



Anybus CompactCom 40 Diagnostic Events for EtherNet/IP™

SCM-1202-037 1.1 ENGLISH

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

1 Preface

1.1 About this Document

This application note is intended to provide a description about how diagnostic events are presented in the engineering tool for the industrial network EtherNet/IP.

It is divided into two parts:

Part one provides a short overview of the Anybus CompactCom 40 Diagnostic Object (02h) and its diagnostic events. Part two is an example, showing how to get diagnostic messages displayed for EtherNet/IP using the PLC engineering tool Studio 5000 Logix Designer.

1.1.1 Target Audience

This document is meant for trained and skilled personnel working with the equipment described.

You need electrical engineering skills for the installation and commissioning of electrical equipment.

You also need general knowledge of automation and programmable logic controllers, in particular about Rockwell Automation software. Additionally knowledge on the EtherNet/IP Industrial Ethernet protocol, the Common Industrial Protocol (CIP) and the Anybus CompactCom (ABCC) object model is necessary.

1.2 History

Revision	Date	Description	Responsible
1.0	2016-10-17	First version	OLB
1.1	2017-02-03	Converted to DOX	KaD

1.3 Referenced Documents

Description	Name / Type	Version
HMS Starter Kit	HMS Development Board 2 Rev 1.07	0314-1.1.1
Anybus CompactCom 40 module	ABCC-M40-EIP	FW V.1.26
Allen Bradley PLC	1769-L27ERM-QBFC1B CompactLogix 5370 Controller	FW 1.010
EDS file for the Anybus CompactCom 40 EtherNet/IP	005A002B00370100.eds	3.11
Anybus CompactCom 40 EtherNet/IP	Network Guide	V2.0
Anybus CompactCom Software Design guide	Software Guide	V3.0
PC with Rockwell Automation PLC programming software	Studio 5000 Logix Designer	V26.01
IDE	KEIL uVision 5	V5.20
Anybus CompactCom Driver	Anybus CompactCom Host Application Example Code	V2.01

1.4 More Information about Networks and Products

The latest manuals and EDS files can be found on the HMS website, www.anybus.com

1.5 Trademark Information

Anybus® is a registered trademark of HMS Industrial Networks AB.

All other trademarks are the property of their respective holders.

2 Introduction

The ability of an automation device to raise diagnostic events is an important benefit. It may not only reflect a higher level of quality setting it apart from competitors in the market, in the end it will also increase the device's reliability, reduce downtimes by relieving preventive maintenance and protect the investment.

Implementing this feature set allows the device to signal significant incidents towards the PLC. This way operators will immediately be informed in case the device experiences errors or faults.

This document describes how developers of a field device implementing an Anybus CompactCom 40 EtherNet/IP network interface can use its features to create diagnostics events.

3 The Diagnostic Object

The Anybus CompactCom concept is based on an object model.

For detailed information about this, please refer to the Anybus CompactCom 40 Software Design Guide.

For creating diagnostics (information from a field device to a PLC) the CompactCom concept contains the diagnostic object (02h) which is located inside the CompactCom. A diagnostic event is created by sending a **create (03h)** command to the **diagnostic object (02h)** of the CompactCom.

3.1 Create a Diagnostic Event

When the CompactCom device has been started and initialized, the diagnostic object (02h) is solely built up of its object instance #0 containing attributes common to all subsequent instances which may be added in the course of the device's operation. The process of reporting a diagnostic event to the network is initiated by the host application. To this end a **create (03h)** event command message is sent from the host application to the diagnostic object (02h).

The CompactCom will internally create a new instance inside the diagnostic object (02h). This new instance corresponds to a **diagnostic event**.

The **create (03h)** command must contain info for describing the diagnostic event: a **Severity** (CMDExt[0]) and an **Event Code** (CMDExt[1]). The **severity** indicates how critical the event is and if it will recover by itself or not. CMDExt[0] additionally contains a bit called **Extended Diagnostic**, which informs the CompactCom if the event message contains additional user specific data. If the **Extended Diagnostic** bit is set, the additional user specific data will be found in MsgData[0..7], and the network specific data will follow in MsgData[8..n].

An **Event Code** informs about the nature of the event, i. e. what caused the device to react, like exceeding temperature or current limit.

Command Details: Create

Details

Command Code:	03h
Valid For:	Object

Description

Creates a new instance, in this case representing a new diagnostic event in the host application.

- Command details:

Field	Contents	Note
CMDExt[0]	Bit 0: Extended Diagnostic Bit 4–6: Severity Other bits: Reserved. Set to zero.	
CMDExt[1]	Event Code, see previous page	
MsgData[0...1]	Slot number associated with the event Set to "0" if unknown or unsupported	These fields only exist if bit 0 (Extended Diagnostic) is set
MsgData[2...3]	ADI associated with the event Set to "0" if unknown or unsupported	
MsgData[4]	Element associated with the event Set to "255" if unknown or unsupported	
MsgData[5]	Bit in element associated with the event Set to "255" if unknown or unsupported	
MsgData[6...7]	Reserved. Set to zero	
MsgData[0/8...n]	Network specific extension (optional, definition is network specific)	MsgData[8-n] if bit 0 in CmdExt[0] is set MsgData[0-n] if bit 0 in CmdExt[0] is not set

- Response details (Success):

Field	Contents
MsgData[0...1]	The number of the created instance

- Response details (Error):

Error	Contents	Comment
Object Specific Error	MsgData[1] = 02h	Error code (Latching event not supported) The event could not be created since the module does not support latching events
	MsgData[1] = FFh	Error code (Network specific error) The event could not be created due to a network specific reason. Information about the event is found in response MsgData[2-n]

3.2 Severity Codes of Diagnostic Events

The following severity codes are defined for the CompactCom:

Severity

This parameter indicates the severity level of the event. Only bits 4 - 6 are used for severity level information.

Severity Levels

Bit Combination	Severity	Comment
000	Minor, recoverable	-
001	Minor, unrecoverable	Unrecoverable events cannot be deleted
010	Major, recoverable	-
011	Major, unrecoverable	Causes a state-shift to EXCEPTION
101	Minor, latching	
110	Major, latching	
(other)	-	(reserved for future use)

Typically, *recoverable* events are generated by the process e.g. if a temperature exceeds a limit value defined by the device manufacturer (e.g. internal temperature of the device exceeds 50° C). The character of this event is typically a warning when the event is defined as *minor*. The temperature of the device is recoverable as the device can cool down again when some heat producers are cut off.

The device manufacturer will add a *major recoverable* event when he wants to inform the PLC that the temperature has reached a critical high temperature which can damage the device.

An ***unrecoverable*** diagnostic event is typically created when the device detects that a sensor is broken. The sensor has to be replaced, it will - normally – not recover by itself. If the device has only one sensor and this sensor is broken the device will create a ***major unrecoverable*** event. The device has to be stopped and to be repaired (replace the broken sensor) before used again otherwise there will be a big risk that the device will be damaged after restart. If the device has some redundant sensors it will create a ***minor unrecoverable*** event informing the users that the broken sensor should be replaced by another one within the next scheduled inspection.

Minor latching and major latching allow the creation of *latching diagnostic event*.

In EtherNet/IP, latching events are not supported.

For more information, see the Software Design Guide and the Network Guide of the Anybus CompactCom 40 device.



The device manufacturer has to define which event will be reported as a diagnostic event to the PLC and which severity has to be used.

3.3 Anybus CompactCom Event Codes

The table below shows a list of the event codes applicable for Anybus CompactCom 40 device.

Event Codes

#	Meaning	Comment
10h	Generic Error	-
20h	Current	-
21h	Current, device input side	-
22h	Current, inside the device	-
23h	Current, device output side	-
30h	Voltage	-
31h	Mains Voltage	-
32h	Voltage inside the device	-
33h	Output Voltage	-
40h	Temperature	-
41h	Ambient Temperature	-
42h	Device Temperature	-
50h	Device Hardware	-
60h	Device Software	-
61h	Internal Software	-
62h	User Software	-
63h	Data Set	-
70h	Additional Modules	-
80h	Monitoring	-
81h	Communication	-
82h	Protocol Error	-
90h	External Error	-
F0h	Additional Functions	-
FFh	NW specific	Definition is network-specific; consult separate network guide for further information.

The event code **FFh** is used if network specific diagnostics are reported (not considered within this application note).

4 EtherNet/IP Diagnostics

This section shows how EtherNet/IP handles diagnostic data.

In EtherNet/IP diagnostic events will affect the Device Status attribute #5 of the CIP **Identity Object** (01h) and the module status LED of the Anybus CompactCom. According to the CIP specification the Device Status attribute #5 has two types of status, a **general device status** and an **extended device status**.

The general device status indicates typical errors and the extended device status can offer details about vendor specific error. If used, the device must implement a vendor specific extended device status field definition. The vendor specific extended device status is not part of the current implementation of the Anybus CompactCom 40 EtherNet/IP device .

The following status bits stands for general device status:

- Bit 8: Minor recoverable fault
- Bit 9: Minor unrecoverable fault
- Bit 10: Major recoverable fault
- Bit 11: Major unrecoverable fault

In CIP instance 1, attribute 5 “status” diagnostic information can be read out. The status attribute is of data type “word” with the following bit definitions as defined within the CIP specification:

Bits	Called	Definition
0	Owned	If set the device has an owner master/scanner has allocated module
1	-	Reserved
2	Configured	This bit tells if the configuration of a feature or functionality of a product that works immediately after installation has been modified
3	-	Reserved
4...7	Extended device status	Default values (see Tab. 2) or applied to vendor specific information
8	Minor recoverable fault	If set the device has detected a problem with itself that can be recovered. The event that sets the bit doesn't cause a state change
9	Minor unrecoverable fault	If set the device has detected a problem with itself that cannot get rid of alone. The event that sets the bit doesn't cause a state change
10	Major recoverable fault	This bit is set to true if the device detected a problem with itself and enter into a major recoverable fault state
11	Major unrecoverable fault	This bit is set to true if the device detected a problem with itself and enter into a major unrecoverable fault state
12...15	Extended device	Reserved or for vendor specific information

Table 1: Bit definition for status instance of identity object

Value of bits 4...7	Description
0000b	Self-testing or unknown
0001b	Firmware update in progress
0010b	At least one faulted I/O connection
0011b	No I/O connections established
0100b	Non-volatile configuration bad
0101b	Major Fault – either bit 10 or bit 11 is true
0110b	At least one I/O connection in run mode
0111b	At least one I/O connection established
1000 and 1001	Reserved
1010 through 1111	Vendor specific

Table 2: Default values for extended device status field (bits 4-7) in status instance attribute

In the Anybus CompactCom 40 EtherNet/IP device, the extended device status defines the following status bit field as follows:

Value of bit(s)	Name
0000b	Unknown
0010b	Faulted I/O connection
0011b	No I/O connections established
0100b	Non-volatile configuration bad
0101b	Major fault
0110b	Connection in run mode
0111b	Connection in idle mode
(others)	Reserved

The next section will show how to setup a configuration of EtherNet/IP network using the CompactLogix 5370 L2 controller.

4.1 Configuration

The PLC system hardware configuration and the EtherNet/IP network configuration are done with the Studio 5000 Logix Designer tool. In the first step the PLC must be configured, then the EtherNet/IP network. In the following we will make a step-by-step configuration.

4.1.1 PLC Configuration

Create a new project. Open the file menu and select new. Select the type of PLC in the dialog box that appears, here a 1769-L27ERM-QBFC1B. Enter the name of the project and click next.

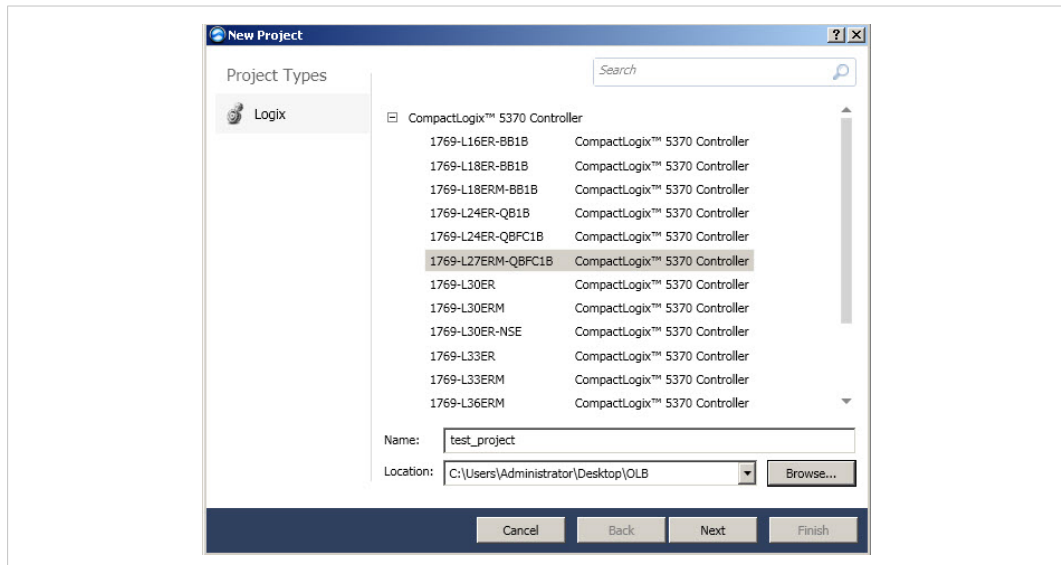


Fig. 1 Adding the PLC to the configuration

Select the revision if needed in the next dialog box and click finish.

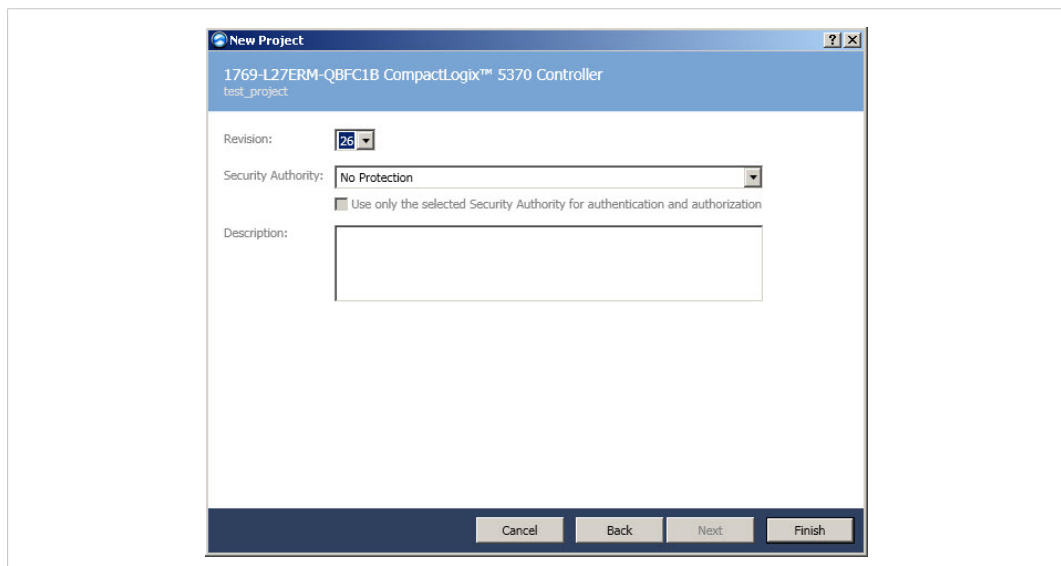


Fig. 2 Selecting the revision

The navigation list will look like below.

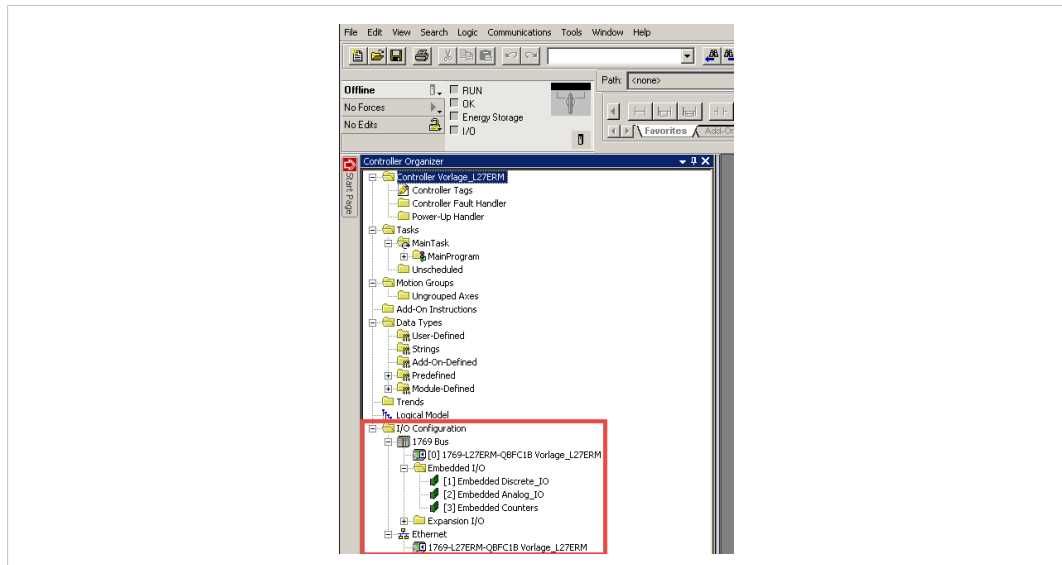


Fig. 3 Navigation list

At this stage, import an EDS file to your project for Anybus CompactCom 40 EtherNet/IP module.

To do this select the menu Tools and click on EDS Hardware Installation Tool. Follow the wizard instruction to import the EDS file.

4.1.2 EtherNet/IP Network Configuration

Add the Anybus CompactCom device to the configuration in the PLC. Start by setting the program in Offline mode (1). Then right-click on the EtherNet/IP Bridge in the I/O configuration, and select New Module as shown below (2).

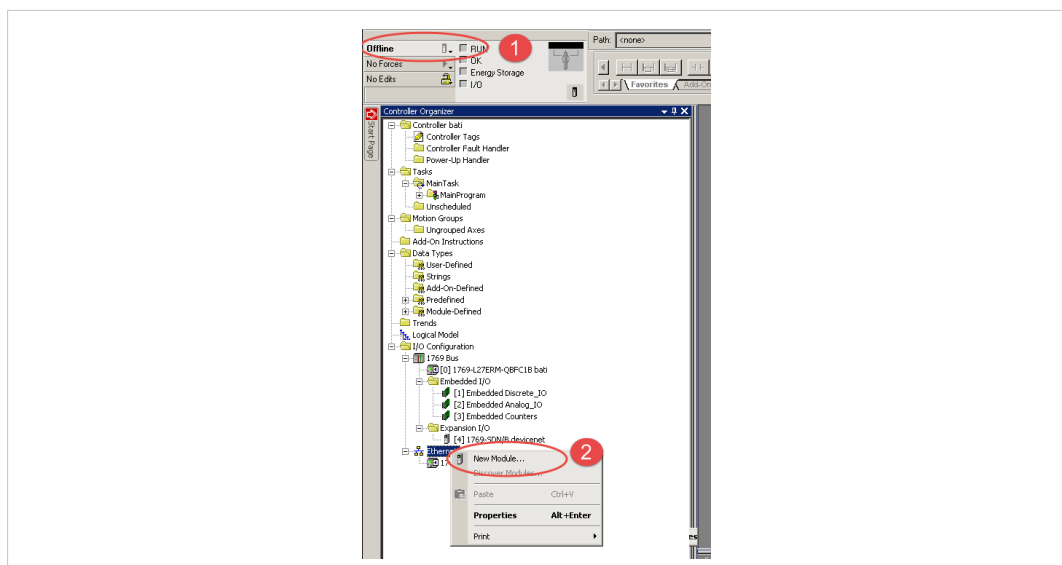


Fig. 4 Adding the CompactCom device

A dialog window will appear. In this dialog box, select Anybus CompactCom 40 EtherNet/IP™ and click Create.

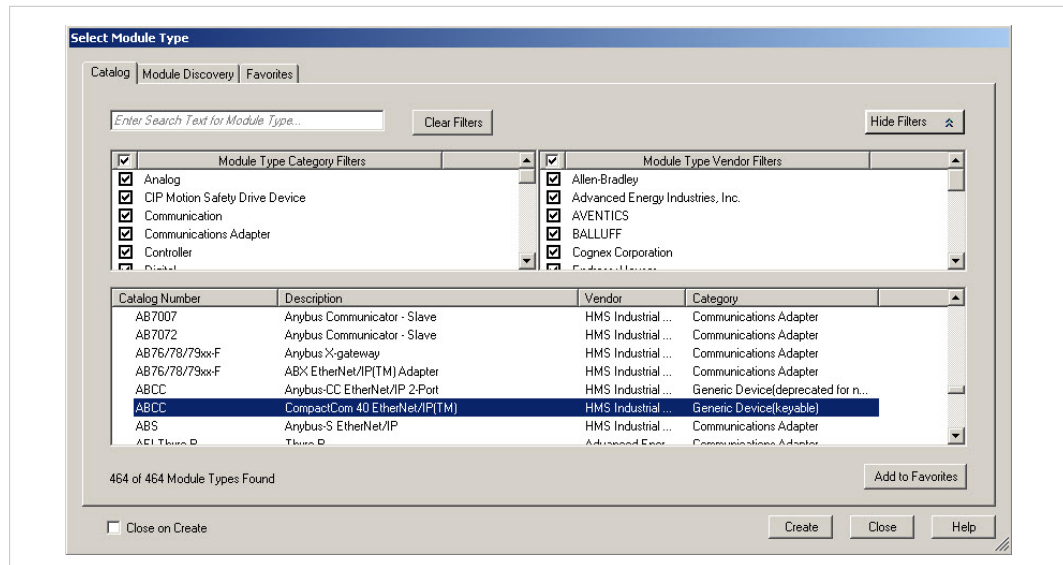


Fig. 5 Selecting module type

Give the module a name and click **Change** in the module definition field to alter the input/output data sizes. See the image below.

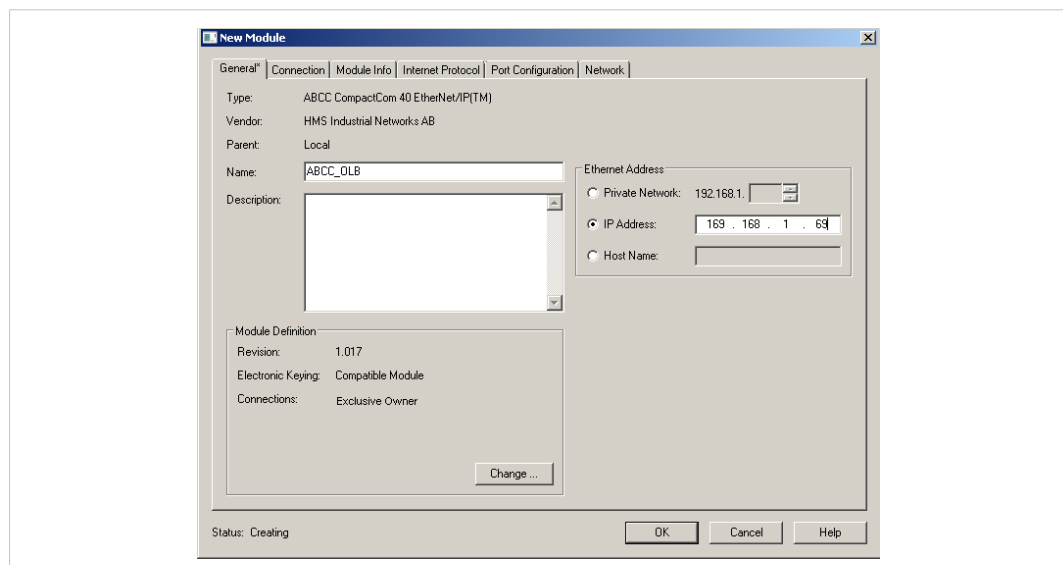


Fig. 6 The New Module window

The module definition window will now appear, in which the type of connection, the size and data type for the input/output data of the Anybus CompactCom module should be set. In this example we use an Exclusive Owner connection and set the data type to INT, which represents the data as 16-bit values. The size of the input and output connections must correspond to the size configured for the Anybus CompactCom module.

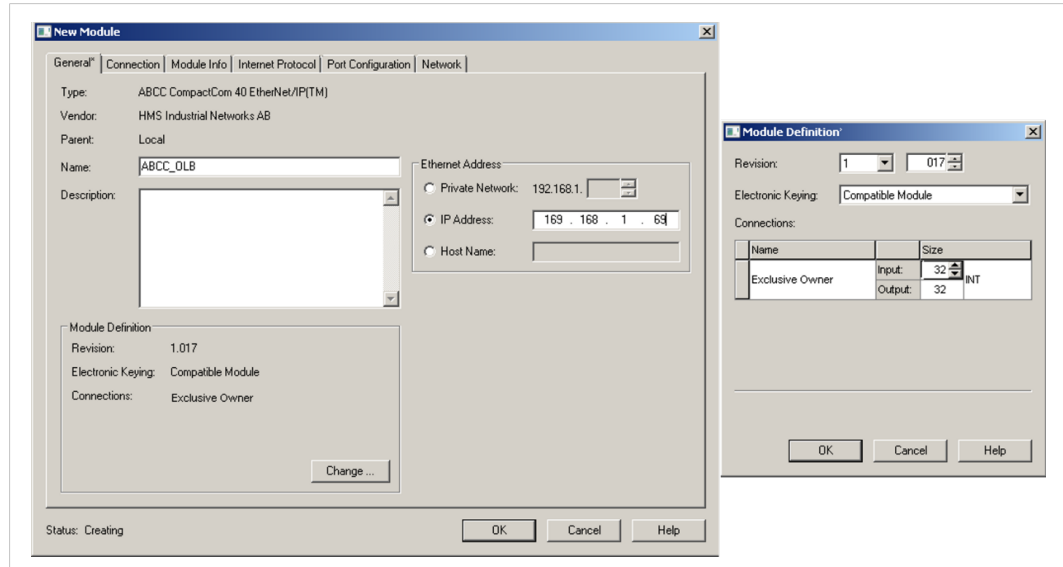


Fig. 7 Module Definition

Click OK.

Finally, the IP address configured for the module should be entered - 192.168.1.69 in this example.

The IP address should be in the same range as the IP address for the PLC system.

Click OK once more to proceed.

In the controller tags for the PLC (to the left), the configured I/O data values for the implemented Anybus CompactCom Module can be seen.

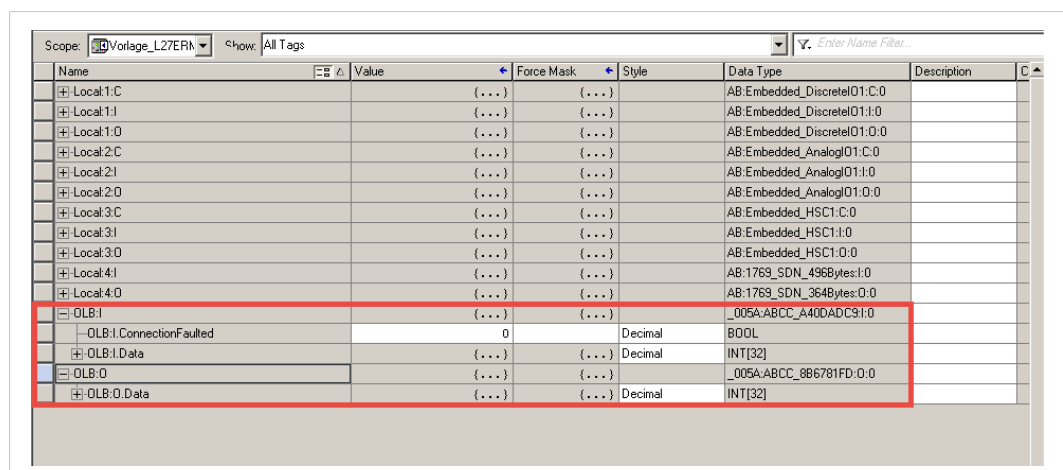


Fig. 8 Configured values

Downloading the Configuration to the PLC

First select the communication path, which can be done by opening the Communications menu and selecting the **Who Active** command. Select the desired communication path as seen below. It may differ in your project.

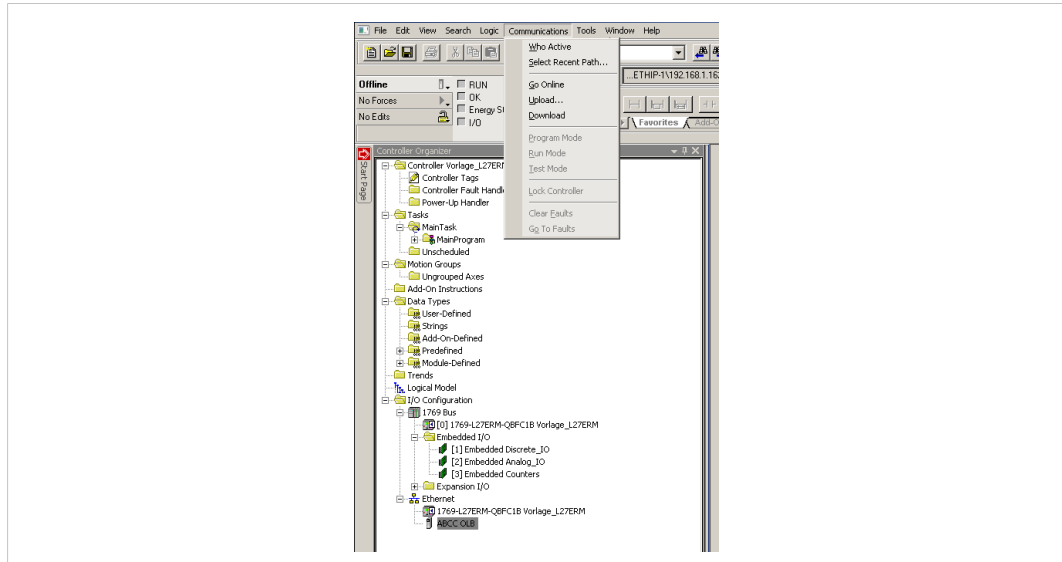


Fig. 9 Configuring the communication path

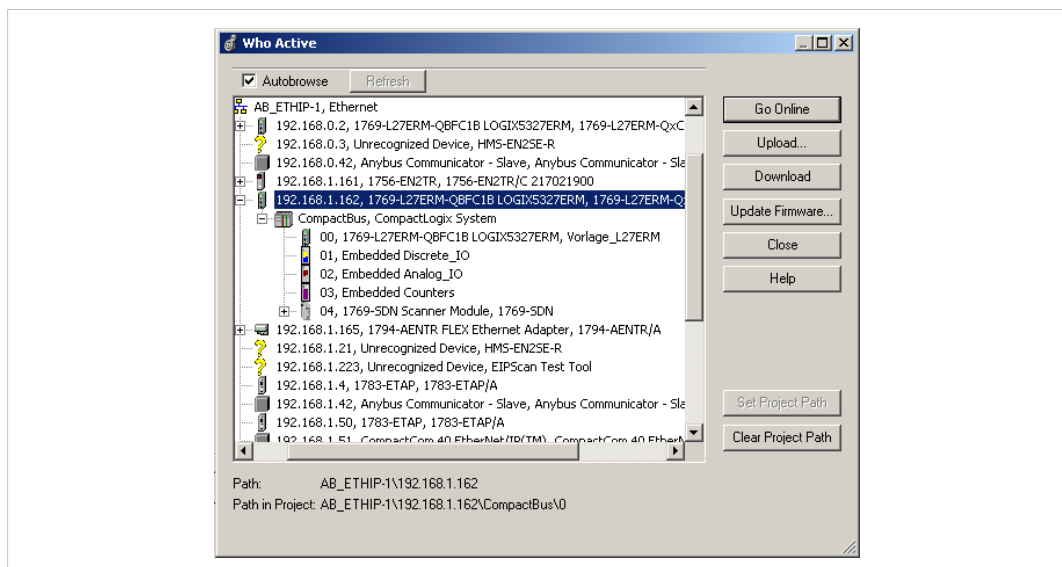


Fig. 10 The Who Active window

Select *Go Online* from the communications menu. In the new window that appears, select Download.

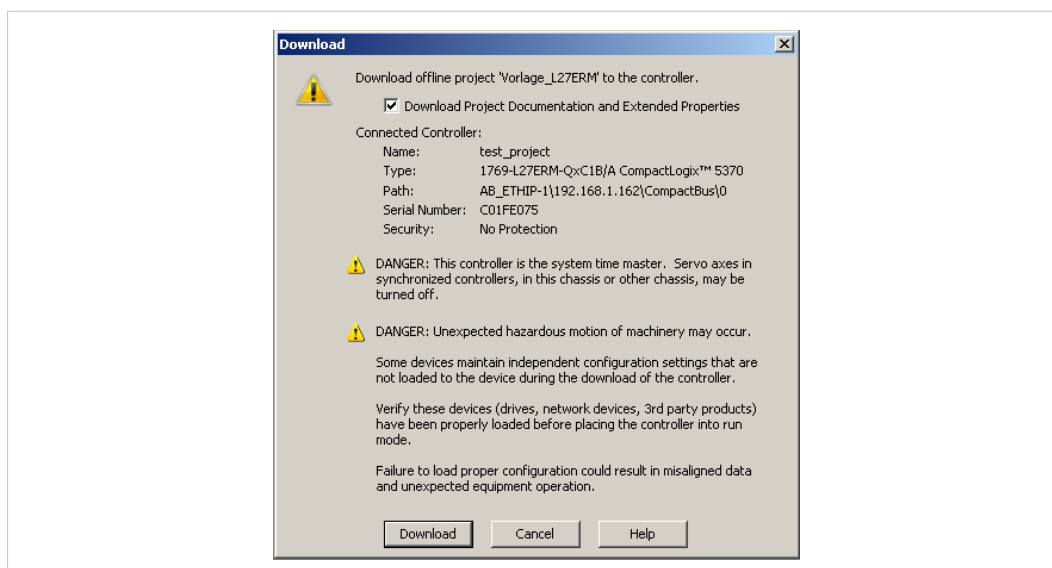


Fig. 11 Downloading the configuration to the PLC

5 Application Example

In this example we are using the Anybus CompactCom Starter Kit in which we integrate the Anybus CompactCom 40 device. We use the host application example code for the Windows platform simulating the host application. We also run the engineering tool Studio 5000 Logix Designer by Rockwell for configuring the PLC. In this example we employ a CompactLogix 5370 L2 controller. The engineering tool will also report the diagnostic events sent by the CompactCom device to the PLC.

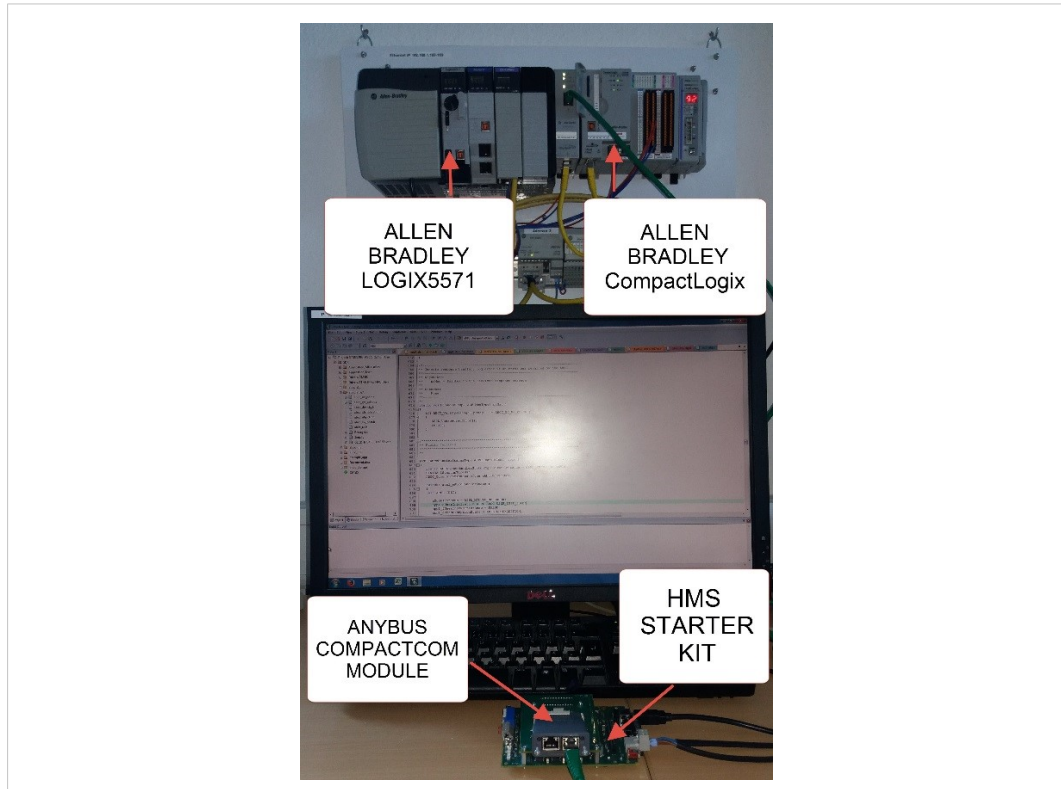


Fig. 12 Hardware Connection Overview

The following section will explain what has to be done from application side to initiate the CompactCom Ethernet module to send field device diagnosis to the network master.

5.1 Code Sample for Creating a Diagnostic Event

This sample is intended to show how to create a diagnostic event in the CompactCom using the CompactCom host application example code. The example below shows the structure of a **create (03h) event command message** that generates a diagnostic event defined as *major recoverable* for the event code *voltage*.

Section 6.1 shows how this event is displayed in Studio 5000 Logix Designer.

```
// message header part
ABCC_SetMsgHeader (psMsg,                                // buffer
                  ABP_OBJ_NUM_DI,                         // diagnostic object
                  0,                                       // instance
                  0,                                       // attribute
                  ABP_CMD_CREATE,                         // command type
                  0,                                       // Message data size
                  ABCC_GetNewSourceId ());                // source id

// severity
psMsg->sHeader.bCmdExt0 = BP_DI_EVENT_SEVERITY_MAJOR_REC;

// event code
psMsg->sHeader.bCmdExt1 = ABP_DI_EVENT_VOLTAGE;

// message data part
ABCC_SetMsgData16 (psMsg, 0, 0);
```

The *create (03h)* event command above has created a diagnostic event including the severity code (**major recoverable (0x20)**) and the event code (**voltage (0x30)**). There is no data included in the message in this case.

Note:

For EtherNet/IP the event code cannot be represented on the network and it is ignored by the CompactCom device.

The severity level of all instances are combined (using logical OR) and **represented on the network through the CIP identity object (01h)**.

The engineering tool will display only the severity codes. See in the software design guide and network guide for more details.

6 Readout of Diagnostics in Studio 5000 Logix Designer

Start your project in Studio 5000 Logix Designer with all the settings done including your Anybus CompactCom 40 EtherNet/IP device. To be able to read out diagnostic information about the slave module, click on the icon representing a slot (1) and select *go online*.

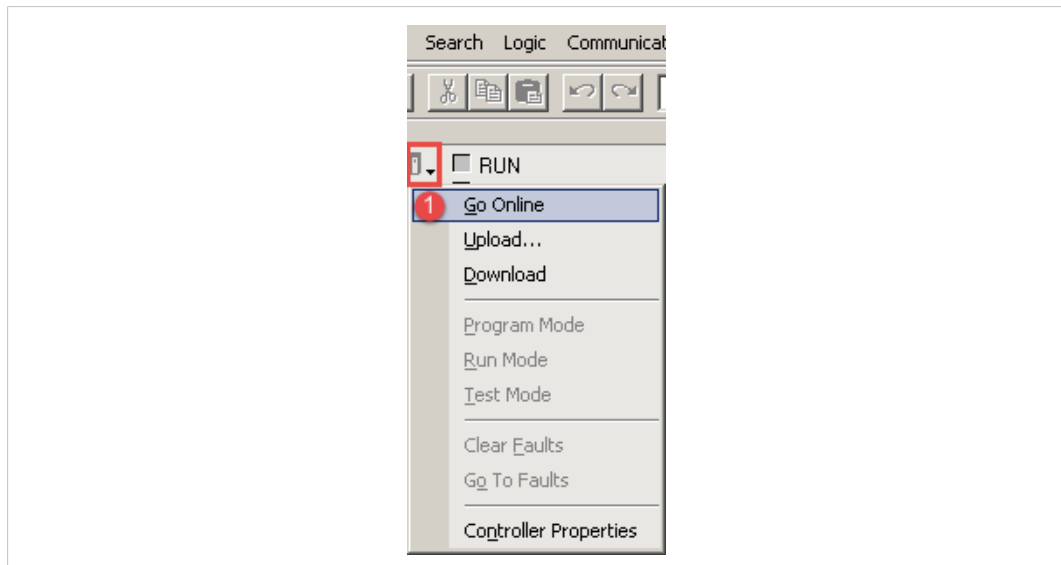


Fig. 13 Go online

In the I/O configuration, double click (2) on the slave module icon and then select the tab *module info* (3). This tab (4) provides general information about the slave module as well as the current status of the device. The figure below shows a running slave module without any error.

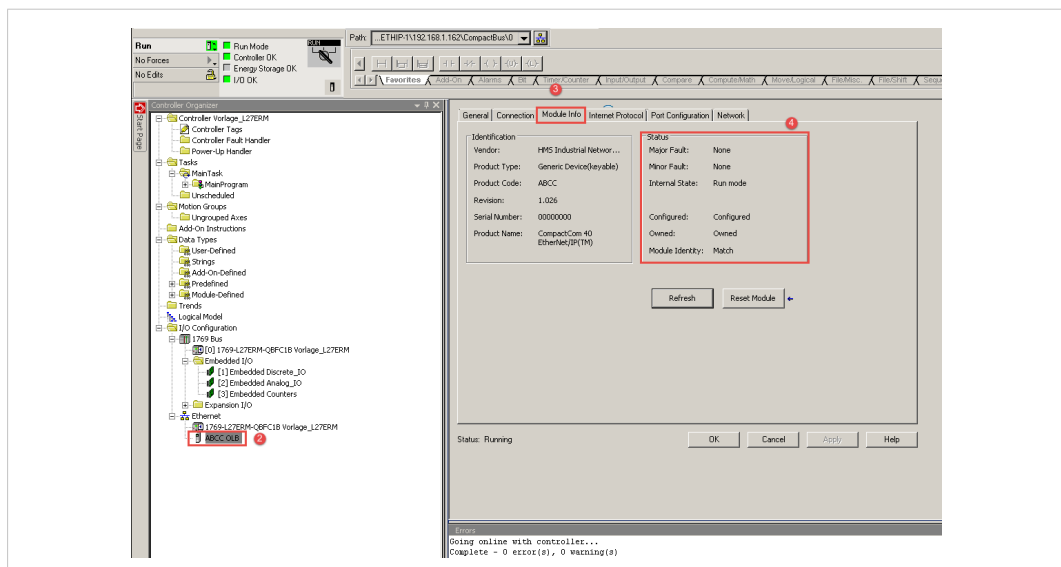


Fig. 14 Studio 5000 Logix Designer diagnostics readout

6.1 Severity: Minor Recoverable and Major Recoverable

Creation of a diagnostic event

The figure below shows one active diagnostic event defined as *minor recoverable*.

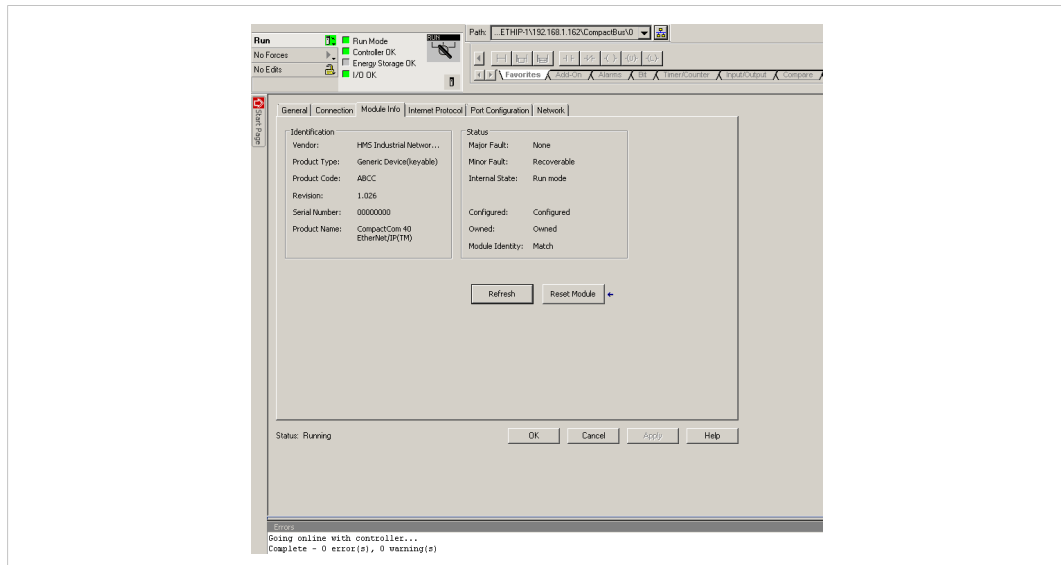


Fig. 15 Minor recoverable

The figure below shows two active diagnostic events: one defined as *major recoverable* and the other as *minor recoverable*. The only information of the diagnostic we get is the severity code. The **difference** to the figure above is the value for the internal status which reflects the extended device status field (bits 4-7). It has changed from *run mode* to *major fault*.

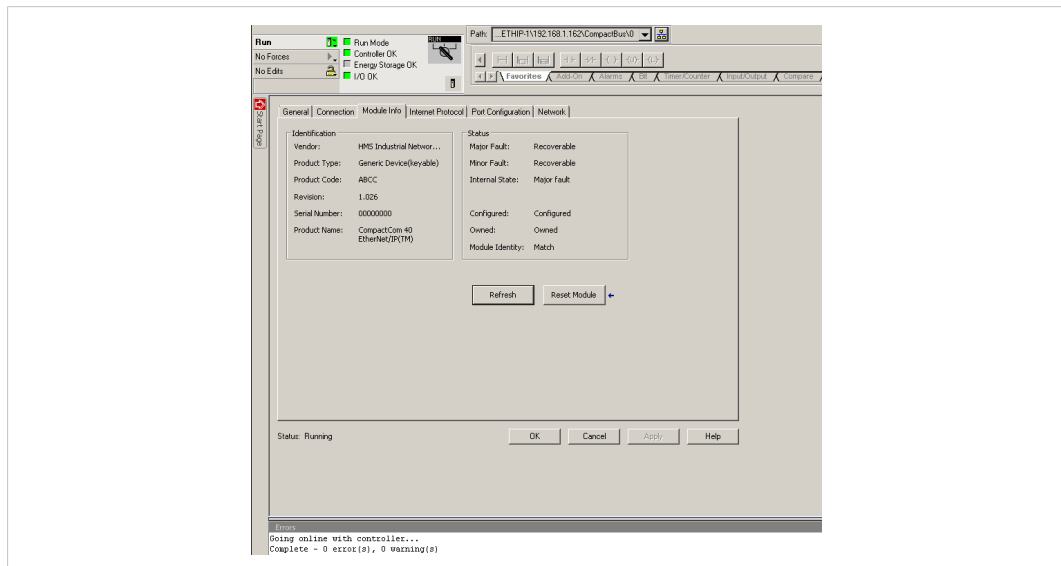


Fig. 16 Minor and major recoverable

Deletion of a diagnostic event

The figure below shows the status after deleting a diagnostic event defined as *minor recoverable*. Because it is minor recoverable, there is no change of the internal state since the major recoverable diagnostic event is still active. Only after deleting the major recoverable event, the internal state will revert to the original state as show in FIGURE 2

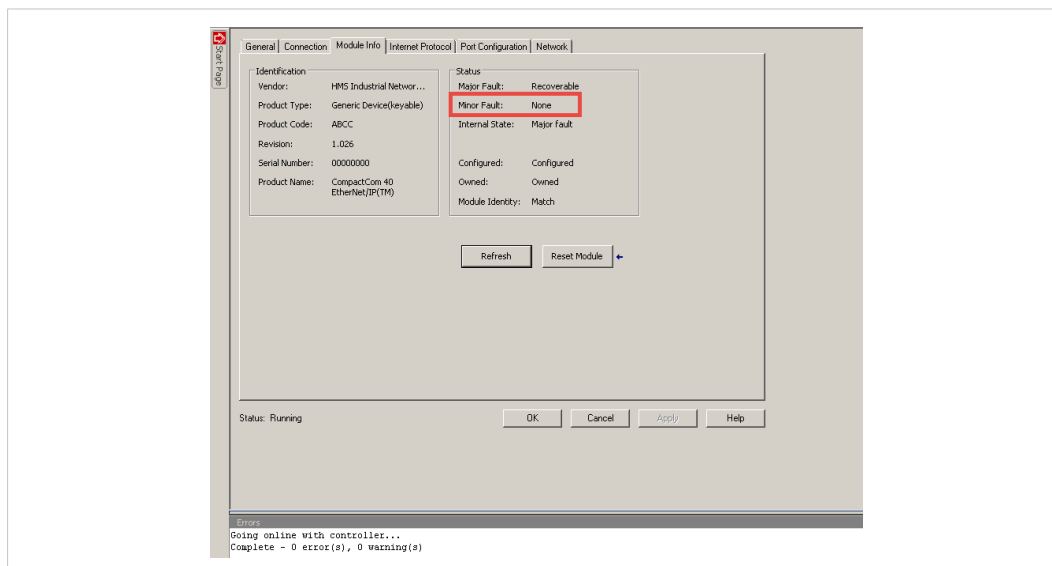


Fig. 17 Deletion of minor recoverable

6.2 Severity: Minor Unrecoverable

Figure 6 shows an active diagnostic event defined as *minor unrecoverable*. This diagnostic event cannot be deleted. The action to take for deleting the diagnostic event is to restart the CompactCom device by resetting the device.

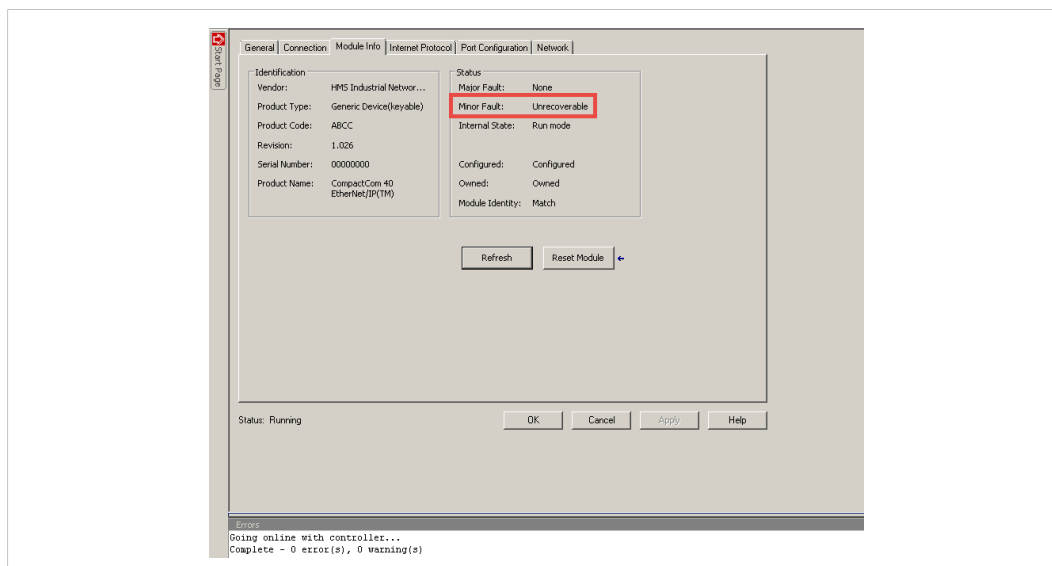


Fig. 18 Minor unrecoverable diagnostic message

6.3 Severity: Major Unrecoverable

Creating a major unrecoverable diagnostic event causes the Anybus CompactCom 40 device to enter EXCEPTION state. It also results into a disconnection of the device from the network. It is not possible to report any diagnostic message by means of the network. This is confirmed by fig. 7. The diagnostic window indicates: the device is not available and a timeout message is sent out to indicate that the communication was interrupted with the network.

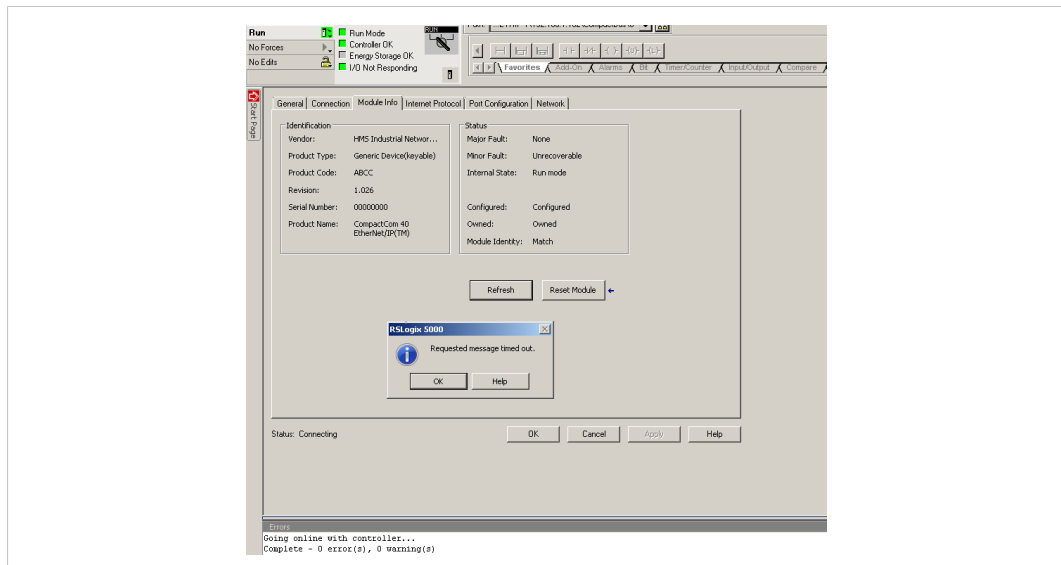


Fig. 19 Disconnection from network

