

### CONTENTS

|  |          |  |          |
|--|----------|--|----------|
| <b>FIRST BASIC COMMUNICATION SEQUENCE</b>                                | <b>1</b> | Status Information2 . . . . .                        | <b>4</b> |
| Test the communication interface . . . . .                               | 1        |  |          |
| Reset and stopping prior started <i>BiSS</i> master activities . . . . . | 1        |  |          |
| <b>BASIC CONFIGURATION SEQUENCE</b>                                      | <b>2</b> |  |          |
| Reset Sensor and Actuator data . . . . .                                 | 2        |  |          |
| Register Data . . . . .  | 2        |  |          |
| Configuration of all Slaves . . . . .                                    | 2        |  |          |
| Control Communication . . . . .  | 2        |  |          |
| Master Configuration . . . . .   | 2        |  |          |
| Channel Configuration . . . . .  | 2        |  |          |
| Instruction Register reset . . . . .                                     | 2        |  |          |
| Status Information 1 . . . . .   | 2        |  |          |
| <b>SOFTWARE BASED SCD TRIGGER</b>  | <b>3</b> |  |          |
| CDM = 0 . . . . .  | 3        |  |          |
| CDM = 1 . . . . .  | 3        |  |          |
| Configured Register communication with full SCD . . . . .                | 3        |  |          |
| Configured Register communication with reduced protocol . . . . .        | 3        |  |          |
| <b>SOFTWARE BASED STATUS POLLING</b>                                     | <b>4</b> |  |          |
| Status Information . . . . .   | 4        |  |          |
|  |          | <b>HARDWARE BASED SCD TRIGGER</b>                    | <b>5</b> |
|  |          | GETSENS . . . . .                                    | 5        |
|  |          | GETSENS and register access synchronisation          | 5        |
|  |          | <b>HARDWARE BASED END OF TRANSMISSION INDICATION</b> | <b>5</b> |
|  |          | EOT Status Information . . . . .                     | 5        |
|  |          | nERR Status Information . . . . .                    | 5        |
|  |          | <b>ERROR STATE DETECTION</b>                         | <b>6</b> |
|  |          | nERR Status Information . . . . .                    | 6        |
|  |          | Status Information 1 . . . . .                       | 6        |
|  |          | Status Information 2 . . . . .                       | 6        |
|  |          | <b>ERROR STATE CORRECTION</b>                        | <b>7</b> |
|  |          | BREAK . . . . .                                      | 7        |
|  |          | INIT . . . . .                                       | 7        |
|  |          | <b>SLAVE CONFIGURATION CONFLICT</b>                  | <b>7</b> |
|  |          | <b>REGISTER ACCESS WITH SSI</b>                      | <b>8</b> |
|  |          | <b>REVISION HISTORY</b>                              | <b>8</b> |

### FIRST BASIC COMMUNICATION SEQUENCE

This chapter recommends first communication details for the host to communicate with the *BiSS* master.

#### Test the communication interface

As a basic test can be combined with the identification of the *BiSS* master. This ensures that the host firmware matched to the found hardware. The related addresses are 0xEA and 0xEB for the device REVISION(7:0) and VERSION(7:0).

#### Reset and stopping prior started *BiSS* master activities

iC-MB4 does reset all RAM registers to 0x00 on a power on. A prior configured and non power cycled *BiSS* master needs to be resetted completely to prevent prior configuration and operational states. It is recommended to execute a full reset of the device and a stop of running sequences by the BREAK instruction.

### BASIC CONFIGURATION SEQUENCE

This chapter explains how to reset and initialize the *BiSS* master.

#### Reset Sensor and Actuator data

The SCDATA RAM range is readable and writable: a reset of the whole SCDATA range 0x00 ... 0x3F is recommended. Resetting the full memory range with an unexpected content like 0x00 supports identifying RAM updates in the evaluation phase.

#### Register Data

The RCDATA RAM range is readable and writable: a reset of the whole SCDATA range 0x80 ... 0xBF is recommended. Resetting the full memory range with an unexpected content like 0x00 supports identifying RAM updates in the evaluation phase.

#### Configuration of all Slaves

iC-MB4 supports up to 8 slaves that are individually configurable. Each slave the configuration range is 4 bytes wide. The slave configuration range is readable and writable: a reset of the whole slaves configuration is recommended. A reset of the complete configuration range of all slaves is 0xC0 ... 0xDF is recommended.

#### Control Communication

The control communication configuration range is readable and writable: a reset or default setting of the whole control communication configuration is recommended. The complete reset or setting control communication configuration range of is 0xE0 ... 0xE5 is recommended.

#### Master Configuration

The master configuration range is readable and writable: a reset or default setting of the whole master configuration is recommended. The complete reset or setting master configuration range of is 0xE6 ... 0xEB is recommended.

#### Channel Configuration

The channel configuration range is readable and writable: a reset or default setting of the whole channel configuration is recommended. The complete reset or setting channel configuration range of is 0xEC ... 0xEE is recommended.

#### Instruction Register reset

The instruction 0xF4 register is partial readable and fully writable: a reset or default setting of the instruction register is recommended. The reset or setting of the instruction register 0xF4 is recommended.

To stop running sequences the instruction BREAK is mandatory and a set BREAK INSTR(2:0) will be reset after execution.

The instruction 0xF5 register is fully readable and fully writable: setting this instruction register to the required state is recommended. The setting of the instruction register 0xF5 is recommended.

#### Status Information 1

The status information 1 range is partial readable and writable: a reset or default setting of the whole status information 1 configuration is recommended. The status information 1 range is 0xF0 ... 0xF3. The possible reset of the status information 1 range of is 0xF1 ... 0xF2 is recommended.

## SOFTWARE BASED SCD TRIGGER

This chapter explains how to trigger SCD transfer by software.

### **CDM = 0**

A single SCD cycle is started with the instruction register 0xF4 and the INSTR(2:0) = 0b010. This SCD with a CDM = 0 will be added to the internal CDM counter for counting the leading CDM = 0 bits.

### **CDM = 1**

A single SCD cycle is started with the instruction register 0xF4 and the INSTR(2:0) = 0b001. This SCD with a CDM = 1 will reset the internal CDM counter for counting the leading CDM = 0 bits.

### **Configured Register communication with full SCD**

A configured control communication e.g. a register access can be started by the the instruction register 0xF4 and the INSTR(2:0) = 0b100 or 0b110.

A software based start of each SCD can be started by repeating the instruction register 0xF4 and the INSTR(2:0) = 0b100 or 0b110 each SCD cycle. This software based start of each SCD requires that sufficiently enough SCD cycles are started to execute the complete register access.

Alternatively the INSTR(2:0) = 0b100 or 0b110 can be used with a set AGS or external GETSENS signals to start all required SCD cycles.

Full SCD cycles are started. The started SCD cycles with a CDM = 1 will reset the internal CDM counter for counting the leading CDM = 0 bits.

### **Configured Register communication with reduced protocol**

A configured control communication e.g. a register access can be started by the the instruction register 0xF4 and the INSTR(2:0) = 0b111. This INSTR does require a set AGS or external GETSENS signals to start all required SCD cycles. Reduced SCD cycles are started. The started SCD cycles with a CDM = 1 will reset the internal CDM counter for counting the leading CDM = 0 bits.

## SOFTWARE BASED STATUS POLLING

This chapter explains how to monitor the *BiSS* master status by polling.

### Status Information

The EOT status bit indicates an end of the SCD cycle, not including the *BiSS* slaves timeout.

The REGEND status bit indicates an end of the register communication. The nREGERR bit indicates that the an error occurred in the register communication.

The nERR bit indicates that the an error occurred or not.

The nAGS bit indicates that the an AGS related error occurred or not. Cause for this error can be a:

- A not finished SCD due to not responding slave
- A not finished SCD due to not leaving ACK state slave
- A collision of running SCD and next to be started SCD

The nSCDERR bit indicates that the SCD data is valid or not.

The SVALIDx bit indicates that the SCD data of slave x is valid or not. On *BiSS* bus structures the defective slave SCD data is identifiable.

### Status Information2

The CDSx bit indicates the last received CDS bit on channel x.

The SLx bit indicates the actual SL line state on channel x.

### HARDWARE BASED SCD TRIGGER

This chapter explains how to trigger SCD transfer by hardware.

#### **GETSENS**

The GETSENS pin is a static input pin for triggering the start of the SCD.

The minimum GETSENS = 1 signal duration needs to be bigger than one period of the internal oscillator period.

The maximum GETSENS = 1 signal duration is recommended to be smaller than the shortest period of the SCD cycle to prevent re-triggering.

#### **GETSENS and register access synchronisation**

Combining software (register access control) and hardware (GETSENS pin to synchronize the SCD cycle start) is also possible. The instruction register 0xF4 and the INSTR(2:0) use AGS and the external GETSENS signal with the following combination:

- `FREQAGS = "AGSINFINITE"` (AGS will wait for the GETSENS signal)
- set AGS bit = 1, generating SCD cycles automatically
- external GETSENS signals that generate SCD cycles automatically
- `INSTR = 0b100` or `0b110` will start the register access with the next GETSENS pulse.

All SCD cycles are started by the GETSENS signal. The started SCD cycles with a `CDM = 1` will reset the internal CDM counter for counting the leading `CDM = 0` bits.

### HARDWARE BASED END OF TRANSMISSION INDICATION

This chapter explains how to monitor the *BiSS* master status by polling.

#### **EOT Status Information**

The EOT output pin indicates the state of the EOT status register bit.

#### **nERR Status Information**

The nERR output pin indicates the state of the nERR status register bit.

### ERROR STATE DETECTION

This chapter explains how to detect error states of the *BiSS* master and the *BiSS* communication.

#### nERR Status Information

The status bit nERR does combine the internal device error states and the external bidirectional pin NER state. If an additional external low active error signal is wired "OR" to the external bidirectional pin NER state, then the error source needs to be identified (internal device error or external provided error to pin NER) .

The internal error state nERR combines:

- nAGSERR
- nDELAYERR
- nSCDERR
- nREGERR
- NER pin input state

If there are no internal device error states then the error state is caused by the external low active error signal that provided at the external bidirectional pin NER.

#### Status Information 1

The status bit nSCDERR combines all enabled slave SVALIDx bits.

To identify all the slaves with failing SCDATA CRC verification, all SVALIDx bits need to be verified.

A mismatching slave configuration may cause a sequence of failing SCDATA verification.

#### Status Information 2

The SLx line state may help identifying SL lines with a static 0 signal state.

The *BiSS* SLx line default state is 1.

The SWBANKFAILS bit indicates that the bank switching was successful or not. If a bank switching was not successful, the *BiSS* master was switching SCDATA banks when the host tried to switch banks with the SWBANK bit in the instruction register 0xF4 bit 5. The SINGLEBANK bit in address 0xE7 bit needs to be 0 to permit bank switching. The recommended host sequence for switching banks:

1. SWBANK = 1
2. If ( SWBANKFAILS ) then
3. {
4.     delay
5.     goto 1
6. }
7. else
8. {
9.     // Bank switched successfully
10. }

### ERROR STATE CORRECTION

This chapter explains how to correct error states of the *BiSS* master and the *BiSS* communication.

#### **BREAK**

To stop running sequences the instruction **BREAK** is mandatory. Set **BREAK INSTR(2:0)** will be stop all running sequences.

The **BREAK INSTR(2:0)** bit will be reset after execution.

#### **INIT**

The **INIT INSTR(2:0)** will generate a clock train on the MA outputs of all channels. The connected *BiSS* slaves will respond with the **IDLOCKING** bits on SLx inputs. The *BiSS* master will measure also the channel specific line delay and store the measured channel 1 line delay in the **SCDATA1(7:0)** and the channel 2 line delay in the **SCDATA5(7:0)**. The unit of this value is the 1/4 of the configured MA clock frequency.

### SLAVE CONFIGURATION CONFLICT

This chapter explains how to enable register communication without knowing a correct *BiSS* slave configuration.

The connected slaves configuration needs to match with the *BiSS* master slave configuration. If the configuration is not matching:

- SCD verification may fail
- SCDATA may be incorrect
- SCD cycle may fail

To enable a SCD cycle without CRC verification all slaves CRC verification need to be deactivated. To deactivate the CRC verification each slaves SCD length needs be extended minimum by the CRC length and the slaves CRC polynomial needs to be disabled by choosing CRC length of 0.

A popular deactivation of SCD verification with iC-MB4 is:

- **SCDLENx = 63**
- **SELCRCSx = 0b0**
- **SCRCLENx = 0x00**
- Required for all enabled and present slaves

With iC-MB4 a CRC polynomial of 0x00 not applicable with **SELCRCSx = 0b1**.

### REGISTER ACCESS WITH SSI

This chapter explains how to enable register write communication with SSI protocol configuration.

With the SSI Protocol a timeout indication may be not available.

On SSI based systems the CDS response is not available.

To write a register the CDM bit needs to be stable for a sufficiently long time. HOLDCDM = 1 does keep the MA clock line constant with the inverted CDM bit state until the next SCD cycle starts. iC-MB4 provides the HOLDCDM option for BiSS and SSI.

HOLDCDM = 1 is neutral to standard BiSS communication.

### REVISION HISTORY

| Rel. | Rel. Date  | Chapter | Modification    | Page |
|------|------------|---------|-----------------|------|
| A1   | 2015-05-05 |         | Initial release |      |

| Rel. | Rel. Date  | Chapter                       | Modification   | Page |
|------|------------|-------------------------------|--|------|
| A2   | 2015-06-03 | SOFTWARE BASED SCD TRIGGER    | Software based SCD retriggering of SCD cycles for register access added. | 3    |
|      |            | SOFTWARE BASED STATUS POLLING | Status bits REGEND and nREGERR added.                                    | 4    |
|      |            | HARDWARE BASED SCD TRIGGER    | GETSENS and register access synchronisation added.                       | 5    |

iC-Haus expressly reserves the right to change its products and/or specifications. An info letter gives details as to any amendments and additions made to the relevant current specifications on our internet website [www.ichaus.com/infoletter](http://www.ichaus.com/infoletter); this letter is generated automatically and shall be sent to registered users by email.

Copying – even as an excerpt – is only permitted with iC-Haus' approval in writing and precise reference to source.

iC-Haus does not warrant the accuracy, completeness or timeliness of the specification and does not assume liability for any errors or omissions in these materials.

The data specified is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given. In particular, this also applies to the stated possible applications or areas of applications of the product.

iC-Haus products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (*Safety-Critical Applications*) without iC-Haus' specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems. iC-Haus products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by iC-Haus.

iC-Haus conveys no patent, copyright, mask work right or other trade mark right to this product. iC-Haus assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.