

# Design Guide **Anybus®-IC**

Doc.Id. HMSI-168-7  
Rev. 2.01

---

# Important User Information

This document is intended to provide a good understanding of the software interface as well as the mechanical and electric properties of the Anybus-IC platform. It does not cover any of the network specific features offered by the various incarnations of the product; this information is instead available as separate documents (Fieldbus Appendix).

The reader of this document is expected to be familiar with hardware design and communication systems in general.

## Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many application of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the application meets all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

## Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

## Trademark Acknowledgements

Anybus® is a registered trademark of HMS Industrial Networks AB. All other trademarks are the property of their respective holders.

|   |
|---|
| <p><b>Warning:</b> This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.</p> <p><b>ESD Note:</b> This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.</p> |
|---|

# Table of Contents

|                  |  |    |
|------------------|--|----|
| <b>Preface</b>   | <b>About This Document</b>             |    |
|                  | Related Documents .....                | 8  |
|                  | Document History .....                 | 9  |
|                  | Document Conventions .....             | 10 |
|                  | Sales and Support .....                | 11 |
| <br>             |  |    |
| <b>Chapter 1</b> | <b>About the Anybus-IC</b>             |    |
|                  | General .....                          | 12 |
|                  | Features .....                         | 12 |
|                  | Overview .....                         | 13 |
|                  | Application Connector .....            | 14 |
| <br>             |  |    |
| <b>Chapter 2</b> | <b>Tutorial</b>                        |    |
|                  | Introduction .....                     | 15 |
|                  | Step by Step Guide.....                | 16 |
|                  | Automatic Baud Rate Detection .....    | 17 |
|                  | Set Initialization Parameters .....    | 17 |
|                  | Set Indata Parameters .....            | 18 |
|                  | Set Outdata Parameters .....           | 19 |
|                  | Set Operation Mode.....                | 20 |
| <br>             |  |    |
| <b>Chapter 3</b> | <b>MIF Interface</b>                   |    |
|                  | General Information .....              | 21 |
|                  | Communication Settings .....           | 22 |
|                  | <i>Baud rate</i> .....                 | 22 |
|                  | <i>Communication Properties</i> .....  | 22 |
|                  | User Interface.....                    | 23 |
|                  | <i>General Information</i> .....       | 23 |
|                  | <i>Main Menu</i> .....                 | 23 |
|                  | <i>'Module Information'-menu</i> ..... | 23 |
|                  | <i>'Parameters'-menu</i> .....         | 24 |
|                  | <i>'Monitor'-menu</i> .....            | 24 |
|                  | <i>Firmware Upgrade'-menu</i> .....    | 24 |

## Chapter 4 SCI Channel

|   |    |
|---|----|
| General Information .....               | 25 |
| Modbus Implementation.....              | 26 |
| <i>Functions Codes</i> .....            | 26 |
| <i>Register Map</i> .....               | 26 |
| <i>Exception Codes</i> .....            | 26 |
| Communication Settings .....            | 27 |
| <i>Baud Rate</i> .....                  | 27 |
| <i>Timeout</i> .....                    | 27 |
| <i>Communication Properties</i> .....   | 27 |
| <i>Modbus RTU Address</i> .....         | 28 |
| Interfacing to RS-232.....              | 29 |
| Interfacing to RS-422.....              | 30 |
| Interfacing to RS-485 (Multidrop) ..... | 30 |

## Chapter 5 SSC Channel

|   |    |
|---|----|
| General Information .....                                 | 31 |
| The Shift Register Loop .....                             | 32 |
| Basic Shift Register Circuit.....                         | 33 |
| Fieldbus Specific Output Register                         |    |
| (Output Register 0).....                                  | 34 |
| <i>General</i> .....                                      | 34 |
| <i>LEDs (no External Drivers)</i> .....                   | 34 |
| <i>LEDs (External Driver Circuit)</i> .....               | 35 |
| Fieldbus Specific Input Register (Input Register 0) ..... | 36 |
| <i>General</i> .....                                      | 36 |
| <i>Binary Switches</i> .....                              | 36 |
| <i>BCD-coded Switches</i> .....                           | 36 |
| Sequence Diagrams .....                                   | 37 |
| <i>Initialization Sequence (Startup Only)</i> .....       | 37 |
| <i>Refresh Sequence (Normal Operation)</i> .....          | 38 |

## Chapter 6 I/O Mapping

|                           |    |
|---------------------------|----|
| General Information ..... | 39 |
| Fieldbus Mapping.....     | 40 |
| SSC Mapping.....          | 41 |
| SCI Mapping .....         | 42 |

## Chapter 7 Initialization

|  |    |
|--|----|
| General Information .....  | 43 |
| Normal Initialization .....  | 43 |
| Automatic Initialization (Standalone).....                           | 44 |
| Fieldbus Specific Initialization.....                                | 44 |
| Initialization Examples, Normal Initialization .....                 | 45 |
| <i>Switches and LEDs on SSC, Data Exchange via SCI</i> .....         | 45 |
| <i>Switches, LEDs and Data Exchange via SCI</i> .....                | 46 |
| <i>Switches and LEDs on SCI, Data Exchange via SSC and SCI</i> ..... | 47 |
| <i>SCI and SSC Used for Data Exchange (No Fieldbus I/O)</i> .....    | 48 |
| Initialization Examples, Automatic Initialization .....              | 49 |
| <i>Switches and LEDs on SSC</i> .....                                | 49 |
| <i>Preconfigured Node Address, no LEDs</i> .....                     | 50 |

## Chapter 8 Parameters

|  |    |
|--|----|
| General Information .....                | 51 |
| General Parameters .....                 | 52 |
| <i>Module Mode (#1)</i> .....            | 53 |
| <i>Module Status (#2)</i> .....          | 54 |
| <i>Module Type (#3)</i> .....            | 55 |
| <i>Fieldbus Type (#4)</i> .....          | 55 |
| <i>LED State (#7)</i> .....              | 56 |
| <i>Configuration Bits (#8)</i> .....     | 57 |
| <i>Switch Coding (#9)</i> .....          | 59 |
| <i>Offline Action Config (#10)</i> ..... | 60 |
| <i>Idle Action Config (#11)</i> .....    | 61 |
| <i>Interrupt Config (#12)</i> .....      | 62 |
| <i>Interrupt Cause (#13)</i> .....       | 63 |
| <i>SCI Rate Config (#14)</i> .....       | 64 |
| <i>SCI Rate Actual (#15)</i> .....       | 65 |
| <i>SCI Settings Config (#16)</i> .....   | 66 |
| <i>SCI Settings Actual (#17)</i> .....   | 67 |
| <i>MIF Rate Config (#18)</i> .....       | 68 |
| <i>MIF Rate Actual (#19)</i> .....       | 69 |
| <i>MIF Settings Config (#20)</i> .....   | 70 |
| <i>MIF Settings Actual (#21)</i> .....   | 71 |
| <i>Modbus RTU Address (#22)</i> .....    | 71 |
| <i>Modbus CRC Disable (#23)</i> .....    | 72 |
| <i>FB Fault Values (#27)</i> .....       | 72 |

|                                       |    |
|---------------------------------------|----|
| I/O Parameters.....                   | 73 |
| <i>FB Byte Order (#40)</i> .....      | 74 |
| <i>FB Out Config (#41)</i> .....      | 74 |
| <i>FB Out Actual (#42)</i> .....      | 75 |
| <i>FB In Actual (#43)</i> .....       | 75 |
| <i>FB In SSC Offset (#44)</i> .....   | 75 |
| <i>FB In SSC Size (#45)</i> .....     | 76 |
| <i>FB In SCI Offset (#46)</i> .....   | 76 |
| <i>FB In SCI Size (#47)</i> .....     | 76 |
| <i>SSC Byte Order (#50)</i> .....     | 77 |
| <i>SSC In Config (#51)</i> .....      | 77 |
| <i>SSC In Auto (#52)</i> .....        | 78 |
| <i>SSC In Actual (#53)</i> .....      | 78 |
| <i>SSC Out Config (#54)</i> .....     | 78 |
| <i>SSC Out Auto (#55)</i> .....       | 79 |
| <i>SSC Out Actual (#56)</i> .....     | 79 |
| <i>SSC Out FB Offset (#57)</i> .....  | 79 |
| <i>SSC Out FB Size (#58)</i> .....    | 80 |
| <i>SSC Out SCI Offset (#59)</i> ..... | 80 |
| <i>SSC Out SCI Size (#60)</i> .....   | 80 |
| <i>SCI Byte Order (#63)</i> .....     | 81 |
| <i>SCI In Config (#64)</i> .....      | 81 |
| <i>SCI In Actual (#65)</i> .....      | 82 |
| <i>SCI Out Actual (#66)</i> .....     | 82 |
| <i>SCI Out FB Offset (#67)</i> .....  | 82 |
| <i>SCI Out FB Size (#68)</i> .....    | 82 |
| <i>SCI Out SSC Offset (#69)</i> ..... | 83 |
| <i>SCI Out SSC Size (#70)</i> .....   | 83 |
| Fieldbus Specific Parameters .....    | 84 |

## Chapter 9 Miscellaneous

|   |    |
|---|----|
| Interrupt (/INT) & Bootloader Enable (BLE)..... | 85 |
| Reset (/RESET) .....                            | 85 |
| Self Test Sequence.....                         | 86 |
| <i>General Information</i> .....                | 86 |
| <i>Basic Procedure</i> .....                    | 86 |
| <i>Test Evaluation</i> .....                    | 86 |

## Appendix A Mechanical Specification

|                           |    |
|---------------------------|----|
| General Information ..... | 87 |
| Measurements .....        | 87 |

## Appendix B Firmware Upgrade

## Appendix C Object Messaging (0x5B)

|                                  |    |
|----------------------------------|----|
| General Information .....        | 89 |
| Message Format.....              | 89 |
| Subfield Contents .....          | 90 |
| <i>Fragment Byte Count</i> ..... | 90 |
| <i>Fragment Protocol</i> .....   | 90 |
| <i>Class ID</i> .....            | 90 |
| <i>Instance ID</i> .....         | 90 |
| <i>Error Codes</i> .....         | 91 |
| <i>Service Code</i> .....        | 92 |
| <i>Attribute</i> .....           | 92 |
| <i>Data</i> .....                | 92 |
| <i>Stuff Byte</i> .....          | 92 |

## Appendix D Technical Specification

|                                     |    |
|-------------------------------------|----|
| Environmental .....                 | 93 |
| Power Supply .....                  | 93 |
| Signal Levels .....                 | 94 |
| Regulatory Compliance.....          | 95 |
| <i>Fieldbus Certification</i> ..... | 95 |
| <i>EMC Compliance (CE)</i> .....    | 95 |
| <i>UL/cUL-Certificate</i> .....     | 95 |

## P. About This Document

For more information, documentation etc., please visit the HMS website, ‘[www.anybus.com](http://www.anybus.com)’.

### P.1 Related Documents

| Document                        | Author  |
|---------------------------------|---------|
| Anybus-IC Profibus DP Appendix  | HMS     |
| Anybus-IC DeviceNet Appendix    | HMS     |
| Anybus-IC EtherNet/IP Appendix  | HMS     |
| Anybus-IC Profinet IO Appendix  | HMS     |
| Anybus-IC CANopen Appendix      | HMS     |
| Modbus Protocol Reference Guide | Modicon |
| -                               | -       |
|                                 |         |
|                                 |         |
|                                 |         |
|                                 |         |



## P.2 Document History

### Summary of Recent Changes (2.00... 2.01)

| Change  | Page(s) |
|---|---------|
| Added note about byte sized parameters                              | 51      |
| Added FBBR bit to Configuration Bits (#8)                           | 57      |
| Altered output signal level from 3.3V to 5V in signal level picture | 94      |
|   |         |
|   |         |

### Revision List

| Revision | Date       | Author | Chapter(s)       | Description                           |
|----------|------------|--------|------------------|---------------------------------------|
| <1.50    | -          | -      | -                | (See previous revisions)              |
| 1.50     | 2006-08-10 | PeP    | -                | Major rewrite                         |
| 1.51     | 2007-01-09 | PeP    | -                | Minor update                          |
| 1.52     | 2007-04-02 | PeP    | 5, 8, 9, C       | Minor corrections                     |
| 1.53     | 2008-04-14 | PeP    | 3, 4, 8, 7, B    | Minor update                          |
| -        | -          | -      | -                | -                                     |
| 1.58     | 2008-11-06 | HeS    | 3, 4, 9, B, C    | Minor updates                         |
| 1.59     | 2008-11-17 | HeS    | 7                | Minor update                          |
| 1.60     | 2011-03-18 | KeL    | P, 1, 3, 6, 7, C | Minor update                          |
| 2.00     | 2012-05-22 | KaD    | 2, 8             | New template, new chapter and updates |
| 2.01     | 2013-02-20 | KaD    | 8, D             | Minor update                          |
|          |            |        |                  |                                       |
|          |            |        |                  |                                       |

## P.3 Document Conventions

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term ‘Anybus’ or ‘module’ is used when referring to the Anybus-IC.
- The terms ‘application’ is used when referring to the hardware that hosts the Anybus-IC
- Hexadecimal values are written in the format 0xNNNN, where NNNN is the hexadecimal value.
- Measurements expressed in this document have a tolerance of  $\pm 0.20$  mm unless otherwise stated.
- Signals which are “pulled to NN” are connected to NN via a resistor.
- Signals which are “tied to NN” are directly connected to NN.
- Modbus register numbers are specified using the protocol convention (base 0), i.e. there is a 1:1 correlation between the register number specified in this document and the actual register value in the message frame.

## P.4 Sales and Support

| Sales                    |                           | Support    |                             |
|--------------------------|---------------------------|------------|-----------------------------|
| HMS Sweden (Head Office) |                           |            |                             |
| E-mail:                  | sales@hms-networks.com    | E-mail:    | support@hms-networks.com    |
| Phone:                   | +46 (0) 35 - 17 29 56     | Phone:     | +46 (0) 35 - 17 29 20       |
| Fax:                     | +46 (0) 35 - 17 29 09     | Fax:       | +46 (0) 35 - 17 29 09       |
| Online:                  | www.anybus.com            | Online:    | www.anybus.com              |
| HMS North America        |                           |            |                             |
| E-mail:                  | us-sales@hms-networks.com | E-mail:    | us-support@hms-networks.com |
| Phone:                   | +1-312 - 829 - 0601       | Phone:     | +1-312-829-0601             |
| Toll Free:               | +1-888-8-Anybus           | Toll Free: | +1-888-8-Anybus             |
| Fax:                     | +1-312-629-2869           | Fax:       | +1-312-629-2869             |
| Online:                  | www.anybus.com            | Online:    | www.anybus.com              |
| HMS Germany              |                           |            |                             |
| E-mail:                  | ge-sales@hms-networks.com | E-mail:    | ge-support@hms-networks.com |
| Phone:                   | +49 (0) 721-989777-000    | Phone:     | +49 (0) 721-989777-000      |
| Fax:                     | +49 (0) 721-989777-010    | Fax:       | +49 (0) 721-989777-010      |
| Online:                  | www.anybus.de             | Online:    | www.anybus.de               |
| HMS Japan                |                           |            |                             |
| E-mail:                  | jp-sales@hms-networks.com | E-mail:    | jp-support@hms-networks.com |
| Phone:                   | +81 (0) 45-478-5340       | Phone:     | +81 (0) 45-478-5340         |
| Fax:                     | +81 (0) 45-476-0315       | Fax:       | +81 (0) 45-476-0315         |
| Online:                  | www.anybus.jp             | Online:    | www.anybus.jp               |
| HMS China                |                           |            |                             |
| E-mail:                  | cn-sales@hms-networks.com | E-mail:    | cn-support@hms-networks.com |
| Phone:                   | +86 (0) 10-8532-3183      | Phone:     | +86 (0) 10-8532-3023        |
| Fax:                     | +86 (0) 10-8532-3209      | Fax:       | +86 (0) 10-8532-3209        |
| Online:                  | www.anybus.cn             | Online:    | www.anybus.cn               |
| HMS Italy                |                           |            |                             |
| E-mail:                  | it-sales@hms-networks.com | E-mail:    | it-support@hms-networks.com |
| Phone:                   | +39 039 59662 27          | Phone:     | +39 039 59662 27            |
| Fax:                     | +39 039 59662 31          | Fax:       | +39 039 59662 31            |
| Online:                  | www.anybus.it             | Online:    | www.anybus.it               |
| HMS France               |                           |            |                             |
| E-mail:                  | fr-sales@hms-networks.com | E-mail:    | fr-support@hms-networks.com |
| Phone:                   | +33 (0) 3 68 368 034      | Phone:     | +33 (0) 3 68 368 033        |
| Fax:                     | +33 (0) 3 68 368 031      | Fax:       | +33 (0) 3 68 368 031        |
| Online:                  | www.anybus.fr             | Online:    | www.anybus.fr               |
| HMS UK & Eire            |                           |            |                             |
| E-mail:                  | uk-sales@hms-networks.com | E-mail:    | support@hms-networks.com    |
| Phone:                   | +44 (0) 1926 405599       | Phone:     | +46 (0) 35 - 17 29 20       |
| Fax:                     | +44 (0) 1926 405522       | Fax:       | +46 (0) 35 - 17 29 09       |
| Online:                  | www.anybus.co.uk          | Online:    | www.anybus.com              |
| HMS Denmark              |                           |            |                             |
| E-mail:                  | dk-sales@hms-networks.com | E-mail:    | support@hms-networks.com    |
| Phone:                   | +45 (0) 35 38 29 00       | Phone:     | +46 (0) 35 - 17 29 20       |
| Fax:                     | +46 (0) 35 17 29 09       | Fax:       | +46 (0) 35 - 17 29 09       |
| Online:                  | www.anybus.com            | Online:    | www.anybus.com              |
| HMS India                |                           |            |                             |
| E-mail:                  | in-sales@hms-networks.com | E-mail:    | in-support@hms-networks.com |
| Phone:                   | +91 (0) 20 40111201       | Phone:     | +91 (0) 20 40111201         |
| Fax:                     | +91 (0) 20 40111105       | Fax:       | +91 (0) 20 40111105         |
| Online:                  | www.anybus.com            | Online:    | www.anybus.com              |

# 1. About the Anybus-IC

## 1.1 General

The Anybus-IC network communication module is a high performance, low cost communication solution for industrial field devices. Typical applications include I/O blocks, temperature controllers, measuring devices, and other devices where size is an issue.

A flexible application interface enables the module to operate either standalone or controlled by a microcontroller. A range of communication channels allows fieldbus data exchange as well as internal discrete I/O. In addition, data can be mapped between the various channels in a flexible manner without any intervention by the host system.

The Anybus-IC software interface is designed to be network protocol independent, making it possible to support several networking systems using the same software driver. The host communication is based on Modbus RTU, a proven protocol which is easy to implement yet flexible enough to provide room for future expansion. This also allows for multidrop configurations where multiple Anybus-IC modules are interfaced to the same microcontroller.

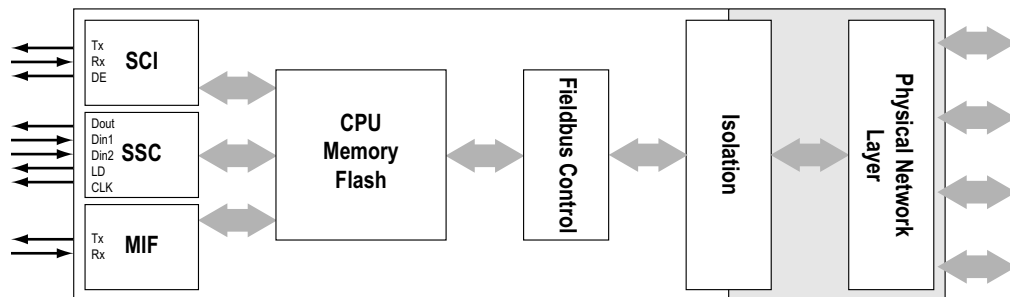
## 1.2 Features

- Standard 32-pin dual in line footprint
- Requires only a single 5 V power supply
- Galvanically isolated fieldbus electronics
- Serial Communications Interface (SCI)
- Synchronous Serial Channel (SSC)
- Text based configuration- and monitoring interface (MIF)
- Up to 144 bytes of fieldbus I/O in each direction
- Up to 128 bytes of serial I/O in each direction (SCI)
- Up to 128 bits of I/O in each direction (SSC)
- Flexible mapping of data
- Standalone or microcontroller operation

## 1.3 Overview

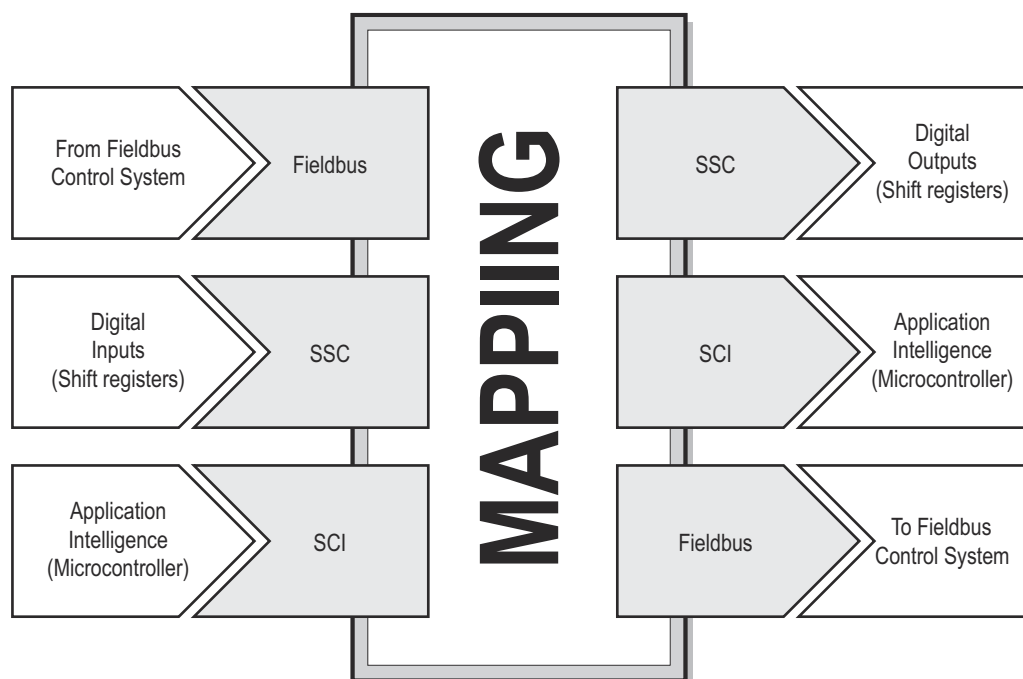
### Internals

The figure below illustrates the basic properties of the Anybus-IC, on-board I/O, fieldbus interface etc.



### Data Mapping

The module features a flexible data mapping scheme; data received on one communication channel can be mapped (i.e copied) to the other, and vice versa. This allows for not just fieldbus connectivity, but also for internal I/O in the application, by mapping SSC I/O to the SCI channel. By mapping fieldbus I/O to the SSC channel, SSC I/O can be accessed directly from the fieldbus.



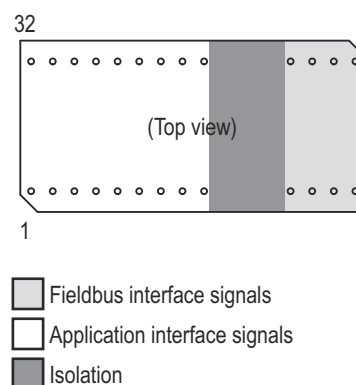
## 1.4 Application Connector

The application connector is based on a standard DIL-32 footprint.

See also...

- “Measurements” on page 87
- “MIF Interface” on page 21
- “SCI Channel” on page 25
- “SSC Channel” on page 31
- “Technical Specification” on page 93

**Note:** Pins 13... 20 are reserved for the fieldbus interface. Consult each separate fieldbus appendix for further information.



| Pin      | Signal         | Description  | Direction                   | Page |
|----------|----------------|--|-----------------------------|------|
| 1        | Vcc            | +5 V Power Supply  | Input                       | D.2  |
| 2        | /SSC_Reset_Out | SSC Reset signal (Active Low)  | Output                      | 5.   |
| 3        | /SSC_LD        | SSC Load signal (Active Low)   | Output                      | 5.   |
| 4        | SSC_DO         | SSC Data Output  | Output                      | 5.   |
| 5        | SSC_DI2        | SSC Data Input 2   | Input                       | 5.   |
| 6        | SSC_DI1        | SSC Data Input 1   | Input                       | 5.   |
| 7        | SSC_CLK        | SSC Clock  | Output                      | 5.   |
| 8        | /RESET         | Module reset (Active Low)  | Input                       | 9.2  |
| 9        | Vcc            | +5 V Power Supply  | Input                       | D.2  |
| 10... 12 | NC             | -  | -                           | -    |
| 13       | FB1            | Fieldbus interface signals<br><br>(consult the fieldbus appendix for further information about the fieldbus interface) | (fieldbus specific)         | -    |
| 14       | FB2            |  |                             |      |
| 15       | FB3            |  |                             |      |
| 16       | FB4            |  |                             |      |
| 17       | FB7            |  |                             |      |
| 18       | FB8            |  |                             |      |
| 19       | FB5            |  |                             |      |
| 20       | FB6            |  |                             |      |
| 21... 23 | NC             | -  | -                           | -    |
| 24       | GND            | GND Power Supply   | -                           | D.2  |
| 25       | NC             | -  | -                           | -    |
| 26       | /INT [BLE]     | Interrupt (Active Low) [Boot loader enable switch]   | Output [Input] <sup>a</sup> | 9.1  |
| 27       | MIF_Tx         | MIF Transmit signal  | Output                      | 3.   |
| 28       | MIF_Rx         | MIF Receive signal   | Input                       | 3.   |
| 29       | SCI_DE [AUTO]  | SCI Data Enable [Autoinitialization]   | Output [Input]              | 4.   |
| 30       | SCI_Tx         | SCI Transmit signal  | Output <sup>a</sup>         | 4.   |
| 31       | SCI_Rx         | SCI Receive signal   | Input                       | 4.   |
| 32       | GND            | GND Power Supply   | -                           | D.2  |

a. Maximum drainable current:  $\pm 2$  mA.

## 2. Tutorial

### 2.1 Introduction

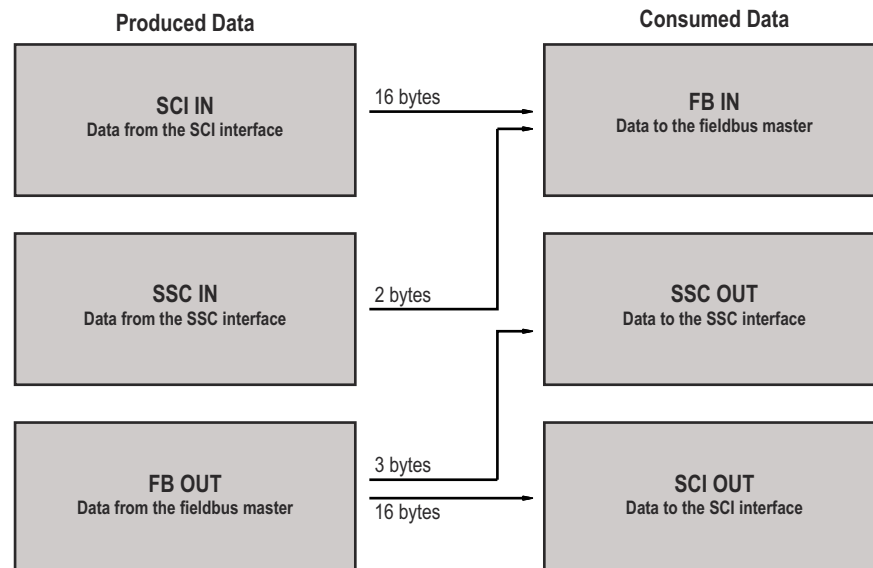
This tutorial will describe how to map data between the three different communication channels (Fieldbus, SSC, SCI) and how to communicate with the Anybus-IC using the SCI interface and a Modbus master. It is assumed that the Anybus-IC is socketed and ready, and that all pins and connectors are connected and ready for use.

The Anybus-IC has three possible modes of operation. For a detailed description of each operation mode, see “Initialization” on page 43. This tutorial will describe how to set up normal mode communication, e.g. where the module shall be controlled by a microcontroller via the SCI channel.

As said above, the Anybus-IC features three possible communication channels:

- **SSC (Synchronous Serial Channel)**  
Data to and from the digital outputs/inputs of the application, e.g. LEDs and switches.
- **SCI (Serial Communications Interface)**  
Data coming from and going to the Modbus master of the host application.
- **FB (Fieldbus)**  
Data coming from and going to the fieldbus side of the application. Data from the fieldbus to the Anybus-IC is called FB OUT, and data from the Anybus-IC to the fieldbus is called FB IN.

This tutorial will feature an example using all three communication channels, with 19 bytes of input data and 18 bytes of output data, mapped according to the picture below:



There are 19 bytes coming from the fieldbus (FB OUT). The first 3 bytes are sent to the SSC interface (SSC OUT), and the remaining 16 bytes to the SCI interface (SCI OUT).

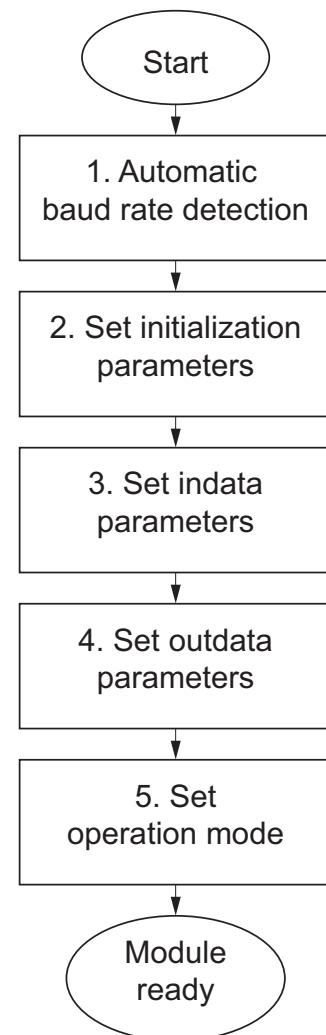
Two bytes of data will be sent from the SSC interface (SSC IN) and 16 bytes from the SCI interface (SCI IN) to the fieldbus (FB IN). A total of 18 bytes.

## 2.2 Step by Step Guide

To implement the application presented above, follow this step by step guide.

1. Detect the baud rate of the SCI channel. See “Automatic Baud Rate Detection” on page 17.
2. Set the initialization parameters. See “Set Initialization Parameters” on page 17.
3. Set the indata parameters. See “Set Indata Parameters” on page 18.
4. Set the outdata parameters. See “Set Outdata Parameters” on page 19.
5. Set operation mode. See “Set Operation Mode” on page 20.

The Anybus-IC module is now configured and should be running smoothly, exchanging data between the three communication interfaces.



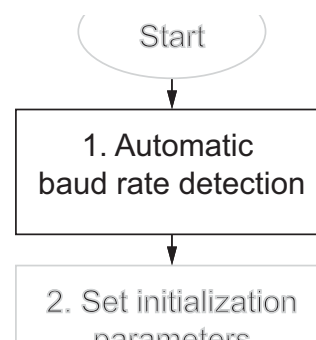


## 2.3 Automatic Baud Rate Detection

By default, the Anybus-IC will try to detect the baud rate of the SCI communication channel automatically.

In order to do that, the Modbus server must repeatedly issue “read holding registers”-requests to the module at start-up.

See “Baud Rate” on page 27 for more information.

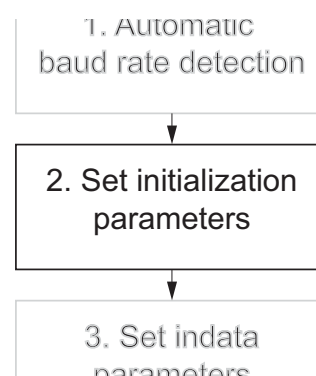


## 2.4 Set Initialization Parameters

The Anybus-IC will now be prepared for operation in normal mode.

To configure the Anybus-IC for normal mode operation, two parameters must be set.

- Parameter “Configuration Bits (#8)” involves setting initialization values for different operation scenarios. For a complete description, see “Configuration Bits (#8)” on page 57.
- Parameter “SCI Rate Config (#14)” is used to set the baud rate for the communication on the SCI channel. The value 0x00 that will be used in this example keeps the default behavior, which is to use automatic baud detection.



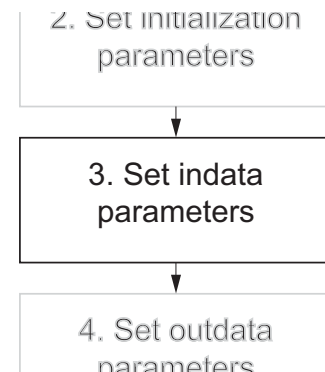
The parameters in the Anybus-IC are set using write register calls issued from the Modbus master. The parameters and their respective values to use are presented in the table below:

| Parameter               | Modbus Address | Setting | Description  |
|-------------------------|----------------|---------|--|
| Configuration Bits (#8) | 0x5008         | FBLP=1  | The Fieldbus Specific Output Register will be used for normal data exchange. See “Fieldbus Specific Output Register (Output Register 0)” on page 34. |
|                         |                | SSCI=1  | The SSC Input size will be configured manually. The size will be set using the SSC In Config parameter (#51).  |
|                         |                | SSCO=1  | The SSC Output size will be configured manually. The size will be set using the SSC Out Config parameter (#54).                                      |
| SCI Rate Config (#14)   | 0x500E         | 0x00    | Use automatic baud rate detection.<br>The default value is 0x00.   |

## 2.5 Set Indata Parameters

Indata is data coming from any of the three possible communication channels (Fieldbus, SCI, SSC) to the Anybus-IC module's incoming data buffers.

When setting the indata parameters, the only thing that needs to be done is to define the amount of data coming from each channel. This is done using the three parameters in the table below.



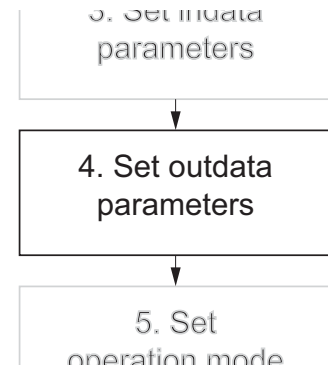
| Parameter           | Modbus Address | Setting | Description   |
|---------------------|----------------|---------|---|
| FB Out Config (#41) | 0x6001         | 0x0013  | The size of the FB Output area. Data coming from the fieldbus.      |
| SSC In Config (#51) | 0x600B         | 0x0002  | The size of the SSC Input area. Data coming from the SSC interface. |
| SCI In Config (#64) | 0x6018         | 0x0010  | The size of the SCI Input area. Data coming from the SCI interface. |

## 2.6 Set Outdata Parameters

Each of the three communication channels' incoming data buffers can transfer data to the outgoing data buffers of the two other channels.

Hence, since every incoming data buffer can contain data that should be transferred to two other outgoing data buffers, two things must be set using the outdata parameters:

- the amount of data bytes to be transferred.
- the location of the data in the incoming data buffer (the offset).



Since the SSCO bit (Configuration Bits #8) was set in the initialization phase above, the SSC Out Config-parameter is used to manually set the size of the SSC output data area.

In this example, data comes from both the SSC and the SCI interfaces to the FB input area. SSC data will always be mapped before SCI data.

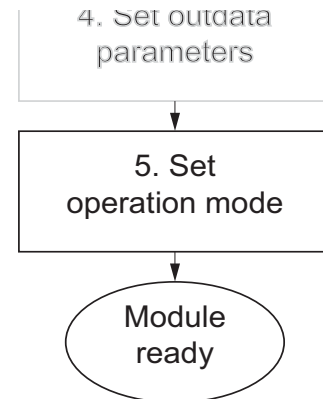
| Parameter               | Modbus Address | Setting | Description   |
|-------------------------|----------------|---------|---|
| FB In SSC Offset (#44)  | 0x6004         | 0x0000  | The source location of the data in the SSC Input area going to the FB Input area.   |
| FB In SSC Size (#45)    | 0x6005         | 0x0002  | The amount of data going from the SSC Input area to the FB Input area.              |
| FB In SCI Offset (#46)  | 0x6006         | 0x0000  | The source location of the data in the SCI Input area going to the FB Input area.   |
| FB In SCI Size (#47)    | 0x6007         | 0x0010  | The amount of data going from the SCI Input area to the FB Input area.              |
| SSC Out Config (#54)    | 0x600E         | 0x0003  | The size of the SSC Output area. Data going to the SSC interface.                   |
| SSC Out FB Offset (#57) | 0x6011         | 0x0000  | The source location of the data in the FB Output area going to the SSC Output area. |
| SSC Out FB Size (#58)   | 0x6012         | 0x0003  | The amount of data going from the FB Output area to the SSC Output area.            |
| SCI Out FB Offset (#67) | 0x601B         | 0x0003  | The source location of the data in the FB Output area going to the SCI Output area. |
| SCI Out FB Size (#68)   | 0x601C         | 0x0010  | The amount of data going from the FB Output area to the SCI Output area.            |

## 2.7 Set Operation Mode

After all parameters are set in the steps above, the operation mode for the application needs to be specified.

There are three possible operation modes for the Anybus-IC. See “Initialization” on page 43 for more information. This tutorial describes the setup for normal mode operation, e.g. where the SCI interface is used to provide intelligence to the application.

To start normal operation, set parameter #1 (“Module Mode”) to 0x0001. It is not possible to set this value until the baud rate has been detected. See “Automatic Baud Rate Detection” on page 17.



| Parameter        | Modbus Address | Setting | Description  |
|------------------|----------------|---------|--|
| Module Mode (#1) | 0x5001         | 0x0001  | Normal operation. See “Module Mode (#1)” on page 53. |

### Module Ready

When normal operation has been started, the Anybus-IC will automatically calculate memory areas, using the parameter values set above, and initiate the communication between the fieldbus and the application.

## 3. MIF Interface

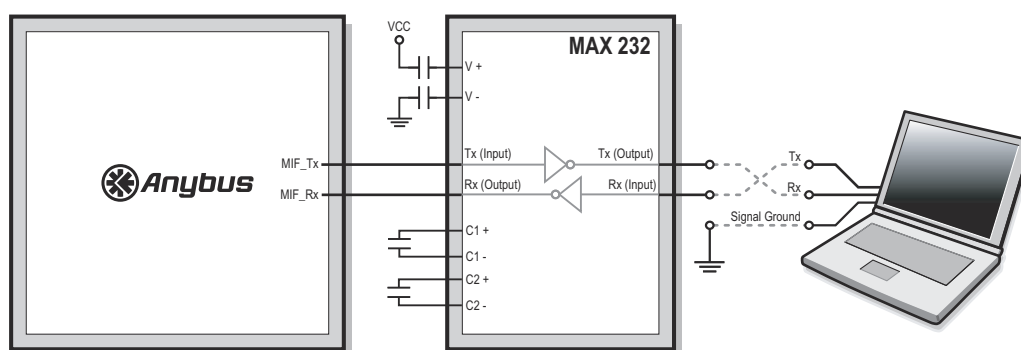
### 3.1 General Information

The MIF interface provides an easy way to monitor data and access parameters through a terminal-based user interface.

This interface uses the following signals:

- **MIF\_Tx (Pin 27)**  
Asynchronous serial output; carries data from the Anybus module to the terminal.  
(If not used, leave this signal unconnected).
- **MIF\_Rx (Pin 28)**  
Asynchronous serial input; carries data from the terminal to the Anybus module.  
(If not used, leave this signal unconnected).

In the following example, the MIF interface is connected to a PC via a MAX232 transceiver and a crossed (a.k.a. “null modem”) cable.



It is important not to confuse the MIF interface with the SCI-channel. Although theoretically possible, it is strongly discouraged to use the MIF interface as a configuration channel for the application, since menu entries may be added, changed, or even removed, in future software revisions.

**Note:** Implementation of this interface is optional. It is however highly recommended to at least implement rudimentary support by adding dedicated solder pads or similar, since the use of this interface may provide valuable clues during development and testing.

See also...

- “Application Connector” on page 14
- “SCI Channel” on page 25

## 3.2 Communication Settings

### 3.2.1 Baud rate

The MIF interface supports baud rates from 4.8 kbps to 57.6 kbps (default is 38.4 kbps). Unlike the SCI interface, automatic baud rate detection is not supported.

See also...

- “MIF Rate Config (#18)” on page 68
- “MIF Rate Actual (#19)” on page 69

### 3.2.2 Communication Properties

The interface supports 1 or 2 stop bits, none, odd or even parity. Default is 1 stop bit, no parity. The number of data bits is fixed to 8.

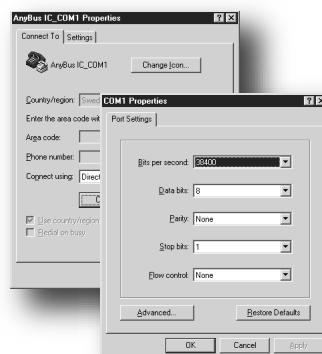
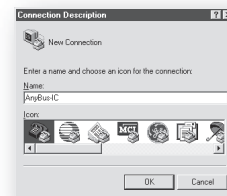
See also...

- “MIF Settings Config (#20)” on page 70
- “MIF Settings Actual (#21)” on page 71

### Terminal Configuration

The following example describes the configuration procedure when using the Windows HyperTerminal. The procedure should be similar when using other terminal emulation software.

1. Start the Windows HyperTerminal.
2. Open a new connection. Name the new connection.
3. Specify the COM-port used for the MIF interface.
4. Specify the baud rate and port settings. Ensure that these settings match the ones specified in the Anybus module. Please note that flow control is not supported.



## 3.3 User Interface

### 3.3.1 General Information

The MIF interface features a menu-based user interface as follows:

- To enter a submenu or parameter, type the corresponding digit and press <Enter>.
- To enter a parameter value, enter the value and press <Enter>.
- To return to a previous menu, or cancel a parameter input, press <ESC>.
- To redraw the current menu, press <Enter>.

### 3.3.2 Main Menu

The main menu provides access to the various submenus. It also displays the fieldbus type.

```

-----
Anybus-IC - Main Menu
Profibus-DP
-----
1 - Module Information
2 - Parameters
3 - Monitor
4 - Firmware Upgrade
-----
>

```

See also...

- “Module Information’-menu” on page 23
- “Parameters’-menu” on page 24
- “Monitor’-menu” on page 24
- “Firmware Upgrade’-menu” on page 24

### 3.3.3 ‘Module Information’-menu

This menu features two additional submenus; ‘Software Versions’, which provides information about the current Anybus-IC firmware, and ‘Product Information’, which provides manufacturing information such as serial number and production date.

```

-----
Anybus-IC - Information
-----
1 - Software Versions
2 - Product Information
-----
>

```

### 3.3.4 ‘Parameters’-menu

This menu provides access to all parameters in the module.

```
-----
Anybus-IC - Parameters
-----
 1 - Anybus-IC
 2 - FB I/O Settings
 3 - SSC I/O Settings
 4 - SCI I/O Settings
 5 - Fieldbus Specific
-----
>
```

See also...

- “Register Map” on page 26
- “Parameters” on page 51

### 3.3.5 ‘Monitor’-menu

All active I/O buffers can be monitored in this menu.

```
-----
Anybus-IC - I/O Areas
-----
 1 - Fieldbus Out
 2 - SSC In
 3 - SCI In
 4 - Fieldbus In
 5 - SSC Out
 6 - SCI Out
-----
>
```

*Example:*

Monitoring of ‘Fieldbus In’-area (submenu #4):

```
-----
Anybus-IC - Fieldbus In
-----
Byte #
 0      1100 0011    0xc3
 1      0000 0000    0x00
-----
>
```

**Note:** The information is not refreshed automatically. Press <Enter> to update the screen.

See also...

- “SCI Channel” on page 25
- “SSC Channel” on page 31

### 3.3.6 ‘Firmware Upgrade’-menu

This menu is used when upgrading the firmware of the module.

See also “Firmware Upgrade” on page 88.



## 4. SCI Channel

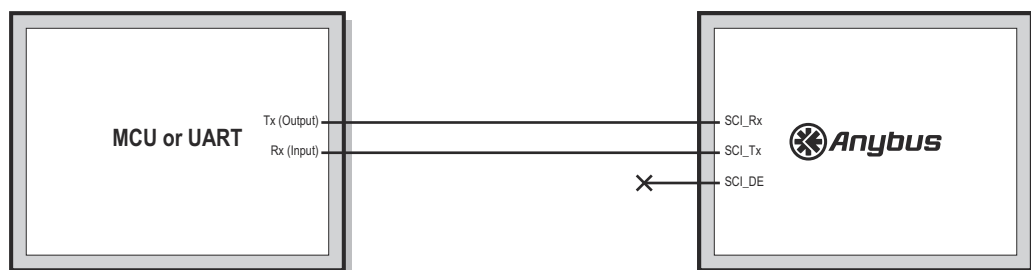
### 4.1 General Information

The SCI channel is a typical asynchronous serial interface and can be used for configuration and data exchange. The communication protocol is based on a subset of the Modbus-RTU standard.

This interface uses only three signals:

- **SCI\_Tx (Output, pin 30)**  
Asynchronous serial output; carries data from the Anybus module to the application.  
(If not used, leave this signal unconnected).
- **SCI\_Rx (Input, pin 31)**  
Asynchronous serial input; carries data from the application to the Anybus module.  
(If not used, leave this signal unconnected).
- **SCI\_DE (Input & Output, pin 29)**  
This signal enables data output on half duplex (e.g. RS-485) networks. Connecting this signal to GND disables the SCI channel altogether and causes the module to initialize automatically (i.e. for standalone operation).  
(If not used, leave this signal unconnected).

In the example below, the SCI\_Tx and SCI\_Rx signals are interfaced directly to an asynchronous serial interface on a microcontroller. SCI\_DE has intentionally been left unconnected.



See also...

- Modbus Protocol Reference Guide
- “Application Connector” on page 14
- “I/O Mapping” on page 39

## 4.2 Modbus Implementation

### 4.2.1 Functions Codes

The following function codes are implemented in the Anybus-IC module.

| #  | Name                                 | No. of simultaneous register R/W | Comments  |
|----|--------------------------------------|----------------------------------|---|
| 3  | Read Multiple Registers <sup>a</sup> | 125/-                            | Reads the contents of a sequence of registers.                      |
| 4  | Read Input Registers <sup>a</sup>    | 125/-                            |   |
| 6  | Write Single Register                |                                  | Writes to a single register <sup>b</sup>                            |
| 16 | Write Multiple Registers             | -/123                            | Writes to a sequence of registers <sup>b</sup>                      |
| 23 | Read / Write Registers               | 125/121                          | Writes to a sequence of registers and returns their previous values |
| 91 | Object Messaging                     |                                  | See "Object Messaging (0x5B)" on page 89                            |

a. These functions are identical in this implementation

b. The module supports broadcast in multidrop configurations

**Note:** All function codes share the same register map.

### 4.2.2 Register Map

Parameters and data are mapped to Modbus registers as follows:

| Register no. (Hex) | Description        | Comments  |
|--------------------|--------------------|---|
| 0000h... 003Fh     | SCI IN buffer      | 128 bytes (64 registers) in each direction; see also "I/O Mapping" on page 39 |
| 1000h... 103Fh     | SCI OUT buffer     |   |
| 5001h... 5033h     | General Parameters | See "General Parameters" on page 52   |
| 6000h... 6020h     | I/O Parameters     | See "I/O Parameters" on page 73   |
| 7000h...           | Fieldbus Specific  | Consult each separate fieldbus appendix                                       |

**Note 1:** Unmapped register regions are reserved for future use.

**Note 2:** In this document, register numbers are specified using the protocol convention (base 0), i.e. there is a 1:1 correlation between the register number specified in this document and the actual register value in the message frame.

### 4.2.3 Exception Codes

The following exception codes are used by the Anybus module.

| # | Name                 | Comments  |
|---|----------------------|---|
| 1 | Illegal Function     | Illegal (unsupported) function code in query                          |
| 2 | Illegal Data Address | Illegal Modbus register address in query                              |
| 3 | Illegal Data Value   | Register value specified in query is not valid                        |
| 4 | Slave Device Failure | An unrecoverable error occurred while processing the requested action |

## 4.3 Communication Settings

### 4.3.1 Baud Rate

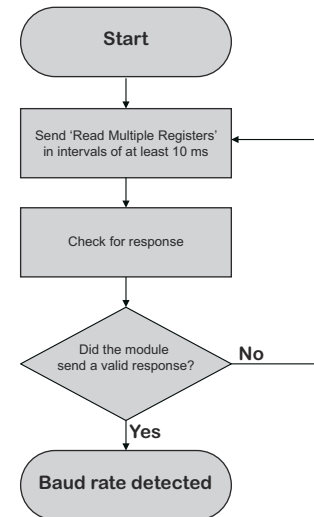
The SCI channel supports baud rates from 4.8 kbps to 57.6 kbps. By default, the module attempts to detect the baud rate automatically. This is not part of the Modbus RTU specification, and requires a special startup sequence (see flowchart).

- During start-up, the application must repeatedly issue 'Read Holding Registers'-requests to slave address 0x01. Wait at least 10 ms before reissuing the request.
- Example:** read data from Modbus register 0x5001 by issuing "01h 03h 50h 01h 00h 01h C4h CAh".
- The module responds when the correct baud rate has been established.
  - Generally the module detects the baud rate within 20 attempts.
  - In order for this functionality to work, the module must be configured to use slave address 0x01. This means that it is not possible to use automatic baud rate detection in multidrop systems.

**Note:** Some Anybus IC modules support further baud rates. Please see "SCI Rate Config (#14)" on page 64".

See also...

- "SCI Rate Actual (#15)" on page 65



### 4.3.2 Timeout

Under normal circumstances the Anybus-IC module will start to send a response within 50 ms of the receipt of a correct message. This time should be considered as a minimum timeout value.

### 4.3.3 Communication Properties

The number of data bits on the SCI channel is fixed to 8. The number of stop bits depends on the parity setting, which is accessed through parameter #16. Flow control is not supported.

Default settings are no parity / 2 stop bits.

See also...

- "SCI Settings Config (#16)" on page 66
- "SCI Settings Actual (#17)" on page 67

**Note:** If using automatic baud rate detection, these settings are detected by the module automatically.

### 4.3.4 Modbus RTU Address

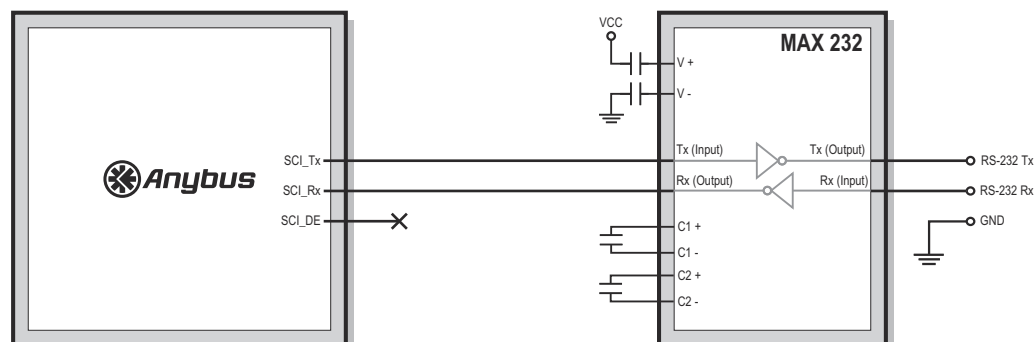
The communication protocol allows multiple Anybus modules to be interfaced to the same microcontroller (e.g. using a multi-drop RS-485 network). In such cases, each module must be assigned a unique Modbus RTU address.

See also...

- “Interfacing to RS-485 (Multidrop)” on page 30
- “Modbus RTU Address (#22)” on page 71

## 4.4 Interfacing to RS-232

In the example below, the MAX232 transceiver from Maxim is used to convert the SCI-signals to RS-232 levels. The SCI\_DE-signal is intentionally left unconnected.



### Cable Considerations

The cable length and quality have great impact on the maximum possible data rate. This is a limitation in the RS-232 standard itself, rather than a limitation in the Anybus module.

The maximum cable length depends on a number of factors, including how well the sender and receiver are implemented regarding rise times, cable capacitance and inductance etc. The original RS-232 specification states a maximum cable length of 15.25 metres at data rates up to 20.0 kbps, however in most real-life situations, the Anybus module can be used well outside these limitations.

The external environment has a large impact on the maximum cable lengths when using unshielded cables. In electrically noisy environments, even very short cables can pick up stray signals.

### General Recommendations

For best operation, be sure to follow the guidelines below.

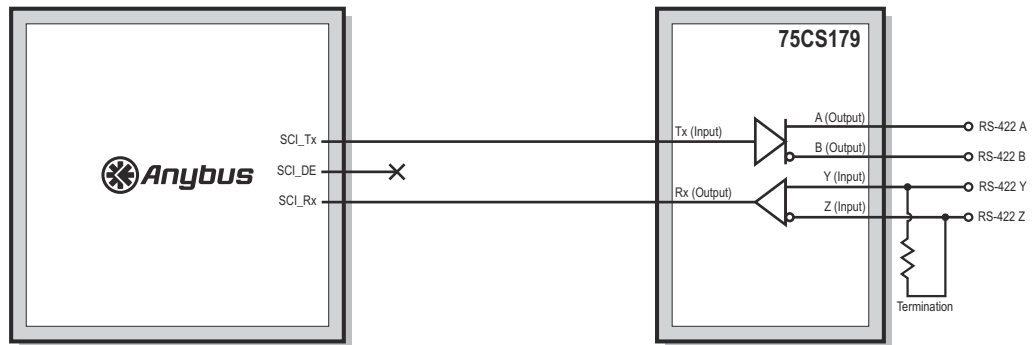
- Always use good quality shielded RS-232 cable
- Do not let the RS-232 cable run parallel to and close to power cables for more than 0.5 m.
- Do not wrap the RS-232 cable around other signal cables.
- For longer distances, use RS-422 or RS-485.

See also...

- “General Information” on page 25
- “Interfacing to RS-422” on page 30
- “Interfacing to RS-485 (Multidrop)” on page 30

## 4.5 Interfacing to RS-422

RS-422 uses balanced signals for data transmission, offering greater noise immunity compared to RS-232. In the example below, a 75CS179 transceiver is used to convert the SCI-signals to RS-422 levels. Again, the SCI\_DE-signal is intentionally left unconnected.



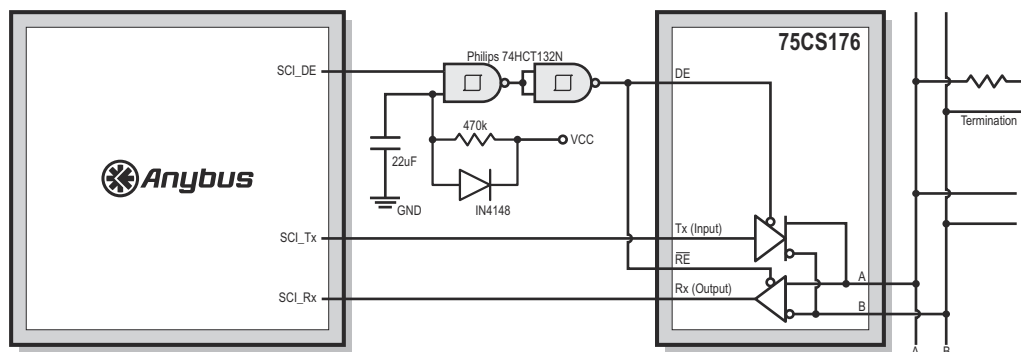
**Note:** It is recommended to use termination resistors to avoid reflections on the serial line.

See also...

- “General Information” on page 25
- “Interfacing to RS-232” on page 29
- “Interfacing to RS-485 (Multidrop)” on page 30

## 4.6 Interfacing to RS-485 (Multidrop)

RS-485 is similar to RS-422 and is used for multipoint communications, allowing multiple Anybus modules to be interfaced to the same microcontroller. In the example below, a 75CS176 transceiver is used to convert the SCI-signals to RS-485 levels.



The SCI\_DE-signal requires a special circuit, which prevents the Anybus module from sending undefined data to the RS-485 network during startup (the circuit prevents the module from accessing the RS-485 network for approximately 4 seconds after power on).

**Note:** It is recommended to use termination resistors to avoid reflections on the serial line.

See also...

- “General Information” on page 25
- “Interfacing to RS-232” on page 29
- “Interfacing to RS-422” on page 30

## 5. SSC Channel

### 5.1 General Information

The SSC channel uses a synchronous serial interface similar to the Motorola SPI, intended for discrete I/O and fieldbus-specific input/output signals such as node address and LED indications.

The Anybus-IC expects that any circuitry that is wired to the SSC is 100% sequence- and timing-compatible to the chain of 165/594 discrete shift registers mounted on the same circuit board as the module. The application designer using the SSC interface must make sure that their circuitry is, from a sequence and timing viewpoint, identical to a 165/594 shift register chain.

This interface uses 6 signals:

- **/SSC\_Reset\_Out (Pin 2)**  
Active low signal used to reset the shift registers. The use of this signal is optional.  
Note that this signal must not be confused with /RESET (pin 8).  
If not used, leave this signal unconnected.
- **/SSC\_LD (Pin 3)**  
Shift register load. Loads the value of the shift register inputs on transition to low.  
If not used, leave this signal unconnected.
- **SSC\_DO (Pin 4)**  
Serial data output. Valid on the falling edge of SSC\_CLK.  
If not used, leave this signal unconnected.
- **SSC\_DI2 (Pin 5)**  
Serial data input 1 from shift registers. Sampled on the rising edge of SSC\_CLK.  
If not used, leave this signal unconnected.
- **SSC\_DI1 (Pin 6)**  
Serial data input 2 from shift registers. Sampled on the rising edge of SSC\_CLK.
  - If only using Input Registers, connect this signal to SCC\_DO.
  - If only using Output Registers, connect this signal to SCC\_DI2.
  - If using both Input- and Output Registers, connect this signal to the serial data line between the Input- and Output Registers.If not used, leave this signal unconnected.
- **SSC\_CLK (Pin 7)**  
Clock output.  
If not used, leave this signal unconnected.

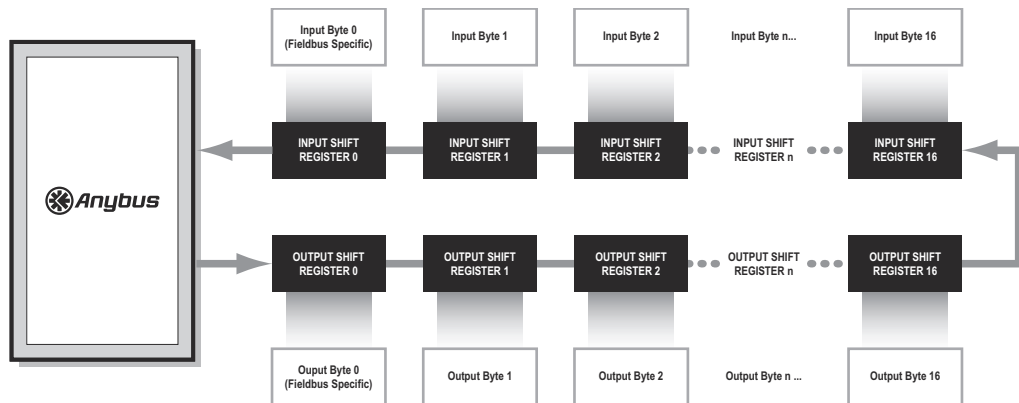
See also...

- “Application Connector” on page 14
- “I/O Mapping” on page 39

## 5.2 The Shift Register Loop

The SSC channel is intended to be used with an external shift register loop to form discrete inputs and outputs. It supports up to 17 registers (136 bits) in each direction, out of which 16 (128 bits) can be used for data exchange.

As illustrated below, the first Input- and Output Registers in the shift register loop are by default reserved for fieldbus specific functions. It is possible to disable this functionality and use those registers for data exchange instead. The maximum total amount of data is however still limited to 16 registers (128 bits) in each direction.



The module detects the number of shift registers automatically during startup. If needed, the number of registers can be specified through parameters #51 and #54. Parameter #8 ('Configuration Bits') determines whether to use manual or automatically detected settings.

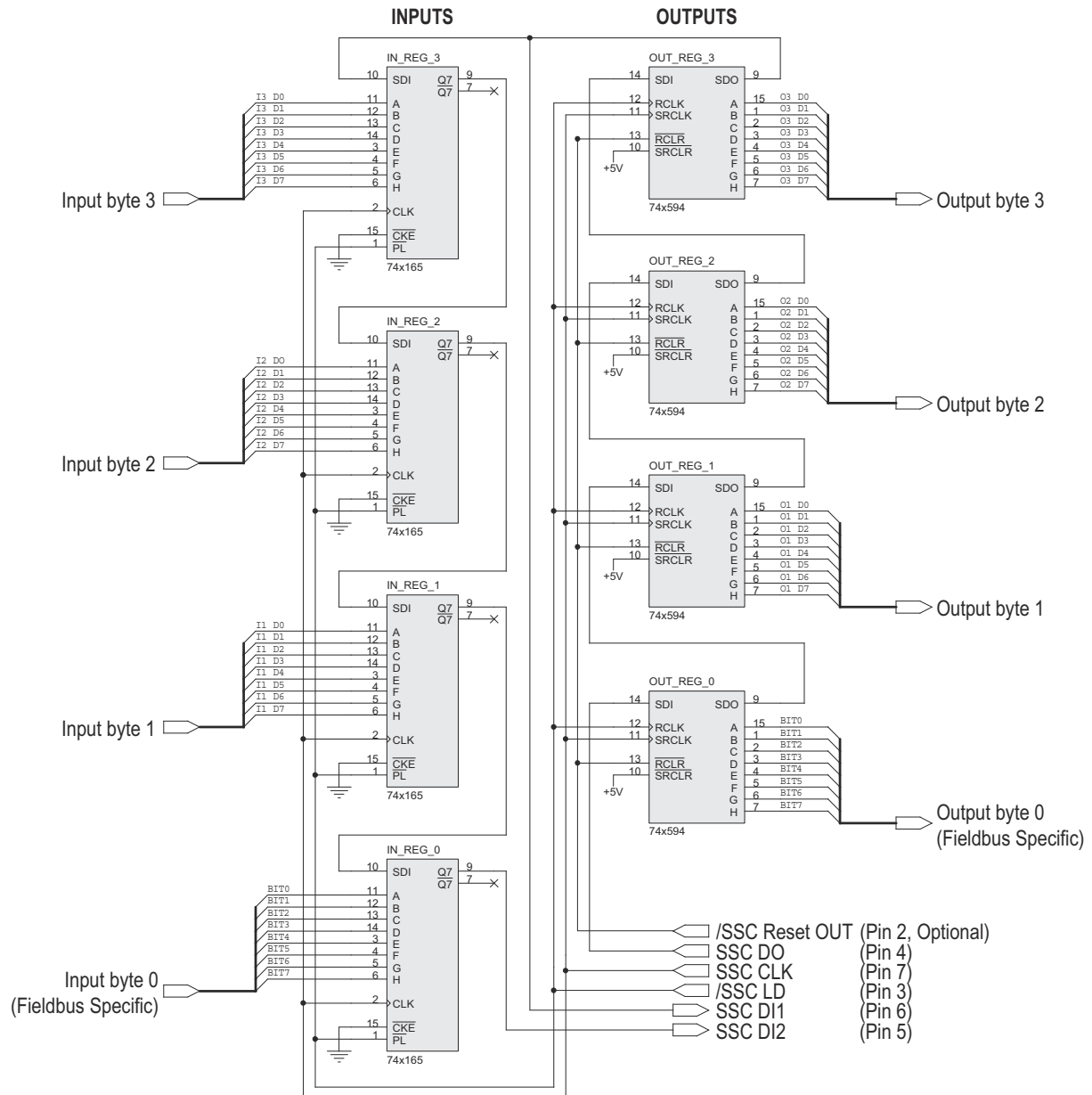
See also...

- "Sequence Diagrams" on page 37
- "Configuration Bits (#8)" on page 57
- "SSC In Config (#51)" on page 77
- "SSC In Auto (#52)" on page 78
- "SSC Out Config (#54)" on page 78
- "SSC Out Auto (#55)" on page 79



## 5.3 Basic Shift Register Circuit

The following schematic illustrates a basic shift register loop.



## 5.4 Fieldbus Specific Output Register (Output Register 0)

### 5.4.1 General

The fieldbus specific output register (Output Register 0), is used for fieldbus specific status indications (LEDs).

As mentioned previously, this functionality can be disabled by setting the 'FBLP'-bit in parameter #8 ('Configuration Bits'). In such a case, this register will be used for normal data exchange instead. Note however that this does not extend the number of possible outputs available for data exchange.

See also...

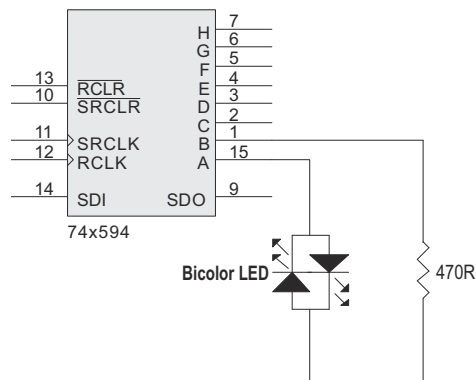
- "Configuration Bits (#8)" on page 57

**Note:** The Fieldbus Specific Output register can be used even if no SSC I/O data is present, e.g. if all data exchange is made through the SCI channel and only the LEDs are used on the SSC channel.

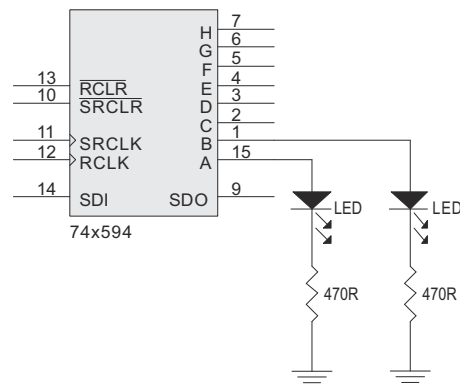
### 5.4.2 LEDs (no External Drivers)

This type of circuit requires shift registers of sufficient current capacity to drive the LEDs directly, e.g. 74x594 or equivalent.

#### Bicolor



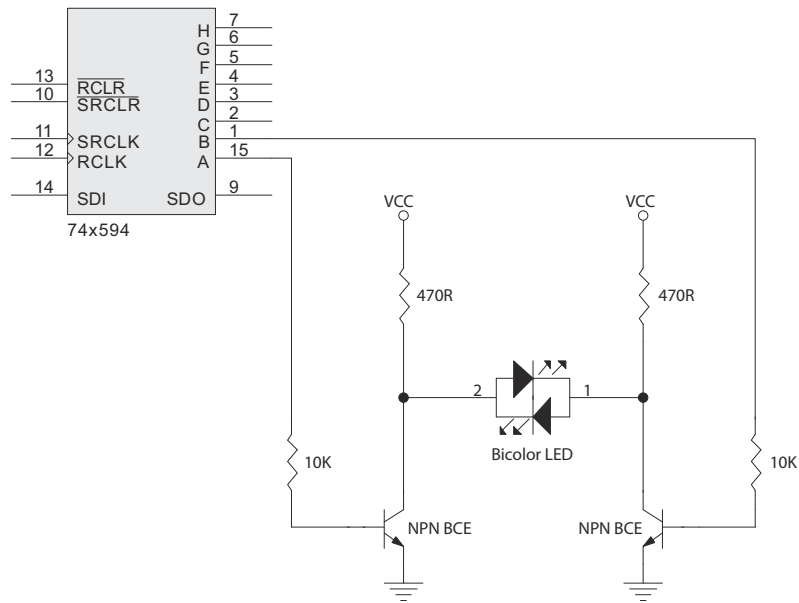
#### Single Color



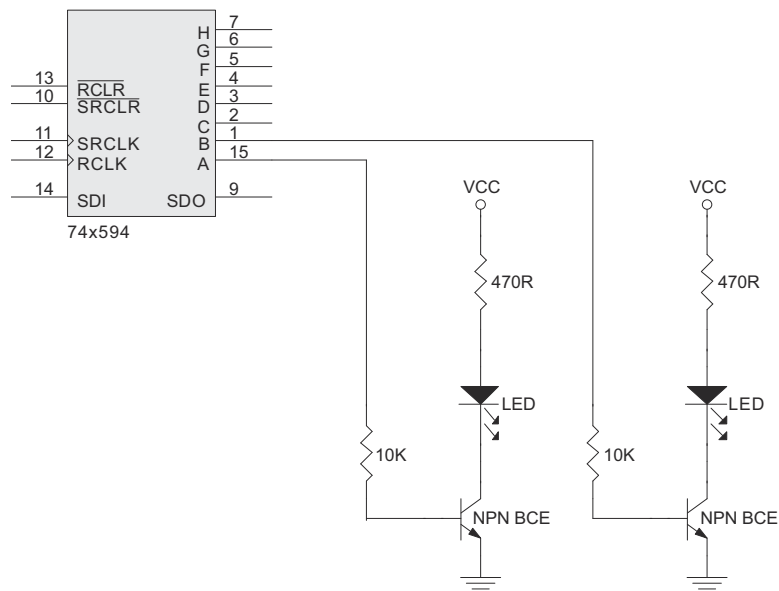
### 5.4.3 LEDs (External Driver Circuit)

This type of connection can be used with 74x594 or equivalent type shift registers.

#### Bicolor



#### Single Color



## 5.5 Fieldbus Specific Input Register (Input Register 0)

### 5.5.1 General

The fieldbus specific input register (Input Register 0) is used for fieldbus specific configuration settings (switches), such as node address and baud rate.

The module supports both binary and BCD-coded switches. The switch type is specified in parameter #9 ("Switch Coding"). Note that the module does not invert the individual bit values of this register; a low input voltage is interpreted as a logical zero (0), and a high input voltage is interpreted as a logical one (1).

This functionality can be disabled by setting the FBNP bit in parameter #8 ("Configuration Bits"). In such a case, this register will be used for normal data exchange instead. Note however that this does not extend the number of possible inputs available for data exchange.

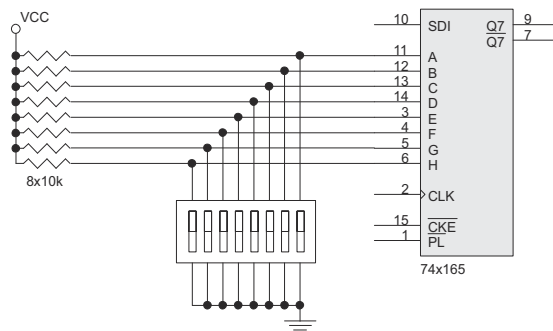
See also...

- "Configuration Bits (#8)" on page 57
- "Switch Coding (#9)" on page 59

**Note:** The Fieldbus Specific Input register can be used even if no SSC I/O data is present, e.g. if all data exchange is made through the SCI channel and only the node address is taken from the SSC channel.

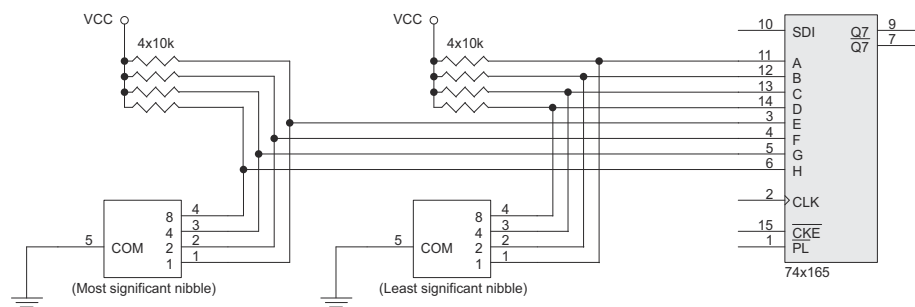
### 5.5.2 Binary Switches

A closed switch in the circuit below produces a logical zero (0).



### 5.5.3 BCD-coded Switches

The switches in the circuit below are of sinking kind (all bits are connected to common when the switch is in position zero).



## 5.6 Sequence Diagrams

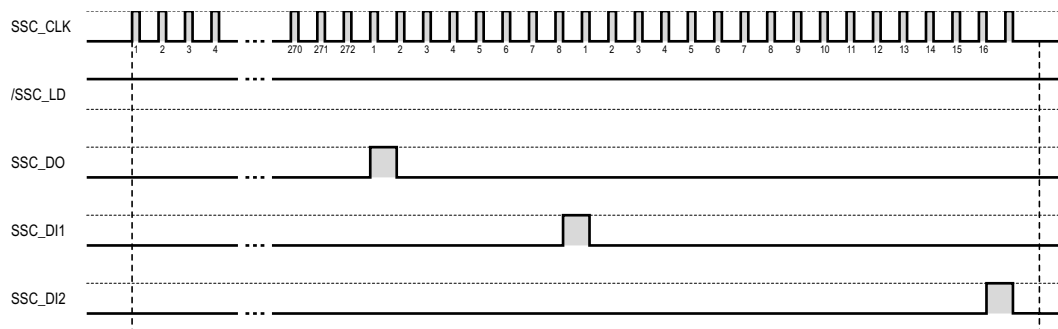
In order to ensure compatibility with all members of the Anybus-IC family the application developer must make sure that any circuitry connected to the SSC interface is fully sequence and timing compatible to a design using a full-size shift register chain, mounted on one single circuit board and built with the 74x165/74x594 shift registers as shown in the sample schematic on page 33.

Other shift registers and connection techniques may also work as long as they are sequence and timing compatible to the above.

### 5.6.1 Initialization Sequence (Startup Only)

A special initialization sequence is performed during startup to establish the number of attached shift registers. The module starts by clearing the internal register in all shift registers by shifting out zeroes the maximum number of times possible (17 Input Registers x 8 bit + 17 Output Registers x 8 bit = 272 times).

When cleared, a logical 1 (one) is shifted through the shift register loop. The module determines the number of Output and Input Registers by counting the number of clock cycles needed before that logical 1 (one) reaches DI1 and DI2. The example below uses 1 Output Register, and 2 Input Registers.

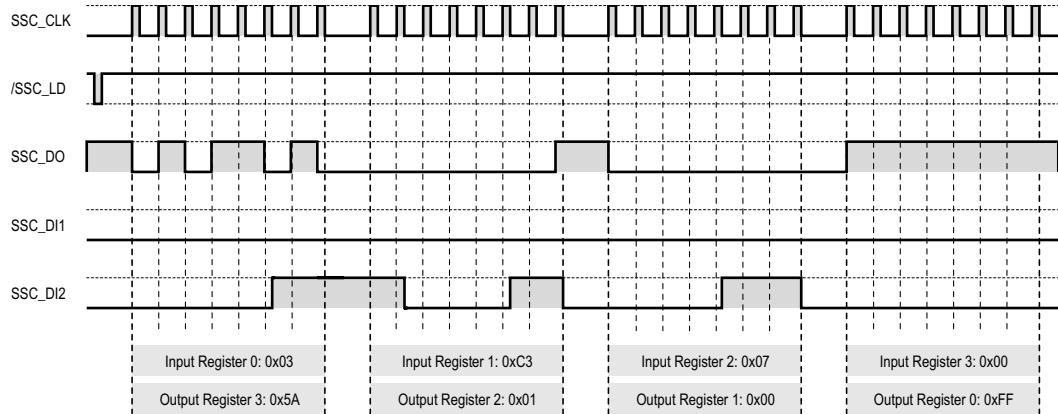


When done, the module starts executing refresh-cycles as described below.

## 5.6.2 Refresh Sequence (Normal Operation)

The refresh sequence is performed every  $5\text{ ms} \pm 2\text{ ms}$ . The Input Registers are read in consecutive order (0... 16), while the Output Registers are read in reverse order (16... 0). Note that the refresh sequence may be temporarily interrupted by other tasks, but will always finish after the interruption in order to keep data consistency.

The following example features 4 Input Registers and 4 Output Registers.



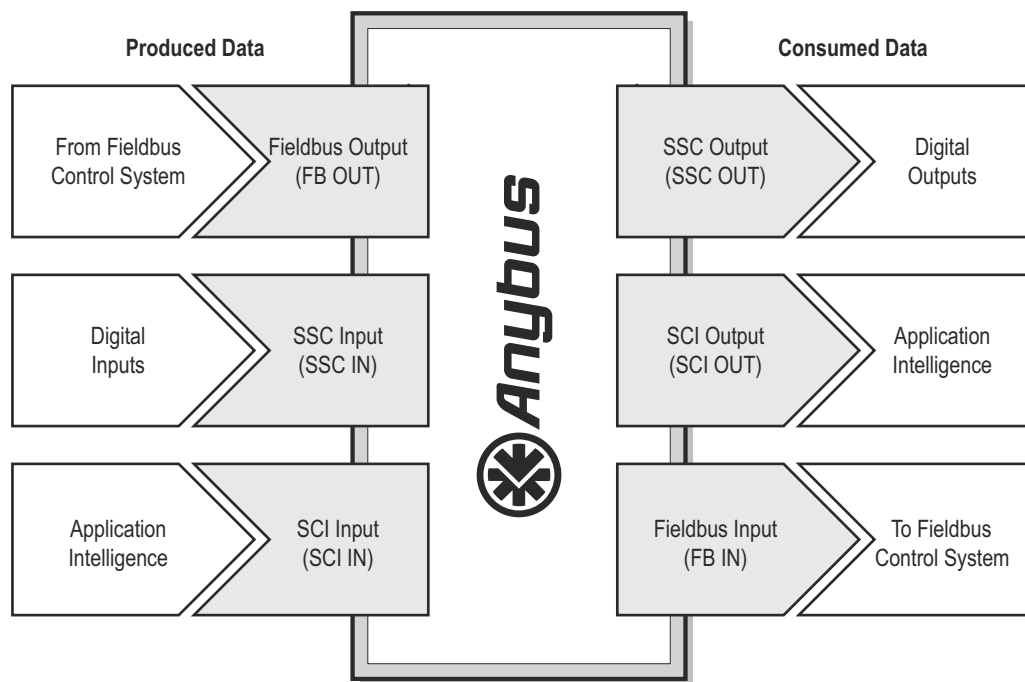
The refresh sequence is initiated by a /SSC\_LD pulse  $\geq 300\text{ ns}$ . This pulse loads the input shift registers with the input parallel data available. The load pulse is followed by a  $\geq 900\text{ ns}$  delay before the clock, SSC\_CLK, is activated. The clock cycle is  $\geq 660\text{ ns}$ , giving a maximum frequency of approximately 1,5 MHz, with the signal staying high/low for  $\geq 300\text{ ns}$ . SSC\_DI2 is sampled on the positive flank of the clock pulse and the next bit in the shift register will be shifted to the output of the register.

The output data from the Anybus-IC is shifted out on the negative flank of the clock. The output shift registers sample the serial data from the Anybus-IC on the positive flank of the clock. Once all registers have been updated, the clock stops. After a delay of  $\geq 900\text{ ns}$ , a new pulse on /SSC\_LD,  $\geq 300\text{ ns}$ , updates the output register circuits.

## 6. I/O Mapping

### 6.1 General Information

Each communication channel (i.e. SSC, fieldbus, SCI) uses two buffers. One which holds incoming (i.e. produced) data, and one which holds outgoing (i.e. consumed) data. Data produced on one channel can be mapped (i.e. copied automatically) to the consuming buffer of another channel. How this mapping shall be performed is specified by the I/O Parameters.



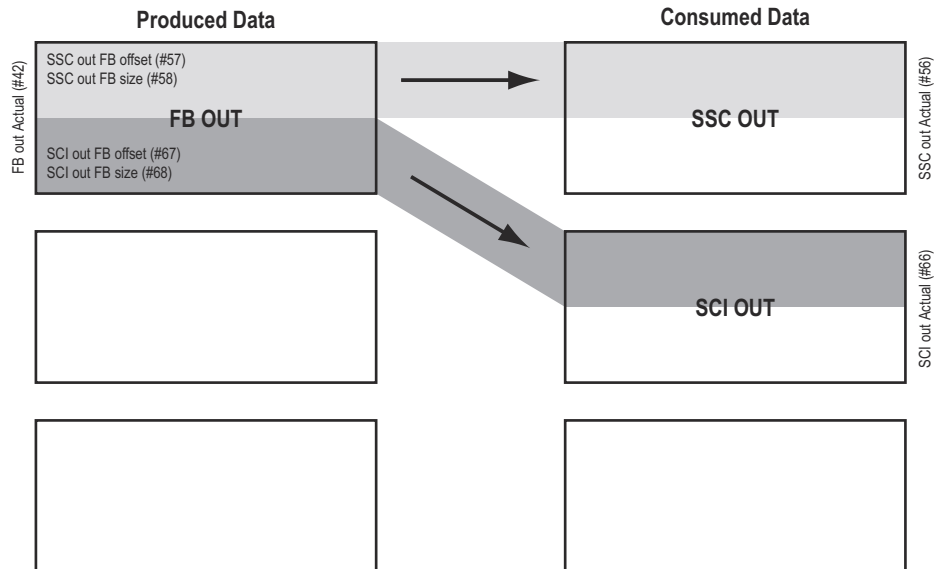
See also...

- “Fieldbus Mapping” on page 40 (mapping of data from FB OUT to SSC OUT and SCI OUT)
- “SSC Mapping” on page 41 (mapping of data from SSC IN to FB IN and SCI OUT)
- “SCI Mapping” on page 42 (mapping of data from SCI IN to FB IN and SSC OUT)
- “I/O Parameters” on page 73

**Note:** Data can only be mapped to another channel; it is not possible to map data produced on one channel to the very same channel’s consuming buffer (i.e. causing a “loop”).

## 6.2 Fieldbus Mapping

Data written by the fieldbus control system resides in the FB OUT buffer. This data may be mapped, i.e. copied automatically, to SCC OUT and/or SCI OUT.



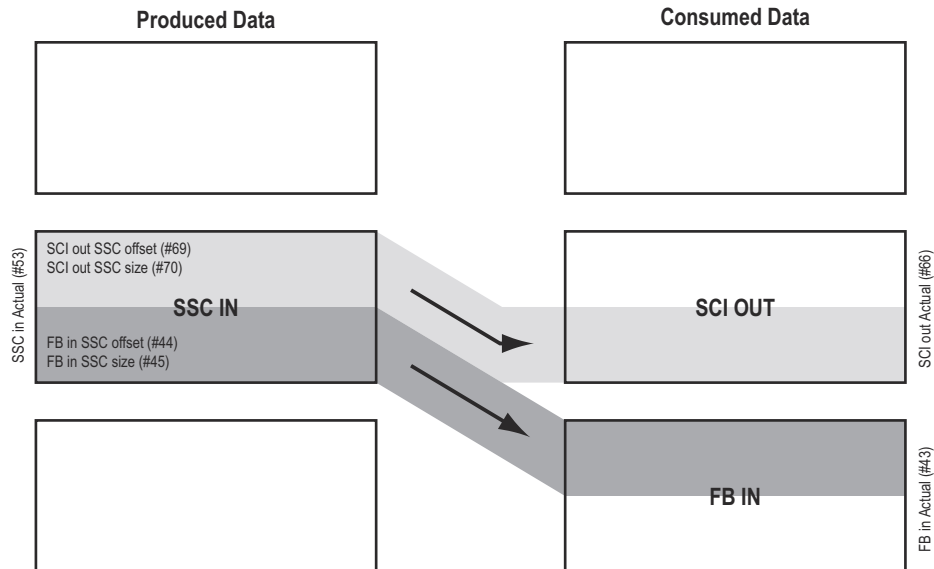
### Related Parameters

| Parameter               | Description  |
|-------------------------|--|
| FB Out Actual (#42)     | Specifies the total size of the FB OUT buffer after initialization. The value specified here is the maximum amount of data that can be mapped to other channels. |
| SSC Out FB Offset (#57) | Specifies the source offset (in the FB OUT buffer) for FB OUT to SCC OUT mapping.  |
| SSC Out FB Size (#58)   | Specifies the number of bytes to copy from FB OUT to SSC OUT.  |
| SCI Out FB Offset (#67) | Specifies the source offset (in the FB OUT buffer) for FB OUT to SCI OUT mapping.  |
| SCI Out FB Size (#68)   | Specifies the number of bytes to copy from FB OUT to SCI OUT.  |



## 6.3 SSC Mapping

Data read from the shift registers on the SSC channel resides in the SSC IN buffer. This data may be mapped, i.e. copied automatically, to FB IN and/or SCI OUT.

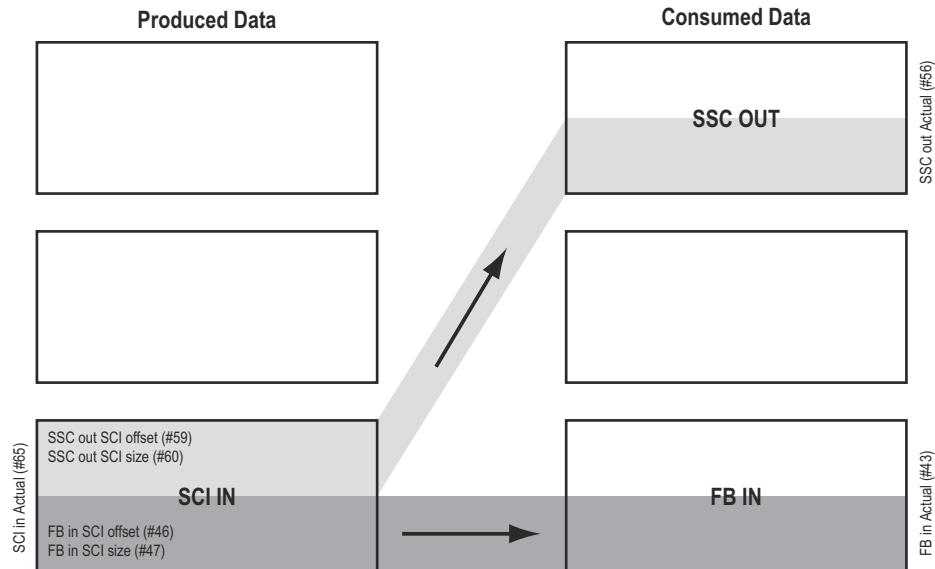


### Related Parameters

| Parameter                | Description  |
|--------------------------|--|
| SSC In Actual (#53)      | Specifies the total size of the SSC IN buffer after initialization. The value specified here is the maximum amount of data that can be mapped to other channels. |
| FB In SSC Offset (#44)   | Specifies the source offset (in the SSC IN buffer) for SSC IN to FB IN mapping.  |
| FB In SSC Size (#45)     | Specifies the number of bytes to copy from SSC IN to FB IN.  |
| SCI Out SSC Offset (#69) | Specifies the source offset (in the SSC IN buffer) for SSC IN to SCI OUT mapping.  |
| SCI Out SSC Size (#70)   | Specifies the number of bytes to copy from SSC IN to SCI OUT.  |

## 6.4 SCI Mapping

Data written by the application on the SCI channel resides in the SCI IN buffer. This data may be mapped, i.e. copied automatically, to FB IN and/or SSC OUT.



### Related Parameters

| Parameter                | Description  |
|--------------------------|--|
| SCI In Actual (#65)      | Specifies the total size of the SCI IN buffer after initialization. The value specified here is the maximum amount of data that can be mapped to other channels. |
| FB In SCI Offset (#46)   | Specifies the source offset (in the SCI IN buffer) for SCI IN to FB IN mapping.  |
| FB In SCI Size (#47)     | Specifies the number of bytes to copy from SCI IN to FB IN.  |
| SSC Out SCI Offset (#59) | Specifies the source offset (in the SCI IN buffer) for SCI IN to SSC OUT mapping.  |
| SSC Out SCI Size (#60)   | Specifies the number of bytes to copy from SCI IN to SSC OUT.  |

## 7. Initialization

### 7.1 General Information

The Anybus module features three modes of operation:

- **Normal Initialization**  
This mode is suitable for intelligent applications where the module shall be controlled by e.g. a microcontroller via the SCI channel.
- **Automatic Initialization (Standalone)**  
This mode is suitable for nonintelligent applications. The SCI channel will be disabled, but parameter settings etc. can still be accessed via the MIF interface.
- **Fieldbus Specific Initialization**  
This mode enables advanced fieldbus specific features when available. Consult each fieldbus appendix for further information.

**Note:** The examples in this chapter assume that all parameters are set to their default values prior to initialization. To reset the module to its factory default settings, set parameter #1 (“Module Mode”) to 0004h.

### 7.2 Normal Initialization

This mode is suitable for intelligent applications where a microcontroller or UART is connected to the SCI channel.

#### Initialization Sequence

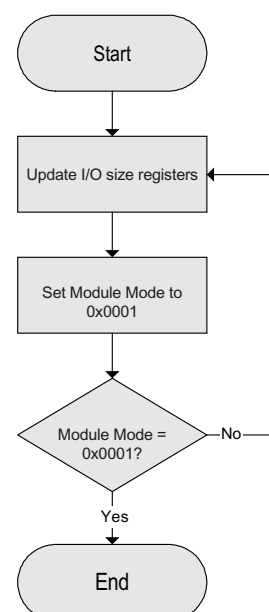
The following steps initialize the module via the SCI channel.

1. Start-up
2. If necessary, perform the automatic baud rate detection sequence for the SCI channel, see “Baud Rate” on page 27
3. Configure the I/O data sizes using the I/O parameters
4. Set parameter #1 (“Module Mode”) to 0x0001 (Normal Mode)
5. Check if initialization was successful
6. End of initialization

**Note:** If required, the procedure above can also be performed using the MIF interface. However, since this interface is designed for debugging and configuration purposes, this initialization method should only be used when evaluating the various functions of the module.

See also...

- “Initialization Examples, Normal Initialization” on page 45



## 7.3 Automatic Initialization (Standalone)

Connecting SCI\_DE [AUTO] (pin 29) to ground (GND) will cause the module to initialize itself automatically to run “standalone”. This mode is intended for nonintelligent applications such as valve terminals and modular I/O devices.

Automatic initialization affects the implementation as follows:

- The SCI channel will be completely disabled, and SCI related parameters will either be set to zero or ignored, depending on context. Parameters can however still be set via the MIF interface.
- SSC- and fieldbus related I/O parameters will be set automatically based on the shift register configuration. All SSC data will be mapped to the fieldbus and vice versa.
- The settings in parameter #8 (“Configuration Bits”) will be set to reflect standalone operation.

**Note:** If parameter #8 “Configuration Bits” is set to a value different than zero (0000h), the automatic initialization may not work as expected (e.g. if the SSCI bit is 1, the SSC Input data size will be taken from parameter #51 “SSC In Config” instead of the automatically detected size). To reset the module to its factory default settings, set parameter #1 (“Module Mode”) to 0004h.

See also...

- “Application Connector” on page 14 (pin 29)
- “MIF Interface” on page 21
- “SSC Channel” on page 31
- “Initialization Examples, Automatic Initialization” on page 49
- “General Parameters” on page 52 ( “Configuration Bits (#8)” on page 57)
- “I/O Parameters” on page 73

## 7.4 Fieldbus Specific Initialization

For advanced implementations, certain Anybus-IC versions feature special initialization modes which provide support for advanced fieldbus specific functions.

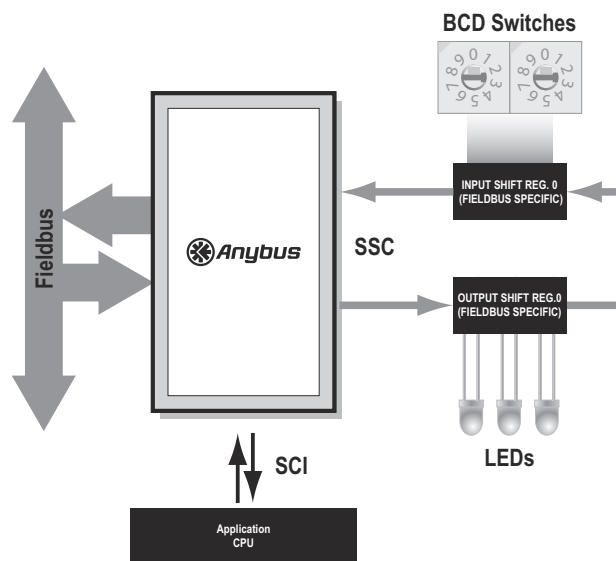
For more information, consult each separate fieldbus appendix.

## 7.5 Initialization Examples, Normal Initialization

### 7.5.1 Switches and LEDs on SSC, Data Exchange via SCI

Properties:

- SCI channel is interfaced to a microcontroller
- Automatic baud rate detection is used (SCI)
- All I/O is mapped between the SCI and Fieldbus channels  
(In this case 8 bytes in each direction)
- LEDs on the Fieldbus Specific Output Register (0)
- BCD-coded switches on the Fieldbus Specific Input Register



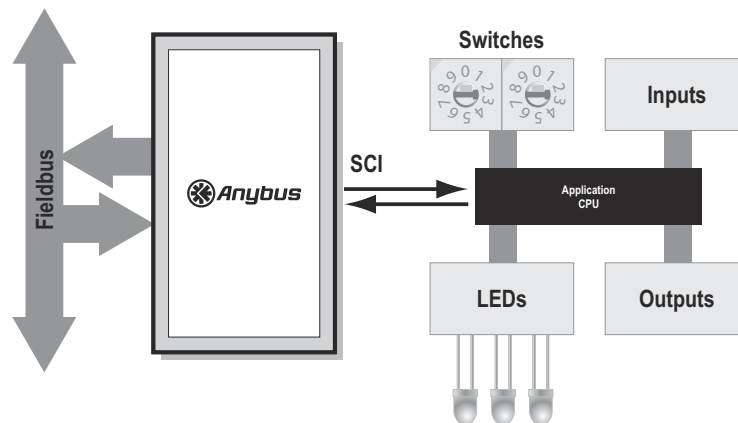
#### Parameter Values

| Parameter               | Setting          |
|-------------------------|------------------|
| Configuration Bits (#8) | SSCI=1, SSC0=1   |
| Switch Coding (#9)      | 0x00             |
| SCI Rate Config (#14)   | 0x00 (default)   |
| FB Out Config (#41)     | 0x0008           |
| FB In SSC Size (#45)    | 0x0000 (default) |
| FB In SCI Offset (#46)  | 0x0000 (default) |
| FB In SCI Size (#47)    | 0x0008           |
| SSC In Config (#51)     | 0x0000 (default) |
| SSC Out Config (#54)    | 0x0000 (default) |
| SCI In Config (#64)     | 0x0008           |
| SCI Out FB Offset (#67) | 0x0000 (default) |
| SCI Out FB Size (#68)   | 0x0008           |
| SCI Out SSC Size (#70)  | 0x0000 (default) |

## 7.5.2 Switches, LEDs and Data Exchange via SCI

Properties:

- SCI channel is interfaced to a microcontroller
- All I/O is mapped between the SCI and Fieldbus channels  
(In this case 8 bytes in each direction)
- LEDs and outputs handled by application
- Switches and inputs handled by application



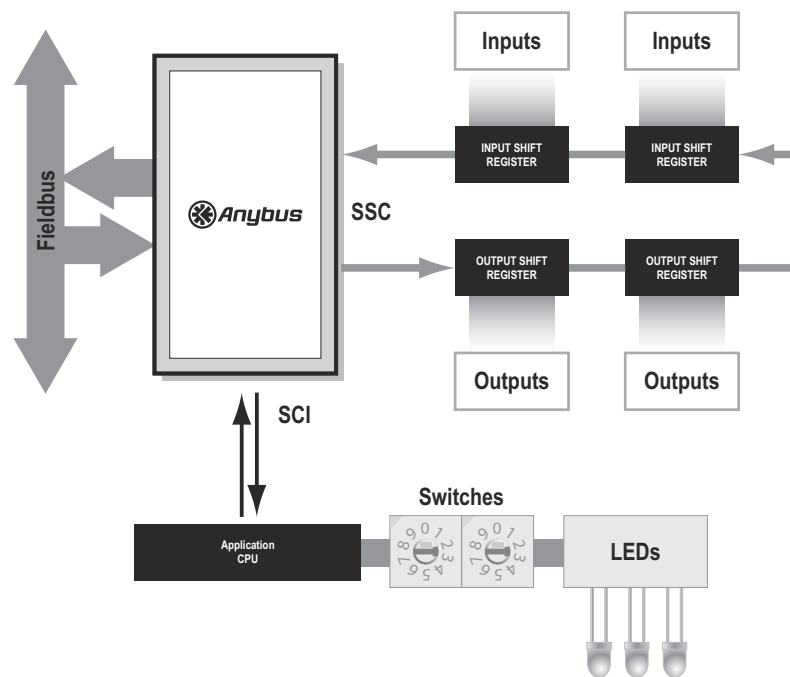
### Parameter Values

| Parameter               | Setting                            |
|-------------------------|------------------------------------|
| Configuration Bits (#8) | FBNP=1, FB LP =1, SSC I=1, SSC O=1 |
| Switch Coding (#9)      | 0x00                               |
| SCI Rate Config (#14)   | 0x00 (default)                     |
| FB Out Config (#41)     | 0x0008                             |
| FB In SSC Size (#45)    | 0x0000 (default)                   |
| FB In SCI Offset (#46)  | 0x0000 (default)                   |
| FB In SCI Size (#47)    | 0x0008                             |
| SSC In Config (#51)     | 0x0000 (default)                   |
| SSC Out Config (#54)    | 0x0000 (default)                   |
| SCI In Config (#64)     | 0x0008                             |
| SCI Out FB Offset (#67) | 0x0000 (default)                   |
| SCI Out FB Size (#68)   | 0x0008                             |
| SCI Out SSC Size (#70)  | 0x0000 (default)                   |

### 7.5.3 Switches and LEDs on SCI, Data Exchange via SSC and SCI

Properties:

- SCI channel is interfaced to a microcontroller
- LEDs handled by application
- Switches handled by application
- 2 bytes of SSC data in each direction
- 6 bytes of SCI data in each direction
- Fieldbus data mapped to both SCI and SSC



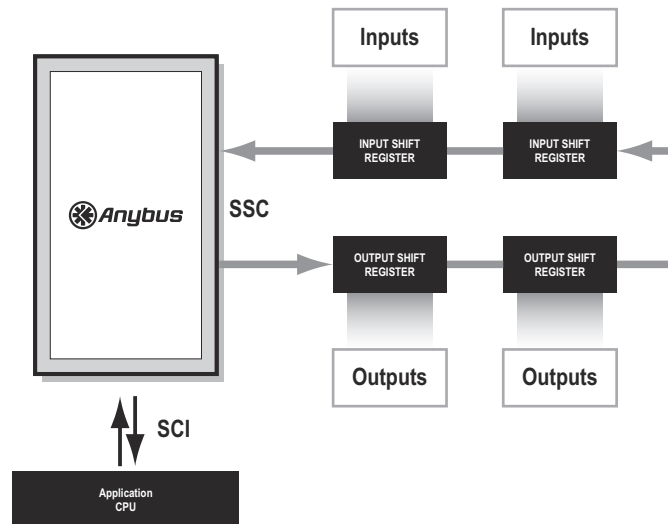
#### Parameter Values

| Parameter               | Setting                         |
|-------------------------|---------------------------------|
| Configuration Bits (#8) | FBNP=1, FBLP =1, SSCI=1, SSCO=1 |
| SCI Rate Config (#14)   | 0x00                            |
| SCI In Config (#64)     | 0x0006                          |
| SCI Out FB Offset (#67) | 0x0002                          |
| SCI Out FB Size (#68)   | 0x0006                          |
| FB Out Config (#41)     | 0x0008                          |
| FB In SSC Size (#45)    | 0x0002                          |
| FB In SCI Offset (#46)  | 0x0000                          |
| FB In SCI Size (#47)    | 0x0006                          |
| FB In SSC Offset (#44)  | 0x0000                          |
| SSC In Config (#51)     | 0x0002                          |
| SSC Out Config (#54)    | 0x0002                          |
| SSC Out FB Offset (#57) | 0x0000                          |
| SSC Out FB Size (#58)   | 0x0002                          |

### 7.5.4 SCI and SSC Used for Data Exchange (No Fieldbus I/O)

Properties:

- SCI channel is interfaced to a microcontroller
- 2 bytes of SSC data in each direction
- 2 bytes of SCI data in each direction
- All data mapped between SSC and SCI



#### Parameter Values

| Parameter               | Setting                         |
|-------------------------|---------------------------------|
| Configuration Bits (#8) | FBNP=1, FBLP =1, SSCI=1, SSCO=1 |
| SCI Rate Config (#14)   | 0x00                            |
| SCI In Config (#64)     | 0x0002                          |
| FB Out Config (#41)     | 0x0000                          |
| FB In SSC Size (#45)    | 0x0000                          |
| FB In SCI Offset (#46)  | 0x0000                          |
| FB In SCI Size (#47)    | 0x0000                          |
| FB In SSC Offset (#44)  | 0x0000                          |
| SSC In Config (#51)     | 0x0002                          |
| SSC Out Config (#54)    | 0x0002                          |
| SSC Out SCI Size (#60)  | 0x0002                          |
| SCI Out SSC Size (#70)  | 0x0002                          |

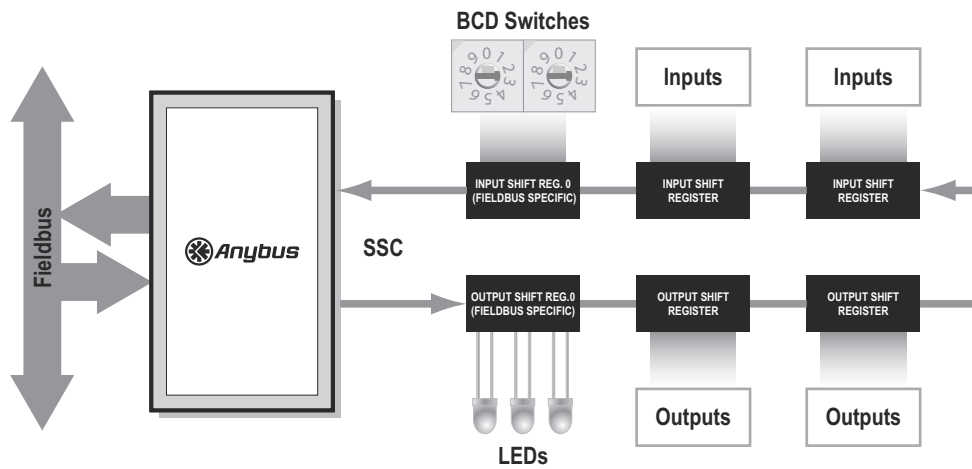


## 7.6 Initialization Examples, Automatic Initialization

### 7.6.1 Switches and LEDs on SSC

Properties:

- Standalone operation (Automatic initialization)
- All data is mapped between Fieldbus and SSC
- LEDs on the Fieldbus Specific Output Register (0)
- BCD-coded switches on the Fieldbus Specific Input Register



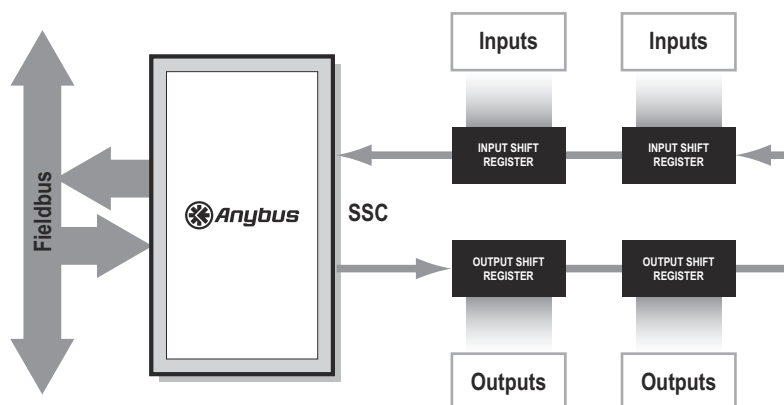
#### Parameter Values

| Parameter               | Setting       |
|-------------------------|---------------|
| Configuration Bits (#8) | 00h (default) |
| Switch Coding (#9)      | 00h           |

## 7.6.2 Preconfigured Node Address, no LEDs

Properties:

- Standalone operation (Automatic initialization)
- All data is mapped between Fieldbus and SSC
- No LEDs or switches



### Parameter Values

| Parameter               | Setting  |
|-------------------------|--|
| Configuration Bits (#8) | FBNP=1, FBLP =1  |
| (fieldbus specific)     | Node address specified via MIF-interface (fieldbus specific parameter) |

## 8. Parameters

### 8.1 General Information

From the host application side, all settings and data in the module are represented through entities called 'Parameters'. Parameters can be accessed by the user through the MIF interface, or by the application via the SCI channel.

The parameters are grouped into three categories based on their usage as follows:

- **General Anybus-IC Parameters**  
Used for configuration and status information, present on all versions of the Anybus-IC.
- **I/O Parameters**  
Specifies data sizes and I/O mapping, present on all versions of the Anybus-IC.
- **Fieldbus Specific Parameters**  
Entirely fieldbus specific; consult each separate fieldbus appendix for further information.

**Note 1:** When accessing parameters via the SCI channel (e.g. using Modbus), byte-sized parameter values are placed in the least significant byte of the word.

**Note 2:** In this document, register numbers are specified using the protocol convention (base 0), i.e. there is a 1:1 correlation between the register number specified in this document and the actual register value in the message frame.

## 8.2 General Parameters

These parameters are used to configure the basic settings of the module.

| #  | Modbus Address   | Name                | Size     | Default                         | Access |
|----|------------------|---------------------|----------|---------------------------------|--------|
| 1  | 0x5001           | Module Mode         | 2 bytes  | -                               | R/W    |
| 2  | 0x5002           | Module Status       | 2 bytes  | -                               | R      |
| 3  | 0x5003           | Module Type         | 2 bytes  | -                               | R      |
| 4  | 0x5004           | Fieldbus Type       | 2 bytes  | -                               | R      |
| 7  | 0x5007           | LED State           | 2 bytes  | -                               | R      |
| 8  | 0x5008           | Config Bits         | 2 bytes  | 0x0000                          | R/W    |
| 9  | 0x5009           | Switch Coding       | 1 byte   | Fieldbus dependent <sup>a</sup> | R/W    |
| 10 | 0x500A           | Offline Action      | 1 byte   | 0x00                            | R/W    |
| 11 | 0x500B           | Idle Action         | 1 byte   | 0x00                            | R/W    |
| 12 | 0x500C           | Interrupt Config    | 2 bytes  | 0x0001                          | R/W    |
| 13 | 0x500D           | Interrupt Cause     | 2 bytes  | -                               | R      |
| 14 | 0x500E           | SCI Rate Config     | 1 byte   | 0x00                            | R/W    |
| 15 | 0x500F           | SCI Rate Actual     | 1 byte   | -                               | R      |
| 16 | 0x5010           | SCI Settings Config | 1 byte   | 0x00                            | R/W    |
| 17 | 0x5011           | SCI Settings Actual | 1 byte   | -                               | R      |
| 18 | 0x5012           | MIF Rate Config     | 1 byte   | 0x04                            | R/W    |
| 19 | 0x5013           | MIF Rate Actual     | 1 byte   | -                               | R      |
| 20 | 0x5014           | MIF Settings Config | 1 byte   | 0x00                            | R/W    |
| 21 | 0x5015           | MIF Settings Actual | 1 byte   | -                               | R      |
| 22 | 0x5016           | Modbus RTU Address  | 1 byte   | 0x01                            | R/W    |
| 23 | 0x5017           | Modbus CRC Disable  | 1 byte   | 0x00                            | R/W    |
| 27 | 0x501B... 0x5032 | FB Fault Values     | 48 bytes | 0x00                            | R/W    |

a. See separate fieldbus appendix

**Note:** In this document, register numbers are specified using the protocol convention (base 0), i.e. register 0042h equals 0042h in the message frame.

## 8.2.1 Module Mode (#1)

This parameter is used to determine the current operating mode of the module.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 1               |
| <b>Modbus Address</b>   | 0x5001          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0005 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R/W             |

### Valid Settings

- **0x0000 - Start up Mode**

This is the initial value of the parameter in all cases after power up except when the module is automatically initialized.

- **0x0001 - Normal Operation Mode**

When all parameters are updated with correct settings, the normal initialization is triggered by writing this value. If the module is automatically initialized, the parameter automatically gets this value.

**Note:** If automatic baud rate detection on the SCI channel is enabled, it is not possible to initialize the module in normal operation mode via the MIF interface until the baud rate has been detected.

- **0x0002 - Fieldbus Specific Init**

If the Anybus module supports fieldbus specific initialization (fieldbus dependent) this value starts the initialization. See fieldbus appendix for more information.

- **0x0003 - Reset Module**

This value makes a reset of the module. The module needs to be reinitialized to start communicating. Note that this is not the same as mode 0x0004, see below.

- **0x0004 - Set Default**

All configurable parameters will be set to their factory default values. Note that password protected parameters will not be affected unless the password is entered before sending "Set default". Note that this is not the same as mode 0x0003, see above.

- **0x0005 - Self Test**

By writing this value to the parameter, an internal self test is initiated on the Anybus module. For more information, see "Self Test Sequence" on page 86 .

## 8.2.2 Module Status (#2)

This parameter holds information about the current status of the module. This parameter is also used to deliver the result of the self test sequence (See “Self Test Sequence” on page 86 )

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 2         |
| <b>Modbus Address</b>   | 0x5002    |
| <b>Default value</b>    | -         |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 2 bytes   |
| <b>Stored in NV RAM</b> | No        |
| <b>Access</b>           | R         |

### Bit Layout

| b15 | b14 | b13 | b12 | b11 | b10  | b9   | b8  | b7 | b6 | b5 | b4 | b3 | b2 | b1  | b0  |
|-----|-----|-----|-----|-----|------|------|-----|----|----|----|----|----|----|-----|-----|
| RCF | FCF | ECF | -   | -   | SCIC | SSCC | FBC | -  | -  | -  | -  | -  | -  | SCI | SSC |

- **SSC**
  - 1: SSC channel running - Shift registers are detected on the SSC channel and the autoinitialized sizes are present in parameter #52 “SSC In Auto” and parameter #55 “SSC Out Auto”.
  - 0: SSC channel stopped/not used - No shift registers are detected on the SSC channel.
- **SCI**
  - 1: SCI channel running
  - 0: SCI channel stopped/not used - Baud rate configuration error / baud rate not detected
- **FBC**
  - 1: FB I/O Configuration failure - I/O data mapping configuration error
  - 0: FB I/O Configuration OK
- **SSCC**
  - 1: SSC I/O Configuration failure - I/O data mapping configuration error
  - 0: SSC I/O Configuration OK
- **SCIC**
  - 1: SCI I/O Configuration failure - I/O data mapping configuration error
  - 0: SCI I/O Configuration OK
- **ECF<sup>1</sup>**
  - 1: EEPROM memory check failed
  - 0: EEPROM memory check OK
- **FCF<sup>1</sup>**
  - 1: FLASH memory check failed
  - 0: FLASH memory check OK
- **RCF<sup>1</sup>**
  - 1: RAM check failed
  - 0: RAM check OK

1. This bit is only set after the self test is performed.

### 8.2.3 Module Type (#3)

This parameter identifies which type of module that is used.

|                         |                  |
|-------------------------|------------------|
| <b>Parameter number</b> | 3                |
| <b>Modbus Address</b>   | 0x5003           |
| <b>Default value</b>    | -                |
| <b>Range</b>            | 0x0000h - 0xFFFF |
| <b>Size</b>             | 2 bytes          |
| <b>Stored in NV RAM</b> | Yes              |
| <b>Access</b>           | R                |

#### Values

- 0x0301 - Standard Anybus-IC

### 8.2.4 Fieldbus Type (#4)

This parameter identifies the fieldbus interface.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 4               |
| <b>Modbus Address</b>   | 0x5004          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0xFFFF |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R               |

#### Values

- 0x0001 - PROFIBUS DP
- 0x0020 - CANopen
- 0x0025 - DeviceNet
- 0x0083 - EtherNet/IP
- 0x0084 - PROFINET

## 8.2.5 LED State (#7)

The state of the LED register can be read using this parameter.

**Note:** This parameter is updated with the LED state, even if the LED register is not used.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 7         |
| <b>Modbus Address</b>   | 0x5007    |
| <b>Default value</b>    | -         |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 2 bytes   |
| <b>Stored in NV RAM</b> | No        |
| <b>Access</b>           | R         |

### Bit Layout

|       |       |       |       |       |       |       |       |    |    |    |    |    |    |    |    |
|-------|-------|-------|-------|-------|-------|-------|-------|----|----|----|----|----|----|----|----|
| b15   | b14   | b13   | b12   | b11   | b10   | b9    | b8    | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| LED 8 | LED 7 | LED 6 | LED 5 | LED 4 | LED 3 | LED 2 | LED 1 |    |    |    |    |    |    |    |    |

Description of LED 1 behavior. All the other LEDs behave according to the same bit pattern.

| LED 1 (bit 1) | LED 1 (bit 0) | Description   |
|---------------|---------------|---|
| 0             | 0             | LED 1 is turned off   |
| 0             | 1             | LED 1 is turned on  |
| 1             | 0             | LED 1 is flashing 1Hz   |
| 1             | 1             | Reserved (LED test - Fieldbus specific - See fieldbus appendix) |



## 8.2.6 Configuration Bits (#8)

This parameter determines which values that will be valid for different initialization parameters.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 8         |
| <b>Modbus Address</b>   | 0x5008    |
| <b>Default value</b>    | 0x0000    |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 2 bytes   |
| <b>Stored in NV RAM</b> | Yes       |
| <b>Access</b>           | R/W       |

### Bit Layout

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7   | b6   | b5 | b4 | b3   | b2   | b1   | b0   |
|-----|-----|-----|-----|-----|-----|----|----|------|------|----|----|------|------|------|------|
| -   | -   | -   | -   | -   | -   | -  | -  | FBBR | FBNA | BR | NA | FBLP | FBNP | SSCO | SSCI |

- **SSCI**
  - 1:** Configured SSC input size is used (see “SSC In Config (#51)” on page 77 ).
  - 0:** Automatically initialized SSC input size is used. See parameter #52 “SSC In Auto”
- **SSCO**
  - 1:** Configured SSC output size is used (see “SSC Out Config (#54)” on page 78 ).
  - 0:** Automatically initialized SSC output size is used. See parameter #55 “SSC Out Auto”
- **FBNP**

(If no input shift registers are connected, this bit is automatically set to 1).

  - 1:** Fieldbus specific input register is not present on the SSC channel
  - 0:** Fieldbus specific input register is present on the SSC channel
- **FBLP**

(If no output shift registers are connected, this bit is automatically set to 1).

  - 1:** Fieldbus specific output register is not present on the SSC channel
  - 0:** Fieldbus specific output register is present on the SSC channel
- **NA<sup>1</sup>**

(If no input shift registers are connected, this bit is automatically set to 1).

  - 1:** Node address determined by fieldbus specific node address parameters<sup>2</sup>
  - 0:** Node address determined by the fieldbus specific input register.

1. If the FBNA bit is set, the value of the NA bit will be ignored.  
 2. Consult each separate fieldbus appendix for further information.

- **BR**

(If no input shift registers are connected, this bit is automatically set to 1).

**1:** Fieldbus baud rate is set using fieldbus specific baud rate parameter<sup>1</sup>

**0:** Fieldbus baud rate is set via fieldbus, or via switches on the fieldbus specific input register.

- **FBNA**

This bit determines the behavior on systems where the node address can be received from the fieldbus. On fieldbus systems that does not feature this functionality, this bit has no function.

**1:** Node address is received from the fieldbus. The value of the NA bit will be ignored.

**0:** Node address source is determined by the NA bit, see above.

- **FBBR**

This bit is fieldbus specific. Consult each separate fieldbus appendix for further information.

**0:** Fieldbus specific features disabled.

**1:** Fieldbus specific features enabled.

---

1. Consult each separate fieldbus appendix for further information.

## 8.2.7 Switch Coding (#9)

If the fieldbus specific input register is enabled, this parameter determines how the value of the switches should be interpreted by the module.

|                         |   |
|-------------------------|---|
| <b>Parameter number</b> | 9   |
| <b>Modbus Address</b>   | 0x5009  |
| <b>Default value</b>    | Fieldbus dependent, see separate fieldbus appendix. |
| <b>Range</b>            | 0x00 - 0x01   |
| <b>Size</b>             | 1 byte  |
| <b>Stored in NV RAM</b> | Yes   |
| <b>Access</b>           | R/W   |

### Values

- 0x00 - BCD-coded switches**

BCD-coded switches encodes each decimal digit using four bits. Two switches are used to specify the decimal value in the range of 0... 99. The table below lists valid switch values and their corresponding decimal value. Values not listed are considered invalid.

| # | Switch Pattern |
|---|----------------|
| 0 | 0000           |
| 1 | 0001           |
| 2 | 0010           |
| 3 | 0011           |
| 4 | 0100           |
| 5 | 0101           |
| 6 | 0110           |
| 7 | 0111           |
| 8 | 1000           |
| 9 | 1001           |

#### *Example*

63 in decimal format becomes 0110 0011 in BCD-coded form.

- 0x01 - Binary switches**

When using binary switches, the correlation is 1:1 as shown in the table below.

| #   | Switch Pattern |
|-----|----------------|
| 0   | 0000 0000      |
| 1   | 0000 0001      |
| 2   | 0000 0010      |
| ... | ...            |
| 254 | 1111 1110      |
| 255 | 1111 1111      |

#### *Example*

63 in decimal format becomes 0011 1111 in binary form.

### 8.2.8 Offline Action Config (#10)

When the fieldbus goes from online to offline or from idle to offline, the fieldbus outputs can be configured to behave in different ways.

**Note:** The module must have been online once for the offline action to take effect.

|                  |             |
|------------------|-------------|
| Parameter number | 10          |
| Modbus Address   | 0x500A      |
| Default value    | 0x00        |
| Range            | 0x00 - 0x02 |
| Size             | 1 byte      |
| Stored in NV RAM | Yes         |
| Access           | R/W         |

#### Values

- **0x00 - Clear**  
Fieldbus outputs are cleared when the fieldbus goes offline.
- **0x01 - Freeze**  
Fieldbus outputs freeze in the states they are in when the fieldbus goes offline.
- **0x02 - Fault values**  
Fault values configured in parameter #27 “FB Fault Values” are copied to the fieldbus outputs when the fieldbus goes offline.

### 8.2.9 Idle Action Config (#11)

When the fieldbus goes from online to idle or from offline to idle, the fieldbus outputs can be configured to behave in different ways.

**Note:** The Anybus module must have been online once for the idle action to take effect.

|                  |             |
|------------------|-------------|
| Parameter number | 11          |
| Modbus Address   | 0x500B      |
| Default value    | 0x00        |
| Range            | 0x00 - 0x02 |
| Size             | 1 byte      |
| Stored in NV RAM | Yes         |
| Access           | R/W         |

#### Values

- **0x00 - Clear**  
Fieldbus outputs are cleared when the fieldbus goes to idle.
- **0x01 - Freeze**  
Fieldbus outputs freeze in the states they are in when the fieldbus goes to idle.
- **0x02 - Fault values**  
Fault values configured in parameter #27 “FB Fault Values” are copied to the fieldbus outputs when the fieldbus goes to idle.

### 8.2.10 Interrupt Config (#12)

This parameter defines events that are allowed to trigger an interrupt. See also parameter #13 “Interrupt Cause”.

|                  |           |
|------------------|-----------|
| Parameter number | 12        |
| Modbus Address   | 0x500C    |
| Default value    | 0x0001    |
| Range            | Bit field |
| Size             | 2 bytes   |
| Stored in NV RAM | Yes       |
| Access           | R/W       |

#### Bit Layout

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6   | b5  | b4  | b3   | b2    | b1   | b0    |
|-----|-----|-----|-----|-----|-----|----|----|----|------|-----|-----|------|-------|------|-------|
| -   | -   | -   | -   | -   | -   | -  | -  | -  | IDLE | RES | DEF | ACYC | FBOFF | FBON | START |

- **START<sup>1</sup>**
  - 1: An interrupt will be generated when the module has started from power on and is ready to communicate.
  - 0: This event will not cause an interrupt.
- **FBON**
  - 1: An interrupt will be generated when the fieldbus goes from offline to online.
  - 0: This event will not cause an interrupt.
- **FBOFF**
  - 1: An interrupt will be generated when the fieldbus goes from online to offline.
  - 0: This event will not cause an interrupt.
- **ACYC<sup>2</sup>**
  - 1: An interrupt will be generated when new acyclic data is received from the fieldbus master.
  - 0: This event will not cause an interrupt.
- **DEF<sup>2</sup>**
  - 1: An interrupt will be generated when “set default” is received from the fieldbus master.
  - 0: This event will not cause an interrupt.
- **RES<sup>2</sup>**
  - 1: An interrupt will be generated when “reset” is received from the fieldbus master.
  - 0: This event will not cause an interrupt.
- **IDLE<sup>2</sup>**
  - 1: An interrupt will be generated when the fieldbus goes from online to idle or from offline to idle.
  - 0: This event will not cause an interrupt.

1. If automatic baud rate detection is used, the interrupt is not generated until the correct baud rate is detected.  
 2. This bit is fieldbus dependent, i.e. it may not be available on all versions of the Anybus-IC.

### 8.2.11 Interrupt Cause (#13)

This parameter indicates the event that has caused an interrupt. The events that shall generate an interrupt are configured in parameter #12 “Interrupt Config”. The parameter is automatically cleared by the Anybus module when read by the application. See “Interrupt (/INT) & Bootloader Enable (BLE)” on page 85 for more information about the interrupt function.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 13        |
| <b>Modbus Address</b>   | 0x500D    |
| <b>Default value</b>    | -         |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 2 bytes   |
| <b>Stored in NV RAM</b> | No        |
| <b>Access</b>           | R         |

#### Bit Layout

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6   | b5  | b4  | b3   | b2    | b1   | b0    |
|-----|-----|-----|-----|-----|-----|----|----|----|------|-----|-----|------|-------|------|-------|
| -   | -   | -   | -   | -   | -   | -  | -  | -  | IDLE | RES | DEF | ACYC | FBOFF | FBON | START |

- **START**
  - 1: The Anybus module has started and is ready to communicate.
  - 0: This event has not caused an interrupt.
- **FBON**
  - 1: A transition from offline to online on the fieldbus has occurred.
  - 0: This event has not caused an interrupt.
- **FBOFF**
  - 1: A transition from online to offline on the fieldbus has occurred.
  - 0: This event has not caused an interrupt.
- **ACYC**
  - 1: New acyclic data is received from the fieldbus master.
  - 0: This event has not caused an interrupt.
- **DEF (Fieldbus Dependent)**
  - 1: “Set default” has been received from the fieldbus master
  - 0: This event has not caused an interrupt.
- **RES (Fieldbus Dependent)**
  - 1: “Reset” has been received from the fieldbus master
  - 0: This event has not caused an interrupt.
- **IDLE (Fieldbus Dependent)**
  - 1: The fieldbus has gone from online to idle or from offline to idle.
  - 0: This event has not caused an interrupt.

### 8.2.12 SCI Rate Config (#14)

This parameter is used to configure the baud rate of the SCI channel. A reset/power cycle of the module is necessary in order for any changes to have effect.

|                         |                         |
|-------------------------|-------------------------|
| <b>Parameter number</b> | 14                      |
| <b>Modbus Address</b>   | 0x500E                  |
| <b>Default value</b>    | 0x00                    |
| <b>Range</b>            | 0x00h - 0x0A (see note) |
| <b>Size</b>             | 1 byte                  |
| <b>Stored in NV RAM</b> | Yes                     |
| <b>Access</b>           | R/W                     |

- Values**

| <b>Value</b> | <b>Baud Rate</b>                        |
|--------------|---|
| 0x00         | Automatic baud rate detection (default) |
| 0x01         | 4.8 kbit/s                              |
| 0x02         | 9.6 kbit/s                              |
| 0x03         | 19.2 kbit/s                             |
| 0x04         | 38.4 kbit/s                             |
| 0x05         | 57.6 kbit/s                             |
| 0x06         | reserved                                |
| 0x07         | 156.25 kbit/s <sup>a</sup>              |
| 0x08         | 208.33 kbit/s <sup>a</sup>              |
| 0x09         | 312.5 kbit/s <sup>a</sup>               |
| 0x0A         | 625 kbit/s <sup>a</sup>                 |

a. These baud rates are only supported by Anybus-IC DeviceNet, PROFIBUS and CANopen modules. These baud rates are not supported by automatic baud rate detection.



### 8.2.13 SCI Rate Actual (#15)

This parameter returns is the actual baud rate of the SCI channel.

|                         |                        |
|-------------------------|------------------------|
| <b>Parameter number</b> | 15                     |
| <b>Modbus Address</b>   | 0x500F                 |
| <b>Default value</b>    | -                      |
| <b>Range</b>            | 0x00 - 0x05 (see note) |
| <b>Size</b>             | 1 byte                 |
| <b>Stored in NV RAM</b> | No                     |
| <b>Access</b>           | R                      |

- **Values**

| <b>Value</b> | <b>Baud Rate</b>           |
|--------------|----------------------------|
| 0x00         | Baud rate not set          |
| 0x01         | 4.8 kbit/s                 |
| 0x02         | 9.6 kbit/s                 |
| 0x03         | 19.2 kbit/s                |
| 0x04         | 38.4 kbit/s                |
| 0x05         | 57.6 kbit/s                |
| 0x06         | reserved                   |
| 0x07         | 156.25 kbit/s <sup>a</sup> |
| 0x08         | 208.33 kbit/s <sup>a</sup> |
| 0x09         | 312.5 kbit/s <sup>a</sup>  |
| 0x0A         | 625 kbit/s <sup>a</sup>    |

a. These baud rates are only supported by Anybus-IC DeviceNet, PROFIBUS and CANopen modules. These baud rates are not supported by automatic baud rate detection.

### 8.2.14 SCI Settings Config (#16)

This parameter is used to configure the port settings of the SCI channel. A reset/power cycle of the module is necessary in order for any changes to have effect.

This parameter has no effect when automatic baud rate detection is enabled.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 16        |
| <b>Modbus Address</b>   | 0x5010    |
| <b>Default value</b>    | 0x00      |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 1 byte    |
| <b>Stored in NV RAM</b> | Yes       |
| <b>Access</b>           | R/W       |

#### Bit Layout

| b7 | b6 | b5 | b4 | b3 | b2 | b1   | b0   |
|----|----|----|----|----|----|------|------|
| -  | -  | -  | -  | -  | -  | PAR1 | PAR2 |

- **PAR2**
  - 1: Enable Parity
  - 0: Disable Parity
- **PAR1**

(This bit has no effect if parity is disabled, see above)

  - 1: Odd Parity
  - 0: Even Parity

### 8.2.15 SCI Settings Actual (#17)

This parameter returns the actual port settings for the SCI channel. If automatic baud rate detection is enabled, the detected port settings are present in this parameter.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 17        |
| <b>Modbus Address</b>   | 0x5011    |
| <b>Default value</b>    | -         |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 1 byte    |
| <b>Stored in NV RAM</b> | No        |
| <b>Access</b>           | R         |

#### Bit Layout

| b7 | b6 | b5 | b4 | b3 | b2 | b1   | b0   |
|----|----|----|----|----|----|------|------|
| -  | -  | -  | -  | -  | -  | PAR1 | PAR2 |

- **PAR2**
  - 1: Enable Parity
  - 0: Disable Parity
- **PAR1**

(This bit has no effect if parity is disabled, see above)

  - 1: Odd Parity
  - 0: Even Parity

### 8.2.16 MIF Rate Config (#18)

This parameter is used to configure the baud rate of the MIF interface. A reset/power cycle of the module is necessary in order for any changes to have effect.

**Note:** Automatic baud rate detection is not supported on this interface.

|                         |                        |
|-------------------------|------------------------|
| <b>Parameter number</b> | 18                     |
| <b>Modbus Address</b>   | 0x5012                 |
| <b>Default value</b>    | 0x04                   |
| <b>Range</b>            | 0x01 - 0x05 (see note) |
| <b>Size</b>             | 1 byte                 |
| <b>Stored in NV RAM</b> | Yes                    |
| <b>Access</b>           | R/W                    |

- **Values**

| <b>Value</b> | <b>Baud Rate</b>      |
|--------------|-----------------------|
| 0x01         | 4.8 kbit/s            |
| 0x02         | 9.6 kbit/s            |
| 0x03         | 19.2 kbit/s           |
| 0x04         | 38.4 kbit/s (default) |
| 0x05         | 57.6 kbit/s           |

**Note:** Additional baud rates may be available on certain Anybus implementations. For more information, consult each separate fieldbus appendix.

### 8.2.17 MIF Rate Actual (#19)

This parameter returns the actual baud rate settings for the MIF interface.

|                         |                        |
|-------------------------|------------------------|
| <b>Parameter number</b> | 19                     |
| <b>Modbus Address</b>   | 0x5013                 |
| <b>Default value</b>    | -                      |
| <b>Range</b>            | 0x01 - 0x05 (see note) |
| <b>Size</b>             | 1 byte                 |
| <b>Stored in NV RAM</b> | No                     |
| <b>Access</b>           | R                      |

- **Values**

| <b>Value</b> | <b>Baud Rate</b>      |
|--------------|-----------------------|
| 0x01         | 4.8 kbit/s            |
| 0x02         | 9.6 kbit/s            |
| 0x03         | 19.2 kbit/s           |
| 0x04         | 38.4 kbit/s (default) |
| 0x05         | 57.6 kbit/s           |

**Note:** Additional baud rates may be available on certain Anybus implementations. For more information, consult each separate fieldbus appendix.

### 8.2.18 MIF Settings Config (#20)

This parameter is used to configure the port settings of the MIF interface. A reset/power cycle of the module is necessary in order for any changes to have effect.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 20        |
| <b>Modbus Address</b>   | 0x5014    |
| <b>Default value</b>    | 0x00      |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 1 byte    |
| <b>Stored in NV RAM</b> | Yes       |
| <b>Access</b>           | R/W       |

#### Bit Layout

| b7 | b6 | b5 | b4 | b3 | b2   | b1   | b0   |
|----|----|----|----|----|------|------|------|
| -  | -  | -  | -  | -  | STOP | PAR1 | PAR2 |

- **PAR2**
  - 1: Enable parity
  - 0: Disable parity
- **PAR1**
  - (If parity is disabled (PAR2 = 0) this bit has no effect)
  - 1: Odd Parity
  - 0: Even Parity
- **STOP**
  - 1: 2 stop bits are used
  - 0: 1 stop bit is used

### 8.2.19 MIF Settings Actual (#21)

This parameter returns the actual port settings for the MIF interface.

|                         |           |
|-------------------------|-----------|
| <b>Parameter number</b> | 21        |
| <b>Modbus Address</b>   | 0x5015    |
| <b>Default value</b>    | -         |
| <b>Range</b>            | Bit field |
| <b>Size</b>             | 1 byte    |
| <b>Stored in NV RAM</b> | No        |
| <b>Access</b>           | R         |

#### Bit Layout

| b7 | b6 | b5 | b4 | b3 | b2   | b1   | b0   |
|----|----|----|----|----|------|------|------|
| -  | -  | -  | -  | -  | STOP | PAR1 | PAR2 |

- **PAR2**
  - 1: Parity is enabled
  - 0: Parity is disabled
- **PAR1**
  - (If parity is disabled (PAR2 = 0) this bit has no effect)
  - 1: Odd Parity
  - 0: Even Parity
- **STOP**
  - 1: 2 stop bits are used
  - 0: 1 stop bit is used

### 8.2.20 Modbus RTU Address (#22)

This parameter is used to configure the Modbus RTU Address used on the SCI channel.

**Note:** If automatic baud rate detection is used, the Modbus RTU address must be 01h.

|                         |             |
|-------------------------|-------------|
| <b>Parameter number</b> | 22          |
| <b>Modbus Address</b>   | 0x5016      |
| <b>Default value</b>    | 0x01        |
| <b>Range</b>            | 0x01 - 0xF7 |
| <b>Size</b>             | 1 byte      |
| <b>Stored in NV RAM</b> | Yes         |
| <b>Access</b>           | R/W         |

### 8.2.21 Modbus CRC Disable (#23)

This parameter is used to disable / enable the Modbus CRC. Disabling this will force the module to skip the CRC field in all Modbus messages (both in query and response messages), i.e. the CRC field will be completely removed from the message frame. Generally, it is not recommended to disable Modbus CRC checking. Use this function only in very special cases.

|                         |             |
|-------------------------|-------------|
| <b>Parameter number</b> | 23          |
| <b>Modbus Address</b>   | 0x5017      |
| <b>Default value</b>    | 0x00        |
| <b>Range</b>            | 0x00 - 0x01 |
| <b>Size</b>             | 1 byte      |
| <b>Stored in NV RAM</b> | Yes         |
| <b>Access</b>           | R/W         |

#### Values

- **0x00 - Enable Modbus CRC checking.**

Resulting Modbus message frame format:

| Start |      |      |  | Address | Function | Data    | CRC    | End  |      |      |  |
|-------|------|------|--|---------|----------|---------|--------|------|------|------|--|
| char  | char | char |  | 8bits   | 8bits    | n*8bits | 16bits | char | char | char |  |

- **0x01 - Disable Modbus CRC checking.**

Resulting Modbus message frame format:

| Start |      |      |  | Address | Function | Data    | End  |      |      |  |
|-------|------|------|--|---------|----------|---------|------|------|------|--|
| char  | char | char |  | 8bits   | 8bits    | n*8bits | char | char | char |  |

### 8.2.22 FB Fault Values (#27)

This parameter holds fault values that can be copied to the fieldbus output area. See parameter #10 “Offline Action” and parameter #11 “Idle Action” for more information.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 27              |
| <b>Modbus Address</b>   | 0x501B - 0x5032 |
| <b>Default value</b>    | 0x00            |
| <b>Range</b>            | 0x00 - 0xFF     |
| <b>Size</b>             | 48 bytes        |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |



## 8.3 I/O Parameters

Each communication channel features its own set of I/O parameters, which determine how produced data shall be mapped to other channels. These parameters must be properly set up during initialization (i.e. prior to setting Module Mode to 0x0001).

| #  | Modbus Address | Name               | Size    | Default | Access |
|----|----------------|--------------------|---------|---------|--------|
| 40 | 0x6000         | FB Byte Order      | 1 byte  | 0x00    | R/W    |
| 41 | 0x6001         | FB Out Config      | 2 bytes | 0x0000  | R/W    |
| 42 | 0x6002         | FB Out Actual      | 2 bytes | -       | R      |
| 43 | 0x6003         | FB In Actual       | 2 bytes | -       | R      |
| 44 | 0x6004         | FB In SSC Offset   | 2 bytes | 0x0000  | R/W    |
| 45 | 0x6005         | FB In SSC Size     | 2 bytes | 0x0000  | R/W    |
| 46 | 0x6006         | FB In SCI Offset   | 2 bytes | 0x0000  | R/W    |
| 47 | 0x6007         | FB In SCI Size     | 2 bytes | 0x0000  | R/W    |
| 50 | 0x600A         | SSC Byte Order     | 1 byte  | 0x00    | R/W    |
| 51 | 0x600B         | SSC In Config      | 2 bytes | 0x0000  | R/W    |
| 52 | 0x600C         | SSC In Auto        | 2 bytes | -       | R      |
| 53 | 0x600D         | SSC In Actual      | 2 bytes | -       | R      |
| 54 | 0x600E         | SSC Out Config     | 2 bytes | 0x0000  | R/W    |
| 55 | 0x600F         | SSC Out Auto       | 2 bytes | -       | R      |
| 56 | 0x6010         | SSC Out Actual     | 2 bytes | -       | R      |
| 57 | 0x6011         | SSC Out FB Offset  | 2 bytes | 0x0000  | R/W    |
| 58 | 0x6012         | SSC Out FB Size    | 2 bytes | 0x0000  | R/W    |
| 59 | 0x6013         | SSC Out SCI Offset | 2 bytes | 0x0000  | R/W    |
| 60 | 0x6014         | SSC Out SCI Size   | 2 bytes | 0x0000  | R/W    |
| 63 | 0x6017         | SCI Byte Order     | 1 byte  | 0x00    | R/W    |
| 64 | 0x6018         | SCI In Config      | 2 bytes | 0x0000  | R/W    |
| 65 | 0x6019         | SCI In Actual      | 2 bytes | -       | R      |
| 66 | 0x601A         | SCI Out Actual     | 2 bytes | -       | R      |
| 67 | 0x601B         | SCI Out FB Offset  | 2 bytes | 0x0000  | R/W    |
| 68 | 0x601C         | SCI Out FB Size    | 2 bytes | 0x0000  | R/W    |
| 69 | 0x601D         | SCI Out SSC Offset | 2 bytes | 0x0000  | R/W    |
| 70 | 0x601E         | SCI Out SSC Size   | 2 bytes | 0x0000  | R/W    |

**Note:** In this document, register numbers are specified using the protocol convention (base 0), i.e. there is a 1:1 correlation between the register number specified in this document and the actual register value in the message frame.

### 8.3.1 FB Byte Order (#40)

This parameter determines if the bytes in the fieldbus I/O area shall be byte swapped or not relative to the other I/O areas.

Note that in order for this function to work properly, the I/O length must be a multiple of 16 bits / 2 bytes.

*Example:*

If data is mapped from the “FB Output area” to the “SSC Output area”, parameter #40 “FB Byte Order” is 0x00, and parameter #50 “SSC Byte Order” is 0x01, the data written to the “FB Output area” will be swapped when it reaches the “SSC Output area”. If the byte order parameters has the same value for both areas, no swap will be made.

**Note:** Both the FB Input area and the FB Output area are affected by this parameter.

|                         |             |
|-------------------------|-------------|
| <b>Parameter number</b> | 40          |
| <b>Modbus Address</b>   | 0x6000      |
| <b>Default value</b>    | 0x00        |
| <b>Range</b>            | 0x00 - 0x01 |
| <b>Size</b>             | 1 byte      |
| <b>Stored in NV RAM</b> | Yes         |
| <b>Access</b>           | R/W         |

#### Values

- **0x00**  
Do not swap bytes in the fieldbus I/O area.
- **0x01**  
Swap bytes in the fieldbus I/O area.

### 8.3.2 FB Out Config (#41)

This parameter configures the size of the FB Out area. The size is specified in bytes.

|                         |                  |
|-------------------------|------------------|
| <b>Parameter number</b> | 41               |
| <b>Modbus Address</b>   | 0x6001           |
| <b>Default value</b>    | 0x0000           |
| <b>Range</b>            | 0x0000 - 0x0090h |
| <b>Size</b>             | 2 bytes          |
| <b>Stored in NV RAM</b> | Yes              |
| <b>Access</b>           | R/W              |

### 8.3.3 FB Out Actual (#42)

This parameter holds the actual size of the FB Out area after initialization. The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 42              |
| <b>Modbus Address</b>   | 0x6002          |
| <b>Modbus Address</b>   |                 |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0090 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.4 FB In Actual (#43)

This parameter holds the actual size of the FB In area after initialization. The value of this parameter is calculated using parameter #45 “FB In SSC Size” and parameter #47 “FB In SCI Size”. The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 43              |
| <b>Modbus Address</b>   | 0x6003          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0090 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.5 FB In SSC Offset (#44)

This parameter is used to set the source location for the SSC Input -> Fieldbus Input mapping.

|                         |                  |
|-------------------------|------------------|
| <b>Parameter number</b> | 44               |
| <b>Modbus Address</b>   | 0x6004           |
| <b>Default value</b>    | 0x0000           |
| <b>Range</b>            | 0x0000h - 0x000F |
| <b>Size</b>             | 2 bytes          |
| <b>Stored in NV RAM</b> | Yes              |
| <b>Access</b>           | R/W              |

### 8.3.6 FB In SSC Size (#45)

This parameter is used to specify how many bytes that will be mapped from the SSC Input area to the Fieldbus Input area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 45              |
| <b>Modbus Address</b>   | 0x6005          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.7 FB In SCI Offset (#46)

This parameter is used to set the source location for the SCI Input -> Fieldbus Input mapping.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 46              |
| <b>Modbus Address</b>   | 0x6006          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x007F |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.8 FB In SCI Size (#47)

This parameter is used to specify how many bytes that will be mapped from the SCI Input area to the Fieldbus Input area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 47              |
| <b>Modbus Address</b>   | 0x6007          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0080 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.9 SSC Byte Order (#50)

This parameter determines if the bytes in the SSC I/O area shall be swapped or not relative to the other I/O areas.

#### *Example*

If data is mapped from the “SSC Input area” to the “FB Input area”, parameter #50 “SSC Byte Order” is 0x00, and parameter #40 “FB Byte Order” is 0x01, the data written to the “SSC Input area” will be swapped when it reaches the “FB Input area”. If the byte order parameters has the same value for both areas, no swap will be made.

Note that in order for this function to work properly, the I/O length must be a multiple of 16 bits / 2 bytes.

**Note:** The entire SSC I/O area is affected by the state of this parameter no matter of the contents.

|                         |             |
|-------------------------|-------------|
| <b>Parameter number</b> | 50          |
| <b>Modbus Address</b>   | 0x600A      |
| <b>Default value</b>    | 0x00        |
| <b>Range</b>            | 0x00 - 0x01 |
| <b>Size</b>             | 1 byte      |
| <b>Stored in NV RAM</b> | Yes         |
| <b>Access</b>           | R/W         |

#### Values

- **0x00**  
Do not swap bytes in the SSC I/O area.
- **0x01**  
Swap bytes in the SSC I/O area.

### 8.3.10 SSC In Config (#51)

This parameter is used to configure the total size of the SSC In area (the FB specific input byte is not included in this size). The size is specified in bytes.

In order for the value of this parameter to be valid after initialization, the SSCI bit in parameter #8 “Configuration Bits” must be set.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 51              |
| <b>Modbus Address</b>   | 0x600B          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.11 SSC In Auto (#52)

This parameter returns the automatically configured size of the SSC In area (the FB specific input byte is not included in this size). The size is specified in bytes.

In order for the value of this parameter to be valid after initialization, the SSCI bit in parameter #8 “Configuration Bits” must be cleared.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 52              |
| <b>Modbus Address</b>   | 0x600C          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.12 SSC In Actual (#53)

This parameter returns the actual size of the SSC In area after initialization (the fieldbus specific input byte is not included in this size). The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 53              |
| <b>Modbus Address</b>   | 0x600D          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.13 SSC Out Config (#54)

This parameter is used to configure the total size of the SSC Out area (the FB specific output byte is not included in this size). The size is specified in bytes.

In order for the value of this parameter to be valid after initialization, the SSCO bit in parameter #8 “Configuration Bits” must be set.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 54              |
| <b>Modbus Address</b>   | 0x600E          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.14 SSC Out Auto (#55)

This parameter returns the automatically configured size of the SSC Out area (the FB specific output byte is not included in this size). The size is specified in bytes.

In order for the value of this parameter to be valid after initialization, the SSC0 bit in parameter #8 “Configuration Bits” must be cleared.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 55              |
| <b>Modbus Address</b>   | 0x600F          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.15 SSC Out Actual (#56)

This parameter returns the actual size of the SSC Out area after initialization (the FB specific output byte is not included in this size). The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 56              |
| <b>Modbus Address</b>   | 0x6010          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.16 SSC Out FB Offset (#57)

This parameter is used to set the source location for the Fieldbus Output -> SSC output mapping.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 57              |
| <b>Modbus Address</b>   | 0x6011          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x008F |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.17 SSC Out FB Size (#58)

This parameter is used to specify how many bytes that will be mapped from the Fieldbus Output area to the SSC Output area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 58              |
| <b>Modbus Address</b>   | 0x6012          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.18 SSC Out SCI Offset (#59)

This parameter is used to set the source location for the SCI Input -> SSC Output mapping.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 59              |
| <b>Modbus Address</b>   | 0x6013          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x007F |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.19 SSC Out SCI Size (#60)

This parameter is used to specify how many bytes that will be mapped from the SCI Input area to the SSC Output area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 60              |
| <b>Modbus Address</b>   | 0x6014          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |



### 8.3.20 SCI Byte Order (#63)

This parameter determines if the bytes in the SCI I/O area shall be swapped or not relative to the other I/O areas.

*Example:*

If data is mapped from the “SCI Input area” to the “FB Input area”, parameter #63 “SCI Byte Order” is 0x00, and parameter #40 “FB Byte Order” is 0x01, the data written to the “SCI Input area” will be swapped when it reaches the “FB Input area”. If the byte order parameters has the same value for both areas, no swap will be made.

Note that in order for this function to work properly, the I/O length must be a multiple of 16 bits / 2 bytes.

**Note:** The entire SCI I/O area is affected by the state of this parameter no matter of the contents.

|                  |             |
|------------------|-------------|
| Parameter number | 63          |
| Modbus Address   | 0x6017      |
| Default value    | 0x00        |
| Range            | 0x00 - 0x01 |
| Size             | 1 byte      |
| Stored in NV RAM | Yes         |
| Access           | R/W         |

#### Values

- **0x00**  
Do not swap bytes in the SCI I/O area.
- **0x01**  
Swap bytes in the SCI I/O area.

### 8.3.21 SCI In Config (#64)

This parameter configures the size of the SCI In area. The size is specified in bytes.

|                  |                 |
|------------------|-----------------|
| Parameter number | 64              |
| Modbus Address   | 0x6018          |
| Default value    | 0x0000          |
| Range            | 0x0000 - 0x0080 |
| Size             | 2 bytes         |
| Stored in NV RAM | Yes             |
| Access           | R/W             |

### 8.3.22 SCI In Actual (#65)

This parameter holds the actual size of the SCI In area after initialization. The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 65              |
| <b>Modbus Address</b>   | 0x6019          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0080 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.23 SCI Out Actual (#66)

This parameter holds the actual size of the SCI Out area after initialization. The value of this parameter is calculated using parameter #68 “SCI Out FB Size” and parameter #70 “SCI Out SSC Size”. The size is specified in bytes.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 66              |
| <b>Modbus Address</b>   | 0x601A          |
| <b>Default value</b>    | -               |
| <b>Range</b>            | 0x0000 - 0x0080 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | No              |
| <b>Access</b>           | R               |

### 8.3.24 SCI Out FB Offset (#67)

This parameter is used to set the source location for the Fieldbus Output -> SCI Output mapping.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 67              |
| <b>Modbus Address</b>   | 0x601B          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x007F |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.25 SCI Out FB Size (#68)

This parameter is used to specify how many bytes that will be mapped from the Fieldbus Output area to the SCI Output area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 68              |
| <b>Modbus Address</b>   | 0x601C          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0080 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.26 SCI Out SSC Offset (#69)

This parameter is used to set the source location for the SSC Input -> SCI Output mapping.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 69              |
| <b>Modbus Address</b>   | 0x601D          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x000F |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

### 8.3.27 SCI Out SSC Size (#70)

This parameter is used to specify how many bytes that will be mapped from the SSC Input area to the SCI Output area.

|                         |                 |
|-------------------------|-----------------|
| <b>Parameter number</b> | 70              |
| <b>Modbus Address</b>   | 0x601E          |
| <b>Default value</b>    | 0x0000          |
| <b>Range</b>            | 0x0000 - 0x0010 |
| <b>Size</b>             | 2 bytes         |
| <b>Stored in NV RAM</b> | Yes             |
| <b>Access</b>           | R/W             |

## 8.4 Fieldbus Specific Parameters

Consult each separate fieldbus appendix for further information.

## 9. Miscellaneous

### 9.1 Interrupt (/INT) & Bootloader Enable (BLE)

During runtime, this pin acts as an active low interrupt output. Events that shall generate an interrupt are specified in parameter #12 (“Interrupt Config”).

When an interrupt has occurred (i.e. when the interrupt pin has gone low), the cause of the interrupt can be read from parameter #13 (“Interrupt Cause”). The value of this register holds its value until it has been read by the application. When read, the interrupt is cleared and the interrupt signal goes high again.

If this pin is connected to ground (GND) during power on, the module will start in a special boot loader mode, allowing new firmware to be downloaded via the MIF interface. Generally, this function should only be used if erroneous data has accidentally been downloaded into the on-board flash. Normally, firmware upgrades should be performed via the ‘Firmware Upgrade’ menu.

**Note:** Do not connect this pin to ground during normal operation.

See also...

- “Application Connector” on page 14
- “MIF Interface” on page 21
- “Interrupt Config (#12)” on page 62
- “Interrupt Cause (#13)” on page 63

### 9.2 Reset (/RESET)

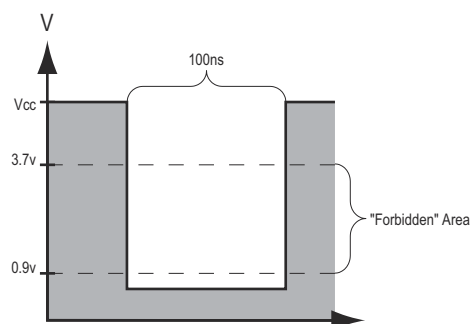
To trigger a hardware reset, a high-to-low transition with a minimum duration of 100 ns is necessary (see figure).

It is generally recommended to connect a 100 nF capacitor between the reset signal and ground (GND).

If not used, this signal can safely be connected directly to Vcc.

See also...

- “Application Connector” on page 14



**Note:** This signal may possess an internal load capacitance of up to 100 nF. This means that a capacitive load in excess of 100 nF must be taken into account when designing the application.

## 9.3 Self Test Sequence

### 9.3.1 General Information

The application can instruct the module to perform a self test sequence using parameter #1 “Module Mode”. The result of the test is presented in parameter #2 “Module Status”.

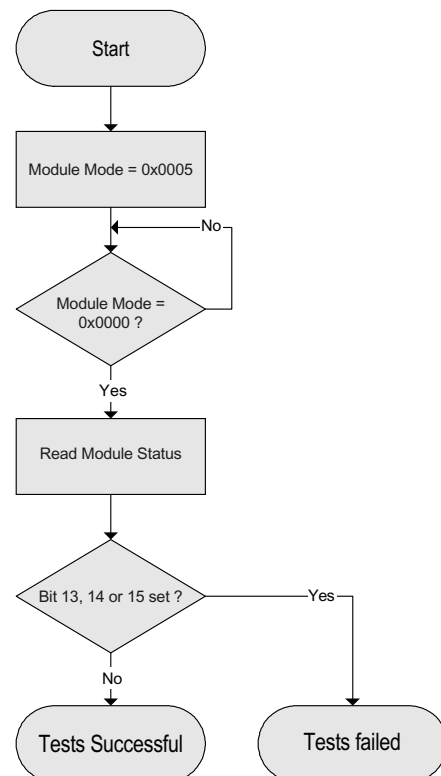
**Note:** This test can only be performed before the module is initialized.

### 9.3.2 Basic Procedure

1. Start the hardware tests by writing 0x0005 to parameter #1 (“Module Mode”).
2. During the test, parameter #1 (“Module Mode”) has the value 0x0005. When tests are finished, the value will be 0x0000.
3. Read parameter #2 (“Module Status”) to get information about the tests.

### 9.3.3 Test Evaluation

If bit 13, 14 or 15 is set, the corresponding test has failed. If the bit is cleared, the corresponding test passed successfully.

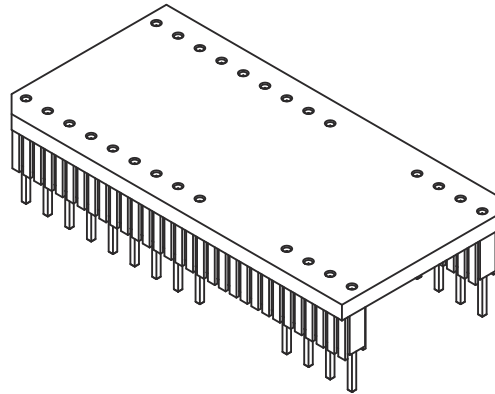


## A. Mechanical Specification

### A.1 General Information

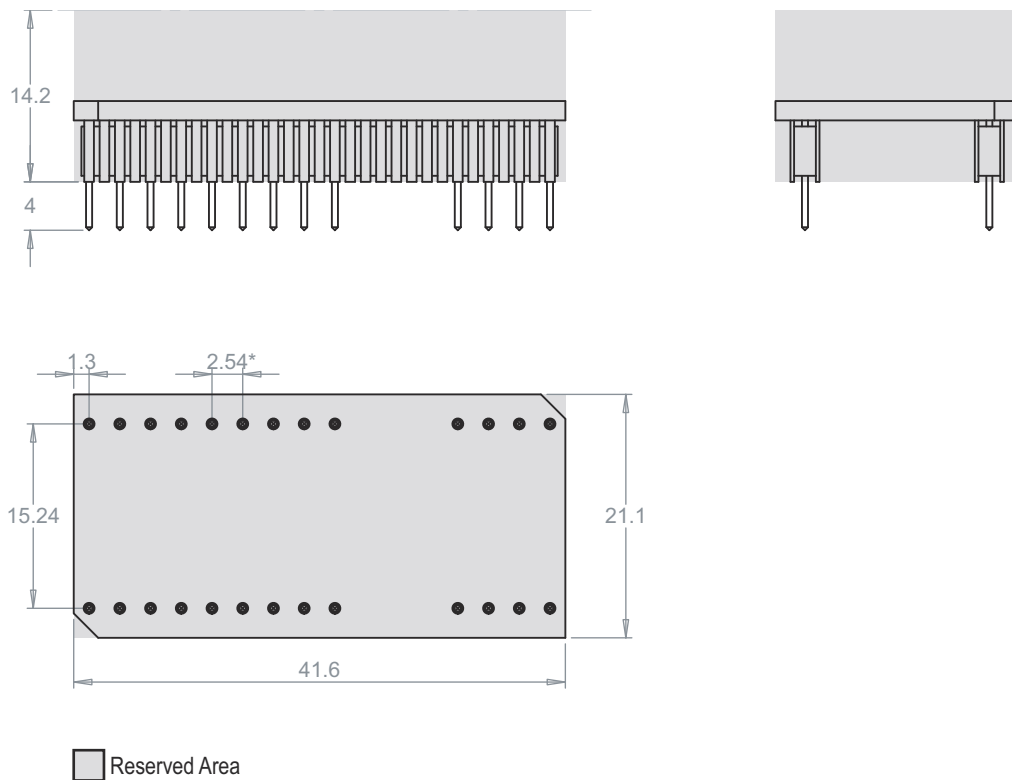
The application interface uses a standard DIL-32 footprint.

It is generally recommended to use a DIL socket instead of soldering the module directly to the application circuit board.



### A.2 Measurements

All measurements are in millimeters, tolerance is  $\pm 0.20$  mm unless otherwise stated.



\*The 2.54 mm measurement from pin to pin conforms to DIL tolerance levels.

## **B. Firmware Upgrade**

The firmware of the module can be updated using the HMS firmware download utility.

For more information, contact HMS technical support.



## C. Object Messaging (0x5B)

### C.1 General Information

Modbus Object Messaging, originally developed for Modbus-TCP, is an extension to the standard Modbus protocol which is used to address functions and data in an object oriented manner. To suit the Anybus-IC, certain changes/additions have been made to the original specification.

Object Messaging is needed for certain fieldbus specific functionality, consult each separate fieldbus appendix for further information.

|                            |                    |
|----------------------------|--------------------|
| <b>Function Name</b>       | 'Object Messaging' |
| <b>Function Code</b>       | 0x5B               |
| <b>Broadcast Supported</b> | No                 |

### C.2 Message Format

The Modbus object messaging is based on dividing the standard data field in Modbus into 7 subfields according to the figure below.

#### Message Frame

| Address   | Function | Subfield    | CRC   |
|-----------|----------|-------------|-------|
| (address) | 91       | (see below) | (CRC) |

#### Query - Subfield Format

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Attribute | Data      | Stuff byte |
|---------------------|-------------------|----------|-------------|--------------|-----------|-----------|------------|
| 8 bits              | 8 bits            | 16 bits  | 16 bits     | 16 bits      | 16 bits   | n*16 bits | 8 bits     |

#### Response - Subfield Format

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Error Code | Data      | Stuff byte |
|---------------------|-------------------|----------|-------------|--------------|------------|-----------|------------|
| 8 bits              | 8 bits            | 16 bits  | 16 bits     | 16 bits      | 16 bits    | n*16 bits | 8 bits     |

**Note:** When using fragmented Object Messaging the receiving part (either master or slave) should acknowledge each fragment by increasing the Fragment Sequence Number and send a response where the Class ID, Instance ID, Service Code, Attribute, eventual Error Code and Data is to be left out, only leaving the Fragment byte count and Fragment protocol of the Object messaging subfield.

## C.3 Subfield Contents

### C.3.1 Fragment Byte Count

This field contains the number of bytes of the current object message (itself excluded).

The maximum number of bytes is 197.

**Note:** If a stuff byte is added at the end of the message, it is not included in the Fragment byte count.

### C.3.2 Fragment Protocol

This field is used when sending fragmented messages. The bits are described below.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| FSN   |       |       | -     | -     | -     | LFI   | FIPI  |

- **FIPI - Fragment In Process Indicator**

1: This message is a fragment of a fragmented message.

0: This message is not fragmented.

- **LFI - Last Fragment Indicator**

1: This is the last fragment in a fragmented message.

0: This not the last fragment in a fragmented message.

- **FSN - Fragment Sequence Number**

This is a counter counting from  $000_2$  to  $111_2$ . Each fragment in a fragmented message has a sequential number in this field.

### C.3.3 Class ID

ID of the object class associated with the service.

### C.3.4 Instance ID

ID of the instance associated with the service.

### C.3.5 Error Codes

In the response from the Anybus-IC, the following error codes may be used.

- **General Errors**

| Value  | Description                    | Comments  |
|--------|--------------------------------|---|
| 0x0000 | Success                        | -   |
| 0x0001 | Invalid Service Code           | The requested service is not implemented or defined for this object.  |
| 0x0002 | Invalid Service Code Parameter | The parameters required to perform the service are invalid.           |
| 0x0003 | Invalid Attribute              | The specified attribute is not supported in this object.              |
| 0x0004 | Attribute out of range         | Set value is out of range for the attribute.                          |
| 0x0005 | Not valid in this state        | The object cannot perform the requested service in its current state. |
| 0x0006 | Fragmentation error            | Error in message fragmentation.                                       |
| 0x00FF | Unspecified error              | -   |

- **Manufacturer Specific Errors**

| Value  | Description            | Comments   |
|--------|------------------------|--|
| 0x0101 | Class Not Supported    | The class specified is not implemented in this device.                             |
| 0x0104 | Instance Not Supported | The specified instance does not exist.   |
| 0x0105 | Instance Already Exist | The requested instance to be created already exists.                               |
| 0x0107 | Attribute Not Settable | A request to modify a nonmodifiable attribute.                                     |
| 0x0109 | Not Enough Data        | The message provides less data than was needed for the attribute.                  |
| 0x010A | Too Much Data          | The message provides more data than was needed for the attribute.                  |
| 0x010C | Resource Unavailable   | Resources needed for the object to perform the requested service were unavailable. |
| 0x010D | Device State Conflict  | The device's current mode/state prohibits the execution of the requested service.  |
| 0x010F | Attribute Not Gettable | A request to a write-only attribute.   |

### C.3.6 Service Code

| Value            | Description            | Comments   |
|------------------|------------------------|--|
| 0x0001           | Get Attribute          | This command retrieves the value of an attribute. The 16 bit attribute number is specified in field Attribute <sup>a</sup> . The Data field is not used.                 |
| 0x0002           | Get Attribute Response | A response to a Get Attribute command returns the value of an attribute in the Data field <sup>a</sup> .   |
| 0x0003           | Set Attribute          | This command assigns a value to an attribute. The 16 bit attribute number is specified in field Attribute <sup>a</sup> . The Data field contains the data to be written. |
| 0x0004           | Set Attribute Response | A response to a Set Attribute command contains no data, i.e. the Data field <sup>a</sup> is not used.  |
| 0x0005           | Create                 | This command creates a new instance within the object.   |
| 0x0006           | Create Response        | -  |
| 0x0007           | Remove                 | -  |
| 0x0008           | Remove Response        | -  |
| 0x0009           | Reset                  | This command performs a reset command on an object.  |
| 0x000A           | Reset Response         | -  |
| 0x000B           | Start                  | -  |
| 0x000C           | Start Response         | -  |
| 0x000D           | Stop                   | -  |
| 0x000E           | Stop Response          | -  |
| 0x000F           | Save                   | -  |
| 0x0010           | Save Response          | -  |
| 0x0011           | Restore                | -  |
| 0x0012           | Restore Response       | -  |
| 0x0013           | Nop                    | -  |
| 0x0014           | Nop Response           | -  |
| 0x0015... 0x007F | -                      | Not used   |
| 0x0080... 0x00FF | -                      | Class specific services  |

a. See page C-89 "Message Format"

### C.3.7 Attribute

If a message is fragmented (i.e. several messages to transmit one big data area) the Anybus-IC expects the Attribute to be a part of every fragment. If an attribute is not used in the service, this word must be 0x0000. This is an HMS specific addition to the official Modbus Object Messaging specification.

### C.3.8 Data

Data associated with the service.

### C.3.9 Stuff Byte

The length of the standard data field in modbus must always be a multiple of 16 bits. If it is not, the stuff byte is added by the end of the fragment to ensure this. The stuff byte does not contain any meaningful data and is not included in the Fragment byte count.

## D. Technical Specification

**Note:** The properties specified in this chapter applies to all Anybus-IC modules unless otherwise stated. Any deviations from what is stated herein is specified separately in each network appendix.

### D.1 Environmental

#### Temperature

Operating: -40 to +85 °C (-40 to 185°F)  
Storage: -40 to +85 °C (-40 to 185°F)

(Tests performed according to IEC 68-2-1 and IEC 68-2-2)

#### Wave Soldering

Temperature: 320°C  
Velocity: 10 mm/s

#### Humidity

5 to 95% noncondensing (tests performed according to IEC 68-2-30)

### D.2 Power Supply

#### Supply Voltage

The module requires a regulated +5 V  $\pm$  5% DC power supply.

The power supply rise time (0.1 V<sub>CC</sub> to 0.9 V<sub>CC</sub>) must be less than 50 ms.

#### Power Consumption

The maximum power consumption may vary. Please consult each separate fieldbus appendix.

#### Isolation

Isolation distances between application and network (according to EN 60950-1; Pollution degree 2; Material Group IIIB):

| Isolation Barrier                   | Working Voltage/Transient Voltage |              | Distance |          |
|-------------------------------------|-----------------------------------|--------------|----------|----------|
|                                     | Creepage                          | Clearance    | External | Internal |
| Application to Network <sup>a</sup> | 250 V/2500 V                      | 250 V/2500 V | 2.5 mm   | 0.4 mm   |

a. Valid for all modules except Ethernet modules which utilize a customer supplied transformer.

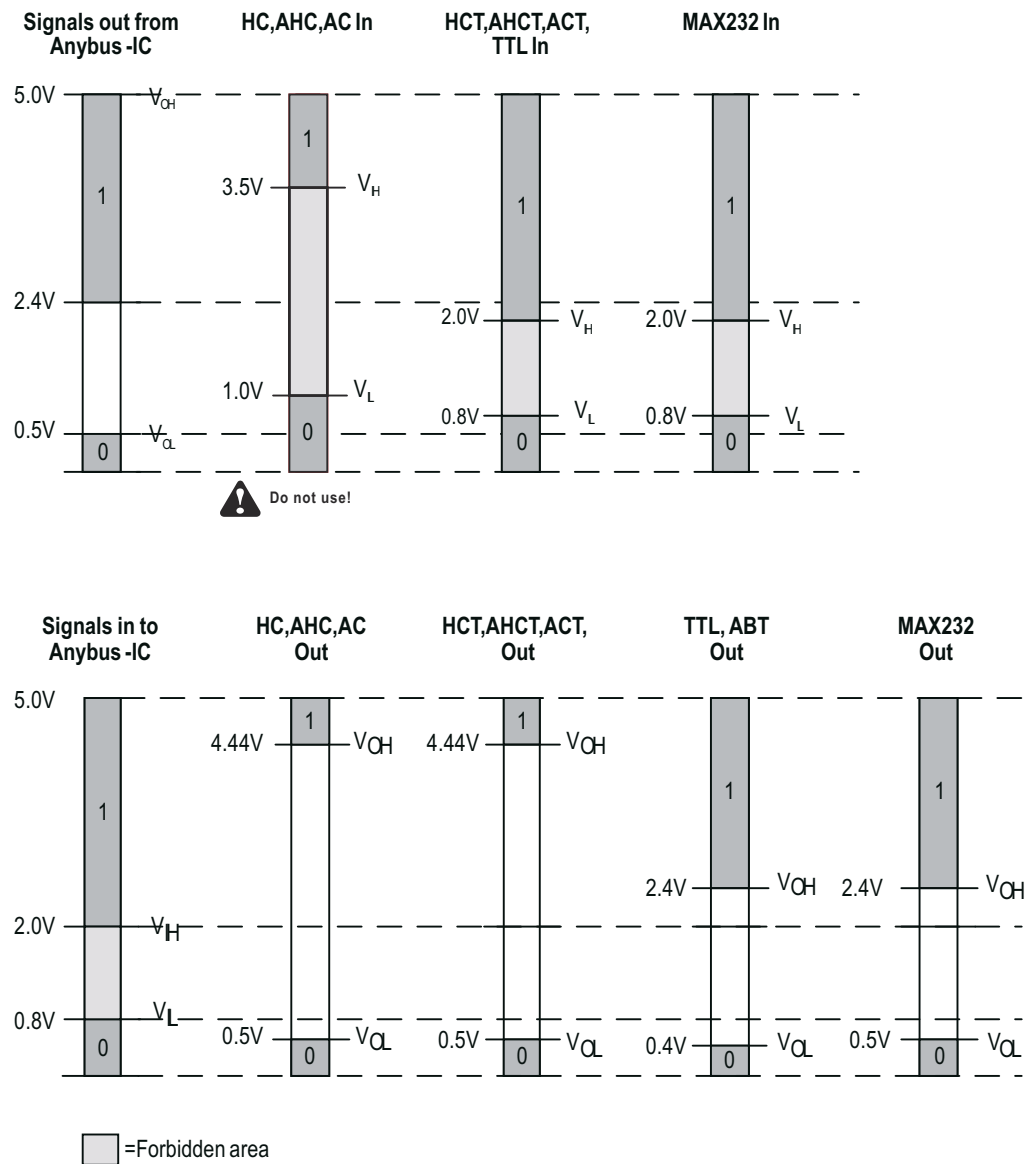
#### Protective Earth & Shielding

PE-requirements are fieldbus dependent. Please consult the separate fieldbus appendix for details.

## D.3 Signal Levels

It is recommended to use 74HCTxxx / 74ACTxxx type TTL circuits. 74HCxxx, 74ACxxx, and 74AHCxxx type circuits are not recommended, and may not work properly as the “forbidden area” of these types of circuits is not within the boundaries allowed by the Anybus-IC. (See figure below)

When connecting current demanding components (e.g LEDs) directly to the outputs of the SSC shift registers, it is required to use shift registers of sufficient current capacity, e.g. 74ACTxxx.



**Note:** For exact signal levels, consult the data sheet for the used logic circuit. Signal levels may depend on temperature and supply voltage.

## D.4 Regulatory Compliance

### D.4.1 Fieldbus Certification

All Anybus-IC modules are precertified and found to comply with each fieldbus standard. Please note that although the module itself has been precertified, the final product may still require recertification depending on the fieldbus standard.

This precertification is valid under the following conditions:

- Standard fieldbus connectors
- No fieldbus specific initialization parameters
- Nonmodified device description file (i.e. '.GSD' or '.EDS')

Any changes to the conditions above invalidates the precertification. For more information, consult each fieldbus standard specification and/or contact HMS.

### D.4.2 EMC Compliance (CE)

Generally, Anybus-IC products are certified according to the European CE standard unless otherwise stated. It is however important to note that although the Anybus module itself is certified, the final product may still require recertification depending on the application.

### D.4.3 UL/cUL-Certificate

The Anybus-IC modules are UL/cUL recognized for the US (NRAQ2) and Canada (NRAQ8) according to UL508, "Programmable Controller".