCI-GSI2 Event Types on page 54.

The following table shows the different options for DAQ with the DCI-GSI2 and some of their characteristics:

	Measurement Without Changes to ECU Software	Measurement with Changes to ECU Software
With Data Consistency[1)]	*Using a data trace interface* ▪ Data is acquired synchronously with ECU tasks. ▪ Very fast data acquisition possible (up to 10 µs raster) ▪ DCI-GSI2 supports up to 32 distinct measurement triggers. For further information, refer to Measurement via Data Trace Interface on page 28.	*dSPACE Calibration and Bypassing Service is integrated in ECU software.* ▪ Data is acquired synchronously with ECU tasks. ▪ You can measure with several thousands of rasters by using a single hardware event. ▪ Service can also be used for bypassing. Refer to Measurement with ECU Service on page 24.
Without Data Consistency	There are two basic event sources: *Using DCI-GSI2-internal timer* ▪ Data is acquired asynchronously to the ECU. *Using available hardware events (e.g., watchpoint pin, polling event)* ▪ Data is acquired synchronously with ECU tasks.	*dSPACE code patch is integrated in ECU software.* ▪ Data is acquired synchronously with ECU tasks. ▪ You can measure with several thousands of rasters by using a single hardware event.

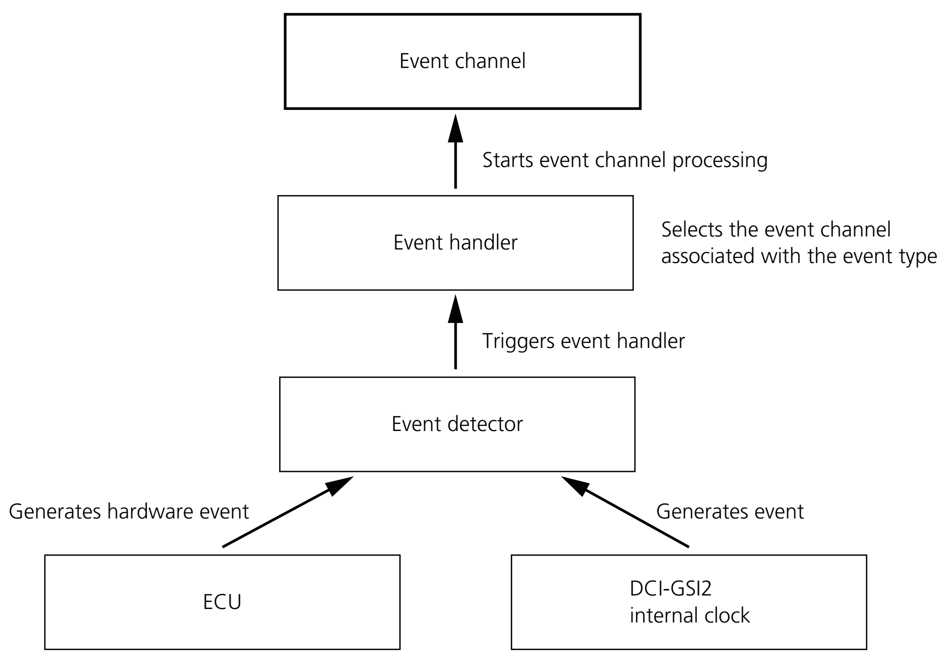
	Measurement Without Changes to ECU Software	Measurement with Changes to ECU Software
	▪ Number of available events depends on the ECU and DCI-GSI2 hardware. Refer to Measurement Without ECU Service on page 26.	▪ Minimal memory consumption and run time overhead.

[1)] Data consistency means that, when measuring several variables, at the time of the measurement trigger a consistent set of data is captured.

Principle of DAQ with the DCI-GSI2

Event channels The DCI-GSI2 supports up to 32,768 so-called *event channels*. An event channel can be configured by the host tool (for example, ControlDesk) to have several DAQ lists containing the variables to be measured. Each configured event channel is associated with an event source. When the event is detected, the event handler initiates event channel processing. All the configured measurement variables are captured from the ECU, and the measurement data is sent to the host tool via DAQ packets. The DAQ packet is given the time stamp at which the event channel was triggered.

The illustration below shows the principle of the processing for one event channel:



Supported event sources The DCI-GSI2 supports two basic event sources:

- *ECU events*: This event source is used for event-triggered tasks, for example, for crankshaft-synchronous sampling. You can define acquisition rasters that are determined by an ECU event, so data points are taken when an ECU event occurs.

- *DCI-GSI2 internal clock events*: This event source is used for DCI-GSI2-synchronous tasks, that is, for time-triggered sampling. You can define acquisition rasters that are determined by the DCI-GSI2's clock. Data points are equidistant according to the clock. DCI-GSI2-triggered sampling can be performed without changing of the ECU code. However, this kind of measurement is not synchronous with any ECU task.

Supported event types To detect ECU events and trigger data acquisition, the DCI-GSI2 supports several event types. Refer to Overview of the DCI-GSI2 Event Types on page 54.

Measurement of parameters

In addition to calibrating ECU parameters, you can also measure them in ControlDesk. However, depending on the architecture of the ECU microcontroller and the configuration and implementation of the calibration method, parameter changes on the working page may not be visible in the measurement. In general, it is therefore recommended to measure parameters only by using the OnChange raster in ControlDesk. For more information, refer to Basics on Measurement Rasters (ControlDesk Measurement and Recording 📖).

Note

Since ECU parameters are not written by the ECU application, they are not transmitted via the data trace interface. Instead, they must be read via the serial interface. This impairs measurement performance.

Related topics

Basics

Measurement with ECU Service

Introduction

The DCI-GSI2 provides service-based measurement. The ECU service provides access to the measurement variables at a dedicated ECU code point. Thus, if a DAQ service is included in the ECU code, the measurement data is acquired synchronously with the ECU tasks.

Basics of service-based measurement

Service-based measurement comprises the following steps:

1. During initialization of the ECU service, the addresses of the variables to be measured are transferred to the ECU service memory.
2. An ECU service call in the ECU task triggers data acquisition.
3. When an ECU service call occurs, the address lists in the service memory are interpreted, and the measured values are collected in the service memory.
4. After the measurement data has been sampled, the ECU triggers an event on the DCI-GSI2 to read the acquired data.

   There are several event types available for the DCI-GSI2 to detect ECU events and trigger data acquisition. Refer to Overview of the DCI-GSI2 Event Types on page 54.
5. In the DCI-GSI2, each measurement is given a time stamp. The measured data is then transferred to the measurement host tool.

The variables are measured synchronously with the ECU task, so the measurement data set consists of the actual values at the time of the ECU service call.

> **Note**
>
> Implementing a service in the ECU application results in a higher processor load.

Supported ECU service

For service-based measurement using the DCI-GSI2, the ECU code must be implemented with the dSPACE Calibration and Bypassing Service. You are recommended to use dSPACE Calibration and Bypassing Service 2.3.0 or later.

Measurements with and without data tracing

The DCI-GSI2 provides measurements with and without data tracing. The data trace mechanism can also be used in combination with the dSPACE Calibration and Bypassing Service. This is useful if the number of required measurement rasters exceeds the number of trace write events supported by the DCI-GSI2, or if bypassing is also to be performed with the service. In this case, measurements will be very fast due to the trace interface, and data stimulation will be done by using the serial ECU interface.

For further information, refer to Measurement via Data Trace Interface on page 28.

Supported acquisition rasters

You can cover a wide measurement range by defining acquisition rasters.

- If you use a dSPACE Calibration and Bypassing Service version earlier than 2.3, you can measure up to 255 acquisition rasters.
- As of dSPACE Calibration and Bypassing Service 2.3, up to 65,535 service IDs for dSPACE Calibration and Bypassing Service calls in the ECU code are supported.

Related topics

Basics

Measurement Without ECU Service

Introduction

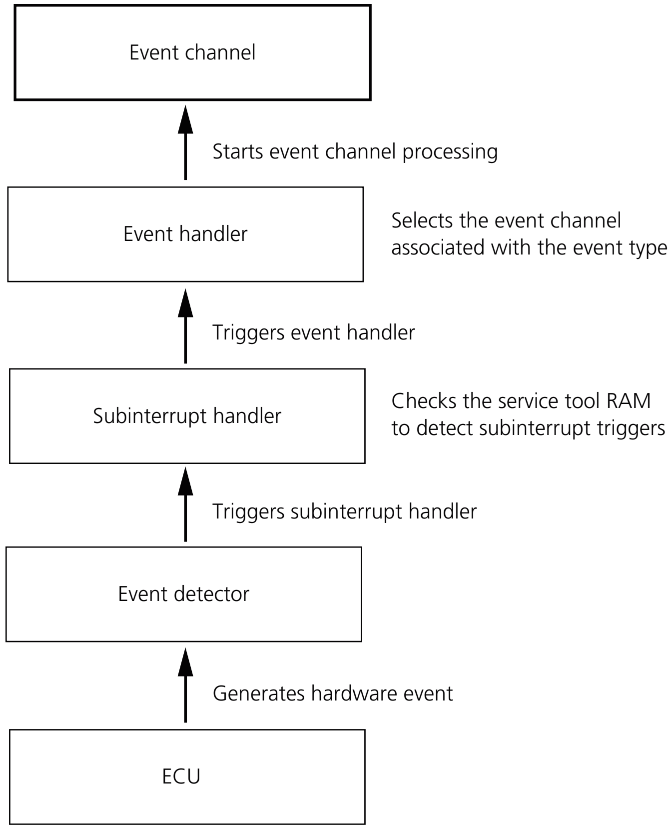
The DCI-GSI2 supports three basic principles of measuring ECU variables if no ECU service is being used:

- Code-patch-based measurement
- Measurement without a code patch
- Measurement using a data trace interface

Code-patch-based measurement

For code-patch-based measurement, a small piece of code needs to be integrated into the ECU software and a function needs to be called wherever a measurement should be triggered. This function sets a subinterrupt and then triggers a hardware interrupt, which in turn triggers the event processing in the DCI-GSI2.

The illustration below shows the principle of the subinterrupt mechanism used in code-patch-based measurements with the DCI-GSI2:



Measurement without a code patch

Without any changes to the ECU software and without the availability of a data trace interface, measurement can be performed using two different kinds of events:

- ECU events that are triggered via watchpoint pins, watchpoint messages or polling events: These events allow measurements that are performed synchronously to the ECU task execution.
- Internal timer events of the DCI-GSI2, which are triggered periodically in configurable rasters: These events allow measurements that are performed asynchronously to the ECU task execution.

Measurements with data tracing

The DCI-GSI2 supports measurement using an ECU data trace interface. Data acquisition using the data trace mechanism allows very fast data capturing of ECU variables while maintaining data consistency.

For further information, refer to Measurement via Data Trace Interface on page 28.

Related topics

Basics

Examples

Measurement via Data Trace Interface

Introduction

The DCI-GSI2 provides measurements with and without data tracing. Both concepts are available for service-based and serviceless data acquisition.

Measurements using the data trace mechanism allow very fast data captures via the data trace interface of your ECU, without having to integrate the dSPACE Calibration and Bypassing Service to achieve data consistency.

Measurement with data tracing

For ECUs/microcontrollers that provide a data trace interface, the DCI-GSI2 supports data acquisition with data tracing. In this case, the on-chip debug controller inside the ECU's microcontroller autonomously transmits written data during run time via the data trace interface to the DCI-GSI2. This allows the DCI-GSI2 to create a local mirror of the contents of the ECU RAM. Measurements from this mirror memory have low latencies and can be performed very fast.

> **Note**
>
> However, optimum performance and data consistency can be reached only if all the variables to be measured are within the configured data tracing segments, that is, within the ECU RAM mirror in the DCI-GSI2.

For measurement with data tracing, data acquisition should be triggered by a *data trace write event* (see Data Trace Write Event on page 55). After a data trace write event is detected, the measurement data associated with it is sampled consistently from the mirror memory and then transmitted to the host tool.

Principle of DAQ using the data trace mechanism As soon as a data trace write event occurs, all the variables belonging to it are sampled and transferred to the RCP system. Variables which are located within one of the data trace segments are sampled by reading their values from the internal ECU RAM mirror in the DCI-GSI2. Variables which are located outside the data trace segments are sampled by reading their values directly from the ECU RAM using the serial interface. However, data consistency cannot be guaranteed for variables located outside the data tracing area without an ECU service.

In other words, DAQ using a data trace write event involves the following steps:

- The DCI-GSI2 watches for data trace write events by watching a variable (see Data Trace Write Event on page 55).
- The ECU application writes a watched variable.
- The DCI-GSI2 detects this write operation and associates a data trace write event to it.
- The variables belonging to the event are sampled. The variable values are read from the internal ECU RAM mirror, if available, or directly from the ECU RAM. The sampled data is transmitted to the RCP system.

  (The variables belonging to the event are specified in the RTI Bypass Blockset. They serve as input variables for calculating the bypassed function(s) on the RCP system.)
- An RCP system task is triggered, and the bypass function(s) are calculated on the RCP system.

Tip

For an example, refer to Example Sequences of DAQ and STIM Functionalities on page 38.

Measurement without data tracing

If the ECU does not provide a data trace interface or data tracing is deactivated for any reason, measurement is performed without an ECU RAM mirror in the DCI-GSI2. When an event channel is triggered, the measurement variables are sampled via the serial interface.

Note

Measurement without data tracing is slower than measurement with data tracing. In addition, the variables can be inconsistent because accesses via the serial interface are always asynchronous to the ECU application. To get consistent measurements without a data trace interface, a service must be integrated into the ECU application.

Related topics

Basics

Examples

Quick Start Measurement

Introduction

The DCI-GSI2 supports quick start measurement on ECUs. The ECU measurement variables are measured during or directly after the ECU startup.

Cold start measurement

Quick start measurement also allows you to perform *cold start measurements*. Cold start means that the vehicle and/or the engine are cooled down to the temperature of the environment and then started. Cold start measurements are performed, for example, to observe the behavior of an engine during the warm-up phase.

Mechanisms for performing quick start measurements

There are two mechanisms available for performing quick start measurements. Both are available in quick start measurements with and without ECU service.

Setting up the measurement and then turning on the ECU If the DCI-GSI2 is powered, a measurement can be configured and started with the host tool while the ECU is powered off. Measurement data is sampled as soon as possible after the ECU power is turned on.

Resume mode If the DCI-GSI2 cannot be turned on before the ECU is powered, the resume mode must be used. The resume mode enables automatic data transfer after ECU power-up. This requires the quick start measurement configuration to be saved to nonvolatile memory in the DCI-GSI2. After the DCI-GSI2 is powered on, it autonomously starts sampling and transmitting measurement data to the host tool.

Loss of measurement data

Depending on whether you perform service-based or serviceless measurement, and depending on your DCI-GSI2's configuration, measurement data may get lost during a quick start measurement.

Measuring without ECU service In serviceless measurements with the DCI-GSI2, no measurement data is lost if the delay time after an ECU reset is suitably configured. You can configure the delay time with the DCI Configuration Tool.

Measuring with ECU service When the ECU is powered up, it performs an alive check (provided by the dSPACE Calibration and Bypassing Service) to detect the dSPACE service implementation. The DCI-GSI2 then writes DAQ tables (which are part of the dSPACE Calibration and Bypassing Service configuration) to the ECU as soon as the defined init delay time has expired. If the DAQ tables are consistent, the measurement starts.

To avoid the loss of any measurement data, the ECU application start-up must be synchronized with the service configuration by means of specific service functions. For further information, refer to the dSPACE Calibration and Bypassing Service Implementation 🕮 document.

Note

Measurement data can be lost at the beginning of a quick start measurement.
Whether measurement data is lost or not, depends on the ECU reset delay time and on how long it takes the ECU to pass through the alive and version information mechanism. You can configure the delay time with the DCI Configuration Tool. For details on the alive and version information mechanism, refer to Alive and Version Information Mechanism (dSPACE Calibration and Bypassing Service Implementation 🕮).

Instructions

You can use ControlDesk to prepare the DCI-GSI2 for quick start measurement. For instructions on how to prepare and perform quick start measurements on an ECU with DCI-GSI2, refer to

- How to Prepare a Quick Start Measurement on an ECU with XCP or DCI-GSI2 (ControlDesk Measurement and Recording 🕮)
- How to Perform a Quick Start Measurement on an ECU with XCP or DCI-GSI2 (ControlDesk Measurement and Recording 🕮)

Related topics

Basics

Bypassing Features

Introduction

For development, test and optimization purposes, ECU tasks can be calculated outside the ECU on an external bypass system, for example, a MicroAutoBox II. The ECU makes the required ECU variables available to the bypass system and receives the calculated results from the bypass system. The results can be used for further processing by the ECU.

The DCI-GSI2 performs the synchronous transfer of input and output data for bypassing and triggers the bypass system.

External bypassing

External bypassing is a technique which allows ECU functionalities to be executed not by the ECU's microcontroller but by an external rapid control prototyping (RCP) system. This enables you to develop, test and optimize ECU functions quickly, without altering the ECU code. Before an external function is called, the function parameters are sent to the bypass system, which then returns the function result to the ECU.

The DCI-GSI2 supports external bypassing by providing an XCP interface supported by dSPACE rapid control prototyping systems.

Bypassing with the DCI-GSI2

Bypassing with the DCI-GSI2 is realized by combining synchronous data acquisition (measurement of ECU variables) and synchronous data stimulation (modification of ECU variables).

The ECU supplies the bypass system with the required input values for calculation. The ECU generates an event when the input data is available. This event triggers the sampling of the input values by the DCI-GSI2. The sampled data is forwarded to the RCP system via Ethernet. The bypass system uses the input values to calculate the function, and generates the output values resulting from the bypass task. The output values are transferred from the RCP system to the DCI-GSI2. The DCI-GSI2 then forwards the output values to the ECU.

The DCI-GSI2 supports external rapid prototyping by means of service-based, serviceless and code-patch-based bypassing.

Options for bypassing

The DCI-GSI2 provides several functionalities for data acquisition (DAQ) and data stimulation (STIM) purposes using different event sources to trigger DAQ and STIM. Refer to Overview of the DCI-GSI2 Event Types on page 54. You can combine the different DAQ and STIM functionalities as required. This allows you to mix different event types to implement various bypassing scenarios. For examples, refer to Example Sequences of DAQ and STIM Functionalities on page 38.

The following table shows the different options for bypassing with the DCI-GSI2 and some of their characteristics:

	Bypassing Without Changes to ECU Software	**Bypassing with Changes to ECU Software**
With Data Consistency	*Using the fast variable overwrite event mechanism* ▪ Requires access to the ECU data trace interface. ▪ There are certain timing restrictions.	*dSPACE Calibration and Bypassing Service is integrated in ECU software.* ▪ Several thousands of bypass events can be integrated based on a single hardware event. ▪ Up to 255 events can be used in parallel for reading and writing ECU variables.

	Bypassing Without Changes to ECU Software	Bypassing with Changes to ECU Software
	▪ Works only with a limited amount of write variables. For further information, refer to Bypassing Without ECU Service on page 35.	▪ Extended bypassing mechanisms (double buffer mechanism, wait mechanism, failure checking mechanism) can be used. ▪ Integrated bypass status detection in ECU software Refer to Bypassing with ECU Service on page 34.
Without Data Consistency	There are two basic event sources: *Using DCI-GSI2-internal timer* ▪ Asynchronously to ECU, oversampling is used. ▪ Data is written very fast and periodically to the ECU. *Using available hardware events (e.g., watchpoint pin, polling event)* ▪ Asynchronously with ECU tasks ▪ Number of available events depends on the ECU and DCI-GSI2 hardware. ▪ ECU timing needs to be known.	*dSPACE code patch is integrated in ECU code.* ▪ Several thousands of bypass events can be integrated based on a single hardware event. ▪ Up to 255 events can be used in parallel for reading and writing ECU variables. ▪ Minimal memory consumption and run time overhead. Refer to Bypassing Without ECU Service on page 35.

Note

To guarantee data consistency in bypassing scenarios, use the following options:

- For DAQ: Either use the data trace mechanism or (if using the data trace mechanism is not possible for any reason) integrate an ECU service.
- For STIM: Either use the fast variable overwrite mechanism (but not in connection with service-based STIM) or integrate an ECU service.

Replacing dSPACE PODs

If you use the DCI-GSI2 for bypassing, the DCI-GSI2 can replace an existing dSPACE plug-on device (POD) for the serial debug interface. For further information, refer to Hardware Adaptation on page 61.

Calibration and bypassing in parallel

Since the DCI-GSI2 supports multiple XCP service instances, you can perform calibration and bypassing in parallel. Simultaneous accesses by the host PC (for calibration and measurement) and by the bypass system are possible.

Tip

If you want to perform service-based measurement and bypassing in parallel, you are recommended to use dSPACE Calibration and Bypassing Service 2.3.0 or later.

Switching the ECU on or off during bypassing	Using the DCI-GSI2, it is possible to switch the connected ECU on or off even if the bypass model is running. After the ECU is rebooted, the bypass algorithm restarts automatically.

Related topics

Basics

Examples

Bypassing with ECU Service

Introduction

The DCI-GSI2 provides service-based bypassing.

Service-based bypassing

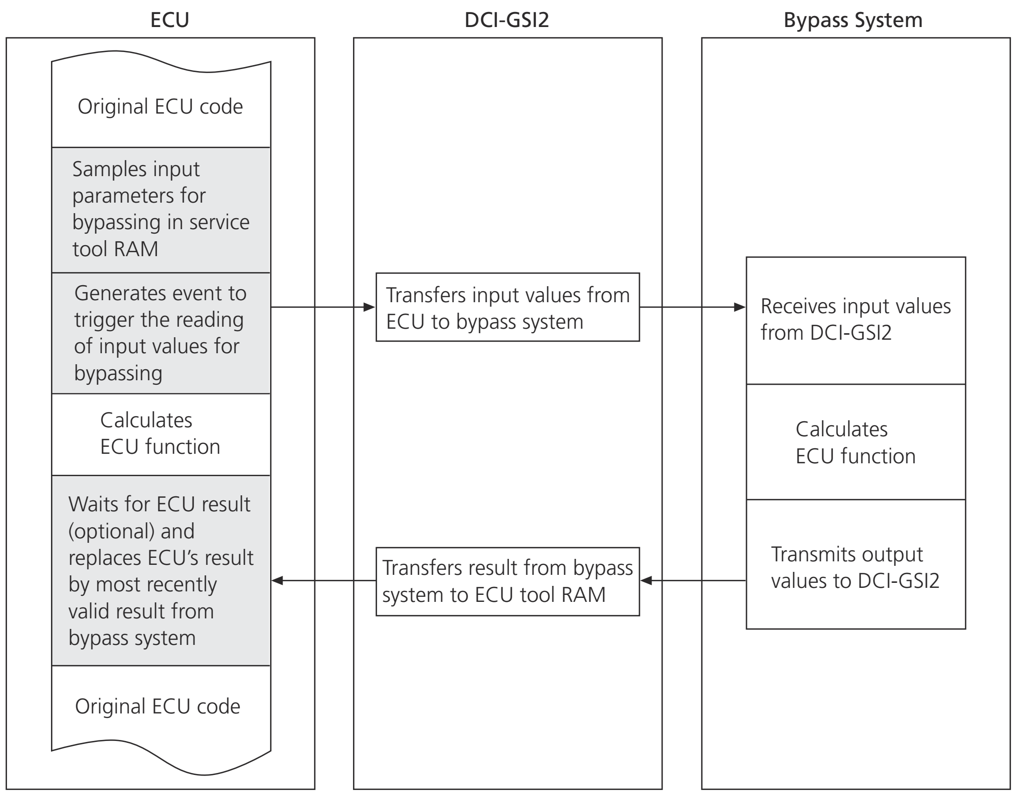
For service-based bypassing with the DCI-GSI2, the ECU code must be implemented with the dSPACE Calibration and Bypassing Service. For details on implementing and configuring the service, refer to the dSPACE Calibration and Bypassing Service Implementation 🕮 document.

When the service is called in the ECU application, all the data to be measured is copied from the ECU memory into the service tool RAM. After that, an event is generated that is used by the DCI-GSI2 to trigger the reading of the data from the service tool RAM. This ensures that a consistent set of data is measured. The measurement data is then transmitted to the RCP system.

When the service is called in the ECU application, all the data to be stimulated is copied from the ECU tool RAM to the ECU memory. The data must have been received by the DCI-GSI2 from the RCP system and instantly copied to the ECU tool RAM beforehand.

If the double buffer mechanism is enabled, the stimulated data set is always consistent.

The illustration below shows the data exchange between the ECU and the bypass system via the DCI-GSI2 if service-based bypassing is performed:



Related topics

Basics

Bypassing Without ECU Service

Introduction

The DCI-GSI2 supports bypassing without an ECU software service, which means serviceless and code-patch-based bypassing.

Serviceless bypassing

For serviceless bypassing, no ECU service needs to be integrated into the ECU code, and no ECU code modifications are required. The triggering of the bypass function on the RCP system can be implemented by means of various event trigger mechanisms supported by the DCI-GSI2. Refer to Overview of the DCI-GSI2 Event Types on page 54.

The following paragraphs describe the basic concepts of some STIM functionalities. You can combine the STIM functionalities with different DAQ functionalities as required. For examples, refer to Example Sequences of DAQ and STIM Functionalities on page 38.

Principle of STIM using data trace write event The DCI-GSI2 supports data stimulation by means of a data trace write event (see Data Trace Write Event on page 55). As soon as a data trace write event is detected by the DCI-GSI2, all the variables belonging to this event are written into the ECU memory. The variable values provided by the RCP system are used for this, i.e., all the variables belonging to this event are overwritten with the appropriate values calculated by the RCP system.

The values to be watched for event generation and the values to be written are separate.

STIM using a data trace write event involves the following steps:

- The results of the bypassed function(s) are transferred from the RCP system to the DCI-GSI2, where they are stored in its data buffer.
- The DCI-GSI2 watches for data trace write events by watching the corresponding variables (see Data Trace Write Event on page 55).
- The ECU application writes to a watched variable.
- The DCI-GSI2 detects this write operation and associates a data trace write event to it.
- The DCI-GSI2 overwrites all the variables belonging to this event with the appropriate variable values previously provided by the RCP system.

Note

DAQ and STIM can be combined within the same data trace write event. In this case, data acquisition is processed before the STIM data is written to the ECU memory.
For information on DAQ using the data trace mechanism, refer to Measurement via Data Trace Interface on page 28.

Principle of STIM using fast variable overwrite The DCI-GSI2 supports data stimulation by means of a fast variable overwrite event (see Fast Variable Overwrite Event on page 57). The fast variable overwrite event mechanism allows single variables to be overwritten directly after the ECU application has written them. As soon as a fast variable overwrite event is detected by the DCI-GSI2, the variable belonging to the event is written to the ECU memory. The variable value last provided by the RCP system is used for this, i.e., the variable belonging to the event is overwritten with the appropriate variable value calculated by the RCP system.

STIM using the fast variable overwrite mechanism therefore involves the following steps:

- The results of the bypassed function(s) are transferred from the RCP system to the DCI-GSI2, where they are stored in an internal data buffer. The RCP system uses a special XCP event channel (fast variable overwrite event) for data transfer for this.

- The DCI-GSI2 watches all the variables belonging to the fast variable overwrite event (see Fast Variable Overwrite Event on page 57).
- The ECU application writes a variable belonging to the fast variable overwrite event.
- The DCI-GSI2 detects the write operation and overwrites the variable with the appropriate variable value previously provided by the RCP system.

Note

Due to technical limitations, there is a short gap between detecting that the ECU application has written a variable and overwriting the variable. As a consequence, if a write access is followed by a read access to the same variable watched by the data trace write event, this might cause the problem that the read access is executed within the gap. The read access then delivers the original ECU value instead of the value provided by the RCP system. This can cause data inconsistencies for the context the variable is used in.

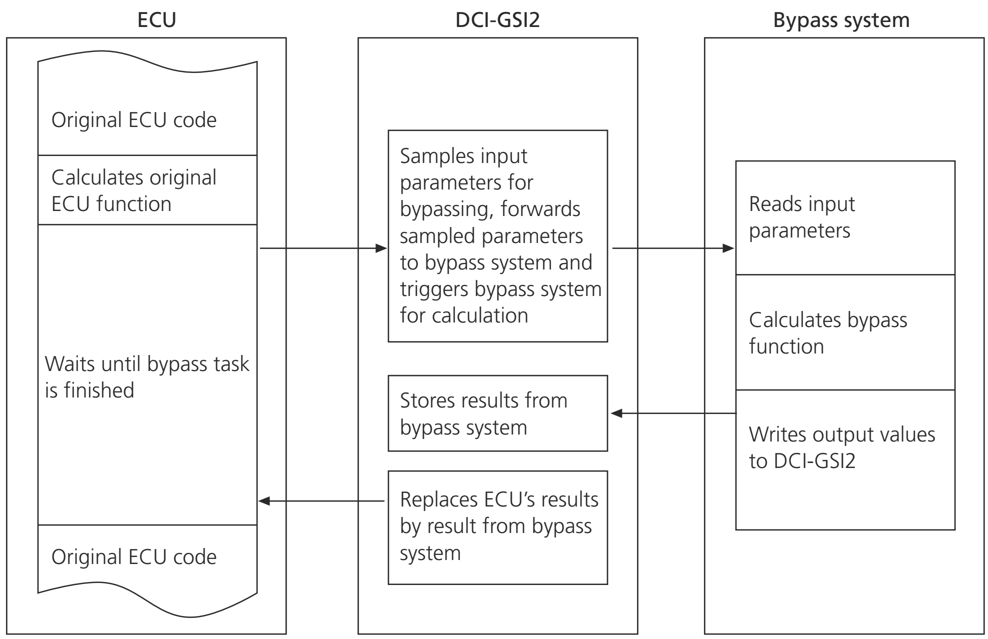
To avoid data inconsistencies, you should check whether a variable is suitable before using it in conjunction with the fast variable overwrite event. Ensure that no read access is executed directly after a write access to the variable.

Code-patch-based bypassing

A code patch can be integrated into the ECU software if the ECU does not have enough resources for the dSPACE Calibration and Bypassing Service or if data consistency is not required. The code patch mechanism uses a subinterrupt handling to provide several DAQ and/or STIM triggers based on a single hardware event trigger. The code patch bypassing mechanism is used when data consistency is not required or if other mechanisms ensuring data consistency are implemented. You can use code patches to generate several different trigger events at different locations of the ECU application. These events can be used by the DCI-GSI2 to measure and/or stimulate variables for implementing a bypass.

At the end of the ECU function to be bypassed, a code patch is integrated in the ECU code to trigger the RCP system. The ECU waits until the RCP system has finished the calculation of the bypass function. The bypass system gets the input values from the DCI-GSI2, calculates the bypass function, and writes the function results back to the DCI-GSI2. Then the DCI-GSI2 replaces the result of the original ECU function with the values provided by the RCP system. Afterwards, the code patch continues the execution of the original ECU code.

The following illustration shows a typical sequence for data exchange between the ECU and the bypass system via the DCI-GSI2 if code-patch-based bypassing is performed:



You can find an example of integrating a code patch into an ECU application in your dSPACE installation. For the demo application, refer to the `%ProgramData%\dSPACE\<InstallationGUID>\Demos\RTIBYPASS` folder.

You can access the `%ProgramData%\dSPACE\<InstallationGUID>` folder via a shortcut in the Windows **Start** menu below **dSPACE RCP and HIL <version>**.

Related topics

Basics

Bypassing Features.. 31
Overview of the DCI-GSI2 Event Types.. 54

Examples

Example Sequences of DAQ and STIM Functionalities.. 38

Example Sequences of DAQ and STIM Functionalities

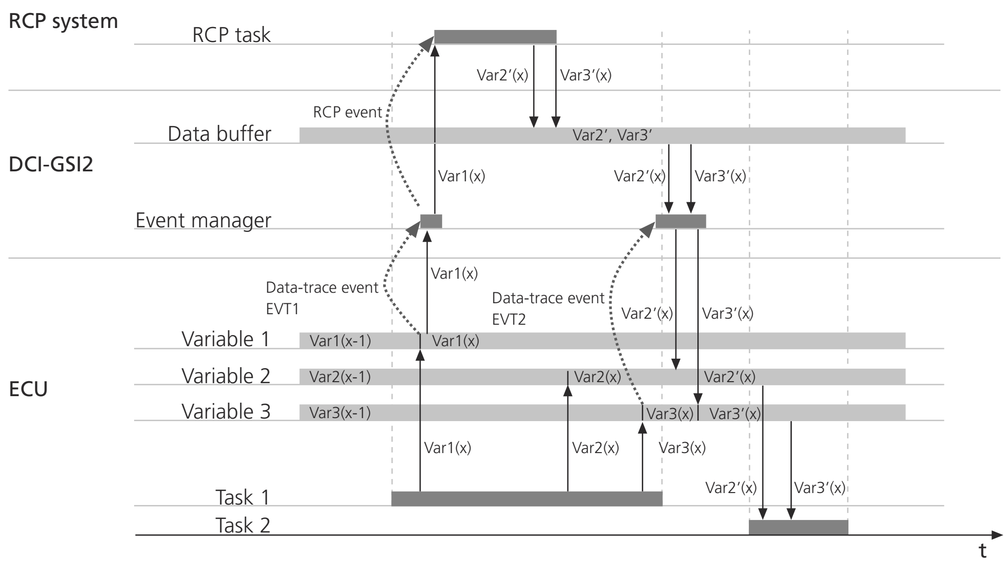
Introduction

The DCI-GSI2 provides several functionalities for DAQ and STIM purposes using different event types to trigger DAQ and STIM. You can combine the different DAQ and STIM functionalities as required. Below are some example sequences.

DAQ and STIM using data trace write event

The following illustration shows an example sequence for DAQ using data trace write event and STIM using data trace write event. It contains two ECU tasks and the variables Var1, Var2, Var3 accessed within the tasks. Task 1 writes the Var1 variable at the beginning and the Var2 and Var3 variables at the end of task execution. Task 2 reads the Var2 and Var3 variables provided by Task 1. The DCI-GSI2 is configured to trigger a data trace write event EVT1 when Var1 is written, and a data trace write event EVT2 when Var3 is written by the ECU application. The RCP system is configured to use EVT1 to sample and transfer Var1 to the RCP task (DAQ) and EVT2 to stimulate Var2 and Var3 with appropriate variable values.

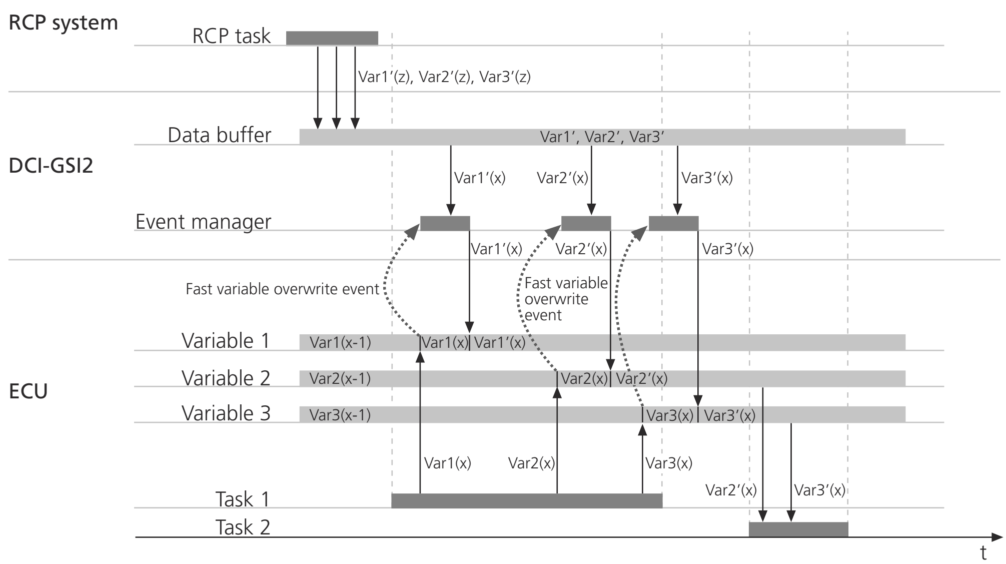
As a result, if the ECU application writes the Var1 variable, the EVT1 data trace write event is generated. This event triggers data acquisition of the Var1 variable and the transfer of the variable value via the DCI-GSI2 to the RCP system. It also triggers the RCP task to calculate new data for Var2 and Var3 using the value of Var1 as the input variable. The calculated values for Var2 and Var3 are transferred to an internal data buffer of the DCI-GSI2 (STIM). When the ECU application writes the Var3 variable, the EVT2 data trace write event is generated, triggering the writing of Var2 and Var3 with the results from the RCP system. As a result, the values of Var2 and Var3 previously provided by Task 1 are overwritten with the values provided by the RCP system.



STIM using fast variable overwrite

The following illustration shows an example sequence for STIM using the fast variable overwrite event mechanism. It contains two ECU tasks and the variables Var1, Var2, Var3 accessed within the tasks. Task 1 writes the Var1, Var2 and Var3 variables within the task execution. Task 2 reads the Var2 and Var3 variables provided by Task 1. The RCP system is configured to use the fast variable overwrite event mechanism to stimulate the Var1, Var2 and Var3 variables. The DCI-GSI2 fast variable overwrite event is configured by the RCP system to watch Var1, Var2 and Var3 for write accesses. The RCP system calculates new variable values Var1', Var2' and Var3'. This calculation can be executed asynchronously to the ECU application (for example, in a timer task of the bypass model), or synchronously triggered by another DCI-GSI2 event. The calculated values are transferred to an internal data buffer of the DCI-GSI2.

The DCI-GSI2 watches for the variables configured to be handled with the fast variable overwrite event. If the ECU application writes a watched variable, an event causes the DCI-GSI2 to overwrite the value of the variable with the corresponding value available in the internal data buffer. In this way, the Var1, Var2 and Var3 values are replaced with the appropriate values Var1', Var2' and Var3' provided by the RCP system each time the ECU application writes the Var1, Var2 and Var3 variables. This forces Task 2 to use the Var2' and Var3' values instead of the values provided by Task 1.



Related topics

Basics

ECU Flash Programming via the DCI-GSI2

Introduction

To update the ECU with new ECU code or data, the ECU's flash memory has to be programmed. This can be necessary, for example, to make new ECU variables available. The new program version can be flashed directly to the ECU via the DCI-GSI2.

ECU flash programming with the DCI-GSI2

The flash programming functionality is not integrated into the standard ECU application, but into a small separate application, called the *flash kernel*. When you request a flashing operation from within the flash tool (for example, the dSPACE ECU Flash Programming Tool), the flash kernel is downloaded from the host PC to the ECU RAM, where it is activated. The ECU then runs from the flash kernel, and the ECU's flash memory is ready to be programmed.

dSPACE flash kernel The dSPACE flash kernel consists of the dSPACE Calibration and Bypassing Service (more precisely, the service's flash programming functions) and the custom flash functions. It also contains the appropriate flash driver(s) for your ECU's flash memory.

> **Note**
>
> A flash kernel has to be adapted to the specific requirements of an ECU and its software. Contact dSPACE to discuss the next steps for creating a flash kernel for your ECU.

Boot check function in ECU code The DCI-GSI2 supports ECU flash programming with and without a boot check:

- To perform ECU flash programming *with* a boot check, a boot check function (small module which is part of the dSPACE Calibration and Bypassing Service) must be integrated into the ECU code. Whenever the ECU boots, the function checks whether there is a request for a flash operation. If there is, the dSPACE flash kernel is downloaded.
- For most processors, the DCI-GSI2 also supports ECU flash programming *without* a boot check. In this case, no boot check function must be integrated in the ECU code. ECU flash programming without a boot check allows you to program the flash memory of brain-dead ECUs, that is, ECUs with an empty flash memory or with corrupt flash memory content (brain-dead flashing).

For further information, refer to Principle of ECU Flash Programming with the DCI-GSI2 (ECU Flash Programming 📖).

dSPACE ECU Flash Programming Tool

If you need to program the flash memory of your ECU, you can use the dSPACE ECU Flash Programming Tool. The tool lets you program either a data set from ControlDesk, or a HEX, MOT, SREC or S19 file from File Explorer, to the ECU flash memory.

For detailed information on the dSPACE ECU Flash Programming Tool and how to work with it, refer to the ECU Flash Programming 📖 document.

Preconditions

Before ECU flash programming is started, the following conditions must be fulfilled:

- The ECU must be switched on.
- The hardware controlled by the ECU must be turned off.

Note

After programming the ECU flash memory, you should perform the following steps:

- Update the DCI-GSI2 configuration if you use data trace write events. The variables and their addresses might have changed with the modified ECU code.
- Add a new variable description (A2L file) to your associated device in ControlDesk. The locations of the variables might have changed with the modified ECU code.
- Import a new ECU Image file into your ControlDesk experiment. Data in the old ECU code usually does not correspond to the new ECU software.
- Replace the database file (A2L file) for the ECU in the RTI Bypass Blockset. The locations of the variables might have changed with the modified ECU code.

Related topics

Basics

Principle of ECU Flash Programming with the DCI-GSI2 (ECU Flash Programming 📖)
Working with the dSPACE ECU Flash Programming Tool (ECU Flash Programming 📖)

References

ECU Flashing Page (DCI Configuration 📖)

Debugging the ECU

Introduction

The interface between the ECU and DCI-GSI2 can be a debug interface like JTAG/Nexus. These interfaces are point-to-point connections: When the DCI-GSI2 is connected, the interface is not free for other tools. Some connector adapters therefore provide a tool connector for connecting an additional tool, for example, a debugger, to the DCI-GSI2. Access to the ECU interface is arbitrated between the DCI-GSI2 and the additional tool.

Tool connector

Some connector adapters have a tool connector for connecting an additional debugging tool to the DCI-GSI2. Thus, the DCI-GSI2 can be put in a tool chain with a host tool (for example, a calibration or bypassing system) and a debugging tool.

> **Note**
>
> When you perform ECU flash programming, especially during brain-dead flashing, you should disconnect the external debugger from the DCI-GSI2 to avoid malfunctions.

Interface arbitration

You can connect a debugging tool to the DCI-GSI2 for simultaneous or alternate calibration and debugging via connector adapters with a tool connector. However, this feature is not supported for all serial interfaces. The serial interface type of the ECU and the connector adapter of the DCI-GSI2 determine whether interface arbitration is possible and which arbitration mode (simultaneous or alternate) can be used.

Simultaneous calibration and debugging If the host tool and the debugging tool are both connected to the DCI-GSI2, calibration tasks and debugging tasks can be executed in parallel.

Alternate calibration and debugging Although the host tool and the debugging tool are both connected to the DCI-GSI2, you can use only one of these tools at a time. Calibration tasks and debugging tasks are executed alternately.

> **Note**
>
> Interface arbitration is not restricted to calibration and debugging. It also applies to measurement or bypassing tasks and to debugging tasks.

Contact dSPACE for information on which arbitration mode is supported by the serial interface type of your ECU.

DebugOverXCP protocol

Using the DebugOverXCP protocol, which was introduced with the XCP standard Ver. 1.5, you can perform ECU debugging without actually connecting a hardware debugger. With this protocol, you can tunnel debug accesses through the XCP interface of the DCI-GSI2. It is therefore no longer necessary to connect a special debugger hardware to the ECU to perform debugging, provided the debugger software supports the DebugOverXCP protocol and the protocol is supported for the specific ECU debug interface.

The DebugOverXCP protocol is a viable alternative to the hardware arbitration mechanism for most debugging scenarios. Due to the tunneling of debug accesses through the XCP protocol, the debug accesses are usually a bit slower than a hardware debugger. However, in most cases this is barely noticeable.

For more information on which debug interfaces support the DebugOverXCP protocol and which vendors offer compatible debugger software, contact dSPACE.

Related topics

References

Tool Arbitration Page (DCI Configuration 🕮)

Time Synchronization

Introduction

The DCI-GSI2 can synchronize its internal time base with other devices via standardized Ethernet protocols, and an accuracy of at least 1 microsecond if 1 Gbit/s Ethernet is used. A synchronized time base ensures that time stamps generated at the same time on separate devices have the same value (within the accuracy of the time synchronization mechanism). Another feature allows for generating DAQ/STIM events on separate DCI-GSI2 devices at the same time.

Protocols

The time synchronization is performed using the standardized IEEE802.1AS protocol. All devices that should be synchronized, as well as all Ethernet switches between those devices, have to support the protocol. The global time base is determined by the grandmaster clock, which is selected by using the best clock master algorithm based on a configurable clock priority.

For setups where an Ethernet switch that supports the IEEE802.1AS protocol is not available, the DCI-GSI2 devices that should be synchronized can be configured to use the 'direct mode'. This mode is less accurate than the standard-compliant mode, and the clock master and slaves have to be configured manually.

Synchronized event timers

The DCI-GSI2 has always featured event channels that are triggered in a configurable period of time by an internal timer. This timer is now synchronized to the global time base, which allows multiple DCI-GSI2 devices to generate timer events at the same time. In addition, a configurable time offset can be applied to adjust the trigger point.

Power Management

Introduction

The power of the DCI-GSI2 is supplied by the battery of the vehicle or an external source, for example, if the DCI-GSI2 is used on a test bench. To meet the conditions in a vehicle and on the test bench, the DCI-GSI2's power management supports the following features.

Power supply

In many cases, the DCI-GSI2 is connected to the vehicle power supply within the ECU via the internal power supply cable, which comes with the DCI-GSI2 enclosure for in-vehicle use. However, in some cases it might be necessary to connect an external power supply to the DCI-GSI2 by using the *PWR_CAB9 Power Supply Cable for DCI-GSI2 with Galvanic Isolation*, which must be ordered separately.

To use the external power supply, you must use the bridge on the internal power connector in the DCI-GSI2. For further information, refer to How to Connect a DCI-GSI2 to an ECU (ECU Interfaces Hardware Installation and Configuration 🕮).

The DCI-GSI2 power can be supplied optionally via the ECU interface connection. The ECU takes the vehicle battery voltage to the interface connector, so there is no additional load on the ECU power supply. Contact dSPACE for details and support.

Note

Keep in mind the recommendations for the power supply of the DCI-GSI2. Refer to Recommendations for Powering the DCI-GSI2 by the Vehicle Battery (ECU Interfaces Hardware Installation and Configuration 🕮).

Voltage range

The voltage range is 4.3 V … 30 V (38 V transient), which allows the DCI-GSI2 to be directly connected to a vehicle battery.

Power consumption

The DCI-GSI2's power consumption is 4 W typ. at room temperature, using 1 Gbit/s Ethernet.

Using 100 Mbit/s Ethernet instead of 1 Gbit/s saves about 0.7 W.

Standby mode

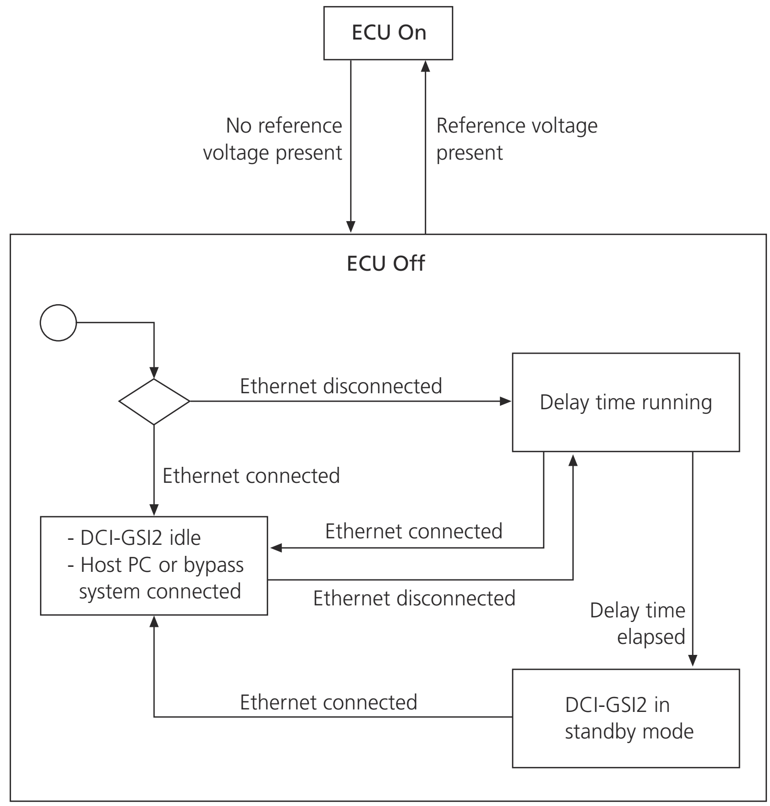
To save battery resources, the DCI-GSI2 switches to standby mode if the following conditions are all fulfilled at the same time:

- The ignition is turned off, the ECU is powered down and the ECU does not feed any reference voltage to the DCI-GSI2.
- The DCI-GSI2 is disconnected from the Ethernet.
- The user-defined standby delay time has elapsed.

In standby mode, the DCI-GSI2 has a power consumption of 30 mW typ.

Standby mode after a delay time In the DCI-GSI2 configuration, you can specify a time for the DCI-GSI2 to delay the standby mode. If no ECU reference voltage is detected by the DCI-GSI2 and no Ethernet link is active during this delay time, the DCI-GSI2 then goes into standby mode.

The illustration below shows the different states of the DCI-GSI2 when the ECU is off.



The DCI-GSI2 is woken up if one of the following conditions is fulfilled:

- The DCI-GSI2 is connected to the Ethernet.
- The ECU supplies the DCI-GSI2 with reference voltage, because the ECU power is turned on or the ignition is turned on.

Note

You can use all the DCI-GSI2's features as long as it is not in standby mode.

Protection against overvoltage	Transient overvoltage up to 38 V caused by load dump does not damage the DCI-GSI2.
Protection against reverse polarity	The DCI-GSI2 is protected against reverse battery voltages of up to -32 V.

Related topics	References Device Settings Page (DCI Configuration 📖)

Operational Reliability Features

Introduction

The following features ensure the operational reliability of the DCI-GSI2 and the connected ECU.

Consistency check of A2L and ECU Image file (EPK check)

If the ECU's A2L file and the ECU Image file contain `ADDR_EPK` and `EPK` attributes, ControlDesk automatically checks the consistency of the two files. The consistency is checked whenever you add an A2L file and an ECU Image file to the associated device.

For details, refer to Consistency Checks (EPK Checks) (ControlDesk Platform Management 📖).

Check on A2L and application (EPK check)

When online calibration is started, ControlDesk checks whether the experiment is consistent with the application on the ECU. The `EPK` string in the A2L file is compared with the `EPK` string read from the hardware. For further information, refer to Consistency Checks (EPK Checks) (ControlDesk Platform Management 📖).

Check on A2L and DCI-GSI2 configuration

When a connection to the DCI-GSI2 is established, ControlDesk checks whether the device configuration is consistent with the A2L file. For example, this ensures that the event channel settings in the DCI-GSI2 configuration are consistent with the XCP event channel configuration in the A2L file. The result of the configuration consistency check is displayed in the **Properties** controlbar of the DCI-GSI2 device. Refer to Common Properties (ControlDesk Platform Management 📖).

Watchdog functionality

The watchdog supervises the DCI-GSI2's firmware execution. If a firmware error occurs, the watchdog resets the DCI-GSI2.

Safety mode In the event of consecutive watchdog resets, for example, caused by hardware failures, the DCI-GSI2 is switched to safety mode. Calibration, data acquisition and bypassing are no longer possible in this mode.

Galvanic isolation of Ethernet signal lines	To prevent ground loops which may lead to erroneous data transmission, the signal lines of the DCI-GSI2's Ethernet interface are galvanically isolated. This prevents transient currents, which can occur especially when the engine is cranked. Thus, transient currents do not affect data transmission.
Power-down during firmware update	A break in the power supply during the programming procedure of the DCI-GSI2 flash memory damages neither the DCI-GSI2 nor the connected ECU.
Power failure behavior	A short power failure can be caused by an event such as cranking the engine or switching the air-conditioning on. The DCI-GSI2's power supply bypasses power failures of several milliseconds.
Related topics	References Connector Hardware Page (DCI Configuration 🕮)

Status Information on the ECU and DCI-GSI2

Introduction

The DCI-GSI2 supports system variables to get status information on the ECU and the DCI-GSI2.

Supported system variables

The following system variables supported by the DCI-GSI2 can be measured in ControlDesk to give feedback:

System Variable	Description
GSI2.CAL.ActivePage	ID number of the currently active page (0 = reference page, 1 … 255 = working page, >255 = neither working nor reference page)
GSI2.DAQ.BufferLevel	Level of the data acquisition buffer (in %)
GSI2.DAQ.MissedEventCount	Number of missed event triggers This system variable counts only the missed event triggers detected by the DCI-GSI2. The actual number of missed event triggers can be higher.
GSI2.DAQ.Overload	Information whether an overload of the ECU's DAQ processor occurred (0 = no overload, 1 = overload occurred, some events may have been lost)
GSI2.DAQ.TraceOverrunCount	Number of trace overruns that occurred
GSI2.DAQ.Workload	Load of the DAQ/STIM processor (in %)
GSI2.ECU.Ignition	Status information for ignition (KL15) (0 = voltage is absent, 1 = voltage is present)

System Variable	Description
GSI2.ECU.Ignition.Voltage[1)]	Voltage of the ECU ignition line (in V)
GSI2.ECU.InterfaceErrors	Number of detected transmission errors of the serial ECU interface
GSI2.ECU.VRef	Status of the ECU reference voltage VRef (0 = reference voltage VRef is absent, 1 = reference voltage VRef is present)
GSI2.ECU.VRef.Voltage[1)]	Level of the ECU reference voltage VRef (in V)
GSI2.ECU.VRef2	Status of the ECU reference voltage VRef2 (0 = reference voltage VRef2 is absent, 1 = reference voltage VRef2 is present)
GSI2.ECU.VRef2.Voltage[1)]	Level of the ECU reference voltage VRef2 (in V)
GSI2.ECU.Reset	Level of the ECU’s reset line (low active hardware signal) (0 = no reset, 1 = ECU is in reset)
GSI2.ECU.Reset.Voltage[1)]	Voltage of the ECU reset line (in V)
GSI2.ECU.VBat	Status of the battery power (0 = voltage is absent, 1 = voltage is present)
GSI2.ECU.VBat.Voltage[1)]	Voltage of the ECU VBat line (in V)
GSI2.ECU.VStandby	Status of the standby power (0 = voltage is absent, 1 = voltage is present)
GSI2.ECU.VStandby.Voltage[1)]	Voltage of the ECU VStandby line (in V)
GSI2.ETH.RxPacketCount	Number of Ethernet packets received by the DCI-GSI2
GSI2.ETH.RxPacketErrors	Number of invalid packets received by the DCI-GSI2 (e.g., due to transmission errors)
GSI2.ETH.TxPacketCount	Number of Ethernet packets transmitted by the DCI-GSI2
GSI2.EVENT.StimFailureCounter	Number of event-channel-specific STIM failures This system variable counts the failures of the event channel it is measured with. (0 = no STIM timeout occurred, 1 ... 253 = number of STIM timeouts that occurred, 254 = no STIM data received yet, 255 = configured failure limit is reached)
GSI2.SVC.MemoryAvail	Number of bytes still available in the service tool RAM
GSI2.Temperature	Temperature of the DCI-GSI2 hardware (in °C)
GSI2.Temperature.Fahrenheit	Temperature of the DCI-GSI2 hardware (in °F)
GSI2.TIMESYNC.Status	Current time synchronization status
GSI2.TIMESYNC.MasterOffset	Measured time offset to grandmaster clock (in ns)
GSI2.TIMESYNC.RateCorrection	Frequency rate correction of time synchronization mechanism (in ppm)

[1)] Availability depends on the connector adapter.

Data acquisition rasters for system variables

In principle, you can use any available data acquisition raster for measuring system variables. Some system variables (for example, the device temperature) are sampled at a fixed interval of 10 ms by default. You can measure them in a faster acquisition raster, if desired.

System variables can be measured even if the ECU is not running, provided that the assigned data acquisition raster is based on a DCI-GSI2's internal clock event and is used exclusively for measuring system variables.

Using the DCI-GSI2 system variables

System variables are described in the A2L file in the same way as normal ECU variables. System variables can be mapped into the ECU's address space by the DCI-GSI2. You can specify the addresses to be used for mapping in the DCI-GSI2 configuration. Refer to System Variables Page (DCI Configuration 📖).

As an alternative, you can use the XCP address extension 254 for the system variables.

Tip

If you want to measure system variables, but they are not in your variable description, copy them from another appropriate A2L file into your variable description, or use the A2L file adaptation functionality of the DCI Configuration Tool.

Related topics

Basics

Adapting or Generating an A2L File (DCI-GSI2 Setup Application Note 📖)

References

System Variables Page (DCI Configuration 📖)

Comparison of DCI-GSI2 and DCI-GSI1 Features

Introduction

The software support for the DCI-GSI1 was discontinued at the end of 2019. You are recommended to use the DCI-GSI2 instead.

Like the DCI-GSI1, the DCI-GSI2 is a serial interface that can be used for ECU calibration, measurement, function bypassing, and ECU flash programming purposes.

DCI-GSI2 versus DCI-GSI1

The following table provides an overview of the individual characteristics of the DCI-GSI2 and DCI-GSI1. It shows you what changes if you use the DCI-GSI2 instead of the DCI-GSI1.

Criteria	DCI-GSI2	DCI-GSI1
Harness complexity	Two connections in total (Ethernet, power supply)	Three connections in total (LVDS, USB, power supply)
Size of enclosure	▪ Enclosure for in-vehicle use (DCI_GSI2_ENC1): 102.5 x 71 x 26.5 mm (4.04 x 2.80 x 1.04 in.) ▪ Enclosure for laboratory use (DCI_GSI2_ENC2): 104 x 69 x 24 mm (4.09 x 2.72 x 0.94 in.)	102.5 x 71 x 21.4 mm (4.04 x 2.80 x 0.84 in.)
Temperature range	–40 … +85 °C (operating temperature) (–40 … +185 °F)	–40 … +85 °C (ambient temperature) (–40 … +185 °F)
Generic design	✓	✓
Open and standardized interfaces[1)]	✓	–
Host interface for measurement and calibration	XCP on Ethernet (UDP/IP or TCP/IP), 1 Gbit/s and 100 Mbit/s	USB 2.0[2)], proprietary protocol
Time synchronization with other devices	IEEE802.1AS	–
Interface to prototyping system for function bypassing	XCP on Ethernet (UDP/IP or TCP/IP), 1 Gbit/s and 100 Mbit/s	LVDS, proprietary protocol
Automatic detection of target microcontroller	✓	–
Performing calibration, measurement and bypassing in parallel	✓	✓
Maximum measurement data throughput	> 50 Mbit/s[3)]	< 6 Mbit/s (via USB interface)
CPU load on ECU for raster-synchronous measurement	▪ None (if data trace interface is used) ▪ Medium ... low (with ECU service)	Medium ... low (ECU service typically required)
Fastest measurement raster	≤ 100 kHz (for measurement rasters ≥ 10 µs)	≤ 10 kHz (for measurement rasters ≥ 100 µs)
Support of address remapping and memory management units (MMUs)	✓	Limited
Data tracing of ECU memory	✓ (≤ 400 kB)	Limited
Function bypassing latencies	Very low	Low
Support of working/reference page concept and calibration page switch	✓	✓
Support of microcontrollers prohibiting read access to ECU flash	✓	–

1) Compatibility with third-party tools via XCP.
2) Compatible with USB 1.1. dSPACE does not guarantee compatibility with USB 3.0.
3) Depending on the debug interface type.

Related topics

References

Technical Specifications of the DCI-GSI2 65

DCI-GSI2 Event Types

Introduction

Several types of events are available for working with the DCI-GSI2. These event types can be used for measurement, bypassing and/or page switching purposes.

Where to go from here

Information in this section

Overview of the DCI-GSI2 Event Types

Overview of event types

The following table gives an overview of the event types supported by the DCI-GSI2 for data acquisition, data stimulation and/or data page switching. The event types are explained in the following topics.

Event Type	DAQ	STIM	Page Switching
Data Trace Write Event	✓	✓	–
dSPACE Calibration and Bypassing Service	✓	✓	–
dSPACE Codepatch	✓	✓	–
Event Pin	✓	✓	✓
Fast Variable Overwrite Event	–	✓ [1)]	–
Instant STIM	–	✓ [2)]	–
Internal Timer	✓	✓	–
Polling Event	✓ [3)]	✓ [3)]	–
Watchpoint Message	✓	✓	–
Watchpoint Pin	✓	✓	✓
Watchpoint Polling	✓	✓	–

1) Do not use the fast variable overwrite event mechanism in connection with service-based STIM.
2) Instant STIM events are unsuitable for service-based bypassing scenarios.
3) Events detected with the polling event mechanism can have a significant time jitter. For this reason, polling events are unsuitable for high-end bypassing scenarios. However, the polling event mechanism is suitable for low-end bypassing scenarios.

Related topics

Basics

Data Trace Write Event

Data trace write event

For ECUs/microcontrollers that provide a data trace interface, the DCI-GSI2 supports the data trace write event. A data trace write event is generated on the DCI-GSI2 when a variable write access is transmitted to the DCI-GSI2 via the data trace interface, and this write access is configured as a data trace event trigger.

As soon as the ECU application writes to a variable in a specific ECU RAM area (data tracing memory area), the data trace write event is generated and indicates the change to the DCI-GSI2. The on-chip debug controller of the ECU's microcontroller autonomously transmits the changed data via the data trace interface to the DCI-GSI2's internal ECU RAM mirror.

To use variable write access by means of a data trace write event, you must specify the variable's address and data size.

Note

If you work with an ECU providing a data trace interface, data trace write events should be your first choice when configuring events for measurement and bypassing purposes.

Related topics

Basics

dSPACE Calibration and Bypassing Service

dSPACE Calibration and Bypassing Service

The dSPACE Calibration and Bypassing Service event type can be used if the dSPACE Calibration and Bypassing Service is implemented in the ECU code. It is not directly a hardware event.

When the service is called in the ECU application, a so-called subinterrupt is set and a hardware event is triggered. This allows many different events to be transmitted via one hardware event.

For information on service-based measurement and bypassing using the DCI-GSI2, refer to Measurement with ECU Service on page 24 and Bypassing Features on page 31.

For further information on the subinterrupt mechanism and the service functionality in general, refer to the dSPACE Calibration and Bypassing Service Implementation document.

Related topics

Basics

Overview of the DCI-GSI2 Event Types........ 54

dSPACE Codepatch

dSPACE Codepatch

To use the dSPACE Codepatch event type, no ECU service needs to be integrated into the ECU code. A code patch must be integrated in the ECU code instead.

The dSPACE Codepatch event type uses the so-called subinterrupt mechanism to generate and handle multiple subinterrupts via a single hardware interrupt line.

The dSPACE Codepatch event type is typically used for bypassing. It can also be used for measurement if only one hardware trigger is available, but several different events need to be transmitted to the DCI-GSI2.

Related topics

Basics

Overview of the DCI-GSI2 Event Types........ 54

Event Pin

Event pin

The fastest way to transmit an event from the ECU to the DCI-GSI2 is to use an event pin.

> **Note**
>
> The DCI-GSI2 supports up to four event pins, depending on the connector adapter used.

An event is generated each time a falling edge is detected on the event pin, i.e., when the signal changes from high to low for the event pin. Usually, these events are generated by a GPIO or event out pin.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Fast Variable Overwrite Event

Fast variable overwrite event

For ECUs/microcontrollers that provide a data trace interface, the DCI-GSI2 supports the fast variable overwrite event. The fast variable overwrite event is a special data trace write event (see Data Trace Write Event on page 55). It allows single variables to be overwritten directly after the ECU application has written them. The RCP system configures each single variable to be watched and overwritten.

The DCI-GSI2 watches each variable that is configured for handling by the fast variable overwrite mechanism. As soon as the ECU writes to a watched variable, a fast variable overwrite event is generated, triggering the DCI-GSI2 to overwrite the value of the variable with the appropriate value provided last by the RCP system.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Instant STIM

Instant STIM

An instant STIM event is not triggered by the ECU, but is generated when a complete DAQ list with STIM direction is received by the DCI-GSI2. As soon as all the related stimulation data is received, it is written to the ECU.

Using the instant STIM event can be useful in the following cases:

- The writing of the variable to be stimulated was disabled in the ECU application.
- After the bypassed function, there is no suitable event to trigger data stimulation.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Internal Timer

Internal timer

Internal timer event generation is based on the internal clock of the DCI-GSI2, i.e., this event source is used for DCI-GSI2-synchronous tasks. The raster of an internal timer event is specified by a time unit and a time cycle. The rasters can be between 10 µs (which is the DCI-GSI2's minimum time-based sample time) and several seconds.

The DCI-GSI2 internal clock can generate events with up to four different rasters concurrently.

When the time synchronization mechanism is active, it is possible to run event channels on several DCI-GSI2 devices that are synchronized with each other. They can either trigger events at the same time or have a configurable time offset.

Note

Events generated this way are completely asynchronous to the ECU tasks.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Polling Event

Polling event

To use the polling event type, your ECU's microcontroller must provide a special hardware-supported trigger register. To generate an event, the ECU application sets a bit in the register. The DCI-GSI2 monitors the trigger register for changes.

If the polling event is configured, the DCI-GSI2 continuously monitors the register. When it finds a set bit, it triggers the corresponding event.

If there is no other traffic on the serial ECU interface, the DCI-GSI2 polls the register as fast as possible. If there is traffic, the DCI-GSI2 polls the register at least once in a certain period of time. The polling event mechanism can cause time jitter depending on the load of the serial ECU interface.

When configuring a polling event, you must specify the number of the bit which triggers the event.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Watchpoint Message

Watchpoint message

Watchpoint messages can be generated whenever the ECU writes to a certain variable.

To configure a watchpoint message event, you must specify the address of the variable to be watched. When the ECU writes to this address, a special trace message is generated and transmitted to the DCI-GSI2 via the data trace interface.

Note

However, you should prefer data trace write events to the watchpoint message mechanism. Refer to Data Trace Write Event on page 55.

Data watchpoint messages are available only with some serial interfaces, for example, JTAG/Nexus.

Related topics

Basics

Overview of the DCI-GSI2 Event Types.. 54

Watchpoint Pin

Watchpoint pin

Unlike the watchpoint message (see Watchpoint Message on page 59), the watchpoint pin does not require a data trace interface. Whenever the ECU writes to a certain variable, a watchpoint pin can be triggered.

When you configure a watchpoint pin as the event source, you must specify the number of the event pin to be used and the address and address extension of the variable to be watched.

Tip

A watchpoint pin is also called an "event out pin", "breakout pin" or "EVTO".

Related topics

Basics

Overview of the DCI-GSI2 Event Types........ 54

Watchpoint Polling

Watchpoint polling

Watchpoint polling events can be generated whenever the ECU writes to a certain variable.

This is very similar to the polling event mechanism (see Polling Event on page 59). The difference is that the polling event bits do not have to be set explicitly, but can be set automatically whenever the ECU writes to a certain variable.

When you configure a watchpoint polling event as an event source, you must specify the address and address extension of the variable to be watched.

Compared with the watchpoint pin mechanism, watchpoint polling does not require a dedicated event pin and can support more than one watchpoint. However, watchpoint polling can incur the same small latency penalty as the polling event mechanism.

Related topics

Basics

Overview of the DCI-GSI2 Event Types........ 54

Adaptation to the ECU

Introduction

The DCI-GSI2 is of generic design, that is, the hardware does not need to be changed if it is used with different ECUs. To adapt the DCI-GSI2 to individual ECUs and connector adapters, dedicated firmware versions have to be downloaded onto the DCI-GSI2.

Where to go from here

Information in this section

Hardware Adaptation

Introduction

The DCI-GSI2 is designed to work with various ECU microcontroller systems. It can be connected to the ECU via a processor-specific connector adapter.

Connection to the ECU

The DCI-GSI2 provides two SAMTEC bus connectors, which interface your individual ECU connector via an appropriate connector adapter. A connector adapter mainly routes the bus signals from the ECU to the DCI-GSI2 and vice versa.

For details on which ECU microcontroller families of the different ECU interfaces are supported by the DCI-GSI2, refer to DCI-GSI2 Data Sheet on page 65.

The DCI-GSI2 can be connected to one ECU at a time. If multiple processors are to be calibrated, measured or bypassed, multiple DCI-GSI2s are necessary.

Replacing dSPACE PODs

There are dSPACE plug-on devices (PODs) that provide the physical connection between the ECU and the RCP system, and perform signal adaptation between the ECU and the bypass system via the serial debug interface. Bypassing with the DCI-GSI2 is also done via the debug interface. Hence, the DCI-GSI2 may be compatible with an existing dSPACE POD, or an existing dSPACE POD for the serial debug interface can be replaced by the DCI-GSI2. That is, if you use the DCI-GSI2 for bypassing, the DCI-GSI2 replaces the POD on the ECU, and the physical connection between the ECU and the RCP system is provided by the DCI-GSI2.

The POD-specific blockset must be be replaced. Replacing dSPACE PODs may also necessitate updating the RCP system software.

For information on whether your existing dSPACE POD can be replaced by the DCI-GSI2, contact dSPACE.

Related topics

Basics

Software Adaptation...62

Software Adaptation

Introduction

Depending on what you want to do with the DCI-GSI2, you have to adapt the code of the ECU. Adapting the ECU code means implementing an additional ECU service. An ECU service provides access to specific memory locations, for example.

Adapting the ECU code

Calibration with the DCI-GSI2 The DCI-GSI2 provides two calibration methods: ECU calibration via overlay units of the ECU processor and ECU calibration in external ECU RAM. For the former, no software adaptations are necessary. For calibration in external RAM, the dSPACE Calibration and Bypassing Service is required. More precisely, the `dsecu_custom_cal_page_switch` function is required for page switching and must therefore be integrated in the ECU code. For further information, refer to the dSPACE Calibration and Bypassing Service Implementation 🕮 document.

On some ECU microcontrollers, the overlay RAM control registers cannot be accessed via the serial interface. In these cases, the dSPACE Calibration and Bypassing Service can be used to provide access to the registers.

Measurement and bypassing with the DCI-GSI2 Using the DCI-GSI2, you can perform data acquisition and bypassing tasks on an ECU with or without an ECU service. For service-based DAQ and function bypassing, the dSPACE Calibration and Bypassing Service is required. For code-patch-based DAQ and bypassing, a code patch must be implemented in the ECU code. The code-patch mechanism can be used if several events need to be generated, but only one hardware event can be generated by the ECU.

For further information, refer to the DCI-GSI2 Setup Application Note 🕮.

Supported ECU service

The DCI-GSI2 supports the dSPACE Calibration and Bypassing Service. For information on the service, refer to the following documents:

- dSPACE Calibration and Bypassing Service Feature Reference 🕮
- dSPACE Calibration and Bypassing Service Implementation 🕮

You are recommended to use dSPACE Calibration and Bypassing Service 2.3.0 or later.

Requirements for the A2L file

To communicate with an ECU via ControlDesk's DCI-GSI2 or XCP on Ethernet device for calibration and measurement, or to bypass an ECU via DCI-GSI2 using the RTI Bypass Blockset, the ECU's A2L file must meet some requirements.

Required interface-specific information (IF_DATA) The A2L file must include an appropriate `IF_DATA` entry describing interface-specific information. This information must comply with a specific format, which is specified in the ASAM MCD-2 MC Metalanguage (AML). The AML specification for the interface must also be contained in the ECU's A2L file. If the AML specification is not contained in the A2L file, it can be defined in separate AML files which must then be included in the A2L file.

The DCI Configuration Tool provides mechanisms to simplify the handling of the A2L file. For further information, refer to the DCI-GSI2 Setup Application Note 🕮 and the DCI Configuration 🕮 document.

Note

If you change the configuration of your DCI-GSI2 (for example, when you define a new event channel), you must ensure the consistency of the DCI-GSI2 configuration and the A2L file.
As of ControlDesk 4.3, this consistency check is performed by ControlDesk for those A2L files that were adapted automatically by the DCI Configuration Tool.

For information on the interface-specific `IF_DATA` entries and details on their data formats, refer to Interface Description Data for dSPACE Communication Interfaces (DCIs) (Interface Description Data Reference 📖).

Related topics

Basics

DCI-GSI2 Data Sheet

Introduction

The DCI-GSI2 data sheet provides the technical data of the DCI-GSI2 and the dimension drawings.

Where to go from here

Information in this section

Technical Specifications of the DCI-GSI2

Technical data

The following table summarizes the technical specifications of the DCI-GSI2:

Parameter	Specification[1)]
General	▪ Microcontroller RAM of 128 MB ▪ Flash memory of 32 MB ▪ ECU RAM mirror of 400 kB ▪ Tool interface for connecting an additional tool (optional)
Host interface	▪ Ethernet (100 Mbit/s, 1 Gbit/s) ▪ Time synchronization: IEEE802.1AS protocol with 1 µs accuracy[2)]

Parameter		Specification[1]

ECU interface and ECU microcontroller families		▪ JTAG/OCDS:
		  ▪ Infineon TriCore
		▪ JTAG/Nexus:
		  ▪ Freescale/NXP MPC55xx
		  ▪ Freescale/NXP MPC56xx
		  ▪ Freescale/NXP MPC57xx
		  ▪ STMicroelectronics SPC56xx
		  ▪ STMicroelectronics SPC57xx
		  ▪ STMicroelectronics SPC58xx
		  ▪ Renesas RH850
		  ▪ Renesas V850E2
		▪ DAP/DAP2:
		  ▪ Infineon TriCore
		  ▪ Infineon XC2000
		▪ NBD:
		  ▪ Renesas RH850
		  ▪ Renesas V850E/V850E1
		  ▪ Renesas M32R
		  ▪ Renesas SH2/SH2A/SH4A
		▪ JTAG/H-UDI:
		  ▪ Renesas SH2A/SH4A
		▪ JTAG/ARM:
		  ▪ Texas Instruments TMS570
		  ▪ Cypress MB9DF
		  ▪ Toshiba TMPV770
		For further support, contact dSPACE.

Electrical characteristics	Power supply	▪ Range: 4.3 V … 30 V (38 V transient) ▪ Protected against reverse battery up to -32 V
	Power consumption	▪ 4 W typ. at room temperature and 1 Gbit/s Ethernet. Using 100 Mbit/s Ethernet instead of 1 Gbit/s saves about 0.7 W. ▪ 30 mW typ. in standby mode

Mechanical characteristics	Connectors	▪ Bus connectors:
		  ▪ Two connectors to interface individual connector adapters.
		  ▪ Various connector adapters are available from dSPACE. Some adapters also have a tool connector for connecting an additional tool, for example, a debugger.
		▪ Ethernet connector:
		  ▪ 8-pin connector for connection to the host PC and/or to a dSPACE real-time system[3]
		▪ Power connectors[4]:
		  ▪ 2-pin LEMO power connector on the DCI-GSI2 enclosure for connection to an external power supply via the PWR_CAB9 Power Supply Cable for DCI-GSI2 with Galvanic Isolation

Parameter		Specification 1)
		▪ Internal power connector within the DCI-GSI2 for connection to the vehicle power supply within the ECU via the internal power supply cable
	Enclosure	There are different variants of aluminum boxes (to be ordered separately): ▪ Enclosure for *in-vehicle use*. Refer to DCI-GSI2 with enclosure for use in a vehicle on page 69. ▪ Enclosure for *laboratory use*. Refer to DCI-GSI2 with enclosure for use in a laboratory on page 70. ▪ dSPACE also provides enclosures for *in-vehicle use* complying with the IP66 protection classification. Contact dSPACE for details.
	Physical size (with enclosure)	Refer to ▪ DCI-GSI2 with enclosure for use in a vehicle on page 69 ▪ DCI-GSI2 with enclosure for use in a laboratory on page 70
	Weight (without connector adapter)	▪ Approx. 220 g (0.49 lb.) with enclosure for in-vehicle use ▪ Approx. 174 g (0.06 lb.) with enclosure for laboratory use
Environmental	Operating temperature	–40 … +85 °C (–40 … +185 °F)
	Storage temperature	–65 … +125 °C (–85 … +257 °F)

1) Unless stated otherwise, the specifications are valid only if the dSPACE hardware is correctly powered, switched on, and ready for operation.

2) Accuracy of 1 µs only with 1 Gbit/s Ethernet.

3) For details, refer to Connecting an ECU with DCI-GSI2 for Simultaneous Calibration and ECU Interfacing (ECU Interfaces Hardware Installation and Configuration 📖).

4) Only one power connector can be used at the same time.

Related topics

Basics

Status LEDs.. 68

References

Certifications of the DCI-GSI2.. 71
DCI-GSI2 Dimension Drawings... 69

Status LEDs

Introduction

The DCI-GSI2 is equipped with three status LEDs.

Location

For the location of the status LEDs, refer to DCI-GSI2 Dimension Drawings on page 69.

LED description

The following table provides a description of the status LEDs:

LED	Status	Description
Ethernet	Off	Ethernet link not established
	Flashing (yellow)	Waiting for a response from a DHCP server
	Lit (yellow)	No response from the DHCP server; the DCI-GSI2's fallback IP address is used instead
	Lit (green)	Ethernet connection established
	Flashing sporadically (green)	Data is being sent or received via Ethernet
	Flashing sporadically (red)	Error in data transmission via Ethernet
	Lit (red)	Error in Ethernet connection
ECU	Off	ECU off
	Lit (red)	ECU in reset
	Lit (yellow)	ECU on, but currently no ECU connection established
	Lit (green)	ECU on and ECU connection established
	Flashing sporadically (green)	Data is being transmitted via the ECU interface (serial interface/data trace interface)
	Flashing sporadically (red)	Error in data transmission via ECU interface
Status	Off	No power
	Lit (green)	Power on, firmware is running
	Flashing (green)	Measurement or ECU interfacing is running
	Flashing quickly (green)	Firmware update is running
	Flashing quickly (yellow)	The 'identify device' action has been triggered by the host tool.
	Flashing (red)	Firmware error

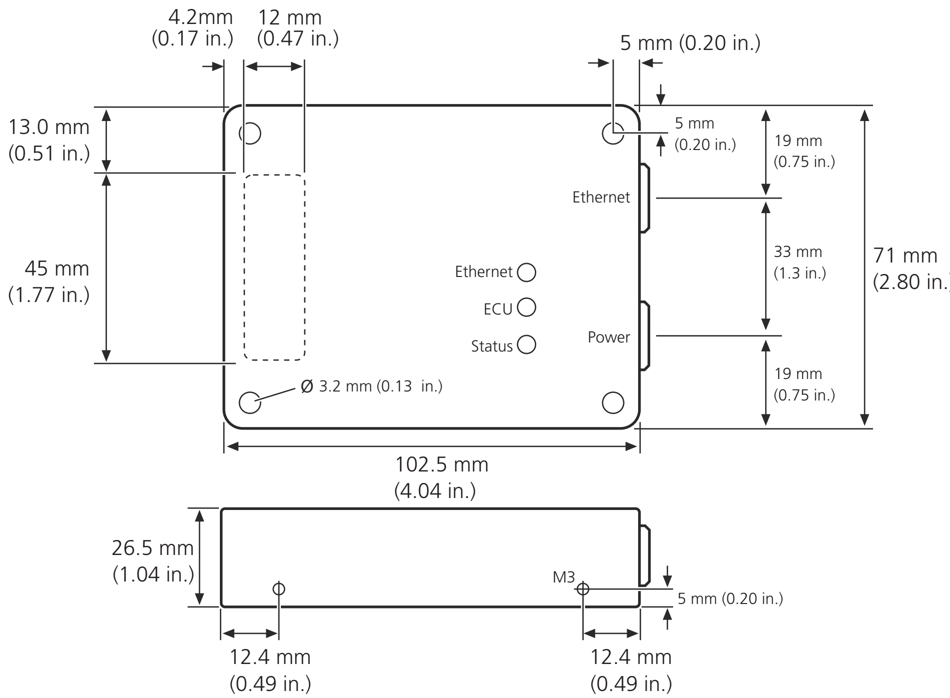
Related topics

References

DCI-GSI2 Dimension Drawings

DCI-GSI2 with enclosure for use in a vehicle

The following illustration shows the physical dimensions of a DCI-GSI2's enclosure for in-vehicle use (DCI_GSI2_ENC1):



The DCI-GSI2's enclosure for in-vehicle use has holes for mounting the DCI-GSI2 on an ECU.

Connector adapter A connector adapter for attaching the DCI-GSI2 to the ECU is integrated in the DCI-GSI2's enclosure. The enclosure also contains two LEMO connectors for the physical connection of the host PC and/or the RCP system and an external power supply to the DCI-GSI2.

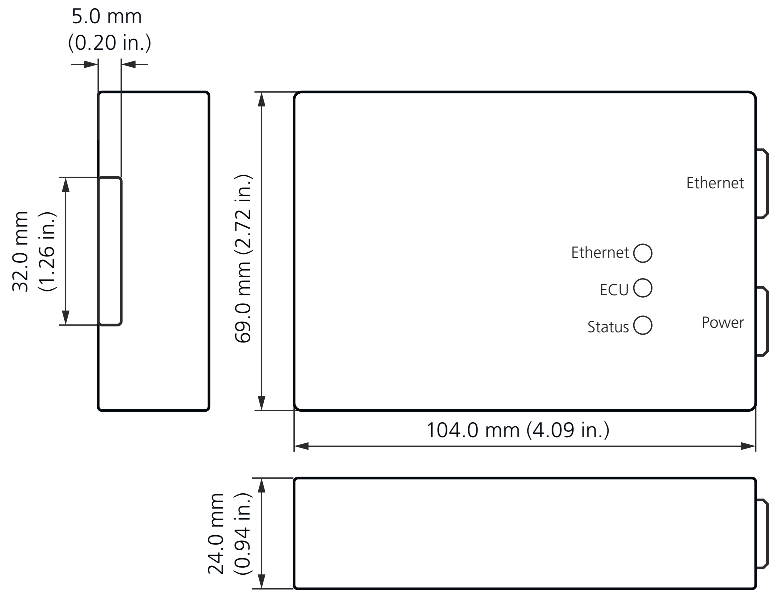
Power supply If the DCI-GSI2's enclosure for in-vehicle use is used, the DCI-GSI2 can alternatively have an internal power supply via a separate power supply cable (also provided by dSPACE). When this cable is used, the power of the DCI-GSI2 is supplied directly by the ECU, and the external power supply LEMO connector in the DCI-GSI2's enclosure is inactive.

Note

- The enclosure is not part of the DCI-GSI2, and has to be ordered separately.
- You can also order a custom enclosure for your DCI-GSI2.

DCI-GSI2 with enclosure for use in a laboratory

The following illustration shows the physical dimensions of a DCI-GSI2's enclosure for laboratory use (DCI_GSI2_ENC2):



The DCI-GSI2's enclosure for laboratory use does not have holes for mounting on an ECU.

Connector adapter A connector adapter for attaching the DCI-GSI2 to the ECU is integrated in the enclosure. The enclosure also contains two LEMO connectors for the physical connection of the host PC and/or the RCP system and an external power supply to the DCI-GSI2.

Note

- The enclosure is not part of the DCI-GSI2, and has to be ordered separately.
- You can also order a custom enclosure for your DCI-GSI2.

Related topics

Basics

Status LEDs........ 68

References

Technical Specifications of the DCI-GSI2........ 65

Certifications of the DCI-GSI2

CE compliance

The DCI-GSI2 meets the requirements of European Directive 2014/30/EU (Electromagnetic Compatibility Directive) for CE marking.

Vibration and shock tests

To verify the reliability of DCI-GSI2 under realistic operating conditions, it was exposed to vibration and shock tests. During the tests, DCI-GSI2 executed a program without any failures.

Applied standards

The characteristics of DCI-GSI2 were tested according to the standards shown in the following table:

Tested Characteristics	Applied Standard	Description
Electromagnetic compatibility (EMC)	EN 61326-1 Table 2	Immunity standard for industrial environments[1)]
	CISPR 11, EN 55011 Group 1, Class A	Emission standard for industrial environments
Vibration	ISO 16750-3:2007 / 4.1.2.4 Test IV	Test conditions: Broad band noise, 4h per axis, RMS-acceleration 27,8 m/s^2
	DO-160F.8 / B1 Test conditions	Test conditions: Broad band noise, 4h per axis, based on DO160F Section 8, Category B1
	ISO 16750-3:2007 / 4.1.2.7 Test VII	Test conditions: Broad band noise, 2h per axis, RMS-acceleration 61,7 m/s^2
	DO-160F.8 / C Test conditions	Test conditions: Broad band noise, 2h per axis, based on DO160F Section 8, Category C
	EN 60068-2-6	Test conditions: ▪ Swept sine, 1 octave per minute, 3-axis test ▪ 5 ... 2000 Hz, up to 5 g, 2 sweeps per axis ▪ Operating
Shock	ISO 16750-3:2007 / 4.2.2.	▪ Linear shock (1/2 sine pulse), 6-axis ▪ 500 m/s^2, 6 ms, 10 pulses per axis ▪ Operating
	RTCA / DO-160F Section 7 Test 7.2 Category A Test type R	Operational shocks test (standard): ▪ Saw-tooth wave, 6-axis ▪ 200 m/s^2, 11 ms, 10 pulses per axis ▪ Operating
	RTCA / DO-160F Section 7 Test 7.2 Category D Test type R	Operational shocks test (low frequency): ▪ Saw-tooth wave, 6-axis ▪ 200 m/s^2, 20 ms, 10 pulses per axis ▪ Operating

[1)] Tested with an ECU I/O cable length < 3 m. Connected cables might affect the specified characteristics due to physical effects such as crosstalk, voltage drops, and influences through electromagnetic fields.

Related topics

References

Technical Specifications of the DCI-GSI2 65