STUP
SpaceWire Protocol

Protocol Specification

Prepared by: P. Rastetter – ASE232
System Engineer

Date: 24/7/09

Checked by: U. Liebstückel – ASE232
System Engineer

Date: 24.07.09

Released by: Dr. S. Fischer – ASE231
Project Manager

Date: 24.07.09
Table of Contents

1. INTRODUCTION
   1.1 LIST OF APPLICABLE DOCUMENTS
   1.2 LIST OF REFERENCE DOCUMENTS

2. PROTOCOL DESCRIPTION
   2.1 INTRODUCTION
   2.2 PROTOCOL FORMAT

3. IMPLEMENTATION OF THE STUP IN THE SMCS116SPW
   3.1 WRITE COMMAND FORMAT
   3.2 READ COMMAND
   3.3 READ REPLY FORMAT
Document Change Record

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Author</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>April 05</td>
<td>P. Rastetter</td>
<td>Initial Issue</td>
</tr>
<tr>
<td>1.1</td>
<td>July 09</td>
<td>P. Rastetter</td>
<td>Removal of copyright</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Insertion of Document Change Record</td>
</tr>
</tbody>
</table>
1. **Introduction**

This document describes in detail the STUP SpaceWire Protocol.

1.1 **List of applicable documents**


1.2 **List of reference documents**

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:

<table>
<thead>
<tr>
<th>Destination Logical Address</th>
<th>Protocol Identifier</th>
<th>Source Logical Address</th>
<th>Data (1 or more byte)</th>
<th>EOP</th>
</tr>
</thead>
</table>

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:

<table>
<thead>
<tr>
<th>Destination Address</th>
<th>Logical Protocol Identifier</th>
<th>Source Address</th>
<th>Logical Command/ Reg.Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data (1 or more byte *)</td>
<td>Checksum</td>
<td>Checksum</td>
<td>EOP</td>
</tr>
</tbody>
</table>

*Note: SMCS116SpW ignores dummy bytes*

**Figure 2: Write Command format**

Byte 4 defines a write command (D7 = 0).
D6-D0 contains the register address where 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:

<table>
<thead>
<tr>
<th>Destination Address</th>
<th>Logical Protocol ID</th>
<th>Source Address</th>
<th>Logical Command/Reg.Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data (0 or more bytes)*</td>
<td>Checksum</td>
<td>Checksum</td>
<td>EOP</td>
</tr>
</tbody>
</table>

*Note: SMCS116SpW ignores dummy bytes*

**Figure 3: Read Command Format**

Byte 4 defines a read (D7 = 1) command.
D6-D0 contains the register address where 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.

<table>
<thead>
<tr>
<th>Source Path Address</th>
<th>Source Path Address</th>
<th>Source Path Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address</td>
<td>Logical Protocol Identifier</td>
<td>Destination Logical Address</td>
</tr>
<tr>
<td>Data</td>
<td>Checksum</td>
<td>Checksum</td>
</tr>
</tbody>
</table>

![Figure 4: Read reply format](image)

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.