



sercos International e. V.  
Kueblerstrasse 1  
73079 Suessen, Germany  
[www.sercos.de](http://www.sercos.de)



Steinbeis-Transferzentrum  
Systemtechnik

Steinbeis-Transferzentrum Systemtechnik  
Prof. Keller und Partner  
Martinstrasse 42-44  
73728 Esslingen, Germany  
[www.steinbeis-tzs.de](http://www.steinbeis-tzs.de)

## sercos EasySlave-IO

---

### Datasheet

Version: 0.4  
Date: 18.11.2012

#### **Abstract:**

The sercos EasySlave-IO is a bitstream variant of the sercos EasySlave for digital IO applications. The technical datasheet of the EasySlave-IO contains a description of the operation mode and the necessary parameters.

Project: **sercos EasySlave-IO – Datasheet**  
Document Type: **Datasheet**  
File: **Datasheet\_sercos\_EasySlave-IO\_YYYYMMDD.doc**

Authors:  
• Christian Hayer (Steinbeis-Transferzentrum Systemtechnik)

The Steinbeis-Transferzentrum Systemtechnik and sercos International e. V. are not liable for any errors in this documentation. Liability for direct and indirect damages arising in connection with the supply of this documentation is excluded in so far as it can be attached legally.

This documentation contains copyright-protected information.

All rights, especially the right of duplication, distribution and translation, are reserved. No portion of the documentation may be reproduced, copied or distributed in any form (photocopy, microfilm, electronic file or other process) without prior written permission of sercos International e. V.

## Revision History

Date	Version	Revision	Authors
19.07.2012	0.1	Document created	C. Hayer
18.09.2012	0.2	First Draft	C. Hayer
07.11.2012	0.3	Minor changes	C. Hayer
16.11.2012	0.4	<ul style="list-style-type: none"><li>- Update of EasySlave FPGA resources</li><li>- Chapter Device status and control words removed, see further information sercos Wiki website</li><li>- Chapter Hot-plug, S/IP, Firmware and parameter update added</li></ul>	C. Hayer

# Contents

1	Definitions and abbreviations.....	5
2	Introduction .....	6
2.1	sercos – the automation bus .....	6
2.2	sercos EasySlave .....	7
3	FPGA devices.....	8
3.1	EasySlave FPGA resources and bit streams.....	8
3.2	EasySlave-IO pinout.....	9
3.2.1	Pin assignment .....	10
4	Short Description of the FPGA structure .....	16
5	EasySlave IO processing .....	18
5.1	Cyclic IO communication in CP4.....	18
5.1.1	Output processing.....	18
5.1.2	Input processing .....	18
5.1.3	Realtime data communication .....	19
5.1.4	IO Bit assignment .....	20
5.1.4.1	Outputs.....	20
5.1.4.2	Inputs.....	21
5.1.5	Output timing .....	22
5.2	Diagnosis and Error management .....	23
5.2.1	Sub-device errors.....	23
5.2.1.1	Supply error .....	23
5.2.1.2	I/O errors.....	23
5.2.1.3	Error handling and error flag reset.....	23
5.2.2	Output fallback operation .....	23
6	Firmware description.....	24
6.1	sercos III version.....	24
6.2	EasySlave parameters .....	24
6.2.1	Communication parameters (SCP) .....	25
6.2.2	Device identification (GDP) .....	27
6.2.3	IO parameters (FSP IO).....	28
6.2.4	SDDML .....	29
6.3	Hot-plug.....	30
6.4	S/IP Services .....	30
6.5	Firmware and parameter update.....	31
7	Notice .....	32
8	References .....	33

# 1 Definitions and abbreviations

<b>Definition</b>	<b>Description</b>
Ethernet	A widely used type of local area network and compatible to ISO/IEC 8802-3 standard.
sercos III	Third generation sercos is an industrial standard which specifies real-time Ethernet communication for automation.
EasySlave	A free IP core for low-cost FPGA chips, which allows sercos III to be integrated into basic I/O slave devices with minimal development and integration effort.
FPGA	Field-programmable gate array is a device where the logic network can be programmed into the device after its manufacture. An FPGA consists of an array of logic elements, either gates or lookup table RAMs, flip-flops and programmable interconnect wiring.
MicroBlaze (MB)	The MicroBlaze is a soft-core processor designed for Xilinx FPGA's.
TFTP	Trivial File Transfer Protocol is a very simple file transfer protocol which is typically based on UDP.
IDN	Identification number of a sercos parameter.
UC channel	Unified Communication Channel (abbr.: UCC), Standard Ethernet Frames like UDP & TCP will be transmitted and services like Ping will be transmitted also.
RT channel	Real-time channel is a certain time span of the communication cycle in which real-time telegrams are sent.
SVC	The service channel (abbr.: SVC) is for non real-time transmission of information upon master request during RT channel
SCP	Sercos Communication Profile (abbr.: SCP)
GDP	Generic Device Profile (abbr.: GDP)
FSP IO	Function Specific Profile (abbr.: FSP) IO is a sercos III device profile for I/O applications.
SDDML	Sercos Device Description Markup Language is a markup language, which contains sercos III-defined tags, in order to describe the functionality that a sercos III device supports, in the form of an XML file.
CC	Cross-communication (abbr.: CC) is a direct data transfer between sercos III devices using connections, i.e., it enables direct data transfer between slaves without active involvement of the master.
Hot-plug	Possibility to insert a slave in the sercos III network, inclusive its initialization, while the sercos III network is running.
S/IP	All IPS services, which are based on TCP and UDP.

## 2 Introduction

### 2.1 sercos – the automation bus

Industrial Ethernet has become the de facto standard for manufacturing information networking, and the market is requesting Ethernet connectivity for servo drives. Ethernet offers high-speed data throughput 10-100 times faster than fieldbus solutions. It uses standard off-the-shelf components and cabling and offers consistent IT implementation from office to the machine level. The problem is that Industrial Ethernet is characterized by high bandwidth and low hardware costs, but is not deterministic. Office communications and certain single-axis motion applications can tolerate delays and data re-transmissions, but that would be disastrous for coordinated multi-axis robots or high-speed machine tools.

The sercos automation bus I optimized for deterministic high-speed motion control, which is required for the accurate synchronization of multiple drives. Sercos also defines a protocol structure and includes various profile definitions for motion and I/O devices.

Sercos III is the open, IEC-standardized third-generation of sercos that redefines Industrial Ethernet for real-time control. Sercos III brings Ethernet and previous sercos designs to provide the highly deterministic bi-directional real-time motion and I/O control that is required by modern production equipment. It overcomes the wasted bandwidth in other TCP/IP-based Ethernet bus solutions, because it is based directly on Ethernet frames, defining a new, registered EtherType for sercos. In addition to real-time communications between drives and the motion control, sercos III provides rich I/O communication capabilities, while also enabling other protocols, such as EtherNet/IP, TCP/IP, UDP and others, to be transmitted over the same Ethernet network efficiently in parallel with sercos real-time communication. Sercos III is a truly a universal automation bus for machine production and system implementation.

Sercos III offers several fundamental performance and technology benefits for OEMs and end-users:

- Cycle times as low as 31.25 microseconds.
- High speed: it uses Fast Ethernet (100 Mb/s).
- Support for either line or ring topologies; in addition hierarchical, synchronized and real-time coupled network structures can be implemented.
- Support for up to 511 slave devices in one network, with multiple networks possible in a system.
- Bumpless cable break recovery in ring mode within 25 microseconds.
- Advanced cross communications—both slave-to-slave and controller-to-controller (sometimes called machine-to-machine).
- Capable of hot-plugging devices and network segments—adding machine or line components to a network with synchronization up and running, without having to reset the network or cycle power.
- Support for safety functions up to SIL3 according to IEC 61508 via CIP Safety for sercos.
- I/O profile that provides an XML-based device and profile description language for I/O device configuration.
- Energy profile that defines parameters and commands for the reduction of energy consumption in a uniform vendor-independent manner.
- Encoder profile that provides a standard method to integrate encoders into a sercos III network.
- Lower hardware costs.

There are several key advantages that manufacturers, systems engineers and machine builders can leverage when using sercos III—advantages that enable drive and control systems with vastly improved flexibility and performance.

## 2.2 sercos EasySlave

The sercos EasySlave is an FPGA-based single-chip controller, enabling inexpensive development of simple sercos III slave devices such as I/O. I/O applications are synchronized in the sercos cycle. An IP core is provided as a netlist for the Xilinx Spartan-6 FPGA family. The IP core contains all the functions of a sercos slave, including the associated software library for I/O devices (e.g., analog inputs, encoders). The user can add IP cores for I/O from Xilinx and his own application code and can integrate the controller into his own board. A reference design is available.

Technical support for the EasySlave is provided by Steinbeis-Transferzentrum Systemtechnik (Tzs) in Esslingen, Germany. Tzs offers an EasySlave evaluation kit to facilitate a quick and easy introduction to sercos slave development. The evaluation kit includes a development board based on a Xilinx Spartan-6 XC6SLX16/25 FPGA and can be extended using plug-in modules. The evaluation kit includes all other required material and documents (Ethernet cable, power supply, and documentation). Furthermore Tzs can provide assistance in the integration and commissioning of EasySlave.

Features and specifications for EasySlave FPGA-based Single-Chip Controller

- FPGA-based single chip controller containing a reduced sercos slave IP core
- Target devices: analog/digital I/Os, simple sensors and actuators
- FPGA-internal RAM for sercos stack, application and data
- Real-time channel with maximum 32 bytes output and maximum 32 bytes input data (1 input and 1 output connection)
- Supported cycle times down to 31.25 µs
- Synchronization of application to sercos cycle
- Parameterization over service channel
- Hot-plug support
- Service channel with hardware support
- IP channel for ARP, Ping and firmware update over TFTP
- Design currently targeted to Xilinx Spartan-6 FPGA series
- FPGA reference design (Xilinx EDK V13.4) and Software-API (Xilinx MicroBlaze / gcc) available

Various EasySlave licensing models are available from sercos international. For sercos member companies special prices are offered. In addition, a license-free IP core version (loaded with a bitstream IP core) with a limited functionality is available, under the product name EasySlave-IO.

### 3 FPGA devices

The EasySlave-IO is designed for the XILINX Spartan-6 family. Actually two types of the Spartan-6 are supported.

FPGA device type	<b>XILINX XC6SLX16</b>	<b>XILINX XC6SLX25</b>
Package	FT256	FT256
Memory	48 KByte	64 KByte
Speed grade	-2	-2

Important:

*Also notice device datasheets and errata-sheets for additional information.*

#### 3.1 EasySlave FPGA resources and bit streams

Both devices are pin compatible and can be used for the EasySlave-IO. For programming the corresponding bit stream file must be chosen. The XC6SLX16 device is sufficient for simple applications like digital I/O. For comprehensive application or for development the larger XC6SLX25 device is recommended. Both FPGA devices are pin compatible, therefore only one PCB layout is required and the device migration is easy.

FPGA device type	<b>XILINX XC6SLX16-2FT(G)256</b>	<b>XILINX XC6SLX25-2FT(G)256</b>
Bit stream filename	S3_ESLV_BIT_XC6SLX[16 25]_DI16_DO16_Vx_y.bit	
COD filename	S3_ESLV_COD_XC6SLX[16 25]_DI16_DO16_Vx_y.bin	
Slice Registers	4553 of 18224 (24%)	4550 of 30064 (15%)
Slice LUTs	5864 of 9112 (64%)	5762 of 15032 (38%)
MUXCYs	588 of 4556 (12%)	588 of 7516 (7%)
IOBs	99 of 186 (53%)	99 of 186 (53%)
RAMB16WERs	30 of 32 (93%)	38 of 52 (73%)
BUFG	7 of 16 (43%)	7 of 16 (43%)
DSP48A1s	3 of 32 (9%)	3 of 38 (7%)
PLL_ADVs	1 of 2 (50%)	1 of 2 (50%)

x: Major version number

y: Minor version number

## 3.2 EasySlave-IO pinout

The FPGA has a fix pinout for the EasySlave-IO bitstream variant. Not all ports are required for the design.

		Spartan-6 XC6SLX16/25 FT(G)256 - top view																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
	A	GND	IO[16]	IO[25]	PHY_RESET	RXD_B[3]	PHY_LINK_B	TXD_B[1]	TX_EN_B	RX_CLK_B	STATE_D_GN	RXD_A[0]	RXD_A[3]	TxD_A[3]	TxD_A[1]	TMS	GND	A	
VCCINT (1,2V)	B	IO[20]	IO[17]	IO[27]	VCCO_0	RXD_B[1]	TXD_B[0]	GND	TXD_B[3]	VCCO_0	STATE_DL_E_DRD	GND	TX_EN_A	VCCO_0	TxD_A[0]	TST[1]	TST[0]	B	
VCCAUX (3,3V)	C	IO[21]	IO[15]	IO[14]	n.c.	LED_LINK_B	RXD_B[2]	RX_DV_B	TXD_B[2]	TX_CLK_B	TX_CLK_A	PHY_LINK_A	TDI	TxD_A[2]	TCK	A[0]	BA[0]	C	
VCCO (3,3V)	D	IO[22]	VCCO_3	IO[23]	GND	RXD_B[0]	LED_LINK_A	VCCO_0	RX_ER_B	S3_LED_RD	VCCO_0	RXD_A[1]	RXD_A[2]	GND	A[4]	VCCO_1	/CAS	D	
FPGA configuration	E	IO[11]	IO[10]	IO[28]	IO[29]	VCCAUX	LED_ACT_B	MDC	MDIO	GND	RX_CLK_A	RX_DV_A	IO[9]	n.c.	TDO	BA[1]	n.c.	E	
JTAG interface	F	UART_TX	UART_RX	IO[30]	IO[31]	IO[26]	IO[24]	LED_ACT_A	VCCAUX	S3_LED_G_N	RX_ER_A	VCCAUX	IO[8]	A[8]	DQ[8]	DQMH	A[3]	F	
	G	EXT[15]	GND	EXT[14]	VCCO_3	IO[18]	IO[19]	VCCINT	GND	VCCINT	VCCINT	CKE	A[9]	VCCO_1	DQ[7]	GND	A[2]	G	
Ethernet PHY Port A	H	EXT[13]	EXT[12]	IO[5]	IO[4]	IO[3]	VCCAUX	GND	VCCINT	GND	VCCINT	A[7]	GND	A[8]	DQ[4]	/RAS	A[1]	H	
Ethernet PHY Port B	J	EXT[11]	VCCO_3	EXT[10]	IO[1]	GND	IO[2]	VCCINT	GND	VCCINT	VCCINT	A[5]	A[12]	A[11]	CLK_25_MHZ	VCCO_1	A[10]	J	
	K	EXT[9]	EXT[8]	IO[0]	VCCO_3	IO[12]	IO[13]	GND	VCCINT	GND	VCCINT	CLK	CLK	VCCO_1	n.c.	DQML	CS	K	
IO ports	L	EXT[7]	GND	EXT[6]	IO[6]	IO[7]	VCCAUX	TSEG_B	DIP[4]	VCCAUX	n.c.	EE_CS	EE_SO	DQ[1]	GND	/WE		L	
Extension ports	M	EXT[5]	EXT[4]	SPI_MISO	SPI_MOSI	SPI_SCLK	TSEG_A	TSEG_C	GND	DIP[5]	n.c.	n.c.	EE_SCK	EE_SI	DQ[5]	DQ[6]		M	
	N	EXT[3]	VCCO_3	EXT[2]	SPI_LSD0	TSEG_F	TSEG_G	VCCO_2	LED[1]	n.c.	VCCO_2	M1	n.c.	GND	DQ[2]	VCCO_1	n.c.	N	
Memory	P	EXT[1]	EXT[0]	GND	DIP[1]	TSEG_D	DIP[3]	LED[0]	TSEG_P	n.c.	DIN	n.c.	n.c.	DONE_2	SUSPEND	DQ[14]	DQ[1]	P	
	R	SPI_CS2	SPI_CS1	INIT_B	VCCO_2	n.c.	GND	DIP[7]	VCCO_2	n.c.	GND	CCLK	DQ[11]	VCCO_1	DQ[0]	DQ[15]	DQ[3]	R	
not connected (n.c.)	T	GND	PROG_B	CSO_B	DIP[0]	n.c.	TSEG[4]	DIP[2]	DIP[6]	n.c.	CSI_B	M0	DQ[9]	DQ[12]	DQ[13]	/RESET	GND	T	
Steinbeis-Transferzentrum Systemtechnik EasySlave Pinout 07.11.2012		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		

### 3.2.1 Pin assignment

Following the detailed pin assignments are listed.

Signal	I/O	Location	Description
<b>Global signals</b>			
CLK_25MHZ	I	J14	Device clock nominal 25 MHz
/RESET	I	T15	Device reset, low-active
<b>Program-Interface</b>			
PROG_B	I	T2	
INIT_B	I	R3	
SUSPEND	I	P14	
DONE	O	P13	
M0	I	T11	FPGA configuration pin
M1	I	N11	FPGA configuration pin
<b>JTAG-Interface</b>			
TDI	I	C12	JTAG serial data input
TDO	O	E14	JTAG serial data output
TCK	I	C14	JTAG clock
TMS	I	A15	JTAG mode
<b>SPI-Flash-Interface</b>			
Flash_CS (CSO_B)	O	T3	SPI Flash Chip select
Flash_CLK (CCLK)	O	R11	SPI Flash Clock
Flash_DI (DIN)	O	P10	SPI Flash Data Output
Flash_DO (CSI_B)	I	T10	SPI Flash Data Input
<b>SIII LED</b>			
S3_LED_GN	O	F9	green LED
S3_LED_RD	O	D9	red LED
<b>Status LED</b>			
STATE_LED_GN	O	A10	green LED
STATE_LED_RD	O	B10	red LED
<b>Output LED</b>			
LED[0]	O	P7	yellow LED >> System active
LED[1]		N8	red LED >> System reboot required (e.g. after update)
<b>Seven segment display</b>			
7SEG_A	O	M6	Segment A
7SEG_B		L7	Segment B
7SEG_C		M7	Segment C
7SEG_D		P5	Segment D
7SEG_E		T6	Segment E
7SEG_F		N5	Segment F
7SEG_G		N6	Segment G
7SEG_P		P8	Segment dot
<b>Dip switch (sercos address)</b>			
DIP[0]	I (PD)	T4	Dip switch ( $2^0$ )
DIP[1]		P4	Dip switch ( $2^1$ )
DIP[2]		T7	Dip switch ( $2^2$ )
DIP[3]		P6	Dip switch ( $2^3$ )
DIP[4]		L8	Dip switch ( $2^4$ )
DIP[5]		M9	Dip switch ( $2^5$ )
DIP[6]		T8	Dip switch ( $2^6$ )
DIP[7]		R7	Dip switch ( $2^7$ )
<b>sercos Testpins</b>			
TST[0]	O	B16	sercos test pin (TS1)
TST[1]		B15	sercos test pin (TS2)

I – input

O – output

I/O – input/output

PD – pull-down

Signal	I/O	Location	Description
<b>Port 1 Phy signals</b>			
RXD_A0	I	A11	Receive data nibble from PHY port 1.
RXD_A1		D11	
RXD_A2		D12	
RXD_A3		A12	
TXD_A0	O	B14	Transmit data nibble from PHY port 1.
TXD_A1		A14	
TXD_A2		C13	
TXD_A3		A13	
RX_DV_A	I	E11	Receive data valid port 1.
RX_ER_A	I	F10	Receive error port 1.
RX_CLK_A	I	E10	Receive clock port 1.
TX_EN_A	O	B12	Transmitter enable port 1.
TX_CLK_A	I	C10	Transmit clock port 1.
PHY_LINK_A	I	C11	Link signal from Phy
LED_LINK_A	O	D6	Low-active LED1 of connector port 1, link info – green
LED_ACT_A	O	F7	Low-active LED2 of connector port 1, activity info – yellow
MDIO	I/O	E8	Data signal of management interface
MDC	O	E7	Clock signal of management interface
PHY_RESET	O	A4	Low-active reset signal for Phy
<b>Port 2 Phy signals</b>			
RXD_B0	I	D5	Receive data nibble from PHY port 1.
RXD_B1		B5	
RXD_B2		C6	
RXD_B3		A5	
TXD_B0	O	B6	Transmit data nibble from PHY port 1.
TXD_B1		A7	
TXD_B2		C8	
TXD_B3		B8	
RX_DV_B	I	C7	Receive data valid port 1.
RX_ER_B	I	D8	Receive error port 1.
RX_CLK_B	I	A9	Receive clock port 1.
TX_EN_B	O	A8	Transmitter enable port 1.
TX_CLK_B	I	C9	Transmit clock port 1.
PHY_LINK_B	I	A6	Link signal from Phy
LED_LINK_B	O	C5	Low-active LED1 of connector port 1, link infor – green
LED_ACT_B	O	E6	Low-active LED2 of connector port 1, activity info – yellow

I – input

O – output

I/O – input/output

PD – pull-down

Signal	I/O	Location	Description
<b>I/O Ports IO[31:0]</b>			
IO0	I/O	K3	Used for digital I/O Ports
IO1		J4	
IO2		J6	
IO3		H5	
IO4		H4	
IO5		H3	
IO6		L4	
IO7		L5	
IO8		F12	
IO9		E12	
IO10		E2	
IO11		E1	
IO12		K5	
IO13		K6	
IO14		C3	
IO15		C2	
IO16		A2	
IO17		B2	
IO18		G5	
IO19		G6	
IO20		B1	
IO21		C1	
IO22		D1	
IO23		D3	
IO24		F6	
IO25		A3	
IO26		F5	
IO27		B3	
IO28		E3	
IO29		E4	
IO30		F3	
IO31		F4	
<b>Extension Ports EXT[15:0]</b>			
EXT0	I/O (PD)	P2	Used for special digital I/O Ports (e.g. diagnosis)  EasySlave-IO: EXT0 >> General_IO_Error A (error active high) EXT1 >> General_IO_Error B (error active high) EXT2 >> Usupply_Error A (error active low) EXT3 >> Usupply_Error B (error active low)
EXT1		P1	
EXT2		N3	
EXT3		N1	
EXT4		M2	
EXT5		M1	
EXT6		L3	
EXT7		L1	
EXT8		K2	
EXT9		K1	
EXT10		J3	
EXT11		J1	
EXT12		H2	
EXT13		H1	
EXT14		G3	
EXT15		G1	

I – input

O – output

I/O – input/output

PD – pull-down

Signal	I/O	Location	Description
<b>SD-RAM interface (not available for EasySlave-IO)</b>			
A[0]	O	C15	Memory address
A[1]		H16	
A[2]		G16	
A[3]		F16	
A[4]		D14	
A[5]		J11	
A[6]		F13	
A[7]		H11	
A[8]		H13	
A[9]		G12	
A[10]		J16	
A[11]		J13	
A[12]		J12	
BA[0]	O	C16	Memory bank address
BA[1]		E15	
DQ[0]	I/O	R14	Memory data
DQ[1]		L14	
DQ[1]		P16	
DQ[2]		N14	
DQ[3]		R16	
DQ[4]		H14	
DQ[5]		M15	
DQ[6]		M16	
DQ[7]		G14	
DQ[8]		F14	
DQ[9]		T12	
DQ[11]		R12	
DQ[12]		T13	
DQ[13]		T14	
DQ[14]		P15	
DQ[15]		R15	
DQMH	O	F15	Byte select high
DQML	O	K15	Byte select low
/CAS	O	D16	Memory column select
/RAS	O	H15	Memory row select
/WE	O	L16	Memory read/write
CKE	O	G11	Memory clock enable
CLK	O	K11	Memory clock
CLK_FB	O	K12	Memory clock (feedback)
CS	O	K16	Memory chip select

I – input

O – output

I/O – input/output

PD – pull-down

Signal	I/O	Location	Description
<b>IO SPI interface (not available for EasySlave-IO)</b>			
SPI_CS[0]	O	N4	SPI Flash Chip select 1
SPI_CS[1]	O	R2	SPI Flash Chip select 2
SPI_CS[2]	O	R1	SPI Flash Chip select 3
SPI_MISO	I	M3	SPI Flash Clock
SPI_MOSI	O	M4	SPI Flash Data Output
SPI_SCLK	O	M5	SPI Flash Data Input
<b>IO UART interface (not available for EasySlave-IO)</b>			
UART_RX	I	F2	SPI Flash Chip select
UART_TX	O	F1	SPI Flash Clock
<b>EEProm (not available for EasySlave-IO)</b>			
EE_CS	O	L12	SPI EEPROM Chip select
EE_SCK	O	M13	SPI EEPROM Clock
EE_SI	O	M14	SPI EEPROM data in
EE_SO	I	L13	SPI EEPROM data out

Signal	I/O	Location	Description
<b>Output SUP voltage</b>			
VCCO_0		B4	Output supply voltage bank 0, has to be connected to 3,3V
VCCO_0		B9	
VCCO_0		B13	
VCCO_0		D7	
VCCO_0		D10	
VCCO_1		D15	Output supply voltage bank 1, has to be connected to 3,3V
VCCO_1		G13	
VCCO_1		J15	
VCCO_1		K13	
VCCO_1		N15	
VCCO_1		R13	
VCCO_1		N15	
VCCO_1		R13	
VCCO_2		N7	Output supply voltage bank 2, has to be connected to 3,3V
VCCO_2		N10	
VCCO_2		R4	
VCCO_2		R8	
VCCO_3		D2	Output supply voltage bank 3, has to be connected to 3,3V
VCCO_3		G4	
VCCO_3		J2	
VCCO_3		K4	
VCCO_3		N2	
<b>Auxiliary SUP voltage</b>			
VCCAUX		E5	Auxiliary supply voltage, connect to 3,3V
VCCAUX		F8	
VCCAUX		F11	
VCCAUX		G10	
VCCAUX		H6	
VCCAUX		J10	
VCCAUX		L6	
VCCAUX		L9	

I – input

O – output

I/O – input/output

PD – pull-down

SUP – power supply

Signal	I/O	Location	Description
<b>Internal SUP voltage</b>			
VCCINT	SUP	G7	Internal supply voltage, connect to 1,2V
VCCINT		G9	
VCCINT		H8	
VCCINT		H10	
VCCINT		J7	
VCCINT		J9	
VCCINT		K8	
VCCINT		K10	
<b>Ground</b>			
GND	SUP	A1	Internal supply voltage, connect to 0V (GND)
GND		A16	
GND		B7	
GND		B11	
GND		D4	
GND		D13	
GND		E9	
GND		G2	
GND		G8	
GND		G15	
GND		H7	
GND		H9	
GND		H12	
GND		J5	
GND		J8	
GND		K7	
GND		K9	
GND		L2	
GND		L15	
GND		M8	
GND		N13	
GND		P3	
GND		R6	
GND		R10	
GND		T1	
GND		T16	

I – input

O – output

I/O – input/output

PD – pull-down

SUP – power supply

The following pins are not connected:

C4, E13, E16, K14, L10, L11, M10, M11, M12, N9, N12, N16, P9, P11, P12, R5, R9, T5, T9

## 4 Short Description of the FPGA structure

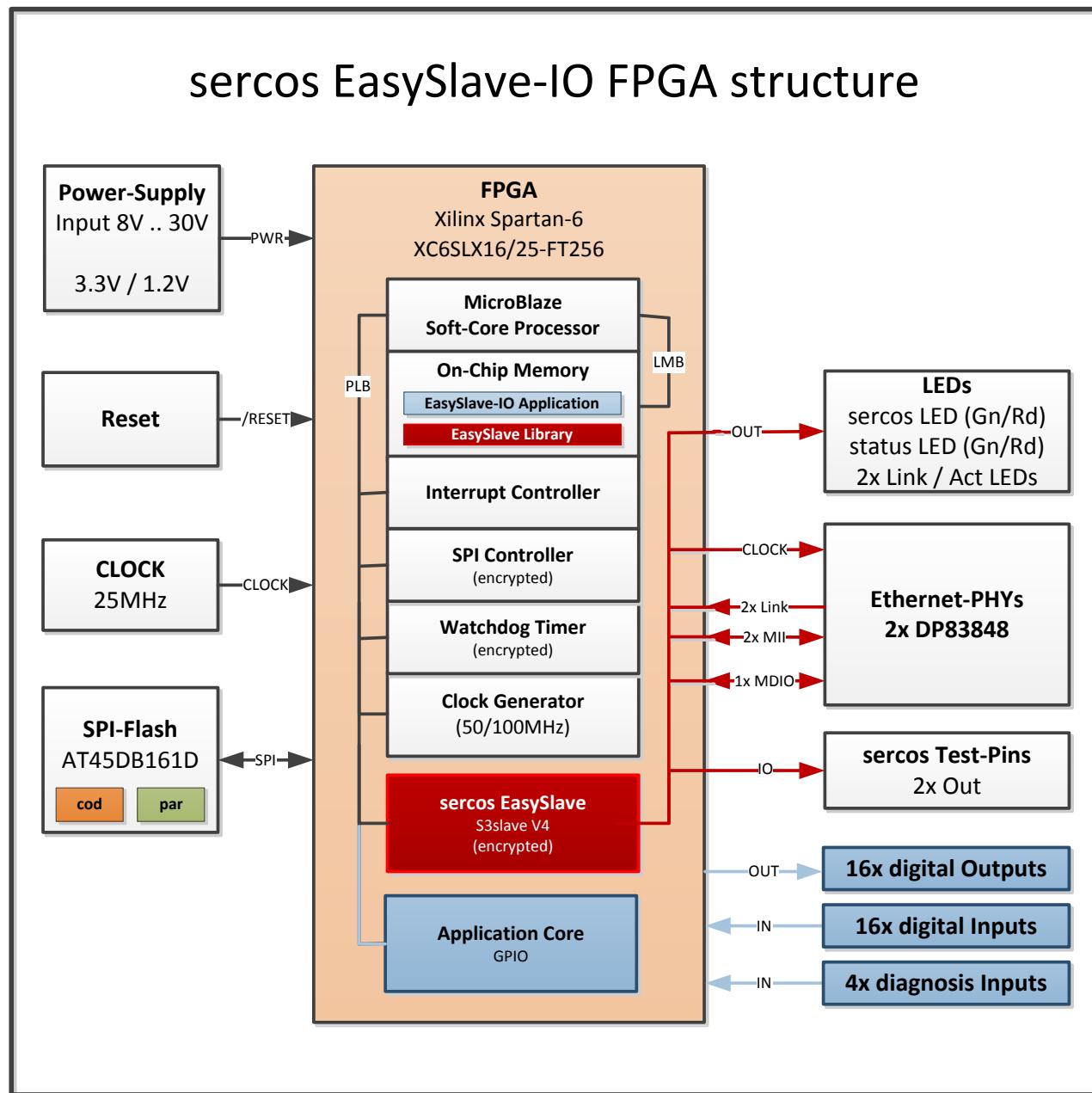
This chapter describes the internal device structure of the EasySlave-IO.

The sercos EasySlave-IO is based on a Xilinx Spartan-6 FPGA XC6SLX16/25 which contains the SERCON100 EasySlave core. It is responsible for sercos III communication and I/O data handling.

The SERCON100 EasySlave core contains the sercos III logic core and a soft-core processor, called "MicroBlaze" (MB) for firmware handling. The whole program and data are stored in the internal FPGA RAM blocks (on-chip memory). There is no additional external program memory for firmware or data. The sercos interface is realized according to sercos specification V1.3.

The MB communicates with several modules over PLB (Processor Local Bus) in order to support the sercos III interface and IO handling:

Hardware Modules	
On-chip Memory	internal RAM for firmware and data which is tightly coupled to the processor
Interrupt Controller	handles multiple interrupts
SPI Controller	accesses the SPI flash
Clock Generator	generates multiple clock signals (PLL) <ul style="list-style-type: none"> <li>- MB (100 MHz)</li> <li>- S3 EasySlave core (50 MHz)</li> </ul>
Watchdog	Hardware watchdog for monitoring the firmware
S3 EasySlave	processes sercos telegrams
GPIO IN / OUT	interface for I/O ports <ul style="list-style-type: none"> <li>- digital I/O</li> <li>- diagnosis inputs</li> <li>- LEDs</li> <li>- seven segment display</li> <li>- switches</li> </ul>



<sup>1</sup>PLB: Processor Local Bus

<sup>2</sup>LMB: Local Memory Bus

sercos EasySlave

Application specific

cod      FPGA configuration data (firmware + application)

par      Non-volatile parameters data

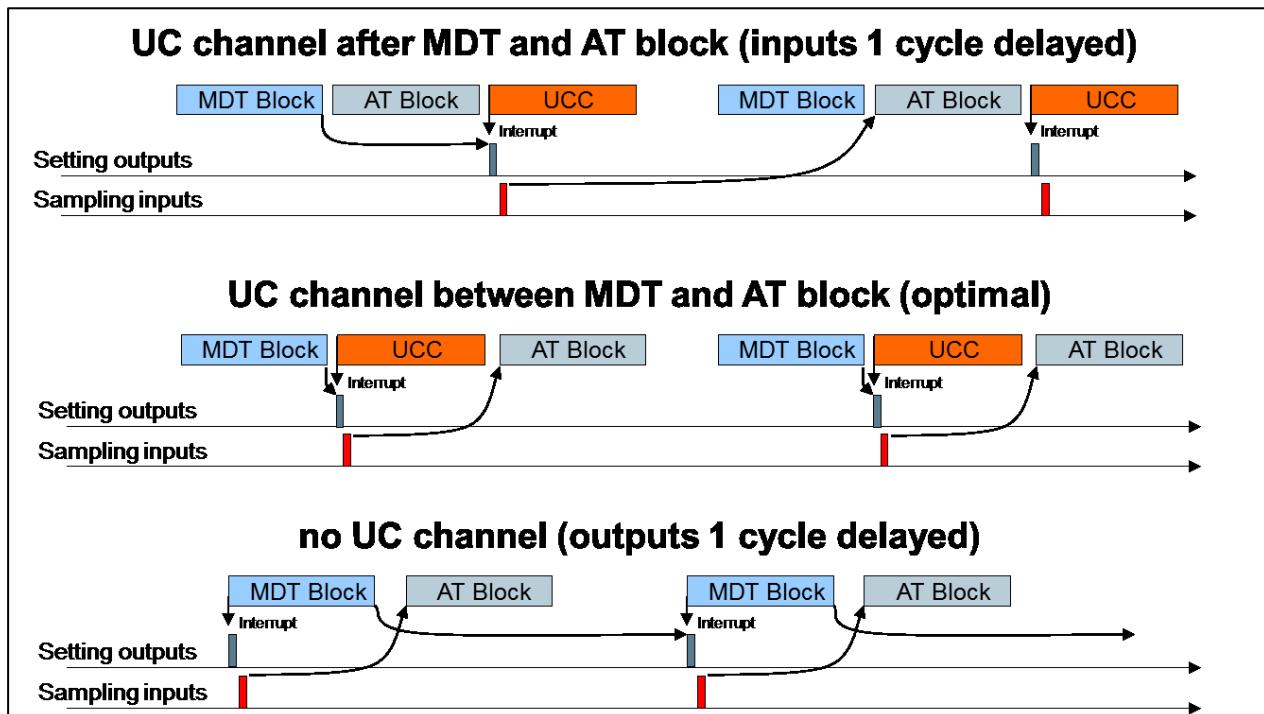
## 5 EasySlave IO processing

### 5.1 Cyclic IO communication in CP4

The EasySlave is synchronized to the sercos bus cycle. There are two types of IO processing:

1. If the UC channel is configured, the interrupt for IO processing is triggered at the start of the UC channel (t6).
2. If no UC channel is configured, the interrupt for IO processing is triggered with the start of the communication cycle MDT0.

Depending of the cycle configuration the timing of setting the outputs and sampling the inputs is varying.



#### 5.1.1 Output processing

The EasySlave-IO provides 2 byte output data. After receiving the output data via the realtime consuming connection the extracted is done by the logic core and stored immediately in the receive buffer. If the interrupt for the IO processing occurred the output values are read out of the receive buffer and transferred directly to the output ports.

The connection which is consumed by the slave, and uses structure instance 1 (S-0-1050.1.y) may either be placed in any MDT or AT. The position of this connection is not dependent on the place where device control is placed. The content of the connections is defined by the slave and cannot be changed by the master.

#### 5.1.2 Input processing

The EasySlave-IO provides 2 byte input data. Within the IO processing interrupt the input data are transferred to sending buffer of the logic core. The logic core insert the data at the next ATx telegram into the producing connection, this mechanism guarantees data integrity.

The connection which is produced by the slave may be placed in any AT, and uses structure instance 0 (S-0-1050.0.y). The position of this connection is not dependent on the place where device status is placed.

### 5.1.3 Realtime data communication

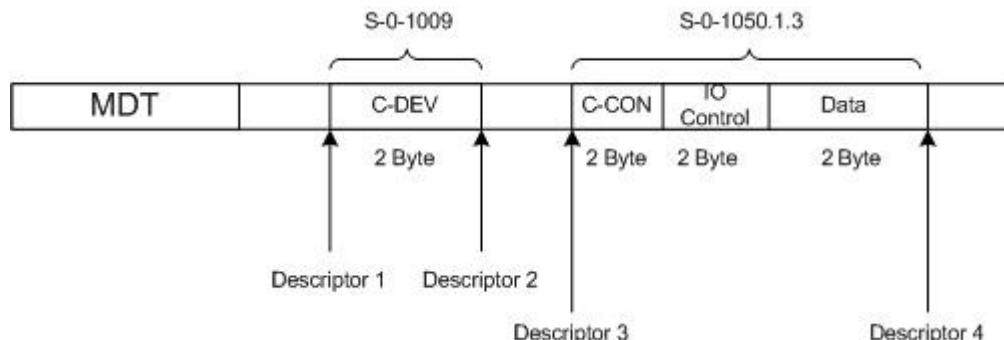
The EasySlave-IO supports the SCP\_FixCFG class where the configuration of the connections is fixed. The data insertion and extraction of real-time data is controlled by several parameters.

- S-0-1009 Device control offset in MDT
- S-0-1011 Device Status offset in AT
- S-0-1050.0.3 Telegram Assignment AT
- S-0-1050.1.3 Telegram Assignment MDT

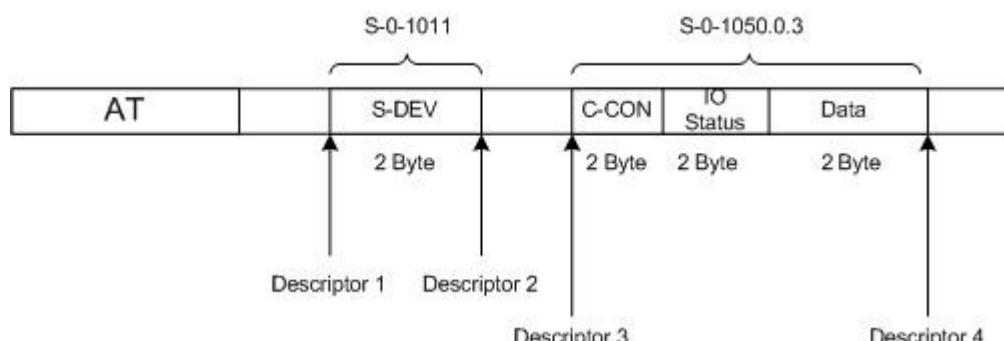
S-0-1009 and S-0-1011 describe the position of C-DEV / S-DEV. The parameters S-0-1050.0.3 and S-0-1050.1.3 describe the position of the connection (C-CON, IO control/status and RT-Data).

The following two illustrates this relation:

#### Consumer data out of a MDT or AT (CC)



#### Producer data into an AT



Further information about the profiles can be found at the sercos Wiki website.

## 5.1.4 IO Bit assignment

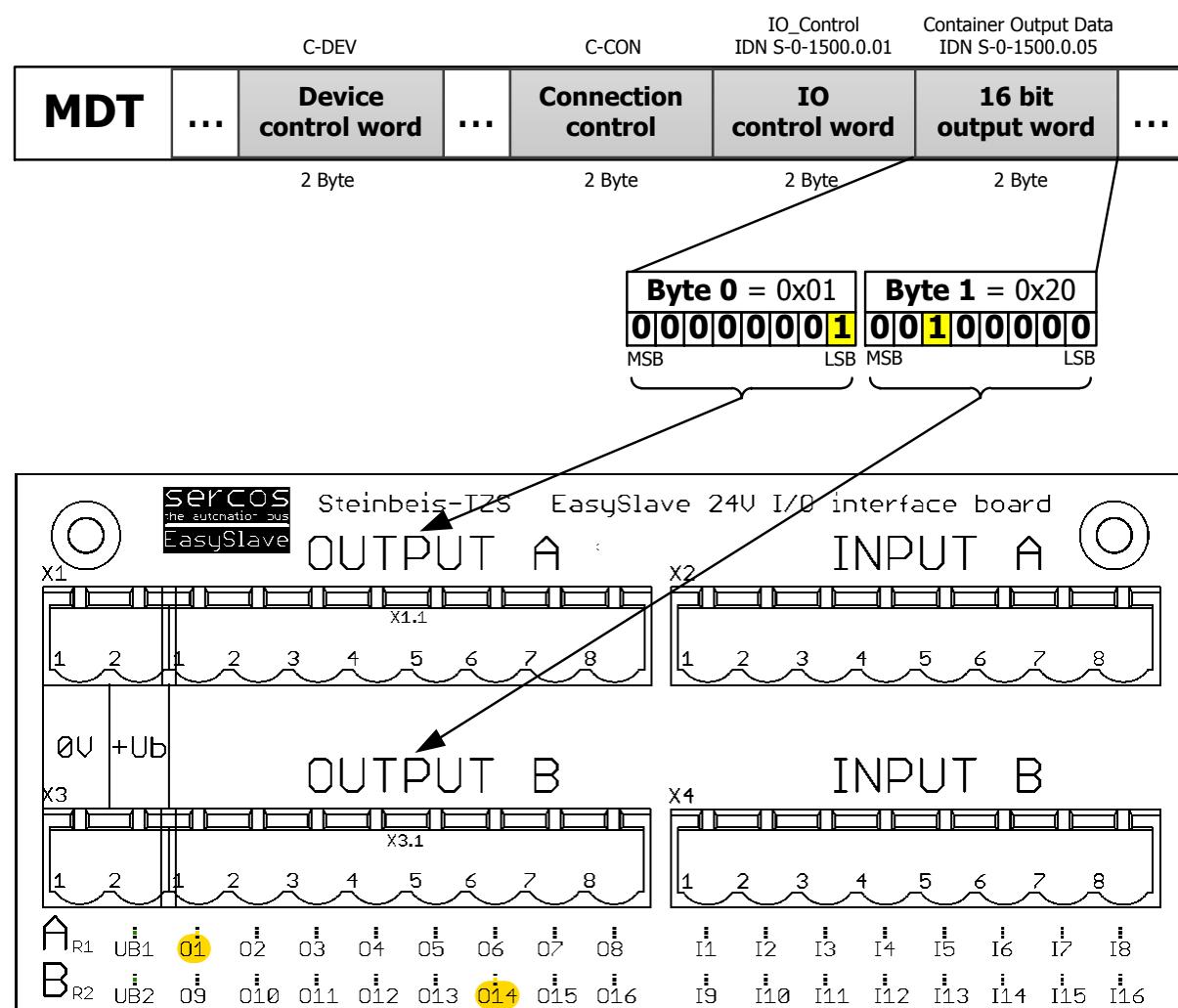
The connection data bytes are mapped to the physical IO ports of the slave.

### 5.1.4.1 Outputs

The bytes are transferred in sercos III telegrams in **LittleEndian** format (LSByte first). The bit assignment is identical for non-cyclic data (over SVC) and cyclic data (RTD). The outputs are accessible via the IDN S-0-1500.x.05 Container OutputData.

Container OutputData Assignment																	
Byte 0								Byte 1									
MSB				LSB				MSB				LSB					
08	07	06	05	04	03	02	01	016	015	014	013	012	011	010	09		

Example (RTD):

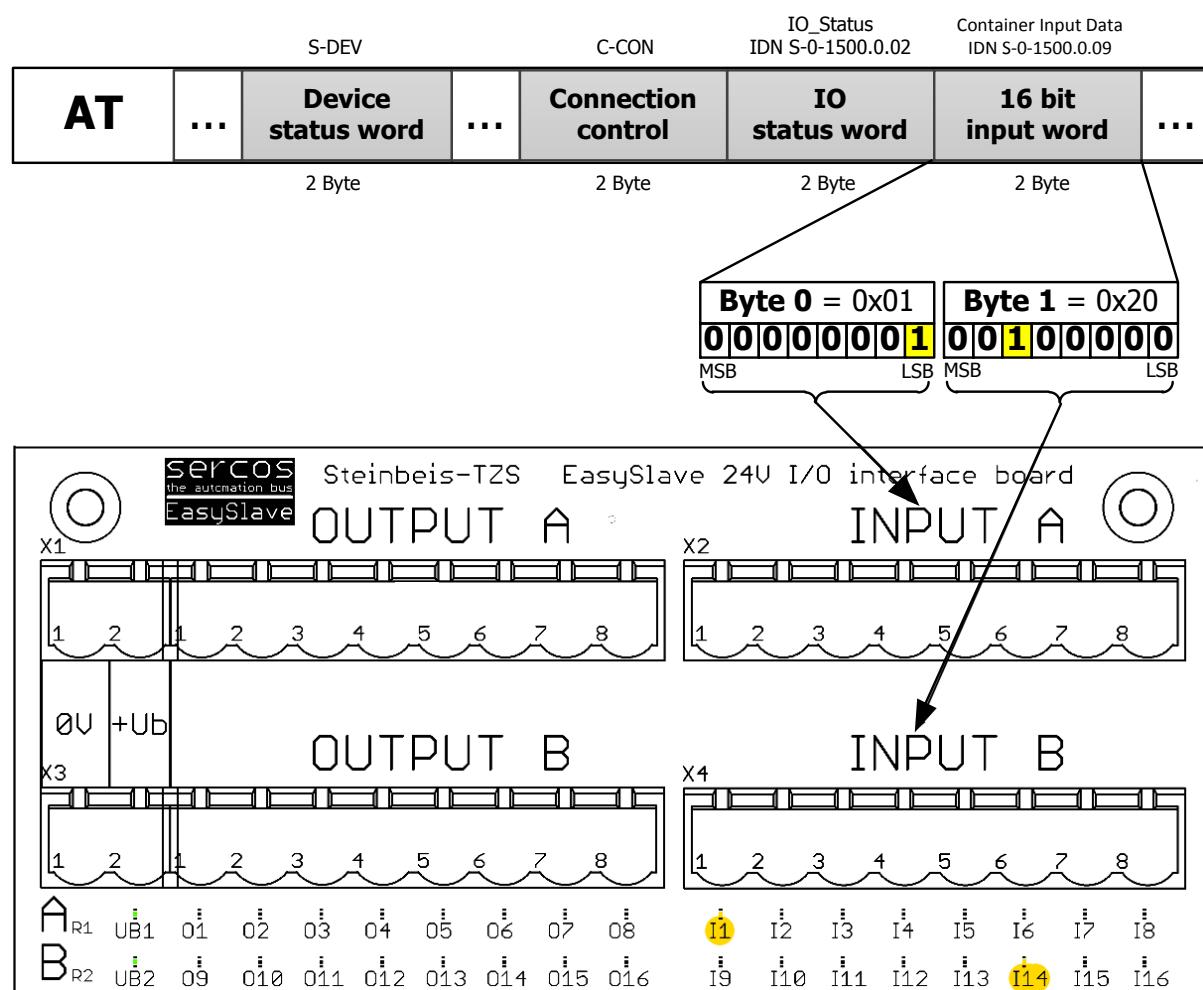


### 5.1.4.2 Inputs

The bit assignment is identical for non-cyclic data (over SVC) and cyclic data (RTD).  
The inputs are accessible via the IDN S-0-1500.x.09 Container InputData.

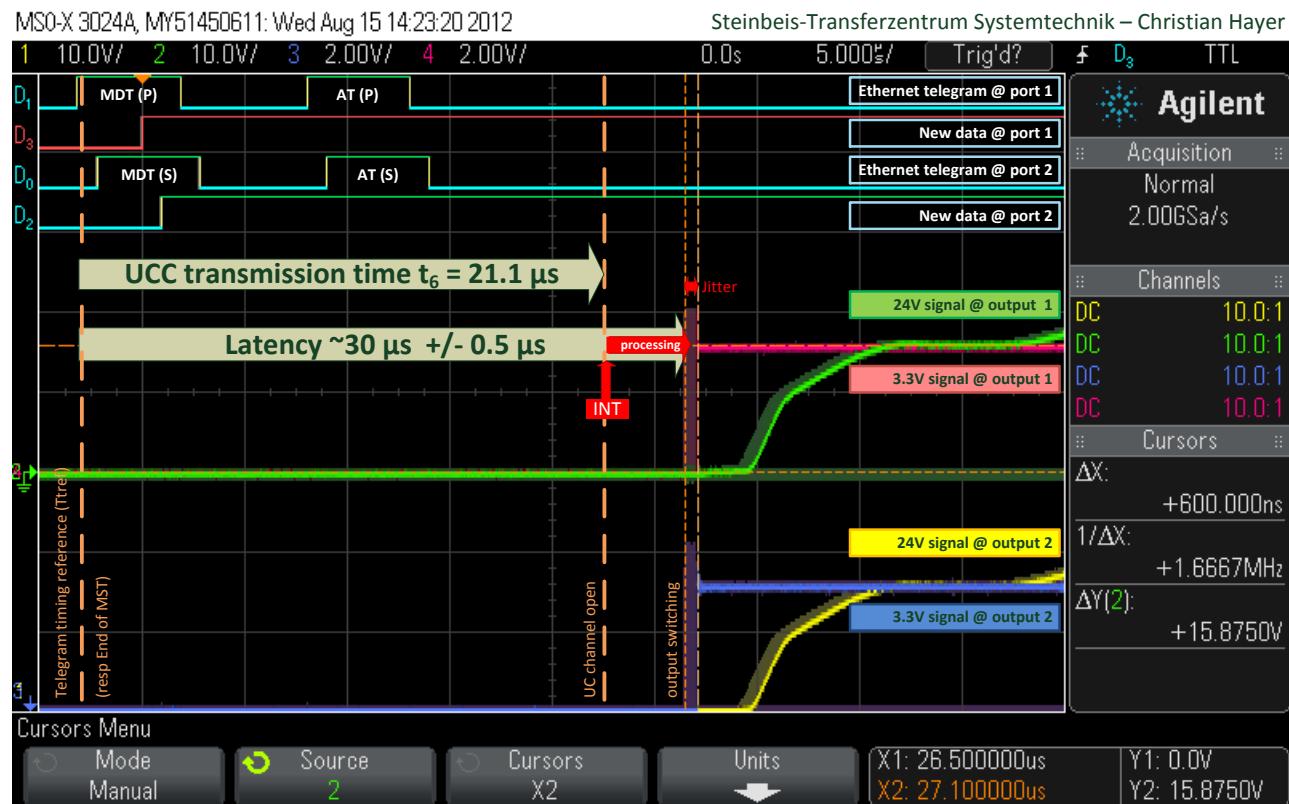
Container InputData Assignment																
Byte 0								Byte 1								
MSB							LSB	MSB								LSB
I8	I7	I6	I5	I4	I3	I2	I1	I16	I15	I14	I13	I12	I11	I10	I9	

Example (RTD):



### 5.1.5 Output timing

If the telegram (MDTx or ATx), in which the output data for the EasySlave-IO slave arrives, the data are stored in the receiving buffer. At the start of the UC channel the IO processing interrupt is generated which reads the output data from the buffer and writes it to the output ports. The FPGA internal processing requires around **9 µs** for output changes at the FPGA pins. Because the sercos telegrams are parsed by the logic core, the output jitter is with **+/- 0.5 µs** very low.



#### Testconditions:

An EasySlave-IO evaluation system (one slave) which is controlled by a SERCANS PC card (sercos conformizer test scripts). The cycle time is 1 ms and the topology is double-ring. Ethernet timing measurement with the Ethernet-Monitor (RZA-EE<sub>2</sub>IOSy) from Distributed Systems Engineering and an Agilent MSO-X 3024A oscilloscope.

The output switching of the 24V outputs of the EasySlave-IO Evaluation Kit are not symmetrically caused by the used high-side switched output driver.

There are two cases:

- from low (0V) to high (24V) output
- from high (24V) to low (0V) output

## 5.2 Diagnosis and Error management

The EasySlave-IO provides diagnosis functions for the IO ports.

### 5.2.1 Sub-device errors

#### 5.2.1.1 Supply error

The EasySlave-IO has two input pins for voltage supply supervising,

- Usupply\_Error A
- Usupply\_Error B

The input pins are monitored permanently. If a low-condition at the input is detected the error bit (C1D) in the device status is set and the error code *0xC10F1000* is inserted in the diagnostic number (S-0-0390).

#### 5.2.1.2 I/O errors

The EasySlave-IO has two inputs for I/O errors.

- General\_IO\_Error A
- General\_IO\_Error B

for supervising I/O malfunctions like

- Output short circuit, or
- Output driver over temperature.

The input pins are monitored permanently. If a high-condition at the input is detected the error bit (C1D) is set and the error code *0xC10F3400* is inserted in diagnostic number (S-0-0390).

#### 5.2.1.3 Error handling and error flag reset

If an error occurs the slave sets the corresponding bit (Bit 7) in the device status word and the IO status. There is only one bit for all errors designated. Every error has its own error number. The error number and the error type can be read out of parameter S-0-0390. If more than one error occurs the error number of the highest priority error is stored in S-0-0390.

Following error codes are possible:

- 0xC10F1000 (General I/O error)
- 0xC10F3400 (Supply error)

The device status error flag and IO status flag can be resetted by the SIII master with the command S-0-0099 if no error is pending.

The EasySlave-IO has no IO warnings defined.

## 5.2.2 Output fallback operation

With a switch to CP0 or NRT state, all outputs are reset to the fallback value (output low).

## 6 Firmware description

The EasySlave firmware consists of an EasySlave-Library (LIB) and the application. The LIB provides the interface to the logic core and the main sercos functions.

### 6.1 sercos III version

The firmware controls sercos III handling. It is implemented according to sercos III Spec V1.3

### 6.2 EasySlave parameters

sercos contains many predefined parameters. They support easy configuration and operation.

Central element for EasySlave parameters is the parameter database.

The parameter database has three departments according to the profiles

- sercos communication profile (SCP)
- generic device profile (GDP)
- IO function specific profile (FSP IO)

The individual parameters, which are supported by the sercos EasySlave, are described in the following sections.

### **6.2.1 Communication parameters (SCP)**

<b>Ident</b>	<b>Read/ Write</b>	<b>Non- volatile</b>	<b>Initialization</b>	<b>Name</b>
S-0-0021	R	No	CP2	IDN-list of invalid operation data for CP2
S-0-0022	R	No	CP2	IDN-list of invalid operation data for CP3
S-0-0127	W	No	CP2	CP3 transition check
S-0-0128	W	No	CP3	CP4 transition check
S-0-1000	R	Yes	NRT	SCP Type & Version
S-0-1002	RW	No	CP2	Communication Cycle time (tScyc)
S-0-1003	RW	No	CP2	Communication Timeout for CP3/CP4
S-0-1009	RW	No	CP2	Device Control offset in MDT
S-0-1010	RW	No	CP2	Lengths of MDTs
S-0-1011	RW	No	CP2	Device Status offset in AT
S-0-1012	RW	No	CP2	Length of Ats
S-0-1013	RW	No	CP2	SVC Offset in MDT
S-0-1014	RW	No	CP2	SVC Offset in AT
S-0-1017	RW	No	CP0	UC transmission time
S-0-1019	RW	No	CP0	MAC address
S-0-1020	RW	No	CP0	IP address
S-0-1021	RW	No	CP0	Subnet Mask
S-0-1022	RW	No	CP0	Gateway address
S-0-1026	R	Yes	CP0	Version of communication hardware
S-0-1027.0.1	R	Yes	CP0	Requested MTU
S-0-1027.0.2	R	Yes	CP0	Effective MTU
S-0-1031	RW	No	CP0	Test pin assignment Port 1 & Port 2
S-0-1034	R	No	CP0	PHY error counter Port 1 & Port 2
S-0-1035.0.0	R	No	CP0	Error counter Port 1 & Port 2
S-0-1035.0.1	R	No	CP0	Error counter (P/S)
S-0-1037	R	No	CP0	Slave Jitter

S-0-1039	RW	No	CP0	Hostname
S-0-1040	RW	No	CP0	sercos Address
S-0-1042	RW	No	CP0	Topology index
S-0-1044	RW	No	CP0	Device Control (C-DEV)
S-0-1045	RW	No	CP0	Device Status (S-DEV)
S-0-1050.0.3	RW	No	CP2	Telegram Assignment AT
S-0-1050.0.5	RW	No	CP2	Actual Length of Connection (AT)
S-0-1050.0.8	RW	No	CP2	Connection Control (C-CON) (AT)
S-0-1050.0.9	RW	No	CP2	Connection State (AT)
S-0-1050.1.3	RW	No	CP2	Telegram Assignment MDT
S-0-1050.1.5	RW	No	CP2	Actual Length of Connection (MDT/AT)
S-0-1050.1.8	RW	No	CP2	Connection Control (C-CON) (MDT/AT)
S-0-1050.1.9	RW	No	CP2	Connection State (MDT/AT)

## 6.2.2 Device identification (GDP)

<b>Ident</b>	<b>Read/ Write</b>	<b>Non- volatile</b>	<b>Initialization</b>	<b>Name</b>
S-0-0017	RW	Yes	NRT	IDN-list of all operation data
S-0-0099	W	No	NRT	Reset class 1 diagnostic
S-0-0390	RW	No	NRT	Diagnostic number
S-0-0420	RW	No	CP0	Activate parameterization level procedure command
S-0-0422	RW	No	CP0	Exit parameterization level procedure command
S-0-0423	R	Yes	CP0	IDN list of invalid data for parameterization level
S-0-0425	W	No	CP3	Sub-device state machine control
S-0-1300.0.1	R	Yes	NRT	Component name
S-0-1300.0.2	R	Yes	NRT	Vendor Name
S-0-1300.0.3	R	Yes	NRT	Vendor code
S-0-1300.0.4	R	Yes	NRT	Device Name
S-0-1300.0.5	R	Yes	NRT	Vendor Device ID
S-0-1300.0.7	R	Yes	NRT	Function Revision
S-0-1300.0.8	R	Yes	NRT	Hardware Revision
S-0-1300.0.9	R	Yes	NRT	Software Revision
S-0-1300.0.10	R	Yes	NRT	Firmware Loader Revision
S-0-1300.0.11	R	Yes	NRT	Order Number
S-0-1300.0.12	R	Yes	NRT	Serial number
S-0-1300.0.13	R	Yes	NRT	Manufacturing Date
S-0-1300.0.14	R	Yes	NRT	QA Date
S-0-1300.0.21	R	Yes	NRT	Service Date
S-0-1300.0.22	R	Yes	NRT	Calibration Date
S-0-1300.0.23	R	Yes	NRT	Calibration Due Date
S-0-1301	R	Yes	NRT	GDP Type & Version
S-0-1302.0.1	R	Yes	NRT	FSP Type & Version
S-0-1302.0.2	R	Yes	NRT	Function Groups
S-0-1399.0.1	W	No	NRT	Test IDN Diagnostic Event

### 6.2.3 IO parameters (FSP IO)

Ident	Read/ Write	Non- volatile	Initialization	Name
S-0-1500.0.1	RW	No	CP0	IO_Control
S-0-1500.0.2	R	No	CP0	IO_Status
S-0-1500.0.3	R	Yes	NRT	Module type code
S-0-1500.0.5	RW	Yes	NRT	Container Output Data
S-0-1500.0.9	R	Yes	NRT	Container Input Data
S-0-1502.0.3	R	Yes	NRT	Channel Amount PDOUT
S-0-1502.0.4	R	Yes	NRT	Channel Width PDOUT
S-0-1502.0.5	RW	No	CP0	PDOUT
S-0-1503.0.7	R	Yes	NRT	Channel Amount PDIN
S-0-1503.0.8	R	Yes	NRT	Channel Width PDIN
S-0-1503.0.9	R	No	CP0	PDIN

## 6.2.4 SDDML

SDDML stands for sercos Device Description Markup Language. A SDDML file describes a sercos III device for offline configuration. The structure of the document is according to the sercos III device model. Within the document there are references to the SCP, GDP and FSP that are implemented by the device described.

### Example EasySlave-IO SDDML

```

<?xml version="1.0" encoding="UTF-8"?>
<Device xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\SDDML.xsd">
  <SDDML_Version>
    <MajorRevision>2</MajorRevision>
    <MinorRevision>1</MinorRevision>
  </SDDML_Version>
  <DeviceInfo>
    <UserName lang="en"/>
    <UserText lang="en"/>
    <Pictures/>
  </DeviceInfo>
  <FileInfo>
    <MajorRevision>1</MajorRevision>
    <MinorRevision>0</MinorRevision>
  </FileInfo>
  <DeviceStructure>
    <SERCOSIII_Interface>
      <DeviceType>
        <Slave/>
      </DeviceType>
      <SubDevice>
        <SCP SERCOSIIIProfileRevisionNumber="1" ProfileName="SCP_FixCFG"/>
        <GDP SERCOSIIIProfileID="11" SERCOSIIIProfileRevisionNumber="1"
             ProfileName="GDP_Basic">
          </GDP>
          <FSP SERCOSIIIProfileRevisionNumber="1" ProfileName="FSP_IO">
            <IOModul Slot="0" Name="Digital Input Output"
                  ModuleTypeCode="0000000000000000">
              <IOFG Removable="false" IO_type="IO Bus Coupler">
                </IOFG>
              <IOFG Removable="false" IO_type="Digital Output">
                <PDOUT IDN="S-0-1502.0.5">
                  <Quantity IDN="S-0-1502.0.3">16</Quantity>
                  <width IDN="S-0-1502.0.4">1</width>
                </PDOUT>
              </IOFG>
              <IOFG Removable="false" IO_type="Digital Input">
                <PDIN IDN="S-0-1503.0.9">
                  <Quantity IDN="S-0-1503.0.7">16</Quantity>
                  <width IDN="S-0-1503.0.8">1</width>
                </PDIN>
              </IOFG>
            </IOModul>
            <ProprietaryConstraint UserParameterEditable="false"/>
          </FSP>
        </SubDevice>
        <ElectronicLabel>
          <ComponentName>sercos III EasySlave-IO</ComponentName>
          <VendorName>Steinbeis-TZS</VendorName>
          <VendorCode>1337</VendorCode>
          <DeviceName>EasySlave-IO 16E/16A Bitstream</DeviceName>
          <DeviceID>0001</DeviceID>
          <Revision>0</Revision>
          <OrderNumber>EasySlave-IO Evaluation Kit</OrderNumber>
        </ElectronicLabel>
      </SERCOSIII_Interface>
    </DeviceStructure>
  </Device>

```

## 6.3 Hot-plug

The hot-plug mechanism provides the ability to bring devices with one or more slaves (multi-slave device) subsequently to CP4. The running S III network with the participant remains in CP4. Hot-plugging is supported in CP4 with line topology only. With ring topology, a ring break has to be initiated first.

The last slave in a line topology shall continuously monitor its inactive port. If an additional device gets connected it receives all telegrams from the last slave.

The Master shall be prepared for the HP slaves, this means the telegram fields used by the hot-plug slaves shall be configured in the MDT and AT. In case of a broken ring topology, the master should activate the Hot-plug function on one channel only (P or S).

For a sercos III slave there are two different ways to participate in the cyclic communication of CP4. The "communication phase run-up" with the sequence from CP0 to CP4 and the "Hot-Plug procedure" with the sequence from HP0 to HP2 and switching to CP4.

An additional device shall start in the NRT state ("store and forward") as described Non-real-time state. If the additional device supports the sercos III protocol, it shall evaluate the hot-plug field in the MDT0. Using the hot-plug fields in MDT0 and AT0, the master and the hot-plug slave shall be able to communicate. After processing through the hot-plug procedure, the hot-plug slave shall become the last slave within the line topology.

The hot-plug procedure is divided in 3 phases (HP0, HP1 and HP2).

## 6.4 S/IP Services

UDP based S/IP requests are generally transmitted to UDP server port 35021 (0x88cd). The client shall use a socket bound to any port except 35021.

Depending on the used service S/IP uses broadcast or unicast UDP telegrams. In general unicasts are used to address single S/IP nodes and broadcasts are used to address multiple nodes.

The EasySlave provides the following S/IP services:

S/IP Service	Message type request	Message type response	Description
SupportedUDPServices	61	62	This service returns a list of all supported "UDP based S/IP" services (unicast and broadcast).
ReadOnlyData	71	72	The ReadOnlyData service requests values of IDNs.
Nameplate	89	90	The nameplate service requests the electronic nameplate of the node. The nameplate information is returned as a list of nameplate entries.
Browse	91	100	The Browse service is used to detect devices within the same subnet.
Identify	93	94	The Identify service is used for the identification of a device, by instructing the device to identify itself, e.g. blinking S3 LED orange.
SetIp	95	96	The SetIp service is used to set an IP configuration of an interface.
NameplateBroadcast	99		The NameplateBroadcast service is used to gather information about a device.

Furthermore the EasySlave supports the TFTP Firmware Management / Update service.

## 6.5 Firmware and parameter update

The firmware update is processed over TFTP. The update routines are implemented in the EasySlave core.

- An update is only possible after power on of the device and without active sercos communication.
- It is not allowed to disturb an update (e.g. because of power fail, ...). If this occurs the firmware will be destroyed
- after successful update the device must be switched off one time

The identification of the desired slave can be done via S/IP services. The identification service can be used to let blink the S3 LED on the slave and then the ReadOnlyData service can be used to evaluate the IP settings of the slave.

TFTP transfer via the command prompt:

Write: tftp -i **%IP\_ADDR%** put file.bin [**cod/par**]

Read: tftp -i **%IP\_ADDR%** get [**cod/par**] file.bin

**%IP\_ADDR%:** IP address of the slave device (e.g. 192.168.0.50)

**put:** write file to device

**get:** read file from device

**cod:** firmware file (code)

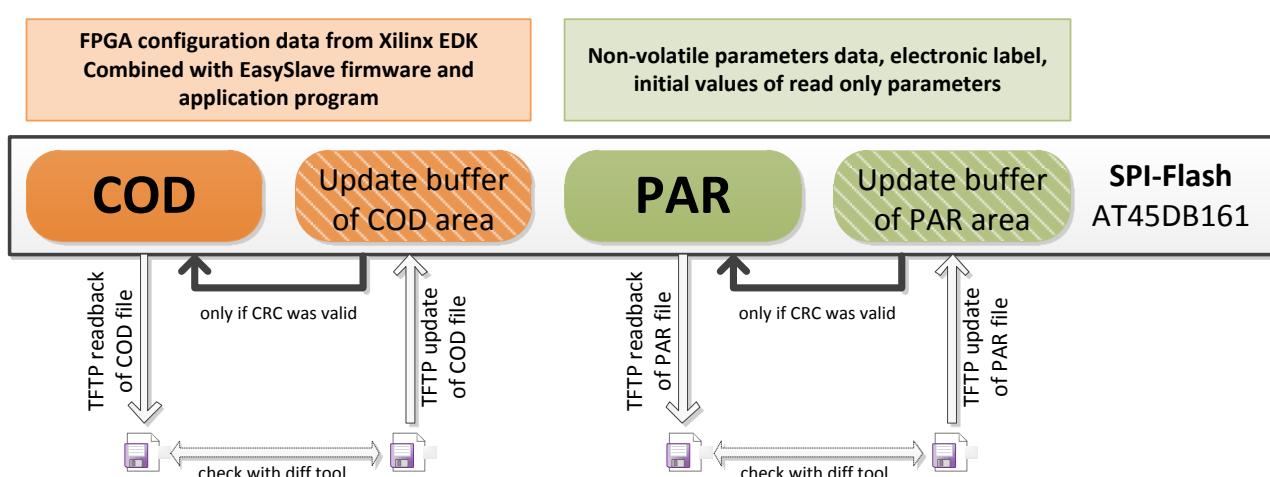
**par:** parameter file

Wait until the TFTP transfer completed.

Console output:

*Übertragung erfolgreich: 801598 Bytes in 27 Sekunde(n), 29688 Bytes/s*

The red sercos LED is blinking while the data are transferred from shadow flash memory to the flash memory. After the red LED is off the power supply of the slave can be disconnected. After the next power up the slave starts with the new firmware or parameter.



The firmware update can also be done via the JTAG interface of the FPGA. Therefore a Xilinx compatible JTAG programming adapter is required, e.g. Xilinx Platform Cable USB II. For programming the bitstream (\*.bit file extension) file is necessary because the binary COD file (\*.bin file extension) contains additional information for firmware update.

## 7 Notice

For the work with the EasySlave a basic understanding of industrial communication, especially sercos II is necessary. Further information can be found at the sercos international website and the sercos Wiki.

- <http://www.sercos.de/en>
- <https://wiki.sercos.org> (User account required)

## 8 References

- [S1] sercos specification General Overview and Architecture V1.3-1.3  
<https://wiki.sercos.org/rc/Documents>
- [S2] sercos specification Generic Device Profile V1.3-1.2  
<https://wiki.sercos.org/rc/Documents>
- [S3] sercos specification Communication V1.3-1.9  
<https://wiki.sercos.org/rc/Documents>
- [S4] sercos specification Communication Profile V1.3-1.3  
<https://wiki.sercos.org/rc/Documents>
- [S5] sercos specification Internet Protocol Services\_V1.3-1.2  
<https://wiki.sercos.org/rc/Documents>
- [S6] sercos specification Function specific Profile IO V1.3-1.5  
<https://wiki.sercos.org/rc/Documents>
- [S7] sercos specification Datasheet SERCON100 for Xilinx\_V1.3-1.9  
<https://wiki.sercos.org/rc/Documents>
- [S8] sercos specification Schematic for Xilinx Master/Slave Design\_V1.3-0.8  
<https://wiki.sercos.org/rc/Documents>
- [S9] Design Guide EasySlave  
<http://www.sercos.de/downloads>
- [S10] Reference schematic for EasySlave-IO  
<http://www.sercos.de/downloads>
- [X1] Spartan-6 FPGA Family Overview (ds160)  
XILINX
- [X2] Spartan-6 FPGA Datasheet – DC and Switching Characteristics (ds162)  
XILINX
- [X3] Spartan-6 FPGA – PCB Design and Pin Planning Guide (ug393)  
XILINX
- [T1] Datasheet – DP83848I  
Texas Instruments (former National Semiconductor)