Serial Communication Bus Controller

Version 1.03

Instruction Manuel

Written by dr. Tamás Zigó
ByteStudio Limited

www.bytestudio.hu
bytestudio@bytestudio.hu

July 20, 2015

Table of contents

Copyright

Disclaimer

ByteStudio provides this document "as is", without warranty of any kind, neither expressed nor implied, including, but not limited to, the particular purpose. ByteStudio may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

Using this Document

This document is intended for the software and hardware engineer's reference and provides detailed information about the Serial Communication Bus of the SCB Controller board. Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide. In that event, please contact ByteStudio (bytestudio@bytestudio.hu) for additional information that may help in the development process.

Author

Author: dr. Tamás Zigó (zigotamas@bytestudio.hu)

1. Serial Communication Bus structure

The SCB Controller based systems have a high data rate two-way serial communication bus (SCB), through which data can be transferred between the SCB Master device (e.g. Gigabit Ethernet Controller II.) and the SCB Slave device. Several slave devices can be connected to the SCB Controller via the SCB-S connectors.

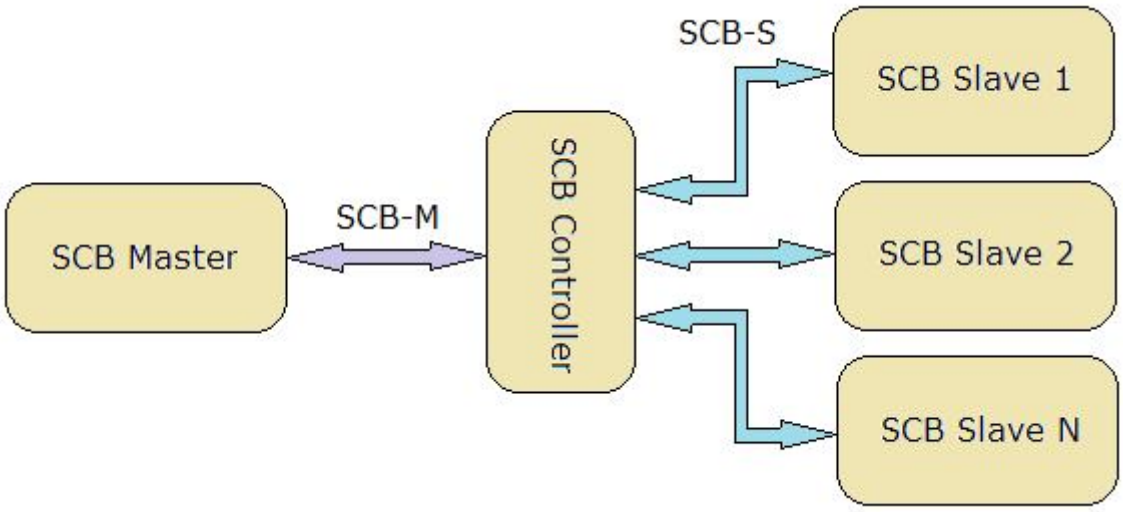


Figure 1. SCB structure

2. SCB Controller

The SCB Controller board has an SCB-M connector for the Master device, 10 SCB-S connectors for the Slave devices and one CW-Net SDI connector to connect standard CW-Net devices. The SCB Controller needs single 3.3 V power supply. The type of the SCB-S connections is point-to-point. Only one Slave device can be connected to one SCB-S connector.

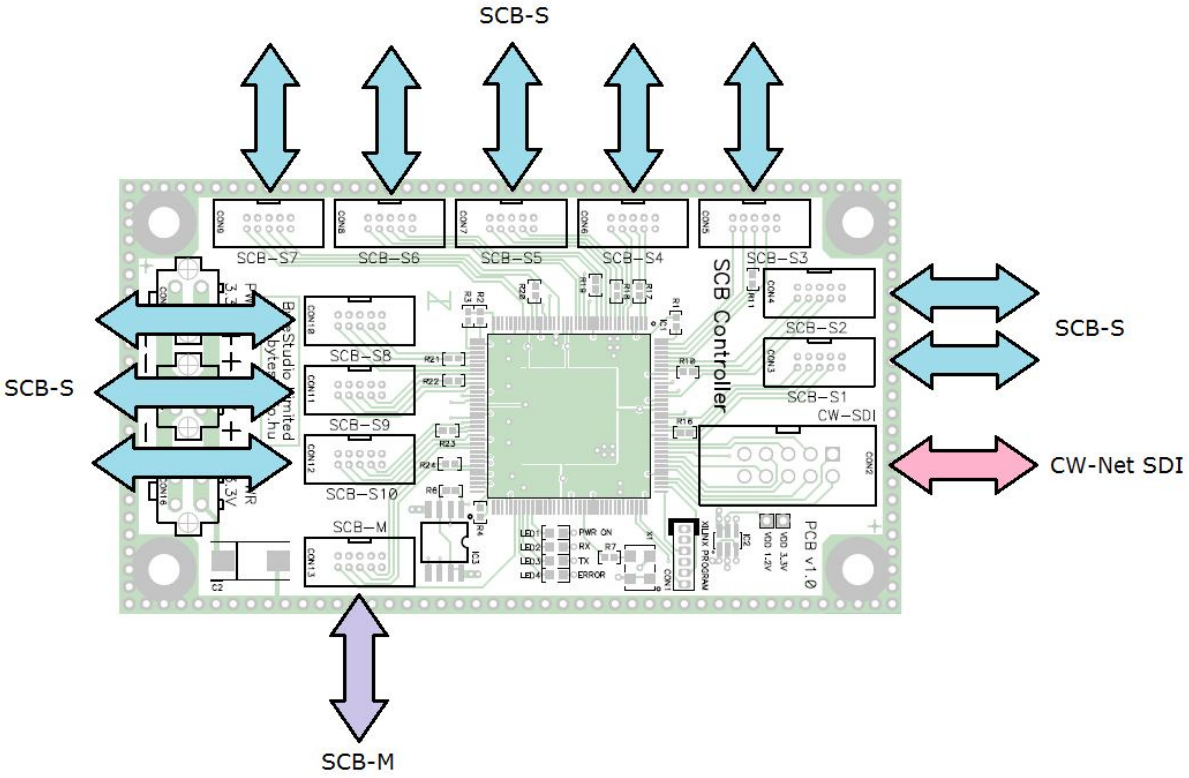
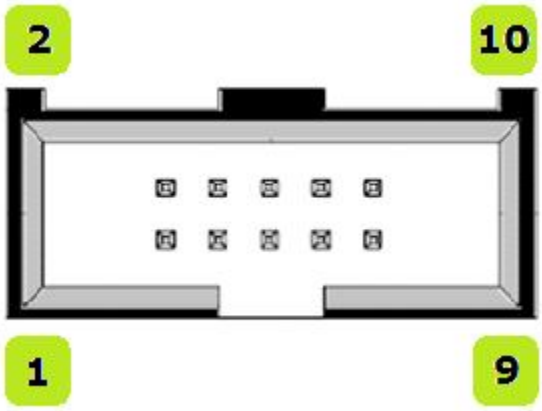


Figure 2. SCB Controller

3. SCB-S Connector

The SCB-S connector is a 10-pin shrouded header connector made by Samtec (www.samtec.com).



Connector type: Samtec SHF-105-01-L-D-TH

Cable strip: Samtec FFSD-05-01-N

SCB-S pinout of the SCB Controller

Pin	Description	Direction	IO Standard
1	GND	-	-
2	CLK	Input	3.3 V LVCMOS
3	GND	-	-
4	IRQ	Output	3.3 V LVCMOS
5	GND	-	-
6	**DO**	**Output**	3.3 V LVCMOS
7	GND	-	-
8	**DI**	**Input**	3.3 V LVCMOS
9	GND	-	-
10	RST	Output	3.3 V LVCMOS

SCB-S pinout of the Slave device

Pin	Description	Direction	IO Standard
1	GND	-	-
2	CLK	Output	3.3 V LVCMOS
3	GND	-	-
4	IRQ	Input	3.3 V LVCMOS
5	GND	-	-
6	**DI**	**Input**	3.3 V LVCMOS
7	GND	-	-
8	**DO**	**Output**	3.3 V LVCMOS
9	GND	-	-
10	RST	Input	3.3 V LVCMOS

4. SCB-S Connector requirements on the Slave device

The following external resistors are required on the PCB of the SCB Slave Device for proper operation.

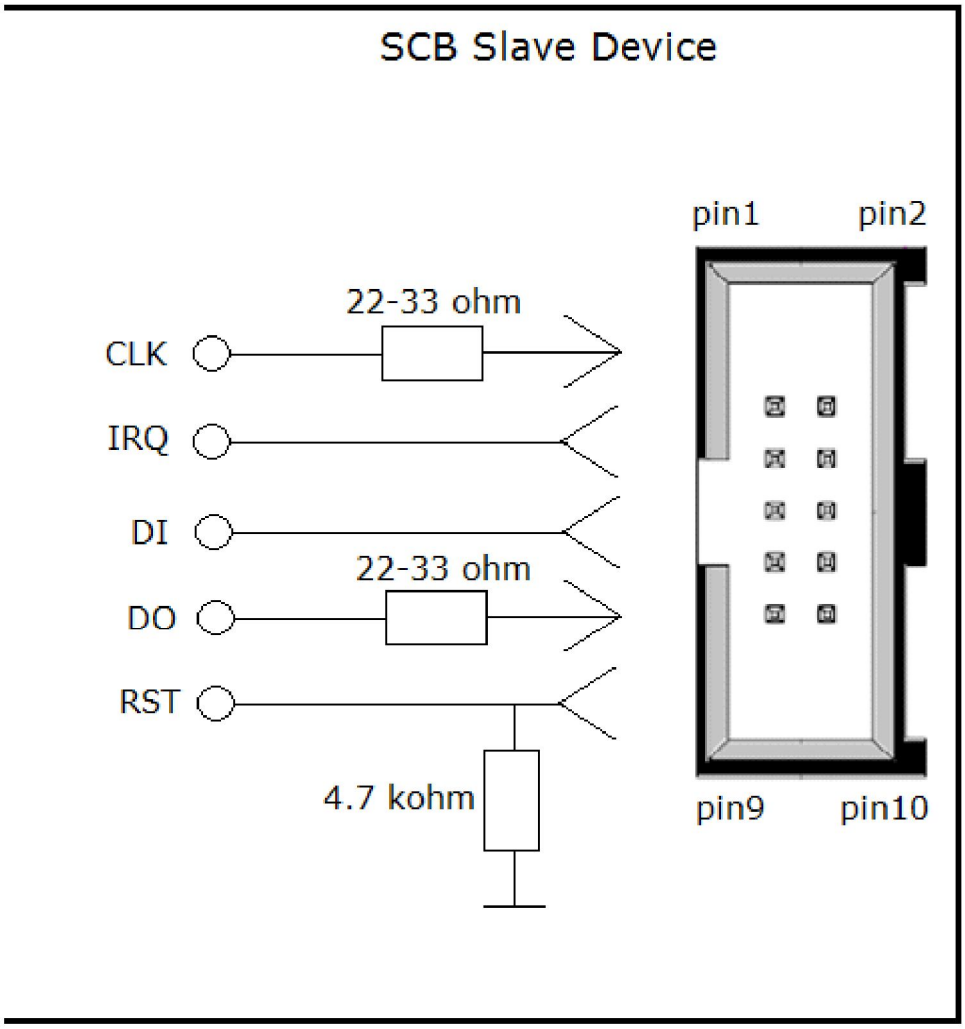


Figure 3. Required SCB resistors

An optional weak pull-down resistor (≥ 4.7 kΩ) can be placed on the IRQ and DI signals.

5. Signal description (Slave side)

The SCB has 10 wires. It has five unidirectional 3.3 V LVCMOS signals. The additional five wires in the cable are ground wires, interleaved with the signal wires to reduce the effects of capacitive coupling between neighboring signal wires, reducing crosstalk. CLK and DO signals are outputs generated by the Slave device. IRQ, DI and RST are inputs.

- Clock signal (CLK) is generated by the Slave device. CLK frequency must be between 0 and 15 MHz. Clock can be either a continuous clock or can be a asserted only during data transfer.
- Interrupt signal (IRQ). A low to high transition on the IRQ signal generates an interrupt in the Slave device. In idle state IRQ is at low.
- Serial data input signal (DI) is generated by the SCB Controller. The Slave device samples the DI line on the falling edge of the CLK.
- Serial data output signal (DO) is generated by the Slave device. The data on DO is sampled by the SCB Controller (it must be stable) on the rising edge of the CLK.
- RST is an active-low reset signal generated by the SCB Controller.

The communication on the serial bus is realized by writing or reading 8 bits long bytes. Every byte transfer need 9 clock cycles.

6. Byte write

Writing one byte onto the serial bus takes nine clock cycles. DO is sampled by the SCB Controller on the rising edge of the clock. DO must be high during the the first clock cycle (start bit) indicating the byte transfer. The start bit is followed by eight data bits. Data is transferred with the most significant bit (MSB) first.

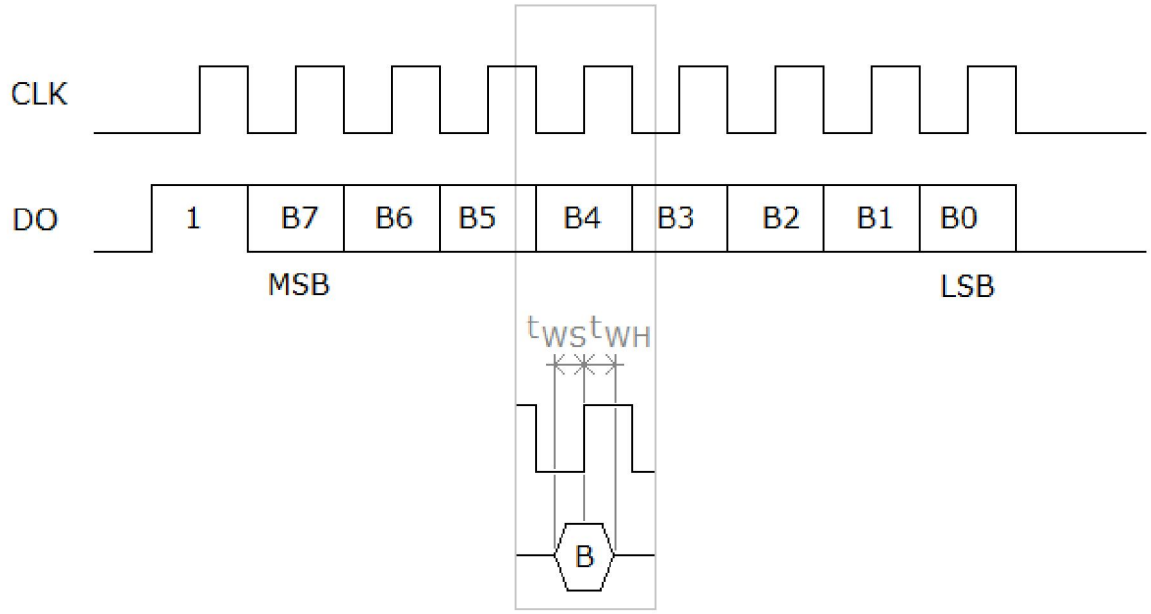


Figure 4. Byte write

Symbol	Item	Min.	Typ.	Max.
t_{WS}	Data valid before rising edge (setup time)	15 ns	-	-
t_{WH}	Data valid after rising edge (hold time)	15 ns	-	-

After the last byte of the message is transferred from the Slave device to the SCB Controller the communication must be finished by a STOP condition. The STOP condition is generated by sending one h00 byte with cleared start bit, so DO must be in low for nine clock cycles.

7. Byte read

Reading one byte from the serial bus takes nine clock cycles. DO is sampled by the SCB Controller on the rising edge of the clock. DI must be sampled by the Slave device on the falling edge of the clock. DO must be high during the the first clock cycle (start bit) indicating the byte transfer. The start bit is followed by eight data bits (MSB first).

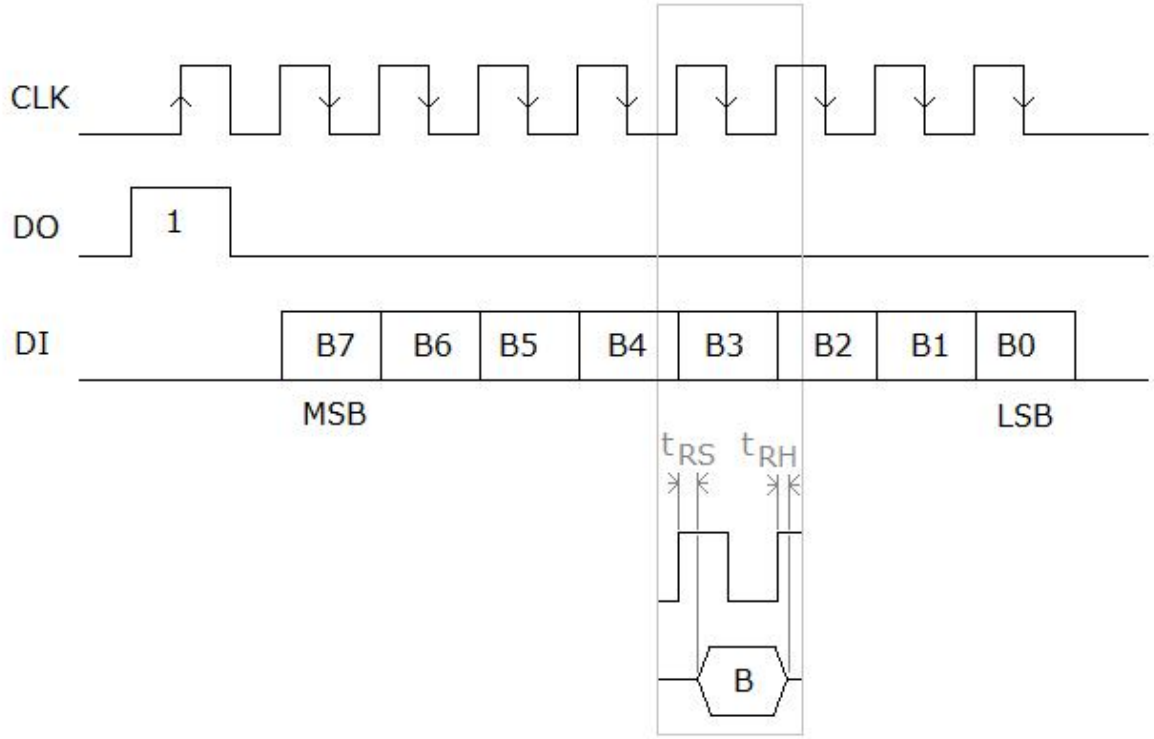


Figure 5. Byte read

Symbol	Item	Min.	Typ.	Max.
t_{RS}	Data valid after the rising edge of the clock	-	-	10 ns
t_{RH}	Data valid after the next rising edge	0 ns	-	-

8. Communication

Via the SCB the Slave device can access the status register, the communication memory area and the register memory area.

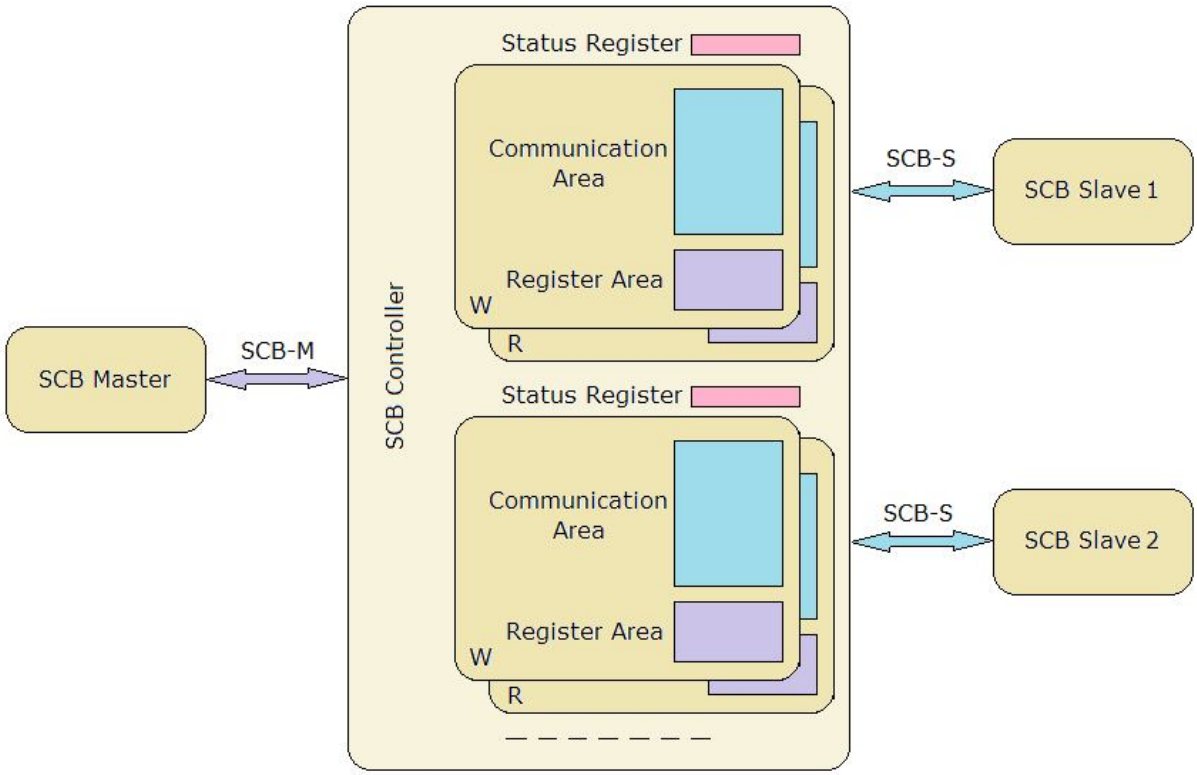


Figure 6. Memory areas

The Slave device can read from or write to these areas. The communication (read or write messages) is always initiated by the Slave device.

The Communication area consists of two 1536 bytes long memory pages, one for sending data from the Master device to the Slave device (page R) and one for sending data from the Slave device to the Master device (page W). The Slave device can read page R and write page W. The Master device can read page W and write page R. If the Master device wants to communicate with the Slave device it has to write the data (e.g. DDToIP instruction chain) into the communication area. After the Master device writes the last byte of the data the SCB Controller generates an interrupt to the Slave device and also set the corresponding flag (SCA) in the status register. After the Slave device reads the data the flag is cleared. The Slave device can also send data to the Master device by writing it into the communication area. After the Slave device writes the

last byte of the data the SCB Controller generates an interrupt to the Master device and set the corresponding flag (MCA) in the status register. After the Master device reads the data the flag is cleared.

The register area consists of two 512 bytes long memory pages. Page R is read-only and page W is write-only for the Slave device. Typically the read-only page is for storing settings, the write-only is for storing variables. The Master device can overwrite the read-only registers. If the Master device modifies one of these registers the corresponding flag (SRA) in the status register is set. If the Slave device read the registers the flag is cleared. The Slave device can overwrite the write-only registers and the Master device can read them. If the Slave device modifies one of these registers the corresponding flag (MRA) in the status register is set. If the Master device read the registers the flag is cleared.

The communication on the serial bus is realized by sending and receiving messages. Messages consist of bytes. Every message is started with an one byte long opcode and finished by a STOP condition.

Every byte transferred on the bus is 8 bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be preceded by a start bit. The next sections use the following symbols to represent messages:

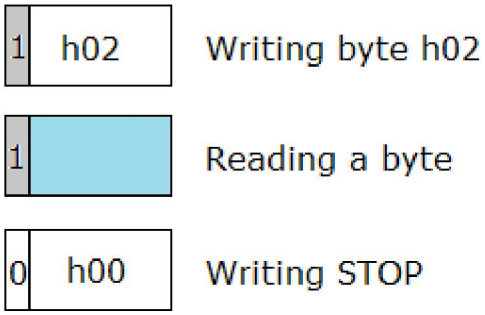


Figure 7. Symbols for byte write and read

9. Status register

The Slave device can access the 3 bytes long status register using the SRW (Status Register Write) and SRR (Status Register Read) messages. The status register has the following fields:

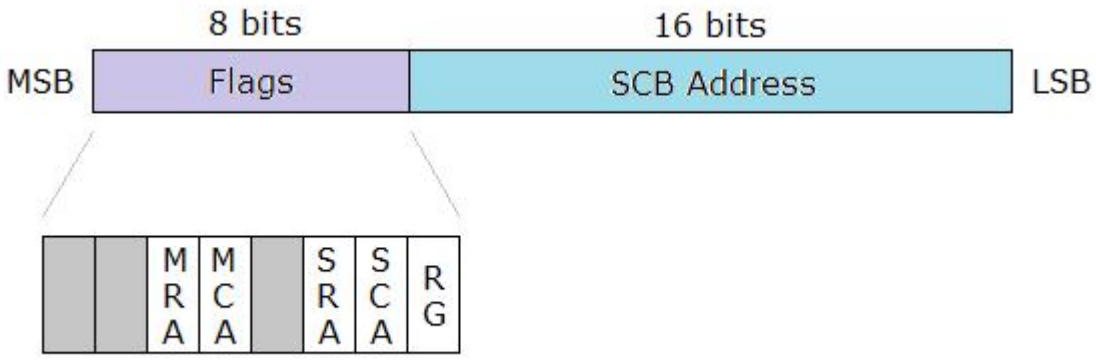


Figure 8. Status register

The flags are set and cleared automatically by the SCB Controller. The SCB Address is set by the Slave device.

Flag	Event
RG	SCB Address is registered
SCA	The SCB Master wrote a new message into the communication area (R)
SRA	Register area (R) is modified by the SCB Master
MCA	The SCB Slave wrote a new message into the communication area (W)
MRA	Register area (W) is modified by the SCB Slave

After power on the value of the status register is h000000. The Slave device first has to write its own SCB Address to the status register (SCB Address Registration). After registering the SCB Address the RG flag (bit 16) is set. (Note: while the RG flag is cleared the SCB Controller processes only SRW and SRR messages.)

The SCB Address is 16-bit long. The first 12 bits identify the type of the slave board, the last 4 bits are used the address the same boards in the system.

The opcode of the SRW message is h02, the length is 4 bytes.

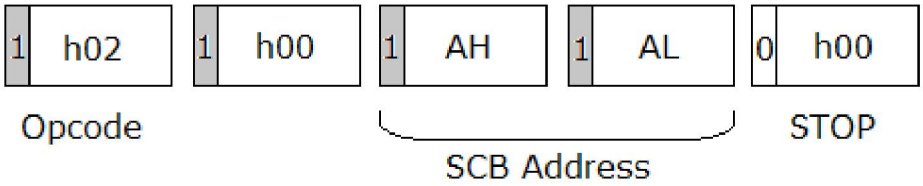


Figure 9. SRW message

The Slave device can read the status register using the SRR message. The opcode of the SRR message is h01, the length is 4 bytes.

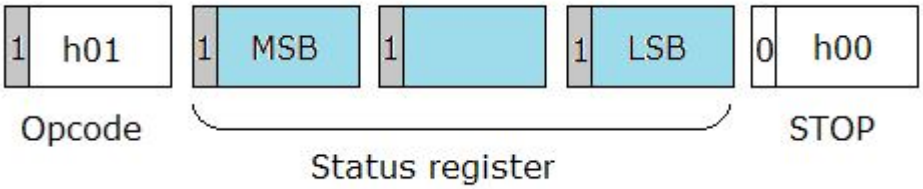


Figure 10. SRR message

10. Writing and reading the communication area

The Slave device can write data to the communication area using the CAW message. The number of data bytes (N) can be between 1 and 1536. (Data bytes are stored in the memory from address h0000.) The opcode of the CAW message is h03.

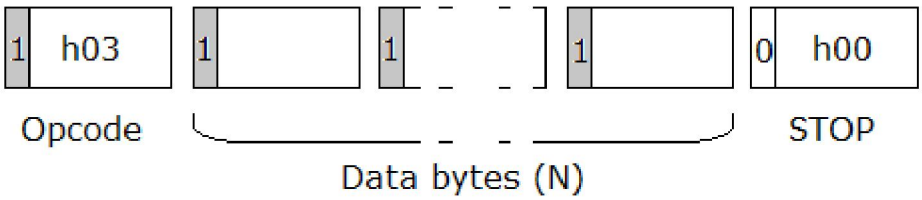


Figure 11. CAW message

The Slave device can read data from the communication area using the CAR message. The number of data bytes (N) can be between 1 and 1536. (Data bytes are read from address h0000.) The opcode of the CAR message is h04.

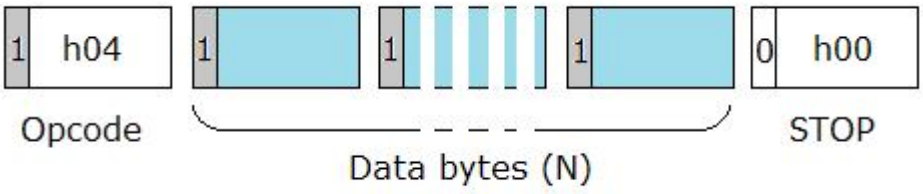


Figure 12. CAR message

11. Writing and reading the register area

The Slave device can write data to the register area using the RAW message. The number of data bytes (N) can be between 1 and 512-RA. Data bytes are stored in the memory from address RA. The opcode of the RAW message is h05.

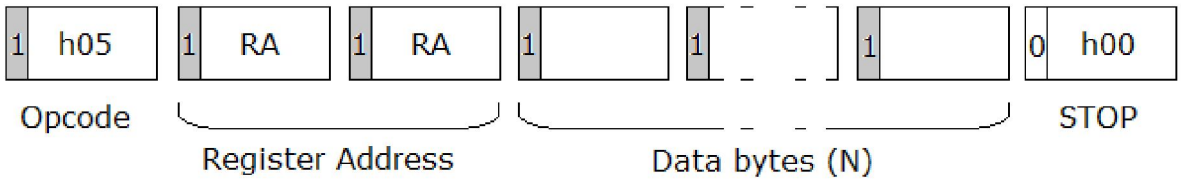


Figure 13. RAW message

The Slave device can read data from the register area using the RAR message. The number of data bytes (N) can be between 1 and 512-RA. Data bytes are read from address RA. The opcode of the RAR message is h06.

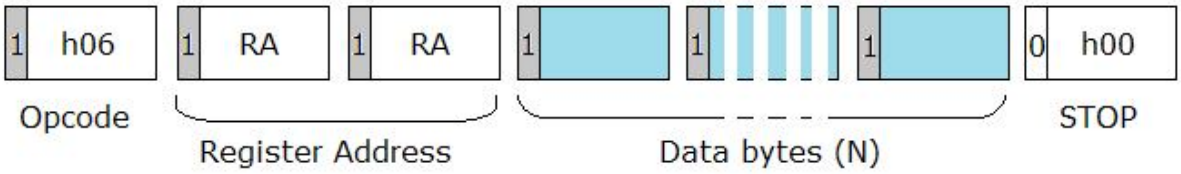


Figure 14. RAR message

12. Reading the version of the SCB Controller

The Slave device can read the version of the SCB Controller using the RVE message. The opcode of the RVE message is h07, the length is 3 bytes. Version format: VH.VL (e.g.: 1.01)

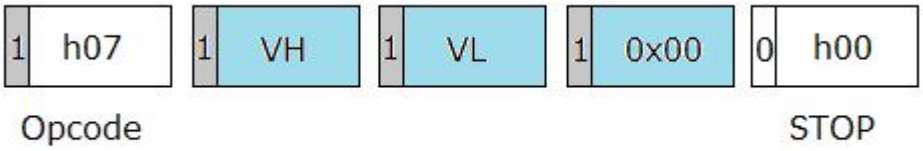


Figure 15. RVE message