

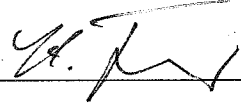




**STUP**  
**SpaceWire Protocol**

EADS Astrium GmbH, ASE4  
Doc No: SMCS-ASTD-PS-001  
Issue: 1.0  
Updated: 15-Apr-2005  
Page: 1 of 6

**STUP**  
**SpaceWire Protocol**  
**Protocol Specification**

<b>Prepared by:</b>	P. Rastetter – ASE421 <i>System Engineer</i>	 _____	<b>Date:</b> <u>15/4/05</u>
<b>Checked by:</b>	U. Liebstückel – ASE421 <i>System Engineer</i>	 _____	<b>Date:</b> <u>15.04.05</u>
<b>Released by:</b>	Dr. S. Fischer – ASE411 <i>Project Manager</i>	 _____	<b>Date:</b> <u>16.04.05</u>

**All Rights Reserved – Copyright per DIN 34:** Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design.



**STUP**  
**SpaceWire Protocol**

EADS Astrium GmbH, ASE4  
Doc No: SMCS-ASTD-PS-001  
Issue: 1.0  
Updated: 15-Apr-2005  
Page: 2 of 6

## Table of Contents

<b>1. INTRODUCTION.....</b>	<b>3</b>
1.1 LIST OF APPLICABLE DOCUMENTS .....	3
1.2 LIST OF REFERENCE DOCUMENTS .....	3
<b>2. PROTOCOL DESCRIPTION .....</b>	<b>4</b>
2.1 INTRODUCTION.....	4
2.2 PROTOCOL FORMAT.....	4
<b>3. IMPLEMENTATION OF THE STUP IN THE SMCS116SPW .....</b>	<b>5</b>
3.1 WRITE COMMAND FORMAT .....	5
3.2 READ COMMAND .....	5
3.3 READ REPLY FORMAT .....	6



# STUP

## SpaceWire Protocol

EADS Astrium GmbH, ASE4  
Doc No: SMCS-ASTD-PS-001  
Issue: 1.0  
Updated: 15-Apr-2005  
Page: 3 of 6

## 1. Introduction

This document describes in detail the STUP SpaceWire Protocol.

### 1.1 List of applicable documents

[AD1] SpaceWire Standard, ECSS-E-50-12A, 24. January 2003

### 1.2 List of reference documents

[RD1] SMCS116SpW User Manual

## 2. Protocol Description

### 2.1 Introduction

Using several different nodes in a SpaceWire network leads to the demand for the handling of distinct protocols. For some applications a complex protocol is needed covering all requirements. However, for some other applications a protocol which is easier to implement is required. The STUP (Serial Transfer Universal Protocol) focuses on these applications and therefore it has a minimized complexity.

The STUP will be implemented in the SMCS116SpW, the new SpaceWire compliant version of the SMCS116 [RD1]. This device requires an easy to implement protocol. The usage of this protocol by the SMCS116SpW is described in Section 3.

### 2.2 Protocol format

The following figure describes the STUP:

	Destination Path Address	Destination Path Address	Destination Path Address
Destination Logical Address	Protocol Identifier	Source Logical Address	Data (1 or more byte)
EOP			

**Figure 1: Protocol format**

The Destination Path Address bytes are optional.

The first byte gives the logical address of the destination (of the command).

The second byte contains the protocol ID of the STUP.

The third byte is the Source Logical Address.

After these three bytes a certain number of data bytes (one or more) follow.

Finally, the packet is terminated by an EOP (End of Packet).

This protocol structure can be used for all kinds of commands like 'read' or 'write' for example.

### 3. Implementation of the STUP in the SMCS116SpW

This section describes how the STUP is implemented in the SMCS116SpW.

#### 3.1 Write Command format

The following figure describes a Write command:

Destination Address	Logical	Protocol Identifier	Source Address	Logical	Command/ Reg.Addr
Data (1 or more byte *)		Checksum	Checksum		EOP

\*Note: SMCS116SpW ignores dummy bytes

**Figure 2: Write Command format**

Byte 4 defines a write command (D7 = 0).

D6-D0 contains the register address were the data is written to.

Two Checksum bytes are appended if checksum generation is enabled.

#### 3.2 Read Command

The following figure describes a Read command:

Destination Address	Logical	Protocol ID	Source Address	Logical	Command/Reg.Addr
Data (0 or more bytes)*		Checksum	Checksum		EOP

\*Note: SMCS116SpW ignores dummy bytes

**Figure 3: Read Command Format**

Byte 4 defines a read (D7 = 1) command.

D6-D0 contains the register address were data is read from

Two Checksum bytes are appended if checksum generation is enabled.

### 3.3 Read Reply Format

Read reply packet is sent in response to a read command.

		Source Path Address	Source Path Address	Source Path Address
Source Address	Logical	Protocol Identifier	Destination Address	Logical Reg. Addr
Data		Checksum	Checksum	EOP

**Figure 4: Read reply format**

The Source path Address bytes contain any required path address bytes to route the reply packet from the destination node back to the source node.

Up to 8 Source Path Address bytes can be pre-pended via header registers.

Two Checksum bytes are appended if checksum generation is enabled.