# ibaPDA-Request-HPCI

## Request Data Interface to HPCi Systems

Manual

Issue 2.2

Measurement Systems for Industry and Energy

www.iba-ag.com

**Manufacturer**

iba AG

Koenigswarterstrasse 44

90762 Fuerth

Germany

**Contacts**

| | |
|---|---|
| Main office | +49 911 97282-0 |
| Support | +49 911 97282-14 |
| Engineering | +49 911 97282-13 |
| E-mail | iba@iba-ag.com |
| Web | www.iba-ag.com |




| Version | Date | Revision | Author | Version SW |
|---|---|---|---|---|
| 2.2 | 07-2025 | Supporting ibaFOB-R and DGM200oE | cv/rm | 8.8.1 |

# Contents


**1 About this documentation ..... 5**

1.1 Target group and previous knowledge ..... 5

1.2 Notations ..... 6

1.3 Used symbols ..... 7

**2 System requirements ..... 8**

**3 ibaPDA-Request-HPCI ..... 10**

3.1 Connection between HPCi and ibaPDA ..... 10

**4 Configuration and engineering HPCi systems ..... 13**

4.1 DASAGNT ..... 13

4.1.1 Changing Network Interface for Control Path ..... 13

4.1.2 Changing Multicast Address ..... 15

4.1.3 Changing the Default Overload Protection Limits ..... 16

4.2 Hardware Definition ..... 17

4.2.1 CC100/DGM200 ..... 17

4.2.2 SM128V ..... 20

4.2.3 Reflective Memory ..... 21

4.2.4 ibaFOB-R ..... 23

4.2.5 ibaLink-VME board in P2P mode ..... 25

4.3 Generating address books with the DAS address book generator ..... 28

4.4 Useful P80i Functions ..... 30

4.4.1 Script Erase ..... 30

4.4.2 FTP Update ..... 30

**5 Configuration and engineering ibaPDA ..... 31**

5.1 General interface settings ..... 31

5.1.1 Diagnostics ..... 33

5.2 HPCi Request Module ..... 35

5.2.1 General module settings ..... 36

5.3 Signal configuration ..... 37

5.3.1 HPCi signal browser ..... 37

5.3.2 Drag and drop with P80i ..... 39

5.4 Request process ..... 40
5.5 HPCi data modules ..... 41
5.6 HPCi Lite ..... 43
**6 Diagnostics ..... 44**
6.1 Checking the license ..... 44
6.2 Log files ..... 44
6.3 Connection diagnostics with PING ..... 45
6.4 Module diagnostics ..... 45
6.5 Check the Historical Logger ..... 46
**7 Support and contact ..... 49**

# 1 About this documentation

This documentation describes the function and application of the software interface *ibaPDA-Request-HPCI*.

**Other documentation**

This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* can be found in the *ibaPDA* manual or in the online help.

## 1.1 Target group and previous knowledge

This documentation is aimed at qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing safety and recognizing possible consequences and risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons who are concerned with the configuration, test, commissioning or maintenance of Programmable Logic Controllers of the supported products. For the handling *ibaPDA-Request-HPCI* the following basic knowledge is required and/or useful:

- Basic knowledge of *ibaPDA*
- Basic knowledge of network technology
- Knowledge of configuration and operation of the relevant control system

## 1.2 Notations

In this manual, the following notations are used:

| Action | Notation |
|---|---|
| Menu command | Menu *Logic diagram* |
| Calling the menu command | *Step 1 – Step 2 – Step 3 – Step x*<br>Example:<br>Select the menu *Logic diagram – Add – New function block*. |
| Keys | <Key name><br>Example: <Alt>; <F1> |
| Press the keys simultaneously | <Key name> + <Key name><br>Example: <Alt> + <Ctrl> |
| Buttons | <Key name><br>Example: <OK>; <Cancel> |
| Filenames, paths | `Filename`, `Path`<br>Example: `Test.docx` |

## 1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

**Danger!**

**The non-observance of this safety information may result in an imminent risk of death or severe injury:**

- Observe the specified measures.

**Warning!**

**The non-observance of this safety information may result in a potential risk of death or severe injury!**

- Observe the specified measures.

**Caution!**

**The non-observance of this safety information may result in a potential risk of injury or material damage!**

- Observe the specified measures

**Note**

A note specifies special requirements or actions to be observed.

**Tip**

Tip or example as a helpful note or insider tip to make the work a little bit easier.

**Other documentation**

Reference to additional documentation or further reading.

# 2 System requirements

- *ibaPDA* v8.8.1 or higher (for ibaFOB-R support)
- Additional license for *ibaPDA-Request-HPCI*
- If Reflective Memory is used as a data path:
  - Additional license *ibaPDA-Interface-Reflective-Memory*
  - Fiber optic card of the Abaco's 5576 or 5565 Reflective Memory family (such as PCIE-5565PIORC) or *ibaFOB-R* in the *ibaPDA* computer
- If DGM200 is used as a data path:
  - Additional license *ibaPDA-Interface-HPCI-DGM200P* + interface card DGM 200-P or
  - Additional license *ibaPDA-Interface-HPCI-DGM200E* + communication adapter DGM 200-E
- HPCi v3.3.x or higher
  - HPCi v4.5.3 or higher for ibaFOB-R support in HPCi controller
- HPC-HWC-Tool v2.46
- “DASAGNT”-HPCi-driver v1.1.0 or higher

**Licenses**

| Order no. | Product name | Description |
|---|---|---|
| 31.001300 | ibaPDA-Request-HPCI | Extension license for an ibaPDA system to be able to use the request functionality with HPCi systems |
| 31.001009 | ibaPDA-Interface-HPCI-DGM200E | Extension license for an ibaPDA system for a DGM200E interface via DGM 200-E communication adapter<br>No. of connections: max. 4 DGM 200 networks with up to 20 controllers each |
| 31.001010 | ibaPDA-Interface-HPCI-DGM200P | Extension license for an ibaPDA system for a DGM200P interface via DGM 200-P PCI card |
| 31.001220 | ibaPDA-Interface-ReflectiveMemory | Extension license for an ibaPDA system for a reflective memory interface |

Table 1: Available licenses

**Hardware**

| Order no. | Product name | Description |
|---|---|---|
| 11.112620 | ibaFOB-R | Bidirectional PCIe, SFP, RFM Network Card - Fiber optic board for the reflective memory network |

Table 2: Hardware

The Reflective Memory card *ibaFOB-R* is manufactured by iba AG.

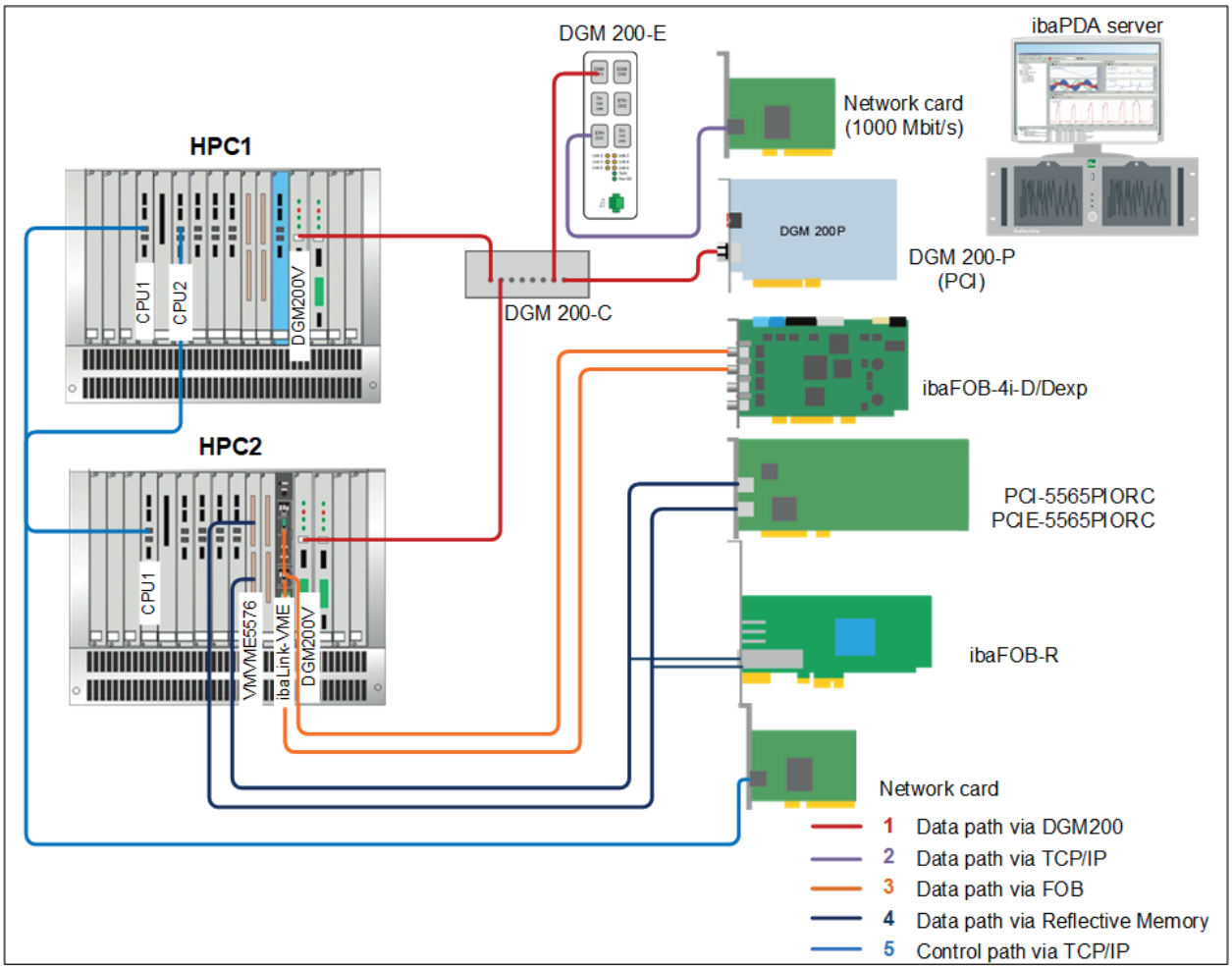
A PCI card DGM 200-P or a communication board DGM 200-E should be obtained directly from GE Energy Power Conversion.

# 3 ibaPDA-Request-HPCI

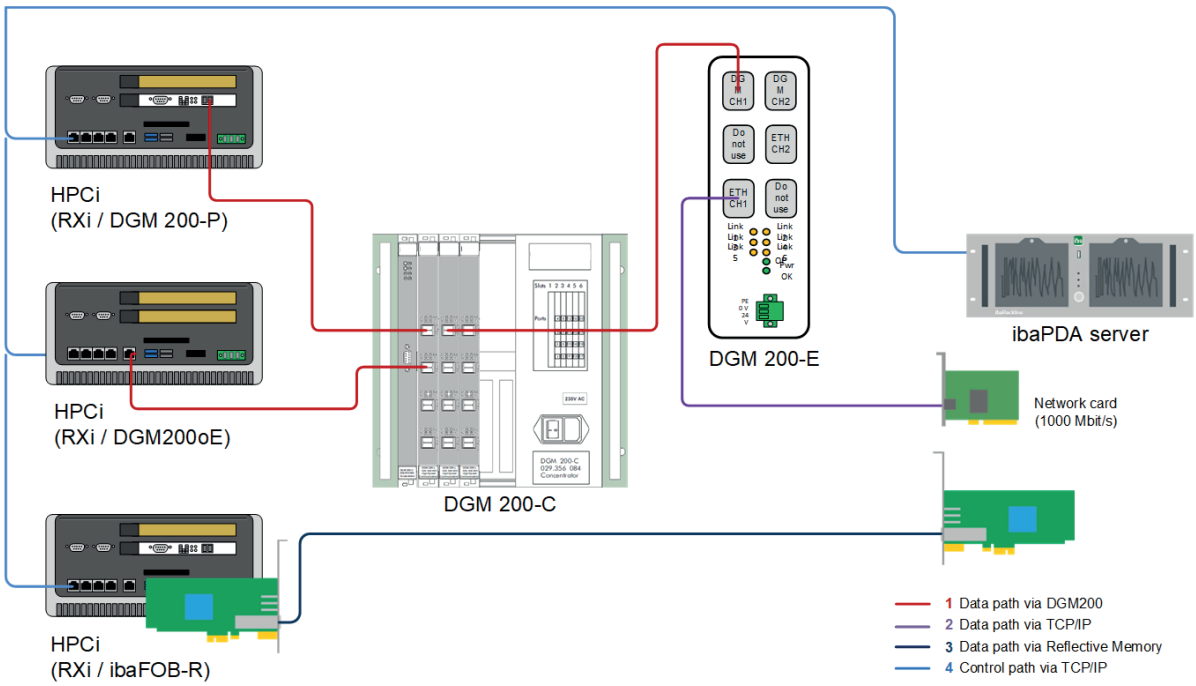
The Request interface *ibaPDA-Request-HPCI*is suitable for the measuring data acquisition with a free symbol selection from HPCi automation systems of GE Energy Power Conversion via different interfaces. The Request interface provides *ibaPDA* with symbolic access to all signals defined in the HPCi system. The user can change the list of signals he wants to measure without having to change anything in the HPCi system.

## 3.1 Connection between HPCi and ibaPDA

An HPCi system consists of one or more controllers. Each controller is a VME-rack or a PC based controller. A VME-based controller can contain up to 4 CPUs. Every CPU has an Ethernet connection. PC based controllers (APC620/810 or RXi042/142) use their Ethernet network adapter respectively.




*Possible connections between HPCi system and ibaPDA, VME-based controller*




*Possible connections between HPCi system and ibaPDA, RXi-based controller*

Currently supported interfaces:

- CC100 system (based on DGM 200 hardware)
- iba SM128V boards and ibaLink-VME board in SM128 compatibility mode
- Reflective Memory 5576 and 5565 boards of GE/ABACO
- ibaFOB-R as 5565-compatible Reflective Memory PCIe board
- ibaLink-VME in P2P mode D

An agent called DASAGNT (data acquisition system agent) needs to be loaded on every CPU or PC respectively. *ibaPDA* communicates with these agents via TCP/IP. The agents are responsible for cyclically sending the requested signals to *ibaPDA*. They can use different data interfaces to transmit the data.

The agents announce their presence via IP multicast. Every 10 seconds they send a status message to a predefined multicast group. This message contains amongst others the name of the CPU, its IP-address and the data interfaces that are available. *ibaPDA* joins the multicast group and listens for these status messages. When *ibaPDA* receives a multicast status message it establishes a TCP connection to the agent. This connection is called the *control path*. The agent will now send the status messages via the TCP connection instead of via multicast. *ibaPDA* responds to the status messages with another status message. This exchange of status messages acts as a watchdog. If *ibaPDA* or the agent doesn't receive a status message every 10 seconds then the connection is closed.

Once the control path is established *ibaPDA* tries to establish the *data path*. It tries to find the data interfaces in the PC that corresponds with the data interfaces the agent has. *ibaPDA* sends a data path discovery message to the agent. The agent then writes a certain pattern onto the data interface. *ibaPDA* then tries to find that pattern on the boards in the PC. This process is

repeated for all data interfaces reported by the agent. This system of automatic discovery of the data path makes the system a lot easier to use because the user doesn't have to configure the data path on the PC.

The user can browse a list of all HPCi signals in *ibaPDA*. He can make a selection of all the signals he wants to measure. He can also decide how fast he wants to measure each signal. There are 4 time classes available (default: 1 ms, 5 ms, 10 ms and 100 ms). When the user starts the measurement *ibaPDA* will send the list of signals via the control path to the agent. The agent will check if all signals are available. He will also check if the sending of the data will not overload the CPU.

See also chapter ↗ *Changing the Default Overload Protection Limits*, page 16

If everything is ok then the agent will start sending the data via the data path to *ibaPDA*.

# 4 Configuration and engineering HPCi systems

## 4.1 DASAGNT

DASAGNT is an HPCi-driver written by GE Energy Power Conversion. It is automatically selected if one of the following boards is inserted into the hardware configuration:

- ibaLink-SM128-V
- ibaLink-VME
- VME 5565/5576 Reflective Memory
- PCI-/PCIE-5565PIORC, PCI-5576 Reflective Memory or ibaFOB-R
- DGM 200-V
- DGM 200-P

### 4.1.1 Changing Network Interface for Control Path

By default the primary network interface of the HPCi CPU is used for TCP/IP communication with the *ibaPDA* PC.

`DASAGNT0.INI` can be adapted to use a specific network interface. The steps below describe how to select the network interface which is used for the control path.

1. Open the file `DASAGNT0.INI`, which is located at `P80_projectname.CTRL\Advanced\Configuration\`
2. Select a specific network interface by adapting the variable *ETHIF*

   fei0 -> primary network interface

   fei1 -> secondary network interface

   For HPCi controllers with a Gigabit Ethernet controller such as the VP325 card, the primary and secondary network interfaces are called **gei0** and **gei1**.

   For more recent HPCi controllers such as RXi-042/142, the network interfaces are numbered from 0 to 4. So, ETHIF = 0 ... 4 corresponds to the five network interfaces as illustrated below:

**CPU Hardware System Diagnosis**

**CPU 1 - CPU1:**

**List of Ethernet interfaces:**

| Port | Name | Usage | MAC | IRQ | IN Errors | OUT Errors |
|---|---|---|---|---|---|---|
| ETH1 | gei3 | IP stack - 192.168.123.47/23 | Unknown | 16 | | |
| ETH2 | gei2 | RAWETH - free | Unknown | 19 | | |
| ETH3 | gei1 | RAWETH - free | Unknown | 18 | | |
| ETH4 | gei0 | RAWETH - free | Unknown | 17 | | |
| ETH5 | gei4 | RAWETH - free | Unknown | 16 | | |

3. Increase the variable *BUILDNO* by one (1), so P80i will notice the settings are changed. In P80i:

   Right-click on the CPU -> Build all

   Right-click on the CPU -> Online -> Load & restart

4. Restart system

### 4.1.2 Changing Multicast Address

**Note**

Changing the multicast address is only necessary in case multiple clusters of HPCi controllers are connected via a factory-wide network and each HPCi cluster has its own dedicated *ibaPDA* server.

To prevent controllers from logging into the wrong *ibaPDA* server, each member of the cluster must be given the same multicast address of their corresponding *ibaPDA* server. Each cluster including its *ibaPDA* Server must therefore have its own unique multicast address.

1. Open the file `DASAGNT0.INI`, which is located at `P80_projectname.CTRL\Advanced\Configuration\`.
2. Assign a multicast address to the variable *MULTICAST_IP.*
3. Increase the variable *BUILDNO* by one (1), so P80i will notice the settings are changed.
   In P80i:

   Right-click on the CPU -> Build all

   Right-click on the CPU -> Online -> Load & restart
4. Restart system.
5. Generate the address book by means of the Address book Builder.
   For more information about this, see ↗ *Generating address books with the DAS address book generator*, page 28.
6. Open the file `TOC.INI`, which is located in the Address book directory.
7. Assign the same multicast address as in `DASAGNT0.INI` to the variable *AGENT_MULTICAST_IP*.
8. Increase the variable *FileVersion* by one (1), so *ibaPDA* will notice the settings are changed.

### 4.1.3 Changing the Default Overload Protection Limits

The DASAGNT driver has a build-in overload protection to prevent CPU overload by requesting too many signals. The default limit for the load caused by the DASAGNT itself is 30 %. The DASAGNT is calculating this value based on a 1 µs VME transfer time per 4 bytes of data. The DASAGNT checks also if the **total load** will not exceed 90 %. (Total load = load caused by the application and estimated load of the DASAGNT itself).

1. To change these default limits, open the file `DASAGNT0.INI`, which is located at `P80_projectname.CTRL\Advanced\Configuration\`
2. Following 2 parameters can be added/changed in the [GENERAL] section:

   MAX_ALLOWED_LOAD=30

   MAX_SYSTEM_LOAD=90
3. The above mentioned values are the default values for the load limits in percentage.

   MAX_ALLOWED_LOAD is the limit for the DASAGNT load

   MAX_SYSTEM_LOAD is the total load limit
4. After changing one of these values, increase the variable *BUILDNO* by one (1), so P80i will notice the settings are changed.
5. In P80i:

   Right-click on the CPU -> Build all

   Right-click on the CPU -> Online -> Load & restart
6. Restart the system.

## 4.2 Hardware Definition

The next step is the definition of the hardware that will be used to transfer the data to *ibaPDA*. There are 4 options:

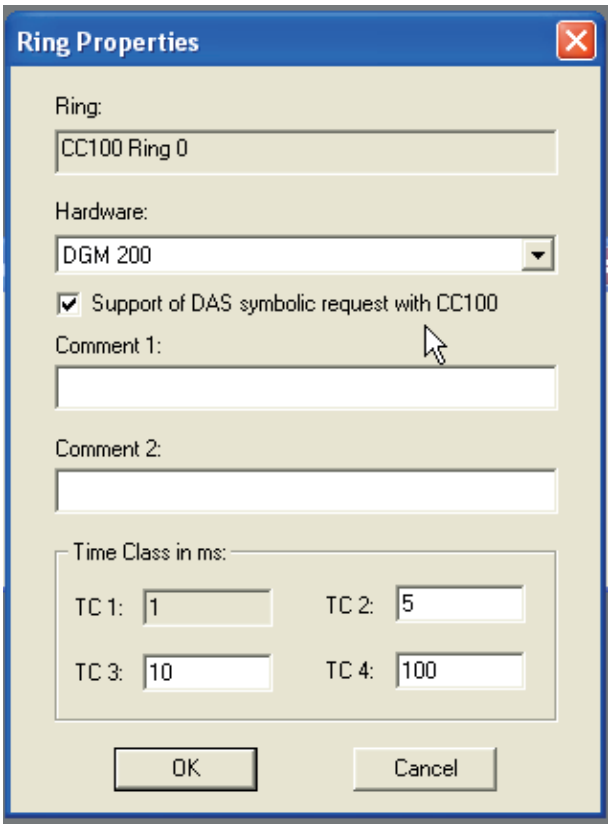
- DGM 200-V/DGM 200-P
- ibaLink-SM-128V boards (or ibaLink-VME in SM128 mode)
- Reflective Memory 5565/5576 boards
- ibaFOB-R board (defined as a 5565 board in the hardware configuration)
- ibaLink-VME board in P2P mode.

The DGM 200 boards cannot be used in combination with *ibaLink-SM-128V* or Reflective Memory. The *ibaLink-SM-128V* and Reflective Memory boards can be used together. The *ibaLink-VME* which simulates a Reflective Memory board cannot be used together with a real Reflective Memory board.

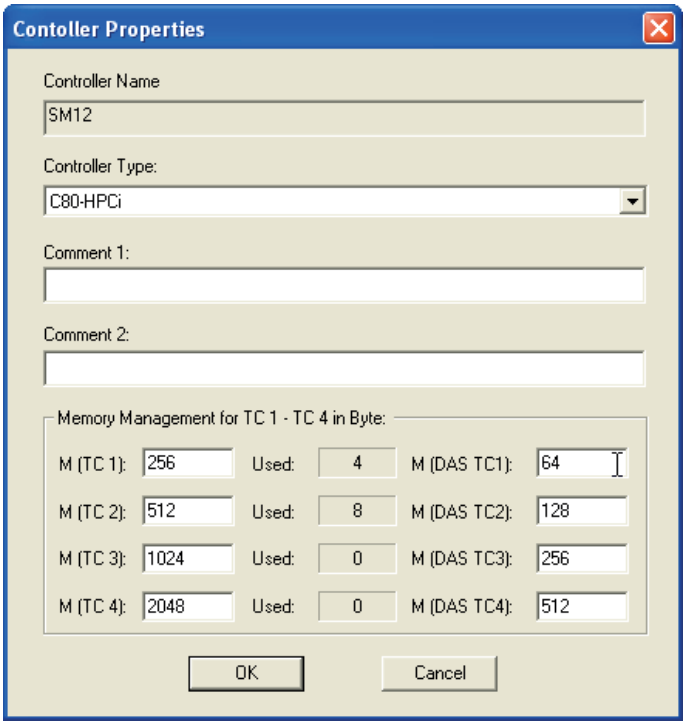
### 4.2.1 CC100/DGM200

DGM stands for Deterministic Global Memory and is an enhanced hardware platform for the CC100 network. The CC100 network has a star topology with the concentrator DGM 200-C in the center. The DGM 200-V is the VME board that fits in the HPCi rack. The DGM 200-P is the PCI board that fits in the APC- or RXi-PC on the HPCi side as well as in the ibaPDA PC (PCI slot required). The DGM 200-E communication board can be used as an external media adapter for the *ibaPDA* PC instead of the DGM 200-P board if the PC has no PCI slots.

The Coordination Channel Manager program (CCM32.exe) is used to configure the CC100 network. Version 2.17a or higher of CCM is required. On the ring properties you have to enable support for HPCi request by checking the *Support of DAS symbolic request with CC100* checkbox.

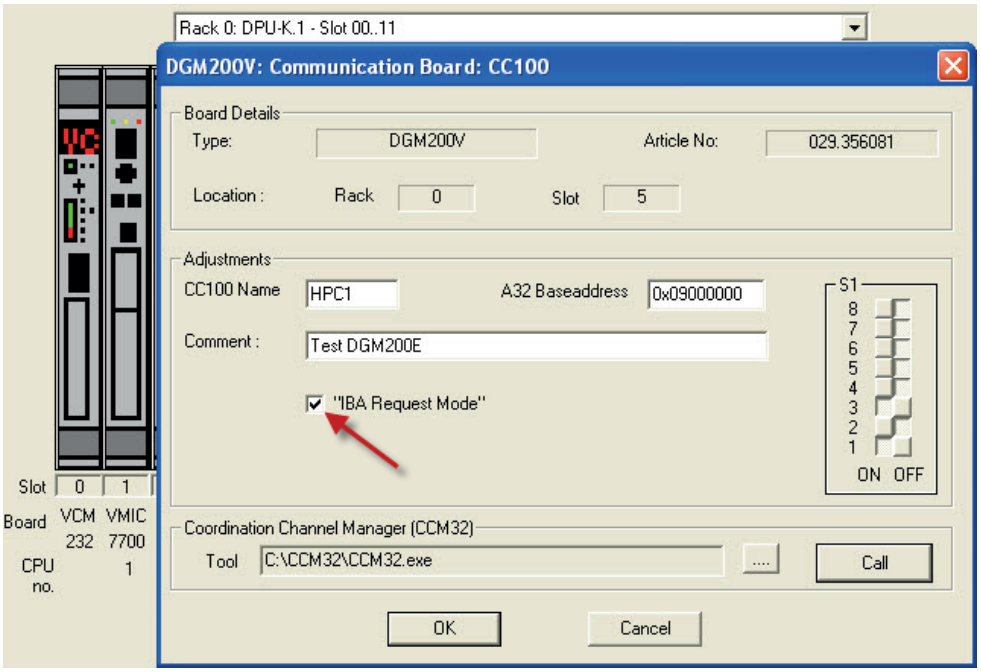



In the properties of each controller on the DGM network you have to reserve some space for the DASAGNT. For each time class you have to specify how many bytes you want to use to transfer HPCi request data. You do this by filling in the DAS column on the controller properties dialog.

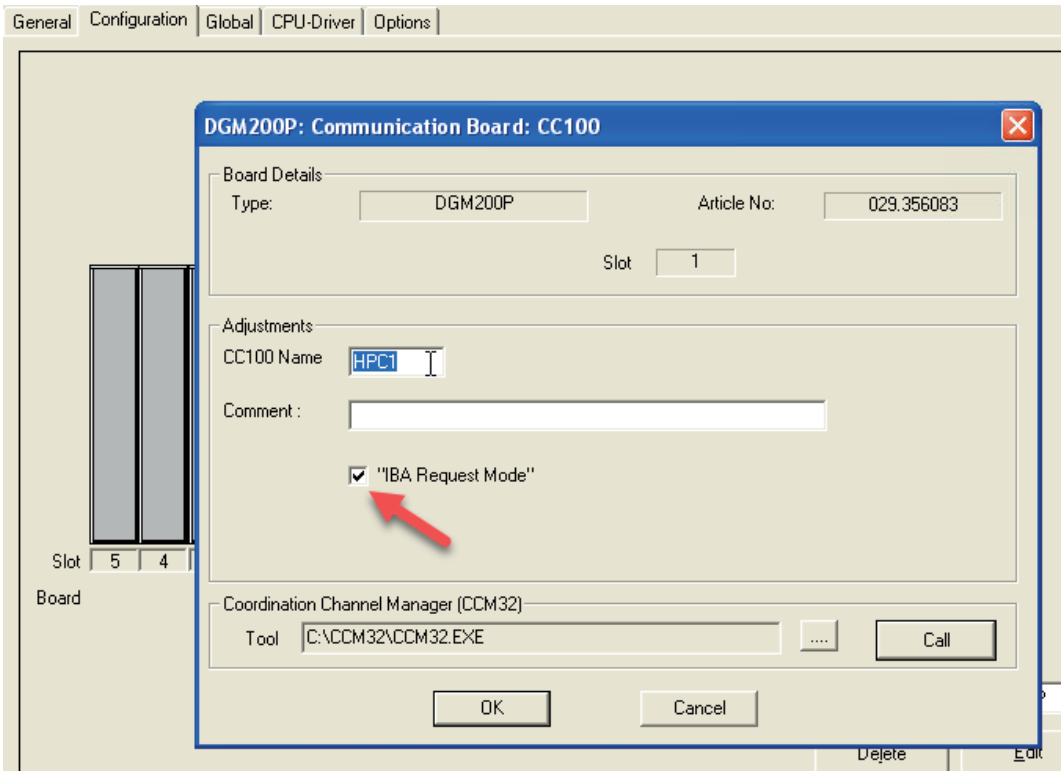



Finally, in the P80i hardware configuration tool, you have to enable HPCi request support in the properties of the DGM 200-V board for VME-based controllers or of the DGM 200-P board

for PC-based controllers. You do this by checking the checkbox called *Support for Data-Acquisition-System (DAS-Agent)* or *"IBA Request Mode"* respectively. This will enable the DGM to be used by the DASAGNT.




*Example DGM 200-V*

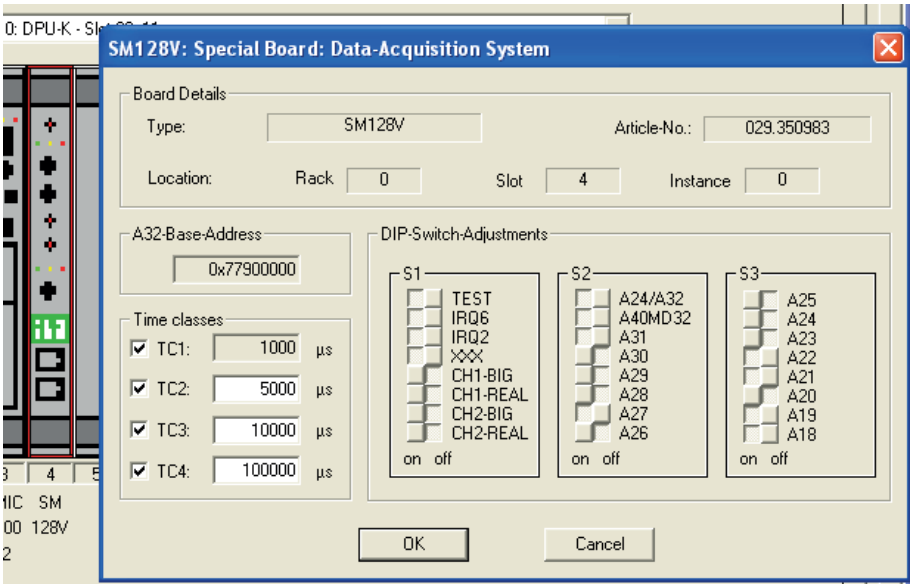



*Example DGM 200-P*

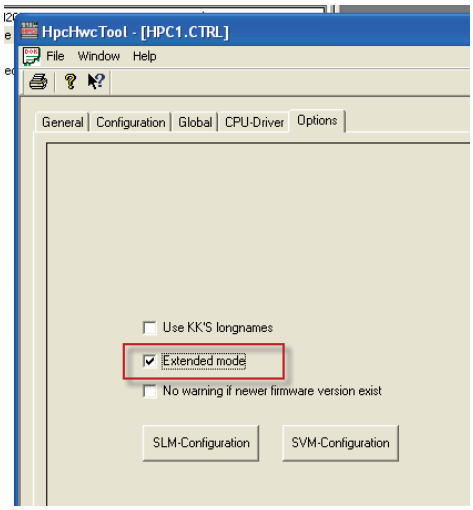
### 4.2.2 SM128V

The ibaLink-SM-128V-i-2o (short: SM128V) board is a VME board that has 2 fiber optical output channels and 1 fiber optical input channel. Only the 2 output channels are used by HPCi request. Each channel can transfer 264 bytes of data. 8 bytes are always reserved for digital values. The other 256 bytes can be used for both analog and digital values. There are 4 SM128V boards supported in one rack.

The only thing you have to do to use the SM128V for HPCi request is add one or more SM128V boards to the hardware configuration of your P80i project.




The VME base address is normally calculated by the hardware tool. If you want to change it then you have to enable extended mode in the hardware tool. The dialog also shows you how to set the DIP switches on the board. The 2 channels must be set to big-endian mode otherwise the data will arrive swapped on the FOB board in the PC.




In the properties of the SM128V board you can also set the 4 different time classes. You can decide which of the time classes can be used on this board. The easiest way is to just select all of the time classes. *ibaPDA* will distribute all requested signals from all time classes over the SM128V boards automatically.

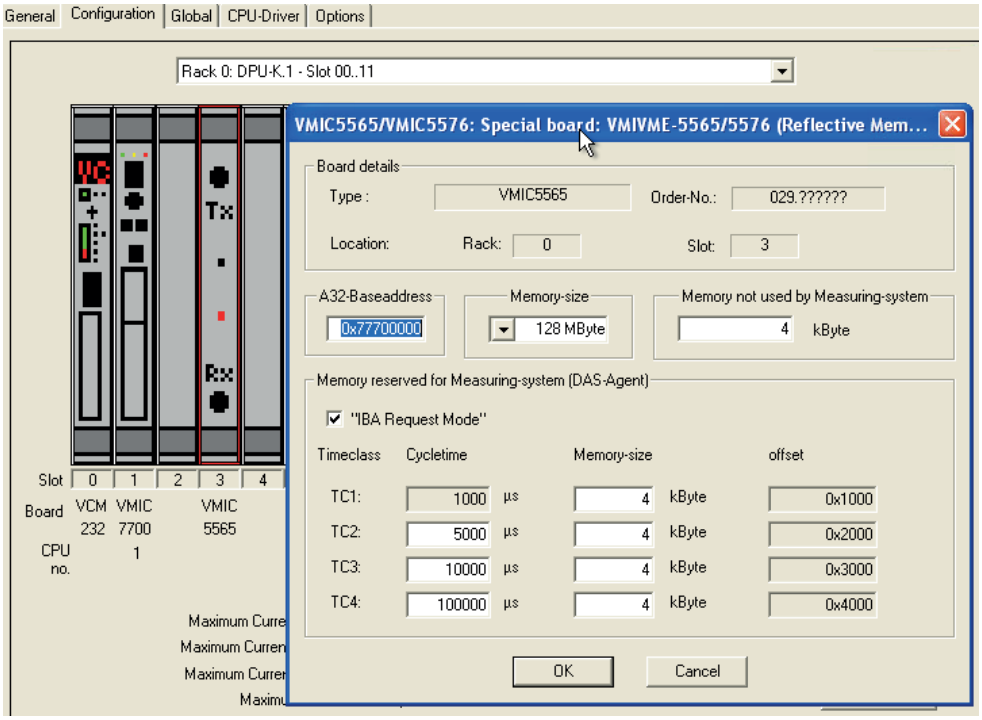
**Note**

Because the board *ibaLink-SM-128V-i-2o* is an outdated model, the successor board *ibaLink-VME* may be used as a replacement or spare part when set on a mode which is compatible to the old board (rotary switch S1 on "0" or "8").

### 4.2.3 Reflective Memory

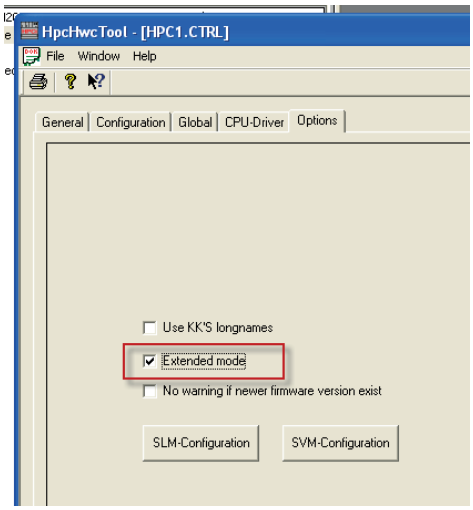
For a VME rack based controller the Reflective Memory boards VME-5565 and former VME-5576 (formerly VMIVME5565/5576) are supported. For the PC-based controllers (APC... and RXi...) the Reflective Memory boards PCI- or PCIE-5565PIORC and former PCI-5576 (formerly VMIPCI5565/5576) are supported. A PCI- or PCIE-board should also be installed in the *ibaPDA* computer.

The only thing you have to do in order to use the Reflective Memory board for HPCi request is to add it to the hardware configuration of your P80i project.




*Example VME- Reflective Memory board VMIVME5565*

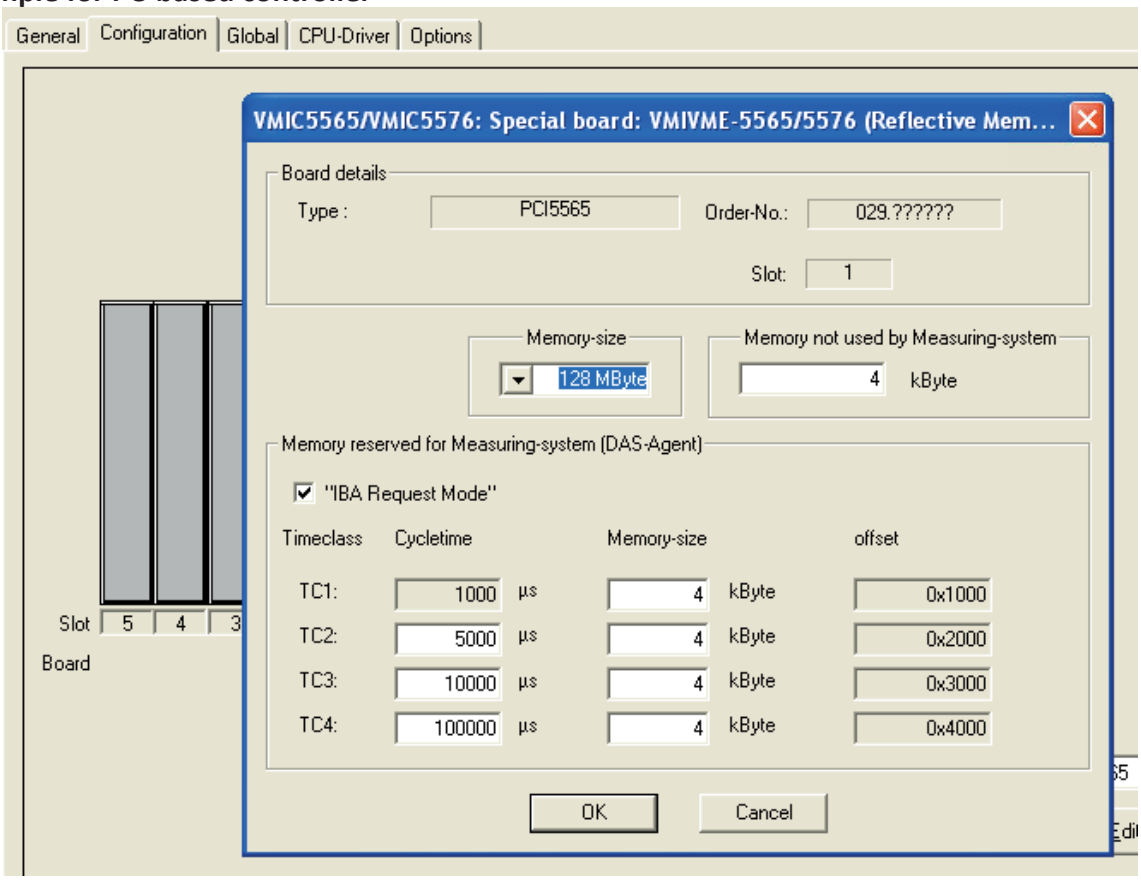
The VME base address is normally calculated by the hardware tool. If you want to change it then you have to enable extended mode in the hardware tool.



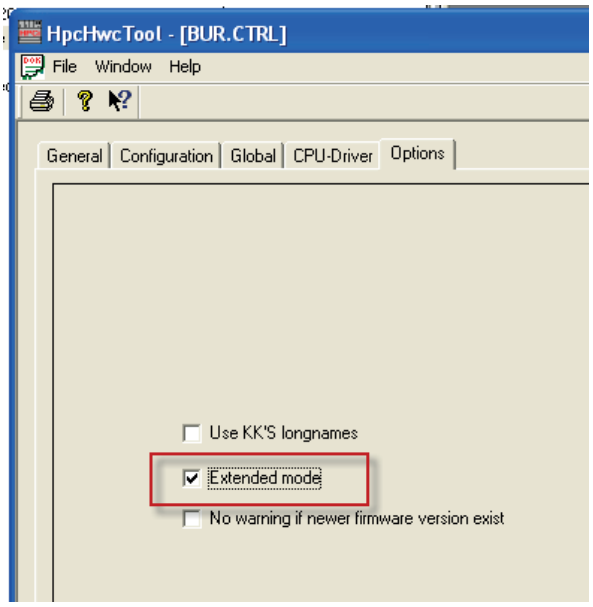

You must setup the correct memory size of the board. You also have to configure how much memory is used by the application and how much memory can be used by the HPCi request system for the 4 time classes.

If you are using the VMIVME-5576 (VME-5576) board then you must set the swap mode of the board VMIPCI-5576 (PCI-5576) **in** *ibaPDA* to *Byte and word swap*. This is needed because the HPCi CPUs perform all VME-bus accesses in big-endian and the PC CPUs are little-endian. If you are using the VMIVME-5565 (VME-5565) board then the swapping is done by the boards automatically.

**Example for PC-based controller**




*Example Reflective Memory board PCI-5565*




### 4.2.4 ibaFOB-R

This chapter describes the integration of the *ibaFOB-R* board into the hardware configuration of the HPCi system.

**Note**

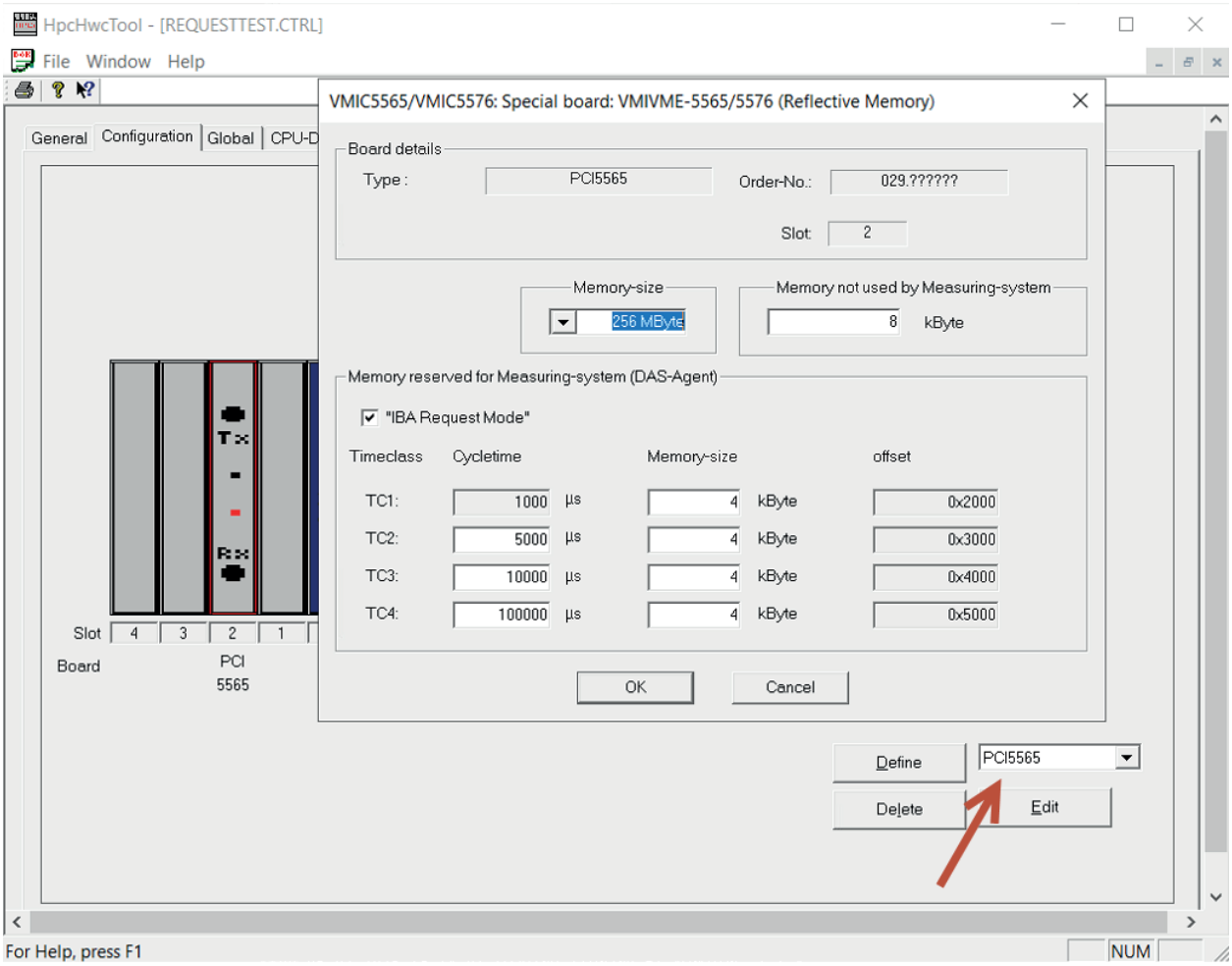
The *ibaFOB-R* board was developed by iba AG after Abaco discontinued their PCIe-based Reflective Memory card, ensuring continued support for Reflective Memory-based industrial automation solutions. To use the *ibaFOB-R* board in the more recent HPCi RXi controllers with a PCIe X4 slot available, customers should upgrade their HPCi version to V4.5.3 or higher .

In order to use the *ibaFOB-R* board for HPCi request you just have to add it to the hardware configuration of your P80i project as a PCI5565 board as depicted below.

This is exactly the same configuration as with a legacy 5565 board, explained in chapter ↗ *Reflective Memory*, page 21 for a pc-based controller.

Enable the option *IBA Request Mode*.

Set the correct memory size (256 Mbyte for the *ibaFOB-R*) and configure how much memory is used by the application and how much memory can be used by the HPCi request system for the 4 time classes.




The settings of this hardware configuration for the DASAGNT driver is saved in the `DASAGNT0.ini` file which is located at `P80_projectname.CTRL\Advanced\Configuration\`.

Since multiple reflective memory cards are now supported which are network compatible with the legacy 5565 board, but have their own specific vendor ID, a field ITFXX_PCIVIDDID is provided in the `DASAGNT0.ini` file to specify the PCI vendor/device id combination.

However, the hardware configuration for the PCI5565 board does not have a way to specify the vendor/device ID settings of the board in the dialog itself.

The first time the `DASAGNT0.ini` file is generated, the value of the ITFXX_PCIVIDDID fields is set by default to 0x114a5565, the original VMIC vendor and device ID combination.

So, to be able to use the iba *ibaFOB-R* board follow these steps:

1. Open the file `DASAGNT0.ini`, which is located at `P80_projectname.CTRL\Advanced\Configuration\` with a text editor.
2. Find the ITF*XX*_PCIVIDDID fields in the [ITFCONF] section where *XX* stands for the 4 time classes 00...04. Change the ITF*XX*_PCIVIDDID field values to 0x167f5565
   Example for time class 00 : **ITF00_PCIVIDDID=0x167f5565**
3. After changing these 4 values, increase the variable BUILDNO by one (1), so P80i will notice the settings are changed.

4. Save the file.
   In the P80:

   Right-click on the CPU -> Build all

   Right-click on the CPU -> Online -> Load & restart

4. Restart the system.

After restarting, observe the Historical logger on the web interface of the controller:

| | | | |
|---|---|---|---|
| Kernel:DASAGNT drv | INFO | 2025-Feb-04 10:51:29.897 | DAS Agent 0 is ready / waiting for connection. (4/4 interfaces,1 resources) |
| Kernel:DASAGNT drv | INFO | 2025-Feb-04 10:51:30.898 | Data acquisition system connected now. (IP: 192.168.122.89) |
| Kernel:DASAGNT drv | ERROR | 2025-Feb-04 10:51:30.898 | Unexpected error occured (0xffffffff,0x2d,0x0, Line: 3208) |

- DASAGNT should be announced as "is ready".
- If an *ibaPDA* system is connected, "Data acquisition system connected now" is reported together with the IP-address of the connected *ibaPDA* system..
- The message "Unexpected error occurred" can be ignored.

### 4.2.5 ibaLink-VME board in P2P mode

The ibaLink-VME board is a VME board that has 2 fiber optical output channels and 1 fiber optical input channel. Only **channel 1 output** is used by HPCi request. The ibaLink-VME has to be set in P2P mode D.

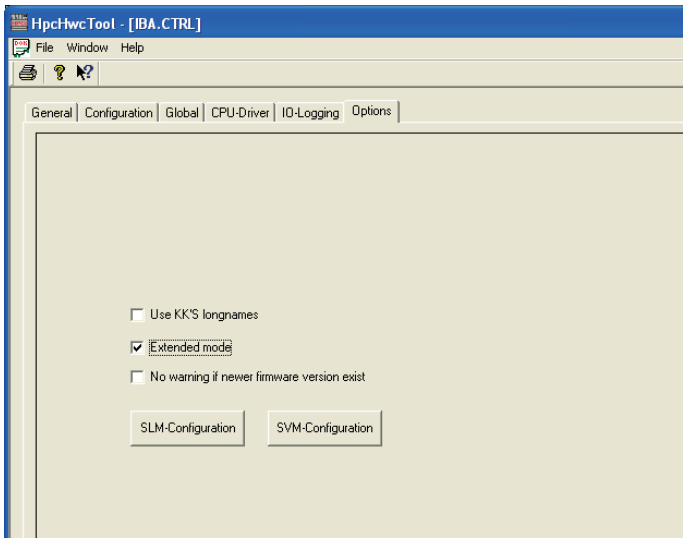
**Other documentation**

For more details about the different modes of the *ibaLink-VME* board, please refer to the *ibaLink-VME* manual.

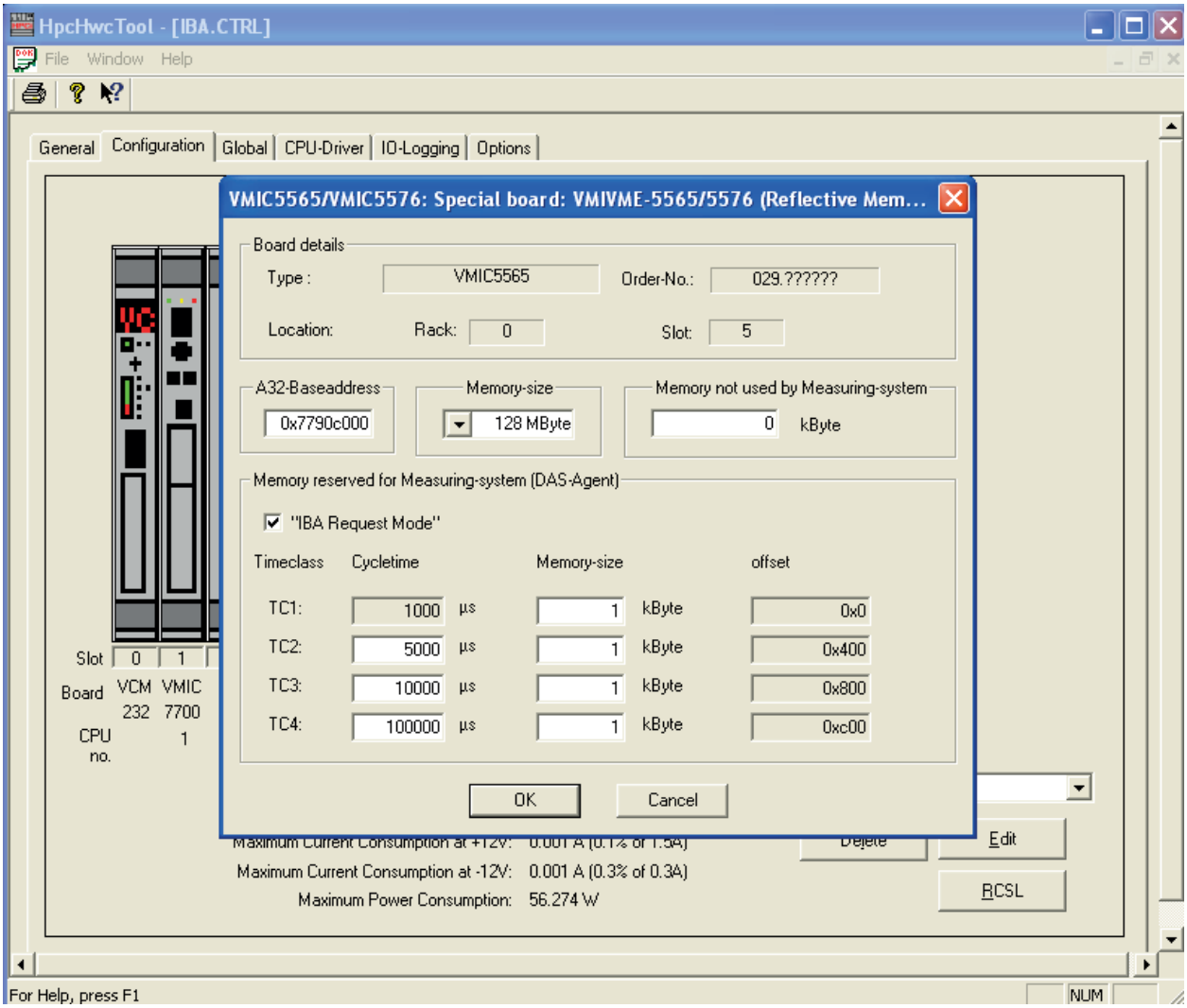
Using this mode the *ibaLink-VME* can transfer 4024 bytes with 1.4 ms refresh rate. These 4024 bytes can be used for both analog and digital values. The *ibaLink-VME* is not natively supported by the DASAGNT driver. To be able to use the *ibaLink-VME*, we declare the *ibaLink-VME* in the HPCi hardware configuration as a Reflective Memory VMIC5565 board. In P80i, the old name of the board is still used.

So from the point of view of the DASAGNT, the *ibaLink-VME* is treated as Reflective Memory and reported as such in loggings and reporting tools. There is only 1 *ibaLink-VME* board supported in one rack. (Limitation due to the Reflective Memory simulation)

The only thing you have to do to use the *ibaLink-VME* board for HPCi request is add the VMIC5565 board to the hardware configuration of your P80i project. But before doing this, you have to enable the extended mode in the hardware tool as depicted below.




Add the VMIC5565 board in the hardware configuration and edit the properties as illustrated below:

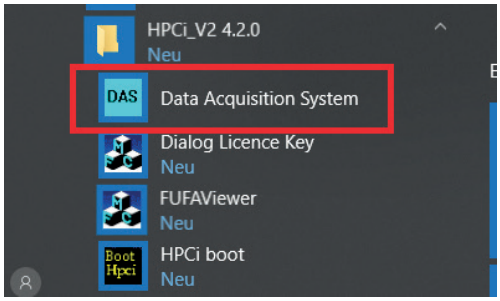


- A32-Baseaddress : board base address + 0xC000
  Base address 0x77900000 is generally used as the base address of the first *ibaLink-VME* (see *ibaLink-VME* manual). However we have to add an offset of 0xC000 to it so the DASAGNT driver can access the P2P send buffer.
- Memory-size: 128 Mbyte
- Memory not used by Measuring-system:
  Set this value to 0 to be able to use the complete range of 4024 bytes
- “IBA Request Mode” must be checked
- Configure the memory-size for each time-class. Keep in mind that there are only 4024 bytes available, which is little less than 4 Kbyte.

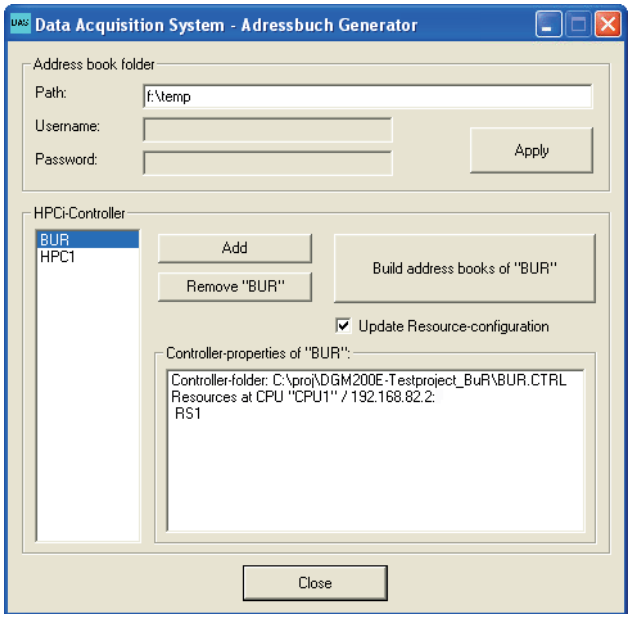
## 4.3 Generating address books with the DAS address book generator

The DAS address book builder is a program developed by GE Energy Power Conversion. It generates a system overview file called `toc.ini` and address book files for all controllers in the HPCi system.

1. Open the DAS address book generator in the Windows start menu with the executable file `DAS_ADDRESSBOOKBUILDER.exe`.




2. Select a directory in which the address book generator will store the generated files. The *ibaPDA* server must be able to access this directory.




3. Click on <Assign>.

→ The directory is checked.

→ If there is no `toc.ini` file in the directory, this file is generated.
If a `toc.ini` file exists, it is read.

4. To add the controllers that make up your overall system, click <Add>.
5. Select the relevant controller in the *Open file* dialog.

→ The controller will appear in the *HPCi-Controller* list.

6. Select a controller from the *HPCi-Controller* list.

7. If the number of resources or the names of the resources have changed, activate the *Update Resource-configuration* option.
8. To create the address books for the selected controller, click <Build address books of …>.

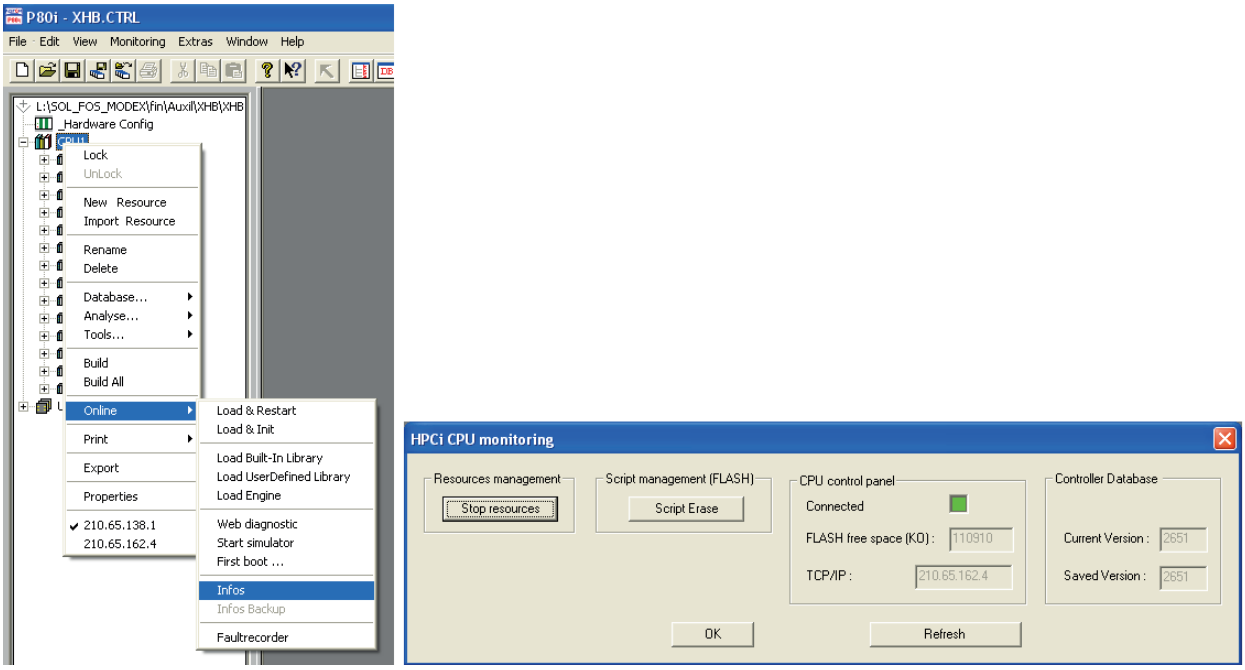
→ An address book file `*.tsv` is created for each resource of a controller and an additional file for the configuration settings `toc.ini`.

## 4.4 Useful P80i Functions

### 4.4.1 Script Erase

Script erase will delete all the data on the flash disk. To apply the erase the system needs to re-boot.

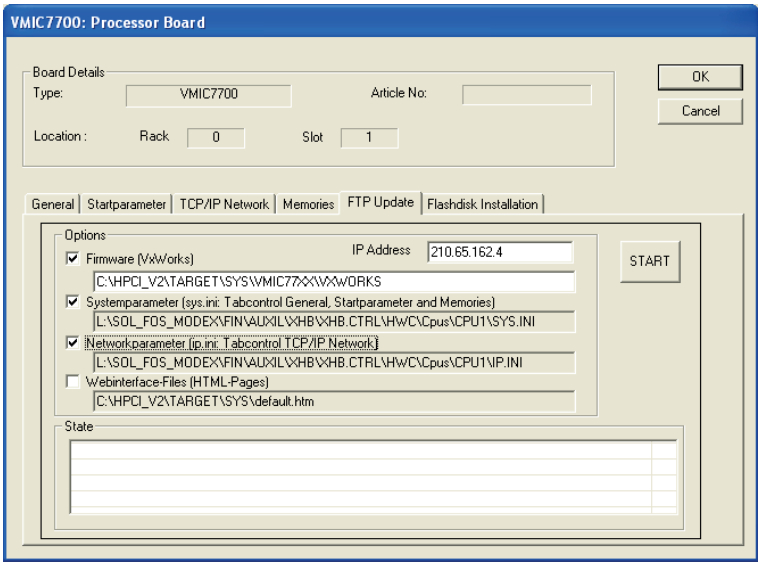
Right-mouse click on *CPU - Online - Infos Script Erase*.




### 4.4.2 FTP Update

FTP Update can be used to load the VxWorks firmware, Systemparameters, Networkparameter and Webinterface Files into the HPCi CPU.

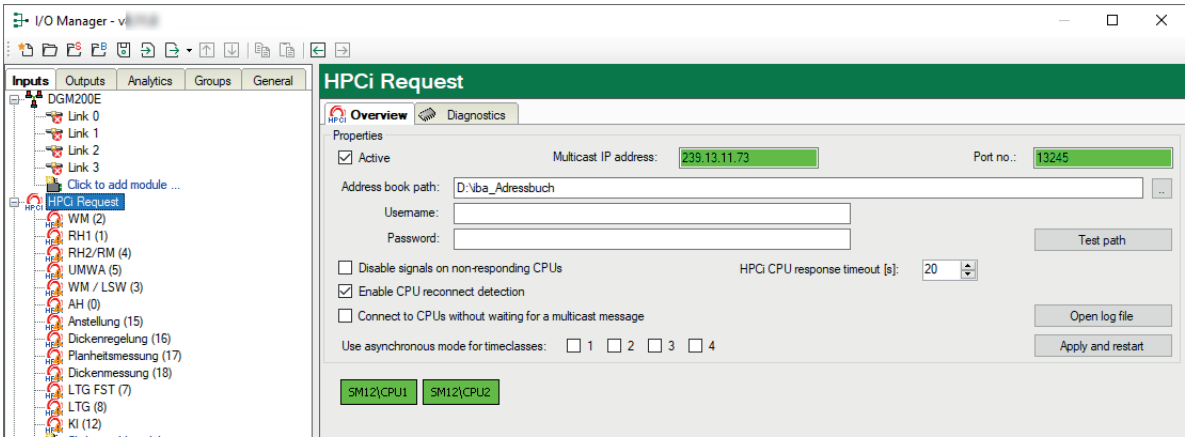
Hardware configuration - Double-Click on CPU - Select *FTP Update* tab



# 5 Configuration and engineering ibaPDA

## 5.1 General interface settings

The interface *ibaPDA-Request-HPCI* is configured in the *ibaPDA* "I/O Manager". If all system requirements are met, the "HPCi Request" interface is displayed in the interface tree.




The interface has the following functions and configuration options:

**Active**
The interface can be enabled or disabled with the *Active* checkbox.

**Addressbook path**
Enter here the path to the address books. Enter user credentials if configured.

Click the <Test path> button to test the access to the path.

With the <Open log file> button you can open the log book entries generated during the connection setup in the default editor.

Click the <Apply and restart> button. You should then see all the CPUs that are configured in the system on the bottom. The color of the CPU corresponds to the status of the connection to the CPU. There are 3 possibilities:

- Red: There is no TCP connection and no data connection to the CPU
- Yellow: There is a TCP connection but no data connection to the CPU
- Green: There is a TCP connection and a data connection to the CPU

A CPU can also be flashing. This means that there is a connection to the CPU but it was not listed in the `toc.ini` file. This also means that there is no address book available for that CPU. If this happens then you should update the `toc.ini` via the DAS address book builder.

**Disable signals on non-responding CPUs / HPCi CPU response timeout**
At the start of acquisition all CPUs are polled. If the *Disable signals on non-responding CPUs* checkbox is checked and a CPU does not respond within the specified *HPCi CPU response timeout* then the related signals will be deactivated and the acquisition will be started without these

signals. The use of this option is recommended during commissioning or maintenance works, when some HPCi stations are switched off. If this option is not enabled the acquisition won't start until all CPUs have replied to the polling at start of acquisition.

**Note**

Choose a generous timeout setting. Recommendation:

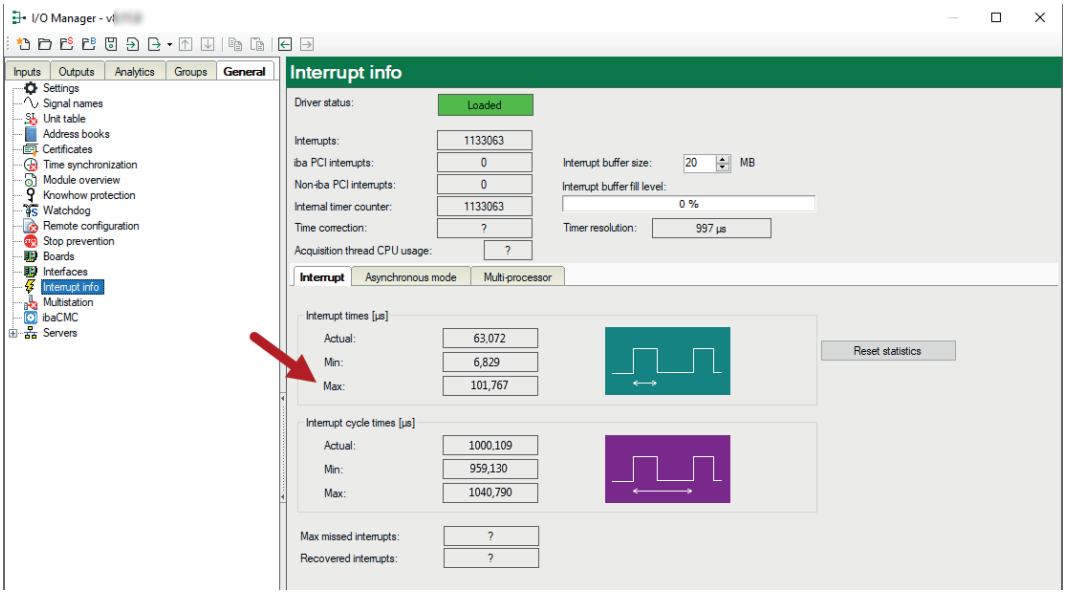
1. Check the time between start of acquisition (<OK> or <Apply> in the I/O Manager) and the time when all stations are available (green).
2. Set timeout to this time + 10 s in order to make sure that all stations can connect.

**Enable CPU reconnect detection**
When *Enable CPU reconnect detection* is enabled then ibaPDA checks on a periodical basis whether an earlier non-responding CPU tries to reconnect. If a reconnection attempt has been detected, ibaPDA stops the acquisition and restarts the acquisition with the new CPU.

**Connect to CPUs without waiting for a multicast message**
When this option is enabled, *ibaPDA* will establish a unicast connection to each CPU, based on the IP address in the `toc.ini` file, as soon as possible. The option is disabled by default.

Generally, when starting the acquisition, *ibaPDA* waits for multicast messages sent by the CPUs before establishing the connection. In some cases the multicast messages can be delayed significantly due to network problems, particularly in networks with routers or switches using IGMP Snooping.

**Use asynchronous mode for time classes**
The asynchronous mode setting for the time classes determines when the driver of *ibaPDA* will copy data from the boards. If asynchronous mode is off then the data is copied during the interrupt service routine. If asynchronous mode is on then the data is copied on a separate thread outside of the interrupt service routine. Normally asynchronous mode should be off. Asynchronous mode is only needed when the interrupt service routine takes more than 1000 µs to copy all the data from the boards. You can check this by going to the *General* node in the I/O Manager and checking the *Interrupt info* tab.




If the maximum interrupt time is larger than 1000 µs then you should enable asynchronous mode for time class 4. Restart the measurement and check the maximum interrupt time again. If it is still larger than 1000 µs then try enabling asynchronous mode also for time class 3. Finally you can also enable asynchronous mode for time class 2 if it is necessary.

**Note**

If using a Reflective Memory PCI-/PCIE-5565PIORC or ibaFOB-R board, it is highly recommended to always use the asynchronous mode for time classes 2, 3 and 4. This board supports DMA which can transfer data with much less CPU overhead.

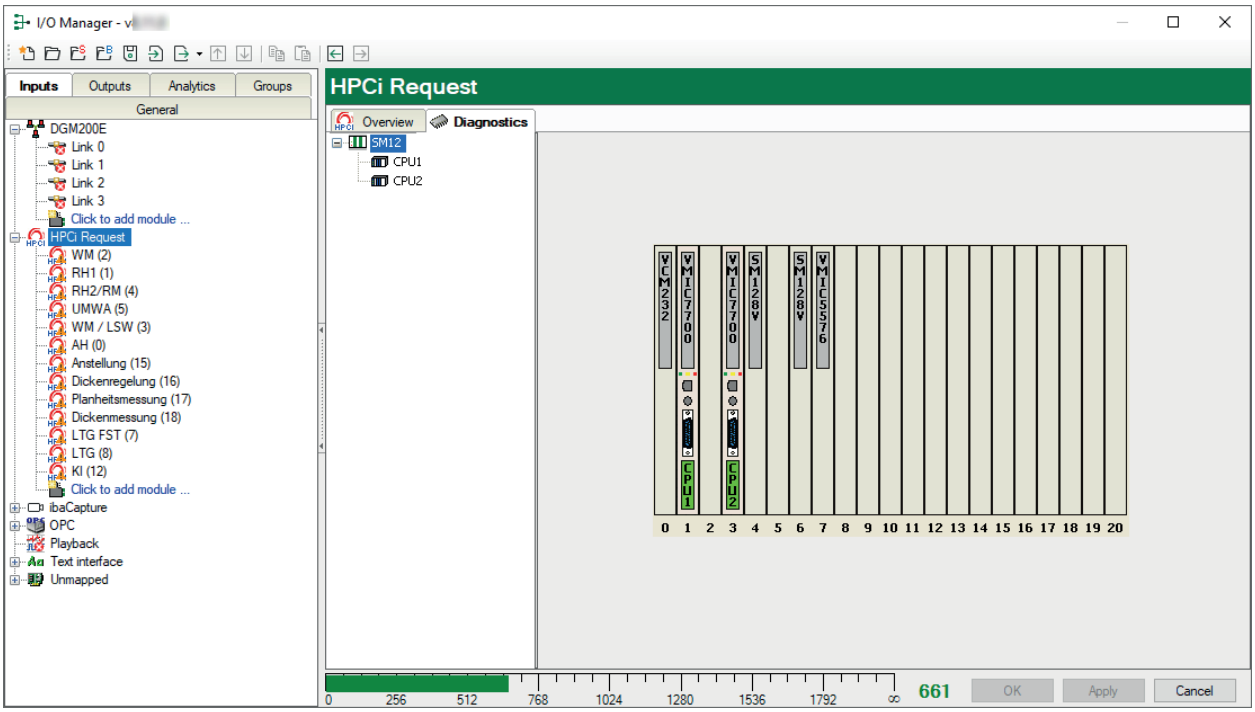
**Note**

If using the DGM200E interface, check the properties settings of the network adapter which is connected to the DGM200E-device. The *Jumbo Packets* should be enabled and set at 9014 bytes, if available.

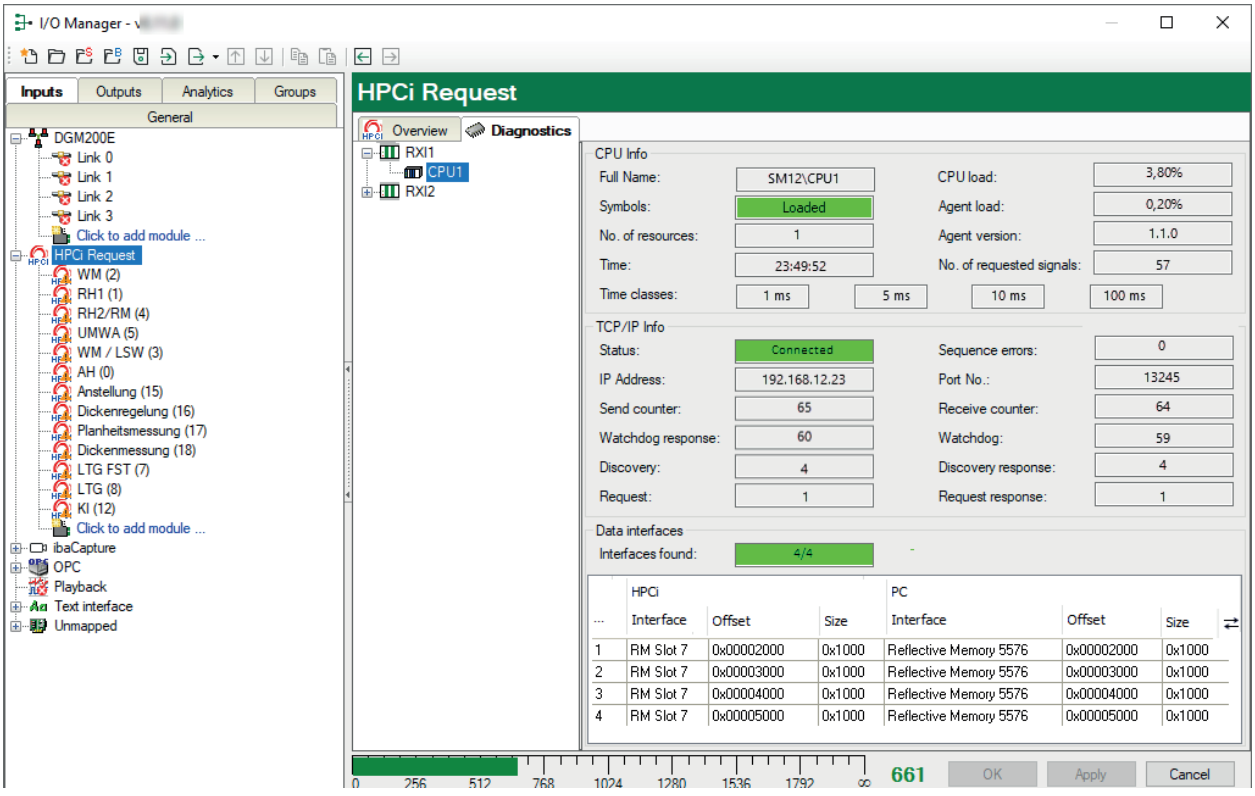
See manual *ibaPDA-Interface-HPCi-DGM200E*, chapter 5.2.1 for details.

### 5.1.1 Diagnostics

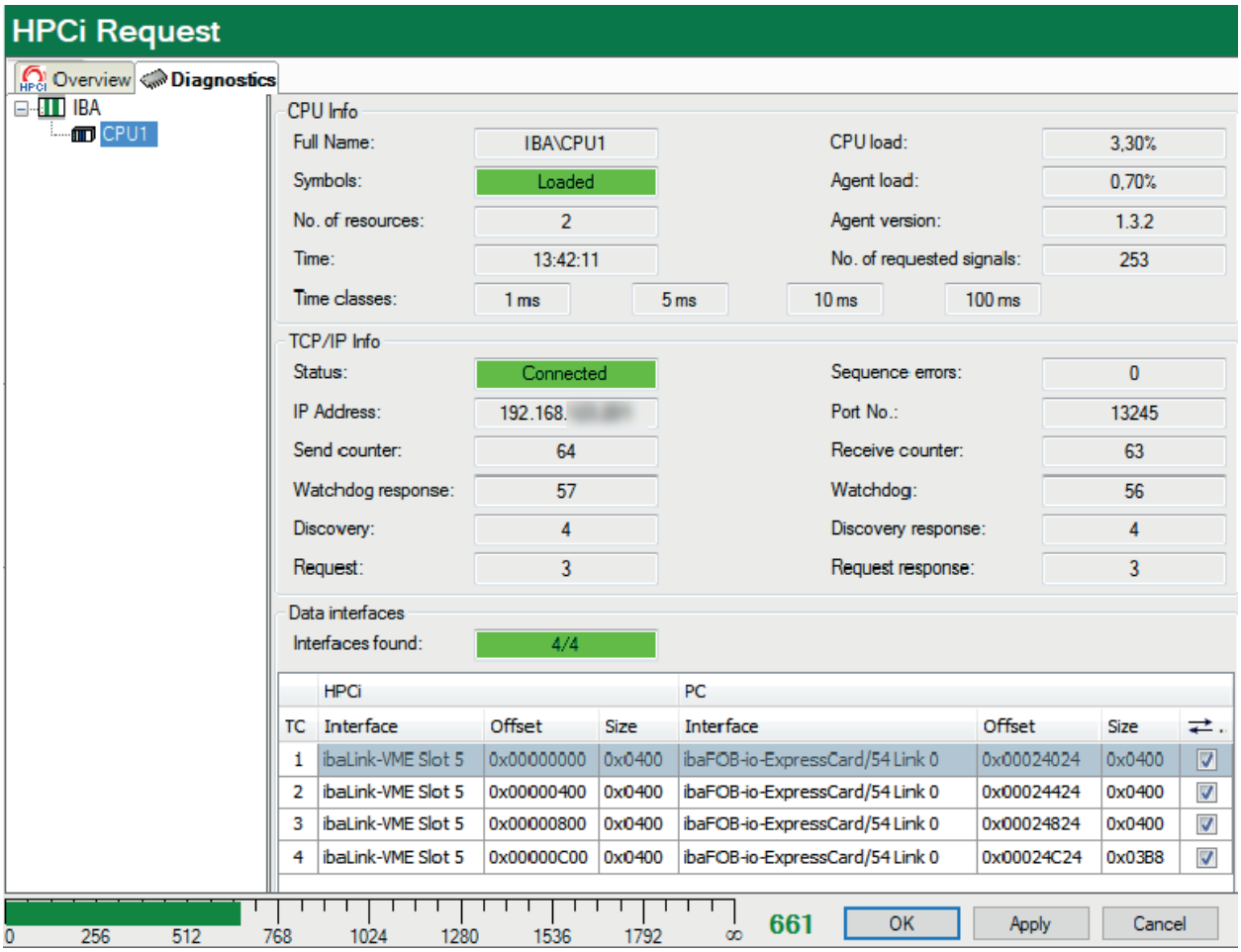
On the *Diagnostics* tab of the *HPCi Request* interface you can find some extra diagnostic info. Connected CPUs send the rack configuration in their status messages. This rack configuration is shown on the *Diagnostics* tab.




If you select the CPU in the tree then you get some extra information about it. You can view the CPU load and the load generated by the agent. You can see the status of the TCP connection and you can also see the data interfaces on the HPCi side and their counterparts on the PC side.

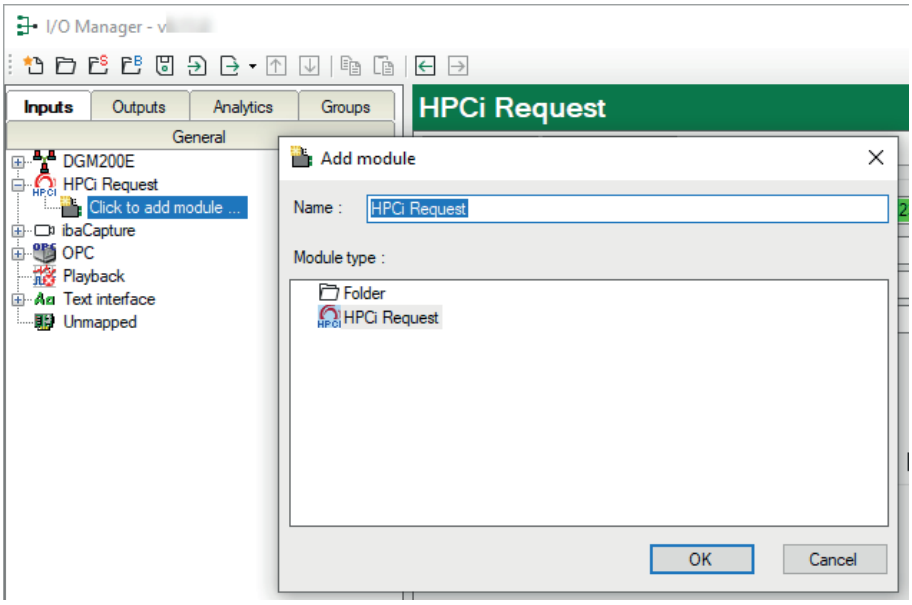



The example below shows the use of the *ibaLink-VME* board on the HPCi side and the *iba-FOB-io-ExpressCard* in a portable PC.

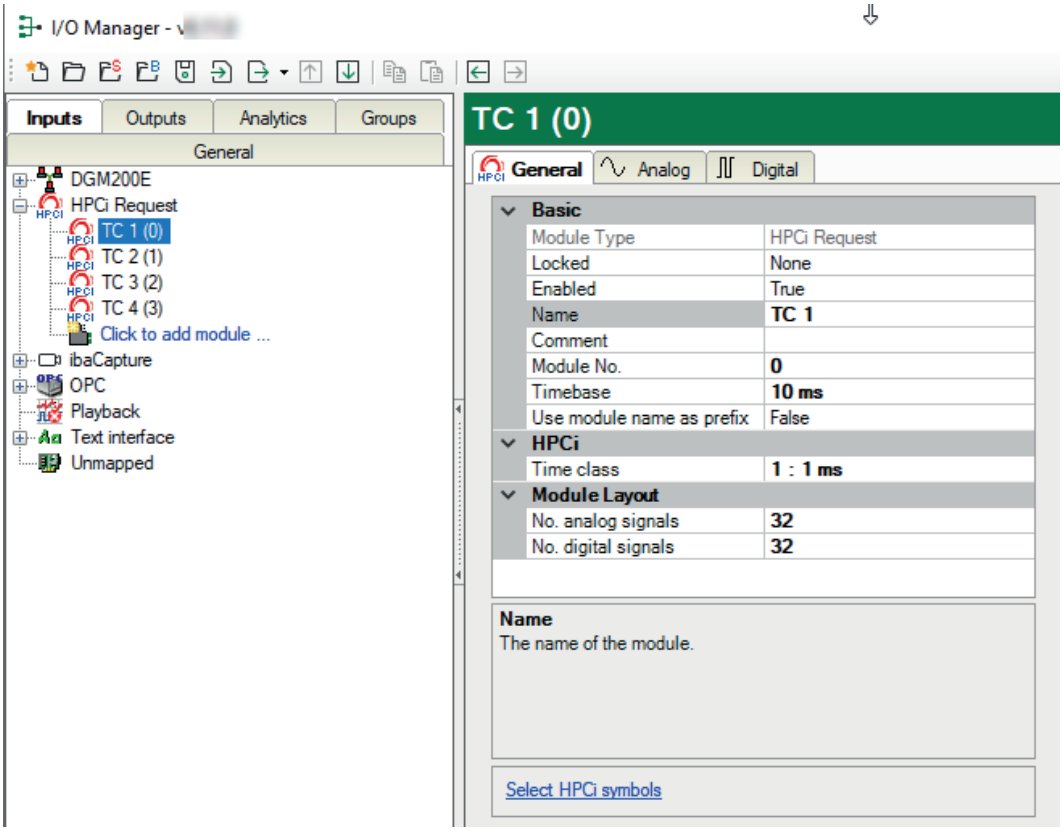



## 5.2 HPCi Request Module

Add an HPCi Request module in the I/O Manager by clicking below the HPCi Request interface. Select the desired module type and click <OK>.



### 5.2.1 General module settings




**Basic settings**

**Module Type (information only)**
Indicates the type of the current module.

**Locked**
You can lock a module to avoid unintentional or unauthorized changing of the module settings.

**Enabled**
Enable the module to record signals.

**Name**
You can enter a name for the module here.

**Comment**
You can enter a comment or description of the module here. This will be displayed as a tooltip in the signal tree.

**Module No.**
This internal reference number of the module determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

**Timebase**
All signals of the module are sampled on this timebase.

**Use module name as prefix**
This option puts the module name in front of the signal names.

**HPCi**

**Time class**
Select the time class from the dropdown menu. The time class is the rate at which the DASAGNT driver will send the data for this module to *ibaPDA*. The time base (in modules's basic settings) is the rate at which *ibaPDA* will sample the data it receives from the DASAGNT. Usually time class and time base are set to the same value.

**Module Layout**

**No. analog / digital signals**
Defines the number of configurable analog/digital signals in the signal tables. A maximum of 1000 signals for each are allowed.

**Select HPCi symbols**
When you click on the *Select HPCi symbols* hyperlink the HPCi signal browser will open.

## 5.3 Signal configuration

Once the connection to the CPU has been successfully established and the address book has been generated, the signals can be configured in the *Analog* and *Digital* signal tables of the module.

You have different methods to do this. The more convenient and save way to configure the signals to be measured is either using the HPCi signal browser or the drag & drop method.

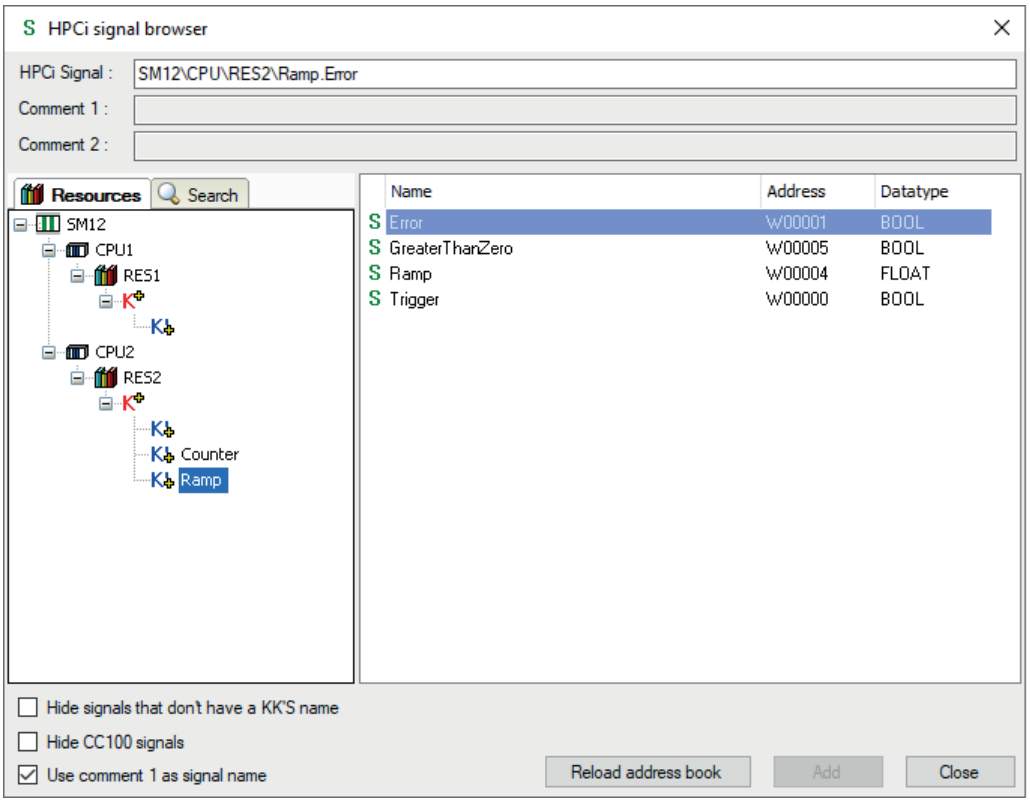
### 5.3.1 HPCi signal browser

Use the HPCi signal browser for selecting the signals to be measured.

In order to display the signals in the browser, make sure that the path of an address book file has been entered in the field *Address book path*, in the *Overview* tab on the *HPCi Request* interface node in the I/O Manager. In this path, there should be a valid file `toc.ini` as well as the corresponding address book files (`*.tsv`) of the HPCi stations.

The signal browser can be opened in different ways:

- In the *General* tab of an HPCi Request module click on the hyperlink *Select HPCi symbols*.
- Click on the little browser button < ... > in the *Analog* tab of an HPCi Request module, in the fields of column *HPCi Symbol*.
- Click on the little browser button < ... > in the *Digital* tab of an HPCi Request module, in the fields of column *HPCi Symbol*.




*HPCi signal browser for HPCi Request modules*

On the left side, in the *Resources* tab, a tree structure will appear containing the signal sources parsed from the address book file. For HPCi Request, the top-level node is the plant, comprising the controllers, the resources and the station nodes which finally contain the signals.

If you select a station node, you can see the signals on the right.

In the upper part of the dialog you find the name of the selected signal and - if available - the comment(s).

You can double-click on a signal to add it to the module. You can also select multiple signals by holding <CTRL> or <SHIFT> while selecting. When you click the <Add> button all selected signals are added to the module.

**Search function**
With the text based search function you can look for available signals by their name.

Click on the <Search> button in the *Search* tab and enter the complete text or a part of it into the *Search signal* dialog. Optionally, you can extend the search on the comments. Alternatively, you can look for KK'S names.

The search result is again a tree structure in the *Search* tab, which contains only the signals matching the search criteria. Proceed in the same way like in the *Resources* tab in order to add the requested signals to the signal table.

**Option "Hide signals that don't have a KK'S name"**
If you enable this option, only signals with KK'S names will be shown.

**Option "Hide CC100 signals"**
If you enable this option, signals which had been configured for CC100 with the CCM32 tool will be hidden.

**Tip**

The CC100 signals can be read in an efficient way by using the HPCi Lite module without generating addional stress on the CPU.

See ↗ *HPCi Lite*, page 43 or the manual of *ibaPDA-Interface-HPCi-DGM200E.*

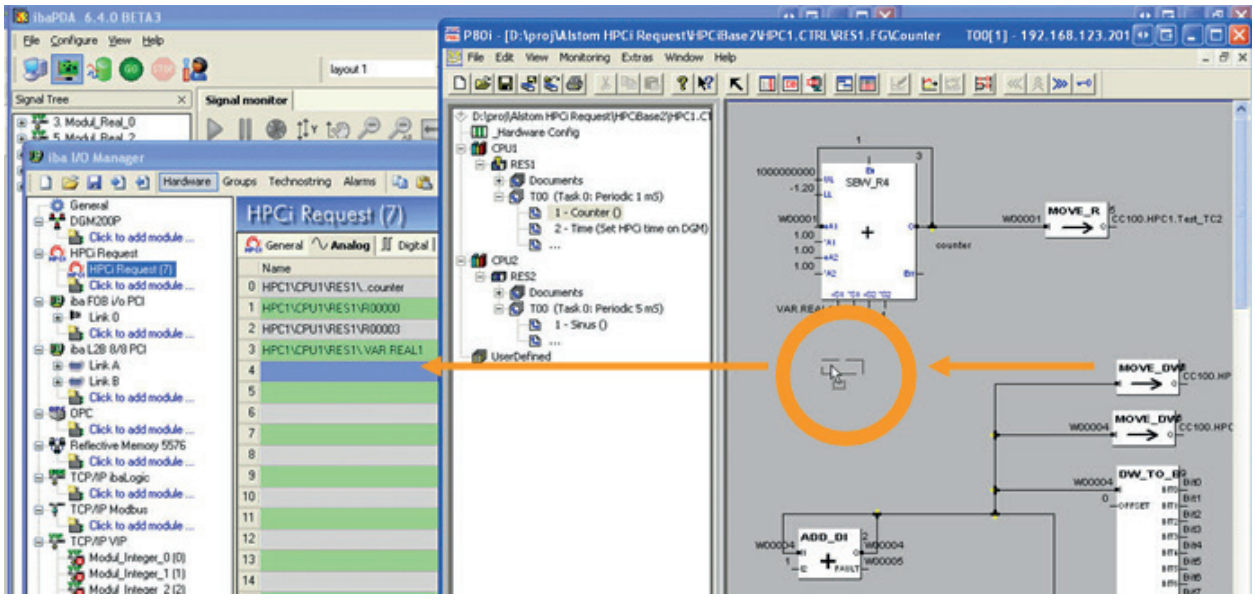
**Option “Use comment 1 as signal name”**
When selecting a signal from the signal browser, the HPCi symbol name will be automatically transmitted into the “Name” column. You may change the name manually afterwards. If you enable this option, the comment 1 will be used for signal name instead of the symbol name.

If required, you can reverse this in the signal table by using the context menu any time.

**Button <Reload address book>**
If no signals are shown in the browser or if the contents is obviously outdated, click on this button in order to reload the address book. If still no signals appear, then there is probably something wrong with the format or the path name of the supplied address book file.

### 5.3.2 Drag and drop with P80i

An alternative way of selecting signals is using drag & drop between P80i and the *ibaPDA* client. If the *ibaPDA* client is installed on a PC where P80i is also installed then you can open your project in P80i and just drag the signals you want to measure from P80i to *ibaPDA.*



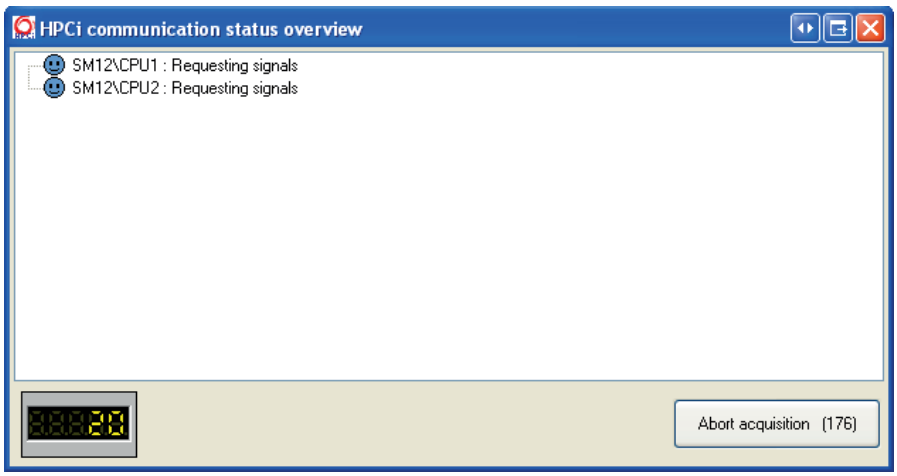

You can select signals from multiple CPUs in the module. They don’t even have to belong to the same controller.

## 5.4 Request process

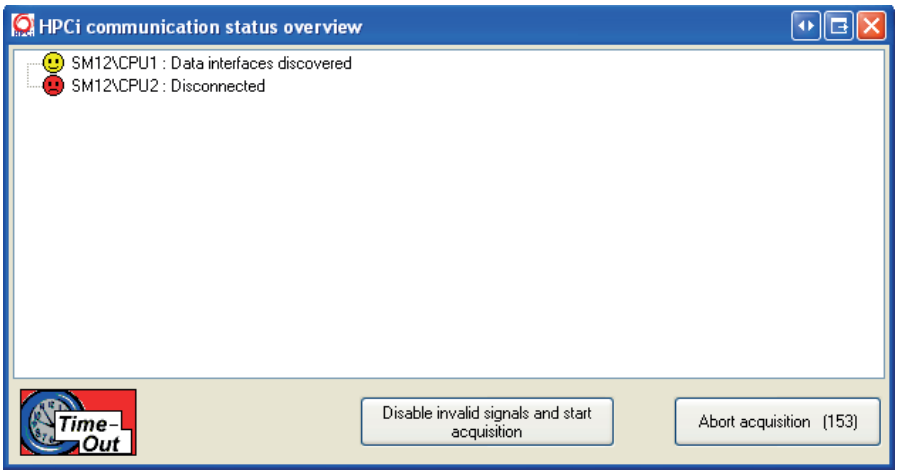
When you have configured the HPCi Request interface and you have added some HPCi Request modules then you can click the <OK> button to start the acquisition. The request process has several steps:

1. A stop message is sent to all connected CPUs.
2. Wait until the data interfaces of all active CPUs are discovered.
3. Map the signals on to the data interfaces.
4. Send request messages to the active CPUs.
5. Wait for the responses to the request messages.
6. If the responses are ok then start acquisition otherwise let the user decide what to do.

The progress of the request process is shown on the *ibaPDA* client.




If there is some error during the request process you can decide what to do next. You can decide to abort the acquisition. You can also decide to temporarily disable the signals on the non-responding CPUs and then try to start the acquisition again.

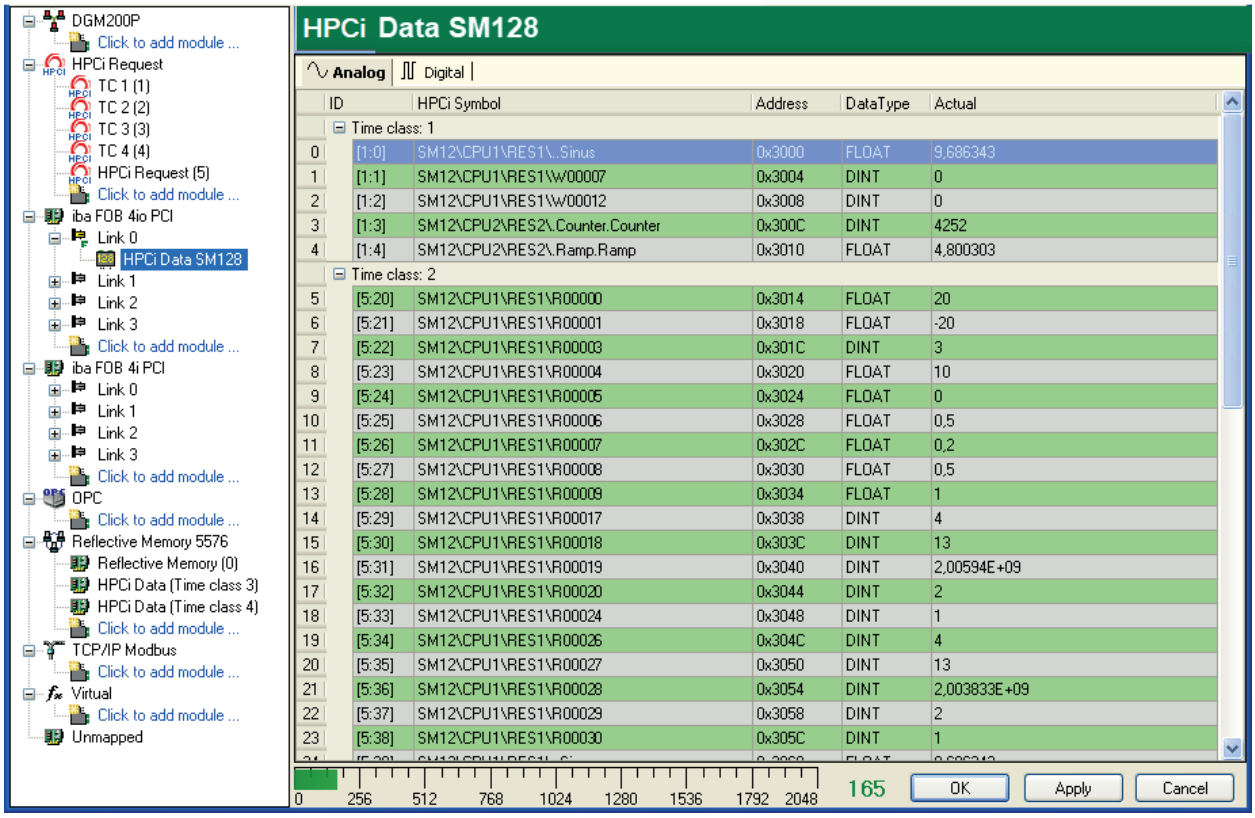


## 5.5 HPCi data modules

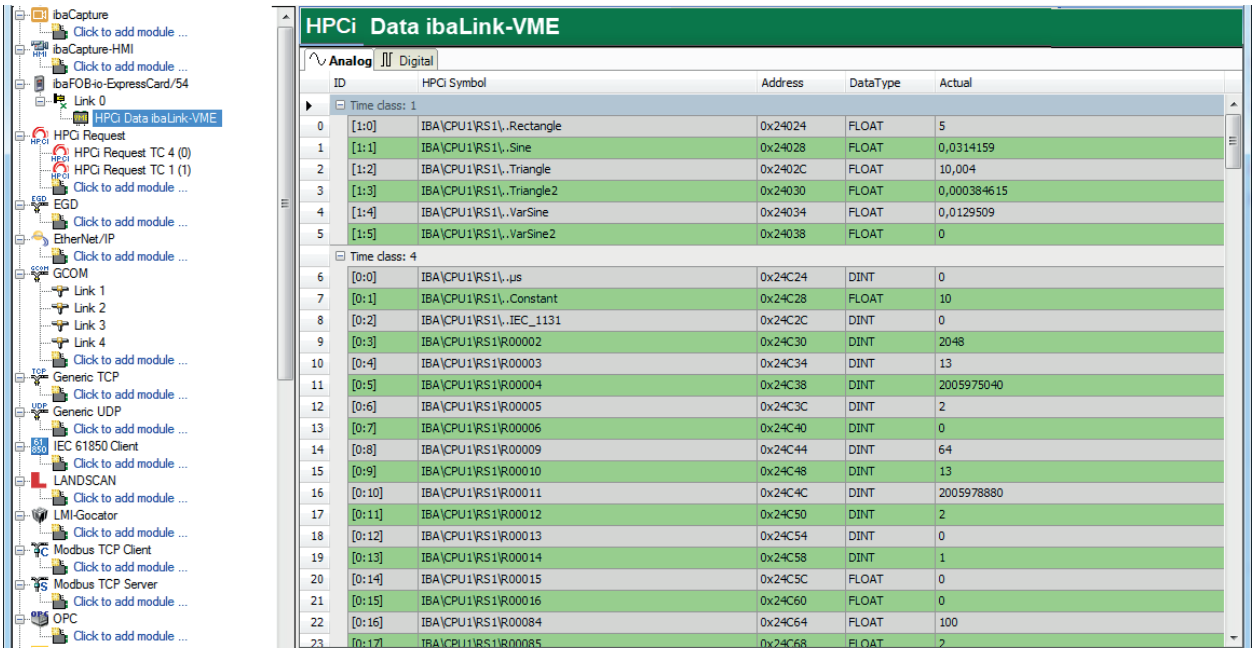
*ibaPDA* automatically maps the requested signals onto the available data interfaces for the CPUs. The HPCi data modules are generated during this mapping. These data modules are also shown in the I/O manager but they are just for diagnostics.

You find the data modules under the interface which is used for the data channel.

| Data Channel Hardware Interface | I/O Manager Interface for Data Modules |
|---|---|
| ibaLink-SM-128V-i-2o | ibaFOB-io...-D/-Dexp/-ExpressCard |
| Reflective Memory | Reflective Memory |
| ibaLink-VME in P2P mode | ibaFOB-io...-D/-Dexp/-ExpressCard |
| DGM 200-P | DGM200P |
| DGM 200-E | DGM200E |

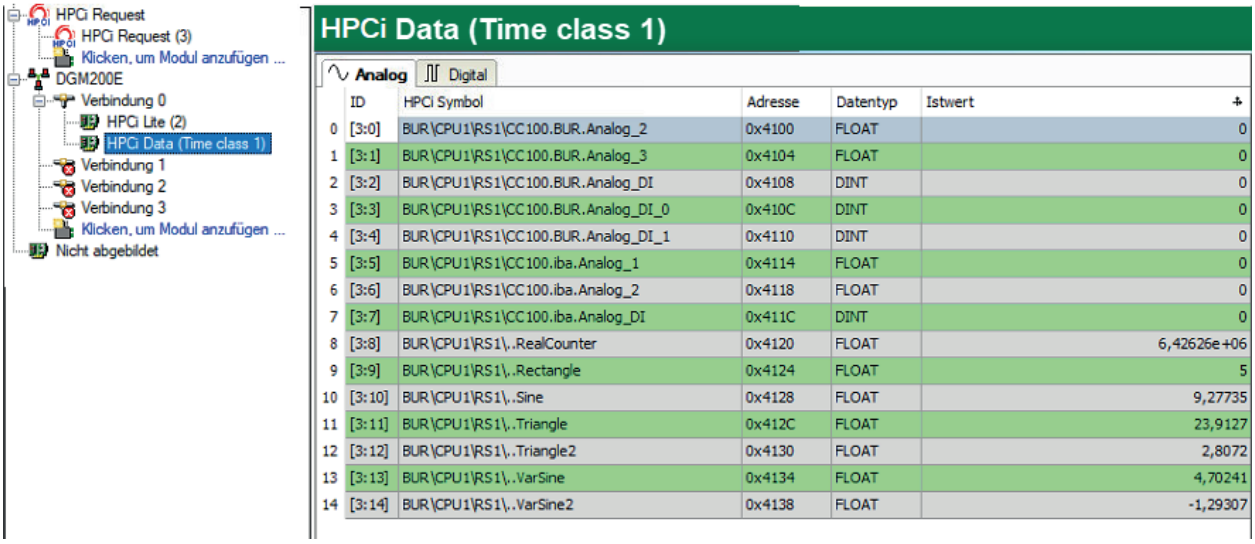



*Example of the ibaLink-SM-128V data module on an ibaFOB-4io card*




| | ID | HPCi Symbol | Address | DataType | Actual |
|---|---|---|---|---|---|
| | Time class: 1 | | | | |
| 0 | [1:0] | IBA\CPU1\RS1\..Rectangle | 0x24024 | FLOAT | 5 |
| 1 | [1:1] | IBA\CPU1\RS1\..Sine | 0x24028 | FLOAT | 0,0314159 |
| 2 | [1:2] | IBA\CPU1\RS1\..Triangle | 0x2402C | FLOAT | 10,004 |
| 3 | [1:3] | IBA\CPU1\RS1\..Triangle2 | 0x24030 | FLOAT | 0,000384615 |
| 4 | [1:4] | IBA\CPU1\RS1\..VarSine | 0x24034 | FLOAT | 0,0129509 |
| 5 | [1:5] | IBA\CPU1\RS1\..VarSine2 | 0x24038 | FLOAT | 0 |
| | Time class: 4 | | | | |
| 6 | [0:0] | IBA\CPU1\RS1\..µs | 0x24C24 | DINT | 0 |
| 7 | [0:1] | IBA\CPU1\RS1\..Constant | 0x24C28 | FLOAT | 10 |
| 8 | [0:2] | IBA\CPU1\RS1\..IEC_1131 | 0x24C2C | DINT | 0 |
| 9 | [0:3] | IBA\CPU1\RS1\R00002 | 0x24C30 | DINT | 2048 |
| 10 | [0:4] | IBA\CPU1\RS1\R00003 | 0x24C34 | DINT | 13 |
| 11 | [0:5] | IBA\CPU1\RS1\R00004 | 0x24C38 | DINT | 2005975040 |
| 12 | [0:6] | IBA\CPU1\RS1\R00005 | 0x24C3C | DINT | 2 |
| 13 | [0:7] | IBA\CPU1\RS1\R00006 | 0x24C40 | DINT | 0 |
| 14 | [0:8] | IBA\CPU1\RS1\R00009 | 0x24C44 | DINT | 64 |
| 15 | [0:9] | IBA\CPU1\RS1\R00010 | 0x24C48 | DINT | 13 |
| 16 | [0:10] | IBA\CPU1\RS1\R00011 | 0x24C4C | DINT | 2005978880 |
| 17 | [0:11] | IBA\CPU1\RS1\R00012 | 0x24C50 | DINT | 2 |
| 18 | [0:12] | IBA\CPU1\RS1\R00013 | 0x24C54 | DINT | 0 |
| 19 | [0:13] | IBA\CPU1\RS1\R00014 | 0x24C58 | DINT | 1 |
| 20 | [0:14] | IBA\CPU1\RS1\R00015 | 0x24C5C | FLOAT | 0 |
| 21 | [0:15] | IBA\CPU1\RS1\R00016 | 0x24C60 | FLOAT | 0 |
| 22 | [0:16] | IBA\CPU1\RS1\R00084 | 0x24C64 | FLOAT | 100 |
| 23 | [0:17] | IBA\CPU1\RS1\R00085 | 0x24C68 | FLOAT | 2 |

*Example of the ibaLink-VME data module on an ibaFOB-io-ExpressCard*




| | ID | HPCi Symbol | Adresse | Datentyp | Istwert |
|---|---|---|---|---|---|
| 0 | [3:0] | BUR\CPU1\RS1\CC100.BUR.Analog_2 | 0x4100 | FLOAT | 0 |
| 1 | [3:1] | BUR\CPU1\RS1\CC100.BUR.Analog_3 | 0x4104 | FLOAT | 0 |
| 2 | [3:2] | BUR\CPU1\RS1\CC100.BUR.Analog_DI | 0x4108 | DINT | 0 |
| 3 | [3:3] | BUR\CPU1\RS1\CC100.BUR.Analog_DI_0 | 0x410C | DINT | 0 |
| 4 | [3:4] | BUR\CPU1\RS1\CC100.BUR.Analog_DI_1 | 0x4110 | DINT | 0 |
| 5 | [3:5] | BUR\CPU1\RS1\CC100.iba.Analog_1 | 0x4114 | FLOAT | 0 |
| 6 | [3:6] | BUR\CPU1\RS1\CC100.iba.Analog_2 | 0x4118 | FLOAT | 0 |
| 7 | [3:7] | BUR\CPU1\RS1\CC100.iba.Analog_DI | 0x411C | DINT | 0 |
| 8 | [3:8] | BUR\CPU1\RS1\..RealCounter | 0x4120 | FLOAT | 6,42626e+06 |
| 9 | [3:9] | BUR\CPU1\RS1\..Rectangle | 0x4124 | FLOAT | 5 |
| 10 | [3:10] | BUR\CPU1\RS1\..Sine | 0x4128 | FLOAT | 9,27735 |
| 11 | [3:11] | BUR\CPU1\RS1\..Triangle | 0x412C | FLOAT | 23,9127 |
| 12 | [3:12] | BUR\CPU1\RS1\..Triangle2 | 0x4130 | FLOAT | 2,8072 |
| 13 | [3:13] | BUR\CPU1\RS1\..VarSine | 0x4134 | FLOAT | 4,70241 |
| 14 | [3:14] | BUR\CPU1\RS1\..VarSine2 | 0x4138 | FLOAT | -1,29307 |

*Example of the DGM200E data module on an DGM 200-E board (via Ethernet)*

## 5.6 HPCi Lite

Even if you are using Request HPCi you can still use HPCi Lite modules, because these modules are part of the DGM200P or DGM200E interface, which is mandatory for using Request HPCi.

With HPCi Lite you can measure the signals that are already available on the CC100/DGM200 bus.

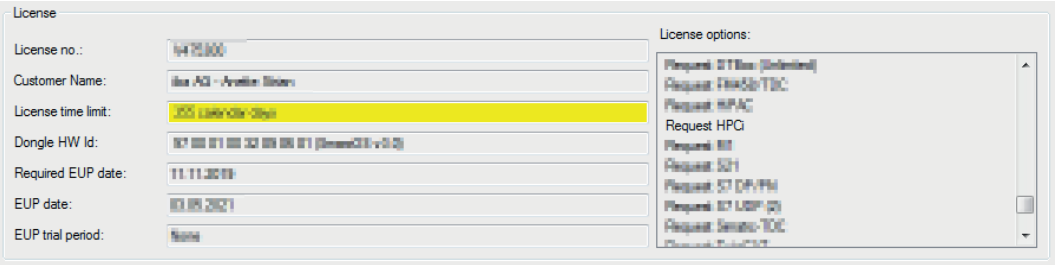
**Other documentation**

For a more detailed description of the HPCi Lite module please refer to the manual of *ibaPDA-Interface-HPCI-DGM200E*.

# 6 Diagnostics

## 6.1 Checking the license

If the "HPCi Request" modules are not available in the signal tree, you can either check in *ibaPDA* in the I/O Manager under *General* - *Settings* - *License* or in the *ibaPDA* service status application if your license is detected properly.




In addition to the license for request HPCi, other licenses must also be present, depending on which data path is to be used.

- For reflective memory: ibaPDA-Interface-Reflective-Memory
- For DGM200: ibaPDA-Interface-HPCI-DGM200P

## 6.2 Log files

If connections to target platforms or clients have been established, all connection-specific actions are logged in a text file. You can open this (current) file and, e.g., scan it for indications of possible connection problems.

You can open the log file via the button <Open log file>. The button is available in the I/O Manager:

- for many interfaces in the respective interface overview
- for integrated servers (e.g. OPC UA server) in the *Diagnostics* tab.

In the file system on the hard drive, you can find the log files of the *ibaPDA* server (`...\ProgramData\iba\ibaPDA\Log`). The file names of the log files include the name or abbreviation of the interface type.

Files named `interface.txt` are always the current log files. Files named `Interface_yyyy_mm_dd_hh_mm_ss.txt` are archived log files.

Examples:

- `ethernetipLog.txt` (log of EtherNet/IP connections)
- `AbEthLog.txt` (log of Allen-Bradley Ethernet connections)
- `OpcUAServerLog.txt` (log of OPC UA server connections)

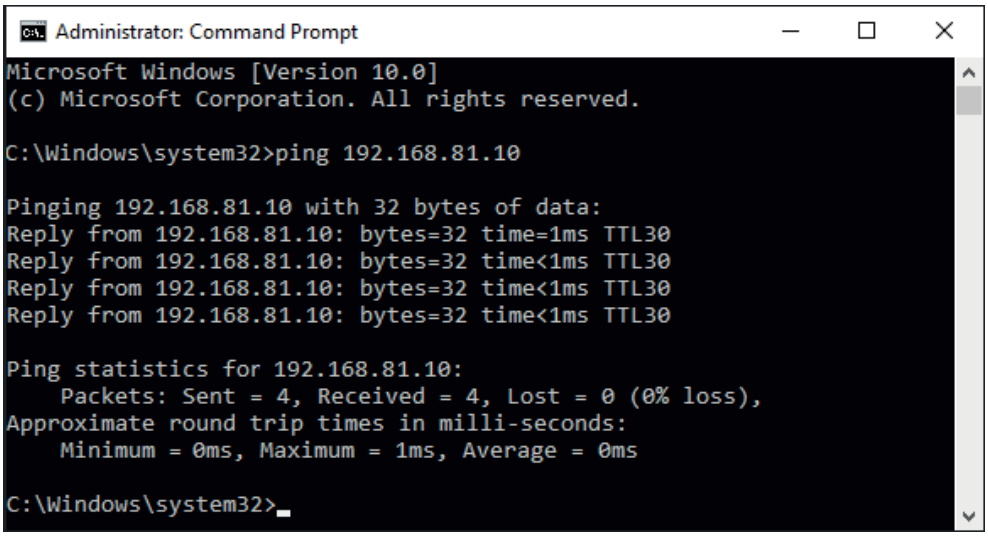
## 6.3 Connection diagnostics with PING

PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

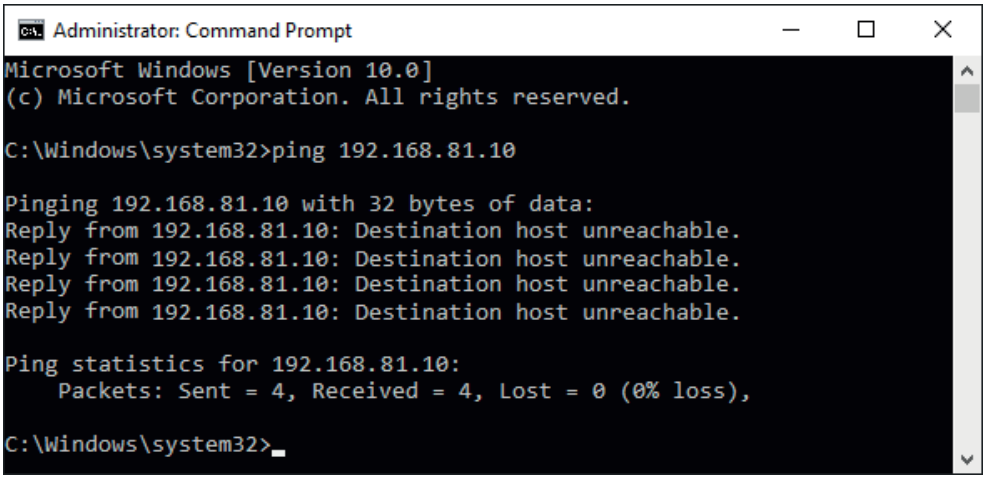
1. Open a Windows command prompt.


2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.




→ With no existing connection you receive error messages.




## 6.4 Module diagnostics

You will find a diagnostic help with a tabular display of the actual analog and digital values and the data types on the *Diagnostics* tab of each HPCi Request module.

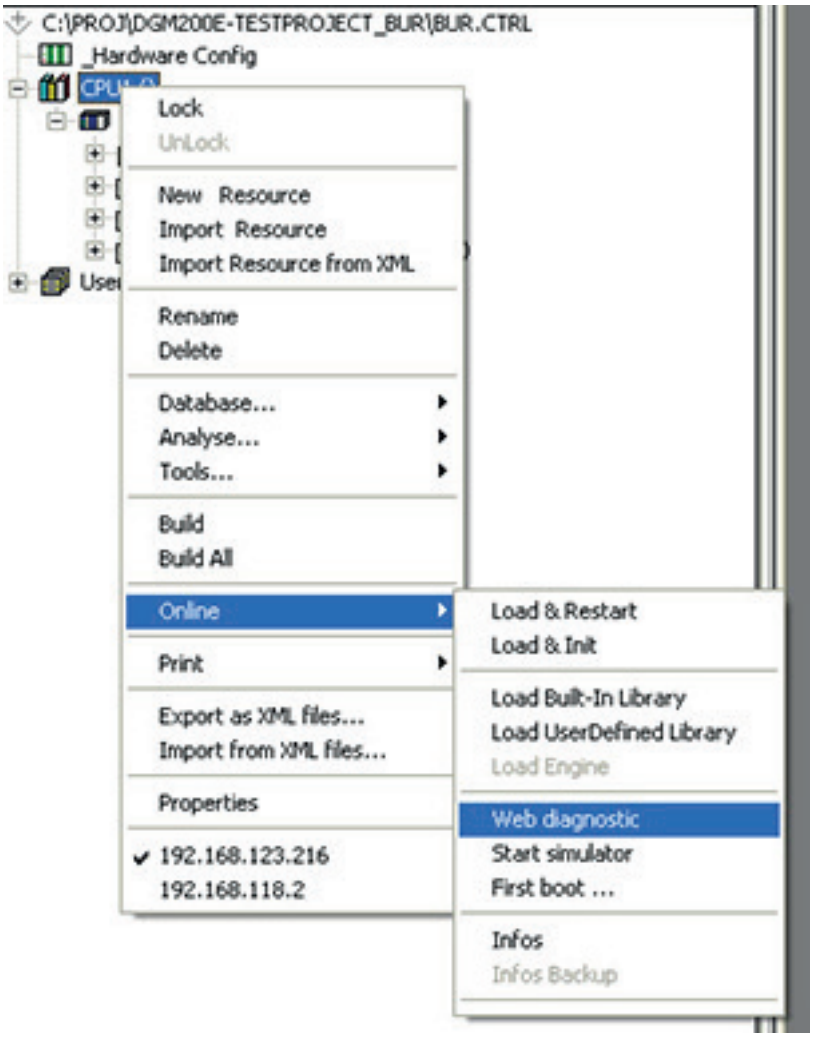
See chapter ↗ *Diagnostics*, page 33

## 6.5 Check the Historical Logger

In order to check if the DAS Agent has started on the HPCi Controller and if the *ibaPDA* system is connected you can use the web interface of the HPCi controller.

If connected to the HPCi controller via a network connection open your web browser and type in the URL of the HPCi controller (*http://IP-address*, e.g. http://192.168.120.215).

Another way to start the web interface is by using the P80i. Right click on the controller and select *Online* and then *Web diagnostic*.

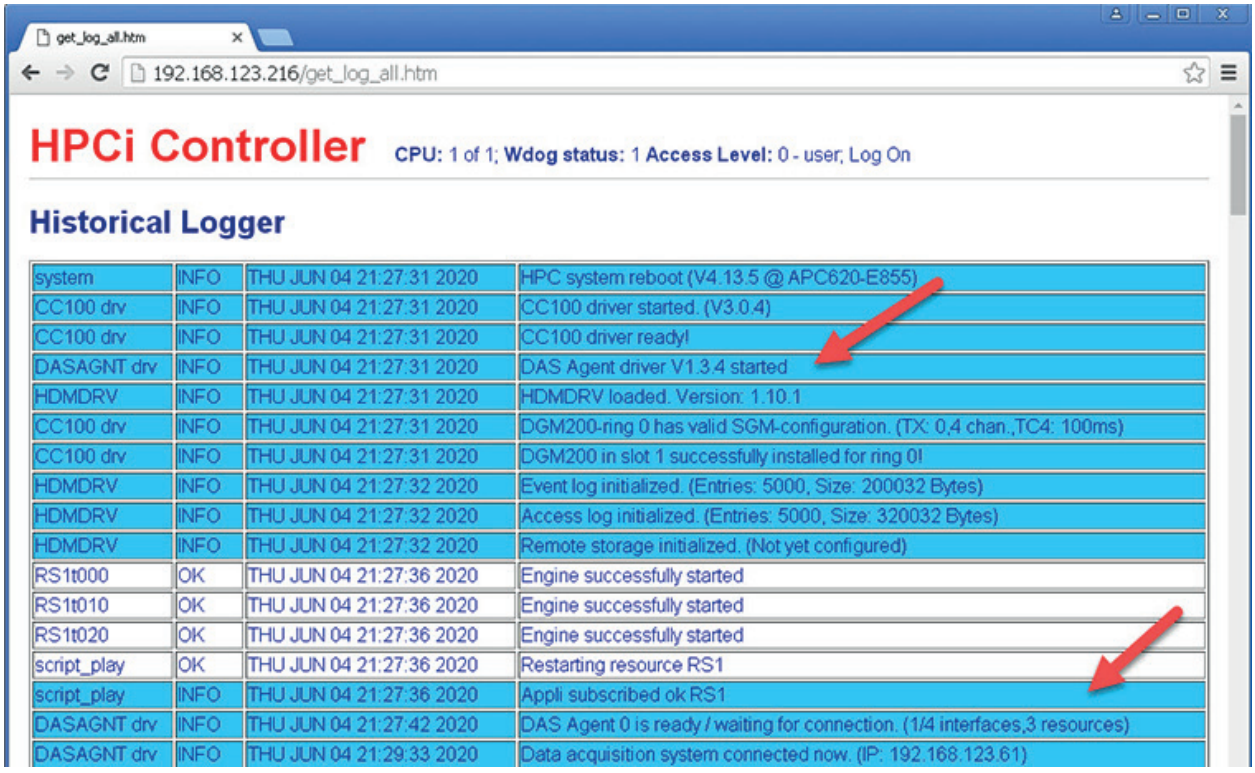



*Opening the web interface by P80i*

Under "System", click on the "Historical Logger" link


*Start page of the controller's web interface*

This will open the Historical Logger page.




*Historical Logger shows start of the DAS Agent driver and ibaPDA connection*

Here, you'll find, among others, entries with messages from the DAS Agent driver (DASAGNT drv).

Check that the DAS Agent driver is started. Also observe the message "Data acquisition system connected now". The IP address mentioned here should be the IP address of the associated *ibaPDA* server.

# 7 Support and contact

**Support**

Phone: +49 911 97282-14

Email: support@iba-ag.com

**Note**

If you need support for software products, please state the number of the license container. For hardware products, please have the serial number of the device ready.

**Contact**

**Headquarters**

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Email: iba@iba-ag.com

**Mailing address**

iba AG
Postbox 1828
D-90708 Fuerth, Germany

**Delivery address**

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

**Regional and Worldwide**

For contact data of your regional iba office or representative
please refer to our web site:

**www.iba-ag.com**