DELTA ELEKTRONIKA B.V.
DC POWER SUPPLIES

Vissersdijk 4, 4301 ND
Zierikzee, the Netherlands

DeltaPowerSupplies.com
Tel. +31 111 413656

IMPLEMENTATION GUIDE

Anybus® Carrier Interface

Firmware Update

It is strongly recommended to first perform a power-supply firmware update before further operation. Detailed instructions are provided in the Quick-Start Manual of the power supply.

Driver & Example Software

Drivers and example projects for a range of applications and interfaces can be downloaded from our website. **Products/Interfaces/Downloads**

INT-MOD-ANY
PSC-ANY-EXT

Document 202603
Firmware P0250 (SM15K)
Firmware P0210 (SM3300)
Firmware P0103 (PSC-ANY)

CONTENTS

1 Information on this document

1.1 Validity

This document is valid for the following products:

- INT-MOD-ANY (Pluggable Anybus-carrier interface module)
  - SM3300-series power supply with firmware version as stated on the cover or higher.
  - SM15K-series power supply with firmware version as stated on the cover or higher.
- PSC-ANY-EXT (External Anybus-carrier interface unit)
  - With firmware version as stated on the cover or higher.

1.2 Target Group

This equipment must be operated only by qualified personnel who understand the instructions and safety instructions provided with the equipment. If the equipment must be operated by unqualified personnel, then he/she must be supervised by qualified personnel.

1.3 Trademarks

Anybus® is a registered trademark of HMS Industrial Networks AB. All other trademarks mentioned in this document are the property of their respective holders.

1.4 Content and Structure

Please consider the safety instructions of all products involved before operating or connecting. This supplementary guide explains how to implement one of the supported pluggable Anybus inserts with an Anybus carrier interface with a power supply into a specific fieldbus.

1.5 Additional information

The following additional product-related documentation is available on our website. Please visit the website to find a list of the latest supplementary materials.

- INT-MOD-ANY: **Products > Interfaces > Anybus module > Downloads**
- PSC-ANY-EXT: **Products > Interfaces > Anybus unit (external) > Downloads**

Document name	Product	Type
Product Manual INT-MOD-ANY	INT-MOD-ANY	Manual
Python Example Code INT-MOD-ANY	INT-MOD-ANY	Software
Product Manual PSC-ANY-EXT	PSC-ANY-EXT	Manual
Python Example Code PSC-ANY-EXT	PSC-ANY-EXT	Software
Device description files Data Format A • CANopen EDS Data Format A • EtherCAT ESI Data Format A • Ethernet/IP EDS Data Format A • PROFIBUS GSD Data Format A • PROFINET GSDML Data Format A	INT-MOD-ANY PSC-ANY-EXT	Configuration files
Device description files Data Format B • CANopen EDS Data Format B • EtherCAT ESI Data Format B • Ethernet/IP EDS Data Format B • PROFIBUS GSD Data Format B • PROFINET GSDML Data Format B	INT-MOD-ANY	Configuration files

This guide describes how to implement a supported Anybus insert in combination with one of the Anybus carrier interfaces with a power supply into a specific fieldbus. To find specific settings related to the HMS Anybus insert itself, such as a transmission cycle rate of a certain protocol. Please consult the Network guide of the specific insert on the HMS Networks website. The inserts can be found on the **HMS Networks' website**.

1.6 Contact

If you need additional guidance or have any remarks or feedback, please visit our website and fill out a **contact form**. A structured technical contact form is available to enter descriptions and to upload files.

In case of malfunction or breakdown, please fill out the **RMA form** to have the product serviced.

2 General

2.1 Introduction

This document provides a guide in setting up a fieldbus with the INT-MOD-ANY and PSC-ANY-EXT products. The fieldbus implementation for a power supply using an Anybus insert for a specific protocol is explained in three steps following the preparation as described in Section 2.2. The three steps are illustrated in Figure 2.1

The first step is to get an overview of available commands and data. A limited set of commands is available for communication through an Anybus insert with a power supply. An overview of the available commands depending on the carrier-product and power supply series is shown in Section 2.4.

The second step is to choose a data format. Four data formats are available to send commands and retrieve data from the Anybus insert connected to a power supply. One of these data formats must be chosen before implementing a fieldbus. Each data format has its own specific advantages and disadvantages. These four global data formats are discussed in Section 2.5.

Step three is implementing specific protocol mapping, based on the previously chosen data format. The protocol-specific implementation is discussed in Chapter 3.

2.2 Preparation

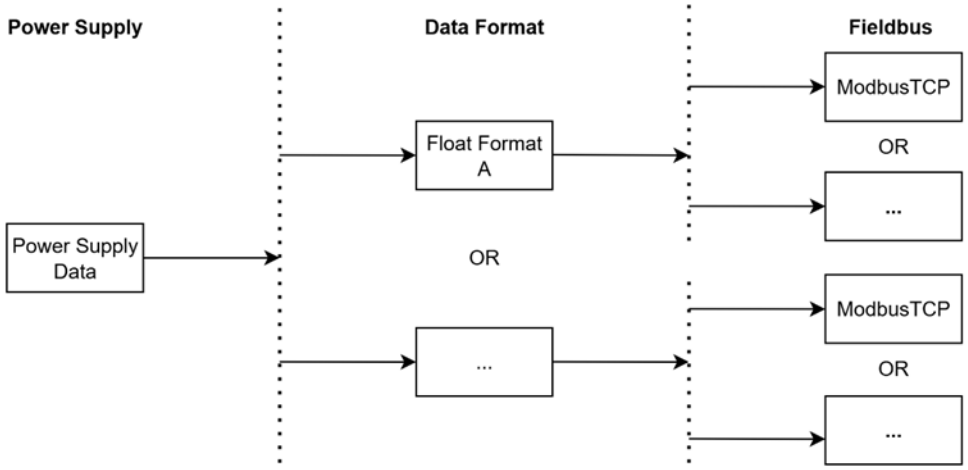

Figure 2.1: Flow of fieldbus implementation

Please make sure to complete the setup of all products as explained in the product manuals before implementing a fieldbus using this document. The combination of power supply, INT-MOD-ANY or PSC-ANY-EXT and an Anybus insert should be configured as explained in the product manuals.

Setup instructions can be found in each product manual, which can be downloaded from our website.

- INT-MOD-ANY: **Products > Interfaces > Anybus module > Downloads**
- PSC-ANY-EXT: **Products > Interfaces > Anybus unit (external) > Downloads**

The next step will be to implement a fieldbus using the equipment.

2.3 Example code

Several Python implementation examples are available for download on the website. These examples provide easy to follow examples as reference for other fieldbus implementations. The examples can be found on the website through the same links as mentioned above for the product manuals in Section 2.2.

2.4 Power supply data

An overview of all commands using the INT-MOD-ANY or PSC-ANY-EXT and compatibility with a power supply series is shown in the table below, see Table 2.1. For reference, a table with command and status descriptions is listed in Appendix A1. The table below, Table 2.1, lists a Refresh Counter. This is a counter that will increment each time the parameters are updated. When the counter hits 65536, it returns to 0.

The registers mentioned in Table 2.1 are implemented as 16-bit binary words. Each bit in this word represents a binary status or setting. The position of each bit is denoted by a bit weight value. For example, the RSD bit in RemCTRL is represented by the first bit, position 0. Please note that the RemCTRL word writes multiple settings at once.

Compatible

Not compatible

Future update

Parameter	Bit Weight	R/W	INT-MOD-ANY		PSC-ANY-EXT				
			SM15K	SM3300	SM6000	SM1500	SM800	ES300	ES150
CVprg	-	R/W	■	■	■	■	■	■	■
CCprg		R/W	■	■	■	■	■	■	■
CCneg		R/W	■						
CPprg		R/W	■						
CPneg		R/W	■						
CVmon		R	■	■	■	■	■	■	■
CCmon		R	■	■	■	■	■	■	■
CPmon		R	■						
CVlim		R	■	■					
CClim		R	■	■					
CClimneg		R	■	■					
CPlim		R	■						
CPlimneg		R	■						
Refresh Counter		R	■	■	■	■	■	■	■
Remote Control Register (RemCTRL)									
├ **RSD**	1	W	■	■	■	■	■	■	■
├ **Output**	2	W	■	■					
├ **RemCV**	512	W	■	■					
├ **RemCC**	1024	W	■	■					
└ **RemCP**	2048	W	■						
Status Register A									
├ **CV**	1	R	■	■					
├ **CC**	2	R	■	■	■	■	■	■	■
├ **CP**	4	R	■						
├ **V_{lim}**	8	R	■	■	■	■	■		
├ **I_{lim}**	16	R	■	■					
├ **P_{lim}**	32	R	■						
├ **DcfVolt**	64	R	■	■	■	■	■		
├ **DcfCurr**	128	R	■	■					
├ **OT**	256	R	■	■	■	■	■		
├ **PSOL**	512	R	■	■	■	■	■		
├ **ACF**	1024	R	■	■	■	■	■		
├ **Interlock**	2048	R	■	■					
├ **RSD**	4096	R	■	■	■	■	■	■	■
├ **Output**	8192	R	■	■					
└ **Frontpanellock**	16384	R	■	■					
Status Register B									
├ **Rem CV**	1	R	■	■					
├ **Rem CC**	2	R	■	■					
├ **Rem CP**	4	R	■						
├ **ProgRunning**	8	R	■	■					
├ **WaitForTrigger**	16	R	■	■					
├ **Master**	32	R	■	■					
├ **Slave**	64	R	■	■					
├ **V_{output} overload**	128	R	■	■	■	■	■		
├ **I_{output} overload**	256	R	■	■	■	■	■		
├ **Sensebreak**	8192	R	■	■					
├ **PROT**	16384	R	■	■					
└ **ProgOpenEndError**	32768	R	■	■					

Table 2.1: Available commands and compatibility with power supply series

2.5 Data formats

In the previous section an overview was given of compatible commands per power supply series. At this point a data format should be chosen that will be used to send the commands and receive data.

Four data formats exist, split in a float and a binary number representation. These are called Float Format A, Float Format B, 16-Bit Format A and 16-Bit Format B. Each format has their own advantages and disadvantages. These are compared and shown in Table 2.2.

The main difference between the float and 16-bit formats is the processing speed of the HMS Anybus insert. This is achieved by sending and requesting smaller packages containing more or fewer parameters with shorter word lengths, which can be seen in Table 2.3 and Table 2.4. The 16-bit format packages are smaller and therefore processed faster.

Please note that the Data Format B variants are only compatible for use with the SM15K-series power supplies in combination with the INT-MOD-ANY. Data Format A variants are compatible with PSC-ANY-EXT, SM3300- and SM15K-series power supplies in combination with an INT-MOD-ANY.

Data format	Advantages	Disadvantages
Float Format A	• Human readability (float representation) • Includes CVlim, CClim and Status Register B. • Avaliable on all Anybus compatible Delta products	• Slower processing speed
16-Bit Format A	• Fastest processing speed • Avaliable on all Anybus compatible Delta products	• Does not include CVlim, CClim and Status Register B. • Less human readable (binary representation)
Float Format B	• Human readability (float representation) • Includes CPprg, CPneg, CPmon, CVlim, CCneg, CClim, CClimneg, CPlim, CPlimneg and Status Register B. •	• Slowest processing speed • Only available for SM15K
16-Bit Format B	• Faster processing speed • Includes CPprg, CPneg, CPmon and CCneg.	• Does not include CVlim, CClim, CClimneg, CPlim, CPlimneg and Status Register B. • Less human readable (binary representation) • Only available for SM15K

Table 2.2: Data format comparison

For each data format, a table is shown below that indicates the datatype for each command and its value range. This is shown in Table 2.3 and Table 2.4. Chapter 3 of this guide will cover implementation for each specific protocol.

Note: When toggling the data format, RSD will be set active. This is a safety precaution.

2.5.1 Float Format A

The parameters in the float format, beside the refresh counter and the status registers are of type REAL32. The word size is 16-bit, so two words are needed for the float type parameters. The register type parameters are all of single-word length. This can be seen in Table 2.3.
The allowed range for the float parameters, as described in the Range column, are the maximum and minimum values of the power supply model.

Parameter	R/W	Data type	Range
CVprg	Read/Write	REAL32	[0 .. V_{max}]
CCprg	Read/Write	REAL32	[0 .. I_{max}]
CVmon	Read	REAL32	[0 .. V_{max}]
CCmon	Read	REAL32	[-I_{max} .. I_{max}]
CVlim	Read	REAL32	-
CClim	Read	REAL32	-
Refresh Counter	Read	UINT16	-
Remote Control Register (RemCTRL)	Write	UINT16	-
Status Register A	Read	UINT16	-
Status Register B	Read	UINT16	-

Table 2.3: Float Format A Datatype and Value Range

2.5.2 16-Bit Format A

All parameters in the 16-bit format are of single word length, this can be seen in Table 2.4. Status Register B, CVlim and CClim are not present in this data format to keep the total data package as small as possible for best processing speed. The parameters like CVprg and CCmon, which are not representing a binary status register, need to be scaled to be interpreted correctly. To program these parameters, the values are scaled to the Nominal parameter value as shown in the Nominal column in Table 2.4. For example, to program a SM66-AR-110 (max. 66 V, max. 110 A) power supply, to a CV setting of 33 V. The value to send to the Anybus insert is (62500/66)*33. See the equations below that illustrate the scaling operation.

$$Programming\ value\ = \left(\frac{Nominal\ Parameter\ value}{Maximum\ value\ of\ the\ power\ supply}\right) * Actual\ desired\ output\ value$$

$$Actual\ Measured\ output\ value = \left(\frac{Maximum\ value\ of\ the\ power\ supply}{Nominal\ parameter\ value}\right) * Monitoring\ value$$

Parameter	**R/W**	**Datatype**	**Range**	**Nominal (100% output)**
CVprg	Read/Write	UINT16	[0 .. 65000]	62500
CCprg	Read/Write	UINT16	[0 .. 32500]	31250
CVmon	Read	UINT16	[0 .. 65535]	62500
CCmon	Read	UINT16	[-32768 .. 32767]	31250
Refresh Counter	Read	UINT16	[0 .. 65535]	-
Remote Control Register (RemCTRL)	Write	UINT16	[0 .. 65535]	-
Status Register A	Read	UINT16	[0 .. 65535]	-

Table 2.4: 16-Bit Format A Datatype and Value Range

2.5.3 Float Format B (Only available for SM15K)

The parameters in the float format, beside the refresh counter and the status registers are of type REAL32. The word size is 16 bit, so two words are needed for the REAL32 type parameters. The register type parameters are all of single word length. This can be seen in Table 2.5.

The allowed range for the REAL32 parameters, as described in the Range column, are the maximum and minimum values of the power supply model.

Programming values are always positive.

Parameter	**R/W**	**Datatype**	**Range**
CVprg	Read/Write	REAL32	[0 .. V_{max}]
CCprg	Read/Write	REAL32	[0 .. I_{max}]
CCprgneg	Read/Write	REAL32	[0 .. -I_{max}]*
CPprg	Read/Write	REAL32	[0 .. P_{max}]
CCprgneg	Read/Write	REAL32	[0 .. -P_{max}]*
CVmon	Read	REAL32	[0 .. V_{max}]
CCmon	Read	REAL32	[-I_{max} .. I_{max}]
CPmon	Read	REAL32	[-P_{max} .. P_{max}]
CVlim	Read	REAL32	-
CClim	Read	REAL32	-
CClimneg	Read	REAL32	-
CPlim	Read	REAL32	-
CPlimneg	Read	REAL32	-
Refresh Counter	Read	UINT16	-
Remote Control Register (RemCTRL)	Write	UINT16	-
Status Register A	Read	UINT16	-
Status Register B	Read	UINT16	-

Table 2.5: Float Format B Datatype and Value Range

2.5.4 16-Bit Format B (Only available for SM15K)

All parameters in the 16-bit format are of single word length, this can be seen in Table 2.6. Status Register B, CVlim and CClim and CPlim are not present in this data format to keep the total data package as small as possible for best processing speed. The parameters like CVprg, CCmon and CPmon, which are not representing a binary status register, need to be scaled to be interpreted correctly. To program these parameters, the values are scaled to the Nominal parameter value as shown in the Nominal column in Table 2.4. For example, to program a SM210-CP-150 (max. 210 V, max. 150 A) power supply, to a CV setting of 33 V. The value to send to the Anybus insert is (62500/210)*33. See the equations below that illustrate the scaling operation.

$$Programming\ value\ = \left(\frac{Nominal\ Parameter\ value}{Maximum\ value\ of\ the\ power\ supply}\right) * Actual\ desired\ output\ value$$

$$Actual\ Measured\ output\ value = \left(\frac{Maximum\ value\ of\ the\ power\ supply}{Nominal\ parameter\ value}\right) * Monitoring\ value$$

The negative programming parameters need to always be entered as positive values. For example: If the negative power setpoint should be set to -7500W, the value for CPprgneg should be 15625(decimal).

Parameter	**R/W**	**Datatype**	**Range**	**Nominal (100% output)**
CVprg	Read/Write	UINT16	[0 .. 65000]	62500
CCprg	Read/Write	UINT16	[0 .. 32500]	31250
CCprgneg	Read/Write	UINT16	[0 .. 32500]	31250
CPprg	Read/Write	UINT16	[0 .. 32500]	31250
CPprgneg	Read/Write	UINT16	[0 .. 32500]	31250
CVmon	Read	UINT16	[0 .. 65535]	62500
CCmon	Read	UINT16	[-32768 .. 32767]	31250
CPmon	Read	UINT16	[-32768 .. 32767]	31250
Refresh Counter	Read	UINT16	[0 .. 65535]	-
Remote Control Register (RemCTRL)	Write	UINT16	[0 .. 65535]	-
Status Register A	Read	UINT16	[0 .. 65535]	-

Table 2.6: 16-Bit Format B Datatype and Value Range

3 Fieldbus implementation

3.1 Modbus-TCP - Read/Write Registers

Implementation of the Anybus insert into a Modbus-TCP network is done using Modbus registers. The Modbus registers are 16-bit wide. They are referenced by an address defined by the device address map. This map is shown for each data format in Table 3.1 and Table 3.2. The data type of each parameter was previously mentioned in Section 2.5. The meaning of each parameter is briefly described in Appendix A1. The minimum processing speed that can be obtained is mentioned in the data sheet of the interface. Together with the network latency, this processing speed defines the maximum polling rate. According to the Modbus-TCP network guide from HMS Networks, the insert can have up to four simultaneous connections. This information, however, needs to be verified with HMS by the user, as this aspect is not within our scope of influence. The Anybus insert has its own web interface, showing current parameter values, which can be used to aid implementation and debugging.

3.1.1 Float Format A

Parameter	Read		Write		Size
	Address	Function Code	Address	Function Code	
CVprg	0x800 + 0x801	0x03	0x000 + 0x001	0x10	2 Words
CVmon	0x802 + 0x803	0x03	-	-	2 Words
CCprg	0x804 + 0x805	0x03	0x002 + 0x003	0x10	2 Words
CCmon	0x806 + 0x807	0x03	-	-	2 Words
CVlim	0x808 + 0x809	0x03	-	-	2 Words
CClim	0x80A + 0x80B	0x03	-	-	2 Words
RemCTRL	-	-	0x004	0x06	1 Word
Refresh Counter	0x80C	0x03	-	-	1 Word
Status Register A	0x80D	0x03	-	-	1 Word
Status Register B	0x80E	0x03	-	-	1 Word

Table 3.1: Float format A Modbus-TCP device address map

3.1.2 16-Bit Format A

Parameter	Read		Write		Size
	Address	Function Code	Address	Function Code	
CVprg	0x800	0x03	0x000	0x06	1 Word
CVmon	0x801	0x03	-	-	1 Word
CCprg	0x802	0x03	0x001	0x06	1 Word
CCmon	0x803	0x03	-	-	1 Word
RemCTRL	-	-	0x002	0x06	1 Word
Refresh Counter	0x804	0x03	-	-	1 Word
Status Register A	0x805	0x03	-	-	1 Word

Table 3.2: 16-Bit format A Modbus-TCP device address map

3.1.3 Float Format B

Parameter	Read		Write		Size
	Address	Function Code	Address	Function Code	
CVprg	0x800 + 0x801	0x03	0x000 + 0x001	0x10	2 Words
CVmon	0x802 + 0x803	0x03	-	-	2 Words
CCprg	0x804 + 0x805	0x03	0x002 + 0x003	0x10	2 Words
CCprgneg	0x806 + 0x807	0x03	0x004 + 0x005	0x10	2 Words
CCmon	0x808 + 0x809	0x03	-	-	2 Words
CPprg	0x80A + 0x80B	0x03	0x006 + 0x007	0x10	2 Words
CPprgneg	0x80C + 0x80D	0x03	0x008 + 0x009	0x10	2 Words
CPmon	0x80E + 0x80F	0x03	-	-	2 Words
CVlim	0x810 + 0x811	0x03	-	-	2 Words
CClim	0x812 + 0x813	0x03	-	-	2 Words
CClimneg	0x814 + 0x815	0x03	-	-	2 Words
CPlim	0x816 + 0x817	0x03	-	-	2 Words
CPlimneg	0x818 + 0x819	0x03	-	-	2 Words
RemCTRL	-	-	0x00A	0x06	1 Word
Refresh Counter	0x81A	0x03	-	-	1 Word
Status Register A	0x81B	0x03	-	-	1 Word
Status Register B	0x81C	0x03	-	-	1 Word

Table 3.3: Float format B Modbus-TCP device address map

3.1.4 16-Bit Format B

Parameter	Read		Write		Size
	Address	Function Code	Address	Function Code	
CVprg	0x800	0x03	0x000	0x06	1 Word
CVmon	0x801	0x03	-	-	1 Word
CCprg	0x802	0x03	0x001	0x06	1 Word
CCprgneg	0x803	0x03	0x002	0x06	1 Word
CCmon	0x804	0x03	-	-	1 Word
CPprg	0x805	0x03	0x003	0x06	1 Word
CPprgneg	0x806	0x03	0x004	0x06	1 Word
CPmon	0x807	0x03	-	-	1 Word
RemCTRL	-	-	0x005	0x06	1 Word
Refresh Counter	0x808	0x03	-	-	1 Word
Status Register A	0x809	0x03	-	-	1 Word

Table 3.4: 16-Bit format B Modbus-TCP device address map

3.2 Ethernet/IP – Implicit Messaging

When writing to the Anybus module, a full array of bytes (BYTE[x]) must be sent.

- For data Format A this must include CVprg, CCprg, and RemCTRL even when only one of them is adjusted. For example, In 'Float Format A', this is a BYTE[10] (array of ten bytes), In '16-Bit Format A' this is a BYTE[6].
- For data Format B this must include CVprg, CCprg, CCprgneg, CPprg, CPprgnegand RemCTRL even when only one of them is adjusted. For example, In 'Float Format B', this is a BYTE[22] (array of ten bytes), In '16-Bit Format B' this is a BYTE[12].

3.2.1 Float Format A

Parameter	**Read** **object: 0x04,** **instance: 0x64,** **attribute: 0x03**	**Write** **object: 0x04,** **instance: 0x96,** **attribute: 0x03**	**Size**
CVprg	0d000 .. 0d031	BYTE[4+4+2]	2 Words
CVmon	0d032 .. 0d063	-	2 Words
CCprg	0d064 .. 0d095	BYTE[4+4+2]	2 Words
CCmon	0d096 .. 0d127	-	2 Words
CVlim	0d128 .. 0d159	-	2 Words
CClim	0d160 .. 0d191	-	2 Words
RemCTRL	-	BYTE[4+4+2]	1 Word
Refresh Counter	0d192 .. 0d207	-	1 Word
Status Register A	0d208 .. 0d223	-	1 Word
Status Register B	0d224 .. 0d239	-	1 Word

3.2.2 16-Bit Format A

Parameter	**Read** ***object: 0x04,*** ***instance: 0x64,*** ***attribute: 0x03***	**Write** **object: 0x04,** **instance: 0x96,** **attribute: 0x03**	**Size**
CVprg	0d000 .. 0d015	BYTE[2+2+2]	1 Word
CVmon	0d016 .. 0d031	-	1 Word
CCprg	0d032 .. 0d047	BYTE[2+2+2]	1 Word
CCmon	0d048 .. 0d063	-	1 Word
RemCTRL	-	BYTE[2+2+2]	1 Word
Refresh Counter	0d064 .. 0d079	-	1 Word
Status Register A	0d080 .. 0d095	-	1 Word

3.2.3 Float Format B

Parameter	Read object: 0x04, instance: 0x64, attribute: 0x03	Write object: 0x04, instance: 0x96, attribute: 0x03	Size
CVprg	0d000 .. 0d031	BYTE[4+4+4+4+4+2]	2 Words
CVmon	0d032 .. 0d063	-	2 Words
CCprg	0d064 .. 0d095	BYTE[4+4+4+4+4+2]	2 Words
CCprgneg	0d096 .. 0d127	BYTE[4+4+4+4+4+2]	2 Words
CCmon	0d128 .. 0d159	-	2 Words
CPprg	0d160 .. 0d191	BYTE[4+4+4+4+4+2]	2 Words
CPprgneg	0d192 .. 0d223	BYTE[4+4+4+4+4+2]	2 Words
CPmon	0d224 .. 0d255	-	2 Words
CVlim	0d256 .. 0d287	-	2 Words
CClim	0d288 .. 0d319	-	2 Words
CClimneg	0d320 .. 0d351	-	2 Words
CPlim	0d352 .. 0d383	-	2 Words
CPlimneg	0d384 .. 0d415	-	2 Words
RemCTRL	-	BYTE[4+4+4+4+4+2]	1 Word
Refresh Counter	0d416 .. 0d431	-	1 Word
Status Register A	0d432 .. 0d447	-	1 Word
Status Register B	0d448 .. 0d463	-	1 Word

3.2.4 16-Bit Format B

Parameter	Read *object: 0x04,* *instance: 0x64,* *attribute: 0x03*	Write object: 0x04, instance: 0x96, attribute: 0x03	Size
CVprg	0d000 .. 0d015	BYTE[2+2+2+2+2+2]	1 Word
CVmon	0d016 .. 0d031	-	1 Word
CCprg	0d032 .. 0d047	BYTE[2+2+2+2+2+2]	1 Word
CCprgneg	0d048 .. 0d063	BYTE[2+2+2+2+2+2]	1 Word
CCmon	0d064 .. 0d079	-	1 Word
CPprg	0d080 .. 0d095	BYTE[2+2+2+2+2+2]	1 Word
CPprgneg	0d096 .. 0d111	BYTE[2+2+2+2+2+2]	1 Word
CPmon	0d112 .. 0d127	-	1 Word
RemCTRL	-	BYTE[2+2+2+2+2+2]	1 Word
Refresh Counter	0d128 .. 0d143	-	1 Word
Status Register A	0d144 .. 0d159	-	1 Word

3.3 CANopen – SDO

Note that the CANopen insert does not have an internal termination resistor and that it should be added externally if required. CANopen EDS-files are available on our website.

3.3.1 Float Format A

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x200C	0x0001	0x200A	0x0001	2 Words
CVmon	0x200C	0x0002	-	-	2 Words
CCprg	0x200C	0x0003	0x200A	0x0002	2 Words
CCmon	0x200C	0x0004	-	-	2 Words
CVlim	0x200C	0x0005	-	-	2 Words
CClim	0x200C	0x0006	-	-	2 Words
RemCTRL	-	-	0x200B	0x0000	1 Word
Refresh Counter	0x200D	0x0000	-	-	1 Word
Status Register A	0x200E	0x0000	-	-	1 Word
Status Register B	0x200F	0x0000	-	-	1 Word

3.3.2 16-Bit Format A

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x2016	0x0001	0x2014	0x0001	1 Word
CVmon	0x2016	0x0002	-	-	1 Word
CCprg	0x2016	0x0003	0x2014	0x0002	1 Word
CCmon	0x2016	0x0004	-	-	1 Word
RemCTRL	-	-	0x2015	0x0000	1 Word
Refresh Counter	0x2017	0x0000	-	-	1 Word
Status Register A	0x2018	0x0000	-	-	1 Word

3.3.3 Float Format B

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x2020	0x0001	0x201E	0x0001	2 Words
CVmon	0x2020	0x0002	-	-	2 Words
CCprg	0x2020	0x0003	0x201E	0x0002	2 Words
CCprgneg	0x2020	0x0004	0x201E	0x0003	2 Words
CCmon	0x2020	0x0005	-	-	2 Words
CPprg	0x2020	0x0006	0x201E	0x0004	2 Words
CPprgneg	0x2020	0x0007	0x201E	0x0005	2 Words
CPmon	0x2020	0x0008	-	-	2 Words
CVlim	0x2020	0x0009	-	-	2 Words
CClim	0x2020	0x000A	-	-	2 Words
CClimneg	0x2020	0x000B	-	-	2 Words
CPlim	0x2020	0x000C	-	-	2 Words
CPlimneg	0x2020	0x000D	-	-	2 Words
RemCTRL	-	-	0x201F	0x0000	1 Word
Refresh Counter	0x2021	0x0000	-	-	1 Word
Status Register A	0x2022	0x0000	-	-	1 Word
Status Register B	0x2023	0x0000	-	-	1 Word

3.3.4 16-Bit Format B

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x202A	0x0001	0x2028	0x0001	1 Word
CVmon	0x202A	0x0002	-	-	1 Word
CCprg	0x202A	0x0003	0x2028	0x0002	1 Word
CCprgneg	0x202A	0x0004	0x2028	0x0003	1 Word
CCmon	0x202A	0x0005	-	-	1 Word
CPprg	0x202A	0x0006	0x2028	0x0004	1 Word
CPprgneg	0x202A	0x0007	0x2028	0x0005	1 Word
CPmon	0x202A	0x0008	-	-	1 Word
RemCTRL	-	-	0x2029	0x0000	1 Word
Refresh Counter	0x202B	0x0000	-	-	1 Word
Status Register A	0x202C	0x0000	-	-	1 Word

3.4 EtherCAT – SDO

EtherCAT ESI-files are available on our website. These configuration files can be imported into the desired programming software.

3.4.1 Float Format A

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x200C	0x0001	0x200A	0x0001	2 Words
CVmon	0x200C	0x0002	-	-	2 Words
CCprg	0x200C	0x0003	0x200A	0x0002	2 Words
CCmon	0x200C	0x0004	-	-	2 Words
CVlim	0x200C	0x0005	-	-	2 Words
CClim	0x200C	0x0006	-	-	2 Words
RemCTRL	-	-	0x200B	0x0000	1 Word
Refresh Counter	0x200D	0x0000	-	-	1 Word
Status Register A	0x200E	0x0000	-	-	1 Word
Status Register B	0x200F	0x0000	-	-	1 Word

3.4.2 16-Bit Format A

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x2016	0x0001	0x2014	0x0001	1 Word
CVmon	0x2016	0x0002	-	-	1 Word
CCprg	0x2016	0x0003	0x2014	0x0002	1 Word
CCmon	0x2016	0x0004	-	-	1 Word
RemCTRL	-	-	0x2015	0x0000	1 Word
Refresh Counter	0x2017	0x0000	-	-	1 Word
Status Register A	0x2018	0x0000	-	-	1 Word

3.4.3 Float Format B

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x2020	0x0001	0x201E	0x0001	2 Words
CVmon	0x2020	0x0002	-	-	2 Words
CCprg	0x2020	0x0003	0x201E	0x0002	2 Words
CCprgneg	0x2020	0x0004	0x201E	0x0003	2 Words
CCmon	0x2020	0x0005	-	-	2 Words
CPprg	0x2020	0x0006	0x201E	0x0004	2 Words
CPprgneg	0x2020	0x0007	0x201E	0x0005	2 Words
CPmon	0x2020	0x0008	-	-	2 Words
CVlim	0x2020	0x0009	-	-	2 Words
CClim	0x2020	0x000A	-	-	2 Words
CClimneg	0x2020	0x000B	-	-	2 Words
CPlim	0x2020	0x000C	-	-	2 Words
CPlimneg	0x2020	0x000D	-	-	2 Words
RemCTRL	-	-	0x201F	0x0000	1 Word
Refresh Counter	0x2021	0x0000	-	-	1 Word
Status Register A	0x2022	0x0000	-	-	1 Word
Status Register B	0x2023	0x0000	-	-	1 Word

3.4.4 16-Bit Format B

Parameter	Read		Write		Size
	Index	Subindex	Index	Subindex	
CVprg	0x202A	0x0001	0x2028	0x0001	1 Word
CVmon	0x202A	0x0002	-	-	1 Word
CCprg	0x202A	0x0003	0x2028	0x0002	1 Word
CCprgneg	0x202A	0x0004	0x2028	0x0003	1 Word
CCmon	0x202A	0x0005	-	-	1 Word
CPprg	0x202A	0x0006	0x2028	0x0004	1 Word
CPprgneg	0x202A	0x0007	0x2028	0x0005	1 Word
CPmon	0x202A	0x0008	-	-	1 Word
RemCTRL	-	-	0x2029	0x0000	1 Word
Refresh Counter	0x202B	0x0000	-	-	1 Word
Status Register A	0x202C	0x0000	-	-	1 Word

3.5 PROFIBUS – DPV1

PROFIBUS GSD-files are available on our website. These configuration files can be imported into the desired programming software. Read and Write represent the size of the variable in bytes. Offset given in number of bytes. All data is found in Slot 0.

3.5.1 Float Format A

Module	Parameter	Read	Write	Offset	Hex code
PROGRAMMING REGISTER	**CVprg**	-	4	0	0x 83 C3 00 0A 00
	CCprg	-	4	4	
MONITORING REGISTER	**CVprg**	4	-	0	0x 43 CB 00 0C 00
	CVmon	4	-	4	
	CCprg	4	-	8	
	CCmon	4	-	12	
	CVlim	4	-	16	
	CClim	4	-	20	
REMOTE CONTROL REGISTER	**RemCTRL**	-	2	0	0x 83 C0 00 0B 00
REFRESH COUNTER	**Refresh Counter**	2	-	0	0x 43 C0 00 0D 00
STATUS REGISTER A	**Status Register A**	2	-	0	0x 43 C0 00 0E 00
STATUS REGISTER B	**Status Register B**	2	-	0	0x 43 C0 00 0F 00

3.5.2 16-Bit Format A

Module	Parameter	Read	Write	Offset	Hex code
PROGRAMMING REGISTER	**CVprg**	-	2	0	0x 83 C1 00 14 00
	CCprg	-	2	2	
MONITORING REGISTER	**CVprg**	2	-	0	0x 43 C3 00 16 00
	CVmon	2	-	2	
	CCprg	2	-	4	
	CCmon	2	-	6	
REMOTE CONTROL REGISTER	**RemCTRL**	-	2	0	0x 83 C0 00 15 00
REFRESH COUNTER	**Refresh Counter**	2	-	0	0x 43 C0 00 17 00
STATUS REGISTER A	**Status Register A**	2	-	0	0x 43 C0 00 18 00

3.5.3 Float Format B

Module	Parameter	Read	Write	Offset	Hex code
PROGRAMMING REGISTER	**CVprg**	-	4	0	0x 83 C9 00 1E 00
	CCprg	-	4	4	
	CCprgneg		4	8	
	CPprg		4	12	
	CPprgneg		4	16	
MONITORING REGISTER	**CVprg**	4	-	0	0x 43 D9 00 20 00
	CVmon	4	-	4	
	CCprg	4	-	8	
	CCprgneg	4		12	
	CCmon	4	-	16	
	CPprg	4		20	
	CPprgneg	4		24	
	CPmon	4		28	
	CVlim	4	-	32	
	CClim	4	-	36	
	CClimneg	4		40	
	CPlim	4		44	
	CPlimneg	4		48	
REMOTE CONTROL REGISTER	**RemCTRL**	-	2	0	0x 83 C0 00 1F 00
REFRESH COUNTER	**Refresh Counter**	2	-	0	0x 43 C0 00 21 00
STATUS REGISTER A	**Status Register A**	2	-	0	0x 43 C0 00 22 00
STATUS REGISTER B	**Status Register B**	2	-	0	0x 43 C0 00 23 00

3.5.4 16-Bit Format B

Module	Parameter	Read	Write	Offset	Hex code
PROGRAMMING REGISTER	**CVprg**	-	2	0	0x 83 C4 00 28 00
	CCprg	-	2	2	
	CCprgneg		2	4	
	CPprg		2	6	
	CPprgneg		2	8	
MONITORING REGISTER	**CVprg**	2	-	0	0x 43 C7 00 2A 00
	CVmon	2	-	2	
	CCprg	2	-	4	
	CCprgneg	2		6	
	CCmon	2	-	8	
	CPprg	2		10	
	CPprgneg	2		12	
	CPmon	2		14	
REMOTE CONTROL REGISTER	**RemCTRL**	-	2	0	0x 83 C0 00 28 00
REFRESH COUNTER	**Refresh Counter**	2	-	0	0x 43 C0 00 2B 00
STATUS REGISTER A	**Status Register A**	2	-	0	0x 43 C0 00 2C 00

3.6 PROFINET IRT

PROFINET GSDML-files are available on our website. These configuration files can be imported into the desired programming software. Offset and sizes are given in number of Words.

3.6.1 Float Format A

Parameter	Read			Write			Size
	Module	Submodule	Offset	Module	Submodule	Offset	
CVprg	0x0000000C	0x00000600	0	0x0000000A	0x10000200	0	2 Words
CVmon	0x0000000C	0x00000600	2	-	-	-	2 Words
CCprg	0x0000000C	0x00000600	4	0x0000000A	0x10000200	2	2 Words
CCmon	0x0000000C	0x00000600	6	-	-	-	2 Words
CVlim	0x0000000C	0x00000600	8	-	-	-	2 Words
CClim	0x0000000C	0x00000600	10	-	-	-	2 Words
RemCTRL	-	-	-	0x0000000B	0x10000100	0	1 Word
Refresh Counter	0x0000000D	0x00000100	0	-	-	-	1 Word
Status Register A	0x0000000E	0x00000100	0	-	-	-	1 Word
Status Register B	0x0000000F	0x00000100	0	-	-	-	1 Word

3.6.2 16-Bit Format A

Parameter	Read			Write			Size
	Module	Submodule	Offset	Module	Submodule	Offset	
CVprg	0x00000016	0x00000400	0	0x00000014	0x10000200	0	1 Word
CVmon	0x00000016	0x00000400	1	-	-	-	1 Word
CCprg	0x00000016	0x00000400	2	0x00000014	0x10000200	1	1 Word
CCmon	0x00000016	0x00000400	3	-	-	-	1 Word
RemCTRL	-	-	-	0x00000015	0x10000100	0	1 Word
Refresh Counter	0x00000017	0x00000100	0	-	-	-	1 Word
Status Register A	0x00000018	0x00000100	0	-	-	-	1 Word

3.6.3 Float Format B

Parameter	Read			Write			Size
	Module	Submodule	Offset	Module	Submodule	Offset	
CVprg	0x00000020	0x00000D00	0	0x0000001E	0x10000500	0	2 Words
CVmon	0x00000020	0x00000D00	2	-	-	-	2 Words
CCprg	0x00000020	0x00000D00	4	0x0000001E	0x10000500	2	2 Words
CCprgneg	0x00000020	0x00000D00	6	0x0000001E	0x10000500	4	2 Words
CCmon	0x00000020	0x00000D00	8	-	-	-	2 Words
CPprg	0x00000020	0x00000D00	10	0x0000001E	0x10000500	6	2 Words
CPprgneg	0x00000020	0x00000D00	12	0x0000001E	0x10000500	8	2 Words
CPmon	0x00000020	0x00000D00	14	-	-	-	2 Words
CVlim	0x00000020	0x00000D00	16	-	-	-	2 Words
CClim	0x00000020	0x00000D00	18	-	-	-	2 Words
CClimneg	0x00000020	0x00000D00	20	-	-	-	2 Words
CPlim	0x00000020	0x00000D00	22	-	-	-	2 Words
CPlimneg	0x00000020	0x00000D00	24	-	-	-	2 Words
RemCTRL	-	-	-	0x0000001F	0x10000100	0	1 Word
Refresh Counter	0x00000021	0x00000100	0	-	-	-	1 Word
Status Register A	0x00000022	0x00000100	0	-	-	-	1 Word
Status Register B	0x00000023	0x00000100	0	-	-	-	1 Word

3.6.4 16-Bit Format B

Parameter	Read			Write			Size
	Module	Submodule	Offset	Module	Submodule	Offset	
CVprg	0x0000002A	0x00000800	0	0x00000028	0x10000500	0	1 Word
CVmon	0x0000002A	0x00000800	1	-	-	-	1 Word
CCprg	0x0000002A	0x00000800	2	0x00000028	0x10000500	1	1 Word
CCprgneg	0x0000002A	0x00000800	3	0x00000028	0x10000500	2	1 Word
CCmon	0x0000002A	0x00000800	4	-	-	-	1 Word
CPprg	0x0000002A	0x00000800	5	0x00000028	0x10000500	3	1 Word
CPprgneg	0x0000002A	0x00000800	6	0x00000028	0x10000500	4	1 Word
CPmon	0x0000002A	0x00000800	7	-	-	-	1 Word
RemCTRL	-	-	-	0x00000029	0x10000100	0	1 Word
Refresh Counter	0x0000002B	0x00000100	0	-	-	-	1 Word
Status Register A	0x0000002C	0x00000100	0	-	-	-	1 Word

A. Appendix

A1. Command and Status Descriptions

Parameter	Read/ Write	Description
CVprg	R/W	Read or Write the Constant Voltage (CV) setting
CCprg	R/W	Read or Write the positive Constant Current (CC) setting
CCneg	R/W	Read or Write the negative Constant Current (CC) setting
CPprg	R/W	Read or Write the positive Constant Power (CP) setting
CPneg	R/W	Read or Write the negative Constant Power (CP) setting
CVmon	R	Read the measured Voltage
CCmon	R	Read the measured Current
CPmon	R	Read the measured Power
CVlim	R	Read the Constant Voltage (CV) Limit setting
CClim	R	Read the Constant Current (CC) positive Limit setting
CClimneg	R	Read the Constant Current (CC) negative Limit setting
CPlim	R	Read the positive Constant Power (CP) Limit setting
CPlimneg	R	Read the negative Constant Power (CP) Limit setting
Refresh Counter	R	Read the Refresh Counter value of the Anybus insert
Remote Control (RemCTRL)		Register of multiple settings
├ **RSD**	W	Set Remote Shut Down (1:ON)
├ **Output**	W	Set Output, enable or disable (1:ON, 0:OFF)
├ **RemCV**	W	Set Constant Voltage (CV) programming source to Remote (1:ON)
├ **RemCC**	W	Set Constant Current (CC) programming source to Remote (1:ON)
└ **RemCP**	W	Set Constant Power (CP) programming source to Remote (1:ON)
Status Register A		Register of multiple statuses (ON or OFF)
├ **CV**	R	Read CV-mode status
├ **CC**	R	Read CC-mode status
├ **CP**	R	Read CP-mode status
├ **V_{lim}**	R	Read Voltage-Limit Alert status
├ **I_{lim}**	R	Read Current-Limit Alert status
├ **P_{lim}**	R	Read Power-Limit Alert status
├ **DcfVolt**	R	Read DC-fail Voltage status
├ **DcfCurr**	R	Read DC-fail Current status
├ **OT**	R	Read Over-Temperature Alert status
├ **PSOL**	R	Read Power Supply Overload status
├ **ACF**	R	Read AC-Failure Alert status
├ **Interlock**	R	Read Interlock status
├ **RSD**	R	Read Remote Shutdown Status
├ **Output**	R	Read Output ON/off status
└ **Frontpanellock**	R	Read Front Panel Lock status
Status Register B		Register of multiple statuses (ON or OFF)
├ **Rem CV**	R	Read Constant Voltage (CV) programming source Status
├ **Rem CC**	R	Read Constant Current (CC) programming source Status
├ **Rem CP**	R	Read Constant Power (CP) programming source Status
├ **ProgRunning**	R	Read Sequencer Program status
├ **WaitForTrigger**	R	Read Sequencer Trigger status
├ **Master**	R	Read Master status, for Master/Slave operation
├ **Slave**	R	Read Slave status, for Master/Slave operation
├ **V_{output} overload**	R	Read Output voltage overload status
├ **I_{output} overload**	R	Read Output current overload status
├ **Sensebreak**	R	Read Sense break status of remote sensing
├ **PROT**	R	Read Self Protect status
└ **ProgOpenEndError**	R	Read Sequencer Program Open End Error status

Additional notes:

- RemCV, RemCC, RemCP set the programming source of a power supply to remote. The SM3300 and SM15K power supplies will automatically set the programming source to the Slot with the Anybus carrier module inserted.
- RSD and Output (ON/off) settings are similar in behavior. However, individual behavior can be customized in case of a power sink option or natively bidirectional power supply like the SM15K-series. Please consult the manuals of the power supply and or power sink option.