# Anybus® CompactCom™ 40

## PROFINET IRT Transparent Ethernet

**NETWORK GUIDE**

**HMS** Connecting Devices™

# Important User Information

## Disclaimer

The information in this document is for informational purposes only. Please inform HMS Networks of any inaccuracies or omissions found in this document. HMS Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Networks and is subject to change without notice. HMS Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the 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.

# Table of Contents

# 1 Preface

## 1.1 About this document

This document is intended to provide a good understanding of the functionality offered by the Anybus CompactCom 40 PROFINET IRT. The document describes the features that are specific to Anybus CompactCom 40 PROFINET IRT. For general information regarding Anybus CompactCom, consult the Anybus CompactCom design guides.

The reader of this document is expected to be familiar with high level software design and communication systems in general. The information in this network guide should normally be sufficient to implement a design. However if advanced PROFINET specific functionality is to be used, in-depth knowledge of PROFINET networking internals and/or information from the official PROFINET specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the PROFINET specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

For additional related documentation and file downloads, please visit the support website at www.anybus.com/support.

## 1.2 Related Documents

| Document | Author | Document ID |
|---|---|---|
| Anybus CompactCom 40 Software Design Guide | HMS | HMSI-216-125 |
| Anybus CompactCom M40 Hardware Design Guide | HMS | HMSI-216-126 |
| Anybus CompactCom B40 Design Guide | HMS | HMSI-27-230 |
| Anybus CompactCom Host Application Implementation Guide | HMS | HMSI-27-334 |
| Anybus CompactCom 40 PROFINET IRT Network Guide | HMS | SCM-1202-023 |
| PROFINET IO specification, rev. 2.3 | Profibus International | |
| PROFIenergy Technical Specification, rev. 1.2 | Profibus International | |
| GSDML Technical Specification for PROFINET IO, rev 2.31 | Profibus International | |

## 1.3 Document History

| Version | Date | Description |
|---|---|---|
| 1.0 | 2016-10-28 | First public release |
| 1.1 | 2016-11-23 | Modular Device<br>Minor Corrections |
| 1.2 | 2017-01-18 | Minor corrections |
| 1.3 | 2017-03-15 | PROFIenergy profile added<br>Asset management and I&M5 added |
| 1.4 | 2017-04-10 | Updated to PROFIenergy 2<br>Minor corrections |
| 1.5 | 2017-07-11 | Added appendix on backward compatibility<br>Added appendix on AIDA LED mode |
| 1.6 | 2017-11-28 | Minor corrections |
| 1.7 | 2017-12-15 | Updated Copyright Appendix |
| 1.8 | 2018-06-18 | SYNC functionality description updated<br>Minor corrections |
| 2.2 | 2018-10-23 | Minor updates |
| 2.3 | 2019-06-10 | Rebranding<br>Minor updates |
| 2.4 | 2019-12-20 | S2 redundancy<br>Minor updates |
| 2.5 | 2021-09-02 | Minor updates |

## 1.4        Document Conventions

Numbered lists indicate tasks that should be carried out in sequence:

1.    First do this

2.    Then do this

Bulleted lists are used for:

•      Tasks that can be carried out in any order

•      Itemized information

▶      An action

　　→    and a result

**User interaction elements** (buttons etc.) are indicated with bold text.

```
Program code and script examples
```

Cross-reference within this document: *Document Conventions, p. 7*

External link (URL): www.hms-networks.com

| ⚠ | **WARNING**<br>Instruction that must be followed to avoid a risk of death or serious injury. |
|---|---|

| ⚠ | **Caution**<br>Instruction that must be followed to avoid a risk of personal injury. |
|---|---|

| ❗ | Instruction that must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk. |
|---|---|

| ⓘ | *Additional information which may facilitate installation and/or operation.* |
|---|---|

## 1.5        Terminology

•      The terms "Anybus" or "module" refers to the Anybus CompactCom module.

•      The terms "host" or "host application" refer to the device that hosts the Anybus.

•      Hexadecimal values are written in the format NNNNh or 0xNNNN, where NNNN is the hexadecimal value.

•      A byte always consists of 8 bits.

•      The terms "basic" and "extended" are used to classify objects, instances and attributes.

## 1.6        Trademarks

Anybus® is a registered trademark of HMS Industrial Networks.

All other trademarks are the property of their respective holders.

# 2      About the Anybus CompactCom 40 PROFINET IRT

## 2.1      General

The Anybus CompactCom 40 PROFINET IRT communication module provides instant PROFINET Real Time connectivity via the patented Anybus CompactCom host interface. Any device that supports this standard can take advantage of the features provided by the module, allowing seamless network integration regardless of network type.

> **!** This network guide covers the Transparent Ethernet version of the product. Transparent Ethernet has to be enabled during setup, or the device will appear as an Anybus CompactCom 40 with full IT functionality. The IT functionality is described in the network guide for the standard Anybus CompactCom 40.
>
> IT functionality is not available when Transparent Ethernet is enabled.

This product conforms to all aspects of the host interface for Active modules defined in the *Anybus CompactCom Hardware- and Software Design Guides*, making it fully interchangeable with any other device following that specification. Generally, no additional network related software support is needed, however in order to take advantage of advanced network specific functionality, a certain degree of dedicated software support may be necessary.

## 2.2      Features

- Transparent Ethernet
- Ethernet or fibre optics connectors
- Up to 128 submodules in total
- Up to 32767 ADIs
- Max. read process data: 1308 bytes
- Max. write process data: 1308 bytes
- Max. process data (read + write, in bytes): 2616 bytes
- Generic and PROFINET specific diagnostic support
- Complies with PROFINET IO Conformance class C
- Supports up to 1440 bytes I/O data in each direction, status bytes included.
- Supports 250 µs cycle time
- SNMP agent
- Device identity customization
- Modular Device Functionality
- PROFIenergy profile supported
- GSD file template provided by HMS
- Supports PROFINET Fast Start Up
- Supports Media Redundancy Protocol (MRP)
- Supports S2 system redundancy
- Supports PROFIsafe

## 2.3        Transparent Ethernet

Transparent Ethernet offers the possibility for a host application, that includes an IT implementation (web pages, file system, a proprietary protocol etc.), to let the Anybus CompactCom handle an industrial Ethernet protocol (in this case PROFINET), without the need for extra Ethernet ports.

Ethernet communication is routed straight to the host application system using an RMII interface. The host application must include an Ethernet controller and a TCP/IP stack. PROFINET protocol messages will be routed to the Anybus CompactCom internal software. Please note that the Transparent Ethernet functionality has to be enabled during startup by setting attribute #16 (instance #1) in the Anybus Object.

16–bit parallell mode can not be used, as specific host application connector pins are reserved for transparent Ethernet. Also TCP/UDP ports may be reserved, and can, in that case, not be used for the transparent Ethernet communication.

See also ...

- *Transparent Ethernet, p. 41*
- Anybus CompactCom 40 Hardware Design Guide
- *Anybus Object (01h), p. 46*

# 3        Basic Operation

## 3.1      General Information

### 3.1.1    Software Requirements

Generally, no additional network support code needs to be written to support the Anybus
CompactCom 40 PROFINET IRT, however due to the nature of the PROFINET networking system
certain things must be taken into account:

- Up to 32767 ADIs can be represented on PROFINET.

- ADI names, types and similar attributes cannot be accessed via PROFINET. They are
  however represented on the network through the built in web server.

- Up to 5 diagnostic instances can be created by the host application. An additional 6th
  instance may be created in event of a major fault.

- For conformance reasons, the host application must implement support for network reset
  types 00h (Power-on) and 02h (Power-on + Factory Default) in the Application Object (FFh).

- PROFINET in itself does not impose any particular timing demands when it comes to acyclic
  requests (i.e. requests towards instances in the Application Data Object), however it is
  generally recommended to process and respond to such requests within a reasonable time
  period (exactly what this means in practice depends on the implementation and the actual
  installation).

- The order in which ADIs are mapped to Process Data is significant and must be replicated in
  the IO Controller when setting up the network communication. In case of a configuration
  mismatch, see *Configuration Mismatch, p. 22* for more information.

See also ...

- *Application Data Instances (ADIs), p. 15*

- Diagnostic Object, *Anybus Module Objects, p. 45*

- Anybus CompactCom 40 Software Design Guide, Application Data Object (FEh)

### 3.1.2    Electronic Data Sheet (GSD)

On PROFINET, the characteristics of a device is stored in an XML data file. This file, referred to as
the "GSD" file, is used by PROFINET engineering tools when setting up the network configuration.
HMS Industrial Networks provides an example GSD file, which must be adapted by the user to
suit the application. To adapt the GSD file, use the HMS PROFINET GSD Generator Tool which can
be downloaded from www.anybus.com.

> (i)   *A GSD checker tool and a GSD upgrade tool is available to PI members on the PI website.*

**Setting Identity and Function Information**

The GSD file must be adapted to your implementation. First thing is the device identity.

In the GSD file there is a section called "DeviceIdentity". It looks like this.

```
<DeviceIdentity VendorID="0x010C" DeviceID="0x0010">
     <InfoText TextId="T_ID_DEV_DESCRIPTION"/>
     <VendorName Value="HMS Industrial Networks"/>
</DeviceIdentity>
```

The identity in the example represents HMS values.

- Replace VendorID value 0x010C with the value which corresponds to your vendor name.

  If you do not have a Vendor ID you can obtain this by contacting PI.

  The VendorID must match what is configured for the PROFINET IO Object (0xF6) attribute #2 (Vendor ID).

- Replace the DeviceID value 0x0010 with the value you have selected for this device.

  The DeviceID must match what is configured for the PROFINET IO Object (0xF6) attribute #1 (Device ID). Please note that if you change the Device ID you MUST also change the Vendor ID, as the Device ID is unique for the Vendor ID.

- Replace the VendorName value "HMS Industrial Networks" with name of your vendor.

  Please note that the keyword VendorName is found not only here, but also at other places in the GSD file. Use "Search"to find all instances and replace them with the name of your vendor.

Specify the function of the device.

```
<DeviceFunction>
      <Family MainFamily="General" ProductFamily="Anybus CompactCom 40 PIR"/>
</DeviceFunction>
```

The example GSD specifies a kind of "General"device as the usage of it is unclear.

- Replace the MainFamily with the class that best describe the device. The following are the allowed values:

  - General

  - Drives

  - Switching Devices

  - I/O

  - Valves

  - Controllers

  - HMI

  - Encoders

  - NC/RC

  - Gateway

  - PLCs

  - Ident Systems

  - PA Profiles

  - Network Components

  - Sensors

- Replace the ProductFamily value "Anybus CompactCom 40 PIR" with a string which describes your device.

In addition to the above, there are a few more places where identity related information is present in the GSD file.

```
DNS_CompatibleName="CompactCom40PIR"
```

For the Device access point (DeviceAccessPointItem) there is a keyword which is called DNS_CompatibleName that may be used by engineering tools to build the Station Name. Locate this by using the search function.

- Replace DNS_CompatibleName value with the default name of the device.

The order number of the device is set with the keyword "OrderNumber".

```
<OrderNumber Value="CompactCom 40 PIR"/>
```

In many cases the value of the OrderNumber equals the Station Type string, but it does not necessarily need to be that way.

- Replace the OrderNumber value "CompactCom 40 PIR"with the order number used for the device.

  The OrderNumber must match what is configured for the PROFINET IO Object (0xF6) attribute #8 (IM Order ID).

It is recommended to change the name of the DAP (keyword Name in the "ModuleInfo" section of the DAP) to something more descriptive. The DAP name is often visible in the PROFINET configuration tool.

- Replace the name "DAP" of the T_ID_DAP text ID with e.g. the name of the product type.

**How to Enable Initial Record Data**

During the establishment of an IO connection between the IO device and the IO controller it is possible for the IO controller to send initial record data. This initial record data is sent using the PROFINET IO service record write. This service can be used at any time and will write data to a defined ADI. The initial record data is defined in the GSD file, and is specified for a submodule of a module. By default, the Anybus CompactCom 40 module will not make use of any initial record data, but that can be enabled if needed.

To enable this functionality, the GSD file needs to be modified as specified below. In this example, 2 bytes are written to ADI 67 (ADI 67 corresponds to index 67) during startup of a PROFINET IO connection (the value can be configured by the end user):

```
<RecordDataList>
    <ParameterRecordDataItem Index="67" Length="2" TransferSequence="0">
       <Name TextId="T_ID_EXAMPLE2"/>
       <Ref DataType="Unsigned16" ByteOffset="0" DefaultValue="0"
          AllowedValues="0..65535" TextId="T_ID_EXAMPLE2_PRM_1"/>
    </ParameterRecordDataItem>
</RecordDataList>
```

It is recommended that the above GSD information is placed directly after the "</IOData>" keyword for the module for which the data is associated.

If more than one ADI needs to be set, the keyword "ParameterRecordDataItem"is duplicated.

Please note that TextId's ("T_ID_xxx" above) need to be added to the "<ExternalTextList>" section of the GSD file (once for each language defined).

```
<Text TextId="T_ID_EXAMPLE2" Value="Config parameter 1"/>
<Text TextId="T_ID_EXAMPLE2_PRM_1" Value="Parameter value description"/>
```

## 3.2       Network Identity

By default, the module identifies itself as a generic Anybus implementation as follows:

| | |
|---|---|
| **Device ID** | 0010h (Anybus CompactCom 40 PROFINET) |
| **Vendor ID** | 010Ch (HMS Industrial Networks) |
| **Station Type** | The Station Type will be assigned a value according to this priority: |
| | If the PROFINET IO Object (F6h) is implemented, the value equals that of attribute #3 in instance #1. |
| | If the PROFINET IO Object is not implemented, the value of attribute #9 (Product name) in the Application Object (FFh), instance #1, will be used (if implemented). |
| | If none of these attributes are implemented, the value equals Anybus CompactCom default value ("CompactCom 40 PIR"). |

It is recommended to customize the identity information so that the Anybus module appears as a vendor specific implementation rather than a generic Anybus product.

See also...

- *Identification & Maintenance (I&M), p. 25*

- *PROFINET IO Object (F6h), p. 118*

- Application Object (FFh) in Anybus CompactCom Software Design Guide

## 3.3       Communication Settings

Network related communication settings are grouped in the Network Configuration Object (04h), and includes:

| | |
|---|---|
| **Ethernet Interface Settings** | The module is locked to 100 Mbit full duplex operation as required by PROFINET. |
| **IP Settings** | These settings must be set in order for the module to be able to participate on the network. |
| | Normally set by the IO Controller. |
| | IP settings must be synchronized between the Anybus CompactCom 40 and the host application. |
| **PROFINET Station Name** | The module needs to be assigned a Station Name in order to participate on PROFINET. |
| | Normally set from the network. |

See also...

- *Identification & Maintenance (I&M), p. 25*

- *Network Configuration Object (04h), p. 55*

- *Secure HICP (Secure Host IP Configuration Protocol), p. 151*

## 3.4 Network Data Exchange

### 3.4.1 Application Data Instances (ADIs)

ADIs can be accessed acyclically from the network by means of Record Data read/write services. If addressed through a given API and Index range, the module translates the service into standard object requests towards the Application Data Object. If the host application responds with an error to such a request, that error will be translated to PROFINET standard.

The following parameters affect the addressing of ADIs onPROFINET:

| | |
|---|---|
| **Application Process Instance (API)** | API 0 (zero) provides access to data in the Application Data Object, i.e. the ADIs. Acyclic requests towards other APIs will be forwarded to the *PROFINET IO Object (F6h), p. 118* by means of the 'Get_Record' and 'Set_Record'-commands.<br><br>The remainder of this section assumes API 0 (zero). |
| **Slot & subslot** | The Slot and subslot values have no impact on the actual addressing of ADIs, except that the actual Slot and subslot needs to be populated with a module/submodule. This is always true for the DAP (Device Access Point), which occupies Slot #0/subslot #1. Other Slot/subslot values can also be used provided that the implementation populates it with a module/submodule. |
| **Index** | There is a 1:1 correlation between ADI and index as long as the index number is less than - or equal to - 7FFFh. Index 0 (zero) is not associated with an ADI and cannot be used. |

| API | Slot | Subslot | Index | ADI | Comments |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0000h | - | (not associated with ADIs) |
| | | | 0001h | 1 | Device Access Point (DAP) |
| | | | 0002h | 2 | |
| | | | ... | ... | |
| | | | 7FFFh | 32767 | |
| | | | 8000h...FFFFh | - | (not associated with ADIs) |
| | X (>0) | Y | 0000h | | |
| | | | 0001h | 1 | Conditional; X and Y must be populated |
| | | | 0002h | 2 | |
| | | | ... | ... | |
| | | | 7FFFh | 32767 | |
| | | | 8000h...FFFFh | - | (not associated with ADIs) |
| >0 | - | - | - | - | See *Application Process Instances (API), p. 17* |

See also...

- *Caveats, p. 16*
- *Application Process Instances (API), p. 17*
- *PROFINET IO Object (F6h), p. 118*

### 3.4.2 Process Data

Mapping an ADI to Write Process Data results in PROFINET input data, and mapping an ADI to Read Process Data results in PROFINET output data. Consistency over all I/O data mapped on PROFINET is guaranteed.

See also...

- *Real Identification (RI), p. 20*

> ! *The order in which ADIs are mapped to Process Data is significant and must be replicated in the IO Controller when setting up the network communication.*

### 3.4.3 Caveats

The length parameter in the Record Data request specifies the number of bytes to read/write.

- When reading more data than the actual size of the ADI, the response will only contain the actual ADI data, i.e. no padding on the data is performed by the module.

- When writing to an ADI, the length parameter is not checked by the module, i.e. the host application must respond with an error if the length differs from the actual size of the requested ADI.

See also..

- *Application Process Instances (API), p. 17*

# 4      PROFINET Implementation Details

## 4.1      General Information

This chapter covers PROFINET specific details in the Anybus implementation. Note that the use of such functionality may require in-depth knowledge in PROFINET networking internals and/or information from the official PROFINET specification. In such cases, the people responsible for the implementation of this product are expected either to obtain these specifications to gain sufficient knowledge or limit their implementation in such a way that this is not necessary. The GSD file must be changed to reflect all changes.

Implementation overview:

| | |
|---|---|
| **Conformance Class** | The Anybus module complies to conformance class C. |
| **Performance Characteristics** | • 100 Mbps, full duplex with autonegotiation enabled as default |
| | • Real Time (RT) communication, 250 µs cycle time |
| | • Isochronous Real Time (IRT) communication, 250 µs cycle time |
| **Device Model** | • One IO Device instance |
| | • The IO Device instance includes an Application Process referenced by its identifier (API). API 0 (zero) is implemented by default. |
| | • The API implements one or more slots |
| | • Each Slot implements one or more subslots |
| | • Each subslot may implement one or more Channels |
| **Slots & Subslots** | Up to 128 subslots in total. |
| **IO Data** | 1440 bytes of IO data in each direction, including status bytes (4 bytes for DAP submodules + 1 byte per additional submodule) |

See also...

- *Electronic Data Sheet (GSD), p. 11*

## 4.2      Application Process Instances (API)

As mentioned previously, acyclic requests towards API 0 are forwarded to the Application Data Object.

Cyclic data exchange is by default carried out through API 0 (i.e. the Anybus associates modules and submodules with API 0).

See also...

- *Application Data Instances (ADIs), p. 15*

## 4.3       Application Relationships (AR)

On PROFINET, a connection between an IO Controller/Supervisor and an I/O device (in this case the Anybus) is called Application Relationship (AR). The Anybus module supports multiple simultaneous Application Relationships, allowing multiple IO Supervisors to access its data and functions.

The host implementation can either ignore this functionality altogether, in which case the Anybus module will handle it automatically, or integrate the establishment and handling of Application Relationships into the host firmware.

Application Relationships are managed through the following functions:

- AR_Check_Ind (for command details see *PROFINET IO Object (F6h), p. 118*)

- Expected_Ident_Ind (for command details see *PROFINET IO Object (F6h), p. 118*)

- AR_Abort_Ind (for command details see *PROFINET IO Object (F6h), p. 118*)

- AR_Abort (for command details see *Network PROFINET IO Object (0Eh), p. 66*)

## 4.4        PROFINET System Redundancy S2

System redundancy relies on the concept of multiple connections to a device to maintain the system in the event of a failure. System redundancy for PROFINET is available in different setups. The S2 functionality, implemented in the Anybus CompactCom 40 PROFINET IRT, makes it possible for two PROFINET controllers, one primary and one backup, to connect to the device. If the primary controller fails, the backup controller instantly takes over the communication without any disturbance on the network. Running S2 requires specific controllers supporting this feature.

S2 redundancy is enabled in by setting instance #1, attribute #28, in the PROFINET IO Object (F6h), for details see *PROFINET IO Object (F6h), p. 118*.

### 4.4.1        Activating S2 System Redundancy in the GSD File

For each DAP, the following additions must be made to an S2 compliant configuration in the GSD file:

- SystemRedundancy element attributes according to:

```
<SystemRedundancy DeviceType="S2" MaxSwitchOverTime="45"
S2MaxInputOnBackupDelay="0" RT_InputOnBackupAR_Supported="true"
NumberOfAR_Sets="1"/>
```

- The attribute PrmBeginPrmEndSequenceSupported according to:

```
<DeviceAccessPointItem ... PrmBeginPrmEndSequenceSupported="true"/>
```

- The attribute PDEV_CombinedObjectSupported according to:

```
<InterfaceSubmoduleItem ... PDEV_CombinedObjectSupported="true"/>
```

- The attribute NumberOfAR (equal two times NumberOfAR_Sets) according to:

```
<ApplicationRelations ... NumberOfAR="2"/>
```

For S2 System Redundancy, MinDeviceInterval has been set to "32" in the GSD file, equivalent to a minimum cycle time of 1ms. This has been done to account for the potentially big load of operating two I/O AR:s.

```
<DeviceAccessPointItem ... MinDeviceInterval="32" .../>
```

Depending on the process data size and layout exchanged on the network in a specific application, it may be possible to set MinDeviceInterval to a lower value (but not lower than 8).

## 4.5        Real Identification (RI)

### 4.5.1      General Information

During the establishment of an IO Connection towards the Anybus CompactCom 40 PROFINET IRT, the configuration derived from the IO Controller (i.e. the Expected Identification) and the actual configuration in the Anybus CompactCom 40 PROFINET IRT (i.e. the Real Identification or RI) are compared.

The RI configuration is either handled by the module (default), or by the host application. In either case the GSD file has to be customized to correspond to the configuration.

**ADI Based Configuration (Default)**

By default (i.e. if the application does not issue API_Add, Plug_Module, Plug_Submodule), the Anybus CompactCom 40 PROFINET IRT handles the plugging of modules and submodules automatically in accordance with the mapped Process Data as follows:

• 	 	A DAP is plugged into Slot 0 (zero)

• 	 	Modules are added in consecutive order (based on the order of the mapping commands)

• 	 	All modules belong to API 0 (zero)

The Anybus CompactCom 40 PROFINET IRT internally creates module/submodule identifiers as described in the picture and the example below. The GSD file has to be customized to define the same modules/submodules with the same identifiers as the Anybus CompactCom 40 PROFINET IRT has created internally.
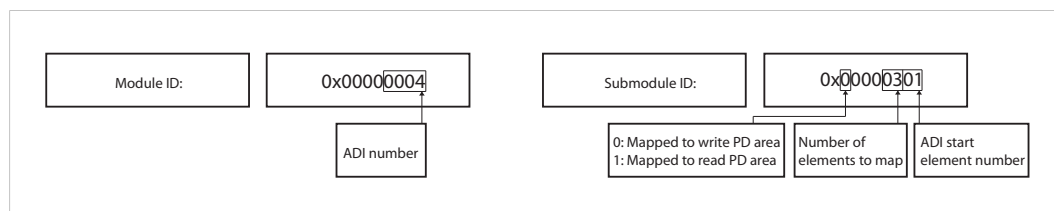


**Fig. 1**

Example (100BASE-TX DAP):

| ADI # | Type | Resulting Real Identification | | |
|---|---|---|---|---|
| | | Module/Submodule ID | Slot/Subslot | IO Data Size (bytes) |
| - | - | 0x80010000/ 0x00000001 0x80010000/ 0x00000002 0x80010000/ 0x00000003 0x80010000/ 0x00000003 | 0/1 (Device Access Point) 0/0x8000 (Interface) 0/0x8001 (Port1) 0/0x8002 (Port2) | - |
| 6 | 1 SINT16 Wr | 0x00000006/ 0x00000100 | 1/1 | 2 |
| 3 | 2 UINT8 Rd | 0x00000003/ 0x10000200 | 2/1 | 2 |
| 3 | 1 UINT8 Rd | 0x00000003/ 0x10000104 | 2/2 | 1 |
| 9 | 1 SINT32 Rd | 0x00000009/ 0x10000100 | 3/1 | 4 |
| 10 | 1 SINT16 Wr | 0x0000000A/ 0x00000100 | 4/1 | 2 |

See also...

- *Application Process Instances (API), p. 17*

- *Flowchart —Establishment of Real Identification (RI), p. 148*

**Modular Device Configuration**

The RI, when implementing a modular device, is generated from the Module IDs and process data mappings as defined in the Modular Device Object (ECh). For more information see *Modular Device, p. 33*.

**Custom Configuration (Advanced Users)**

Optionally it is possible to override the default configuration during the SETUP state by means of the PROFINET specific commands API_add, Plug_Module, Plug_Submodule, and Plug_ Submodule_Ext. This way, the host application can define exactly how ADIs are represented on PROFINET by defining custom modules and submodules. These commands need to be sent after the process data is mapped, and before sending setup complete to the Anybus CompactCom 40 PROFINET IRT.

See also...

- *Network PROFINET IO Object (0Eh), p. 66*

- *Flowchart —Establishment of Real Identification (RI), p. 148*

## 4.5.2        Configuration Mismatch

**General**

A configuration mismatch occurs when the Real Identification (RI) does not match the Expected Identification. Depending on how the RI configuration is established, the Anybus CompactCom 40 PROFINET IRT will first try to resolve the mismatch as described in the applicable section below (Resolving Mismatch for default configuration or custom configuration). If this attempt to resolve the mismatch fails, the Anybus CompactCom 40 PROFINET IRT will provide indications as described in *Further Actions to Resolve Mismatch, p. 23*.

**Resolving Mismatch (Default Configuration)**

If the Real Identification has been established according to the default mode, the Anybus CompactCom 40 PROFINET IRT will try to remap the Real Identification to match the Expected Identification.

> **!** The application must have implemented support for the remap commands in the Application Data object (FEh), for remap to be possible.

As the Module ID contains the ADI number and the Submodule ID describes I/O direction and element section, all information required to perform a remap is available.

> **ⓘ** *The application must be able to respond with the data type for every ADI, or a remap is not possible.*

The remap is performed as follows:

- A request is sent to remap the read area. If this request is rejected no change is made to the process data map and the Real Identification.

- If the read area remap succeeds, a write remap request is sent. If this is rejected, the process data map is in an inconsistent state and new remap commands are sent that will remove all mappings.

- If both the read and the write remap requests succeed a new Real Identification will be built to match the Expected Identification.

See also ...

- *Default Configuration Mismatch, p. 149* (flowchart)

- Application Data Object (FEh) (see Anybus CompactCom 40 Software Design Guide)

**Resolving Mismatch (Custom Configuration)**

If a configuration mismatch occurs for a custom configuration, the Anybus CompactCom 40 PROFINET IRT will issue the command Expected_Ident_Ind to the host application. If the host application intends to change the Real Identification based on the Expected Identification, it responds with "Block" and performs the required Pull/Plug operations, before sending an Ident_ Change_Done command to the Network PROFINET IO Object (0Eh).

See also ....

- *Custom Configuration mismatch, p. 150* (flowchart)

- Expected_Ident_Ind, see command details in *PROFINET IO Object (F6h), p. 118*

- Ident_Change_Done, see command details in *Network PROFINET IO Object (0Eh), p. 66*

**Further Actions to Resolve Mismatch**

If the mismatch remains unsolved, either for default or custom configuration, the following will be performed by the Anybus CompactCom 40 PROFINET IRT to find a solution that will make exchange of valid data possible:

| | |
|---|---|
| **Incomplete output mapping:** | If the controller tries to connect to fewer output submodules than are plugged by the application, the controller will not be granted ownership of any output submodule. The Anybus state is set to ERROR and the LEDs will indicate configuration mismatch. Matching input submodules will be owned by controller and work normally. |
| **Incomplete input mapping:** | The controller may choose to connect to a subset of the available input submodules without any restrictions. Anybus state is set to PROCESS_ACTIVE and no error is indicated on any LED. |
| **Mismatch of submodule(s):** | As long as all of the output submodules of the Real Identification are present and matching in the Expected Identification, the Anybus state is set to PROCESS_ACTIVE. However, if there is any mismatch among the other submodules the LEDs will indicate configuration mismatch. |

See also...

• The Remap_ADI_Write_Area and Remap_ADI_Read_Area commands in the Application Data Object (FEh), found in Anybus CompactCom 40 Software Design Guide.

## 4.6 Diagnostics

### 4.6.1 Standard Diagnostics

PROFINET IO uses alarms when informing the IO Controller of diagnostic entries. In the Anybus implementation, it is possible for the application to create alarms via diagnostic entries by means of the Diagnostic Object (02h).

Up to 5 diagnostic instances can be created by the host application. An additional 6th instance can always be created in event of a major unrecoverable fault.

Creating a diagnostic instance is done by issuing the command Create. If the module is in state IDLE or PROCESS_ACTIVE, the created instance will be communicated on the network as an "appear"-alarm. If the module is in another state, the PLC will be notified in the connect response by a module diff block.

Deleting a diagnostic instance is done by issuing the command Delete. This will trigger a "disappear"-alarm on the network. Supply the instance ID that was returned by the create-command.

Every diagnostic instance has a severity level and an event code associated to it. Major unrecoverable events will cause the module to disconnect itself from the network, thus preventing network participation. Other severity levels either produce a Channel Diagnostic alarm or a Generic Diagnostic alarm, depending on the Event Code, according to the table below.

| Severity | Event code != network specific | Event code = network specific |
|---|---|---|
| Minor, recoverable | Channel Diagnostic Alarm | Generic Diagnostic Alarm (See *Extended Diagnostics, p. 24*) |
| Minor, unrecoverable | | |
| Major, recoverable | | |
| Major, unrecoverable | Anybus enters Exception state | |

ℹ️  *Process alarms can not be created.*

See also..

• *Diagnostic Object (02h), p. 47*

### 4.6.2 Extended Diagnostics

Using the network specific event code (FFh) creates a Generic Diagnostic Alarm on the network. This type of alarm can carry extended diagnostic information and more details about the source of the problem.

Generic Diagnostic Alarm instances can be tagged with a source API and slot- and subslot number. It can also contain additional network specific diagnostic data like:

• Standard channel diagnosis with manufacturer extension

• Manufacturer specific channel diagnosis

• Manufacturer specific channel diagnosis with manufacturer specific extension that provide also network specific diagnostic data

For more information, see *Details: Network Specific Data, p. 48*.

# 4.7     Identification & Maintenance (I&M)

## 4.7.1     General Information

Identification & Maintenance (I&M) provides a standard way of gathering information about an I/O device. The I&M information can be accessed by the IO Controller by means of acyclic Record Data Read/Write services.

The application should provide application specific I&M0 information during start-up. See *PROFINET IO Object (F6h), p. 118* for more information.

It is possible for the application to handle I&M records. Activate this using the IM_Options command. See *Network PROFINET IO Object (0Eh), p. 66* for more information.

While I&M0 information describes the application, the I&M5 information describes the communication unit i.e. the Anybus CompactCom 40 PROFINET IRT, to the PROFINET network. Most of the information is provided by the module, but the application can change the order ID and the IM annotation (attributes #25 and #26, see *PROFINET IO Object (F6h), p. 118*). I&M5 is enabled by default, but can be disabled using attribute #27 (IM5 enabled), see *PROFINET IO Object (F6h), p. 118*. If the example GSD file is used, I&M5 has to be disabled there aswell, se below.

> **ⓘ**   *I&M5 is not available for Anybus IP. See below how to disable I&M5 in the GSD file for Anybus IP.*

Default I&M0 information:

| | |
|---|---|
| **IM Manufacturer ID** | 010Ch (HMS Industrial Networks) |
| **IM Order ID** | "CompactCom 40 PIR" |
| **IM Serial Number** | (unique serial number, set during manufacturing) |
| **IM Hardware Revision** | (Anybus hardware revision ID, set during manufacturing) |
| **IM Software Revision** | (Anybus software revision, set during manufacturing) |
| **IM Revision Counter** | (Revision counter) |
| **IM Profile ID** | 0000h (Generic Device) |
| **IM Profile Specific Type** | 0004h (No profile) |
| **IM Version** | 0101h |
| **IM Supported** | For submodules belonging to a "non–zero API",the returned value is zero. For submodules belonging to API 0, the returned value is 000Eh (IM0-3 supported) |

**Disabling I&M5 in the GSD File**

> **ⓘ**   *I&M5 is always disabled and can not be enabled on Anybus IP. It has to be disabled in the example GSD file, if this is to be used*

I&M5 is enabled by default in the GSD file.

The settings for I&M are located at the Device Access Point, within `<VirtualSubmoduleItem...>`. To disable I&M5 , change GSDML entry `<IM5_ Supported="true">` to `<IM5_Supported="false">`.

### 4.7.2    I&M Data Structures

The I&M records uses the following data structures.

| Record | Content | Size | Description |
|--------|---------|------|-------------|
| I&M0 | Manufacturer Id | 2 bytes | PROFINET IO Object (F6h), attribute #2 (Vendor ID/I&M Vendor ID) |
| | Order Id | 20 bytes | PROFINET IO Object (F6h), attribute #8 (I&M Order ID) |
| | Serial number | 16 bytes | The content will be assigned in the following priority order: 1. PROFINET IO Object (F6h), attribute #9 (I&M Serial number)  2. Application Object (FFh), attribute #3 (Serial number)  3. Default value, assigned during manufacturing |
| | Hardware revision | 2 bytes | The content will be assigned in the following priority order: 1. PROFINET IO Object (F6h), attribute #10 (I&M Hardware revision)  2. Application Object (FFh), attribute #11 (Hardware version)  3. Default value, assigned during manufacturing |
| | Software revision | 4 bytes | The content will be assigned in the following priority order: 1. PROFINET IO Object (F6h), attribute #11 (I&M Software revision)  2. Application Object (FFh), attribute #10 (Firmware version)  3. Default value, assigned during manufacturing |
| | Revision counter | 2 bytes | PROFINET IO Object (F6h), attribute #12 (I&M Revision counter) |
| | Profile Id | 2 bytes | PROFINET IO Object (F6h), attribute #13 (I&M Profile ID) |
| | Profile specific type | 2 bytes | PROFINET IO Object (F6h), attribute #14 (I&M Profile specific type) |
| | IM version | 2 bytes | 0101h (Internal, constant value) |
| | IM supported | 2 bytes | For submodules belonging to a "non–zero API",the returned value is zero. For submodules belonging to API 0, the returned value is 000Eh (IM0-3 supported) |
| I&M1 | Tag Function | 32 bytes | Default: All bytes set to blanks |
| | Tag Location | 22 bytes | Default: All bytes set to blanks |
| I&M2 | Installation date | 16 bytes | Default: All bytes set to blanks |
| I&M3 | Descriptor | 54 bytes | Default: All bytes set to blanks |
| I&M5 | IM_Annotation | 64 bytes | PROFINET IO object; attribute #26 (IM Annotation). If the application doesn't support the attribute: Anybus CompactCom 40 PROFINET IRT |
| | IM_OrderID | 64 bytes | PROFINET IO object; attribute #25 (IM Module Order ID). If the application doesn't support the attribute: Anybus CompactCom 40 PROFINET IRT |
| | VendorIDHigh | 1 byte | 01h |
| | VendorIDLow | 1 byte | 0Ch |
| | Serial number | 16 bytes | Serial number of the Anybus CompactCom |
| | Hardware revision | 2 bytes | Hardware version of the Anybus CompactCom |
| | Software revision | 4 bytes | Product version of the Anybus CompactCom |

See also..

- *PROFINET IO Object (F6h), p. 118*

- Anybus CompactCom 40 Software Design Guide, Application Object (FFh)

## 4.8    Asset Management

Asset management provides means to collect information on non PROFINET automation components, that are connected to PROFINET networked devices. These components are not part of the PROFINET system, but the collected information will facilitate troubleshooting and

exchange of faulty components. The host application can describe up to 32 different non PROFINET components in the instances of the Asset Management Object.

See also...

- *Asset Management Object (E5h), p. 102*

### 4.8.1      Activating Asset Management in the GSD File

Asset management is not a default feature, and the example GSD files distributed do not contain the keyword "AssetManagement". To activitate it, the user must implement the Asset Management object for the device to accept the Asset Management Read Record from the network. In addition the keyword "AssetManagement" must be added to the GSD.

The settings for Asset Management have to be added at the end of the Device Access Point, directly after `</Graphics>`, or the GSD checker tool will invalidate the file:

```
...
<DeviceAccessPointItem...>
  ....
  <Graphics>
    <GraphicItemRef Type="DeviceSymbol" GraphicItemTarget"1"/>
  </Graphics>
  <AssetManagement/>
</DeviceAccessPointItem>
...
```

## 4.9        Fast Start Up

### 4.9.1        General Information

The Fast Start Up (FSU) function enables PROFINET IO devices, connected to the network, to power up quickly. This is useful in, for example, robot applications, where rapid retooling is necessary. With FSU activated, the module will send a DCP Hello message as soon as possible after power-on.

This function is enabled by two GSD keywords: PowerOnToCommReady and DCP_HelloSupported. The activation is made from the PLC configuration tool.

The FSU time is defined as the number of milliseconds (ms) from hardware reset (or power-on) until the module enters the PROCESS_ACTIVE state. On PROFINET, it is recommended to try to reach a FSU time <= 500 ms.

To enable FSU, set values according to the following (listed for the Device Access Point(s)):

| | |
|---|---|
| **PowerOnToCommReady** | FSU time, in milliseconds (ms). This value must be measured by the customer. |
| **DCP_HelloSupported** | Value: true. |
| **ParameterizationSpeedupSupported** | Value: true |

To disable FSU, set the keywords to the following values:

| | |
|---|---|
| **PowerOnToCommReady** | Remove this keyword from the GSD file. |
| **DCP_HelloSupported** | Value: false. |
| **ParameterizationSpeedupSupported** | Remove this keyword from the GSD file. |

### 4.9.2        How to Improve the FSU Time

- Issue as few commands as possible to the module during the SETUP state.

- Respond as fast as possible to all commands issued by the module after setup is complete.

- If there is record data in the GSD file, use attribute #4 in the Application Object (FFh), instance #1 (Parameter Control Sum). During the next start-up, this parameter data is already saved in the module's nonvolatile storage and will not be sent to the application.

### 4.9.3        Fast Start Up Configuration with STEP7

The example below shows the procedure when the Siemens tool STEP7 is used for configuration.

**Activation of Fast Start Up**

1. Start the configuration tool. The figure below shows the HW Config window of the STEP7 tool.
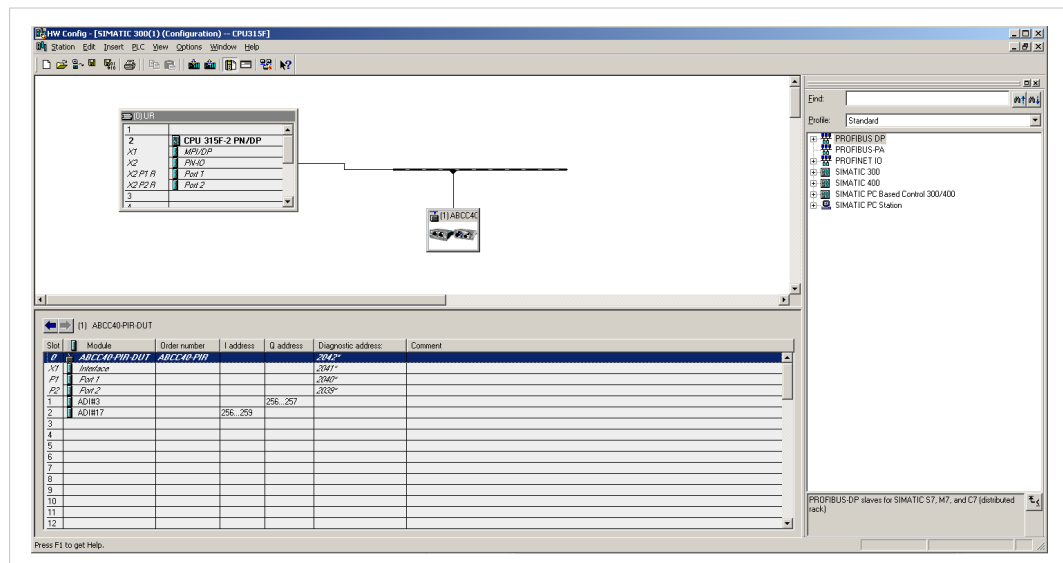


**Fig. 2**

2. Double click on "Interface" in the Module column. The window shown to the right will appear. Choose the **General**tab and check the box **Prioritized startup**.
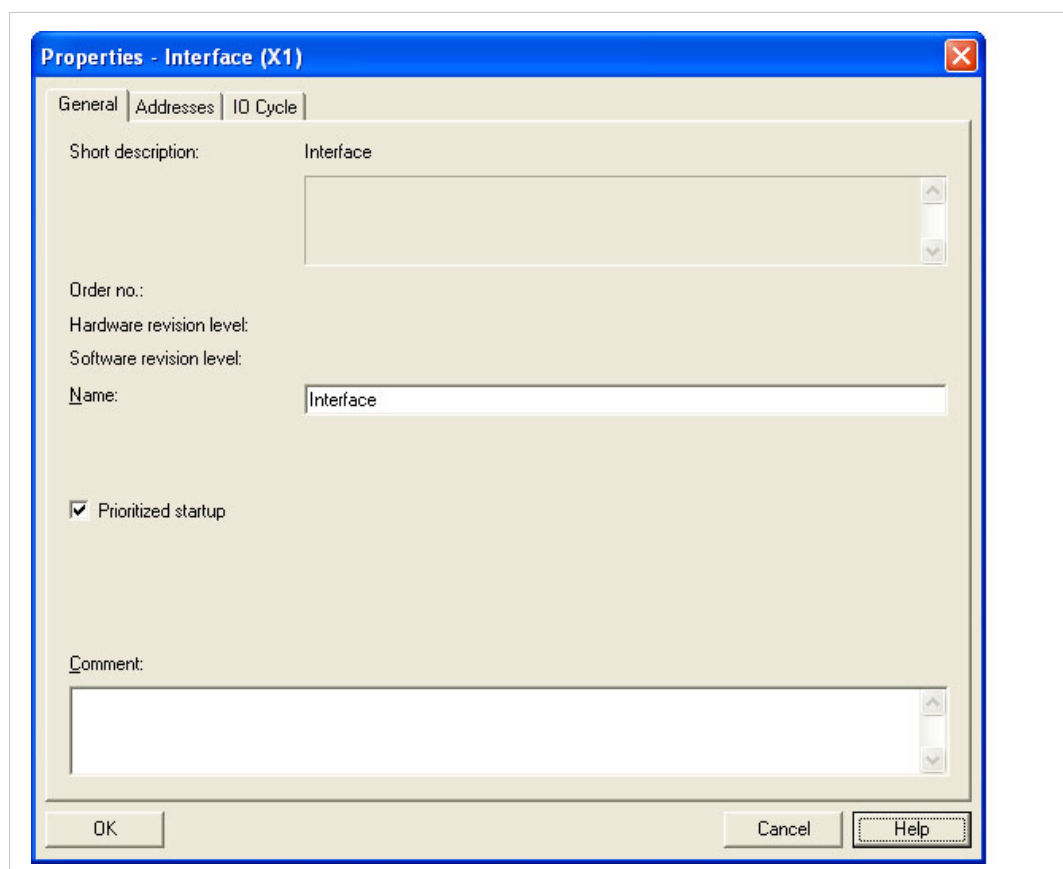


**Fig. 3**

3.   Return to the HW Config window. Double click on **Port 1** in the Module column. The window shown to the right will appear. Choose the **Options** tag. To configure fastest possible startup, choose transmission medium/duplex **TP/ITP 100 Mbps, full duplex**and check the **Disable autonegotiation**box.
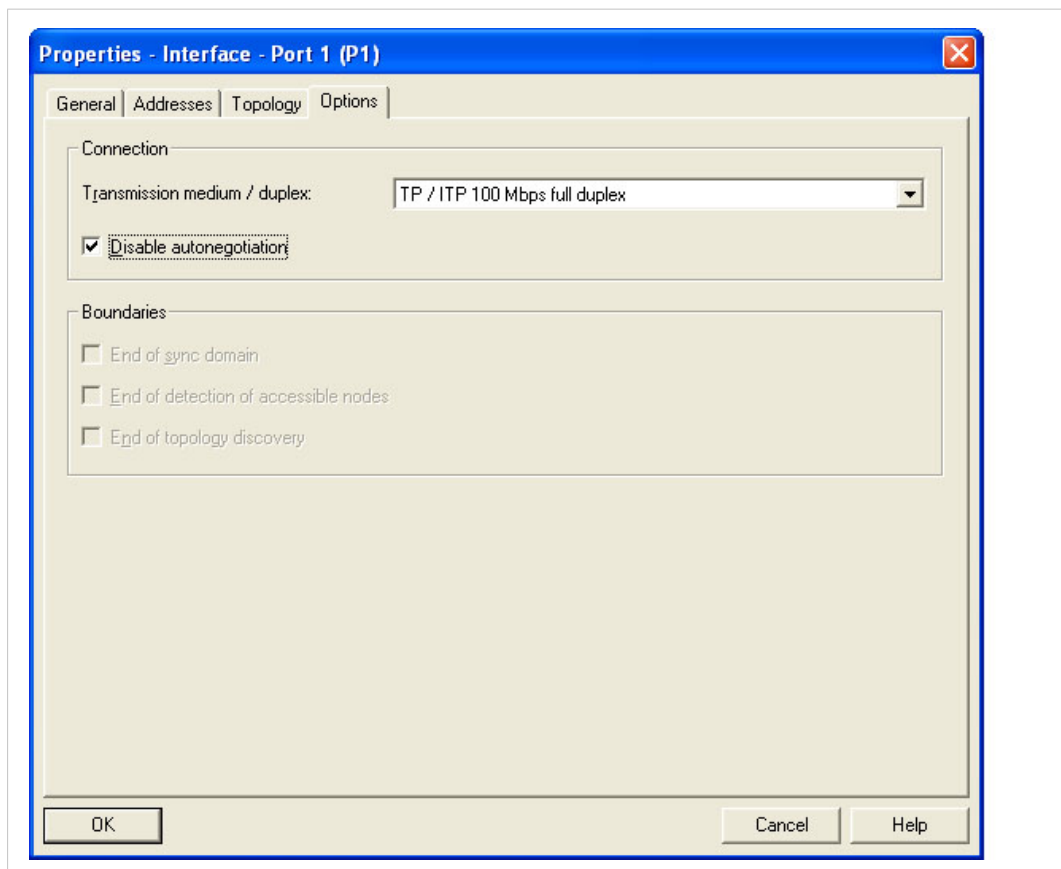


**Fig. 4**

4.   Repeat for Port 2.

## 4.10      Address Conflict Detection (ACD)

The Anybus CompactCom 40 PROFINET IRT supports Address Conflict Detection (ACD). This mechanism involves the following two aspects:

• Initial probing: before using an IP address, the module issues ARP probes to see if the address is already in use (three probes with a 100 ms delay).

• Address announcement: after the initial probing, the module issues ARP announcements.

If an IP address conflict is detected, IP address error will be indicated on the Network Status LED. The module will use address 0.0.0.0. A new address can be configured via the Anybus IPconfig tool.

> **!** If Fast Start Up is used, ACD initial probing is automatically disabled to ensure a fast startup. Address announcement is still used, as it will not affect the actual startup time.

To enable/disable ACD, see*Ethernet Host Object (F9h), p. 138*.

## 4.11    PROFIenergy Profile

The Anybus CompactCom 40 PROFINET IRT supports the PROFIenergy profile, according to the PROFIenergy Technical Specification, rev. 1.2. This profile makes it possible for a user to temporarily put a device in energy saving mode, e.g. during a lunch break or during weekends. The amount of power used by machines, when they are not in active use is thus reduced. Each device can be set individually to the energy saving mode that is the most optimal depending on the length of the production stop. Operators in factories, for example, can easily set all devices at the same time, in the, for each device, optimal energy saving mode.

The application defines the time for how long the device will stay in energy saving mode, and the device decides which mode will be the most appropriate. Transitions between the "Ready to operate" mode and all saving modes are mandatory, while transitions between different energy saving modes are optional. The transition from "Ready to operate mode" to "Power off mode" is not mandatory, as repowering the device may mean hands on restart.

### 4.11.1    Implementation

The PROFIenergy profile is implemented in the Anybus CompactCom 40 PROFINET IRT according to the state machine described in the PROFIenergy Technical Specification (available from PROFIBUS International). PROFIenergy commands arriving from the network will be translated into the Anybus CompactCom implementation as follows:

| PROFIenergy command | Anybus CompactCom 40 implementation |
|---|---|
| Start_Pause | Translated into a StartPause command to instance #0 in the Energy Control Object. |
| Start_Pause_with_time_response | Same as Start_Pause but will do several reads of the Energy Control object, and the currently used energy saving mode to complement the response. |
| End_Pause | Translated into a EndPause command to instance #0 in the Energy Control Object. |
| List_Energy_Saving_Modes | Instances present in the Energy object will be listed. |
| Get_Mode | Supported attributes for the requested mode/instance will be read. |
| PEM_Status | Results in several reads of Energy Control object and the currently used energy saving mode. |
| PEM_Status_with_Ext1 | |
| PE_Identify | Returns a static list of the supported, and also mandatory, PROFIenergy commands. |
| Get_Measurement_List | Returns the list of supported measurement values from the Energy Measurement object. |
| Get_Measurement_Values | Returns the actual measurement values from the Energy Measurement object. |
| Get_measurement_List_with_object_number | Returns the list of supported measurement values from the Energy Measurement object. |
| Get_Measurement_Values_with_object_number | Returns the actual measurement values from the Energy Measurement object. |
| Query_Version | Return the supported PROFIenergy version. |
| Info_Sleep_Mode_WOL | Not supported. |
| Go_Sleep_Mode_WOL | |

The PROFIenergy profile is valid for all sub-slots and is accessed through index 80A0h. Maximum number of instances/modes in the Energy Control Object is 8.

See also...

- *Energy Measurement Object (E4h), p. 91*

- *Energy Control Object (F0h), p. 112*

## 4.12        PROFIsafe

The Anybus CompactCom 40 PROFINET IRT supports the PROFIsafe profile. This profile makes it possible for a user to send data on a black channel interface, i.e. a safe channel over PROFINET using an add on Safety Module, e.g. the IXXAT Safe T100/PS. For more information about this module, see IXXAT Safe T100 Manual, available at www.ixxat.com.

The Safety Module can be located in any valid slot. This is true for when the configuration is handled by the module, as well as when the configuration is handled by the application.

For an application to support PROFIsafe, the Functional Safety Object in the application have to be implemented. Slots are assigned using the Command Add_Safety_Module to the Network PROFINET IO object (0Eh). The safe communication is enabled in the host application Functional Safety Object (E8h).

The Anybus CompactCom serial channel is used for functional safety communication. When this channel is used for the host application, a second separate serial channel, is implemented for the functional safety communication, see Anybus CompactCom Hardware Design Guide.

See also...

*Functional Safety Module Object (11h), p. 83*

*Functional Safety Object (E8h), p. 106*

*PROFINET IO Object (F6h), p. 118*

Command details: Add_Safety_Module in *Network PROFINET IO Object (0Eh), p. 66*

*Anybus CompactCom Hardware Design Guide*

# 5        Modular Device

## 5.1       General

The modular device concept is useful when creating a configuration based on reusable modules, for example when modelling a backplane device with pluggable modules.

The first slot (slot 0) is always occupied by the Device Access Point (DAP) module, which represents the head node where the Anybus CompactCom is located. It is not allowed to add a process data mapping for any ADI within the range of slot 0. The remaining slots are assigned fixed ranges of ADIs, set up using the Modular Device Object. A GSD module can represent different ADIs, depending on which slot it occupies, making it possible to create a configuration based on reusable modules. This is not possible in ADI based RI mode.

See also ...

- " Modular Device Object (ECh)" (see Anybus CompactCom 40 Software Design Guide)

## 5.2       Modular Device RI

The RI, when implementing a modular device, is generated from the Module IDs as defined in the Modular Device Object (ECh) (see the Anybus CompactCom 40 Software Design Guide) and from the process data mappings as shown in the example below.

| Module ID: | 0x0000004 | | | | |
|---|---|---|---|---|---|
| Submodule ID: | 0x00010203 | | | | |
| | 0x0 | | | | 0: Mapped to Write PD area<br>1: Mapped to Read PD area |
| | | 001 | | | ADI index |
| | | | 02 | | Number of elements to map |
| | | | | 03 | ADI start element # |

A set of conditions must be fulfilled for the Anybus CompactCom to use the Modular Device RI:

- The Modular Device object implemented must be of revision 1.

- Modular device object attributes #11 and #12 must be readable and valid

    - Number of slots must be: 0 < NumSlots <= 256

    - ADIs per slot must be: 0 < AdisPerSlot <= 4096

- The list with Module Type IDs must contain "NumSlots" items.

## 5.3       Remap

For a host application not supporting remap, the modules must contain a fixed set of submodules (GSD attribute FixedInSubslots), where the process data mapping of each slot is performed in accordance with the GSD definition of each particular module type. See *Modular Device Example, p. 34*, where slot 1 and slot 3 have the same module ID and also matching process data mapping commands.

If remap is supported by the application the GSD modules may be designed in a flexible manner with pluggable submodules (GSD attributes AllowedInSubslots / UsedInSubslots). If the configuration results in an expected identification that differs from the real identification the Anybus CompactCom will attempt to resolve the mismatch according to *Configuration Mismatch, p. 22*.

## 5.4        Safety Module

A safety module, such as the IXXAT Safe T100/PS may be used within the modular device concept. If enabled, the safety module can be plugged into any valid slot and the host application will not be allowed to add a process data mapping for any ADI within the range this slot. Please note that this module can only be used once and that should be reflected in the GSD file e.g. by setting AllowedInSlots/UsedInSlots/FixedInSlots to "[safety module slot no]".

## 5.5        Record Data

In non-transparent record data mode, ADIs are addressed acyclically, based on slot and index in a one-to-one relationship, according to ADI number = Slot * ADIsPerSlot + Index. Since the address space is not divided per slot/module, it is recommended to use transparent record data mode when implementing the modular device concept. This way the host application will be informed which slot, subslot and index is addressed by the controller.

## 5.6        Modular Device Example

The following example shows a device with two defined types of modules (in addition to the DAP). In its current configuration the device has Module Type B plugged in slots 1 and 3, while Module Type C is plugged in slot 2. No safety module is used so process data can be added to all slots except slot 0.

### 5.6.1      Mapping Commands

Please note that the mapping commands below omit parts not relevant for the example.

Slot 1:

Map_ADI_Write_Ext_Area( ADI=257, FirstIndex=0, NumElems=8, DataTypes=(UINT8,) )

Map_ADI_Read_Ext_Area( ADI=300, FirstIndex=3, NumElems=2, DataTypes=(UINT16,) )

Map_ADI_Read_Ext_Area( ADI=304, FirstIndex=0, NumElems=1, DataTypes=(INT32,) )

Slot 2:

Map_ADI_Read_Ext_Area( ADI=513, FirstIndex=0, NumElems=4, DataTypes=(UINT8,) )

Map_ADI_Write_Ext_Area( ADI=700, FirstIndex=0, NumElems=2, DataTypes=(UINT16,) )

Map_ADI_Write_Ext_Area( ADI=708, FirstIndex=0, NumElems=16, DataTypes=(UINT32,) )

Map_ADI_Read_Ext_Area( ADI=760, FirstIndex=0, NumElems=3, DataTypes=(BIT4, PAD2, BIT2) )

Slot 3:

Map_ADI_Write_Ext_Area( ADI=769, FirstIndex=0, NumElems=8, DataTypes=(UINT8,) )

Map_ADI_Read_Ext_Area( ADI=812, FirstIndex=3, NumElems=2, DataTypes=(UINT16,) )

Map_ADI_Read_Ext_Area( ADI=816, FirstIndex=0, NumElems=1, DataTypes=(INT32,) )

### 5.6.2 List of All Module IDs:

The response to the Get_List command (list type 01h) contains the following modules:

1. 0xA0000001 (DAP)

2. 0xB0000002 (Module type B)

3. 0xC0000003 (Module type C)

4. 0xB0000002 (Module type B)

### 5.6.3 Device Layout
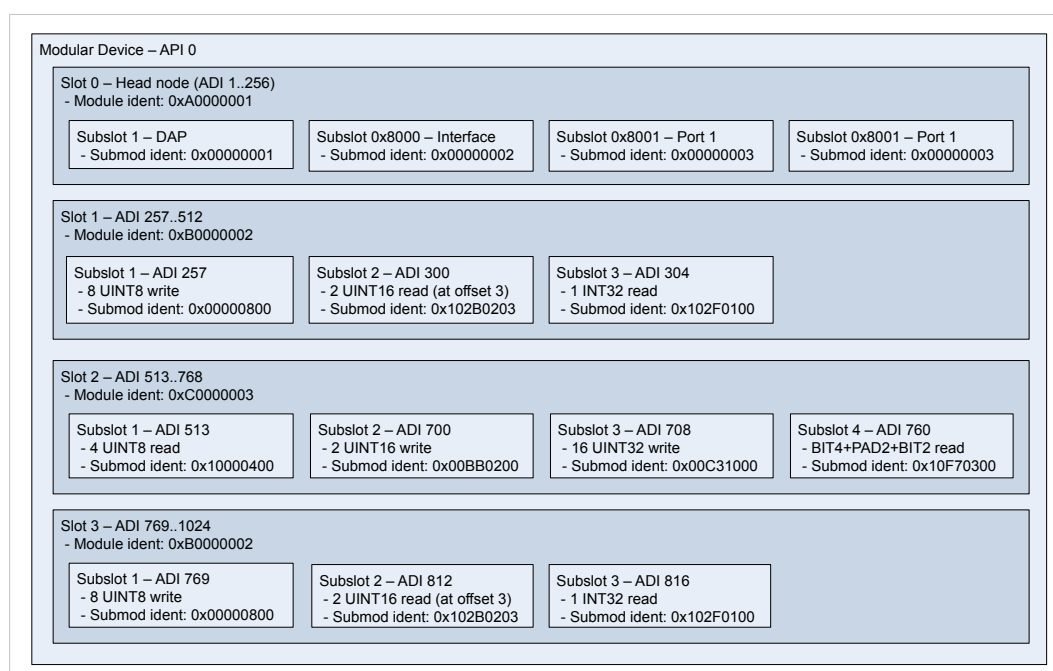


**Fig. 5**

### 5.6.4 Example View in TIA Portal

Note that the GSD instantiates the same module, Module Type B in both slot 1 and slot 3.
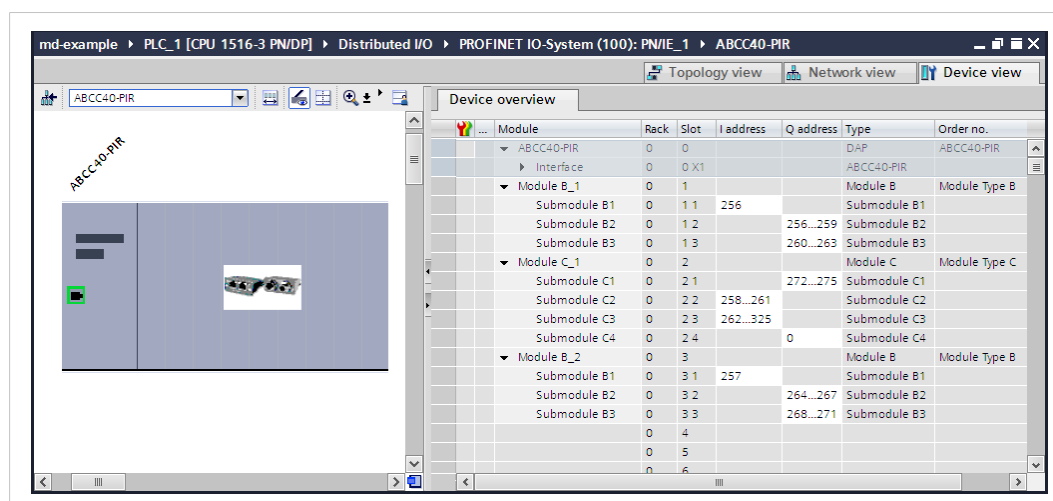


**Fig. 6**

## 5.7        Recommendations

### 5.7.1     I&M Data

Identification & Maintenance data may be used by the engineering tool to assist in resolving mismatching configurations. By comparing the OrderID from the expected GSD module with the OrderID from the I&M data of the actual, plugged module, the user can adapt the configuration to match the real identification. To ease integration and tracking of assets it is highly recommended to implement transparent I&M data for modules. When implementing I&M0 data for modules, one of the submodules in each module must be selected as a module representative. This submodule shall be listed as I&M0 Carrier and Module Representative in the I&M0 Filter Data of the IM_Options command.
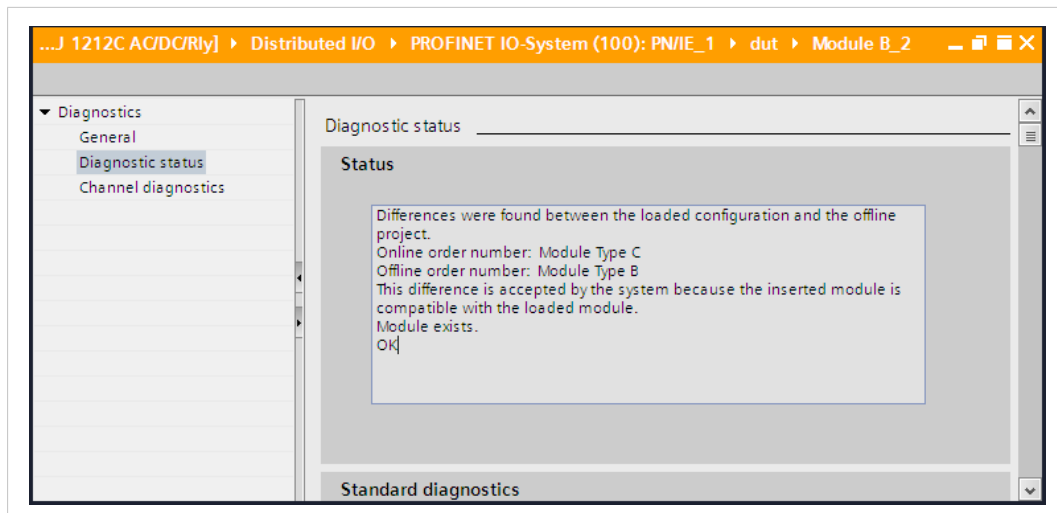


**Fig. 7**

ⓘ   *After a successful process data remap the entire RI is regenerated and any I&M0 Filter Data must be reconfigured. Refer to the Expected_Ident_Ind command for finding out the new RI.*

# 6 SNMP Agent

## 6.1 General

Simple Network Management Protocol (SNMP, see RFC1157 standard) is used in network management systems to monitor network-attached devices for conditions that warrant administrative attention. A management agent is installed in the managing station, and exchanges data via get and set requests.

## 6.2 Community Strings

SNMP uses community strings for authentication. SNMP support and default community strings are required for PROFINET comformance. For security reasons it is strongly recommended to change the values of these strings. See attributes #20 and #21 in the *Ethernet Host Object (F9h), p. 138*.

## 6.3 Management Information (MIB)

A MIB is a device database that is accessed by an SNMP agent. The Anybus CompactCom 40 PROFINET IRTsupports standardized MIBs: LLDP-MIB and MIB-II. Standardized MIBs are defined in RFC standards and contain variables that are divided into so called groups. The host application can change the values of some of the variables for the MIB-II.

## 6.4 MIB_II

The MIB-II of the Anybus CompactCom 40 PROFINET IRT contains the system- and interfaces group. The following tables show the variables according to the MIB-II standard (RFC1213) for monitoring the device status. The access authorizations refer to access via the SNMP protocol.

### 6.4.1 System Group Variables

| Variable | Access Authorizations | Description | Source of Origin |
|---|---|---|---|
| sysDescr | Read only | Description of the device. Data type: DisplayString (only printable ASCII characters). Max 255 characters. | Internal |
| sysObjectID | Read only | N/A. Value=0 | Internal |
| sysUpTime | Read only | Time since last power up (in hundredths of a second) | Internal |
| sysContact | Read/Write | Identification of the contact person for the device, including contact information. Data type: Displaystring. Max 255 characters. Factory default setting: empty string | Internal |
| sysName | Read/Write | Name of the device. Data type: Displaystring. Max 255 characters. Factory default setting: empty string | Internal |
| sysLocation | Read/Write | Physical location of the device (IM Tag Location). Data type: (DisplayString). Max 255 characters. Factory default setting: empty string | Internal |
| sysServices | Read only | Functionality of the device. Value=74, which indicates that the device has functionality that represents layers 2 (switch), 4(TCP) and 7(Application) in the OSI model. | Internal |

## 6.4.2 Interfaces Group Variables

Access authorizations for all variables are read only with values from internal sources. The number in brackets refers to the port number (1 - Port 1, 2 - Port 2, 3 - Internal port)

If nothing else is specified, the value of a variable is 0

| Variable | Data Type | Value | Description |
|---|---|---|---|
| ifNumber | integer | 3 | Number of network interfaces present. Constant |
| IfIndex(1..3) | integer | ifIndex(1) = 1<br>ifIndex(2) = 2<br>ifIndex(3) = 3 | Unique value for each interface. Constant |
| ifDescr(1..3) | octetstring | ifDescr(1) = "port-001"<br>ifDescr(2) = "port-002"<br>ifDescr(3) = "port-internal" | Information about the interface. ifDescr(1) must equal "port-001"and ifDescr(2) = "port-002" to be compatible with the STEP7 topology scanner. |
| ifType(1..3) | integer | 6 ("Ethernet-csmacd") | Type of interface |
| ifMtu(1..3) | integer | 1500 | Size of largest datagram that can be sent/received on the interface, specified in octets |
| ifSpeed(1..3) | gauge | 0 or 100 000 000 | Data transfer rate of the Ethernet port in bits per second. The speed is only shown for ports where the link status is "up". |
| ifPhysAddress(1..3) | octetstring | | MAC address for the ports |
| ifAdminStatus(1..3) | integer | 1 ("up") | Desired state of the Ethernet port |
| ifOperStatus(1..3) | integer | 1 ("up") or 2 ("down") | Current operating state of the Ethernet port. (Link = "up", No link = "down".) |
| ifLastChange(1..3) | timeticks | Time when state changed, except ifLastChange(3) = 0 | Time (since start-up) when the port changed to its current state, see previous variable. Indicated in multiples of hundredths of a second |
| ifInOctets(1..3) | counter | ifInOctets(1..3) = Number of octets | Total number of octets received on the interface, including framing characters |
| ifInUcastPkts(1..3) | counter | ifInUcastPkts(1..3) = Number of unicast packets | Number of subnetwork-unicast packets delivered to a higher-layer protocol |
| ifInNUcastPkts(1..3) | counter | ifInNUcastPkts(1..3) = Number of non-unicast packets | Number of non-unicast (i.e. subnetwork-broadcast or subnetwork-multicast) delivered to a higher-layer protocol. |
| ifInDiscards(1..3) | counter | ifInDiscards(1..3) = number of discarded packets | Number of inbound packets which were discarded, without any error detected, not to be delivered to a higher-layer protocol. (One reason to discard packages might be to free up buffer space) |
| ifInErrors(1..3) | counter | ifInErrors(1..3) = number of error packets | Number of inbound packets with errors |
| ifInUnknownProtos (1..3) | counter | ifInUnknownProtos (1..3) = Number of unknown packets | Number of packets received via the interface, discarded because of an unknown or unsupported protocol. |
| ifOutOctets(1..3) | counter | ifOutOctets(1..3) = Number of octets | Total number of octets transmitted out from the interface, including framing characters |
| ifOutUcastPkts(1..3) | counter | ifOutUcastPkts(1..3) = Number of unicast packets | Total number of packets that higher-level protocols requested to be transmitted to a subnetwork-unicast address, including those that were discarded or not sent. |
| ifOutNUcastPkts(1..3) | counter | ifOutNUcastPkts(1..3) = Number of non-unicast packets | Total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e. a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent. |
| ifOutDiscards(1..3) | counter | ifOutDiscards(1..3) = Number of discarded packets | Number of outbound packets which were discarded without any error detected, not to be transmitted. (One reason to discard packages might be to free up buffer space) |

| Variable | Data Type | Value | Description |
|---|---|---|---|
| ifOutErrors(1..3) | counter | ifOutErrors(1..3) = Number of error packets | Number of outbound packets that could not be transmitted due to errors |
| ifOutQLen(1..3) | gauge | ifOutQLen(1..3) = Number of packets in queue | Length of the output packet queue (in packets), |
| ifSpecific(1..3) | objid | .0.0 | Reference to MIB definitions specific to the particular media being used to realize the interface. Here no reference is available, so a fixed value is used for all ports. |

# 7 Media Reduncancy Protocol (MRP)

## 7.1 General

Media Redundancy Protocol (MRP) is a PROFINET specific ring protocol ensuring redundancy in the network, which can significantly decrease network downtime. It is a token based ring protocol with a master-slave hierarchy.

All the nodes in the PROFINET network part of the ring are connected using ring topology (that is, the last node is connected directly to the first node). If, at any point, the connection between two nodes would break, the data will flow the other way instead, guaranteeing that data can be sent to/from the IO Controller to the IO Device(s). The self-healing time is approximately 200 ms.

The Media Redundancy Master (MRM) is responsible for checking the functional capability of the ring network, by sending out cyclic tokens. The Media Redundancy Clients (MRC) basically work as switches that pass on the tokens. The Anybus CompactCom supports acting as a Media Redundancy Client (MRC). It also supports propagating link change to the Media Redundancy Master.

If port 2 of the device is inactivated in the Ethernet Host Object (F9h), support for MRP is removed. See *Ethernet Host Object (F9h), p. 138* , instance attribute #13, for more information.

## 7.2 GSDML Entries

MRP functionality is enabled by default in the GSD file. The settings for MRP are located at the Device Access Point (DAP). within the `<InterfaceSubmoduleItem …>`. The Anybus CompactCom is defined as "Client" with the keyword `<MediaRedundancy SupportedRole="Client"/>`.

For each physical port there are two keywords in the `<PortSubmoduleItem …>` section `<PortSubmoduleItem … SupportsRingportConfig= "true"IsDefaultRingport="true" … />`. These are set to "true" by default. To disable MRP, these two shall be set to"false".

# 8 Transparent Ethernet

## 8.1 General Information

Transparent Ethernet offers the possibility for a host application, that includes an IT implementation, to let the Anybus CompactCom handle an industrial Ethernet protocol (in this case PROFINET), without the need for extra Ethernet ports.

Ethernet communication that is not related to PROFINET is internally routed via the RMII interface to the Ethernet port and the TCP/IP stack of the host application. The IP configurations and the MAC addresses of the host application and the Anybus CompactCom must be the same.

The RMII interface is accessed through the host application connector. Please note that the 16 bit parallel interface is not available when transparent Ethernet is enabled. See the *Anybus CompactCom M40 Hardware Design Guide* for more information.
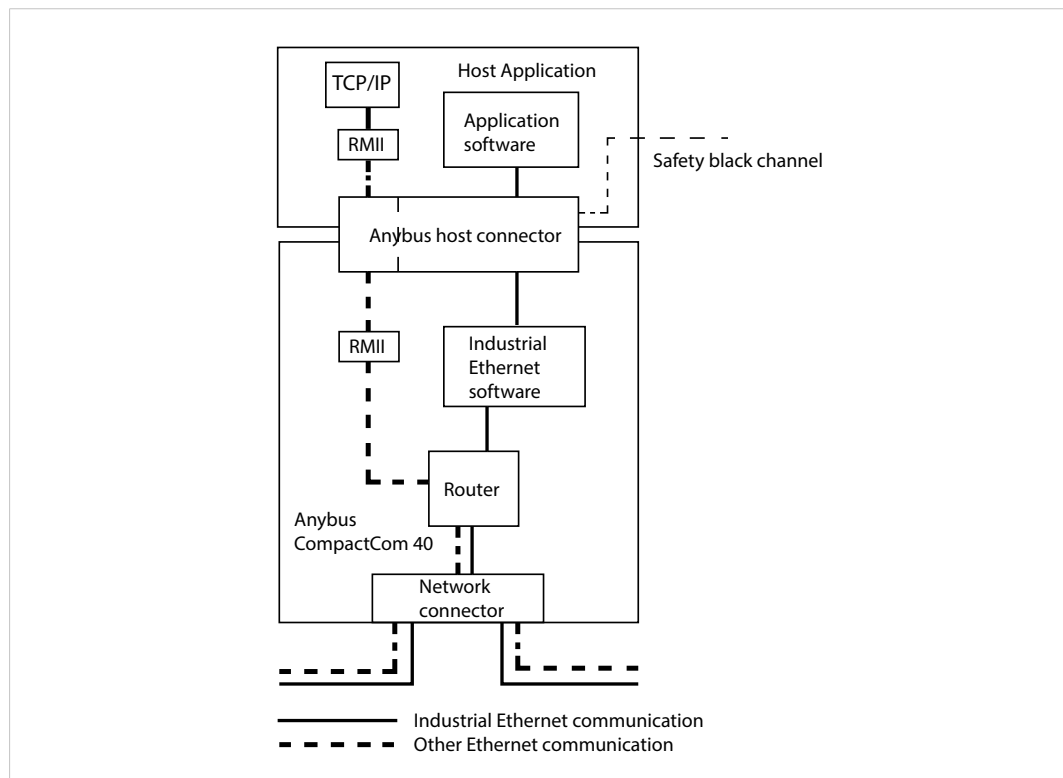


**Fig. 8**

> Transparent Ethernet has to be enabled by setting instance attribute #16 in the Anybus Object (01h) during setup, see below.
>
> 16–bit parallell mode cannot be used when using transparent Ethernet.
>
> MAC addresses and IP configurations have to be synchronized, see below.
>
> Some EtherTypes ,TCP/UDP ports and multicast MAC addresses may be reserved for use by the industrial Ethernet network. These must not be used for transparent Ethernet communication. See below for more information.
>
> The Transparent Ethernet interface only supports 100 Mbit, full duplex operation.

## 8.2 Enabling Transparent Ethernet

Transparent Ethernet is not enabled at delivery. Attribute #16 (instance #1) in the Anybus Object (01h) has to be set to 0002h during setup. If this attribute is not changed, the Anybus

CompactCom 40 PROFINET IRT will start up with full IT functionality instead of transparent Ethernet functionality. Transparent Ethernet cannot be enabled after setup is finished. Once Transparent Ethernet is enabled, no IT functionality is enabled.

## 8.3        MAC Address Synchronization

The host application and the Anybus CompactCom must use the same MAC address when communicating on Ethernet. The host application must make sure that this is the case. This can be accomplished in ether of the two ways described below:

- The pre-programmed MAC address in attribute #1 (instance #1) in the Network Ethernet Object (0Ch) is read and used by the host application when communicating on Ethernet.

- The Ethernet Host Object (F9h, instance #1, attribute #1) is implemented in the application, set with a MAC address provided and used by the application. At initialization, the Anybus CompactCom will read and then use the application provided MAC address from this object.

## 8.4        IP Configuration Synchronization

The host application TCP/IP stack and the Anybus CompactCom 40 PROFINET IRT TCP/IP stack must use the same IP configuration when communicating on Ethernet. The Anybus CompactCom 40 PROFINET IRT will write its currently used IP configuration to instance attribute #16 in the Ethernet Host Object (F9h) whenever the configuration is assigned or changed. The host application must use this configuration. DNS server and domain names can be read from the Network Configuration Object (04h) after an IP configuration update.

## 8.5        Routing Restrictions

The internal router receives all frames from the network. The frames that are intended for the industrial Ethernet network internal software, are recognized and routed to the Anybus CompactCom. The remaining Ethernet frames will be routed to the host application. Some restrictions apply to the use of e.g. UDP and TCP ports, sometimes also depending on industrial Ethernet network. If the host application is intended only for use with PROFINET, the restrictions for the other networks can be ignored.

> **!** The host application is responsible for taking the following restrictions into consideration. If they are not followed, the Ethernet communication will not work correctly.

### 8.5.1 EtherTypes

The Anybus CompactCom internally uses bit 12 and bit 13 (mask 3000h) in the EtherType. Thus the host application cannot implement protocols using Ethertype, where these bits are used..

The following EtherTypes are used by PROFINET and should not be used by the host application:

- 8892h (PNIO)

- 88CCh (LLDP)

- 88E3h (MRP)

### 8.5.2 Multicast MAC Addresses

The host application must not transmit or receive any data from and to the following MAC addresses as they may be used by the industrial Ethernet network (PROFINET):

- 01-0E-CF-XX-XX-XX

- 01-80-C2-00-00-0E

- 01-00-5E-40-F8-00 ... 01-00-5E-40-FB-FF

- X3-XX-00-00-00-00

(X: any number 0-F)

### 8.5.3 UDP/TCP Ports

The following ports may be used by the Anybus CompactCom, and must not be used by the host application:

- UDP 67 & 68 (DHCP)

- UDP 161 (SNMP)

- UDP 3250 (HICP)

The following ports are reserved for use by PROFINET:

- UDP 34962 (PROFINET RT Unicast)

- UDP 34963 (PROFINET RT Multicast)

- UDP 34969 (PROFINET RPC Context Manager)

- UDP 53247 (PROFINET RPC Client/Server)

The following ports are reserved for use by EtherNet/IP:

- UDP 2222 (Implicit messaging)

- UDP & TCP 44818 (Explicit messaging)

The following port is reserved for use by Modbus TCP:

- TCP 502 (Modbus messaging)

# 9    Firmware Upgrade

The Anybus CompactCom 40 PROFINET IRT firmware can be updated either by running the Firmware Manager II tool (FMII), available at www.anybus.com/support, or by downloading the firmware upgrade file directly to the host application file system. For any of these methods to work the following needs to be implemented and/or performed:

- HICP needs to be enabled (FMII only).

- An FTP server needs to be implemented in the host application.

- A directory named "firmware" in the host application FTP root.

- The file module.nfo in the "firmware" directory in theAnybus CompactCom file system has to be copied to the "firmware" directory in the host application file system (FMII only). The Anybus CompactCom file system is accessed via the Anybus File System Object (0Ah).

Once a firmware file has been downloaded, the host application must be able to:

- detect a new file in the "firmware" directory

- download this file to the "firmware" directory in the Anybus CompactCom (The Anybus CompactCom file system is accessed via the Anybus File System Object (0Ah).)

The firmware will be updated upon the next reset of the Anybus CompactCom 40 PROFINET IRT.

# 10      Anybus Module Objects

## 10.1    General Information

Standard Objects:

Network Specific Objects:

## 10.2        Anybus Object (01h)

### Category

Basic

### Object Description

This object assembles all common Anybus data, and is described thoroughly in the general *Anybus CompactCom 40 Software Design Guide*.

> **!**    Instance attribute #16 has to be set to 0002h during SETUP state to enable Transparent Ethernet functionality.

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |
| | Set_Attribute |
| | Get_Enum_String |

### Object Attributes (Instance #0)

(Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.)

### Instance Attributes (Instance #1)

Basic

| # | Name | Access | Type | Value |
|---|---|---|---|---|
| 1 | Module type | Get | UINT16 | 0403h (Standard Anybus CompactCom 40) |
| 2... 11 | - | - | - | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 12 | LED colors | Get | struct of:<br>UINT8 (LED1A)<br>UINT8 (LED1B)<br>UINT8 (LED2A)<br>UINT8 (LED2B) | <u>Value:</u> <u>Color:</u><br>01h  Green<br>02h  Red<br>01h  Green<br>02h  Red |
| 13... 15 | - | - | - | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 16 | GPIO configuration | Get/Set | UINT16 | Configuration of the host interface GIO pins. To enable Transparent Ethernet, this attribute has to be set to 0002h during SETUP state. |
| 17 | Virtual attributes | Get/Set | - | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 18 | Black list/White list | Get/Set | | |
| 19 | Network time | Get | UINT64 | 0 (Not supported) |

## 10.3 Diagnostic Object (02h)

### Category

Extended

### Object Description

This object provides a standardized way of handling host application events & diagnostics, and is thoroughly described in the general *Anybus CompactCom 40 Software Design Guide*.

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| | Create |
| | Delete |
| **Instance:** | Get_Attribute |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|---|---|---|---|
| 1... 4 | - | - | - | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 11 | Max no. of instances | Get | UINT16 | 5+1 (Of the maximum number of instances there should always be one instance reserved for an event of severity level 'Major, unrecoverable', to force the module into the 'EXCEPTION'-state.) |
| 12 | Supported functionality | Get | BITS32 | Bit 0: "0" (Latching events are not supported)<br>Bit 1 - 31: reserved (shall be "0" ) |

### Instance Attributes (Instance #1)

Extended

| # | Name | Access | Data Type | Value |
|---|---|---|---|---|
| 1 | Severity | Get | UINT8 | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 2 | Event Code | Get | UINT8 | |
| 3 | NW specific data | Get | Array of UINT8 | Optional network specific information, see below. |
| 4 | Slot | Get | UINT16 | Consult the general Anybus CompactCom 40 Software Design Guide for further information. |
| 5 | ADI | Get | UINT16 | |
| 6 | Element | Get | UINT8 | |
| 7 | Bit | Get | UINT8 | |

Major unrecoverable events cause the module to disconnect itself from the network, thus preventing network participation. Other severity levels either produce a Channel Diagnostic entry/alarm or a Generic Diagnostic entry/alarm, depending on the Event Code:

| Event Code | Result |
|---|---|
| 0...FEh | Module issues a Channel Diagnostic entry/alarm. The Event Code will be translated and represented as the Channel Error Type as follows: |

| Code: | Event (Anybus): | Channel Error Type (PROFINET): |
|---|---|---|
| 10h | Generic Error | Error |
| 20h | Current | Short circuit |
| 21h | Current, device input side | Short circuit |
| 22h | Current, inside the device | Short circuit |
| 23h | Current, device output side | Short circuit |
| 30h | Voltage | Overvoltage |
| 31h | Mains Voltage | Overvoltage |
| 32h | Voltage inside the device | Overvoltage |
| 33h | Output Voltage | Overvoltage |
| 40h | Temperature | Overtemperature |
| 41h | Ambient Temperature | Overtemperature |
| 42h | Device Temperature | Overtemperature |
| 50h | Device Hardware | Error |
| 60h | Device Software | Error |
| 61h | Internal Software | Error |
| 62h | User Software | Error |
| 63h | Data Set | Error |
| 70h | Additional Modules | Error |
| 80h | Monitoring | Error |
| 81h | Communication | Error |
| 82h | Protocol Error | Error |
| 90h | External Error | Error |
| F0h | Additional Functions | Error |

| Event Code | Result |
|---|---|
| FFh | Module issues a Generic Diagnostic entry/alarm based on network specific data. |
| | See details below. |

## Details: Network Specific Data

Network specific diagnostic data serves as the payload in the PROFINET diagnostic alarm. The data contains an identifier (UserStructureIdentifier) that describes the structure of the data.

The following identifier values are supported:

- 8000h (Channel Diagnostic)

- 8002h (Extended Channel Diagnostic)

- 8003h (Qualified Channel Diagnostic)

| Byte | Contents |
|---|---|
| 1 | UserStructureIdentifier, low byte |
| 2 | UserStructureIdentifier, high byte |
| 3... 16 | Data |

## Object Error Codes

| Code | Error |
|------|-------|
| 03h | API does not exist |
| 04h | No module inserted in the specified slot |
| 05h | No submodule inserted in the specified subslot |
| 06h | Slot number specified is out-of-range |
| 07h | Subslot number specified is out-of-range |
| 08h | Failed to add the channel diagnostic entry |
| 09h | Failed to send the channel diagnostic alarm |
| 0Ah | Channel number out-of-range |
| 0Bh | ChannelPropType out-of-range |
| 0Ch | ChannelPropDir out-of-range |
| 0Dh | ChannelPropAcc out-of-range |
| 0Eh | ChannelPropMaintReq out-of-range |
| 0Fh | ChannelPropMaintDem out-of-range |
| 10h | UserStructIdent out-of-range |
| 11h | ChannelErrType out-of-range |
| FFh | Unknown error |

## 10.4       Network Object (03h)

**Category**

Basic

**Object Description**

PROFINET specific information for this object is given below. For more information regarding this object, consult the general *Anybus CompactCom 40 Software Design Guide*.

**Supported Commands**

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |
| | Set_Attribute |
| | Get_Enum_String |
| | Map_ADI_Write_Area |
| | Map_ADI_Read_Area |
| | Map_ADI_Write_Ext_Area |
| | Map_ADI_Read_Ext_Area |

**Object Attributes (Instance #0)**

(Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.)

## Instance Attributes (Instance #1)

Basic

| # | Name | Access | Type | Value |
|---|---|---|---|---|
| 1 | Network type | Get | UINT16 | 0089h (PROFINET IRT) or<br>009Dh (PROFINET IRT Fiber Optic) |
| 2 | Network type string | Get | Array of CHAR | "PROFINET IRT" or<br>"PROFINET IRT Fiber Optic" |
| 3 | Data format | Get | ENUM | 01h (MSB first) |
| 4 | Parameter data support | Get | BOOL | True |
| 5 | Write process data size | Get | UINT16 | Current write process data size (in bytes)<br>Updated on every successful Map_ADI_Write_Area. (Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.) |
| 6 | Read process data size | Get | UINT16 | Current read process data size (in bytes)<br>Updated on every successful Map_ADI_Read_Area. (Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.) |
| 7 | Exception Information | Get | UINT8 | Additional information available if the module has entered the EXCEPTION state.<br><br>Value:    Meaning:<br>00h    No information<br>01h    Illegal value<br>02h    Wrong data size<br>03h    Illegal response<br>04h    Missing MAC address (Only valid for Anybus IP)<br>05h    Command timeout (an expected command not received within time limit)<br>06h    The implementation of the Modular Device Object in the host application is not correct. Either Number of ADIs per module exceeds 4095 or an error response is received on the Get_List command.<br>07h    Wrong version in PROFIenergy object<br>08h    Wrong number of instances in PROFIenergy object<br>09h    Invalid value of MaxAr in PROFINET IO object |

## Command Details: Map_ADI_Write_Ext_Area

**Details**

| Command Code: | 12h |
|---|---|
| **Valid For:** | Instance |

**Description**

This command is only supported by Anybus CompactCom 40 devices.

This command is equivalent to Map_ADI_Write_Area, but can map more than 256 bytes of data. It supports mapping fractional byte size types, and it can be used to map only specific parts of an ADI.

It maps an ADI as Write Process Data. If successful, the response data contains the offset, in bits, for the mapped ADI from the start of the Write Process Data area.

• Mapping an ADI more than once (i.e. map it multiple times to the Read- or Write Process Data, or map it to both the Read- and Write Process Data) is not accepted by all networks.

• It is not allowed to mix mapping commands Map_ADI_Read/Write_Area and Map_ADI_Read/Write_Ext_ Area within one area (Read/Write).

• It is recommended to only map one item for each mapping command during initial development, since data area offset is only given for the first mapping item, and all mapping items may be rejected using one single error code.

• All mapped elements, except those of types BIT1-BIT7 and PADx, must be byte aligned.

• The only implicit padding done is from the very last mapped item up to byte alignment, since the process data needs to be of byte size when the setup is complete.

• Explicit padding is done either through available ADI elements of PADx type, or through the imaginary ADI 0, which is assumed to be an array with 255 elements of type PAD1. Explicit padding of process data is the only correct use of ADI 0. Padding bits might not be visible on the network.

• This command may permanently alter the state of the Anybus CompactCom 40 PROFINET IRT even though the command is returned with an error. Network specific restrictions may lead to n mapping items to be accepted, but with an error on mapping item n+1. If so, the mappings up to and including n will be accepted, but all other mapping items, starting with n+1, are rejected. The number of accepted mappings is declared in CmdExt[ 0 ] of the answer.

• Certain Anybus implementations allow the network to remap the Process Data during runtime. (Consult the general *Anybus CompactCom 40 Software Design Guide*, Application Data Object (FEh) for further information.)

See also...

*Anybus CompactCom 40 Software Design Guide*, Application Object (FFh)

> **!** Error control is only performed on the command parameters. The Anybus module does <u>not</u> verify the correctness of these parameters by a read of the actual ADI attributes.

- Command details:

| Field | Contents |
|---|---|
| CmdExt[0] | The number of mapping items to add (0-217) |
| CmdExt[1] | Reserved. Set to 0 |
| MsgData[0-1] | New mapping item 1: ADI number |
| MsgData[2] | New mapping item 1: Number of elements in the ADI |
| MsgData[3] | New mapping item 1: Index to the first element to map (0-254) |
| MsgData[4] | New mapping item 1: Number of consecutive elements to map (1-255) |
| MsgData[5] | New mapping item 1: Number of type descriptors (1-255) |
| MsgData[6..n] | New mapping item 1: Array of type specifiers for each mapped element |
| ... | Repeat MsgData[0-n] (as above) for mapping item 2 and onwards. |

- Response details (Success):

| Field | Contents |
|---|---|
| CmdExt[0] | The number of accepted mapping items (0-217) |
| MsgData[0] | Bit offset of the mapped ADI from the start of the Write Process Data (Least significant byte) |
| MsgData[1] | Bit offset of the mapped ADI from the start of the Write Process Data |
| MsgData[2] | Bit offset of the mapped ADI from the start of the Write Process Data |
| MsgData[3] | Bit offset of the mapped ADI from the start of the Write Process Data (Most significant byte) |

- Response details (Error):

| Error | Contents | |
|---|---|---|
| Invalid CmdExt[0] | The number of accepted mapping items, before an error occurred | |
| Invalid State | Mapping of ADIs is only allowed in the SETUP state | |
| Object Specific Error | Object specific error, see MsgData[1] for details: | |
| | 01h: Invalid data type | The data type is not valid for Process Data |
| | 02h: Invalid number of elements | The number of elements is not valid (zero, or too many elements) |
| | 03h: Invalid total size | The requested mapping is denied because the resulting total data size would exceed the maximum permissible (depending on network type) |
| | 06h: Invalid map command sequence | The order in which the commands were received is invalid |
| | 07h: Invalid mapping command | Inconsistencies in the command makes it impossible to parse |
| | 08h: Bad alignment | The alignment rules for process data are not followed |
| | 09h: Invalid use of ADI 0 | ADI 0 is an array (255 elements) of type PAD1 |
| | FFh: Network specific restriction | Mapping an ADI other than the previous (non-padding) ADI, must be done on a byte boundary. Bits from multiple ADIs cannot be mapped to the same byte. Indicated by network specific error code 01h (total error response is FFh FFh 01h). |

Error control is only performed on the command parameters. The Anybus module does not verify the correctness of these parameters by a read of the actual ADI attributes.

## Command Details: Map_ADI_Read_Ext_Area

**Details**

| | |
|---|---|
| **Command Code:** | 13h |
| **Valid For:** | Instance |

**Description**

This command is only supported by Anybus CompactCom 40 devices.

This command is equivalent to Map_ADI_Read_Area, but can map more than 256 bytes of data.

It is identical to Map_ADI_Write_Ext_Area, described above, except that it maps ADIs to Read Process Data.

## 10.5        Network Configuration Object (04h)

### Category

Extended

### Object Description

This object holds network specific configuration parameters that may be set by the end user. A reset command (factory default) issued towards this object will result in all instances being set to their default values.

As soon as the used combination of IP address, Subnet mask and Gateway is changed, the module informs the application by writing the new set to instance #1, attribute #16 in the Ethernet Host Object (F9h).

See also...

- *Communication Settings, p. 14*

- 

- *Ethernet Host Object (F9h), p. 138*

> **!** Allowing the following instances to be set by the host application during start-up will inhibit the possibility to pass conformance tests.

### Supported Commands

| **Object:** | Get_Attribute |
| --- | --- |
| | Reset |
| **Instance:** | Get_Attribute |
| | Set_Attribute |
| | Get_Enum_String |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value | Description |
| --- | --- | --- | --- | --- | --- |
| 3 | Number of instances | Get | UINT16 | 10 | Supported number of instances |
| 4 | Highest instance number | Get | UINT16 | 21 | Highest instance number |

(Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.)

## Instance Attributes (Instance #3, IP Address)

Value is used after module reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "IP address" <br> (Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 04h (= UINT8) |
| 3 | Number of elements | Get | UINT8 | 04h (four elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of UINT8 | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset. <br> Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |
| 6 | Configured Value | Get | Array of UINT8 | Holds the configured value, which will be written to attribute #5 after the module has been reset. <br> Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |

> **!** This attribute should not be set by the application at every power on, as this would cause certification problems.

## Instance Attributes (Instance #4, Subnet Mask)

Value is used after module reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "Subnet mask" <br> (Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 04h (= UINT8) |
| 3 | Number of elements | Get | UINT8 | 04h (four elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of UINT8 | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset. <br> Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |
| 6 | Configured Value | Get | Array of UINT8 | Holds the configured value, which will be written to attribute #5 after the module has been reset. <br> Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |

> **!** This attribute should not be set by the application at every power on, as this would cause certification problems.

## Instance Attributes (Instance #5, Gateway Address)

Value is used after module reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "Gateway"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 04h (= UINT8) |
| 3 | Number of elements | Get | UINT8 | 04h (four elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of UINT8 | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |
| 6 | Configured Value | Get | Array of UINT8 | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |

> **!** This attribute should not be set by the application at every power on, as this would cause certification problems.

## Instance Attributes (Instance #6, DHCP Enable)

Value is used after module reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "DHCP"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 08h (= ENUM) |
| 3 | Number of elements | Get | UINT8 | 01h (one element) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | ENUM | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>(Multilingual, see page *61*)<br><br>Value     String     Meaning<br>00h     "Disable"     DHCP disabled (default)<br>01h     "Enable"     DHCP enabled |
| 6 | Configured Value | Get | ENUM | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br><br>Value     String     Meaning<br>00h     "Disable"     DHCP disabled<br>01h     "Enable"     DHCP enabled |

> **!** Do not set this unless the end user explicitly would like to turn DHCP on. Normally the PROFINET IO Controller assigns the IP address.

## Instance Attributes (Instance #9, DNS1)

This instance holds the address to the primary DNS server. Changes are valid after reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "DNS1"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 04h (= UINT8) |
| 3 | Number of elements | Get | UINT8 | 04h (four elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of UINT8 | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |
| 6 | Configured Value | Get | Array of UINT8 | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |

## Instance Attributes (Instance #10, DNS2)

This instance holds the address to the secondary DNS server. Changes are valid after reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "DNS2"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 04h (= UINT8) |
| 3 | Number of elements | Get | UINT8 | 04h (four elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of UINT8 | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |
| 6 | Configured Value | Get | Array of UINT8 | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br>Valid range: 0.0.0.0 - 255.255.255.255 (Default =0.0.0.0) |

## Instance Attributes (Instance #11, Host name)

This instance holds the host name of the module. Changes are valid after reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "Host name"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 07h (= CHAR) |
| 3 | Number of elements | Get | UINT8 | 40h (64 elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of CHAR | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>Host name, 64 characters |
| 6 | Configured Value | Get | Array of CHAR | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br>Host name, 64 characters |

## Instance Attributes (Instance #12, Domain name)

This instance holds the domain name. Changes are valid after reset.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "Host name"<br>(Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 07h (= CHAR) |
| 3 | Number of elements | Get | UINT8 | 30h (48 elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of CHAR | If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset.<br>Domain name, 48 characters |
| 6 | Configured Value | Get | Array of CHAR | Holds the configured value, which will be written to attribute #5 after the module has been reset.<br>Domain name, 48 characters |

## Instance Attributes (Instances #13 - #19)

(Reserved)

## Instance Attributes (Instance #20, Station Name)

The Station Name identifies the Anybus module on PROFINET. If this value is changed by the host application during runtime, a reset is required in order for changes to have effect. Changes made through DCP will have immediate effect, however.

The Station Name field shall be coded as data type CHAR with 1 to 240 characters. The definition of RFC 5890 and the following syntax applies:

• 1 or more labels, separated by [.]

• Total length is 1 to 240

• Label length is 1 to 63

• Labels consist of [a-z, 0-9, -]

• Labels do not start with [-]

• Labels do not end with [-]

• The first label must not have the form "port-xyz" or "port-xyz-abcde", where a, b, c, d, e, x, y, z = 0...9, to avoid similarity with the field AliasNameValue

• Station names must not have the form n.n.n.n, where n = 0...999

> ❗ If the Station name does not pass validation by the Anybus module, an error response with error code 0Ch will be returned to the application and the Station name will not be changed/saved.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "Station name" (Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 07h (= CHAR) |
| 3 | Number of elements | Get | UINT8 | F0h (240 elements) |
| 4 | Descriptor | Get | UINT8 | 07h (read/write/shared access) |
| 5 | Value | Get/Set | Array of CHAR | The current Station name<br>If read, the actual value will be received. If written, the written value is reflected in attribute #6 until a reset. |
| 6 | Configured value | Get | Array of CHAR | Holds the configured value, which will be written to attribute #5 after the module has been reset. |

> ❗ This attribute shall normally not be set by the application. The station name is normally set by the end user via the network. The host application shall use this attribute when the end user has the possibility to edit the station name through the application, and chooses to do so.
>
> This attribute should not be set by the application at every power on, as this would cause certification problems.

## Instance Attributes (Instance #21, F-Address)

This instance holds the F-Address, which is the PROFIsafe address for the safety module. This instance has no effect unless the attribute #1 (Safety enabled) is set to TRUE in the Functional Safety host object (instance #1). If the attribute is set to FALSE in the Functional Safety host object the application is advised to hide this instance to the end-user. See *Functional Safety Object (E8h), p. 106*

If this value is changed by the host application during runtime, a reset is required in order for changes to have effect.

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | Name | Get | Array of CHAR | "F-Address" (Multilingual, see page *61*) |
| 2 | Data type | Get | UINT8 | 05h (= UINT16) |
| 3 | Number of elements | Get | UINT8 | 01h (one elements) |
| 4 | Descriptor | Get | UINT8 | 03h (read/write access) |
| 5 | Value | Get/Set | UINT16 | F-Address set by the host application. Range: 1 - 65534 (Default: 1) |
| 6 | Configured value | Get/Set | UINT16 | The configured value that will be used after restart |

## Multilingual Strings

The instance names and enumeration strings in this object are multilingual, and are translated based on the current language settings as follows:

| Instance | English | German | Spanish | Italian | French |
|----------|---------|--------|---------|---------|--------|
| 3 | IP address | IP-Adresse | Dirección IP | Indirizzo IP | Adresse IP |
| 4 | Subnet mask | Subnetzmaske | Masac. subred | Sottorete | Sous-réseau |
| 5 | Gateway | Gateway | Pasarela | Gateway | Passerelle |
| 6 | DHCP | DHCP | DHCP | DHCP | DHCP |
| | Enable | Einschalten | Activado | Abilitato | Activé |
| | Disable | Ausschalten | Desactivado | Disabilitato | Désactivé |
| 7 | Comm 1 | Komm 1 | Comu 1 | Connessione 1 | Comm 1 |
| 8 | Comm 2 | Komm 2 | Comu 2 | Connessione 2 | Comm 2 |
| 9 | DNS1 | DNS 1 | DNS Primaria | DNS1 | DNS1 |
| 10 | DNS2 | DNS 2 | DNS Secundia. | DNS2 | DNS2 |
| 11 | Host name | Host name | Nombre Host | Nome Host | Nom hôte |
| 12 | Domain name | Domain name | Nobre Domain | Nome Dominio | Dom Domaine |
| 20 | Station name | Stationsname | Nom. Estacion | Nome Stazione | Nom Station |
| 21 | F-Address | F-Adresse | Dirección-F | Indirizzo-F | F-Adresse |

## Command Details: Reset

**Category**

**Details**

| Command Code: | 05h |
|---|---|
| **Valid for:** | Object Instance |

**Description**

A reset command to this object will result in that all instances are set to their default values.

It is optional to implement support for this command.

- Command Details

| Field | Comments |
|---|---|
| CmdExt[0] | 00h (Reserved) |
| CmdExt[1] | 01h (Factory default reset) |

- Response Details

    (No data)

## 10.6      File System Interface Object (0Ah)

**Category**

Extended

**Object Description**

This object provides an interface to the built-in file system. Each instance represents a handle to a file stream and contains services for file system operations.

This object is thoroughly described in *Anybus CompactCom 40 Software Design Guide*.

## 10.7       Network Ethernet Object (0Ch)

### Category

Extended

### Object Description

This object provides Ethernet-specific information to the application.

The object has three instances, each corresponding to a port:

| Instance # | Port |
|------------|------|
| 1 | Internal port |
| 2 | Port 1 |
| 3 | Port 2 |

Each instance provides statistic counters for the port with the MAC address given in attribute #4. This information can e.g be presented on internal web pages, if present, using the JSON script language.

> **ⓘ** *Instance attributes #1 - #3 are reserved and used for backwards compatibility with earlier applications.*

### Supported Commands

| **Object:** | Get_Attribute |
|---|---|
| **Instance:** | Get_Attribute |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "Network Ethernet" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | 3 |
| 4 | Highest instance no. | Get | UINT16 | 3 |

### Instance Attributes (Instances #1 - #3)

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | MAC Address | Get | Array of UINT8 | Reserved, used for backwards compatibility.<br>(Device MAC address.)<br>(See also *Ethernet Host Object (F9h), p. 138*) |
| 2 | Port 1 MAC Address | Get | Array of UINT8 | Reserved, used for backwards compatibility.<br>(MAC address for port 1 (mandatory for the LLDP protocol))<br>(See also *Ethernet Host Object (F9h), p. 138*) |
| 3 | Port 2 MAC Address | Get | Array of UINT8 | Reserved, used for backwards compatibility.<br>(MAC address for port 2 (mandatory for the LLDP protocol))<br>(See also *Ethernet Host Object (F9h), p. 138*) |
| 4 | MAC Address | Get | Array of UINT8 | MAC address of the port. |

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 5 | Interface Counters | Get | Array of UINT32 | Array containing MIB-II interface counters (rfc1213)<br>See table below for array indices. |
| 6 | Media Counters | Get | Array of UINT32 | Array containing Ethernet-Like MIB counters for the port.<br>See table below for array indices.<br>Note: This attribute is not valid for the internal port, i.e. instance #1 of the object |

## Interface Counters

Array indices of Interface Counters attribute (#5)

| Index | Name | Description |
|-------|------|-------------|
| 0 | In octets | Octets received on the interface |
| 1 | In Unicast Packets | Unicast packets received on the interface |
| 2 | In Non-Unicast Packets | Non-unicast packets (multicast/broadcast) packets received on the interface |
| 3 | In Discards | Inbound packets received on the interface but discarded |
| 4 | In Errors | Inbound packets that contain errors (does not include In Discards) |
| 5 | In Unknown Protos | Inbound packets with unknown protocol |
| 6 | Out Octets | Octets transmitted on the interface |
| 7 | Out Unicast packets | Unicast packets transmitted on the interface |
| 8 | Out Non-Unicast Packets | Non-unicast (multicast/broadcast) packets transmitted on the interface |
| 9 | Out Discards | Outbound packets discarded |
| 10 | Out Errors | Outbound packets that contain errors |

## Media Counters

Array indices of Media Counters attribute (#6)

| Index | Name | Description |
|-------|------|-------------|
| 0 | AlignmentErrors; | Frames received that are not an integral number of octets in length |
| 1 | FCSErrors; | Frames received that do not pass the FCS check |
| 2 | SingleCollisions; | Successfully transmitted frames which experienced exactly one collision |
| 3 | MultipleCollisions; | Successfully transmitted frames which experienced more than one collision |
| 4 | SQETestErrors; | Number of times SQE test error is generated |
| 5 | DeferredTransmissions; | Frames for which first transmission attempt is delayed because the medium is busy |
| 6 | LateCollisions; | Number of times collision is detected later than 512 bit-times into the transmission of a packet |
| 7 | ExcessiveCollisions; | Frames for which transmission fails due to excessive collisions |
| 8 | lMACTransmitErrors; | Frames for which transmission fails due to an internal MAC sublayer transmit error |
| 9 | lCarrieSenseErrors; | Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame |
| 10 | lFrameTooLong; | Frames received that exceed the maximum permitted frame size |
| 11 | lMACRecieveErrors; | Frames for which reception on an interface fails due to an internal MAC sublayer receive error |

## 10.8     Network PROFINET IO Object (0Eh)

### Category

Extended

### Object Description

When the application maps ADIs to process data during start-up, the Anybus CompactCom 40 PROFINET IRT will create the module configuration as described in *Real Identification (RI), p. 20*. The modules in the GSDML file must then be described in the same way. The GSDML file provided by HMS provides a few examples based on this way of describing modules.

If the end-user wishes to define modules in another way the application must provide the module configuration to the Anybus CompactCom 40 PROFINET IRT. This is achieved by using the following commands:

- API_Add

- Plug_Module

- Plug_Submodule

- Plug_Submodule_Ext

These commands need to be sent after the process data is mapped, and before sending setup complete to the Anybus CompactCom 40 PROFINET IRT.

**Example:**

Initially, the application maps ADIs as process data by calling all or some of the functions below:

- Map_ADI_Write_Area (10h)

- Map_ADI_Read_Area (11h)

- Map_ADI_Write_Ext_Area (12h)

- Map_ADI_Read_Ext_Area (13h)

Modules and submodules are now created based on this information as described in *Real Identification (RI), p. 20*.

1.  Call API_Add to add an API.

2.  Call Plug_Module to add a module to the API.

3.  Call Plug_Submodule one or more times to add submodules to the module.

4.  Repeat steps 2 and 3 to add modules to the API.

After the configuration is complete, call setup complete.

See also ...

**Removing and Exchanging Modules and Submodules**

If the RI has been created by the host application through custom configuration, there are ways of removing modules and plug new modules during runtime.

The application will be notified by the command Cfg_Mismatch_Ind for every submodule that does not match. This information will also be provided in the command Expected_Ident_Ind. The application can then decide to remove the plugged module by issuing the command Pull_Module. This will remove the whole module and its submodules. Then, based on the information received from either Cfg_Mismatch_Ind or Expected_Ident_Ind, the application can adopt to the PLC configuration by issuing new Plug_Module, Plug_Submodule and Plug_ Submodule_Ext commands.

See also ...

- *Configuration Mismatch, p. 22*
- *Custom Configuration mismatch, p. 150* (flowchart)

## Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| | Plug_Module (see below) |
| | Plug_Submodule (see below) |
| | Plug_Submodule_Ext (see below) |
| | Pull_Module (see below) |
| | Pull_Submodule (see below) |
| | API_Add (see below) |
| | Appl_State_Ready (see below) |
| | AR_Abort (see below) |
| | IM_Options (see below) |
| | Ident_Change_Done (see below) |
| | Add_Safety_Module (see below) |
| **Instance:** | Get_Attribute |

## Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of Char | "Network PROFINET IO" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | 0001h |
| 4 | Highest instance number | Get | UINT16 | 0001h |

## Instance Attributes (Instance #1)

### Extended

| # | Name | Access | Type | Value |
|---|------|--------|------|-------|
| 1 | OnLineTrans | Get | UINT32 | Diagnostic counters; keeps track of the number of on-line transitions |
| 2 | OffLineTrans | Get | UINT32 | Diagnostic counters; keeps track of the number of off-line transitions |
| 3 | (reserved) | | | |
| 4 | Last AbortInd ReasonCode | Get | UINT16 | Reason code for most recent Abort indication<br>See command details for command AR_Abort |
| 5 | AddedApis | Get | UINT16 | Returns the number of APIs added (including API 0) |
| 6 | ApiList | Get | Array of UINT32 | First element will always be zero and the second element will contain an additional API. Length of the array is determined by parameter "AddedApis". |
| 7 | EstablishedArs | Get | UINT16 | The number of Application Relationships currently established |
| 8 | ArList | Get | Array of UINT16 | Array of Application Relationship handles. Length of array is determined by parameter "EstablishedArs". |
| 9 | - | - | - | - |
| 10 | Port 1 MAC Address | Get | Array of UINT8 | 6 Byte PROFINET Port 1 MAC address<br>See also...<br>  - *Ethernet Host Object (F9h), p. 138* |
| 11 | Port 2 MAC Address | Get | Array of UINT8 | 6 Byte PROFINET Port 2 MAC address<br>See also...<br>  - *Ethernet Host Object (F9h), p. 138* |

## Command Details: Plug_Module

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 10h |
| **Valid for:** | Object Instance |

### Description

This command may be called during start-up to specify the Real Identification. It may also be called during runtime in case there are changes to the Real Identification.

> (i) *It is only permitted to issue this command if **API_Add** has been issued first.*

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | | |
| Data[0] | API (low word, low byte) | Application Process Instance (API) |
| Data[1] | API (low word, high byte) | See also... |
| Data[2] | API (high word, low byte) | - *Application Data Instances (ADIs), p. 15* |
| Data[3] | API (high word, high byte) | |
| Data[4] | SlotNr (low byte) | Destination slot for module. |
| Data[5] | SlotNr (high byte) | Range: 0... 0x7FFF |
| Data[6] | ModIdent (low word, low byte) | Module identified as state in the GSD-file |
| Data[7] | ModIdent (low word, high byte) | |
| Data[8] | ModIdent (high word, low byte) | |
| Data[9] | ModIdent (high word, high byte) | |

See also...

## Command Details: Plug_Submodule

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 11h |
| **Valid for:** | Object Instance |

### Description

This command may be called during start-up to specify the Real Identification. It may also be called during runtime in case there are changes to the Real Identification. In such case, the Anybus will automatically issue a **Plug** or **Plug Wrong Submodule**-alarm to the IO Controller.

A submodule plugged with this command can hold IO data to the master, from the master or data in both directions. It is also possible to plug submodules which do not carry any data at all.

The Anybus CompactCom 40 PROFINET IRT supports up to 128 submodules in total.

---

ⓘ  *In case the slot number in the command is set to 0 (zero), the ADI number must also be 0 (zero), since slot 0 cannot hold any actual data.*

*It is only permitted to issue this command if **API_Add** has been issued first.*

*The **Interface**- and **Port** submodules have to be plugged in order to pass certification tests.*

*The interface and port submodule can only be plugged during the SETUP-state. Any attempt to plug these submodules during runtime will result in error.*

*No more than one safety submodule can be plugged at any time. If, during NW_INIT, it is determined that no safety module is attached, and a safety submodule is plugged in state SETUP, an exception will be raised. If an attempt to plug a safety submodule is done during runtime, when no safety module is attached, the request will be rejected.*

---

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | Bit 0: Submodule type | 0: Normal I/O submodule<br>1: Safety submodule (no ADI mapping, set Data[0..5] to 0) |
| | Bit 1-7: (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | (reserved, set to zero) | (reserved for future use) |
| Data[0] | ADI number (low byte), Read | Reference to the ADI previously mapped with Map_ADI_Read_Area. |
| Data[1] | ADI number (high byte), Read | |
| Data[2] | ADI element, Read | Reference to the element of the ADI mapped with Map_ADI_Read_Area for the specified SlotNr (See Data[10... 11]).<br>Range:<br> 1... 255 ADI element associated with the submodule<br> 0 Entire ADI is associated with the submodule |
| Data[3] | ADI number (low byte), Write | Reference to the ADI previously mapped with Map_ADI_Write_Area. |
| Data[4] | ADI number (high byte), Write | |
| Data[5] | ADI element, Write | Reference to the element of the ADI mapped with Map_ADI_Write_Area for the specified SlotNr (See Data[10... 11]).<br>Range:<br> 1... 255 ADI element associated with the submodule<br> 0 Entire ADI is associated with the submodule |
| Data[6] | API (low word, low byte) | Application Process Instance (API)<br>See also...<br> - "Application Process Instances (API)" on page 22 |
| Data[7] | API (low word, high byte) | |
| Data[8] | API (high word, low byte) | |
| Data[9] | API (high word, high byte) | |
| Data[10] | SlotNr (low byte) | Destination slot for submodule.<br>Range:<br> 0... 0x7FFF |
| Data[11] | SlotNr (high byte) | |
| Data[12] | SubSlotNr (low byte) | Destination subslot for submodule.<br>Range:<br> For API 0: 1... 0x8002<br> For API >0: 1... 0x7FFF |
| Data[13] | SubSlotNr (high byte) | |
| Data[14] | SubModIdent (low word, low byte) | Submodule identifier as stated in the GSD-file.<br>If the submodule is a safety module only high word is used. Low word is specified by the safety module. |
| Data[15] | SubModIdent (low word, high byte) | |
| Data[16] | SubModIdent (high word, low byte) | |
| Data[17] | SubModIdent (high word, high byte) | |

See also...

## Command Details: Plug_Submodule_Ext

### Category

Extended

### Details

| Command Code: | 19h |
| --- | --- |
| **Valid for:** | Object Instance |

### Description

This is an extended version of the **Plug_Submodule** command. This command may be called during start-up to specify the Real Identification. It additionally features the possibility to associate a submodule with several consecutive ADI elements. (The **Plug_Submodule** command only allows association with one ADI element or all ADI elements.)

This command can also be called during operation if there are changes to the Real Identification. A **Plug** or **Plug Wrong Submodule**-alarm is automatically sent to the master as a result of this action.

A submodule plugged with this command can hold IO data to the master, from the master or data in both directions. It is also possible to plug submodules which do not carry any data at all.

The Anybus CompactCom 40 PROFINET IRT supports up to 128 submodules in total.

> ⓘ *In case the slot number in the command is set to 0 (zero), the ADI number must also be 0 (zero), since slot 0 cannot hold any actual data.*
>
> *It is only permitted to issue this command if **API_Add** has been issued first.*
>
> *The **Interface**- and **Port** submodules have to be plugged in order to pass certification tests.*
>
> *The interface and port submodule can only be plugged during the SETUP-state. Any attempt to plug these submodules during runtime will result in error.*
>
> *It is not recommended to mix **Plug_Submodule** and **Plug_Submodule_Ext** commands.*
>
> *No more than one safety submodule can be plugged at any time. If, during NW_INIT, it is determined that no safety module is attached, and a safety submodule is plugged in state SETUP, an exception will be raised. If an attempt to plug a safety submodule is done during runtime, when no safety module is attached, the request will be rejected.*

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | Bit 0: Submodule type | 0: Normal I/O submodule<br>1: Safety submodule (no ADI mapping, set Data[0..7] to 0) |
| | Bit 1-7: (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | (reserved, set to zero) | (reserved for future use) |
| Data[0] | ADI number (low byte), Read | Reference to the ADI previously mapped with Map_ADI_ Read_Area. |
| Data[1] | ADI number (high byte), Read | |
| Data[2] | First ADI element, Read | Reference to the first ADI element associated with the submodule.<br>Range:<br>  0 to 255 - ADI element (0 is the first element) |
| Data[3] | Number of consecutive ADI elements, Read | Number of consecutive elements associated with the submodule.<br>Range:<br>1 to 255 (Number of elements) |
| Data[4] | ADI number (low byte), Write | Reference to the ADI previously mapped with Map_ADI_ Write_Area. |
| Data[5] | ADI number (high byte), Write | |
| Data[6] | ADI element, Write | Reference to the first ADI element associated with thesubmodule.<br>Range:<br>  0 to 255 - ADI element (0 is the first element) |
| Data[7] | Number of consecutive ADI elements, Write | Number of consecutive elements associated with the submodule.<br>Range:<br>1 to 255 (Number of elements) |
| Data[8] | API (low word, low byte) | Application Process Instance (API)<br>See also...<br>  - "Application Process Instances (API)" on page 22 |
| Data[9] | API (low word, high byte) | |
| Data[10] | API (high word, low byte) | |
| Data[11] | API (high word, high byte) | |
| Data[12] | SlotNr (low byte) | Destination slot for submodule.<br>Range:<br>  0... 0x7FFF |
| Data[13] | SlotNr (high byte) | |
| Data[14] | SubSlotNr (low byte) | Destination subslot for submodule.<br>Range:<br>  For API 0: 1... 0x8002<br>  For API >0: 1... 0x7FFF |
| Data[15] | SubSlotNr (high byte) | |
| Data[16] | SubModIdent (low word, low byte) | Submodule identifier as stated in the GSD file.<br>If the submodule is a safety module only high word is used. Low word is specified by the safety module. |
| Data[17] | SubModIdent (low word, high byte) | |
| Data[18] | SubModIdent (high word, low byte) | |
| Data[19] | SubModIdent (high word, high byte) | |

See also...

## Command Details: Pull_Module

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 12h |
| **Valid for:** | Object Instance |

### Description

This command removes a module from the configuration. Can be issued at any time. During runtime, it can be called in case there are changes to the Real Identification. The Anybus CompactCom 40 PROFINET IRT then automatically issues a **Pull** or **Pull Module** alarm to the master.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | | |
| Data[0] | API (low word, low byte) | Application Process Instance (API) |
| Data[1] | API (low word, high byte) | See also... |
| Data[2] | API (high word, low byte) | –  *Application Process Instances (API), p. 17* |
| Data[3] | API (high word, high byte) | |
| Data[4] | SlotNr (low byte) | Slot number of module. |
| Data[5] | SlotNr (high byte) | Range:<br>  0... 0x7FFF |

## Command Details: Pull_Submodule

**Category**

Extended

**Details**

| | |
|---|---|
| **Command Code:** | 13h |
| **Valid for:** | Object Instance |

**Description**

This command removes a submodule from the configuration and can be issued at any time. During runtime, it can be called in case there are changes to the Real Identification. The Anybus CompactCom 40 PROFINET IRT0 then automatically issues a **Pull** alarm to the master.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | | |
| Data[0] | API (low word, low byte) | Application Process Instance (API) |
| Data[1] | API (low word, high byte) | See also... |
| Data[2] | API (high word, low byte) | – *Application Process Instances (API), p. 17* |
| Data[3] | API (high word, high byte) | |
| Data[4] | SlotNr (low byte) | Slot number of submodule. |
| Data[5] | SlotNr (high byte) | Range:<br>  0... 0x7FFF |
| Data[6] | SubSlotNr (low byte) | Subslot number of submodule. |
| Data[7] | SubSlotNr (high byte) | Range:<br>  For API 0: 1... 0x8002<br>  For API >0: 1... 0x7FFF |

## Command Details: API_Add

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 14h |
| **Valid for:** | Object Instance |

### Description

By default, the module only supports API 0 (zero). If additional APIs are to be supported, or if the host application shall handle plugging/unplugging of modules and submodules, this command must be used to specify the API implementation. Note that if using this command, it is mandatory to declare API 0 (zero) prior to defining other APIs or plugging/unplugging modules/submodules. API numbers are assigned by (PROFIBUS & PROFINET International (PI)).

> ℹ️ This command may only be issued prior to setting the **Setup Complete**-attribute in the Anybus Object.
>
> This command clears the default Real Identification created by the Anybus module while mapping ADIs to Process Data. Therefore, issuing this command effectively makes it mandatory to specify the actual Real Identification by means of the **Plug_Module** and **Plug_Submodule**-commands.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | | |
| Data[0] | API (low word, low byte) | Application Process Instance (API) |
| Data[1] | API (low word, high byte) | See also... |
| Data[2] | API (high word, low byte) | – *Application Process Instances (API), p. 17* |
| Data[3] | API (high word, high byte) | |
| Data[4] | (reserved, set to zero) | (reserved) |
| Data[5] | | |
| Data[6] | | |
| Data[7] | | |

See also...

## Command Details: Appl_State_Ready

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 15h |
| **Valid for:** | Object Instance |

### Description

This command is only applicable if the host application implements support for **End_Of_Prm_Ind**, and signals to the module (and in turn the I/O Controller) that the host application is ready for data exchange.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |

See also...

*Application Process Instances (API), p. 17*

- End_Of_Prm_Ind, command details in *PROFINET IO Object (F6h), p. 118*

## Command Details: AR_Abort

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 16h |
| **Valid for:** | Object Instance |

### Description

This command indicates to the module that the current application relationship shall be aborted.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |

See also...

- *Application Process Instances (API), p. 17*

- AR_Check_Ind, command details in *PROFINET IO Object (F6h), p. 118*

- Expected_Ident_Ind, command details in *PROFINET IO Object (F6h), p. 118*

- Appl_State_Ready, command details in *PROFINET IO Object (F6h), p. 118*

## Command Details: IM_Options

### Category

Extended

### Details

| | |
|---|---|
| **Command Code:** | 18h |
| **Valid for:** | Object Instance |

### Description

During startup, this command can be called to specify if I&M data for Slot0 (DAP) and/or Slot > 0 should be forwarded transparently by the Anybus CompactCom module. Additionally, it provides a way for the application to specify the I&M0 Filter Data.

I&M0 Filter Data is composed of three blocks: I&M0 Carrier Data, Module Representative Data and Device Representative Data (see table below). A submodule can belong to several blocks.

| I&M0 Filter Data Contents | |
|---|---|
| **Content** | **Description** |
| I&M0 Carrier Data | List of all submodules being a carrier of discrete I&M data<br>Block is mandatory to support according to the PROFINET specification |
| Module Representative Data | List of all submodules acting as module representative<br>Block is optional to support according to the PROFINET specification |
| Device Representative Data | List of at least one submodule where I&M1, I&M2, and I&M3 can be written<br>Block is mandatory to support according to the PROFINET specification |

When transparent I&M data for slot > 0 is enabled, the application must store I&M data for modules located in slots > 0 to nonvolatile memory. In this case, all modules that carry discrete I&M data shall be included in the **I&M0 Carrier Data**. (The Anybus CompactCom 40 PROFINET IRT will include the DAP submodule (located in slot 0, subslot 1) in the **I&M0 Carrier Data** and **Device Representative Data**.)

When transparent I&M data for slot 0 is enabled, the application must store I&M data for slot 0 to nonvolatile memory. The DAP submodule is by default included in the **I&M0 Carrier Data** and **Device Representative Data** but may be removed from any of the blocks using the **IM_Options** command.

See also...

- *Flowchart — I&M Record Data Handling, p. 147*.

- Get Record, command details in *PROFINET IO Object (F6h), p. 118*

- Set_Record, command details in *PROFINET IO Object (F6h), p. 118*

- Command Details

  If the I&M0 Filter Data is of no interest, the Data Field is left out (command length = 0).

  The command may contain one or several I&M0 Filter Data entries. The maximum amount of entries depends on the application. For a 256 bytes message channel the maximum amount of entries is 51 (256 / 5 = 51). For a 1524 bytes message channel, the maximum amount of entries is 304 (1524 / 5 = 304).

  > ⚠ For submodules to be listed in the I&M0 Filter data, the command must be sent when the Real Identification has been determined. This means that for the "ADI Based RI" method, the command must be sent when the module has shifted to WAIT_PROCESS state. For the "Application specific RI" method the command can be sent in SETUP state but after the plugging of modules/ submodules is finished (Plug_Module/Plug_Submodule).
  >
  > The table below contains two entries as an example.

| Field | Contents | Comments | Example |
|---|---|---|---|
| CmdExt[0] | IM_Transparent | Bitmask for enabling transparent I&M data | 02h |
| CmdExt[1] | Reserved | See table below | |
| Data[0] | SlotNr (low byte) | Location of the submodule that shall be part of the I&M0 Filter Data | Entry #1 - module with discrete I/M data in slot 1, subslot 1 |
| Data[1] | SlotNr (high byte) | | |
| Data[2] | SubSlotNr (low byte) | | |
| Data[3] | SubSlotNr (high byte) | | |
| Data[4] | IM0_Filter_Data | Bitmask specifying which I&M0 Filter Data block(s) the submodule shall belong to<br>See table below | 01h |
| Data[5] | SlotNr (low byte) | | Entry #2 - module with discrete I&M data in slot 2, subslot 1 |
| Data[6] | SlotNr (high byte) | | |
| Data[7] | SubSlotNr (low byte) | | |
| Data[8] | SubSlotNr (high byte) | | |
| Data[9] | IM0_Filter_Data | | 01h |

**IM_Transparent**

| Bit | Value | Description |
|---|---|---|
| 0 | 0 | |
| | 1 | Transparent I&M data for SlotNr = 0 (DAP) |
| 1 | 0 | |
| | 1 | Transparent I&M data for SlotNr not equal to 0 |
| 2 - 7 | - | Reserved |

**IM0_Filter_Data**

| Bit | Value | Description |
|---|---|---|
| 0 | 0 | |
| | 1 | Submodule will be included in the I&M0 Carrier Data block |
| 1 | 0 | |
| | 1 | Submodule will be included in the Device Representative Data block |
| 2 | 0 | |
| | 1 | Submodule will be included in the Module Representative Data block |
| 3 - 7 | - | Reserved |

## Command Details: Ident_Change_Done

### Category

Extended

### Details

| Command Code: | 1Ah |
|---|---|
| **Valid for:** | Object Instance |

### Description

This command shall be sent to the module when the host application has finished its adaptations of the Real Identification.

It is optional to implement support for this command, except for that it must be issued if the host application previously has responded with **Block** to the command Expected_Ident_Ind (1Bh).

* Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |

* Response Details

    (No data)

## Command Details: Add_Safety_Module

**Category**

Extended

**Details**

| Command Code: | 17h |
|---|---|
| **Valid for:** | Object Instance |

**Description**

Usage of this command is conditional.

- If API_Add has been issued, ADD_Safety_Module must be issued. Any slot can be specified (1... 7FFFh).

- If API_Add has not been issued, ADD_Safety_Module is rejected. A safety module will be placed in slot 1.

The command must be called during start-up, before setup is complete. The Safety Module can be located in a valid slot (1...7FFFh). In addition to the slot number, the 16 most significant bits of the 32 bit module identification number for the Safety Module are specified. The 16 least significant bits are specified by the Safety Module itself.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved, set to zero) | (reserved for future use) |
| CmdExt[1] | | |
| Data[0] | SlotNr (low byte) | Number of the slot where to insert the Safety Module. Valid values are 1... 7FFFh. |
| Data[1] | SlotNr (high byte) | |
| Data[2] | ModIdent_High (low byte) | Module identifier as stated in the GSD file (highest 16 bits). |
| Data[3] | ModIdent_High (high byte) | |

- Response Details

  See" Object Specific Error Codes" below.

## Object Specific Error Codes

| Code | Meaning |
|------|---------|
| 01h | The ADI has not been mapped with command Map_ADI_Write_Area |
| 02h | The ADI has not been mapped with command Map_ADI_Read_Area |
| 03h | Element does not exist for the ADI |
| 04h | This ADI/element is already mapped |
| 05h | API 0 must be added first |
| 06h | API does not exist |
| 07h | Trying to add an API already present |
| 08h | There is no room for any more APIs |
| 09h | Module in slot 0 cannot have any IO data |
| 0Ah | Prior to plugging the requested module/submodule, slot 0 must be populated with a module and a submodule (in subslot 1) |
| 0Bh | Slot occupied |
| 0Ch | subslot occupied |
| 0Dh | No module inserted in the specified slot |
| 0Eh | No submodule inserted in the specified subslot |
| 0Fh | Slot number specified is out-of-range |
| 10h | subslot number specified is out-of-range |
| 11h | The AR handle provided is not valid |
| 12h | There is no application ready pending |
| 13h | Unknown error (PROFINET IO stack denied the request) |
| 14h | Max number of submodules have already been plugged |
| 15h | Safety module has not been plugged |
| 16h | ADI data type constraint |
| 17h | Safety Module already plugged |

## 10.9 Functional Safety Module Object (11h)

### Category

Extended

### Object Description

This object contains information provided by the Safety Module connected to the Anybus CompactCom module. Please consult the manual for the Safety Module used, for values of the attributes below.

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| | Error_Confirmation |
| | Set_IO_Config_String |
| | Get_Safety_Output_PDU |
| | Get_Safety_Input_PDU |
| **Instance:** | Get_Attribute |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "Functional Safety Module" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | 0001h |
| 4 | Highest instance no. | Get | UINT16 | 0001h |

## Instance Attributes (Instance #1)

| # | Name | Access | Data Type | Description |
|---|------|--------|-----------|-------------|
| 1 | State | Get | UINT8 | Current state of the Safety Module<br>Please consult the manual for the Safety Module used. |
| 2 | Vendor ID | Get | UINT16 | Identifies vendor of the Safety Module.<br>E.g. 0001h (HMS Industrial Networks)<br>Please consult the manual for the Safety Module used. |
| 3 | IO Channel ID | Get | UINT16 | Describes the IO Channels that the Safety Module is equipped with.<br>Please consult the manual for the Safety Module used. |
| 4 | Firmware version | Get | Struct of<br>UINT8 (Major)<br>UINT8 (Minor)<br>UINT8 (Build) | Safety Module firmware version.<br>Format: version "2.18.3" would be represented as: first byte = 0x02, second byte = 0x12, third byte = 0x03. |
| 5 | Serial number | Get | UINT32 | 32 bit number, assigned to the Safety Module at production.<br>Please consult the manual for the Safety Module used. |
| 6 | Output data | Get | Array of UINT8 | Current value of the Safety Module output data, i.e. data FROM the network<br>Note: This data is unsafe, since it is provided by the Anybus CompactCom module. |
| 7 | Input data | Get | Array of UINT8 | Current value of the Safety Module input data, i.e. data sent TO the network.<br>**Note**: This data is unsafe, since it is provided by the Anybus CompactCom module. |
| 8 | Error counters | Get | Struct of<br>UINT16 (ABCC DR)<br>UINT16 (ABCC SE)<br>UINT16 (SM DR)<br>UINT16 (SM SE) | Error counters (each counter stops counting at FFFFh)<br><br>ABCC DR: Responses (unexpected) from the Safety Module, discarded by the Anybus CompactCom module.<br>ABCC SE: Serial reception errors detected by the Anybus CompactCom module.<br>SM DR: Responses (unexpected) from the Anybus CompactCom module, discarded by the Safety Module.<br>SM SE: Serial reception errors detected by the Safety Module. |
| 9 | Event log | Get | Array of UINT8 | Latest Safety Module event information (if any) is logged to this attribute. Any older event information is erased when a new event is logged.<br>For evaluation by HMS support. |
| 10 | Exception information | Get | UINT8 | If the Exception Code in the Anybus object is set to "Safety communication error" (09h), additional exception information is presented here, see table below. |
| 11 | Bootloader version | Get | Struct of<br>UINT8 Major<br>UINT8 Minor | Safety Module bootloader version.<br>Format: version "2.12" would be represented as: first byte = 0x02, second byte = 0x0C |
| 12 | Vendor block safe uc1 | Get | Array of UINT8 | The Safety Module may supply additional vendor-specific data to the Anybus CompactCom. If such data is available it is presented in this attribute. |
| 13 | Vendor block safe uc2 | Get | Array of UINT8 | The Safety Module may supply additional vendor-specific data to the Anybus CompactCom. If such data is available it is presented in this attribute. |

**Exception Information**

If Exception Code 09h is set in the Anybus object, there is an error regarding the functional safety module in the application. Exception information is presented in instance attribute #10 according to this table:

| Value | Exception Information |
|-------|----------------------|
| 00h | No information |
| 01h | Baud rate not supported |
| 02h | No start message |
| 03h | Unexpected message length |
| 04h | Unexpected command in response |
| 05h | Unexpected error code |
| 06h | Safety application not found |
| 07h | Invalid safety application CRC |
| 08h | No flash access |
| 09h | Answer from wrong safety processor during boot loader communication |
| 0Ah | Boot loader timeout |
| 0Bh | Network specific parameter error |
| 0Ch | Invalid IO configuration string |
| 0Dh | Response differed between the safety microprocessors (e.g. different module types) |
| 0Eh | Incompatible module (e.g. supported network) |
| 0Fh | Max number of retransmissions performed (e.g. due to CRC errors) |
| 10h | Firmware file error |
| 11h | The cycle time value in attribute #4 in the Functional Safety Host Object can not be used with the current baud rate |
| 12h | Invalid SPDU input size in start-up telegram |
| 13h | Invalid SPDU output size in start-up telegram |
| 14h | Badly formatted input SPDU |
| 15h | Anybus to safety module initialization failure |

## Command Details: Error_Confirmation

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 10h |
| **Valid for:** | Object |

### Description

When the Safety Module has entered the Safe State, for any reason, it must receive an error confirmation before it can leave the Safe State. With this command it is possible to reset all safety channels of the safety which, for any reason, are in the Safe State at the same time. The application issues this command to the Anybus CompactCom module, when an error has been cleared by for example an operator. The Anybus CompactCom forwards the command to the Safety Module.

The channel Safe State can also be confirmed by the safety PLC or by the safety module.

With this command

- Command Details

    (no data)

- Response Details

    (no data)

## Command Details: Set_IO_Config_String

**Category**

Extended

**Details**

| Command Code | 11h |
|---|---|
| **Valid for:** | Object |

**Description**

This command is sent from the host application when there is a need to change the default configuration of the safety inputs and outputs. This string is used by networks where there are no other means (e.g. PLC or some other tool) to provide the configuration to the safety module. Consult the specification of the safety module for more information. The byte string passed is generated by HMS and need to be passed unmodified using this command.

Information about this string is located in the specification of the safety module to which the string shall be sent.

• Command Details

| Field | Contents |
|---|---|
| CmdExt[0] | (not used) |
| CmdExt[1] | |
| Data[0... n] | Data (byte string)<br>The data consists of an IO configuration string, where the data format depends on the safety network. |

• Response Details

   (no data)

## Command Details: Get_Safety_Output_PDU

### Category

Extended

### Details

| | |
| --- | --- |
| **Command Code** | 12h |
| **Valid for:** | Object |

### Description

This command can be issued by the application to get the complete safety output PDU sent by the PLC. The Anybus CompactCom 40 PROFINET IRT will respond with the complete safety PDU, that the application then has to interpret.

- Command Details

  (no data)

- Response Details

| Field | Contents |
| --- | --- |
| CmdExt[0] | (not used) |
| CmdExt[1] | |
| Data[0... n] | Safety PDU from PLC |

## Command Details: Get_Safety_Input_PDU

### Category

Extended

### Details

| | |
| --- | --- |
| **Command Code** | 13h |
| **Valid for:** | Object |

### Description

This command can be issued by the application to get the complete safety input PDU sent by the safety module. The Anybus CompactCom 40 PROFINET IRT will respond with the complete safety PDU, that the application then has to interpret.

- Command Details

  (no data)

- Response Details

| Field | Contents |
| --- | --- |
| CmdExt[0] | (not used) |
| CmdExt[1] | |
| Data[0... n] | Safety PDU from safety module |

## Object Specific Error Codes

| Error Code | Description | Comments |
|---|---|---|
| 01h | The safety module rejected a message. | Error code sent by safety module is found in MsgData[2] and MsgData[3]. |
| 02h | Message response from the safety module has incorrect format (for example, wrong length). | - |

# 11  Host Application Objects

## 11.1  General Information

This chapter specifies the host application object implementation in the module. The objects listed here may be implemented within the host application firmware to expand the PROFINET implementation.

Standard Objects

- *Energy Measurement Object (E4h), p. 91*

- *Functional Safety Object (E8h), p. 106*

- " Application File System Object (EAh)" (see Anybus CompactCom 40 Software Design Guide)

- " Modular Device Object (ECh)" (see Anybus CompactCom 40 Software Design Guide)

- *Sync Object (EEh), p. 108*

- *Energy Control Object (F0h), p. 112*

- "Application Data Object (FEh)" (see Anybus CompactCom 40 Software Design Guide)

- " Application Object (FFh)" (see Anybus CompactCom 40 Software Design Guide)

Network Specific Objects:

- *Asset Management Object (E5h), p. 102*

- *PROFINET IO Object (F6h), p. 118*

- *Ethernet Host Object (F9h), p. 138*

## 11.2        Energy Measurement Object (E4h)

### Category

Extended

### Object Description

This object defines a standardized way of reporting different types of measurement values (current, voltage, power, energy etc.) from a measuring device.

Each instance, logical device, represents a group of measurement values for one separate measurement device. Physically, measuring devices might be connected to a higher level network directly, or indirectly via a coupling device. To enable host applications to implement either a direct or an indirect measuring device, the Anybus CompactCom object model implies a 1:1 relationship between measuring device (a.k.a. logical device) and instance number, where each instance attribute represent a group of measurement values (see list of instance attributes below). The host application does not have to support all instance attributes listed.The command Get_Attribute_Measurement_List is used towards the host application, to read out which measurement values are supported by a device.

The table below shows an example of how the instances of the Anybus CompactCom object model maps to PROFIenergy. PROFIenergy uses the Object number to address a logical device, while the Measurement ID refers to the actual measurement value.

| Instance number | Attribute number | PROFINET | | Description |
|---|---|---|---|---|
| (logical device) | (measurement value) | Obj nr | Measurement ID | |
| 1 | 1 | 0 | 1,2,3,31 | Logical device #1 |
| | 4 | | 4,5,6,32 | #1-Voltage PH-N |
| | 10 | | 7,8,9,33 | #4-Voltage PH-PH #10-Current |
| 2 | 1 | 1 | 1,2,3,31 | Logical device #2 |
| | 4 | | 4,5,6,32 | #1-Voltage PH-N |
| | 10 | | 7,8,9,33 | #4-Voltage PH-PH #10-Current |
| 3 | 13 | 2 | 10,11,12,36 | Logical device #3 |
| | 16 | | 13,14,15,34 | #13-Apparent Power |
| | 19 | | 16,17,18,35 | #16-Active Power #19-Reactive Power |

See also...

• *PROFIenergy Profile, p. 31*

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |
| | Get_Attribute_Measurement_List |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "Energy Measurement" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | One instance per supported measurement device |
| 4 | Highest instance no. | Get | UINT16 | Max 8, depending of how many instances have been implemented |

### Instance Attributes (Instance #1 - #8)

Each instance contains a set of attributes, where each attribute represents a group of measurement values. Each value can be associated with a timestamp, indicating when the recording took place. If the timestamp is set to 0, the value is not associated with a timestamp. The timestamp is given as number of milliseconds elapsed since 00.00.00 Coordinated Universal Time (UTC), Thursday, January 1st 1970, not counting leap seconds. Example:

```
2017-02-01 12:16:30 = 1 485 951 390 000 ms
```

It is advised that the host application store attributes #29 - #31 in non-volatile memory.

- Abbreviations used:
  - TS – Timestamp
  - Avg – Average
  - PH – Phase (L1–Line 1, L2 –Line 2, L3 – Line 3)
  - N – Neutral, G – Ground
- "-" indicates the measurement value has no representation on PROFIenergy.

The attributes can be divided in different categories, depending on the type of the variable they represent. See the table immediately following this one, for more information.

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|---|---|---|---|---|---|
| 1 | Get | VoltagePH-N | UINT64 FLOAT UINT32 | 0: L1-N TS 1: L1-N Value 2: Reserved | 1 |
| | | | UINT64 FLOAT UINT32 | 3: L2-N TS 4: L2-N Value 5: Reserved | 2 |
| | | | UINT64 FLOAT UINT32 | 6: L3-N TS 7: L3-N Value 8: Reserved | 3 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-N TS 10: Avg L-N Value 11: Reserved | 31 |
| 2 | Get | Min VoltagePH-N | UINT64 FLOAT UINT32 | 0: L1-N TS 1: L1-N Value 2: Reserved | 70 |
| | | | UINT64 FLOAT UINT32 | 3: L2-N TS 4: L2-N Value 5: Reserved | 71 |
| | | | UINT64 FLOAT UINT32 | 6: L3-N TS 7: L3-N Value 8: Reserved | 72 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-N TS 10: Avg L-N Value 11: Reserved | 92 |
| 3 | Get | Max VoltagePH-N | UINT64 FLOAT UINT32 | 0: L1-N TS 1: L1-N Value 2: Reserved | 40 |
| | | | UINT64 FLOAT UINT32 | 3: L2-N TS 4: L2-N Value 5: Reserved | 41 |
| | | | UINT64 FLOAT UINT32 | 6: L3-N TS 7: L3-N Value 8: Reserved | 42 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-N TS 10: Avg L-N Value 11: Reserved | 62 |
| 4 | Get | VoltagePH-PH | UINT64 FLOAT UINT32 | 0: L1-L2 TS 1: L1-L2 Value 2: Reserved | 4 |
| | | | UINT64 FLOAT UINT32 | 3: L2-L3 TS 4: L2-L3 Value 5: Reserved | 5 |
| | | | UINT64 FLOAT UINT32 | 6: L3-L1 TS 7: L3-L1 Value 8: Reserved | 6 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-L TS 10: Avg L-L Value 11: Reserved | 32 |

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|---|---|---|---|---|---|
| 5 | Get | Min VoltagePH-PH | UINT64 FLOAT UINT32 | 0: L1-L2 TS 1: L1-L2 Value 2: Reserved | 73 |
| | | | UINT64 FLOAT UINT32 | 3: L2-L3 TS 4: L2-L3 Value 5: Reserved | 74 |
| | | | UINT64 FLOAT UINT32 | 6: L3-L1 TS 7: L3-L1 Value 8: Reserved | 75 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-L TS 10: Avg L-L Value 11: Reserved | 93 |
| 6 | Get | Max VoltagePH-PH | UINT64 FLOAT UINT32 | 0: L1-L2 TS 1: L1-L2 Value 2: Reserved | 43 |
| | | | UINT64 FLOAT UINT32 | 3: L2-L3 TS 4: L2-L3 Value 5: Reserved | 44 |
| | | | UINT64 FLOAT UINT32 | 6: L3-L1 TS 7: L3-L1 Value 8: Reserved | 45 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-L TS 10: Avg L-L Value 11: Reserved | 63 |
| 7 | Get | Voltage PH-G | UINT64 FLOAT UINT32 | 0: L1-G TS 1: L1-G Value 2: Reserved | - |
| | | | UINT64 FLOAT UINT32 | 3: L2-G TS 4: L2-G Value 5: Reserved | |
| | | | UINT64 FLOAT UINT32 | 6: L3-G TS 7: L3-G Value 8: Reserved | |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-G TS 10: Avg L-G Value 11: Reserved | |
| 8 | Get | Min VoltagePH-G | UINT64 FLOAT UINT32 | 0: L1-G TS 1: L1-G Value 2: Reserved | - |
| | | | UINT64 FLOAT UINT32 | 3: L2-G TS 4: L2-G Value 5: Reserved | |
| | | | UINT64 FLOAT UINT32 | 6: L3-G TS 7: L3-G Value 8: Reserved | |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-G TS 10: Avg L-G Value 11: Reserved | |
| 9 | Get | Max VoltagePH-G | UINT64 FLOAT UINT32 | 0: L1-G TS 1: L1-G Value 2: Reserved | - |
| | | | UINT64 FLOAT UINT32 | 3: L2-G TS 4: L2-G Value 5: Reserved | |
| | | | UINT64 FLOAT UINT32 | 6: L3-G TS 7: L3-G Value 8: Reserved | |
| | | | UINT64 FLOAT UINT32 | 9: Avg L-G TS 10: Avg L-G Value 11: Reserved | |

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|---|---|---|---|---|---|
| 10 | Get | Current | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 7 |
| | | | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 8 |
| | | | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 9 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L TS 10: Avg L Value 11: Reserved | 33 |
| | | | UINT64 FLOAT UINT32 | 12: N TS 13: N Value 14: Reserved | - |
| 11 | Get | Min Current | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 76 |
| | | | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 77 |
| | | | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 78 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L TS 10: Avg L Value 11: Reserved | 94 |
| | | | UINT64 FLOAT UINT32 | 12: N TS 13: N Value 14: Reserved | - |
| 12 | Get | Max Current | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 46 |
| | | | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 47 |
| | | | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 48 |
| | | | UINT64 FLOAT UINT32 | 9: Avg L TS 10: Avg L Value 11: Reserved | 64 |
| | | | UINT64 FLOAT UINT32 | 12: N TS 13: N Value 14: Reserved | - |
| 13 | Get | Apparent Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 10 |
| | | | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 11 |
| | | | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 12 |
| | | | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 36 |

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|---|---|---|---|---|---|
| 14 | Get | Min Apparent Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 79 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 80 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 81 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 97 |
| 15 | Get | Max Apparent Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 49 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 50 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 51 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 67 |
| 16 | Get | Active Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 13 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 14 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 15 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 34 |
| 17 | Get | Min Active Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 82 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 83 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 84 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 95 |
| 18 | Get | Max Active Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 52 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 53 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 54 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 65 |

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|----------|-------------|------|-----------|----------------------------|----------------------------|
| 19 | Get | Reactive Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 16 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 17 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 18 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 35 |
| 20 | Get | Min Reactive Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 85 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 86 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 87 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 96 |
| 21 | Get | Max Reactive Power | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 55 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 56 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 57 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 66 |
| 22 | Get | Power factor | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 19 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 20 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 21 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 37 |
| 23 | Get | Min Power factor | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 88 |
|  |  |  | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 89 |
|  |  |  | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 90 |
|  |  |  | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 98 |

| Attr ID# | Access Rule | Name | Data Type | Element number:description | PROFIenergy Measurement ID |
|---|---|---|---|---|---|
| 24 | Get | Max Power factor | UINT64 FLOAT UINT32 | 0: L1 TS 1: L1 Value 2: Reserved | 58 |
| | | | UINT64 FLOAT UINT32 | 3: L2 TS 4: L2 Value 5: Reserved | 59 |
| | | | UINT64 FLOAT UINT32 | 6: L3 TS 7: L3 Value 8: Reserved | 60 |
| | | | UINT64 FLOAT UINT32 | 9: Total TS 10: Total Value 11: Reserved | 68 |
| 25 | Get | Frequency | UINT64 FLOAT UINT32 | 0: Line frequency TS 1: Line frequency Value 2: Reserved | 30 |
| 26 | Get | Min Frequency | UINT64 FLOAT UINT32 | 0: Line frequency TS 1: Line frequency Value 2: Reserved | 91 |
| 27 | Get | Max Frequency | UINT64 FLOAT UINT32 | 0: Line frequency TS 1: Line frequency Value 2: Reserved | 61 |
| 28 | Get | Field rotation | UINT64 FLOAT UINT32 | 0: Field rotation TS 1: Field rotation Value 2: Reserved | - |
| 29 | Get | Total Active energy (Sum = Consumed – Generated) | UINT64 FLOAT UINT32 | 0: Consumed TS 1: Consumed Value 2: Reserved | 200 |
| | | | UINT64 FLOAT UINT32 | 3: Generated TS 4: Generated Value 5: Reserved | 201 |
| | | | UINT64 FLOAT UINT32 | 6: Sum TS 7: Sum Value 8: Reserved | 205 |
| 30 | Get | Total Reactive energy(Sum = Consumed – Generated) | UINT64 FLOAT UINT32 | 0: Consumed TS 1: Consumed Value 2: Reserved | 202 |
| | | | UINT64 FLOAT UINT32 | 3: Generated TS 4: Generated Value 5: Reserved | 203 |
| | | | UINT64 FLOAT UINT32 | 6: Sum TS 7: Sum Value 8: Reserved | 206 |
| 31 | Get | Total Apparent energy | UINT64 FLOAT UINT32 | 0: Consumed TS 1: Consumed Value 2: Reserved | - |
| | | | UINT64 FLOAT UINT32 | 3: Generated TS 4: Generated Value 5: Reserved | - |
| | | | UINT64 FLOAT UINT32 | 6: Sum TS 7: Sum Value 8: Reserved | 204 |

The table below describes how the generic values for the different variable categories are represented on PROFINET.

| Attr # | Variable category | Unit | Comments |
|---|---|---|---|
| 1-9 | Voltage | V | RMS value<br>Absolute value |
| 10-12 | Current | A | RMS value<br>Positive values indicate consumed current; negative values indicate generated current. |
| 13-15 | Apparent Power | VA | Positive values indicate consumed apparent power; negative values indicate generated apparent power. |
| 16-18 | Active Power | W | Positive values indicate consumed active power; negative values indicate generated active power. |
| 19-21 | Reactive Power | VAR | Positive values indicates inductive reactive power; negative values indicates capacitive reactive power. |
| 22-24 | Power factor | | Range:-1.0 to 1.0. Positive values indicate leading power factor; negative values indicate lagging power factor. |
| 25-27 | Frequency | Hz | Absolute value |
| 28 | Field Rotation | | 0: no rotation recognized (not supported)<br>1: Clock-wise<br>-1: Counter clock-wise |
| 29 | Total Active energy | Wh | For the Import- and Export variables an absolute value is used.For the Sum variable a positive value indicates consumed energy; a negative value indicates generated energy. |
| 30 | Total Reactive energy | VARh | For the Consumed and Generated variables an absolute value is used. For the Sum variable a positive value indicates consumed reactive energy (inductive load); a negative value indicates generated reactive energy (capacitive load). |
| 31 | Total Apparent energy | VAh | Absolute value |

## Command Details: Get_Attribute_Measurement_List

**Details**

| | |
|---|---|
| **Command Code** | 10h |
| **Valid for:** | Instance |

**Description**

This command is used to gather all supported attribute elements of an instance. If several instances are supported, the command will be sent numerous times (implied by object attribute Highest instance number). Based on the response(s) the Anybus CompactCom will provide information to the network indicating which measurement values that are supported by a device.

The command response shall contain all attributes up to the last supported attribute number. For unsupported attributes within this range the bit field shall be set to zero.

The table below shows an example of the expected response for a (multiple phase) device supporting several measurement values for attributes #1, #4 and #10.

| Data[0-1] | 0x000F | Attribute #1 - measurement values 1-4 supported (i.e. L1-L2-L3-Avg) |
|---|---|---|
| Data[2-3] | 0x0000 | Attribute #2 - attribute not supported |
| Data[4-5] | 0x0000 | Attribute #3 – attribute not supported |
| Data[6-7] | 0x000F | Attribute #4 - measurement values 1-4 supported (i.e. L1-L2-L3-Avg) |
| Data[8-9] | 0x0000 | Attribute #5 – attribute not supported |
| Data[10-11] | 0x0000 | Attribute #6 – attribute not supported |
| Data[12-13] | 0x0000 | Attribute #7 – attribute not supported |
| Data[14-15] | 0x0000 | Attribute #8 – attribute not supported |
| Data[16-17] | 0x0000 | Attribute #9 – attribute not supported |
| Data[18-19] | 0x000F | Attribute #10 – measurement values 1-4 supported (i.e. L1-L2-L3-Avg) |

The table below shows an example of the expected response for a (single phase) device supporting one measurement value for attributes #1, #4 and #10.

| Data[0-1] | 0x0001 | Attribute #1 - measurement value 1 supported (i.e. L1) |
|---|---|---|
| Data[2-3] | 0x0000 | Attribute #2 - attribute not supported |
| Data[4-5] | 0x0000 | Attribute #3 – attribute not supported |
| Data[6-7] | 0x0001 | Attribute #4 - measurement value 1 supported (i.e. L1) |
| Data[8-9] | 0x0000 | Attribute #5 – attribute not supported |
| Data[10-11] | 0x0000 | Attribute #6 – attribute not supported |
| Data[12-13] | 0x0000 | Attribute #7 – attribute not supported |
| Data[14-15] | 0x0000 | Attribute #8 – attribute not supported |
| Data[16-17] | 0x0000 | Attribute #9 – attribute not supported |
| Data[18-19] | 0x0001 | Attribute #10 – measurement values 1 supported (i.e. L1) |

- Command Details

| Field | Contents |
|---|---|
| CmdExt[0 - 1] | Reserved, set to 00h |

- Response Details

| Field | Contents | Description |
|---|---|---|
| Data[0 - 1] | Attribute #1 | Data type: BITS16 |
| Data[2 - 3] | Attribute #2 | Bit 0 – First element<br>Bit 1 – Second element |
| Data[...] | ... | … |
| Data[(N*2-2) – (N*2-1)] | Attribute #N (last supported attribute number) | Bit 15 – Last element<br>Bit = 0: element not supported<br>Bit = 1: element supported |

## 11.3    Asset Management Object (E5h)

**Category**

Extended

**Object Description**

This object implements asset management information, i.e. identification information about one or several non PROFINET components of a device. Each component is represented by one object instance. The application collects identification information and assembles it in the instance attributes for each object instance.

**Supported Commands**

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |

**Object Attributes (Instance #0)**

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "Asset management" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | Max 32. Depends on number of non PROFINET components. |
| 4 | Highest instance no. | Get | UINT16 | Max 32. Depends on number of non PROFINET components. |

## Instance Attributes (Instance #1 - #32)

Asset management data for a non PROFINET component is represented by one of three different formats depending on the complexity of the component. For a complex component based on both hardware and software the format AM_Full information comply. For hardware-only components the format AM_HardwareOnly complies, and for software-only components the AM_FirmwareOnly format is used.

Which format to use for a specific component is selected by attribute AM_InfoType.

Both AM_HardwareOnly and AM_FirmwareOnly are subsets of AM_Full. If AM_HardwareOnly is selected the attributes AM_SoftwareRevision and IM_Software_Revision are of no interest and thus not read by the Anybus CompactCom. In the same manner, if AM_FirmwareOnly is selected the attributes AM_HardwareRevision and IM_Hardware_Revision are of no interest and thus not read by the Anybus CompactCom.

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| 1 | AM info Type | Get | UINT8 | - | Asset management data representation<br><br>0:       AM_FullInfo<br><br>1:       AM_HardwareOnlyInfo; attributes AM Software Revision and IM Software Revision are irrelevant<br><br>2:       AM_FirmwareOnlyInfo; attributes AM Hardware Revision and IM Hardware Revision are irrelevant |
| 2 | IM Unique Identifier | Get | Array of UINT8 | 0 (unused) | 128 bit UUID. Manufacturer created unique defined identifier. Used as source identification if issuing an event using reporting system or delivering asset management information<br>ISO/IEC 9834-8:2014 specifies the format and generation rules that enable users to produce 128-bit unique identifiers.<br>Number of elements in array: 16. |
| 3 | Location Type | Get | UINT8 | - | Asset management includes two concepts how to locate an asset. See section on AM_Location below for more information.<br><br>1:       Twelve level tree format (LT)<br><br>2:       Slot- and SubslotNumber format (SS) |
| 4 | Location LT | Get | Array of UINT16 | - | Mandatory if Location Type is "1" (LT). If not present any request will increase the error counter and the instance will not be reported on the network. See section on AM_Location below for usage. Maximum number of elements in array: 12. |
| 5 | Location SS | Get | struct of UINT16 BeginSlot UINT16 BeginSubslot UINT16 EndSlot UINT16 EndSubslot | - | Mandatory if Location Type is "2" (´SS). If not present any request will increase the error counter and the instance will not be reported on the network. See section on AM_Location below for usage. Maximum number of elements in array: 12. |
| 6 | IM Annotation | Get | Array of CHAR | 64 blanks | Manufacturer defined description, The value shall be set to a manufacturer specific identifier, and can be used for the name of the component,<br>Maximum number of elements in array: 64. |
| 7 | IM Order ID | Get | Array of CHAR | 64 blanks | Manufacturer defined order id<br>Maximum number of elements in array: 64. |
| 8 | IM Serial Number | Get | Array of CHAR | 16 blanks | Manufacturer defined serial number<br>Maximum number of elements in array: 16. |
| 9 | AM Device Identification | Get | Array of UINT16 | - | The content of this attribute can be used for coding of device, vendor and organization identification. See details below.<br>Number of elements in array: 4. |
| 10 | AM Type Identification | Get | UINT16 | 0 | 0000h    Unspecified<br><br>0001h    PLC |

| # | Name | Access | Data Type | Default Value | Comment | |
|---|------|--------|-----------|---------------|---------|---|
| | | | | | 0002h | PC based station |
| | | | | | 0003h | IO Module or IO Submodule |
| | | | | | 0004h | Communication Module or Communication Submodule |
| | | | | | 0005h | Interface Module or Interface Submodule |
| | | | | | 0006h | Active Network Infrastructure Component |
| | | | | | 0007h | Media converter |
| | | | | | 0100h - 7FFFh | Shall be defined by the manufacturer of the reporting entity |
| | | | | | Other | (reserved) |
| 11 | AM Software Revision | Get | Array of CHAR | 64 blanks | Manufacturer defined software revision.<br>Maximum number of elements in array: 64.<br>Example: "V1.3.0"<br>If attribute #12, IM Software Revision, is supported, this attribute should not be implemented. | |
| 12 | IM Software Revision | Get | struct of CHAR 3*UINT8 | V0.0.0 | Manufacturer defined software revision. | |
| 13 | AM Hardware Revision | Get | Array of CHAR | 64 blanks | Manufacturer defined hardware revision<br>Maximum number of elements in array: 64<br>If attribute #14, IM Hardware Revision, is supported, then this attribute should not be implemented. | |
| 14 | IM Hardware Revision | Get | UINT16 | 0 | Manufacturer defined hardware revision<br>0000h-FFFFh<br>(FFFFh indicates availability of profile specific information) | |

## AM_Location

Asset management include two concepts of how to locate an asset; "Twelve level tree format" and "Slot- and Subslot number format".

### Location Concept: Slot- and Subslot Number

In case of a remote IO device the location of a component (e.g. rack module) is clearly defined by the slot of the IO device. Other components may span several slots, such as a backplane, or several subslots, such as a terminal block. This means each asset can be located by "BeginSlotNumber/BeginSubslotNumber" and "EndSlotNumber/EndSubslotNumber".

For this concept the following parameters have to be considered:

| Array index | Description |
|---|---|
| 0 | BeginSlotNumber |
| 1 | BeginSubslotNumber |
| 2 | EndSlotNumber |
| 3 | EndSubslotNumber |

> **ⓘ** *It is important that the defined slot-/subslot numbers (i.e. BeginSlotNumber, EndSlotNumber, BeginSubslotNumber and EndSubslotNumber) are present in the Real Identification. This is checked at conformance testing.*

### Location Concept: Twelve Level Tree Format

In case of a remote IO device the location of a component is clearly defined by the level information. The Twelve level tree format may be used for modular devices with hierarchical device structure. This means each asset can be located by a level number 0-11 (e.g. x.y.z level notation). If LT format is used a value must be provided at least for level 0. If a shorter array is supplied value 03FFh is implied for the missing levels.

For this concept the following parameters have to be considered:

| Array index | Description | Value |
|---|---|---|
| 0 | Level 0 | 0000h - 03FEh: Address information to identify a reported node |
| 1 | Level 1 | 03FFh: Level not used |
| 2 | Level 2 | 0400h-0xFFFFh: Reserved |
| 3 | Level 3 | |
| 4 | Level 4 | |
| 5 | Level 5 | |
| 6 | Level 6 | |
| 7 | Level 7 | |
| 8 | Level 8 | |
| 9 | Level 9 | |
| 10 | Level 10 | |
| 11 | Level 11 | |

## AM_DeviceIdentification

| Array index | Description |
|---|---|
| 0 | Organization (0=PI)<br>0: Must be used<br>1–FFFFh: reserved |
| 1 | Vendor ID |
| 2 | Device ID |
| 3 | Device Sub ID:<br>0: Must be used<br>1–FFFFh: reserved |

## 11.4        Functional Safety Object (E8h)

**Category**

Extended

**Object Description**

> **!**     Do not implement this object if a safety module is not used.

This object specifies the safety settings of the application. It is mandatory if Functional Safety is to be supported and a Safety Module is connected to the Anybus CompactCom module.

**Supported Commands**

| **Object:** | Get_Attribute |
|---|---|
| **Instance:** | Get_Attribute |

**Object Attributes (Instance #0)**

| # | Name | Access | Data Type | Value |
|---|---|---|---|---|
| 1 | Name | Get | Array of CHAR | "Functional Safety" |
| 2 | Revision | Get | UINT8 | 01h |
| 3 | Number of instances | Get | UINT16 | 0001h |
| 4 | Highest instance no. | Get | UINT16 | 0001h |

**Instance Attributes (Instance #1)**

| # | Name | Access | Data Type | Default Value | Comment |
|---|---|---|---|---|---|
| 1 | Safety enabled | Get | BOOL | - | When TRUE, enables communication with the Safety Module. **Note**: If functional safety is not supported, this attribute must be set to FALSE. |
| 2 | Baud Rate | Get | UINT32 | 1020 kbit/s | This attribute sets the baud rate of the communication in bits/s between the Anybus CompactCom and the Safety Module. Valid values: <br>• 625 kbit/s <br>• 1000 kbit/s <br>• 1020 kbit/s (default) <br>Any other value set to this attribute, will cause the module to enter the EXCEPTION state. The attribute is optional. If not implemented, the default value will be used. **Note**: The host application shall never implement this attribute when using the IXXAT Safe T100. |
| 3 | (reserved) | | | | |

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| 4 | Cycle Time | Get | UINT8 | - | Communication cycle time between the Anybus and the Safety module in milliseconds. **Note**: The host application shall never implement this attribute when using the IXXAT Safe T100. Valid values: <br><br>• 2 ms <br><br>• 4 ms <br><br>• 8 ms <br><br>• 16 ms <br><br>If another value is set in this attribute the Anybus will enter Exception state. Optional attribute; If not implemented the minimum cycle time for the chosen baud rate will be used: <br><br>• 2 ms for 1020 kbit/s <br><br>• 2 ms for 1000 kbit/s <br><br>• 4 ms for 625 kbit/s <br><br>The Anybus CompactCom validates the cycle time according to the minimum values above. If e.g. baud rate is 625 kbit/s and the cycle time is set to 2 ms the Anybus CompactCom will enter the EXCEPTION state. |
| 5 | FW upgrade in progress | Set | BOOL | False | Indicates if the Anybus CompactCom is upgrading the connected Safety module firmware. This means that the Anybus CompactCom will stay in the NW_ INIT state longer than normal. |

## 11.5 Sync Object (EEh)

### Category

Extended

### Object Description

This object contains the host object sync settings.

The implementation of this object is optional. If it is not implemented the Anybus CompactCom 40 PROFINET IRT can not be used for a synchronous application.

If there is any problem with the configuration of the sync functionality as a whole, the application must indicate this in the application status register. The Anybus CompactCom will then change states and inform the problem to the PROFINET IO Controller, see *Application Status Register, p. 144*.

See also ...

*   Anybus CompactCom 40 Software Design Guide, "Sync"

*   Anybus CompactCom 40 Software Design Guide, "Sync Object"

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |
| | Set_Attribute |

### Object Attributes (Instance #0)

(Consult the general *Anybus CompactCom 40 Software Design Guide* for further information.)

## Anybus CompactCom Sync and PROFINET Isochronous Mode

The sync functionality is described differently in the PROFINET network specification than how it is described in the specification for Anybus CompactCom in general. See the Anybus CompactCom 40 Software Design Guide for a detailed description of the Anybus CompactCom sync functionality.

This section describes the correlation between the specifications

The figure below shows a timing diagram for PROFINET isochronous mode.
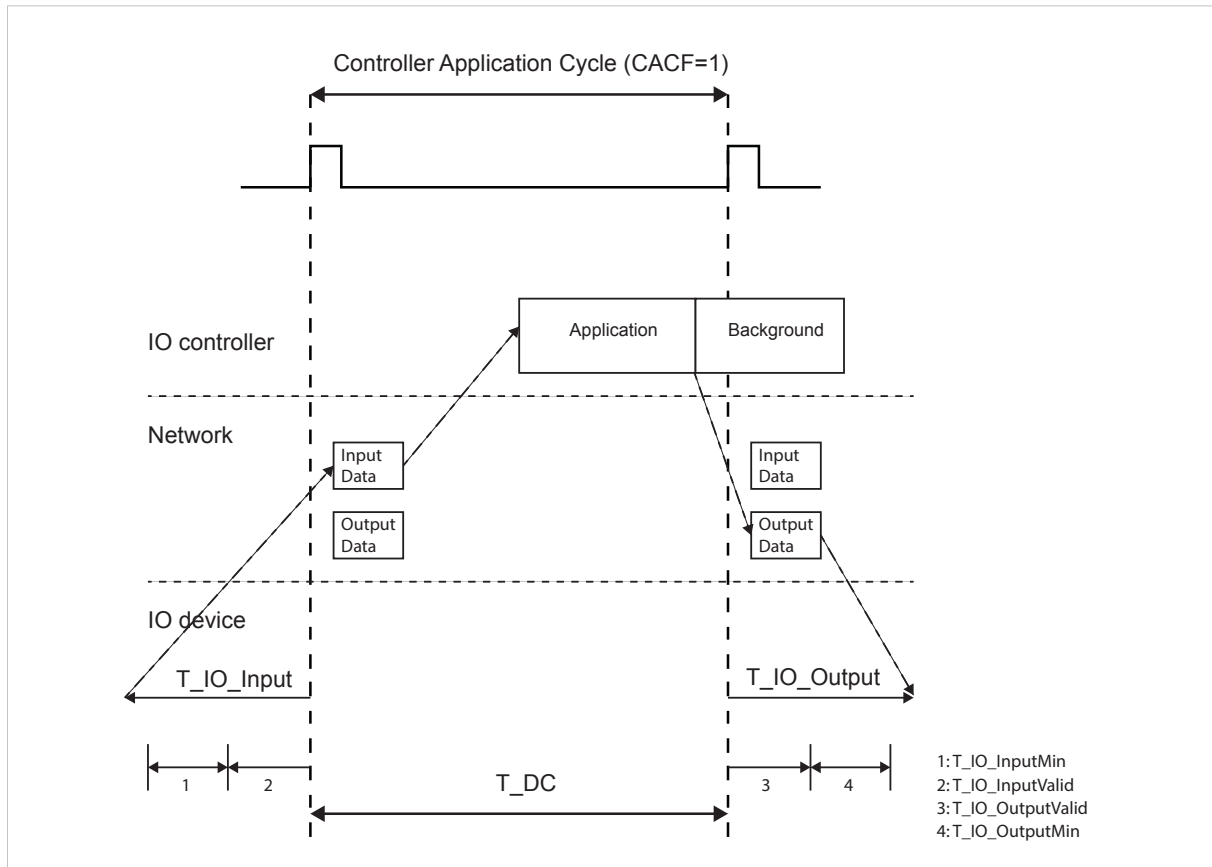


**Fig. 9**

### T_IO_OutputMin

T_IO_OutputMin consists of two delays:

- The delay added by the Anybus CompactCom. This is the time from when the message is available on the network until it is available to the application. This delay is 0 in the current implementation.

- The delay added by the application. This is the time it takes for the application from when it is notified that new process data has arrived, to when the process data is copied and the output is valid. This delay must be measured by the application designer. The measured value shall be written to Attribute #4 (Output Processing).

The value of T_IO_OutputMin must be entered in the GSD file for every submodule supporting synchronous operation. See example GSD entries below.

**T_IO_InputMin**

T_IO_InputMin consists of two delays:

- The delay added by the Anybus CompactCom. This is the time from when the message is available in the device until it is available on the network. This delay is 12 µs in the current implementation.

- The delay added by the application. This is the time it takes for the application from when inputs are captured, to when the input data is available to the Anybus CompactCom. This delay must be measured by the application designer. The measured value shall be written to Attribute #5 (Input Processing).

The value of T_IO_InputMin must be entered in the GSD file for every submodule supporting synchronous operation. See example GSD entries below.

**T_IO_Output**

T_IO_Output is equivalent to attribute #2 (Output Valid). The value is configured from the IO controller.

**T_IO_Input**

T_IO_Input is the configured Cycle Time (T_DC) minus the Input Capture time (Attribute #3). The value is configured from the IO controller.

**T_DC**

T_DC = Cycle Time (Attribute #1)

## GSDML Entries

The following must be added to the GSDML file for modules supporting isochronous operation:

```
IsochroneModeInRT_Classes="RT_CLASS_3" in the <InterfacesubmoduleItem>
<IsochroneMode T_DC_Base="8" T_DC_Min="1" T_DC_Max="16" T_IO_Base="1000"
   T_IO_OutputMin="X" T_IO_InputMin="Y"/>
```

## Instance Attributes (Instance #1)

| # | Name | Access | Corresponding term for PROFINET | Data Type | Value |
|---|------|--------|---------------------------------|-----------|-------|
| 1 | Cycle time | Get/Set | T_DC | UINT32 | Application cycle time in nanoseconds |
| 2 | Output valid | Get/Set | T_IO_Output | UINT32 | Output valid point relative to SYNC events, in nanoseconds<br>Default value: 0 |
| 3 | Input capture | Get/Set | T_DC - T_IO_Input | UINT32 | Input capture point relative to SYNC events, in nanoseconds<br>Default value: 0 |
| 4 | Output processing | Get | T_IO_OutputMin | UINT32 | Minimum required time, in nanoseconds, between RDPDI interrupt and "Output valid" |
| 5 | Input processing | Get | T_IO_InputMin - 12 µs | UINT32 | Maximum required time, in nanoseconds, from "Input capture" until write process data has been completely written to the Anybus CompactCom module |
| 6 | Min cycle time | Get | T_DC_Min | UINT32 | Minimum cycle time supported by the application, in nanoseconds |
| 7 | Sync mode | Get/Set | - | UINT16 | This attribute is used to select synchronization mode. It enumerates the bits in attribute 8<br>0: Nonsynchronous mode. (Default value if nonsynchronous mode is supported)<br>1: Synchronous mode<br>2 - 65535: Reserved. Any attempt to set sync mode to an unsupported value shall generate an error response |
| 8 | Supported sync modes | Get | - | UINT16 | A list of the synchronization modes the application supports.<br>Each bit corresponds to a mode in attribute 7<br>Bit 0: 1 = Nonsynchronous mode supported<br>Bit 1: 1 = Synchronous mode supported (Required for synchronous application)<br>Bit 2 - 15: Reserved (0) |
| 9 | Control Task cycle factor | Get/Set | CACF (Controller Application Cycle Factor) | UINT16 | If the synchronous control task operates at a cycle that is longer than the data cycle (see attribute #1, Cycle time) then this attribute provides a scaling factor for the cycle time such that:<br>Control task duration = Control task cycle factor x Cycle time.<br>The information may be used e.g. to interpolate output values, if required by the process. Note that synchronization to the Control Task cycle must be done by the host application.<br>Default: 1 |

The figure below shows an example of Control Task cycle and Cycle time, with Control Task cycle factor = 3
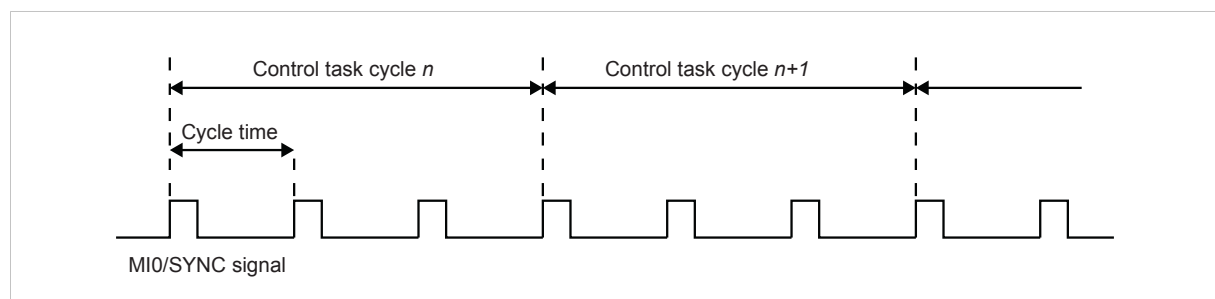


**Fig. 10**

## 11.6        Energy Control Object (F0h)

**Category**

Extended

**Object Description**

This object implements energy control functionality, i.e. energy specific settings, in the host application. The implementation of this object is optional. All instance attributes shall be seen as required and must be implemented in the application. If the Anybus module detects that an attribute is missing during run time an appropriate network error is sent and the Discard Responses counter is increased in the Anybus Object instance attribute Error Counter.

Each enabled instance in the object corresponds to an Energy saving mode. The number of available modes is device specific, and must be defined by the application. The higher the instance number, the more energy is saved. The instance with the highest number always corresponds to the "Power off" mode, i.e. the state where the device is essentially shut down. Instance 1 of the object represents "Ready to operate", i.e. the mode where the device is fully functional and does not save energy at all. Consequently a meaningful implementation always contains at least two instances, one for energy saving and one for operating. If this object is implemented for PROFINET, at least three instances are needed: "Ready to operate", "Energy saving mode 1", and "Power off".

Highest number of instances is 8. Please note that these modes are always present – they are not dynamically created or deleted. It is not allowed to leave holes in the list of instances.
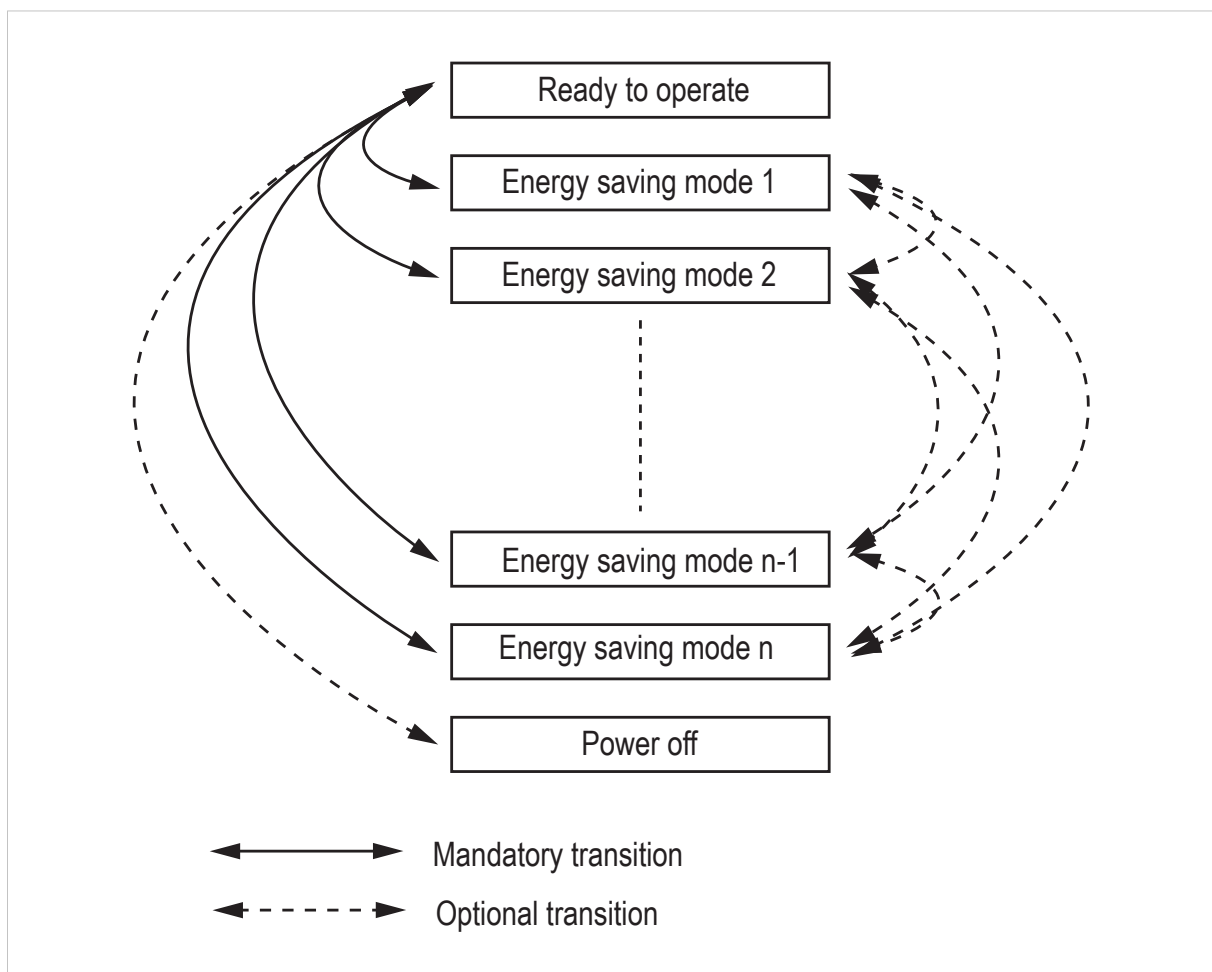


**Fig. 11**

## Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| | StartPause |
| | EndPause |
| | Preview_Pause_Time (not PROFINET) |
| **Instance:** | Get_Attribute |

## Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "Energy Control" |
| 2 | Revision | Get | UINT8 | 02h |
| 3 | Number of instances | Get | UINT16 | - |
| 4 | Highest instance no. | Get | UINT16 | Highest created instance number. Maximum value is 8. |
| 11 | Current Energy Saving Mode | Get | UINT16 | Instance number of the currently used Energy saving mode. During a mode transition the new Energy saving mode shall be presented. "Ready to operate" will equal instance #1, and "Power off" mode will equal Highest instance number. |
| 12 | Remaining time to destination | Get | UINT32 | When changing mode this parameter will reflect the actual time (in milliseconds) remaining until the shift is completed.<br>If a dynamic value cannot be generated the static value for the transition from the source to destination mode shall be used.<br>If the value is infinite, or unknown, the maximum value 0xFFFFFFFF shall be used. |
| 13 | Energy consumption to destination | Get | FLOAT | When changing mode this parameter will reflect the actual energy (in kWh) which will be consumed until the shift is completed.<br>If a dynamic value cannot be generated the static value for the transition from the source to destination mode shall be used.<br>If the value is undefined the value 0.0 shall be used. |
| 14 | Transition to "Power off" mode supported | Get | BOOL | Indicates whether transition to "Power off" mode is supported or not.<br><br>0: Not supported<br>1: Supported |

## Instance Attributes (Instance #1 - #8)

| # | Name | Access | Data Type | Description | |
|---|------|--------|-----------|-------------|---|
| 1 | ModeAttributes | Get | BITS16 | Bit 0: | Meaning: |
| | | | | 0: | Only static time and energy values are available (Value of bit 0 attribute is not implemented) |
| | | | | 1: | Dynamic time and energy values are available |
| | | | | Bit 1-15: | Reserved |
| 2 | TimeMinPause | Get | UINT32 | Minimum pause time in milliseconds. ($t_{pause}$)<br>If the value is infinite, or unknown, the maximum value 0xFFFFFFFF shall be used. | |
| 3 | TimeToPause | Get | UINT32 | Maximum time to go to this Energy saving mode.(ms, $t_{off}$)<br>If the value is infinite, or unknown, the maximum value 0xFFFFFFFF shall be used. | |
| 4 | TimeToOperate | Get | UINT32 | Maximum time needed to go to the "Ready to operate" mode. (ms, $t_{on}$)<br>If the value is infinite, or unknown, the maximum value 0xFFFFFFFF shall be used. | |
| 5 | TimeMinLengthOfStay | Get | UINT32 | The minimum time that the device must stay in this mode. In milliseconds.(ms, $t_{off\_min}$)<br>If the value is infinite, or unknown, the maximum value 0xFFFFFFFF shall be used. | |
| 6 | TimeMaxLengthOfStay | Get | UINT32 | Maximum time that is allowed to stay in this mode. In milliseconds.<br>If no maximum value is available or if not implemented, the maximum value FFFFFFFFh shall be used. | |
| 7 | ModePowerConsump-tion | Get | FLOAT | Amount of power consumed in this mode. (kW)<br>If the value is undefined the value 0.0 shall be used. | |
| 8 | EnergyConsumptionTo-Pause | Get | FLOAT | Amount of energy required to go to this mode. (kWh)<br>If the value is undefined the value 0.0 shall be used. | |
| 9 | EnergyConsumption-ToOperate | Get | FLOAT | Amount of energy required to go to the "Ready to operate" mode from this mode. (kWh)<br>If the value is undefined the value 0.0 shall be used. | |
| 10 | Availability | Get | BOOL | Indicates if this energy saving mode is available given the current device state.<br>Not used for PROFINET. | |
| | | | | False | Not available |
| | | | | True | Available (Value if attribute not implemented) |
| 11 | Power Consumption | Get | UINT32 | Indicates the power consumption of the device when in this state.<br>Not used for PROFINET. | |

**Command Details: Start_Pause**

**Details**

| Command Code | 10h |
|---|---|
| **Valid for:** | Object |

**Description**

This command is sent to the host application when the system wants to initialize a pause of the system. The length of the pause is specified in milliseconds. The response of the message contains the destination mode (i. e. the instance number of the selected energy saving mode).

- Command Details

| Field | Contents | Comments |
|---|---|---|
| Data[0] | Pause time (low word, low byte) | Pause time (ms) |
| Data[1] | Pause time (low word, high byte) | |
| Data[2] | Pause time (high word, low byte) | |
| Data[3] | Pause time (high word, high byte) | |

- Response Details

| Field | Contents | Comments |
|---|---|---|
| Data[0] | Instance number (low byte) | Instance number of selected Energy mode |
| Data[1] | Instance number (low byte) | |

If the application is unable to select a state, given the requested pause time, it shall return one of the error codes in the table below.

| # | Error code | Description |
|---|---|---|
| 0x0D | Invalid state | Given the state of the device and the requested pause time it is currently not possible to enter any energy saving mode |
| 0x12 | Value too low | The requested pause time is too short |

**Command Details: End_Pause**

**Details**

| | |
|---|---|
| **Command Code** | 11h |
| **Valid for:** | Object |

**Description**

This command is sent to the host application when the system wants to return the system from a pause mode back to "Ready to operate" mode. In the response message the number of milliseconds to actualize the switch is returned.

- Command Details

  (none)

- Response Details

| Field | Contents | Comments |
|---|---|---|
| Data[0] | Time To Operate (low word, low byte) | Time needed to switch to "Ready to operate" |
| Data[1] | Time To Operate (low word, high byte) | |
| Data[2] | Time To Operate (high word, low byte) | |
| Data[3] | Time To Operate (high word, high byte) | |

If the application is unable to end the pause it shall return the error code in the table below.

| # | Error code | Description |
|---|---|---|
| 0x0D | Invalid state | Given the state of the device, it is currently not possible to end the pause |

**Command Details: Preview_Pause_Time**

**Details**

| | |
|---|---|
| **Command Code** | 12h |
| **Valid for:** | Object |

**Description**

Not used for PROFINET devices.

This command is sent to the host application when the system wants to preview the application's choice of Energy saving mode. The length of the pause is specified in milliseconds. The response shall contain the destination mode the application would have chosen if the StartPause service was sent (that is, the instance number of the selected energy saving mode). No transition to an Energy saving mode occurs.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| Data[0] | Pause time (low word, low byte) | Pause time (ms) |
| Data[1] | Pause time (low word, high byte) | |
| Data[2] | Pause time (high word, low byte) | |
| Data[3] | Pause time (high word, high byte) | |

- Response Details

| Field | Contents | Comments |
|---|---|---|
| Data[0] | Instance number (low byte) | Instance number of selected Energy mode |
| Data[1] | Instance number (low byte) | |

If the application is unable to select a state, given the requested pause time, it shall return one of the error codes in the table below.

| # | Error code | Description |
|---|---|---|
| 0x0D | Invalid state | Given the state of the device and the requested pause time it is currently not possible to enter any energy saving mode |
| 0x12 | Value too low | The requested pause time is too short |

## 11.7 PROFINET IO Object (F6h)

### Category

Basic, extended

### Object Description

This object implements PROFINET IO related settings in the host application.

The implementation of this object is optional; the host application can support none, some, or all of the attributes specified below. The module will attempt to retrieve the values of these attributes during startup; if an attribute is not implemented in the host application, simply respond with an error message (06h, "Invalid CmdExt[0]"). In such a case, the module will use its default value.

If the module attempts to retrieve a value of an attribute not listed below, respond with an error message (06h, "Invalid CmdExt[0]").

See also...

- *Anybus CompactCom 40 Software Design Guide*, "Error Codes"

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute (01h) |
| | Get_Record (10h, see below) |
| | Set_Record (11h, see below) |
| | AR_Check_Ind (14h, see below) |
| | Cfg_Mismatch_Ind (15h, see below) |
| | Expected_Ident_Ind (1Bh, see below) |
| | End_Of_Prm_Ind (17h, see below) |
| | AR_Abort_Ind (19h, see below) |
| | Indicate_Device (1Eh, see below) |
| | Prm_Begin_Ind (1Fh, see below) |
| **Instance:** | Get_Attribute (01h) |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|------|--------|-----------|-------|
| 1 | Name | Get | Array of CHAR | "PROFINET IO" |
| 2 | Revision | Get | UINT8 | 02h |
| 3 | Number of instances | Get | UINT16 | 0001h |
| 4 | Highest instance no. | Get | UINT16 | 0001h |

## Instance Attributes (Instance #1)

**Basic**

| # | Name | Access | Data Type | Default | Comment |
|---|------|--------|-----------|---------|---------|
| 1 | Device ID | Get | UINT16 | 0010h | Identifies the device.<br>(Assigned by manufacturer)<br>If not implemented, the default value will be used. |
| 2 | Vendor ID<br>(I&M Manufacturer ID) | Get | UINT16 | 010Ch (HMS Industrial Networks) | (Assigned by PROFIBUS & PROFINET International (PI))<br>If not implemented, the default value will be used. |
| 3 | Station Type | Get | Array of CHAR | "CompactCom 40 PIR" | Characterizes the device.<br>Assigned by manufacturer); up to 25 characters.<br>If not implemented, the value of attribute #9 (Product name) in the Application Object (FFh), will be used. If this attribute is not implemented, the default value will be used. |
| 8 | I&M Order ID | Get | Array of CHAR | "CompactCom 40 PIR" | I&M0 Parameter:<br>Order ID of device; up to 20 characters.<br>If not implemented, the default value will be used. |
| 9 | I&M Serial Number | Get | Array of CHAR | Assigned during manufacturing | I&M0 Parameter:<br>Serial number of device; up to 16 characters.<br>If not implemented, the value of attribute #3 (Serial number) in the Application Object (FFh), will be used. If this attribute is not implemented, the default value will be used. |

**GSD Entries**

The GSDML entries below must match the values of the corresponding attributes in the PROFINET IO object.

- Attributes #1 and #2 correspond to the following entry in the GSDML file:

  `<DeviceIdentity VendorID="0x010C" DeviceID="0x0010">`

- Attribute #8 correspond to the following entry in the GSDML file:

  <OrderNumber Value="CompactCom 40 PIR"/>

**Extended**

- If an attribute is not implemented, the default value will be used.

- The Anybus module in itself does not alter its behavior based on the value of attributes #13 and #14. The host application has to implement the corresponding functionality.

- The module is preprogrammed with a valid Mac address. To use that address, do not implement attributes #17 and #18.

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| 4 | MaxAr | Get | UINT32 | 0000 0003h | Max. no.of simultaneous ARs. (Range 1... 4) **Note 1**: With S2Enabled = True, MaxAr needs to be at least 2. GSDML attribute NumberOfDeviceAccessAR must be set to MaxAr - 2. **Note 2**: With S2Enabled = False, MaxAr needs to be at least 1. GSDML attribute NumberOfDeviceAccessAR must be set to MaxAr - 1. |
| 5 6 | (Reserved) | - | - | - | Reserved for future use |
| 7 | Record Data Mode | Get | UINT8 | 00h | This setting affects how Record Data requests are treated, and holds a bit field as follows: Bit 0:Index 0... 7FFFh:  0:  Normal Mode  1:  Transparent Mode Bit 1:Index AFF0h... AFFFh:  Reserved (replaced with the command IM_Options, see *Network PROFINET IO Object (0Eh), p. 66*) |
| 10 | I&M Hardware Revision | Get | UINT16 | Hardware Rev. | I&M0 Parameter: Hardware revision of device; FFFFh indicates availability of profile specific information If not implemented, the value of attribute #11 (Hardware version) in the Application Object (FFh), will be used. If this attribute is not implemented, the default value will be used. |
| 11 | I&M Software Revision | Get | Struct of CHAR 3*UINT8 | Software Rev. | I&M0 Parameter: Software revision of device. Byte:      Value:         Meaning: 0:        "V"            Official release          "R"            Revision          "P"            Prototype          "U"            Under test          "T"            Test Device 1:        0... 255      Major Version 2:        0... 255      Minor Version 3:        0... 255      Internal Change Please note that version V255.255.255 indicates availability of profile specific information. If not implemented, the value of attribute #10 (Firmware version) in the Application Object (FFh), will be used. If this attribute is not implemented, the default value will be used. |
| 12 | I&M Revision Counter | Get | UINT16 | 0000h | I&M0 Parameter: Revision counter of device; a changed value marks a change of the hardware or its parameters. |
| 13 | I&M Profile ID | Get | UINT16 | 0000h (Non Profile Device) | I&M0 Parameter: If the application supports a specific profile, the Profile ID must be defined here. It shall equal the API number for the profile.. Please not that: this will not change the behavior of the module, since it does not handle profiles. |
| 14 | I&M Profile Specific Type | Get | UINT16 | Generic Profile: 0004h (Communica-tion Module) | I&M0 Parameter: If the application supports a specific profile, the profile specific type must be defined here (value is defined by the profile). |

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| | | | | | Please not that: this will not change the behavior of the module, since it does not handle profiles. |
| 15-16 | (Reserved) | - | - | - | Reserved, not used |
| 17 | Port 1 MAC address | Get | Array of UINT8 | - | 6 byte physical address value; overrides the preprogrammed Mac address. Do not implement this attribute if the preprogrammed Mac address is to be used. |
| 18 | Port 2 MAC address | Get | Array of UINT8 | - | 6 byte physical address value; overrides the preprogrammed Mac address. Do not implement this attribute if the preprogrammed Mac address is to be used. |
| 19-23 | (Reserved) | - | - | - | Reserved, not used |
| 24 | Custom Station Name | Get | Array of CHAR | Product specific. (default: "CompactCom 40 PIR") | When using shift register mode, this value is used to create the station name. If for example the attribute has the value abcc40-pir, the station name will be abcc40-pir-XYZ where XYZ is the value of DIP2. Valid characters are letters "a" to "z", numbers "0" to "9", hyphens and periods. The attribute value must begin with a letter. Maximum number of elements in array: 58 For more information, see the *Anybus CompactCom M40 Hardware Design Guide*. |
| 25 | IM Module Order ID | Get | Array of CHAR | Anybus CompactCom 40 PROFINET IRT | I&M5 parameter. Order ID of the communication interface. Maximum number of elements in array is 64. Not available for Anybus IP. |
| 26 | IM Annotation | Get | Array of CHAR | Anybus CompactCom 40 PROFINET IRT | I&M5 parameter Description of the communication interface. Maximum number of elements in array: 64. Not available for Anybus IP. |
| 27 | IM5 Enabled | Get | BOOL | True | Enable I&M5 support. True: I&M5 enabled False: I&M5 disabled |
| 28 | S2 Enabled | Get | BOOL | False | Enable S2 system redundancy. True: S2 enabled False: S2 disabled |
| 29 | S2 Primary AR handle | Set | UINT16 | 0 | AR handle of currently primary AR when S2 is enabled (attribute #28). A switchover (Backup to Primary) or loss of Primary triggers an update of the AR handle indicating change of primary AR. The handle may be used by the application to reference AR specific commands; e.g. AR_Check_Ind that contains IP address and Station name of the IO controller. The value 0 indicates "No Primary AR" or "No S2 connection". |

## Command Details: Get_Record

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 10h |
| **Valid for:** | Object Instance |

### Description

The module issues this command in the following situations:

- Module receives a Record Data Read request towards an API other than 0 (zero).

- Module receives a Record Data Read request towards API 0, but the record in question is handled in Transparent Mode.

  See instance attribute #7 for more information about Transparent Mode.

See *Flowchart — Record Data Access, p. 146* for more information.

It is optional to implement support for this command. If not implemented, the original network request will be rejected and an error is returned to the IO Controller/Supervisor.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | API (low word, low byte) | Application Process Instance (API) |
| MsgData[1] | API (low word, high byte) | |
| MsgData[2] | API (high word, low byte) | |
| MsgData[3] | API (high word, high byte) | |
| MsgData[4] | Slot (low byte) | Slot number of request |
| MsgData[5] | Slot (high byte) | |
| MsgData[6] | Subslot (low byte) | Subslot number of request |
| MsgData[7] | Subslot (high byte) | |
| MsgData[8] | Index (low byte) | Index of request |
| MsgData[9] | Index (high byte) | |
| MsgData[10] | Length (low byte) | Range: 1 - 1524 |
| MsgData[11] | Length (high byte) | MsgData[11] is only available if the length value exceeds 255. |

- Response Details (Success)

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0... 1] | (reserved) | (set to zero) |
| MsgData[0... n] | Data (up to 1524 bytes) | Data to be returned in the Record Data Read response. |

- Response Details (Error)

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0... 1] | (reserved) | (set to zero) |
| MsgData[0] | FFh | Object specific error |
| MsgData[1] | Error Code 1 | See *Details: Error Code 1, p. 137* |
| MsgData[2] | Error Code 2 | User specific error code |
| MsgData[3] | Additional Data 1 | API specific. Set to zero if no Additional Data 1 is defined. |
| MsgData[4] | Additional Data 2 | User specific. Set to zero if no Additional Data 2 is defined. |

See also...

- Details for the command Set_Record (below)

-

## Command Details: Set_Record

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 11h |
| **Valid for:** | Object Instance |

### Description

The module issues this command in the following situations:

- Module receives a Record Data Write request towards an API other than 0 (zero).

- Module receives a Record Data Write request towards API 0, but the record in question is handled in Transparent Mode

    See instance attribute #7 for more information about Transparent Mode.

See "Flowchart - Record Data Access" on page 154 for more information.

It is optional to implement support for this command. If not implemented, the original network request will be rejected and an error is returned to the IO Controller/Supervisor.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | API (low word, low byte) | Application Process Instance (API) |
| MsgData[1] | API (low word, high byte) | |
| MsgData[2] | API (high word, low byte) | |
| MsgData[3] | API (high word, high byte) | |
| MsgData[4] | Slot (low byte) | Slot number of request |
| MsgData[5] | Slot (high byte) | |
| MsgData[6] | Subslot (low byte) | Subslot number of request |
| MsgData[7] | Subslot (high byte) | |
| MsgData[8] | Index (low byte) | Index of request |
| MsgData[9] | Index (high byte) | |
| MsgData[10] | (reserved) | (set to zero) |
| MsgData[11..n] | Data (up to 1512 bytes) | Data from the Record Data Write request. |

- Response Details (Success)

    (no data)

- Response Details (Error)

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0... 1] | (reserved) | (set to zero) |
| MsgData[0] | FFh | Object specific error |
| MsgData[1] | Error Code 1 | See *Details: Error Code 1, p. 137* |
| MsgData[2] | Error Code 2 | User specific error code |
| MsgData[3] | Additional Data 1 | API specific. Set to zero if no Additional Data 1 is defined. |
| MsgData[4] | Additional Data 2 | User specific. Set to zero if no Additional Data 2 is defined. |

See also...

- Command details for "Get_Record"

- *Flowchart — Record Data Access, p. 146*

## Command Details: AR_Check_Ind

### Category

Extended

### Details

| Command Code | 14h |
| --- | --- |
| **Valid for:** | Object Instance |

### Description

The module issues this command to inform the host application that an Application Relationship (AR) is to be established. It is optional to implement support for this command.

- Command Details

| Field | Contents | Comments |
| --- | --- | --- |
| CmdExt[0] | AR handle (low byte) | Handle for Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | IP address (low word, low byte) | IP address of the remote station (IO Controller/Supervisor) |
| MsgData[1] | IP address (low word, high byte) | |
| MsgData[2] | IP address (high word, low byte) | |
| MsgData[3] | IP address (high word, high byte) | |
| MsgData[4] | AR Type (low byte) | Indicates the type of AR as follows: |
| MsgData[5] | AR Type (high byte) | Value: Meaning:<br><br>0001h IOCARSingle (RT connection to single IO Connector)<br><br>0006h IOSAR (Used in combination with ARProperties. DeviceAccess for connections without process data)<br><br>0010h IOCARSingle using RT_CLASS_3 (IRT connection to single IO Controller)<br><br>0020h IOCARSR (Variant of IOCARSingle indicating usage of System Redundancy) |
| MsgData[6] | AR Properties (low word, low byte) | Bit-field indicating the properties of the AR as described in the table below.: |
| MsgData[7] | AR Properties (low word, high byte) | |
| MsgData[8] | AR Properties (high word, low byte) | |
| MsgData[9] | AR Properties (high word, high byte) | |
| MsgData[10] | Remote station name length | Length of remote station name, in bytes |
| MsgData[11..n] | Remote station name | Remote station name (IO Controller/Supervisor) |

AR Properties:

| Bit(s) | Value / Description |
| --- | --- |
| 0-2 | State:<br><br>0: (reserved)<br><br>1: Active<br><br>2-7: (reserved) |
| 3 | Supervisor take over allowed:<br><br>0: Not allowed<br><br>1: Allowed |
| 4 | Parameterization server:<br><br>0: Externak PrmServer |

| Bit(s) | Value / Description | |
|--------|--------------------|---|
| | 1: | CM Initiator |
| 5-7 | (reserved) | |
| 8 | Device Access: | |
| | 0: | Only submodules from ExpectedSubmoduleBlock are accessible |
| | 1: | Submodule access is controlled by IO device application |
| 9-10 | Companion AR: | |
| | 0: | Single AR |
| | 1-3: | (reserved) |
| 11 | AcknowledgeCompanion AR: | |
| | 0: | No companion AR |
| | 1: | (reserved) |
| 12-27 | (reserved) | |
| 28 | Time aware system: | |
| | 0: | Non time aware |
| | 1: | Time aware |
| 29 | Combined object container (COC) | |
| | with StartupMode = Legacy | |
| | 0: | No COC |
| | 1: | (reserved) |
| | with StartupMode = Advanced | |
| | 0: | No COC |
| | 1: | Usage of COC required |
| 30 | Startup mode: | |
| | 0: | Legacy |
| | 1: | Advanced |
| 31 | Pull module alarm allowed: | |
| | 0: | Subslot number 0 shall indicate pulling of module in conjunction with AlarmType(=Pull) |
| | 1: | Sublslot number 0-9FFFh shall indicate pulling of submodule in conjunction with AlarmType (=Pull) |

• Response Details

(no data)

## Command Details: Cfg_Mismatch_Ind

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 15h |
| **Valid for:** | Object Instance |

### Description

The module issues this command to inform the host application that the configuration in the IO Controller (i.e. the Expected Identification) does not match the configuration defined by the host application (i.e. the Real Identification).

It is optional to implement support for this command.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | API (low word, low byte) | Application Process Instance (API) |
| MsgData[1] | API (low word, high byte) | |
| MsgData[2] | API (high word, low byte) | |
| MsgData[3] | API (high word, high byte) | |
| MsgData[4] | Slot (low byte) | Slot number of mismatch |
| MsgData[5] | Slot (high byte) | |
| MsgData[6] | Subslot (low byte) | Subslot number of mismatch |
| MsgData[7] | Subslot (high byte) | |
| MsgData[8] | Expected Module Identifier (low word, low byte) | Module identifier (as stated in the GSD file) derived from the IO Controller configuration |
| MsgData[9] | Expected Module Identifier (low word, high byte) | |
| MsgData[10] | Expected Module Identifier (high word, low byte) | |
| MsgData[11] | Expected Module Identifier (high word, high byte) | |
| MsgData[12] | Expected Submodule Identifier (low word, low byte) | Submodule identifier (as stated in the GSD file) derived from the IO Controller configuration |
| MsgData[13] | Expected Submodule Identifier (low word, high byte) | |
| MsgData[14] | Expected Submodule Identifier (high word, low byte) | |
| MsgData[15] | Expected Submodule Identifier (high word, high byte) | |

- Response Details

  (no data)

## Command Details: Expected_Ident_Ind

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 1Bh |
| **Valid for:** | Object Instance |

### Description

The module issues this command to inform the host application of the Expected Identification (Module/ Submodule List) that the IO Controller will use for the established AR.

Note that this information may be split in multiple segments, which means that this command will be issued multiple times by the module, each time containing different parts of the configuration.

Expected_Ident_Ind is similar to AR_Info_Ind but uses a different segmentation protocol that shall be used for the 40 series concept, see Anybus CompactCom 40 Software Design Guide (Message Segmentation).

For very large configurations where the Expected Identification cannot fit into one message, this segmentation protocol will be used. If the number of modules/sub-modules exceeds the capabilities of the Anybus CompactCom the message will be truncated. The size of the configuration can be up to 2370 bytes at maximum number of modules/sub-modules.

It is optional to implement support for this command.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | Reserved | For segmented messages the CmdExt[1] byte has been reserved for segment bits,see Anybus CompactCom 40 Software Design Guide (Message Segmentation).. |
| CmdExt[1] | Cmd segment bits | |
| MsgData[0] | AR handle (low byte) | Handle for the Application Relationship. |
| MsgData[1] | AR handle (high byte) | |
| MsgData[2.. n] | Data field | The first two bytes in the initial block of the Data field indicates the number of modules in the configuration. Each module is represented by a Module block, followed by a number of Submodule blocks (provided that the module in question contains submodules). See "Data Format" below for coding of the data field. |

- Response Details

| Field | Contents | Comments | |
|-------|----------|----------|---|
| CmdExt[0] | Reserved | (set to zero) | |
| CmdExt[1] | | | |
| MsgData[0] | Continue/Block | Value: | Meaning: |
| | | 0 | Continue. The host application will not perform any changes to the Real Identification, A connect response may be sent on the network immediately following this response. |
| | | 1 | Block. The host application will perform changes in the Real identification in response to the received Expected identification. A connect response will not be sent on the network until Ident_Change_Done is sent by the host application |
| | | Returning any error response or providing no data/ too much data will give the ssame result as responding with Continue (0). | |
| | | The timeout is 300 seconds. A "Continue" response or the Ident_Change_Done command should be sent well within this time limit. | |

**Data Format**

When all data has been received, the resulting data shall be interpreted as follows:

| Type | Name | | | | Description |
|------|------|---|---|---|-------------|
| UINT16 | iNbrApi | | | | Number of APIs in configuration |
| UINT32 | | iApiNbr | | | Initial module block including API number and number of module blocks in the API. |
| UINT16 | | iNbrMod | | | |
| UINT16 | | | iSlotNbr | | Module block (8 bytes), see below. |
| UINT16 | | | iNbrSubMod | | |
| UINT32 | | | lModIdent | | |
| UINT16 | | | | iSubSlotNbr | Submodule block (10 bytes), see below. |
| UINT32 | | | | lSubModIdent | |
| UINT16 | | | | iInDataLength | |
| UINT16 | | | | iOutData-Length | |

The initial API block (iNbrApi) defines the number of APIs in the configuration.

Each API has an initial module block, that includes information on the API number (iApiNbr) and the number of modules (or slots) in the API.

Each module starts with a module block, which holds the slot number, the number of submodules (or subslots) and the module identity number.

Finally each submodule block holds subslot number, submodule identification number, input and output data lengths.

**Example**



| Initial API Block | No. of APIs | 0002h |
| Initial Module Block (API #0) | API no.<br>No. of Modules | 00 00 00 00h<br>0002h |
| Module Block (Module #1) | Slot no.<br>No. of Submodules<br>Module ID | 0001h<br>0003h<br>4A 6F 48 62h |
| Submodule Block (Module #1) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0001h<br>65 6C 69 65h<br>0004h<br>0010h |
| Submodule Block (Module #1) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0002h<br>76 65 73 69h<br>0008h<br>0002h |
| Submodule Block (Module #1) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0003h<br>6E 53 61 6Eh<br>0008h<br>0002h |
| Module Block (Module #2) | Slot no.<br>No. of Submodules<br>Module ID | 0002h<br>0001h<br>74 61 43 6Ch |
| Submodule Block (Module #2) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0001h<br>61 75 73 21h<br>0010h<br>0001h |
| Initial Module Block (API #2) | API no.<br>No. of Modules | 00 00 00 02h<br>0001h |
| Module Block (Module #1) | Slot no.<br>No. of Submodules<br>Module ID | 0001h<br>0002h<br>4A 6F 48 82h |
| Submodule Block (Module #1) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0001h<br>65 6C 67 65h<br>0004h<br>0010h |
| Submodule Block (Module #1) | Subslot no.<br>Submodule ID<br>Input Data Length<br>Output Data Length | 0002h<br>76 65 74 69h<br>0008h<br>0002h |

**Fig. 12**

In this example, the configuration contains two APIs with the following properties:

- API #0 contains two modules, the first with three submodules, the second with one submodule

- API #2 contains one module with two submodules

## Command Details: End_Of_Prm_Ind

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 17h |
| **Valid for:** | Object Instance |

### Description

The module may issue this command to indicate to the host application that the parameterization phase is completed. It is optional to implement support for this command.

If implemented, the host application may, depending on the response issued to this command, be required to issue 'Appl_State_Ready' at a later stage to indicate that it is ready for data exchange. If not implemented, this is handled automatically by the module.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | API (low word, low byte) | Application Process Instance (API) <br> - Only valid if subslot > 0 |
| MsgData[1] | API (low word, high byte) | |
| MsgData[2] | API (high word, low byte) | |
| MsgData[3] | API (high word, high byte) | |
| MsgData[4] | Slot (low byte) | Slot number affected by the command <br> - Only valid if subslot > 0 |
| MsgData[5] | Slot (high byte) | |
| MsgData[6] | Subslot (low byte) | Subslot number affected by the command |
| MsgData[7] | Subslot (high byte) | Value:   Meaning: <br> 0:    Command applies to all modules in the <br> other:  configuration <br>          Command applies only to the specified slot/ <br>          subslot |

- Response Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved) | (set to zero) |
| CmdExt[1] | | |
| MsgData[0] | Subslot (high byte) | Value:   Meaning: <br> 0:    Ready for Data Exchange <br> 1:    Not ready for data exchange <br>       (Appl_State_Ready must be issued at a later <br>       stage) |

See also...

- Appl_State_Ready, details in *Network PROFINET IO Object (0Eh), p. 66*

## Command Details: AR_Abort_Ind

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 19h |
| **Valid for:** | Object Instance |

### Command Details: AR_Abort_Ind

**Description**

This command is optional. The module issues this command to indicate to the host application that an
Application Relationship (AR) is aborted (by the application or any other source).

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| MsgData[0] | Reason code (low byte) | Reason code for the offline transition |
| MsgData[1] | Reason code (high byte) | <table><tr><td>Value:</td><td>Reason:</td></tr><tr><td>0</td><td>No reason (unknown reason)</td></tr><tr><td>3</td><td>Out of memory</td></tr><tr><td>4</td><td>Add provider or consumer failed</td></tr><tr><td>5</td><td>Miss (consumer)</td></tr><tr><td>6</td><td>Cmi timeout</td></tr><tr><td>7</td><td>Alarm-open failed</td></tr><tr><td>8</td><td>Alarm-send.cnf(-)</td></tr><tr><td>9</td><td>Alarm-ack-send.cnf(-)</td></tr><tr><td>10</td><td>Alarm-data too long</td></tr><tr><td>11</td><td>Alarm.ind(err)</td></tr><tr><td>12</td><td>Rpc-client call.cnf(-)</td></tr><tr><td>13</td><td>Ar-abort.req</td></tr><tr><td>14</td><td>Re-run aborts existing</td></tr><tr><td>15</td><td>Got release.ind</td></tr><tr><td>16</td><td>Device passivated</td></tr><tr><td>17</td><td></td></tr><tr><td>18</td><td>Protocol violation</td></tr><tr><td>19</td><td>NARE error</td></tr><tr><td>20</td><td>RPC-Bind error</td></tr><tr><td>21</td><td>RPC-Connect error</td></tr><tr><td>22</td><td>RPC-Read error</td></tr><tr><td>23</td><td>RPC-Write error</td></tr><tr><td>24</td><td>RPC-Control error</td></tr><tr><td>25</td><td>Forbidden pull or plug after check.rsp and before in-data.ind</td></tr><tr><td>26</td><td>AP removed</td></tr><tr><td>27</td><td>Link down</td></tr><tr><td>28</td><td>Could not register multicast-mac</td></tr><tr><td>29</td><td>Not synchronized (cannot start companion-ar)</td></tr><tr><td>30</td><td>Wrong topology (cannot start companion-ar)</td></tr><tr><td>31</td><td>Dcp, station-name changed</td></tr><tr><td>32</td><td>Dcp, reset to factory-settings</td></tr><tr><td>33</td><td>Cannot start companion AR because of parameter error</td></tr></table> |

- Response Details

  (no data)

## Command Details: Indicate_Device

### Category

Extended

### Details

| | |
|---|---|
| **Command Code** | 1Eh |
| **Valid for:** | Object Instance |

### Description

This command is optional. The module issues this command to inform the application that the DCP command Set Control/Signal has been received on the network. This is used by engineering tools to identify the node on the network. The application should flash its dedicated Network Status LED for 3 seconds after receiving this command.

- Command Details

| Field | Contents | Comments |
|---|---|---|
| CmdExt[0] | (reserved) | |
| CmdExt[1] | | |

- Response Details

  (no data)

## Command Details: Prm_Begin_Ind

**Category**

Extended

**Details**

| Command Code | 1Fh |
| --- | --- |
| **Valid for:** | Object Instance |

**Description**

This command is optional. The module issues this command to indicate to the host application that a parameterization phase, outside of the connection establishment, is initiated. During the parameterization phase the host application may receive Set_Record commands targeting the submodules addressed by the Prm_Begin_Ind command.

- Command Details

| Msg field | Contents | Comments |
| --- | --- | --- |
| CmdExt[0] | AR handle (low byte) | Handle for the Application Relationship (AR) |
| CmdExt[1] | AR handle (high byte) | |
| Data[0] | API (low word, low byte) | Application Process Instance<br>Only valid if Sub-slot > 0 |
| Data[1] | API (low word, high byte) | |
| Data[2] | API (high word, low byte) | |
| Data[3] | API (high word, high byte) | |
| Data[4] | Slot (low byte) | Slot number affected by the command<br>Only valid if Sub-slot > 0 |
| Data[5] | Slot (high byte) | |
| Data[6] | Sub-slot (low byte) | Sub-slot number affected by the command |
| Data[7] | Sub-slot (high byte) | 0 — Command applies to all modules in the configuration<br>>0 — Command applies to the specified slot/sub-slot |

- Response Details

  (no data)

## Details: Error Code 1

The error codes below shall be used when providing error responses to the following commands:

- Get_Record

- Set_Record

- Get_IM_Record

- Set_IM_Record

| High nibble (bits 4... 7) | | Low nibble (bits 0... 3) | |
|---|---|---|---|
| ErrorClass | Meaning | ErrorCode | Meaning |
| 0... 9 | Reserved | (reserved) | (reserved) |
| 10 | Application | 0 | Read error |
| | | 1 | Write error |
| | | 2 | Module error |
| | | 3... 6 | (reserved) |
| | | 7 | Busy |
| | | 8 | Version conflict |
| | | 9 | Feature not supported |
| | | 10... 15 | User specific |
| 11 | Access | 0 | Invalid index |
| | | 1 | Write length error |
| | | 2 | Invalid slot/subslot |
| | | 3 | Type conflict |
| | | 4 | Invalid area |
| | | 5 | State conflict |
| | | 6 | Access denied |
| | | 7 | Invalid range |
| | | 8 | Invalid parameter |
| | | 9 | Invalid type |
| | | 10 | Backup |
| | | 11... 15 | User specific |
| 12 | Resource | 0 | Read constrain conflict |
| | | 1 | Write constrain conflict |
| | | 2 | Resource busy |
| | | 3 | Resource unavailable |
| | | 4... 7 | (reserved) |
| | | 8... 15 | User specific |
| 13... 15 | User specific | (user specific) | User specific |

See also...

- Command details for Get_Record

- Command details for Set_Record

## 11.8    Ethernet Host Object (F9h)

### Object Description

This object implements Ethernet features in the host application.

### Supported Commands

| | |
|---|---|
| **Object:** | Get_Attribute |
| **Instance:** | Get_Attribute |
| | Set_Attribute |

### Object Attributes (Instance #0)

| # | Name | Access | Data Type | Value |
|---|---|---|---|---|
| 1 | Name | Get | Array of CHAR | "Ethernet" |
| 2 | Revision | Get | UINT8 | 02h |
| 3 | Number of instances | Get | UINT16 | 0001h |
| 4 | Highest instance no. | Get | UINT16 | 0001h |

### Instance Attributes (Instance #1)

- If an attribute is not implemented, the default value will be used.

- The module is preprogrammed with a valid MAC address. To use that address, do not implement attribute #1.

- Do not implement attributes #9 and #10, only used for PROFINET devices, if the module shall use the preprogrammed MAC addresses.

- If new MAC addresses are assigned to a PROFINET device, these addresses (in attributes #1, #9, and #10) have to be consecutive, e.g. (xx:yy:zz:aa:bb:01), (xx:yy:zz:aa:bb:02), and (xx:yy:zz:aa:bb:03).

| # | Name | Access | Data Type | Default Value | Comment |
|---|---|---|---|---|---|
| 1 | MAC address | Get | Array of UINT8 | - | 6 byte physical address value; overrides the preprogrammed Mac address. Note that the new Mac address value must be obtained from the IEEE.<br>Do not implement this attribute if the preprogrammed Mac address is to be used. |
| 2 | Enable HICP | Get | BOOL | True (Enabled) | Enable/Disable HICP |
| 3 | Enable Web Server | Get | BOOL | True (Enabled) | Enable/Disable Web Server<br>(Not used if Transparent Ethernet is enabled.) |
| 4 | (reserved) | | | | Reserved for Anybus CompactCom 30 applications. |
| 5 | Enable Web ADI access | Get | BOOL | True (Enabled) | Enable/Disable Web ADI access<br>(Not used if Transparent Ethernet is enabled.) |
| 6 | Enable FTP server | Get | BOOL | True (Enabled) | Enable/Disable FTP server<br>(Not used if Transparent Ethernet is enabled or if device supports IIoT secure functionality.) |
| 7 | Enable admin mode | Get | BOOL | False (Disabled) | Enable/Disable admin mode<br>(Not used if Transparent Ethernet is enabled.) |
| 8 | Network Status | Set | UINT16 | - | See below. |

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| 9 | Port 1 MAC address | Get | Array of UINT8 | - | **Note**: This attribute is only valid for PROFINET devices.<br>6 byte MAC address for port 1 (mandatory for the LLDP protocol).<br>This setting overrides any Port MAC address in the host PROFINET IO Object.<br>Do not implement this attribute if the preprogrammed Mac address is to be used. |
| 10 | Port 2 MAC address | Get | Array of UINT8 | - | **Note**: This attribute is only valid for PROFINET devices.<br>6 byte MAC address for port 2 (mandatory for the LLDP protocol).<br>This setting overrides any Port MAC address in the host PROFINET IO Object.<br>Do not implement this attribute if the preprogrammed Mac address is to be used. |
| 11 | Enable ACD | Get | BOOL | True (Enabled) | Enable/Disable ACD protocol.<br>If ACD functionality is disabled using this attribute, the ACD attributes in the CIP TCP/IP object (F5h) are not available. |
| 12 | Port 1 State | Get | ENUM | 0 (Enabled) | The state of Ethernet port 1.<br><br>• This attribute is not read by EtherCAT and Ethernet POWERLINK devices, where Port 1 is always enabled.<br><br>00h: Enabled<br><br>01h: Disabled.<br>The port is treated as existing. References to the port can exist, e.g. in network protocol or on website. |
| 13 | Port 2 State | Get | ENUM | 0 (Enabled) | The state of Ethernet port 2.<br><br>• This attribute is not read by EtherCAT and Ethernet POWERLINK devices, where Port 2 is always enabled.<br><br>00h: Enabled<br><br>01h: Disabled.<br>The port is treated as existing. References to the port can exist, e.g. in network protocol or on website.<br><br>02h: Inactive.<br>The attribute is set to this value for a device that only has one physical port. All two-port functionality is disabled. No references can be made to this port.<br>**Note:** This functionality is available for PROFINET, Ethernet/IP and Modbus-TCP devices. |
| 14 | (reserved) | | | | |
| 15 | Enable reset from HICP | Get | BOOL | 0 = False | Enables the option to reset the module from HICP. |
| 16 | IP configuration | Set | Struct of:<br>UINT32 (IP address)<br>UINT32 (Subnet mask)<br>UINT32 (Gateway) | N/A | Whenever the configuration is assigned or changed, the Anybus CompactCom module will update this attribute. |

| # | Name | Access | Data Type | Default Value | Comment |
|---|------|--------|-----------|---------------|---------|
| 17 | IP address byte 0–2 | Get | Array of UINT8 [3] | [0] = 192 [1] = 168 [2] = 0 | First three bytes in IP address. Used in standalone shift register mode if the configuration switch value is set to 1-245. In that case the IP address will be set to: Y[0].Y[1].Y[2].X Where Y0-2 is configured by this attribute and the last byte X by the configuration switch. |
| 18 | Ethernet PHY Configuration | Get | Array of BITS16 | 0x0000 for each port | Ethernet PHY configuration bit field. The length of the array shall equal the number of Ethernet ports of the product. Each element represents the configuration of one Ethernet port (element #0 maps to Ethernet port #1, element #1 maps to Ethernet port #2 and so on). **Note**: Only valid for EtherNet/IP and Modbus-TCP devices. Bit 0:       Auto negotiation fallback duplex 0 = Half duplex 1 = Full duplex Bit 1–15:       Reserved |
| 20 | SNMP read-only community string | Get | Array of CHAR | "public" | **Note**: This attribute is only valid for PROFINET devices. Sets the SNMP read-only community string. Max length is 32. |
| 21 | SNMP read-write community string | Get | Array of CHAR | "private" | **Note**: This attribute is only valid for PROFINET devices. Sets the SNMP read-write community string. Max length is 32. |
| 22 | DHCP Option 61 source | Get | ENUM | 0 (Disabled) | **Note**: This attribute is currently only valid for Ethernet/IP devices. See below (DHCP Option 61, Client Identifier) |
| 23 | DHCP Option 61 generic string | Get | Array of UINT8 | N/A | **Note**: This attribute is currently only valid for Ethernet/IP devices. See below (DHCP Option 61, Client Identifier) |
| 24 | Enable DHCP Client | Get | BOOL | 1 = True | **Note**: This attribute is currently valid for Ethernet/IP and PROFINET devices. Enable/disable DHCP Client functionality 0:       DHCP Client functionality is disabled 1:       DHCP Client functionality is enabled |
| 25 | Enable WebDAV Server | Get | BOOL | 1 = True | **Note**: This attribute is currently valid for devices with IIoT Secure functionality. Enable/disable WebDAV server 0:       WebDAV functionality is disabled 1:       WebDAV functionality is enabled |

## Network Status

This attribute holds a bit field which indicates the overall network status as follows:

| Bit | Contents | Description | Comment |
|-----|----------|-------------|---------|
| 0 | Link | Current global link status 1= Link sensed 0= No link | EtherCAT only: This link status indicates whether the Anybus CompactCom is able to communicat using Ethernet over EtherCAT (EoE) or not. That is, it indicates the status of the logical EoE port link and is not related to the link status on the physical EtherCAT ports. |
| 1 | IP established | 1 = IP address established 0 = IP address not established | |
| 2 | (reserved) | (mask off and ignore) | |
| 3 | Link port 1 | Current link status for port 1 1 = Link sensed 0 = No link | EtherCAT only: This link status indicates whether the Anybus CompactCom is able to communicat using Ethernet over EtherCAT (EoE) or not. That is, it indicates the status of the logical EoE port link and is not related to the link status on the physical EtherCAT ports. |

| Bit | Contents | Description | Comment |
|-----|----------|-------------|---------|
| 4 | Link port 2 | Current link status for port 2<br>1 = Link sensed<br>0 = No link | Not used for EtherCAT |
| 5... 15 | (reserved) | (mask off and ignore) | |

## DHCP Option 61 (Client Identifier)

(i)   *Only valid for EtherNet/IP devices*

The DHCP Option 61 (Client Identifier) allow the end-user to specify a unique identifier, which has to be unique within the DHCP domain.

Attribute #22 (DHCP Option 61 source) is used to configure the source of the Client Identifier. The table below shows the definition for the Client identifier for different sources and their description.

| Value | Source | Description |
|-------|--------|-------------|
| 0 | Disable | The DHCP Option 61 is disabled. This is the default value if the attribute is not implemented in the application. |
| 1 | MACID | The MACID will be used as the Client Identifier |
| 2 | Host Name | The configured Host Name will be used as the Client Identifier |
| 3 | Generic String | Attribute #23 will be used as the Client Identifier |

Attribute #23 (DHCP Option 61 generic string) is used to set the Client Identifer when Attribute #22 has been set to 3 (Generic String). Attribute #23 contains the Type field and Client Identifier and shall comply with the definitions in RFC 2132. The allowed max length that can be passed to the module via attribute #23 is 64 octets.

Example:

If Attribute #22 has been set to 3 (Generic String) and Attribute #23 contains 0x01, 0x00, 0x30, 0x11, 0x33, 0x44, 0x55, the Client Identifier will be represented as an Ethernet Media Type with MACID 00:30:11:33:44:55.

Example 2:

If Attribute #22 has been set to 2 (Host Name) Attribute #23 will be ignored and the Client Identifier will be the same as the configured Host Name.

# A        Categorization of Functionality

The objects, including attributes and services, of the Anybus CompactCom and the application
are divided into two categories: basic and extended.

## A.1       Basic

This category includes objects, attributes and services that are mandatory to implement or to use.
They will be enough for starting up the Anybus CompactCom and sending/receiving data with
the chosen network protocol. The basic functions of the industrial network are used.

Additional objects etc, that will make it possible to certify the product also belong to this
category.

## A.2       Extended

Use of the objects in this category extends the functionality of the application. Access is given to
the more specific characteristics of the industrial network, not only the basic moving of data to
and from the network. Extra value is given to the application.

Some of the functionality offered may be specialized and/or seldom used. As most of the
available network functionality is enabled and accessible, access to the specification of the
industrial network may be required.

# B Anybus Implementation Details

## B.1 SUP-Bit Definition

The supervised bit (SUP) indicates that the network participation is supervised by another network device. In the case of PROFINET, this bit is set when one or more IO connections are established.

## B.2 Anybus State Machine

The table below describes how the Anybus State Machine relates to the PROFINET network.

| Anybus State | Implementation | Comment |
| --- | --- | --- |
| WAIT_PROCESS | The Anybus stays in this state until an IO connection with an IO controller is opened. | - |
| ERROR | Configuration data mismatch, initial parameter error, or error reported from application status register. | See *Application Status Register, p. 144* for more information. |
| PROCESS_ACTIVE | IO connection established with IO controller and valid output data has been provided at least once. | - |
| IDLE | IO controller with which an IO connection is established is in STOP mode, the IO controller has not provided valid output at least once, or error reported from application status register.. | • Some IO controllers will not provide valid output data in the first cycles following a successful connection.<br>• See *Application Status Register, p. 144* for more information. |
| EXCEPTION | Turn module status LED red, to indicate major fault, turn network status LED off, and hold Ethernet MAC in reset. | Some kind of unexpected behavior, for example watchdog timeout. See also instance 1, attribute 7 in*Network Object (03h), p. 50*. |

## B.3 Application Status Register

The application status register is primarily used in SYNC applications. It is used in applications where the network in question supports the ability to indicate critical process data errors to the master. If the application sets an error status to the application status register, the module cannot operate in the PROCESS_ACTIVE state. The Anybus CompactCom will accept and handle the below listed status codes written by the application.

| Value | Error | Description | Anybus CompactCom 40 PROFINET IRT Response |
| --- | --- | --- | --- |
| 0000h | No error | Ready for transition to state PROCESS_ACTIVE (Default) | If Isochronous mode is active, the Anybus CompactCom monitors write process data updates according to the table below (Input time failure). |
| 0001h | Not yet synchronized | Not ready for transition to state PROCESS_ACTIVE | Alarm: Application not synchronized<br>• The Anybus CompactCom is set to state IDLE<br>• IOxS is set to BAD. |
| 0002h | Sync configuration error | A problem with the current attribute values in the Sync object prevents the transition to state PROCESS_ACTIVE | Alarm: Sync configuration error<br>• The Anybus CompactCom is set to state IDLE<br>• IOxS is set to BAD. |
| 0003h | Read process data configuration error | A problem with the current read process data mapping prevents the transition to state PROCESS_ ACTIVE | Alarm: Output configuration error<br>• The Anybus CompactCom is set to state IDLE<br>• IOxS is set to BAD. |
| 0004h | Write process data | A problem with the current write process data mapping | Alarm: Input configuration error<br>• The Anybus CompactCom is set to state IDLE |

| Value | Error | Description | Anybus CompactCom 40 PROFINET IRT Response |
|-------|-------|-------------|--------------------------------------------|
| | configuration error | prevents the transition to state PROCESS_ACTIVE | • IOxS is set to BAD. |
| 0005h | Synchroniza-tion loss | The application has lost synchronization lock<br>If the Anybus CompactCom is in the state PROCESS_ACTIVE, it will change to the state ERROR | Alarm: Application not synchronized<br>• The Anybus CompactCom is set to state ERROR<br>• IOCS is set to BAD. |
| 0006h | Excessive data loss | The application has detected a significant loss of process data from the network<br>If the Anybus CompactCom is in the state PROCESS_ACTIVE, it will change to the state ERROR | |
| 0007h | Output error | Application malfunction<br>If the Anybus CompactCom is in the state PROCESS_ACTIVE, it will change to the state ERROR | Alarm: Output disabled<br>• The Anybus CompactCom is set to state ERROR<br>• IOxS is set to BAD. |

The Anybus CompactCom produces synchronous mode related diagnosis as shown in the table below, where the cause is reported from the Application status register.

| Cause | ChannelErrorType | ExtChannelErrorType | Description |
|-------|------------------|---------------------|-------------|
| 0007h | 0018h | - | Output disabled |
| 0001h or 0005h or 0006h | 7000h (vendor specific – "Isochronous mode error") | 0001h | Application not synchronized |
| 0002h | | 0002h | Sync configuration error |
| 0003h | | 0003h | Output configuration error |
| 0004h | | 0004h | Input configuration error |
| 0000h | 8004h (Isochronous mode mismatch) | 8001h | Input time failure is reported if the application status register reports 0000h and IsochronousMode is active and no new write process data update occurs before input processing. |

For GSDML entries, see the SYNC example GSDML file that is included in the configuration files package, that can be downloaded from the Anybus CompactCom 40 PROFINET IRT page at www.anybus.com/support.

## B.4 Application Watchdog Timeout Handling

Upon detection of an application watchdog timeout, the module will cease network participation and shift to the state EXCEPTION. No other network specific actions are performed.

# C Flowcharts

## C.1 Flowchart — Record Data Access

This flowchart illustrates how Record Data requests are handled by the Anybus module.



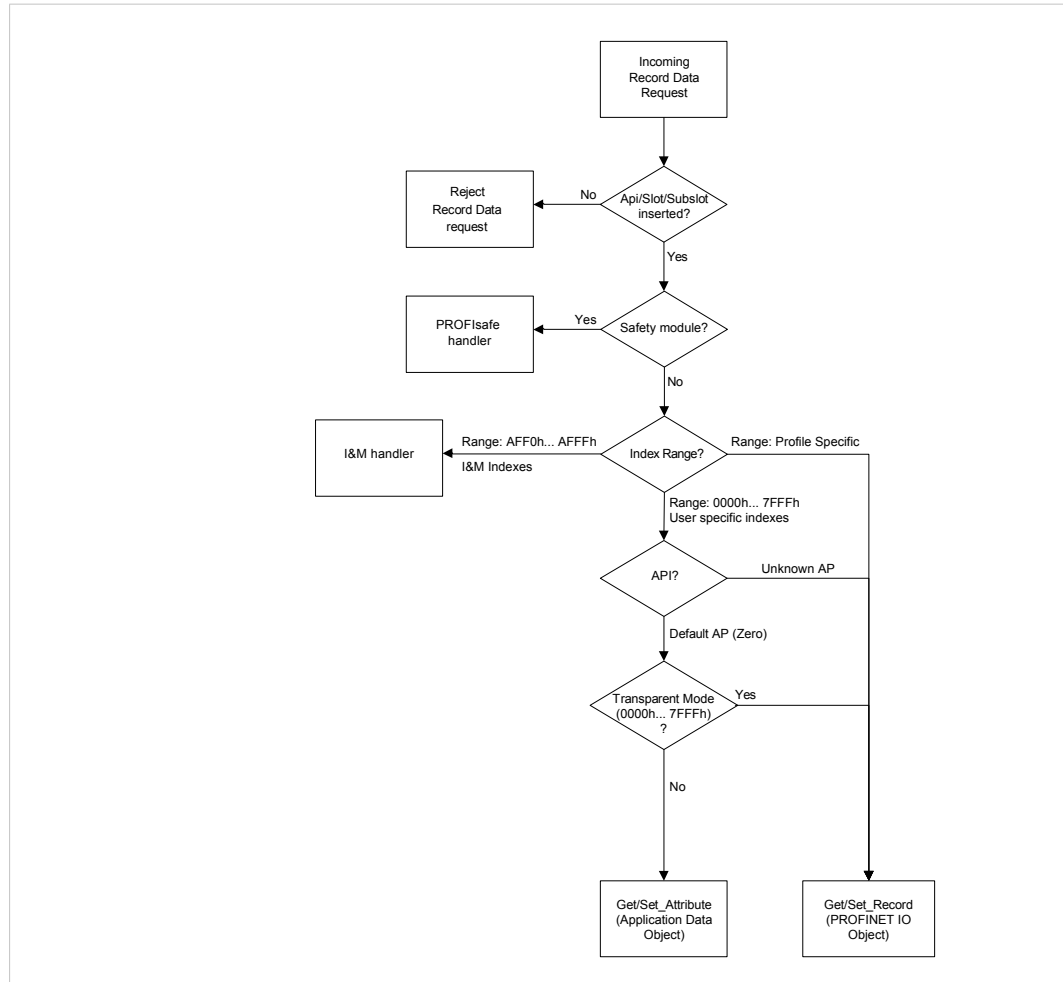**Fig. 13**

See also...

- *Application Data Instances (ADIs), p. 15*

- *PROFINET IO Object (F6h), p. 118*

- Details for command Get_Record in the *PROFINET IO Object (F6h), p. 118*

- Details for command Set_Record in the *PROFINET IO Object (F6h), p. 118*

## C.2 Flowchart — I&M Record Data Handling

This flowchart illustrates how I&M Record Data requests are handled by the Anybus module.
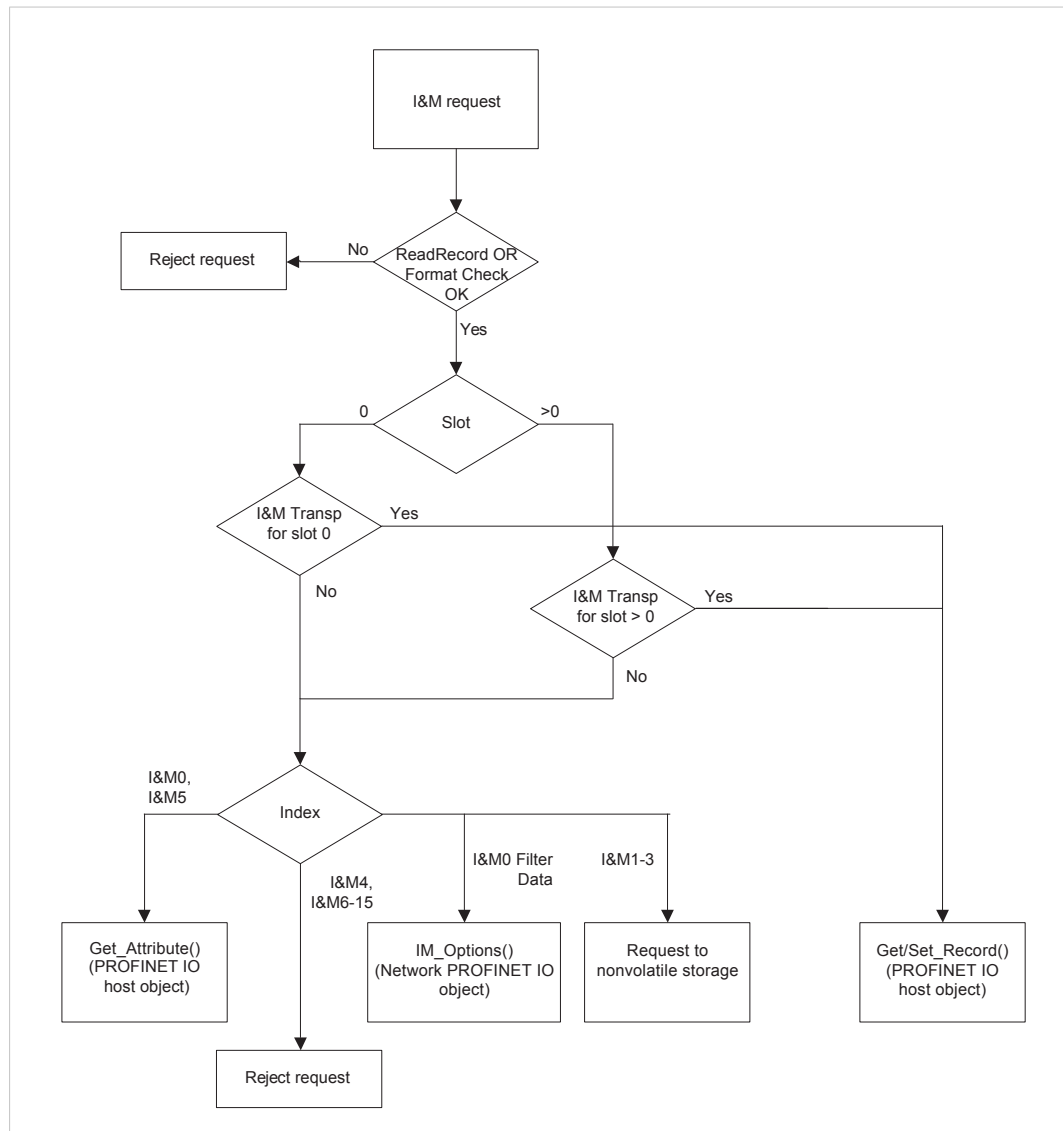


**Fig. 14**

- *PROFINET IO Object (F6h), p. 118*

- Details for command Get_Record in the *PROFINET IO Object (F6h), p. 118*

- Details for command Set_Record in the *PROFINET IO Object (F6h), p. 118*

## C.3 Flowchart —Establishment of Real Identification (RI)

This flowchart illustrates the establishment of the Real Identification.
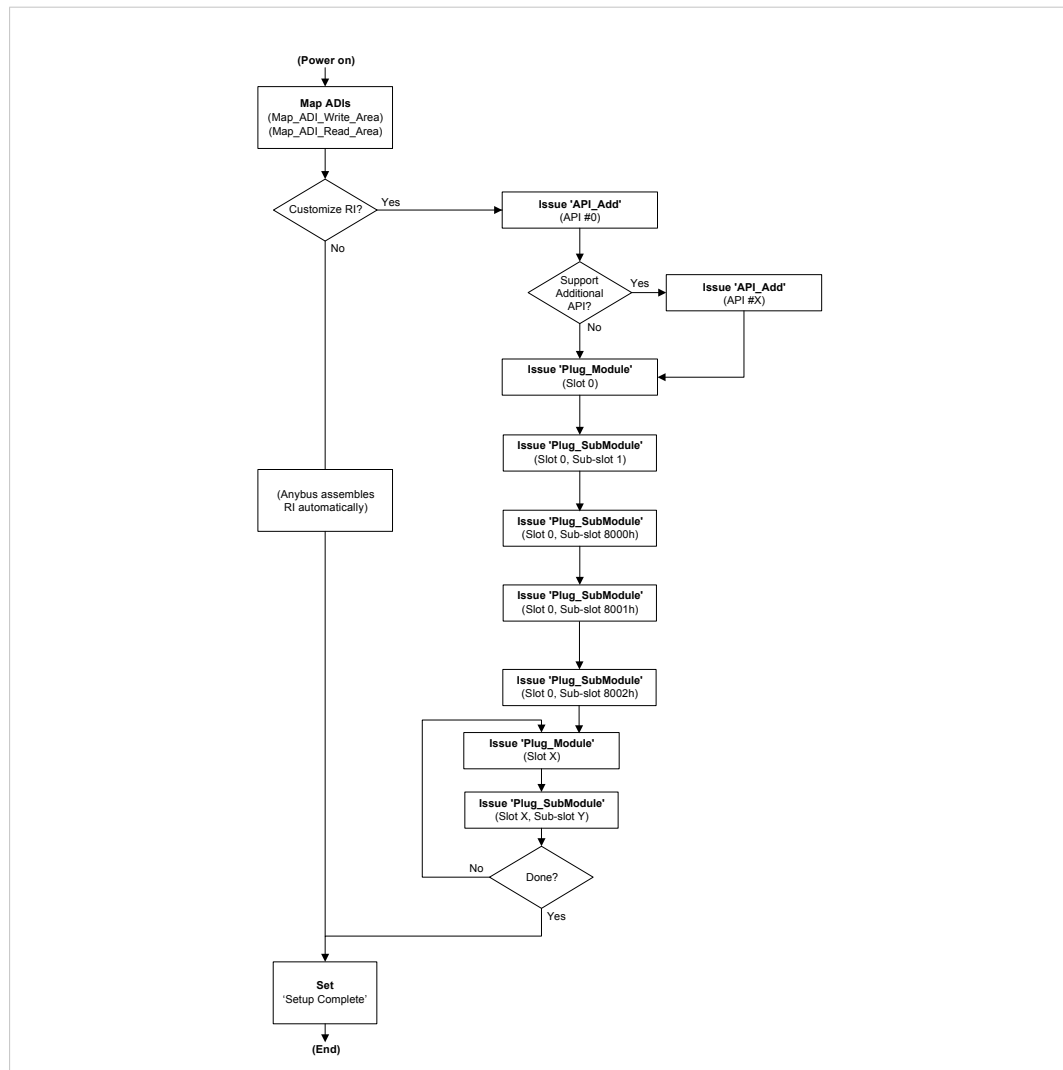


**Fig. 15**

- *Process Data, p. 16*

- *Real Identification (RI), p. 20*

- Details for command Set_Record in the *PROFINET IO Object (F6h), p. 118*

## C.4        Flowcharts — Handling of Configuration Mismatch

### C.4.1      Default Configuration Mismatch

This flowchart shows how the Anybus CompactCom automatically handles a configuration mismatch when the Real Identification has been established by the default configuration method.



**Fig. 16**

## C.4.2      Custom Configuration mismatch

This flowchart shows how to handle a configuration mismatch when the Real Identification has been established by the host application (custom configuration).
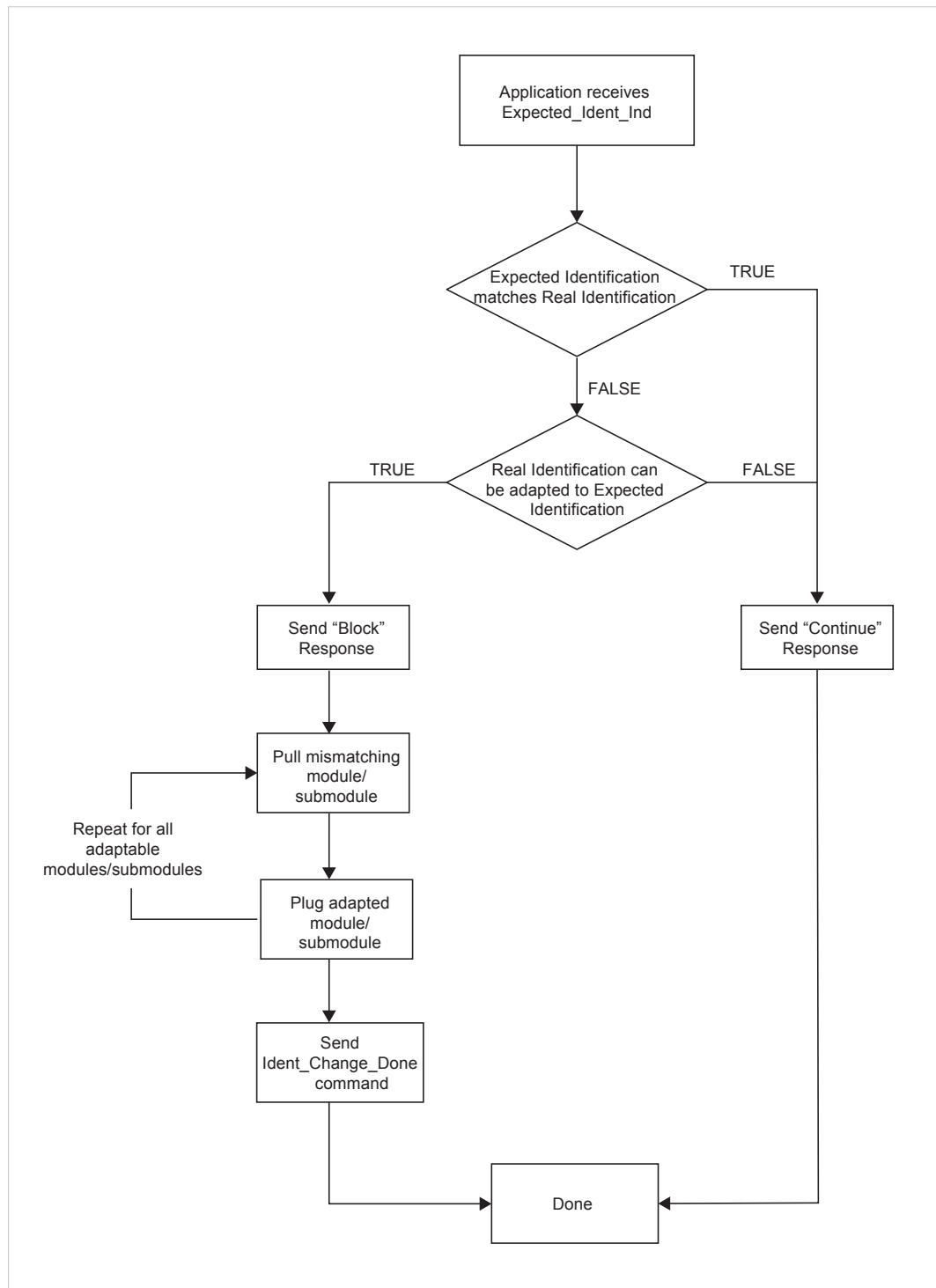


**Fig. 17**

# D Secure HICP (Secure Host IP Configuration Protocol)

## D.1 General

The Anybus CompactCom 40 PROFINET IRT supports the Secure HICP protocol used by the Anybus IPconfig utility for changing settings, e.g. IP address, Subnet mask, and 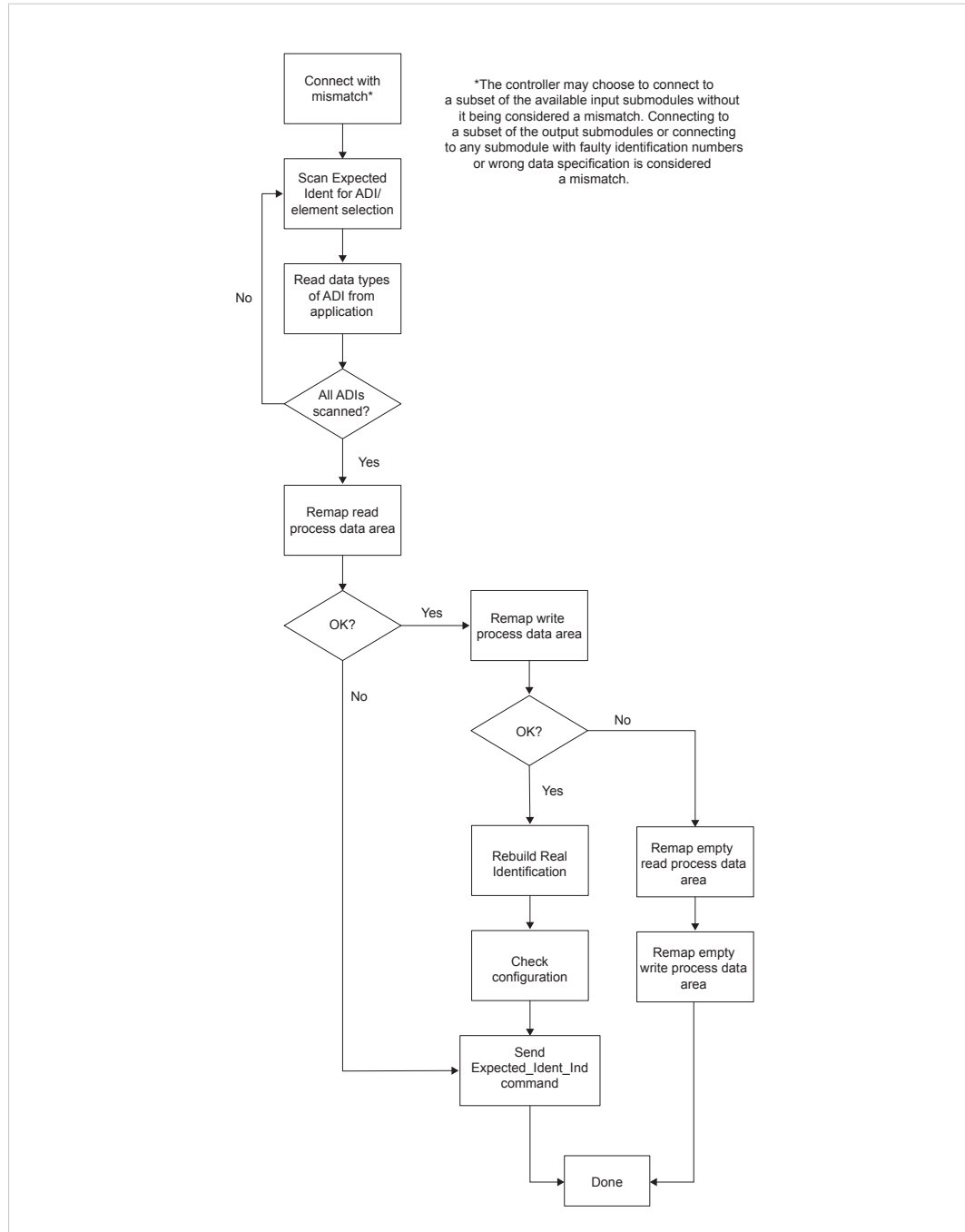enable/disable DHCP. The protocol offers secure authentication. Anybus IPconfig can be downloaded free of charge from the HMS website, www.anybus.com. This utility may be used to access the network settings of any Anybus product connected to the network via UDP port 3250.

## D.2 Operation

When the application is started, the network is automatically scanned for Anybus products. The network can be rescanned at any time by clicking **Scan**.

To alter the network settings of a module, click on its entry in the list. The settings for the module will appear to the right.



**Fig. 18**

Click **Apply** to send and apply the settings. Settings are saved in non-volatile memory in the device. Optionally, the configuration can be protected from unauthorized access by a password.

# E        Technical Specification

## E.1      Front View

### E.1.1        Front View (PROFINET IRT, Ethernet Connectors)

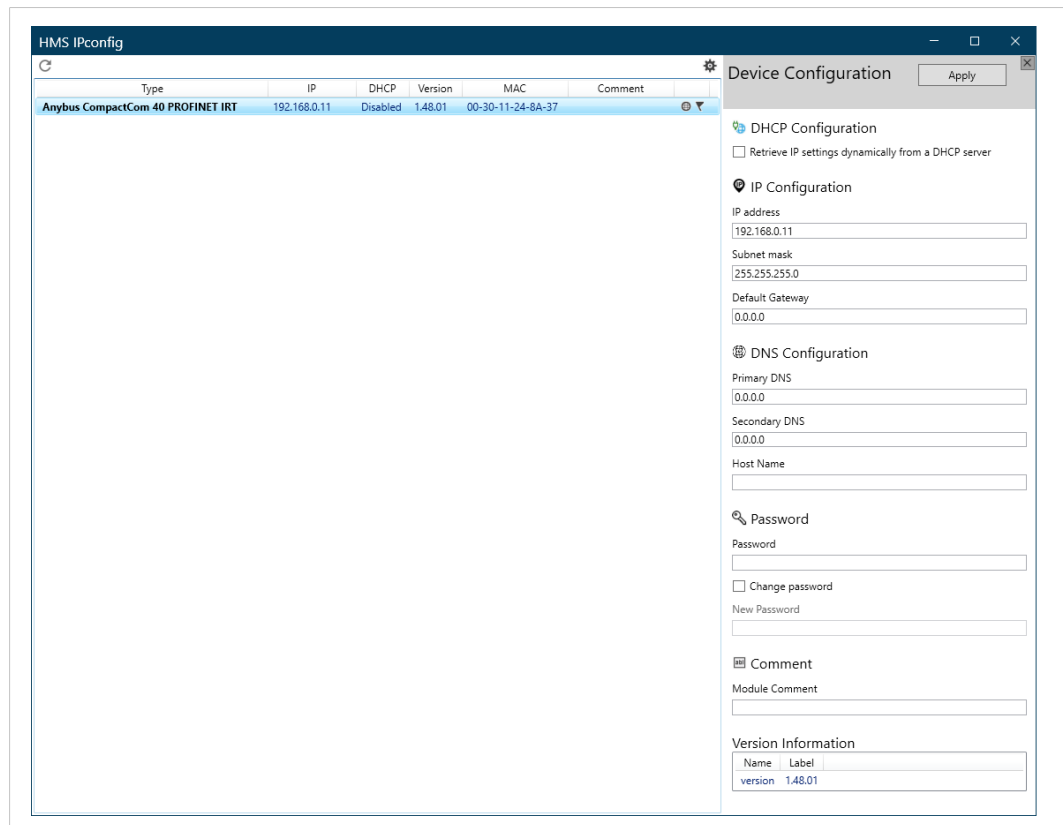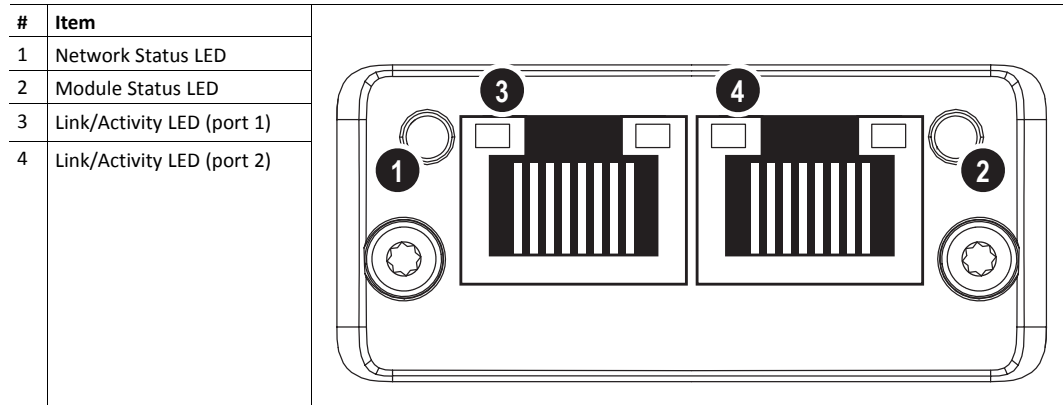| # | Item |
|---|------|
| 1 | Network Status LED |
| 2 | Module Status LED |
| 3 | Link/Activity LED (port 1) |
| 4 | Link/Activity LED (port 2) |

Test sequences are performed on the Network and Module Status LEDs during startup.

### E.1.2        Front View (Fiber Optics Connectors)

| # | Item |
|---|------|
| 1 | Network Status LED |
| 2 | Module Status LED |
| 3 | Optical signal from the Anybus CompactCom module |
| 4 | Optical signal to the Anybus CompactCom module |
| 5 | Link/Activity LED (port 1) |
| 6 | Link/Activity LED (port 2) |
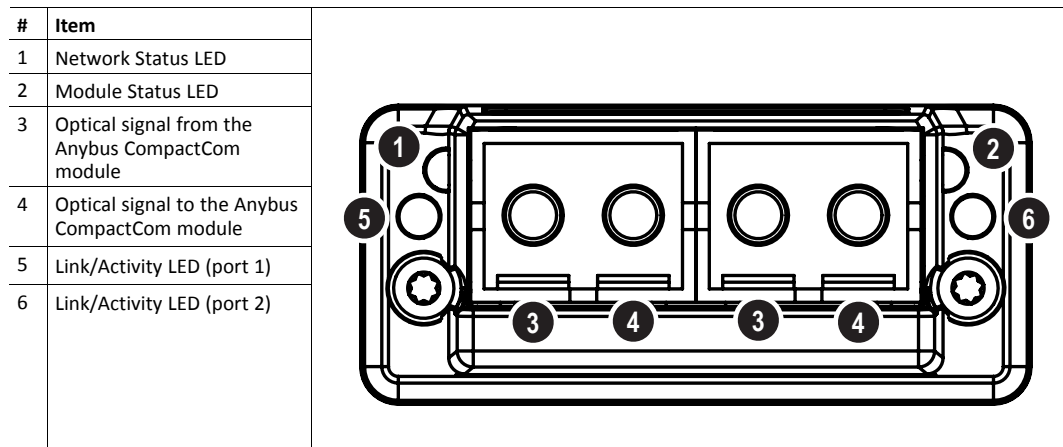
Test sequences are performed on the Network and Module Status LEDs during startup.

### E.1.3        Network Status LED

| LED State | Description | Comments |
|-----------|-------------|----------|
| Off | Offline | • No power<br>• No connection with IO Controller |
| Green | Online (RUN) | • Connection with IO Controller established<br>• IO Controller in RUN state |
| Green, 1 flash | Online (STOP) | • Connection with IO Controller established<br>• IO Controller in STOP state or IO data bad<br>• IRT synchronization not finished |
| Green, blinking | Blink | Used by engineering tools to identify the node on the network |
| Red | Fatal event | Major internal error (this indication is combined with a red module status LED) |

| LED State | Description | Comments |
|---|---|---|
| Red, 1 flash | Station Name error | Station Name not set |
| Red, 2 flashes | IP address error | IP address not set |
| Red, 3 flashes | Configuration error | Expected Identification differs from Real Identification |

## E.1.4     Module Status LED
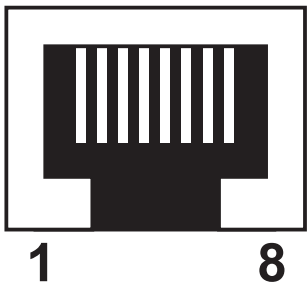
| LED State | Description | Comments |
|---|---|---|
| Off | Not Initialized | No power OR Module in SETUP or NW_INIT state. |
| Green | Normal Operation | Module has shifted from the NW_INIT state. |
| Green, 1 flash | Diagnostic Event(s) | Diagnostic event(s) present |
| Red | Exception error | Device in state EXCEPTION. |
|  | Fatal event | Major internal error (this indication is combined with a red network status LED) |
| Alternating Red/Greed | Firmware update | Do NOT power off the module. Turning the module off during this phase could cause permanent damage. |

## E.1.5     LINK/Activity LED

| LED State | Description | Comments |
|---|---|---|
| Off | No Link | No link, no communication present |
| Green | Link | Ethernet link established, no communication present |
| Green, flickering | Activity | Ethernet link established, communication present |

### E.1.6 Ethernet Interface (RJ45 connectors)

The Ethernet interface operates at 100 Mbit, full duplex, as required by PROFINET.

| Pin no | Description |
|--------|-------------|
| 4,5,7,8 | Connected to chassis ground over serial RC circuit |
| 6 | RD- |
| 3 | RD+ |
| 2 | TD- |
| 1 | TD+ |
| Housing | Cable Shield |

For information on how to connect the PROFINET cable, see *Functional Earth (FE) Requirements, p. 154*

## E.2 Functional Earth (FE) Requirements

In order to ensure proper EMC behavior, the module must be properly connected to functional earth via the FE pad/FE mechanism described in the *Anybus CompactCom 40 Hardware Design Guide*. Proper EMC behavior is not guaranteed unless these FE requirements are fulfilled.

> ℹ️ *The shield of the RJ45 connector is not connected directly to FE. As all nodes in a PROFINET network have to share chassis ground connection, the PROFINET cable shield has to be connected to the chassis ground at each node in the network.*

For further information, see *PROFINET Installation Guideline for Cabling and Assembly*, available for download at www.profinet.com.

## E.3 Power Supply

### E.3.1 Supply Voltage

The Anybus CompactCom 40 PROFINET IRT requires a regulated 3.3 V power source as specified in the general *Anybus CompactCom 40 Hardware Design Guide*.

### E.3.2 Power Consumption

TheAnybus CompactCom 40 PROFINET IRT is designed to fulfil the requirements of a Class B module. The current hardware design consumes up to 390 mA

TheAnybus CompactCom 40 PROFINET IRT FO is designed to fulfil the requirements of a Class C module. The current hardware design consumes up to 740 mA

In line with HMS policy of continuous product development, we reserve the right to change the exact power requirements of this product without prior notification. However, in any case, the Anybus CompactCom 40 PROFINET IO will remain as a Class B module.

## E.4 Environmental Specification

Consult the *Anybus CompactCom 40 Hardware Design Guide* for further information.

## E.5          EMC Compliance

Consult the *Anybus CompactCom 40 Hardware Design Guide* for further information.


## E.6          Fiber Optics Compliance (MAU type Compliance)

The optical interface of theAnybus CompactCom 40 PROFINET IRT is compliant with the non IEEE802.3 MAU type POF.

The supported cables are stated in the PI document *PROFINET Cabling and Interconnection Technology*;

for (SI-POF/PCF):

Physical Layer Medium Dependent Sublayer on 650 nm Fiber Optics

PROFIBUS & PROFINET International Order No: 2.432

# F        AIDA LED Mode

The German car manufacturers have through the Automation Initiative of German Domestic Automobile Manufacturers (AIDA) defined an AIDA specific LED behavior for PROFINET devices. It is necessary to implement this behavior, to pass conformance tests for AIDA. Any application using the Anybus CompactCom 40 PROFINET IRT can be designed to conform to the AIDA LED behavior standard.

## F.1      How to Design for AIDA LED Mode

The default version of the Anybus CompactCom 40 PROFINET IRT has one LED for Network Status, one LED for Module Status and one LED per port to indicate both link and activity. In the AIDA LED indication mode, one LED indicates Bus Failure and one LED indicates System Failure, but link and activity for each port are shown separately on different LEDs.

The front of the Anybus CompactCom 40 PROFINET IRT does not carry enough LEDs to show both the link and the activity indications that are demanded in the AIDA LED mode, but the signals are available on the host application connector.

To implement the AIDA LED Mode in an Anybus CompactCom 40 PROFINET IRT, do the following:

- Select LED indication mode by setting attribute #7 in the Application Object (FFh) to 1 (AIDA LED mode). This will make the link and the activity signals separately available in the host application connector.

- In your hardware design, connect orange LEDs to the appropriate pins of the host application connector, to indicate activity for each port.

> **!** The Port Activity and Port Link signals are available in the host application connector in all operating modes except 16 bit parallel. The signals are always available in the LED status register, see Anybus CompactCom 40 Software Design Guide for more information.
>
> When Transparent Ethernet is activated, the LED status signals are only available in the LED status register.

## F.2      Anybus CompactCom AIDA LED Mode

### F.2.1     Front View

| # | Default Mode (see *Front View, p. 152*) | AIDA Mode (see description below) |
|---|---|---|
| 1 | Network Status LED | Bus Failure LED |
| 2 | Module Status LED | System Failure LED |
| 3 | Link/Activity LED (port 1) | Link (port 1) |
| 4 | Link/Activity LED (port 2) | Link (port 2) |



### F.2.2     Bus Failure LED

| LED State | Description | Comments |
|---|---|---|
| Off | Ok | No problem |
| Red | Fatal event | Major internal error (this indication is combined with a red system failure LED) |
| | Station Name error | Station Name not set |
| | IP address error | IP address not set |

| LED State | Description | Comments |
|---|---|---|
|  | Configuration error | Expected Identification differs from Real Identification |
|  | Online (STOP) | Connection with IO Controller established<br><br>• IO Controller in STOP state or IO data BAD<br><br>• IRT synchronization not finished |
|  | Connection error | No connection with IO Controller |

### F.2.3 System Failure LED

| LED State | Description | Comments |
|---|---|---|
| Off | Not Initialized | No problems detected |
| Red | Exception error | Device in state EXCEPTION. |
|  | FATAL event | Major internal error (this indication is combined with a red bus failure LED) |
|  | Diagnostic Event(s) | Diagnostic event(s) present |

### F.2.4 Port Link LED

| LED State | Description | Comments |
|---|---|---|
| Off | No Link | No link, no communication present |
| Green | Link | Ethernet link established |
| Green, blinking | DCP Identify | Used by engineering tools to identify the node on the network 500 ms on, 500 ms off during 3 seconds |

The signal is also available in the host application connector:

| AIDA indication | Signal Name, Application Interface Connector |
|---|---|
| Link on port 1 | LED3A |
| Link on port 2 | LED4A |

The Port Link signals are always available in the host application connector in all operating modes except 16 bit parallel. For more information consult the Anybus CompactCom M40 Hardware Design Guide (module) or the Anybus CompactCom B40-1 Design Guide (brick).

When Transparent Ethernet is activated, the LED status signals are only available in the LED status register.

### F.2.5 Port Activity

| AIDA LED State | Description |
|---|---|
| Off | No Activity |
| Orange, flashing | Activity |

There is no LED on the front of the Anybus CompactCom 40 PROFINET IRT showing this indication, but the signal is available in the host application connector according to this table:

| AIDA indication | Signal Name, Host Application Connector |
|---|---|
| Activity on port 1 | LED3B |
| Activity on port 2 | LED4B |

The Port Activity signals are always available in the host application connector in all operating modes except 16 bit parallel. For more information consult the Anybus CompactCom M40 Hardware Design Guide (module) or the Anybus CompactCom B40-1 Design Guide (brick).

When Transparent Ethernet is activated, the LED status signals are only available in the LED status register.

# G        Conformance Test Guide

## G.1        General

When using the default settings of all parameters, the Anybus CompactCom 40 PROFINET IRT is precertified for network compliance. This precertification is done to ensure that your product *can* be certified.

Changes in the parameters in the example GSD file, supplied by HMS Industrial Networks, will require a certification. A vendor ID can be obtained from PNO and is compulsory for certification. This chapter provides a guide for successful conformance testing your product, containing the Anybus CompactCom 40 PROFINET IRT, to comply with the demands for network certification set by the PNO.

Independent of selected operation mode, the actions described in this appendix have to be accounted for in the certification process. The identity of the product needs to be changed to match your company and device.

For multiple-API implementations, e.g. PROFIdrive, one of the submodules belonging to the "non-zero API" must be inserted in the I&M0 Filter Data by the application, using the command IM_Options.

> **!**    *This appendix provides guidelines and examples of what is needed for certification. Depending on the functionality of your application, there may be additional steps to take.*

Please contact HMS Industrial Networks at www.anybus.com/support for more information.

## G.2       Reidentifying Your Product

After successful setting of the **Setup Complete** attribute in the Anybus Object (01h), the Anybus module asks for identification data from the host PROFINET IO Object (F6h). Therefore, the attributes listed below shall be implemented and proper values returned.

| Object/ Instance | Attribute | Explanation | Default | Customer sample | Comment |
|---|---|---|---|---|---|
| PROFINET IO Object (F6h), Instance 1 | #1,Device ID | With this attribute you set the Device ID of the device | Device ID: 0010h | Device ID: YYYYh | This information must match the keys of the "DeviceIdentity" of the GSD-file. Note that the GSD file keyword "VendorName" must correspond to the Vendor ID value. |
| PROFINET IO Object (F6h), Instance 1 | #2,Vendor ID | With this attribute you set the Vendor ID of the device | Vendor ID: 010Ch (HMS) | Vendor ID: XXXXh | |
| PROFINET IO Object (F6h), Instance 1 | #3,Station Type | With this attribute you set the station type of the device | "CompactCom 40 PIR" | "Cust-PNIO-Dev" | This information matches, in the case of Anybus CompactCom 40 PROFINET IRT, GSD keywords "DNS_CompatibleName" and "OrderNumber". The Station Type must be equal to the "DNS_CompatibleName", but it is allowed to have a completely different "OrderNumber", see also I&M Order ID below. |
| PROFINET IO Object (F6h), Instance 1 | #8, I&M Order ID | With this attribute you set the Order ID that is used in the I&M data | "CompactCom 40 PIR" | "Cust-PNIO-Dev" | This information must match the keys of "theOrderNumber" of the GSD-file. |
| PROFINET IO Object (F6h), Instance 1 | #10, I&M Hardware Revision | With this attribute you set the I&M Hardware Revision | (Hardware Rev.) | "0002h" | Optional. This information must match the keys of the "HardwareRelease" of the GSD-file. |
| PROFINET IO Object (F6h), Instance 1 | #11, I&M Software Revision | With this attribute you set the I&M Software Revision | (Software Rev.) | "V2.5.3" | Optional. This information must match the keys of the "SoftwareRelease" of the GSD-file. |
| PROFINET IO Object (F6h), Instance 1 | #13, I&M Profile ID | With this attribute you set the I&M Profile ID | 0xF600 (Generic device) | API number of profile. | Optional, Only relevant for multi-API implementations like PROFIdrive. |
| PROFINET IO Object (F6h), Instance 1 | #14, I&M Profile Specific type | With this attribute you set the I&M Profile Specific typ | 0x0004 (Communica-tion Module) | Profile specific type, defined by profile. | Optional, Only relevant for multi-API implementations like PROFIdrive. |

### G.2.1 Additional GSD File Information

The GSD file keyword "ProductFamily" shall correspond to the vendor's name of the device.

The GSD file keyword "MainFamily" lists the kinds of devices for which the product shall be listed. As of GSD specification v2.3, the following "families" are available:

"General", "Drives", "Switching Devices", "I/O", "Valves", "Controllers", "HMI","Encoders", "NC/RC", "Gateway", "PLCs", "Ident Systems", "PA Profiles", "Network Components", "Sensors".

## G.3 Factory Default Reset

Reset command to Application Object (FFh) must be supported

When PROFINET IO modules are delivered, they are required to be in their "Factory Default" state. For PROFINET devices this means that their Station Name is empty ("""") and that the IP suite is not assigned (IP 0.0.0.0). When a Factory Default Reset command is received from the network, the Anybus module will erase all IP and Station Name information and inform the host application that a hardware reset of the Anybus module is required. This is done by sending a Reset command to the Application Object (FFh) of the host. For more details, please consult the *Anybus CompactCom Software Design Guide*.

## G.4 IP Address

Normally the IP numbers of PROFINET IO devices are assigned via the PROFINET network via DCP (Discovery and Configuration Protocol). HMS Industrial Networks recommends not using the Network Configuration Object (04h, instances #3 - #6) during the initialization phase for PROFINET modules, unless the end user has requested the IP address to be set to a specific value (by for example using a keypad). The reason is that when a factory default reset command is received from the PROFINET network (via DCP) the node must be available after a hardware reset with the default IP-address (0.0.0.0).

If the IP suite is set using the Network Configuration Object (04h), the key word AddressAssignment = "DCP;LOCAL" must be added to the GSD file in the section DeviceAccessPointItem:

```
<DeviceAccessPointItem AddressAssignment="DCP;LOCAL"...>
```

## G.5 Station Name

Normally the Station Name of a PROFINET device is assigned by the end user via the PROFINET DCP protocol. HMS recommends not using the Station Name instance in the Network Configuration Object during the initialization phase for PROFINET modules. If this attribute is used, it is recommended that it is sent explicitly when the end user changes the Station Name with e.g. a keypad. The reason is that when a factory default reset command is received from the PROFINET network (via DCP), the node must be available after a hardware reset with the default Station Name ("""").

> (i) *The Anybus module will forward all information about the connection being established to the IO Controller, as commands to the host PROFINET IO Object (F6h). Even though the host application might not need this information, a response must always be generated (such as 05h, "Unsupported command")*

## G.6            Documentation Considerations

To obtain a certificate, the following information must be present in the customer's user manual:

1.   Behavior of the outputs if IOPS=BAD.

2.   Behavior of the outputs if connection is aborted.

3.   Behavior of the outputs at power on.

The Anybus CompactCom handles these situations in the following ways:

1.   State change to IDLE. The network is informed that the I/O data of the submodule with IOPS=BAD is substituted with zeros (clear). No read process data is updated in the host interface.

2.   State change to WAIT_PROCESS. The network is informed that the I/O data of all submodules is substituted with zeros (clear). No process data is updated in the host interface.

3.   The network is informed that the I/O data of all submodules is substituted with zeros (clear). No process data is updated in the host interface.

4.   A shift register application must use the PA signal to clear outputs when the Anybus CompactCom 40 PROFINET IRT is not in the state PROCESS_ACTIVE.

## G.7            Certification in Generic Anybus Mode

In Generic Anybus Mode (when the command API_add in the Network PROFINET IO Object (0Eh) is not used) there is normally nothing that needs to be considered apart from what is mentioned earlier in this appendix. The example HMS GSD file has to be modified with respect to the process mapping and identity of the product and this requires a certification of the product
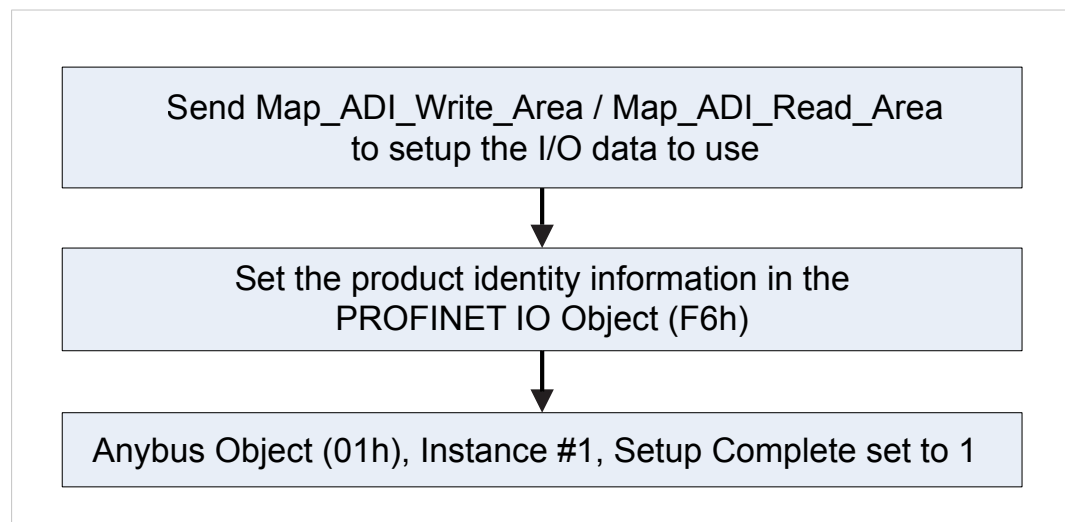


**Fig. 19**

## G.8            Certification in Advanced Mode

In advanced mode (Network PROFINET IO Object (0Eh) is used), the most important thing is to use a Device Access Point (DAP) that conform to PROFINET IO Specification v2.0 or later (DAP2). From specification version 2.0 it is possible to describe the physical Ethernet interface and its ports (PDEV, or Physical Device) with a special mechanism. This is done with special submodules at slot 0 (the module at slot 0 is the access point for the device). HMS recommends following the flow below for setting up a DAP2.
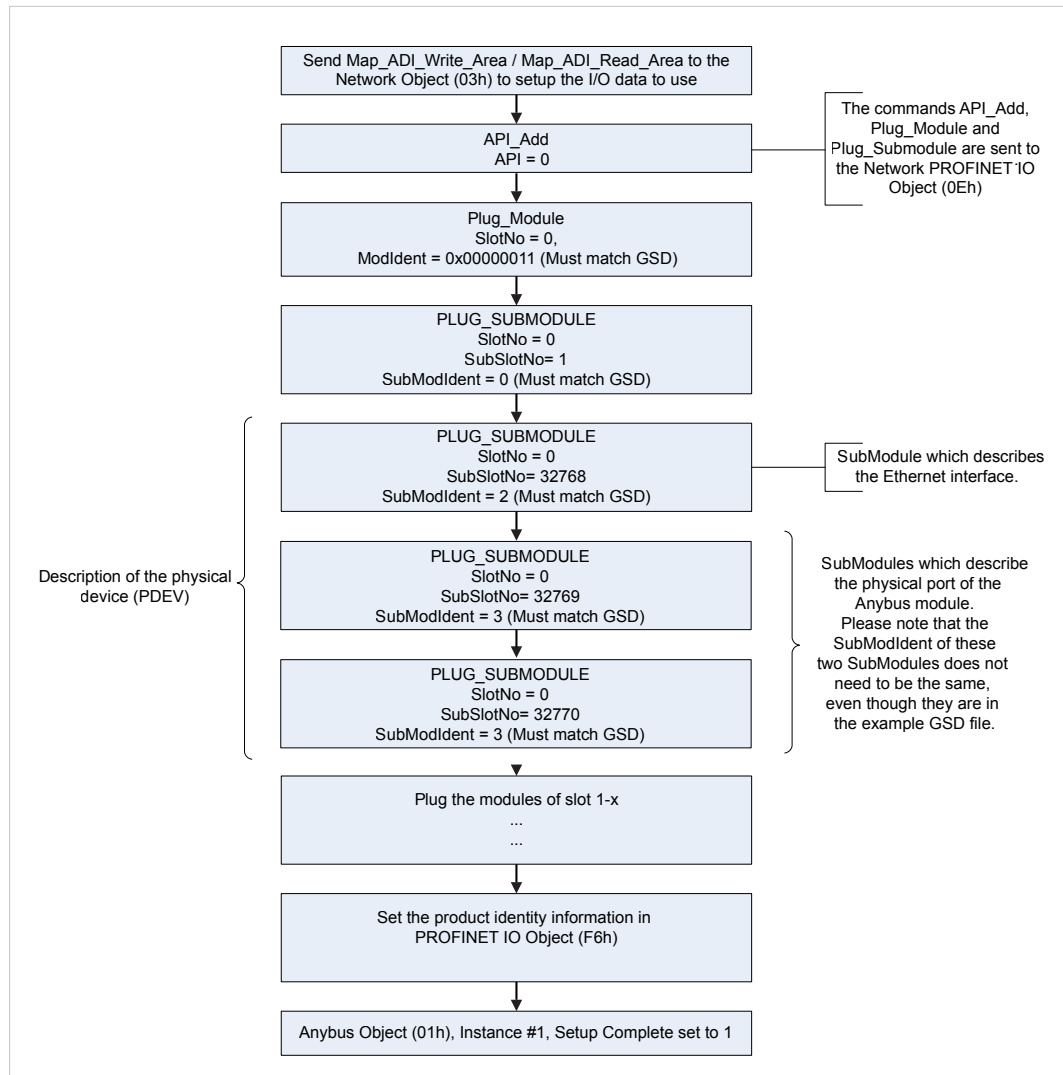
**Fig. 20**

The figure shows how to set up a PROFINET compatible DAP. Please note that for some commands only the relevant parameters are shown.

Please note that the values of "SubModIdent" in the above flowchart are the values of the example HMS GSD file. They can be changed if necessary, but there is no real need for it, the important thing is that it matches the GSD file. To be able to pass the PROFINET conformance test a "DAP2" is mandatory.

## G.9        Changes in GSD File for Conformance Class B

The example GSD file, supplied by HMS Industrial Networks, is adapted for testing a Anybus CompactCom 40 PROFINET IRT for conformance class C. If the implementation does not need the isochronous features of the device, the GSD file can be modified to mirror this. The implementation can then be conformance tested for conformance class B instead. The list below describe the changes needed in the example GSD file to accomplish this.

1.   The value of the `ConformanceClass` attribute in the `<CertificationInfo...>` element in each DAP must be changed from "C" to "B".:

     `<CertificationInfo ConformanceClass="B" ApplicationClass=""`
     `NetloadClass="III"/>`

2.   The value of the `SupportedRT_Classes` attribute in the `<InterfaceSubmoduleItem ...>` element in each DAP must be "RT_CLASS_1". I.e. "RT_CLASS_3" must be removed:

     `<InterfaceSubmoduleItem ID="Interface" SubslotNumber="32768"`

     `SubmoduleIdentNumber="0x00000002"`

     `SupportedRT_Classes="RT_CLASS_1" TextId="T_ID_INTERFACE"`

     `SupportedProtocols="SNMP;LLDP" DCP_HelloSupported="true"`

     `PTP_BoundarySupported="true" DCP_BoundarySupported="true"`
     `DelayMeasurementSupported="true">`

3.   The elements `<RT_Class3Properties ...>`, `<SynchronisationMode ...>`, and `<RT_Class3TimingProperties ...>` must be removed from each DAP

## G.10       SYNC Pin Measurements for Conformance Class C Test

For a conformance class C (IRT) test, access to the SYNC pin must be provided to the test lab.

(i)   *The Sync Object (EEh) does not have to be enabled for the SYNC pin to function.*

# H        Backward Compatibility

The Anybus CompactCom M40 series of industrial network modules have significantly better performance §d include more functionality than the modules in the Anybus CompactCom 30 series. The 40 series is backward compatible with the 30 series in that an application developed for the 30 series should be possible to use with the 40 series, without any major changes. Also it is possible to mix 30 and 40 series modules in the same application.

This appendix presents the backwards compatibility issues that have to be considered for Anybus CompactCom 40 PROFINET IRT, when designing with both series in one application, or when adapting a 30 series application for the 40 series.

## H.1        Initial Considerations

There are two options to consider when starting the work to modify a host application developed for Anybus CompactCom 30-series modules to also be compatible with the 40-series modules:

- Add support with as little work as possible i.e. reuse as much as possible of the current design.

    - This is the fastest and easiest solution but with the drawback that many of the new features available in the 40-series will not be enabled (e.g. enhanced and faster communication interfaces, larger memory areas, and faster communication protocols).

    - You have to check the hardware and software differences below to make sure the host application is compatible with the 40-series modules. Small modifications to your current design may be needed.

- Make a redesign and take advantage of all new features presented in the 40-series.

    - A new driver and host application example code are available at www.anybus.com/starterkit40 to support the new communication protocol. This driver supports both 30-series and 40-series modules.

    - You have to check the hardware differences below and make sure the host application is compatible with the 40-series modules.

> ⓘ   *This information only deals with differences between the 30-series and the 40-series.*

Link to support page: www.anybus.com/support.

## H.2        Hardware Compatibility

Anybus CompactCom is available in three hardware formats; Module, Chip, and Brick.

### H.2.1        Module

The modules in the 30-series and the 40-series share physical characteristics, like dimensions, outline, connectors, LED indicators, mounting parts etc. They are also available as modules without housing.



**Fig. 21        Anybus CompactCom M30/M40**

### H.2.2        Chip

The chip (C30/C40) versions of the Anybus CompactCom differ completely when it comes to physical dimensions.

> **!**  There is no way to migrate a chip solution from the 30-series to the 40-series without a major hardware update.

### H.2.3        Brick

The Anybus CompactCom B40-1 does not share dimensions with the Anybus CompactCom B30. The B40-1 is thus not suitable for migration. However HMS Industrial Networks has developed a separate brick version in the 40-series, that can be used for migration. This product, B40-2, shares dimensions etc. with the B30. Please contact HMS Industrial Networks for more information on the Anybus CompactCom B40-2.
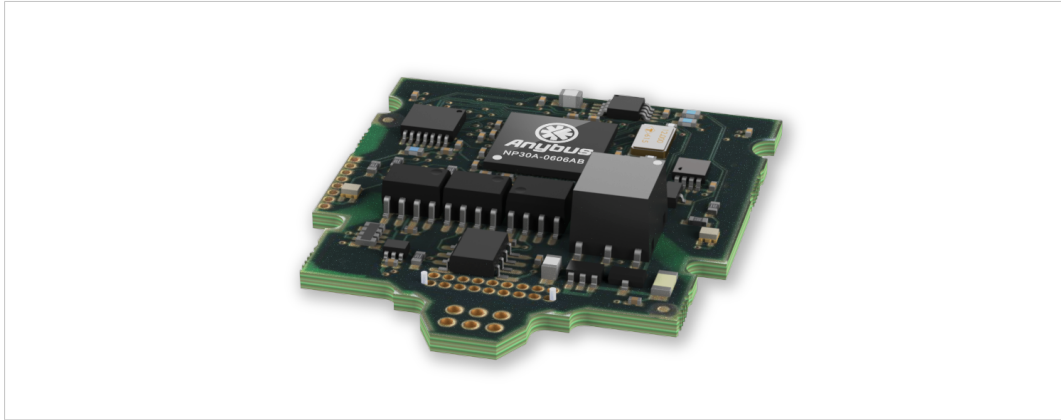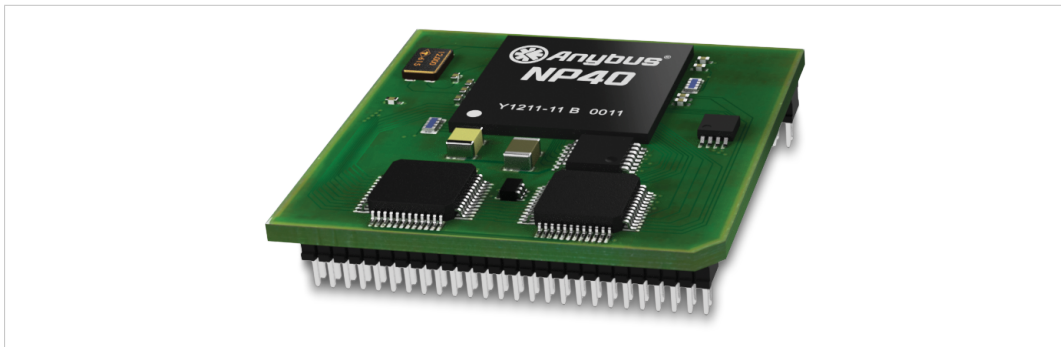


**Fig. 22        Anybus CompactCom B30**



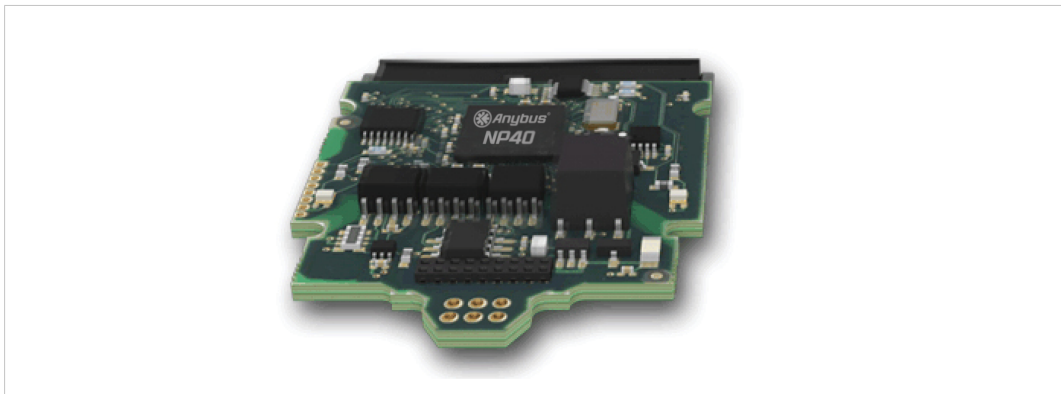**Fig. 23        Anybus CompactCom B40–1 (not for migration)**



**Fig. 24        Anybus CompactCom B40–2**

### H.2.4        Host Application Interface
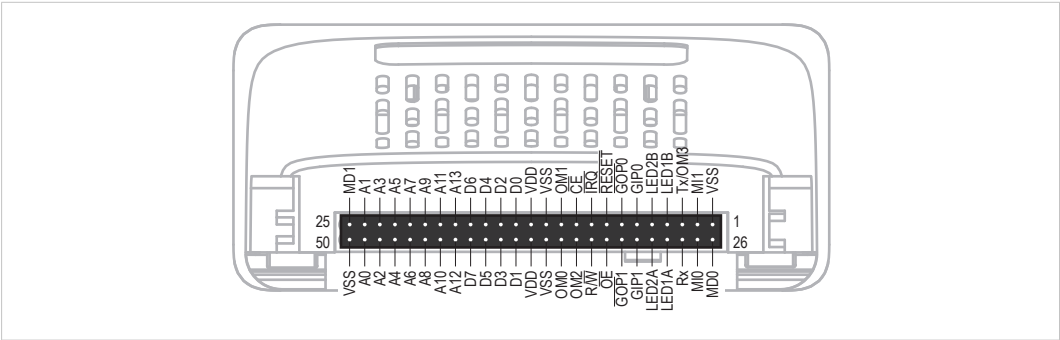


**Fig. 25**

Some signals in the host application interface have modified functionality and/or functions which must be checked for compatibility. See the following sections.

#### Tx/OM3

In the 30-series, this pin is only used for Tx. It is tri-stated during power up, and driven by the Anybus CompactCom UART after initialization. In the 40-series this pin is used as a fourth operating mode setting pin (OM3). During startup after releasing the reset, this pin is read to determine the operating mode to use. The pin is then changed to a Tx output.

In the 40-series, this pin has a built-in weak pull-up. If this pin, on a 30-series module or brick is unconnected, pulled high, or connected to a high-Z digital input on the host processor, it will be compatible with the 40-series. An external pull-up is recommended, but not required.

> **!**  If this pin is pulled low by the host during startup in a 30-series application, any 40-series module or brick, substituted in the application, will not enter the expected operating mode.

Related Information: Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126), Section "Application Connector Pin Overview"

#### Module Identification (MI[0..1])

These pins are used by the host application (i.e. your product) to identify what type of Anybus CompactCom that is mounted. The identification differs between the 30-series and the 40-series.

> **ⓘ**  *If your software use this identification you need to handle the new identification value.*

| MI1 | MI0 | Module Type |
|-----|-----|-------------|
| LOW | LOW | Active Anybus CompactCom 30 |
| HIGH | LOW | Active Anybus CompactCom 40 |

MI[0..1] shall only be sampled by the application during the time period from power up to the end of SETUP state. The pins are low at power up and before reset release.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section "Settings/Sync".

### GIP[0..1]/LED3[A..B]

These pins are tri-stated inputs by default in the 30-series. In the 40-series, these pins are tri-stated until the state NW_INIT. After that they become open-drain, active low LED outputs (LED3A/LED3B).

No modification of the hardware is needed, if your current design has

- tied these pins to GND

- pulled up the pins

- pulled down the pins

- left the pins unconnected

However, if the application drive the pins high, a short circuit will occur.

If you connect the pins to LEDs, a pull-up is required.

In the 40-series, there is a possibility to set the GIP[0..1] and GOP[0..1] in high impedance state (tri-state) by using attribute #16 (GPIO configuration) in the Anybus object (01h). I.e. if it is not possible to change the host application hardware, this attribute can be configured for high impedance state of GIP and GOP before leaving NW_INIT state.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section "LED Interface/D8-D15 (Data Bus)".

### GOP[0..1]/LED4[A..B]

These pins are outputs (high state) by default in the 30-series. In the 40-series, these pins are tri-stated until the state NW_INIT, and after that they become push-pull, active low LED outputs (LED4A/LED4B).

This change should not affect your product.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section 3.2.3, "LED Interface/D8-D15 (Data Bus)".

### Address Pins A[11..13]

The address pins 11, 12, and 13 are ignored by the 30-series. These pins must be high when accessing the 40-series module in backwards compatible 8-bit parallel mode. If you have left these pins unconnected or connected to GND, you need to make a hardware modification to tie them high.

### Max Input Signal Level ($V_{IH}$)

The max input signal level for the 30-series is specified as $V_{IH}=V_{DD}+0,2$ V, and for the 40-series as $V_{IH}=3.45$ V. Make sure that you do not exceed 3.45 V for a logic high level.

**RMII Compatibility**

If the RMII mode is being used on an Anybus CompactCom 40 module and it is desired to remain compatible with the 30 series, it is important to disable this connection when switching to an Anybus CompactCom 30 module due to pin conflicts. The RMII port of the host processor should be set to tristate by default, and only be enabled if an RMII capable Anybus CompactCom 40 is detected. In case the RMII connection cannot be disabled through an internal hardware control on the host processor, it will be necessary to design in external hardware (i.e. a FET bus switch) to prevent short circuits

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section 3.2.5, "RMII — Reduced Media-Independent Interface".

## H.3        General Software

### H.3.1       Extended Memory Areas

The memory areas have been extended in the 40-series, and it is now possible to access larger sizes of process data (up to 4096 bytes instead of former maximum 256 bytes) and message data (up to 1524 bytes instead of former maximum 255 bytes). The 30-series has reserved memory ranges that the application should not use. The 40-series implements new functionality in some of these memory areas.

> ⓘ   *To use the extended memory areas you need to implement a new communication protocol which is not part of this document.*
>
> *Memory areas not supported by the specific network cannot be used. Make sure you do not access these areas, e.g. for doing read/write memory tests.*

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, Section "Memory Map"

### H.3.2       Faster Ping-Pong Protocol

The ping-pong protocol (the protocol used in the 30-series) is faster in the 40-series. A 30-series module typically responds to a so called ping within 10-100 µs. The 40-series typically responds to a ping within 2 µs.

Interrupt-driven applications (parallel operating mode) may see increased CPU load due to the increased speed.

### H.3.3       Requests from Anybus CompactCom to Host Application During Startup

All requests to software objects in the host application must be handled and responded to (even if the object does not exist). This applies for both the 30-series and the 40-series. The 40-series introduces additional objects for new functionality.

There may also be additional commands in existing objects added to the 40-series that must be responded to (even if it is not supported).

If your implementation already responds to all commands it cannot process, which is the expected behavior, you do not need to change anything.

### H.3.4       Anybus Object (01h)

| Attribute | 30-series | 40-series | Change/Action/Comment |
|---|---|---|---|
| #1, Module Type | 0401h | 0403h | Make sure the host application accepts the new module type value for the 40-series. |
| #15, Auxiliary Bit | Available | Removed | It is not possible to turn off the "Changed Data Indication" in the 40-series. Also see "Control Register CTRL_AUX-bit" and "Status Register STAT_AUX-bit" below. |
| #16, GPIO Configuration | Default: General input and output pins | Default: LED3 and LED4 outputs | See also ..<br>• *GIP[0..1]/LED3[A..B], p. 169*<br>• *GOP[0..1]/LED4[A..B], p. 169* |

### H.3.5 Control Register CTRL_AUX-bit

| | |
|---|---|
| **30-series** | The CTRL_AUX bit in the control register indicates to the Anybus CompactCom if the process data in the current telegram has changed compared to the previous one. |
| **40-series** | The value of the CTRL_AUX bit is always ignored. Process data is always accepted. |

All released Anybus CompactCom 30 example drivers from Anybus CompactCom comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section "Control Register".

### H.3.6 Status Register STAT_AUX-bit

| | |
|---|---|
| **30-series** | The STAT_AUX bit in the status register indicates if the output process data in the current telegram has changed compared to the previous one. This functionality must be enabled in the Anybus object (01h), Attribute #15. By default, the STAT_AUX bit functionality is disabled. |
| **40-series** | The STAT_AUX bit indicates updated output process data (not necessarily changed data) from the network compared to the previous telegram. The functionality is always enabled. |

All released Anybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section "Status Register".

### H.3.7 Control Register CTRL_R-bit

| | |
|---|---|
| **30-series** | The application may change this bit at any time. |
| **40-series** | For the 8-bit parallel operating mode, the bit is only allowed to transition from 1 to 0 when the STAT_M-bit is set in the status register. When using the serial operating modes, it is also allowed to transition from 1 to 0 in the telegram immediately after the finalizing empty fragment. |

All released Anybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section "Control Register".

### H.3.8 Modifications of Status Register, Process Data Read Area, and Message Data Read Area

In the 40-series, the Status Register, the Process Data Read Area, and the Message Data Read Area are write protected in hardware (parallel interface). If the software for some reason writes to any of those areas, a change is needed.

All releasedAnybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

## H.4        Network Specific — PROFINET

Related Information:

*Anybus CompactCom 40 PROFINET IRT Network Guide, HMSI-27-226*

*Network Interface Appendix, Anybus CompactCom 30, PROFINET IO 2-Port, HMSI-168-49*

### H.4.1        Network Object (03h)

| Attribute | Default | Anybus CompactCom Product | Comment |
|---|---|---|---|
| #1, Network Type | 0084h | 30-series, PROFINET IO 1-Port | The 30-series module is a PROFINET RT module without IRT capabilities. The 40-series module has PROFINET IRT capabilities. The Network Type value differs between the different Anybus CompactCom PROFINET versions. |
| | 0096h | 30-series, PROFINET IO 2-Port | |
| | 0089h | 40-series, PROFINET IRT | |
| | 009Dh | 40-series, PROFINET IRT Fiber Optic | |
| | 00ADh | 40-series, PROFINET IRT + IIoT | |
| | 00AEh | 40-series, PROFINET IRT Fiber Optic +IIoT | |
| #2, Network Type String | "PROFINET IO" | 30-series, PROFINET IO 1-Port | The 30-series module is a PROFINET RT module without IRT capabilities. The 40-series module has PROFINET IRT capabilities. The Network Type value differs between the different Anybus CompactCom PROFINET versions. |
| | "PROFINET IO 2-Port" | 30-series, PROFINET IO 2-Port | |
| | "PROFINET IRT" | 40-series, PROFINET IRT and PROFINET IRT + IIoT | |
| | "PROFINET IRT Fiber Optic" | 40-series, PROFINET IRT Fiber Optic 40-series, PROFINET IRT Fiber Optic +IIoT | |

## H.4.2        PROFINET IO Object (F6h)

| Attribute | Default | Anybus CompactCom Product | Comment |
|---|---|---|---|
| #1, Device ID | 0007h<br><br>0009h<br><br>0010h | 30-series, PROFINET IO 1-Port<br><br>30-series, PROFINET IO 2-Port<br><br>40-series, PROFINET IRT | The Device ID controls how your product identifies itself on the PROFINET network.<br>If the attribute is implemented in the host application, it overrides the default value and there is no difference between the 30-series and the 40-series.<br>If the attribute is not implemented, the default value differs between the different Anybus CompactCom PROFINET types. |
| #3, Station Type | "ABCC-PRT"<br><br>"ABCC-PRT (2-Port)"<br><br>"CompactCom 40 PIR" | 30-series, PROFINET IO 1-Port<br><br>30-series, PROFINET IO 2-Port<br><br>40-series, PROFINET IRT | The Station Type defines the name your product uses to identify itself on the PROFINET network.<br>If the attribute is implemented in the host application, it overrides the default value and there is no difference between the 30-series and the 40-series.<br>If the attribute is not implemented, the default value differs between the different Anybus CompactCom PROFINET types. |
| #8, I&M Order ID | "ABCC-PRT"<br><br>"ABCC-PRT (2-Port)"<br><br>"CompactCom 40 PIR" | 30-series, PROFINET IO 1-Port<br><br>30-series, PROFINET IO 2-Port<br><br>40-series, PROFINET IRT | If the attribute is implemented in the host application, it overrides the default value and there is no difference between the 30-series and the 40-series.<br>If the attribute is not implemented, the default value differs between the different Anybus CompactCom PROFINET types. |
| #19, System Description | "HMS Industrial Networks Anybus CompactCom"<br><br>"Anybus CompactCom PROFINET IO 2-Port"<br><br>"HMS Industrial Networks Anybus-CompactCom 40" | 30-series, PROFINET IO 1-Port<br><br><br>30-series, PROFINET IO 2-Port<br><br><br>40-series, PROFINET IRT | If the attribute is implemented in the host application, it overrides the default value and there is no difference between the 30-series and the 40-series.<br>If the attribute is not implemented, the default value differs between the different Anybus CompactCom PROFINET types. |

| Attribute | 30-series | 40-series | Change/Action/Comment |
|---|---|---|---|
| #7, Record Data Mode | See Comment | | I&M Record data transparent mode (bit 1) is replaced with the IM_Options command in the Network PROFINET IO object (0Eh). If this bit is 0 in the current implementation, no action is needed |
| #13, I&M Profile ID | Used for all APIs | Only used for "non-zero" APIs | In the 40-series, this attribute is only read for submodules belonging to a "non-zero" API (e.g. PROFIdrive profile). Constant values (F600h, Generic Device and 0004h, Communication Module) are used for submodules belonging to API 0. In the 30-series this parameter is used for all API:s. |
| #14, I&M Profile Specific Type | Used for all APIs | Only used for "non-zero" APIs | In the 40-series, this attribute is only read for submodules belonging to a "non-zero API" (e.g. PROFIdrive profile). Constant values (F600h, Generic Device and 0004h, Communication Module) are used for submodules belonging to API 0. In the 30-series this parameter is used for all API:s. |
| #15, I&M Version | Available | Removed | Attribute removed in the 40-series. The Anybus CompactCom will never request this attribute. The host application cannot influence the I&M implemented by the Anybus CompactCom. |
| 16, I&M Supported | Available | Removed | Attribute removed in the 40-series. The Anybus CompactCom will never request this attribute. The host application cannot influence the I&M supported by the Anybus CompactCom. |
| #20, Interface description | Available | Removed | Attribute removed in the 40-series. The Anybus CompactCom will never request this attribute. |
| #21, Module ID Assignment Mode | Available | Removed | Attribute removed in the 40-series (Incremental Module Identification removed in the 40-series). The Anybus CompactCom will never request this attribute. |
| #22, System Contact | Available | Removed | Attribute removed in the 40-series. The Anybus CompactCom will never request this attribute. System Contact can now only be reached from the network via SNMP (sysContact). Also see *SNMP MIB-II, p. 179*. |
| #23, PROFIenergy functionality | Available | Removed | Attribute removed in the 40-series. The Anybus CompactCom will never request this attribute. PROFIenergy is now enabled by the application having implemented the Energy Control Object (F0h). |

| Command | 30-series | 40-series | Change/Action/Comment |
|---|---|---|---|
| Ar_Info_Ind | Available | Removed | Command removed in the 40-series. The Anybus CompactCom will never use this command in a request. The command is replaced by the command Expected_Ident_Ind |
| Ar_Offline_Ind | Available | Removed | Command removed in the 40-series. The Anybus CompactCom will never use this command in a request. The 30-series module issues this command to indicate to the host application that the module enters an offline state. Use Ar_Abort_Ind instead. |
| Plug_Submodule_Failed | Available | Removed | Command removed in the 40-series. The Anybus CompactCom will never use this command in a request. |
| Get_IM_Record | Available | Removed | Command removed in the 40-series. The Anybus CompactCom will never use this command in a request. It will use the general Get_Record-command instead (filter on index AFF0h-AFFFh). |
| Set_IM_Record | Available | Removed | Command removed in the 40-series. The Anybus CompactCom will never use this command in a request. It will use the general Set_Record-command instead (filter on index AFF0h-AFFh). |

### H.4.3 PROFINET Additional Diagnostic Object (0Fh)

Object removed in the 40-series. All diagnostics are handled via the standard diagnostic object (02h), Event code FFh. Only Channel diagnostics can be created.

### H.4.4 Diagnostic Object (02h)

The structure of network specific event information has changed in the create command. Instead of including diagnostic source information such as API, Slot, and Subslot in the data field, it is extracted from the extended diagnostic fields in the create command. API, Slot, and Subslot are determined with the help of Slot and ADI given by the extended diagnostic mode.

Process alarms cannot be created in the 40-series.

### H.4.5        Network Configuration Object (04h)

The instances in the Network Configuration Object have been rearranged for the Ethernet based modules for consistency. Network specific instances are moved to instance number 20 and onwards. This is done to increase the number of instances in the section that is not network specific.

If the host application is using any of the parameters below, the software must be updated to use the new instance numbers.

| Parameter Name | 30-series Instance # | 40-series Instance # |
| --- | --- | --- |
| DNS1 | 8 | 9 |
| DNS2 | 9 | 10 |
| Host Name | 10 | 11 |
| Domain Name | 11 | 12 |
| SMTP Server | 12 | 13 |
| SMTP User | 13 | 14 |
| SMTP Password | 14 | 15 |
| Station Name | 15 | 20 |
| F-Address | 20 | 21 |
| Function Tag | 16 | Parameter removed |
| Location Tag | 17 | Parameter removed |
| Installation Date | 18 | Parameter removed |
| Description | 19 | Parameter removed |

The following instances are removed in the 40-series. They are only possible to set via the network.

•       Instance #16 in 30-series, Function Tag

•       Instance #17 in 30-series, Location Tag

•       Instance #18 in 30-series, Installation Date

•       Instance #19 in 30-series, Description

### H.4.6        Network PROFINET IO Object (0Eh)

| Attribute | 30-series | 40-series | Change/Action/Comment |
| --- | --- | --- | --- |
| #3, Last OffLineInd ReasonCode | - | Removed | Attribute removed in the 40-series. Since the command Ar_Offline_Ind is removed, this attribute makes no sense.<br>If the host application is accessing this attribute, it will receive an error response when using the 40-series. |
| #9, ProfinetIoStack Init ErrorCode | - | Removed | Attribute removed in the 40-series.<br>If the host application is accessing this attribute, it will receive an error response when using the 40-series. |

### H.4.7        I&M4

I&M4 is removed in the 40-series. Writeable I&M records in GSD must be updated (see GSD section below).

### H.4.8        LED Indications

Changes has been made regarding the specification of the LED indications. See tables below for differences.
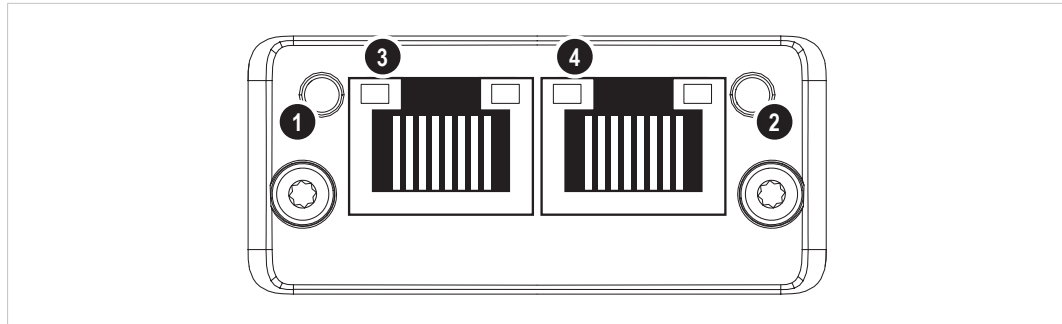
**Fig. 26**

### Network Status LED (LED1[A..B])

| LED State | 30-series | 40-series | Comments |
|---|---|---|---|
| Off | Offline | Offline | - No power- No connection with IO controller |
| Green | Online (RUN) | Online (RUN) | Connection with IO controller established, IO controller in Run state |
| Green, 1 flash | - | Online (STOP) | Connection with IO controller established, IO controller in STOP state or IO data bad, IRT synchronization not finished |
| Green, 3 flashes | - | Identify | Flashing 3 times (1Hz) continuously to identify slave (DCP_Identify). |
| Green, cont. flash | Online (STOP) | - | Connection with IO controller established, IO controller in STOP state |
| Red | Fatal event | Fatal event | Major internal error (this indication is combined with a red module status LED) |
| Red, 1 flash | - | Station Name error | Station name not set |
| Red, 2 flashes | - | IP address error | IP address not set |
| Red, 3 flashes | - | Configuration error | Expected Identification differs from Real Identification. |

**Module Status LED (LED2[A..B])**

| LED State | 30-series | 40-series | Comments |
|---|---|---|---|
| Off | Not Initialized | Not Initialized | No power or Module in SETUP or NW_INIT state |
| Green | Normal Operation | Normal Operation | Module has shifted from the NW_INIT state |
| Green, 1 flash | Diagnostic Event | Diagnostic Event | Diagnostic event(s) present |
| Green, cont. flash | Identify | - | Flashing 1Hz continuously to identify slave (DCP_Identify). |
| Red | Exception error | Exception error | Module in state Exception |
| | Fatal Event | Fatal Event | Major internal error (this indication is combined with a red Network Status LED) |
| Red, 1 flash | Configuration error | - | |
| Red, 2 flashes | IP address error | - | |
| Red, 3 flashes | Station Name Error | - | |
| Red, 4 flashes | Internal error | - | |
| Alternating Red/Green | - | Firmware Update | Do NOT power off the module. Turning the module off during this phase could cause permanent damage. |

## H.4.9    SNMP MIB-II

sysContact, sysLocation and sysName are used to give the product identification in the end installation.

In the 40-module these variables are only set from the network using SNMP protocol by the end user.

Due to this, sysContact (PROFINET IO object (F6h), Attribute 22), sysLocation (Network Configuration Object (04h), Attribute 17) and sysName (Network Configuration Object (04h), Attribute 15) are no longer used.

Nothing needs to be updated in your implementation but attribute values will no longer be used.

### H.4.10          ADI Based Configuration

When using the ADI based configuration, the structure of module identifiers and submodule identifiers are changed in the 40-series to be able to support the re-map functionality.

**30-series (2-Port)**

DAP V2.0: Module ID: 0x00000011, Submodule ID: 0x00000001

Modules:

Fixed module IDs for different data sizes and data directions

Output 1 byte - Module ID: 0x00000020, Submodule ID: 0x00000000

Output 1 word - Module ID: 0x00000030, Submodule ID: 0x00000000

Output 2 word - Module ID: 0x00000040, Submodule ID: 0x00000000

Output 4 word – Module ID: 0x00000050, Submodule ID: 0x00000000

Input 1 byte – Module ID: 0x00000002, Submodule ID: 0x00000000

Input 1 word – Module ID: 0x00000003, Submodule ID: 0x00000000

Input 2 word - Module ID: 0x00000004, Submodule ID: 0x00000000

Input 4 word – Module ID: 0x00000005, Submodule ID: 0x00000000

One submodule per module

**40-series**

DAP: Module ID: 0x80010000, Submodule ID: 0x00000001

Modules:

The Module IDs and Submodule IDs are built up according to the figure below.
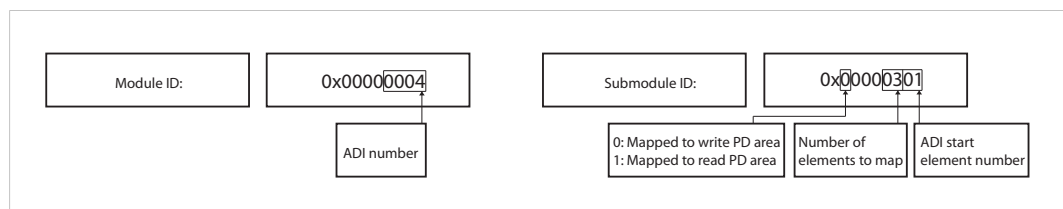


**Fig. 27**

### H.4.11        Configuration Mismatch

The 40-series are stricter regarding outputs and more tolerant regarding inputs compared to the 30-series. All outputs are locked if at least one output is missing. The network engineering tool may use anywhere between 0 and all actual input submodules without error indication.

ABCC40 provides a window for "seamless" adaptation of Real Identification (RI). In ADI based RI mode through remap, and in advanced mode by blocking in the connect request handling. It is not ok to adapt RI by rebooting.

For detailed information about how a configuration mismatch is handled in the 30-series, see *Anybus CompactCom 30 PROFINET IO Network Appendix, HMSI-168-74*, Section "Configuration Mismatch".

For detailed information about how a configuration mismatch is handled in the 40-series, see *Anybus CompactCom 40 PROFINET IRT Network Guide, HMSI-27-226*, Section "Configuration Mismatch".

### H.4.12        Media Redundancy Protocol (MRP)

MRP is supported in the 40-series and cannot be disabled by the host application. You need to add new keywords in your GSD file, see GSD section below.

### H.4.13        GSD File (PROFINET configuration file used by engineering tool)

**GSD Generator Tool**

When using the ADI-based configuration, it is possible to use a GSD Generator from HMS which will generate a basic GSD file (up to date with the above differences) with correct ADI modules.

> (**i**)    *The GSD generator only works for the 40-series*

The generator can be downloaded from www.anybus.com/starterkit40

**Modifications for Conformance Class B (if you want to disable support for IRT)**

The example GSD file, supplied by HMS Industrial Networks, is prepared for testing an Anybus CompactCom 40 PROFINET IRT for conformance class C, which includes PROFINET isochronous communication mode (IRT). If the implementation does not need the isochronous features of the device, the GSD file can be modified to reflect this. The product can then be conformance tested for conformance class B instead. The list below describe the changes needed in the example GSD file to accomplish this.

> **!** If IRT support is removed from the GSD file, it is not possible to use the device in the middle of an IRT line. It has to be connected to the line using an IRT capable switch or placed at the end of the line.

1. The value of the ConformanceClass attribute in the `<CertificationInfo...>` element in each DAP must be changed from "C" to "B":

   ```
   <CertificationInfo ConformanceClass="B" ApplicationClass=""
   NetloadClass="III"/>
   ```

2. The value of the SupportedRT_Classes attribute in the `<InterfaceSubmoduleItem...>` element in each DAP must be "RT_CLASS_1". I.e. the "RT_CLASS_2" and "RT_CLASS_3" values must be removed.

   ```
   <InterfaceSubmoduleItem ID="Interface" SubslotNumber="32768"

   SubmoduleIdentNumber="0x00000002" SupportedRT_Classes="RT_
   CLASS_1"

   TextId="T_ID_INTERFACE "SupportedProtocols="SNMP;LLDP" DCP_

   HelloSupported="true" PTP_BoundarySupported="true" DCP_
   BoundarySupported="true"
   ```

3. The elements `<RT_Class3Properties ...>`, `<SynchronisationMode ...>`, and `<RT_Class3TimingProperties ...>` must be removed from each DAP.

### Keywords

| Keyword | Comments |
|---|---|
| GSDML-DeviceProfile version | 2.31 for 30-series, 2.32 for 40-series |
| DeviceIdentity->DeviceID | Must be changed to reflect the Device ID you have configured in the PROFINET IO object (F6h), Attribute 1. |
| DeviceAccessPointItem->CheckDeviceID_Allowed | False for 30-series, true for 40-series |
| DeviceAccessPointItem->ImplementationType | NP30 for 30-series, NP40 for 40-series |
| DeviceAccessPointItem->WebServer | Optional keyword. Can be removed if not used. |
| DeviceAccessPointItem->LLDP_NoD_Supported | Manadatory, =true for 40-series |
| DeviceAccessPointItem->PowerOnToCommReady | Required if Fast Start-Up is supported |
| DeviceAccessPointItem->ResetToFactoryModes | 2 in 30-series, 2 and 8 in 40-series (2 mandatory) |
| DeviceAccessPointItem->PNIO_Version | V2.2 in 30-series, V2.32 in 40-series |
| ModuleInfo->HardwareRelease Value | Optional. Removed in 40-series |
| ModuleInfo->SoftwareRelease Value | Optional. Removed in 40-series |
| CertificationInfo->ConformanceClass | B in 30-series, B or C in 40-series depending on IRT. |
| CertificationInfo->NetloadClass | Mandatory. III in 40-series |
| VirtualSubmoduleItem->API | Removed in 40-series. API=0 default. |
| VirtualSubmoduleItem->IOData IOPS_Length | Removed in 40-series. Default=1 |
| VirtualSubmoduleItem->IOData IOCS_Length | Removed in 40-series. Default=1 |
| VirtualSubmoduleItem->Writeable_IM_Records | 1 2 3 4 in 30-series. 1 2 3 in 40-series. |
| VirtualSubmoduleItem->MayIssueProcessAlarm | Mandatory for 2.32. False in 40-series. |
| InterfaceSubmoduleItem->PTP_BoundarySupported | False in 30-series, true in 40-series |
| InterfaceSubmoduleItem->DCP_HelloSupported | True - Required if FSU is supported |
| InterfaceSubmoduleItem->DelayMeasurementSupported | Only for IRT (40-series) |
| InterfaceSubmoduleItem->RT_Class3Properties | Only for IRT (40-series) |
| InterfaceSubmoduleItem->SynchronisationMode | Only for IRT (40-series) |
| ApplicationRelations->StartupMode | Legacy in 30-series. Legacy and Advance in 40-series (mandatory). |
| MediaRedundancy SupportedRole="Client" | New MRP keyword in the 40-series. MRP not supported in 30-series. |
| SupportsRingportConfig="true" | New MRP keyword in the 40-series. MRP not supported in 30-series. |
| IsDefaultRingport="true" | New MRP keyword in the 40-series. MRP not supported in 30-series. |
| VirtualSubmoduleItem->Writeable_IM_Records= "1 2 3" | I&M4 removed in the 40-series |

# I    License Information

This product includes software developed by Carnegie Mellon, the Massachusetts Institute of Technology, the University of California, and RSA Data Security:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright 1986 by Carnegie Mellon.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright 1983,1984,1985 by the Massachusetts Institute of Technology

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (c) 1988 Stephen Deering.

Copyright (c) 1982, 1985, 1986, 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Stephen Deering of Stanford University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' ANDANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C) 1990-2, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Copyright 2013 jQuery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

"Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION

OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-
\*

rsvp.js

Copyright (c) 2013 Yehuda Katz, Tom Dale, and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of

this software and associated documentation files (the "Software"), to deal in

the Software without restriction, including without limitation the rights to

use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies

of the Software, and to permit persons to whom the Software is furnished to do

so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

*******************************************************************************-
*

libb (big.js)

The MIT Expat Licence.

Copyright (c) 2012 Michael Mclaughlin

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

'Software'), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-
\*

FatFs - FAT file system module R0.09b (C)ChaN, 2013

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial

developments under license policy of following trems.

Copyright (C) 2013, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for

personal, non-profit or commercial products UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-
\*

Copyright (c) 2002 Florian Schulze.

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. Neither the name of the authors nor the names of the contributors

may be used to endorse or promote products derived from this software

without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS ``AS IS'' AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGE.

ftpd.c - This file is part of the FTP daemon for lwIP

*****************************************************************************-

*

Format - lightweight string formatting library.

Copyright (C) 2010-2013, Neil Johnson

All rights reserved.

Redistribution and use in source and binary forms,

with or without modification,

are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

* Neither the name of nor the names of its contributors

may be used to endorse or promote products derived from this software

without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER

OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*****************************************************************************-

*

Print formatting routines

Copyright (C) 2002 Michael Ringgaard. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

3. Neither the name of the project nor the names of its contributors

may be used to endorse or promote products derived from this software

without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGE.

**************************************************************************-
*

lwIP is licenced under the BSD licence:

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED

WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT

OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING

IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY

OF SUCH DAMAGE.

*****************************************************************************-
*

Copyright (c) 2016 The MINIX 3 Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions
and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided
with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Author: David van Moolenbroek <david@minix3.org>

*****************************************************************************-
*

MD5 routines

Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

This software is provided 'as-is', without any express or implied

warranty. In no event will the authors be held liable for any damages

arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,

including commercial applications, and to alter it and redistribute it

freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not

claim that you wrote the original software. If you use this software

in a product, an acknowledgment in the product documentation would be

appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be

misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch

ghost@aladdin.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\_
\*

The "inih" library is distributed under the New BSD license:

Copyright (c) 2009,
Ben Hoyt All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions
and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided
with the distribution.
* Neither the name of Ben Hoyt nor the names of its contributors may be used to endorse or
promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY BEN HOYT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL BEN HOYT BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*******************************************************************************-
*

open62541 is licensed under the Mozilla Public License v2.0

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of
the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

To obtain customized changes please contact foss@anybus.com.

*******************************************************************************-
*

musl as a whole is licensed under the following standard MIT license:

----------------------------------------------------------------------

Copyright © 2005-2014 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and
associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to
do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

----------------------------------------------------------------------

*******************************************************************************-
*

PCG Random Number Generation for C.

Copyright 2014 Melissa O'Neill <oneill@pcg-random.org>

Licensed under the Apache License, Version 2.0 ("the License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For additional information about the PCG random number generation scheme, including its license and other licensing options, visit

http://www.pcg-random.org

*******************************************************************************-*

queue.h

Copyright (c) 1991, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@(#)queue.h 8.5 (Berkeley) 8/20/94