SIMATIC Sensors

RFID systems
RF182C communication module

Operating Instructions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Description</td>
<td>2</td>
</tr>
<tr>
<td>Mounting</td>
<td>3</td>
</tr>
<tr>
<td>Connecting</td>
<td>4</td>
</tr>
<tr>
<td>Parameterizing</td>
<td>5</td>
</tr>
<tr>
<td>Communication interface</td>
<td>6</td>
</tr>
<tr>
<td>Maintenance and Service</td>
<td>7</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>8</td>
</tr>
<tr>
<td>Error messages</td>
<td>9</td>
</tr>
<tr>
<td>Examples/applications</td>
<td>10</td>
</tr>
<tr>
<td>Technical data</td>
<td>11</td>
</tr>
<tr>
<td>Dimension drawings</td>
<td>12</td>
</tr>
<tr>
<td>Connecting cable to the reader/SLG</td>
<td>13</td>
</tr>
<tr>
<td>Ordering data</td>
<td>14</td>
</tr>
<tr>
<td>Command and acknowledgement telegrams</td>
<td>A</td>
</tr>
<tr>
<td>Addressing of the RFID tags</td>
<td>B</td>
</tr>
<tr>
<td>Transfer scheme for hexadecimal tag data via XML</td>
<td>C</td>
</tr>
<tr>
<td>Service &amp; support</td>
<td>D</td>
</tr>
</tbody>
</table>
Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

**DANGER**
indicates that death or severe personal injury will result if proper precautions are not taken.

**WARNING**
indicates that death or severe personal injury may result if proper precautions are not taken.

**CAUTION**
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

**CAUTION**
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

**NOTICE**
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

**WARNING**
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.
Table of contents

1 Introduction ................................................................................................................................................ 5
1.1 Introduction ....................................................................................................................................5

2 Description............................................................................................................................................. 7

3 Mounting.................................................................................................................................................. 13
3.1 Mounting position, mounting dimensions.....................................................................................13
3.2 Mounting the I/O module..............................................................................................................14
3.3 Mounting the connection block ....................................................................................................16
3.4 Replacing labels...........................................................................................................................18
3.5 Disassembling the RF182C.........................................................................................................19

4 Connecting .............................................................................................................................................. 21
4.1 Wiring connection block M12, 7/8"...............................................................................................24
4.2 Wiring of the push-pull connection block .....................................................................................27
4.3 Loop-through of Ethernet and supply voltage..............................................................................30
4.4 Wiring an RF182C to a controller with Ethernet connection........................................................32
4.5 Connecting the RF182C to functional ground (PE) .....................................................................33

5 Parameterizing ........................................................................................................................................ 35
5.1 Address assignment for Ethernet.................................................................................................35
5.2 Data communication between client and RF182C.......................................................................37
5.3 Factory setting of the RF182C.....................................................................................................40
5.4 Assigning the IP address.............................................................................................................41
5.4.1 Overview ......................................................................................................................................41
5.4.2 Web server...................................................................................................................................41
5.4.3 Primary Setup Tool ......................................................................................................................45
5.5 Troubleshooting: Assigning the IP address .................................................................................48

6 Communication interface ......................................................................................................................... 49
6.1 Overview of commands ...............................................................................................................49
6.2 Configuration parameters of the RF182C....................................................................................50
6.3 Input parameters of the RF182C ..................................................................................................52
6.4 Commands of the communication module ..................................................................................53
6.4.1 writeTagData...........................................................................................................................53
6.4.2 readTagData............................................................................................................................54
6.4.3 initializeTag............................................................................................................................55
6.4.4 getReaderStatus.......................................................................................................................56
6.4.5 getTagStatus............................................................................................................................57
6.4.6 setAnt...........................................................................................................................................58
6.4.7 heartbeat......................................................................................................................................59
6.5 Asynchronous message frames ................................................................. 60
6.5.1 tagPresent ............................................................................................. 60
6.5.2 alarm ........................................................................................................ 61

7 Maintenance and Service ........................................................................ 63
7.1 Replacing the RF182C communication module .................................. 63
7.2 Firmware update ...................................................................................... 65
7.3 Reader update .......................................................................................... 65

8 Diagnostics .............................................................................................. 67
8.1 Diagnostics using LEDs ................................................................. 67

9 Error messages ....................................................................................... 71
9.1 Response without error entry .............................................................. 71
9.2 Response with error entry ..................................................................... 71
9.3 Error messages of the RF182C ............................................................ 72
9.4 Diagnostics via Web server ................................................................. 78
9.4.1 Saving/reading of I&M data records .................................................... 78
9.4.2 Communication status query .......................................................... 79
9.4.3 Event and message frame overview ................................................ 80

10 Examples/applications ........................................................................... 81
10.1 Basic principles of socket programming, exemplary in C .................. 81
10.1.1 Socket programming requirements ................................................ 81
10.1.2 Basic client/server principle ........................................................ 82
10.1.3 Important basic commands .......................................................... 82
10.1.4 Partial programming example of a client in C/Windows operating system .................. 83
10.2 RF182C user application ................................................................. 85
10.2.1 User interface layout ................................................................. 86
10.2.2 Extracts example code of the user application in C# ................. 87
10.2.3 Functions of the RF182C applications ........................................ 95
10.3 Example application for a PLC according to DIN IEC 61131 .............. 99

11 Technical data ....................................................................................... 101

12 Dimension drawings ............................................................................. 103
12.1 Dimension drawing for RF182C with fixing holes ................................ 103

13 Connecting cable to the reader/SLG .................................................... 105
13.1 Routing of standard cables .............................................................. 105
13.2 Self-assembled cable .......................................................................... 107

14 Ordering data ....................................................................................... 109
A Command and acknowledgement telegrams ....................................... 111
B Addressing of the RFID tags ................................................................. 123
C Transfer scheme for hexadecimal tag data via XML .............................. 125
D Service & support ................................................................................... 127
1.1 Introduction

Purpose of these operating instructions
The information provided in these Operating Instructions enables you to operate the RF182C communication module on a standard PC or a PLC.

Basic knowledge required
These operating instructions assume general knowledge of automation engineering and identification systems.
You also require basic knowledge of socket programming (TCP/IP communication via Ethernet). Socket programming depends on the programming language or the operating system used (Windows, Linux, or Unix).

Scope of this manual
The Operating Instructions apply to the RF182C communication module.

Position in the information landscape
- The manual of the relevant RFID family contains information on the readers/SLGs to be connected.
- Special information on parameterizing the RF620R/RF630R readers in conjunction with the RF182C communication module can be found in the "RF620R/RF630R parameterization manual".

Guide
These Operating Instructions describe the hardware and the communications interface of the RF182C communication module. They comprise introductory sections and reference sections (e.g. technical data).
The operating instructions include the following subject areas:
- Connection of the RF182C communication module
- Parameterization and programming of the RF182C communication module
- Diagnostics information
- Display elements of the RF182C communication module
- Information on repair and maintenance (e.g. firmware update)
- Technical data as well as dimension drawings of the RF182C communication module
- Ordering data
Recycling and disposal

- Due to its environmentally compatible equipment, the RF182C communication module can be recycled.
- Contact a certified electronic-waste disposal company to recycle and dispose of your old equipment in an environment-friendly manner.
Description

Area of application

The RF182C communication module is a module that is used for operating RFID components on a standard PC or PLC over Ethernet.

RF182C communication module
With connection block M12, 7/8"      With push-pull connection block

When using it on a standard PC, please follow the appropriate instructions for parameterization and integration in the system.

The following RFID families can be operated with the RF182C (only with normal addressing):

- RF300
- RF600
- MOBY D
- MOBY U

Features

Up to 2 readers/SLGs can be operated on the RF182C at the same time. The user can issue a command on 2 readers/SLGs simultaneously.

The tag data is accessed by means of physical addressing of the tag.
Other features

- Degree of protection IP67
- System integration with M12, 7/8” concept or with push-pull concept
- Standardized Ethernet interface
- Diagnostics support via web server
- Routing capability
- Firmware update via web server
- Support of identification and maintenance data sets (I&M): Mechanism for reading out information via the communication module, and saving system information such as function, installation date, installation location, and comments
- Module supports SNMP

Layout

The RF182C has the same enclosure as the RFID communication module ASM 456 for PROFIBUS and the RFID communication module RF180C for PROFINET.

For connecting to Ethernet, the RF182C communication module features a connection block in one of the following designs:

- Connection block in M12 design, either with
  - 5-pin 7/8” connector (standard) or
  - 4-pin 7/8” connector (option)
- Push-pull connection block design, RJ45
The following figure shows the basic design of the RF182C.

**Figure 2-1  Basic design of the RF182C**
Potential

Ungrounded installation of a system is possible with the RF182C. The following circuit shows the internal relationships of the reference potentials.

24 V supply, RF182C and reader

Internal supply, reader interface

Ethernet
Auxiliary voltage for bus connection

Schirm

Figure 2-2 Galvanic isolation of RF182C
Integration

The following figure shows how the RF182C with connection block M12, 7/8" is integrated in an automation system. The push-pull connection block is integrated in the same manner as the connection block M12, 7/8".

Figure 2-3   RF182C configurator with connection block M12, 7/8"
Mounting

The RF182C communication module is designed for easy assembly.

3.1 Mounting position, mounting dimensions

Mounting position

There are no restrictions regarding the mounting position for the RF182C.

Mounting dimensions and spacing

Table 3- 1 Mounting dimensions of base module with connection block M12, 7/8” (without connector)

<table>
<thead>
<tr>
<th>Designation</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting width</td>
<td>60 mm</td>
</tr>
<tr>
<td>Mounting height</td>
<td>210 mm</td>
</tr>
<tr>
<td>Mounting depth</td>
<td>54 mm</td>
</tr>
</tbody>
</table>

Table 3- 2 Mounting dimensions of base module with push-pull connection block (without connector)

<table>
<thead>
<tr>
<th>Designation</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting width</td>
<td>60 mm</td>
</tr>
<tr>
<td>Mounting height</td>
<td>216 mm</td>
</tr>
<tr>
<td>Mounting depth</td>
<td>100 mm</td>
</tr>
</tbody>
</table>
3.2 Mounting the I/O module

Features

- The base unit is mounted on a stable surface

---
**Note**

**Functional ground (PE)**

If a grounded metal mounting surface is used, the bottom mounting screw of the RF182C module already establishes a reliable grounding connection. This eliminates the need for a separate grounding cable. If you use the fixing screw as grounding connection, the thread of the fixing screw or the contact facing of the fastening nut on the base must be unpainted. This ensures a low-resistance connection.

---

Requirements

Screws:

<table>
<thead>
<tr>
<th>Screw type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5 cylindrical head screw to ISO 1207/ISO 1580 (DIN 84/DIN 85)</td>
<td>The screw should be at least 20 mm long. You will also need washers according to DIN 125.</td>
</tr>
<tr>
<td>Cylindrical head screw with M5 hexagonal recessed hole according to DIN 912</td>
<td></td>
</tr>
</tbody>
</table>

Required tools

Medium-sized cross-head screwdriver or 8 mm socket wrench.
Procedure

Fix the base unit onto a level surface using the screws. The base unit must be screwed to the surface (3 Nm tightening torque) at both fixing points (front, top and bottom).

Figure 3-1 Mounting the I/O module
3.3 Mounting the connection block

Features
The connection block connects the RF182C with the Ethernet and supplies the base unit with voltage.

Requirements
The base unit is already mounted

Required tools
Cross-head screwdriver, medium.

Mounting the connection block
1. Plug the connection block into the base unit
2. Screw the connection block onto the base unit (torque 1 to 1.3 Nm) Tighten the screws evenly, working in cross-wise passes. 4 screws are already located in the connection block (see Figure).

Figure 3-2 Plug the connection block M12, 7/8” onto the base unit and screw it on

Figure 3-3 Plug the push-pull connection block onto the base unit and screw it on

**Note**

IP65, IP66 or IP67 degree of protection only exists when the connection block is screwed to the base unit.
3.4 Replacing labels

Features

You can use the labels to mark every channel on the base unit and the connection block. The labeling strips are supplied with clipped on label.

- 2 labels on the base module
- 1 label on connection block M12, 7/8"
- 2 labels on push-pull connection block

Requirements

If you want to replace the labels, you can reorder them. You will find the order number in section Ordering data (Page 109).

Required tools

Screwdriver, size 2.5 mm to 4 mm.

Replacing labels

1. Push the screwdriver into the small opening of the label, and then lever it out.

2. With your finger push the new label into the holder of the module.
3.5 Disassembling the RF182C

Procedure

The RF182C is wired up and operating.
1. Switch off the supply voltage for the RF182C.
2. Disconnect the wiring on the connection block.
3. Remove the 4 fixing screws from the connection block and pull the connection block off the base unit.
4. Disconnect the wiring on the base unit.
5. Remove the fixing screws from the base unit.

Note
See also section Loop-through of Ethernet and supply voltage (Page 30).
Mounting

3.5 Disassembling the RF182C
Connecting

Proper use

When connecting non-specified devices to the RF182C, it is possible that the connected device may be destroyed.

**NOTICE**

The device must **not** be connected to the public telephone network without a HUB / Switch because the voltage intervals are designed for 500 V.

Ethernet setups

Ethernet communication can be established in BUS or STAR topology. Also note the information in section Loop-through of Ethernet and supply voltage (Page 30).

![RF182C with BUS topology](image_url)

Figure 4-1 RF182C with BUS topology
Connecting RF182C communication module

Operating Instructions, 10/2010

PROFINET IO
Master module
SIMATIC S7

PC

Switch (e.g. SCALANCE)

RF182C

RF182C

24 V DC

PROFINET IO  [Industrial Ethernet]

Figure 4-2  RF182C with STAR topology
Reader/SLG connection system

One reader/SLG always occupies one M12 connection socket on the RF182C. A pre-assembled cable therefore provides the optimum easy connection for the reader/SLG. The connection cable is 2 m long in the standard version.

Figure 4-3   Overview of wiring
4.1 Wiring connection block M12, 7/8"

Features

- Connect the supply voltages and Ethernet to the connection block M12, 7/8":
  - M12 connection in D coding: Ethernet
  - 7/8" connection: Supply voltages
- You can loop the supply voltages and Ethernet through via the second M12 or 7/8" circular socket connectors.

Requirements

- Wire connection block M12, 7/8" when the supply voltage is switched off.

Required tools

- Stripping tool, screwdriver for wiring the M12 and/or 7/8" connector if you are not using a pre-assembled cable.

Accessories required

- Pre-assembled cable with connector
- If you are not using a pre-assembled cable:
  - M12: 4-core Ethernet cable (Twisted Pair), shielded and M12 connector, 4-pole, D coding (see Table Pin assignment of M12 connector, 4-pole, D coding (Ethernet))
  - 7/8": 5-core cable and 7/8" connector (see Table Pin assignment for 7/8" connector (supply voltages))
- For order numbers, refer to Section Ordering data.

Wiring M12, 7/8" connector

The tables below contain the pin assignment for the M12 and 7/8" connectors:

Table 4-1 Pin assignment of M12 connector, 4-pole, D coding (Ethernet)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Assignment</th>
<th>View of M12 connector, 4-pole, D coding (wiring side)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data line TxP</td>
<td>Data line RxP</td>
</tr>
<tr>
<td>2</td>
<td>Data line RxP</td>
<td>Data line TxP</td>
</tr>
<tr>
<td>3</td>
<td>Data line TxN</td>
<td>Data line RxN</td>
</tr>
<tr>
<td>4</td>
<td>Data line RxN</td>
<td>Data line TxN</td>
</tr>
</tbody>
</table>

Infeed and loop-through of Ethernet X3, X4

Any connector can be used for infeed and looping through

Ethernet cable (Twisted Pair)
Connecting

4.1 Wiring connection block M12, 7/8”

Table 4-2  Pin assignment of 7/8” connector, 4-pole (supply voltages)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Assignment</th>
<th>View of 7/8” connector (wiring side)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load voltage ground (2M)</td>
<td>Supply X01</td>
</tr>
<tr>
<td>2</td>
<td>Ground for electronics/encoder supply (1M)</td>
<td>2L+</td>
</tr>
<tr>
<td>3</td>
<td>Functional ground (PE)</td>
<td>1L+</td>
</tr>
<tr>
<td>4</td>
<td>Electronics/encoder supply (1L+) (voltage supply for RF182C and reader/SLG)</td>
<td>5-core cab</td>
</tr>
<tr>
<td>5</td>
<td>Load voltage supply (2L+) (unused on RF182C)</td>
<td>Loop-through connection X02</td>
</tr>
</tbody>
</table>

Table 4-3  Pin assignment of 7/8” connector, 4-pole (supply voltages)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Assignment</th>
<th>View of 7/8” connector (wiring side)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1L+ electronics/encoder supply (power supply for RFID 181EIP and reader/SLG)</td>
<td>1L+</td>
</tr>
<tr>
<td>2</td>
<td>unused</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>unused</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Ground for electronics/encoder supply (1M)</td>
<td></td>
</tr>
</tbody>
</table>

Note
When connecting the supply voltage, we recommend the cables specified in section Ordering data (Page 109) (5 x 1.5 mm² pre-assembled cable with 7/8” connectors).

If you want to assemble the cable yourself, then the conductor cross-section should be 1.5 mm².
Connecting M12, 7/8" connectors

1. Press the connector (M12 or 7/8") into the relevant round socket on the connection block. Ensure that the correct stop is provided between the connector and bush (groove and spring).

2. Use the knurled locking ring to secure the connector.

Sealing unused sockets

Always close all unused sockets using M12 or 7/8" seal caps in order to achieve the degree of protection IP65, IP66 or IP67. See section Ordering data (Page 109) for order numbers.
4.2 Wiring of the push-pull connection block

Features
- Connect the power supplies and Ethernet to the push-pull connection block:
  - Push-pull connection (RJ45), D-coded: Ethernet
  - Push-pull connection: Supply voltages
- You can loop through the supply voltages and Ethernet via the second push-pull connection.

Requirements
- Wire the push-pull connection block with the supply voltage switched off.

Required tools
- Screwdriver
- Stripping tool for wiring the push-pull cable connector if you assemble your own cables.

Accessories required
- Pre-fabricated cables with push-pull cable connector for 1L+/2L+ and RJ45. The cables are available in various lengths from appropriate manufacturers.
- If you assemble your own cables:
  - 5-core cable and push-pull cable connector for 1L+/2L+
  - 4-core, shielded cable (bus cable) and push-pull cable connector for RJ45

Note
Refer to the manufacturer's documentation if you are fabricating the cables with the push-pull cable connectors.
### Wiring of push-pull connectors

The tables below contain the pin assignment for the push-pull connectors:

**Table 4-4 Pin assignment of push-pull cable connectors (RJ45)**

<table>
<thead>
<tr>
<th>View of push-pull cable connectors (RJ45)</th>
<th>Terminal</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>X03 PN1</td>
<td>X04 PN2</td>
<td>X03 PN1 for infeed from Ethernet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X04 PN2 for loop-through from Ethernet</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Transmit Data+ TD Receive Data+ RD</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Transmit Data- TD_N Receive Data- RD_N</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Receive Data+ RD Transmit Data+ TD</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Ground GND (RJ45)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Ground GND (RJ45)</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Receive Data- RD_N Transmit Data- TD_N</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Ground GND (RJ45)</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Ground GND (RJ45)</td>
</tr>
</tbody>
</table>

**Table 4-5 Pin assignment of push-pull cable connectors (1L+ and 2L+ supply voltages)**

<table>
<thead>
<tr>
<th>View of push-pull cable connectors (1L+ and 2L+ supply voltages)</th>
<th>Terminal</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>X01 DC 24 V for infeed</td>
<td>X02 DC 24 V for looping through</td>
<td></td>
</tr>
<tr>
<td>1 Electronic/encoder supply 1L+ground</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Ground for electronic/encoder supply 1M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 2L+ load voltage supply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Ground for load voltage supply 2M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Functional ground (PE)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Connecting

4.2 Wiring of the push-pull connection block

Note

When connecting the power supply, we recommend the cables specified in section Ordering data (Page 109) (5 x 1.5 mm² pre-assembled with push-pull connectors).

If you want to assemble the cable yourself, then the conductor cross-section should be 1.5 mm².

A cable cross-section of 2.5 mm² is mandatory for an amperage > 8 A.

Connecting push-pull cable connectors

Plug the push-pull cable connectors for 1L+/2L+ and RJ45 into the associated sockets (see figure below). Ensure that the locking mechanism between the connector and socket is properly applied. The connectors must engage.

Sealing unused sockets

Cover all unused push-pull sockets with caps in order to achieve degree of protection IP65, IP66, or IP67. Refer to section Ordering data for order numbers.
4.3 Loop-through of Ethernet and supply voltage

Features

The connection block features one connector for the incoming supply and one socket for loop-through connection of the supply voltage. The connector and the socket for the supply are linked with one another internally.

Two sockets are available for the infeed and loop-through of Ethernet. The sockets are not connected to each other in the connection block. The switch in the base unit creates the logical connection.

Note

If you disassemble the connection block during operation, only the power supply will be looped through. Data communication to subsequent devices will be interrupted from this module onwards.

CAUTION

The IP65, IP66 or IP67 degree of protection is no longer guaranteed when the connection block is dismounted.
Notes for wiring

- If you are wiring your structure, then you must take into account the impact of cable length on supply voltage to the RF182C.

  **Example:**
  When using a 10 m long cable with a diameter of 1.5 mm², the voltage drop is 2.5 V with a loading of 10 A. This corresponds to 0.25 V at a 1 A load.

- The maximum infeed current for connection block M12, 7/8” is 6 A at 1L+ and 8 A at 2L+. These values must not be exceeded.

- The maximum infeed current of the push-pull connection block is 12 A for 1L+ and 2L+ at up to 40 °C and 8 A for 1L+ and 2L+ at up to 60 °C. These values must not be exceeded.

- Adhere to the current carrying capacity of the connected cables, which depends on the conductor material, the conductor cross-section and the ambient temperature.

  **CAUTION**
  If you do not observe the maximum infeed current and the cable cross-section required, this may result in the cable isolation and contacts overheating and to the device being damaged.

  **CAUTION**
  **Damage**
  A cable cross-section of 2.5 mm² is mandatory for an amperage > 8 A!
4.4 Wiring an RF182C to a controller with Ethernet connection

A connection from Ethernet to an M12 connection can be easily implemented.

Self-assembly of an Ethernet M12 cable

- You will need a pre-assembled PROFINET/Ethernet cable with M12 connectors at both ends twice the required length. You will also need two Ethernet connectors for self-assembly. Cut the M12 cable in the center and connect one Ethernet connector to each free cable end. This will result in two Ethernet M12 cables.

- You will need the following individual parts: Ethernet plug-in connector, M12 plug-in connector, and PROFINET/Ethernet standard cable (unassembled). The parts can be found in the ordering data. You can make up a cable to your own length requirements using these parts.

Using a cabinet bushing Ethernet M12

This connection variant must always be used when the controller electronics is installed in a cabinet. The following figure shows the connection layout.

![Diagram of Ethernet M12 connection](image-url)

Figure 4-7  Cabinet bushing
4.5 Connecting the RF182C to functional ground (PE)

Features

- You have to connect the RF182C to the functional ground (PE). For this purpose, a grounding screw for one grounding cable is provided on the communication module.
- If a grounded metal mounting surface is used, the bottom mounting screw of the RF182C module already establishes a reliable grounding connection. This eliminates the need for a separate grounding cable.
- The connection to functional ground (PE) is also required to deflect the interference currents and for electromagnetic compatibility.

Requirements

- Always make sure there is a low-resistance connection to the functional ground (PE).
- If you use the fixing screw as grounding connection, the thread of the fixing screw or the contact facing of the fastening nut on the base must be unpainted. This ensures a low-resistance connection.

Required tools (only if grounding via the grounding cable is required).

- Screwdriver
- Stripping tool
- Crimp tool

Required accessories (only if grounding via the grounding cable is required).

- M5 x 10 grounding screw and washers
- Grounding cable (copper braided cable) with minimum cross-section of 4 mm²
- Cable lug
Connecting the RF182C to functional ground (PE)

<table>
<thead>
<tr>
<th>Standard grounding via the fixing screw</th>
<th>Optional grounding via a grounding cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Mount the module on the grounded, metallic base as described in section</td>
<td>1. Isolate the grounding cable and secure the cable lug.</td>
</tr>
<tr>
<td>Mounting the I/O module.</td>
<td>2. Screw the cable lug on to the communication module</td>
</tr>
<tr>
<td>① Grounded, metallic base</td>
<td>(M5 grounding screw). The tightening torque is 3 Nm.</td>
</tr>
<tr>
<td>② Unpainted thread or nut base</td>
<td></td>
</tr>
</tbody>
</table>

1. Mount the module on the grounded, metallic base as described in section Mounting the I/O module.

① Grounded, metallic base
② Unpainted thread or nut base
5.1 Address assignment for Ethernet

The reader is connected to Ethernet via the RF182C communication module. Communication between the application in the PC (client) and the reader (via the RF182C as server) only functions with a unique address assignment:

MAC-ADD

The physical address, MAC-ADD (Media Access Address), is defined by the manufacturer for each RF182C.

Example MAC-ADD: 67-89-AB-CD-EF-01
You will find the MAC-ADD printed on the side of the RF182C.

IP address

In addition, each RF182C requires a logical address, an IP address (Internet protocol), which is used to address it on the network.

An IP address may be present only once within a network. It must be parameterized in the RF182C. In the user application in the PC, the IP address is specified when establishing a connection.

The IP address always comprises 32 bits and is represented in decimal format (value range from 0 to 255). It therefore comprises a string of four numbers in ASCII format which are each separated by a point.

Example of an IP address: "157.163.170.12";

Subnet mask

The subnet mask is required to specify the network. The subnet mask is similar to the IP address. It comprises four numbers which are each separated by a point (default value: 255.255.255.0).

Example of a subnet mask "255.255.0.0"

Socket

A socket is a communication end point that is defined by an IP address and a port.
Parameterizing

5.1 Address assignment for Ethernet

**Port**

A port is an access point that can be addressed by means of a specific function on the device. For the RF182C, for example, the reader is addressed through port numbers.

**Note**

**Port assignment via web server**

The assignment of the ports can be changed via the web server. If no connection could be established with the RF182C, check the port setting via the web server. The default port setting is: 10001/10002.

You can then establish a connection via the set port numbers.

---

**Figure 5-1 TCP/IP connection**

For more detailed information on socket programming, see section Socket programming requirements (Page 81).
5.2 Data communication between client and RF182C

Basic sequence

- The module has run up and has not been parameterized yet.

---

**Note**

**Connection problems?**

If no connection could be established with the RF182C, check the communication settings (IP address, port number) of the communication module via the web server (Page 41).

---

- Optionally, a configuration message frame (comDevSetConfig) can be sent to the RF182C to change the communication mode stored in the RF182C by default. The client that configures the RF182C first also defines the RF182C settings.

Further configurations during operation are not possible. The module must be de-energized or restarted by means of a "Reset" via the web server so that it can be reconfigured.
If no configuration message frame is sent and the default port setting was not changed via the web server, the RF182C is operated in the default setting. The default values are shown in the following figure:

![Diagram](image)

---

**Parameter** | **Value**
---|---
Baud rate | 115.2 kBaud
Port 10001 | Reader 1 assignment (see chapter Web server (Page 41))
Port 10002 | Reader 2 assignment (see chapter Web server (Page 41))
Asynchronous message frames (alarm/presence message) are assigned to the opened port. | 
LED suppression | NONE
Mode | U/D/RF300/RF600

---

**NOTICE**

**SLG D11S and D12S cannot run with the standard baud rate**

Please note that the MOBY D readers SLG D11S and D12S cannot run if the standard baud rate is set.

If you want to operate these readers with the communication module, you must first parameterize the communication module and set the baud rate to 19.2 kBaud.
● The RESET message frame created by the user is sent to the corresponding reader.
● The process continues with a command message frame depending on the application.
● After longer periods without message frames (approx. 3 s), the application (client) can automatically send a heartbeat frame (line monitoring) to test the connection. The RF182C communication module then acknowledges the message frame. You yourself must ensure that the connection is monitored and define the interval after which the client should automatically send a heartbeat message frame. If, in case of an error, no response is sent in answer to the heartbeat message frame from the RF182C, the client must then initiate further actions (disconnect/connect/parameterize the reader).
● The connections (including TCP/IP) can be canceled by both sides due to the following causes:
  – Inactivity (timeout, keep-alive on TCP level)
  – Connection error
  – Disconnection request
● After disconnection by the server, the client must reconnect, send a RESET command, etc.

**Data communication**

The graphic below shows the principle of data communication between application (client) and RF182C:

![Data communication principle](image)
5.3 Factory setting of the RF182C

Each RF182C is assigned a unique device ID (MAC address) before it leaves the factory. The communication module is addressed via the IP address during configuration and programming.

Therefore, you must first assign the IP address data (IP address and subnet mask) to the RF182C so that it can be used in the Ethernet network. An IP address can be assigned to the RF182C using the PST tool or via the web server.

Factory setting

- Default IP address setting: 192.168.0.100
- Default port setting (default:
  - 10001 Reader 1
  - 10002 Reader 2

Use the "Primary Setup Tool" (http://support.automation.siemens.com/WW/view/de/19440762) software (V4-0 or higher) to assign an IP address to the communication module.
5.4 Assigning the IP address

5.4.1 Overview

There are two ways of assigning an IP address to the RF182C communication module:

- Using the "Primary Setup Tool V4-0"
- Via the web server of the communication module

Both alternative procedures are described in brief below.

5.4.2 Web server

Procedure

1. Enter the IP address of the communication module in the address field of your browser.
The web server of the communication module opens.

**Note**

**No contact with web server RF182C**

If the web server of the communication module does not open, you should make sure that all cables are correctly connected and check whether the RF182C communication module has powered up.

2. Check the IP address of the PC and the address of the subnet mask in the "Communication" menu, "Parameter" tab.

**Note**

If a connection is established, make sure that an IP address of the same subnet is assigned to the PC, laptop computer or PLC, unless a router is used. The IP address of the communication module and PC must have the same subnet mask.
3. In the "Settings" tab you will find the settings that are currently valid on the RF182C device. Here you can change the IP address, subnet mask, default router or the port numbers of the individual readers.

**NOTICE**

**Termination of communication**

If you change the settings during ongoing operation, communication will be terminated. The application must then reconnect again with the new settings form the RF182C communication module. The same applies to the web server.

At the next startup of the communication module, the default settings are no longer active. The changed settings are now active.
Resetting to factory settings

Via the "Reset" menu, you can reset all settings to the factory settings.

1. To do this, click on the button "Reset RF182C to factory settings".

Note

Soft reset

If serious communication errors occur, you can also execute a so-called "soft reset" using the "Reset 182C" button. The communication module will then restart as if the power was turned off and back on again.
5.4.3 Primary Setup Tool

Procedure

1. Open the "Primary Setup Tool V4-0" via "Start > SIMATIC > Primary Setup Tool".
2. Select the network card of the PC in the menu under "Settings" and click the "Search" button.

A window opens that indicates that a device was found in the network.

If no RF182C communication module is shown, make sure that all cables are connected properly and check if the RF182C communication module has started.

Check the IP address of the PC and the subnet mask.

Note

If a connection is established, make sure that an IP address of the same subnet is assigned to the PC, laptop computer or PLC, unless a router is used. The IP address of the communication module and PC must have the same subnet mask.
3. Click the "+" sign next to the folder icon to display the settings of the communication module:

![Figure 5-4 Settings of the communication module](image)

- If you want to change the data under "Assign IP Parameters", check the adjacent option.
- Click the button "Assign Name" to explicitly assign a name to the device, e.g. RF182C.

4. If you want to transfer the changed data to the communication module, select the higher-level folder and select "Module > Download" in the menu. Confirm the subsequent window with "Yes":

<table>
<thead>
<tr>
<th>NOTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Waiting time</strong></td>
</tr>
</tbody>
</table>
Wait until the IP address has been updated. Activate the search function by clicking the button 💾 to view the change.
5.4 Assigning the IP address

Result

You can now address the device using a browser or user program.

Note

IP address stored on connection block

The IP address is stored on the connection block. Therefore, if you replace the base module, no new IP address has to be assigned.
5.5 Troubleshooting: Assigning the IP address

If you are having problems when assigning an IP address to the SIMATIC RF182C communication module, proceed according to the checklist below:

**Procedure**

1. Connect the RF182C directly or via the hub/switch on your PC/notebook/PLC. Do not connect any other module/device to the network. Do **not** switch on the RF182C communication module yet.

2. Remove all other network cables from your PC/notebook and make sure that the RF182C is the only network device connected to your PC/notebook.

3. Now switch on the RF182C. Pay attention to the following LEDs:
   - The error LEDs 1 and 2 on the device should flash every 3 seconds after run up.
   - The "SF" LED should be lit.
   - The "ON" LED should be lit.
   - The "DC 24 V" LED should be lit.
   - The "BF" LED should be flashing.
   - One of the two green link LEDs should be lit (green). One or two "RxTx" LEDs should flicker depending on the communication load in the network.

4. Start the "Primary Setup Tool V4-0" software and configure the network settings if you have not already done so.

5. Click "Search" to update the view.
   The communication module should now be visible in the "Primary Setup Tool V4-0" software.

6. In the software, click the "+" sign next to the folder icon of the communication module.

7. Check "Assign IP parameter".
   - Enter a valid IP address.

8. To transfer the data, select the higher-level folder of the communication module and select "Module > Load" in the menu. Confirm the subsequent window with "Yes".

9. The IP address should be assigned to the communication module the next time the "Primary Setup Tool V4-0" inquires.
Communication interface

6.1 Overview of commands

<table>
<thead>
<tr>
<th>Name command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comDevSetConfig</td>
<td>Optional: Reconfigure RF182C</td>
</tr>
<tr>
<td>reset</td>
<td>Reset and parameterize RF182C and reader of a channel</td>
</tr>
<tr>
<td>writeTagData</td>
<td>Write to tag address</td>
</tr>
<tr>
<td>readTagData</td>
<td>Read from tag address</td>
</tr>
<tr>
<td>initializeTag</td>
<td>Initialize the tag</td>
</tr>
<tr>
<td>getReaderStatus</td>
<td>Status of the connected readers</td>
</tr>
<tr>
<td>setAnt</td>
<td>Antenna on/off</td>
</tr>
<tr>
<td>heartbeat</td>
<td>Line monitoring</td>
</tr>
<tr>
<td>getTagStatus</td>
<td>Status of the tag</td>
</tr>
</tbody>
</table>

Note

Text file for structure of XML commands

The structure of the XML commands of this section can also be found in a text file (RF182C_XML_Commands.txt). You can find this file on the RFID CD "Software & Documentation". Follow the link "CM > ASM > RF182C > Tools".

Thus you can transfer the basic structure of the commands to your application program by means of copy and paste.

Note

Structure of the XML commands

Please note that a value must always be entered between an opening and a closing XML tag. Otherwise, the message frame will be acknowledged with the error 3550.

Example:

<baud rate>115200</baud rate>

If an XML tag pair does not contain a value, omit the complete pair.
6.2 Configuration parameters of the RF182C

The RF182C is already configured at the factory. In most cases this command can therefore be omitted in the application.

XML command

```xml
<command>
  <comDevSetConfig>
    <signature>RF182C</signature>
    <protocolVersion>Version</protocolVersion>
    <parameter>
      <transmissionPoint>
        <mode>Mode</mode>
        <baudrate>Baud rate</baudrate>
        <startupLedSupression>LEDSupression</startupLedSupression>
      </transmissionPoint>
    </parameter>
  </comDevSetConfig>
</command>
```

XML response

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <comDevSetConfig/>
</reply>
```
## 6.2 Configuration parameters of the RF182C

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>ASCII</td>
<td>0 ... 9, A ... Z&lt;br&gt;Here you must enter the following version of the protocol as string:&lt;br&gt;• V1.0</td>
</tr>
<tr>
<td>RF182C</td>
<td>ASCII</td>
<td>Here you must enter the following values:&lt;br&gt;• RF182C</td>
</tr>
<tr>
<td>Baud rate</td>
<td>ASCII</td>
<td>Here you must enter one of the following baud rates (in baud):&lt;br&gt;• 19200&lt;br&gt;• 57600&lt;br&gt;• 115200</td>
</tr>
<tr>
<td>LEDSuppression</td>
<td>ASCII</td>
<td>Both channels, or channel 1 or channel 2 are flashing&lt;br&gt;• NONE = run-up flashing is not suppressed by any of the two reader terminals&lt;br&gt;• TRANSM_POINT_1 = run-up flashing of the err_LED at the reader 1 terminal is being suppressed&lt;br&gt;• TRANSM_POINT_2 = run-up flashing of the err_LED at the reader 2 terminal is being suppressed</td>
</tr>
<tr>
<td>Mode</td>
<td>ASCII</td>
<td>U/D/RF300_DIRECT_ADDRESSING</td>
</tr>
</tbody>
</table>

**Note**

The baud rate and mode cannot be set individually; they apply for both channels.
6.3 Input parameters of the RF182C

The RESET command is always required after switching on/run-up of the module. It contains the settings for the reader connected to the RF182C.

Below the XML command for the RESET command is described:

**XML command**

```xml
<command>
  <reset>
    <param>Param</param>
  </reset>
</command>
```

**XML response**

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <reset/>
</reply>
```

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>ASCII hex</td>
<td>00 ... FF (ASCII string of a length of 16 ASCII characters; corresponding to 8 hexadecimal numbers) e.g. 00 25 00 00 01 00 00 (presence activated RF300, no multitag) ¹</td>
</tr>
</tbody>
</table>

¹ The ASCII string may not contain any blank. Blanks have been inserted in the example for better readability.

A byte-by-byte breakdown of the RESET command can be found in section Command and acknowledgement telegrams (Page 111).
6.4 Commands of the communication module

6.4.1 writeTagData

With this command you can write to subareas or the complete tag insofar as there are address ranges in linear order physically on the tag.

Below the XML command for the write command (writeTagData) is described:

XML command

```xml
<command>
  <writeTagData>
    <startAddress>Address</startAddress>
    <data>Data</data>
  </writeTagData>
</command>
```

Information on the memory sizes of the tags can be found in section Addressing of the RFID tags (Page 123).

XML response

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <writeTagData/>
</reply>
```

Note

The length of the data to be written is derived from the number of characters transferred in the variable <data>. Please note that in each case 2 ASCII characters of the transferred data flow are converted into a hex character on the tag in the communication module.

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 to maximum length of the (user data - 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The user data is written to the tag from this start address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Note that this parameter depends on the tag. For detailed information on the address, refer to the respective MOBY System Manual or Appendix B Addressing of the RFID tags (Page 123).</td>
</tr>
<tr>
<td>Data</td>
<td>ASCII hex</td>
<td>00...FF (ASCII string (max. 128 KB ASCII))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• User data to be written to the tag.</td>
</tr>
</tbody>
</table>
6.4 Commands of the communication module

6.4.2 readTagData

With this command you can read subareas or the complete tag insofar as there are address ranges in linear order physically on the tag.

Below the XML command for the read command (readTagData) is described:

**XML command**

```xml
<command>
  <readTagData>
    <startAddress>Address</startAddress>
    <dataLength>Datalength</dataLength>
  </readTagData>
</command>
```

Information on the memory sizes of the tags can be found in section Addressing of the RFID tags (Page 123).

**XML response**

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <readTagData>
    <returnValue>
      <data>Data</data>
    </returnValue>
  </readTagData>
</reply>
```

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 to maximum length of the (user data -1); the user data is read from</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the tag starting from this address. (Address + data length) must be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smaller than the end address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Note that this parameter depends on the tag. For detailed information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on the address, refer to the respective MOBY System Manual.</td>
</tr>
<tr>
<td>Datalength</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The contents of Datalength refer to the number of bytes to be read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from the tag. Twice the number of characters is transmitted in the &quot;Data&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>field of the XML response.</td>
</tr>
<tr>
<td>Data</td>
<td>ASCII hex</td>
<td>max. 128 KB ASCII per command</td>
</tr>
</tbody>
</table>
6.4.3 *initializeTag*

With this command the complete user memory area of the memory is deleted or overwritten with a defined value.

Below the XML command for initializing tags (*initializeTag*) is described:

**XML command**

```xml
<command>
  <initializeTag>
    <memorySize>MemorySize</memorySize>
    <defaultValue>Value</defaultValue>
  </initializeTag>
</command>
```

**XML response**

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <initializeTag/>
</reply>
```

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemorySize</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Memory size of the tag to be initialized</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Note that this parameter depends on the tag. For detailed information on the memory size, refer to the respective MOBY System Manual or Appendix B Addressing of the RFID tags (Page 123).</td>
</tr>
<tr>
<td>Value</td>
<td>ASCII hex</td>
<td>00 ... FF (2 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Hex value that is written to the tag.</td>
</tr>
</tbody>
</table>
6.4.4 **getReaderStatus**

The command requests diagnostic/status information from the connected reader. Below the XML command for the reader status (`getReaderStatus`) is described:

**XML command**

```xml
<command>
  <getReaderStatus>
    <mode>Mode</mode>
  </getReaderStatus>
</command>
```

**XML response**

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <getReaderStatus>
    <returnValue>
      <data>Data</data>
    </ returnValue>
  </getReaderStatus>
</reply>
```

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>ASCII hex</td>
<td>00 ... FF (2 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01 = reader status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>02 = MOBY U (SLG diagnostics I, function calls)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>03 = MOBY U (SLG diagnostics II, error messages)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>04 = MOBY U (SLG diagnostics III, identified MDS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>05 = MOBY U (SLG diagnostics IV, communication performance)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>06 = RF300 reader diagnostics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>07 = RF600 reader diagnostics</td>
</tr>
<tr>
<td>Data</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (ASCII string (max. 400 x ASCII))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An acknowledgment message frame on the reader status is returned.</td>
</tr>
</tbody>
</table>

**Reader status acknowledgement message frame**

An acknowledgment message frame on the reader status is returned via the "Data" parameter. Different acknowledgment message frames are returned depending on the set "mode". All available acknowledgment message frames can be found in section Command and acknowledgement telegrams (Page 111).
6.4.5 getTagStatus

The command requests different status information on a tag in the field.

Below the XML command for the tag status (getTagStatus) is described:

XML command

```xml
<command>
  <getTagStatus>
    <mode>Mode</mode>
    <week>Week</week>
    <year>Year</year>
  </getTagStatus>
</command>
```

Note

Only with MOBY U

The italic XML tags <week> and <year> only apply for the MOBY U system. These italic tags must be omitted for other MOBY systems!

XML response

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```xml
<reply>
  <resultCode>Errorcode</resultCode>
  <getTagStatus>
    <returnValue>
      <data>Data</data>
    </returnValue>
  </getTagStatus>
</reply>
```

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>ASCII hex</td>
<td>00 = MOBY U 01 = RF300: Type and write protection status 02 = RF300: Diagnostic data 03 = RF300/MOBY D: Type and write protection status for ISO tags 04 = RF600: Diagnostic data</td>
</tr>
<tr>
<td>Week</td>
<td>ASCII hex</td>
<td>00 ... FF (2 x ASCII)</td>
</tr>
<tr>
<td>Year</td>
<td>ASCII hex</td>
<td>00 ... FF (2 x ASCII)</td>
</tr>
<tr>
<td>Data</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (ASCII string: max. 400 x ASCII)</td>
</tr>
</tbody>
</table>
Tag status acknowledgement message frame

An acknowledgment message frame on the tag status is returned via the "Data" parameter. Different acknowledgment message frames are returned depending on the set "mode". All available acknowledgment message frames can be found in section Command and acknowledgement telegrams (Page 111).

6.4.6 setAnt

Below the XML command for switching the antenna(s) on/off (setAnt) is described:

XML command

```
<command>
  <setAnt>
    <mode>Mode</mode>
  </setAnt>
</command>
```

XML response

Below the XML response without error entry (for error entries, see section Error messages (Page 71)) is described:

```
<reply>
  <resultCode>Errorcode</resultCode>
</setAnt/>
</reply>
```

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
</table>
| Mode      | ASCII     | D/U/RF300: 01 = switch antenna on 02 = standby; switch antenna off
|           |           | RF600: 00 = Antenna 1 off and Antenna 2 off 01 = Antenna 1 on and Antenna 2 off 02 = Antenna 1 off and Antenna 2 on 03 = Antenna 1 and 2 on |
6.4.7 heartbeat

The heartbeat command has no parameters. Evaluation of the resultCode in the XML response shows if the connection to the communication module is still functioning properly. A missing connection between the communication module and reader causes an alarm and/or an error message.

Below the XML command for monitoring the connection (heartbeat) is described:

**XML command**

```xml
<command>
  <heartbeat/>
</command>
```

**XML response**

Below is the XML response without error entry

```xml
<reply>
  <resultCode>0000</resultCode>
  <heartbeat/>
</reply>
```

**See also**

Error messages (Page 71)
6.5 Asynchronous message frames

6.5.1 tagPresent

The "tagPresent" notification will only be signaled if this is activated accordingly beforehand in the RESET message frame. If "tagPresence" is activated, this message frame is sent when a tag enters or exits the field of a reader.

XML message frame

```xml
<notification>
  <id>Sequencenumber</id>
  <origin>Origin</origin>
  <tagPresent>
    <tagCount>TagCount</tagCount>
  </tagPresent>
</notification>
```

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence number</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The sequence number is a number that is automatically set to 0000 after</td>
</tr>
<tr>
<td></td>
<td></td>
<td>switching on the RF182C. After sending a notification on the same</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP/IP channel, the sequence number is increased by 1. After FFFF,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this number is also set to 0000 again.</td>
</tr>
<tr>
<td>Origin</td>
<td>ASCII</td>
<td>• TRANSM_POINT_1 - Channel1/Reader1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• TRANSM_POINT_2 - Channel2/Reader2</td>
</tr>
<tr>
<td>TagCount</td>
<td>ASCII hex</td>
<td>4 x ASCII</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000 = no tag present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0001 = there is a tag in the field of the reader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0002 = there are two tags in the field of the reader and so on, up to max. 00FF</td>
</tr>
</tbody>
</table>
6.5.2 alarm

If no command is pending, the RF182C communication module sends an alarm.

XML message frame

```xml
<alarm>
  <id>Sequencenumber</id>
  <origin>Origin</origin>
  <deviceName>DeviceName</deviceName>
  <deviceTime>Time</deviceTime>
  <content>
    <code>Errorcode</code>
  </content>
</alarm>
```

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence number</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The sequence number is a number that is automatically set to 0000 after switching on the RF182C. After sending an alarm on the same TCPIP channel, the sequence number is increased by 1. After FFFF, this number is also set to 0000 again.</td>
</tr>
<tr>
<td>Origin</td>
<td>ASCII</td>
<td>0 .. 9, A ... Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source channel TRANSM_POINT_1 or TRANSM_POINT_2</td>
</tr>
<tr>
<td>DeviceName</td>
<td>ASCII</td>
<td>Name of the device (default: RF182C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The device name can be up to 256 bytes long.</td>
</tr>
<tr>
<td>Time</td>
<td>ASCII hex</td>
<td>00000000...FFFFFFFF (8 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time of the alarm in milliseconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This time is reset to 0000 when the RF182C is switched on. It cannot be set by the user. It is therefore a relative time</td>
</tr>
<tr>
<td>Error code</td>
<td>ASCII hex</td>
<td>0000 ... FFFF (4 x ASCII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The error code specifies the cause that triggered this alarm message frame. More information on the error codes can be found in section Error messages of the RF182C (Page 72).</td>
</tr>
</tbody>
</table>
7.1 Replacing the RF182C communication module

Initial situation

- The RF182C communication module is already mounted. A new RF182C communication module of the same type should be installed.
- The RF182C is wired up and operating.

Procedure

1. Remove the 4 fixing screws from the connection block and pull the connection block off the communication module.

   Note
   
   If you disassemble the connection block during operation, only the power supply will be looped through. Ethernet communication will be interrupted during module replacement from this node onwards. For more information, refer to section Loop-through of Ethernet and supply voltage (Page 30).

2. On the communication module, remove the screwed M12 plug-in connections to the readers.

3. Remove the fixing screws from the communication module and remove it.

4. Locate the new communication module and screw it down firmly.

5. Place the connection block on the new communication module and tighten the 4 fixing screws.

Result

Since the IP address of the communication module remains saved in the connection block, the new RF182C communication module is included in the data communication by the Ethernet controller.

Note

If the connection block is replaced in addition to the base unit, the RF182C may not start up automatically. In this case, proceed as follows:
What should I do if the RF182C can no longer be addressed

If the connection block is replaced in addition to the base unit, it is possible that the RF182C can no longer be addressed. This is indicated by a permanently lit or flashing BF LED.

In this case, check the network configuration. Load (e.g. using the PST tool) the required network parameters into the RF182C.

Check the diagnostic messages via the web server or check the settings of the IP address or the port number setting.

See also

Parameterizing (Page 35)
7.2 Firmware update

The firmware for the RF182C communication module can be updated via the Ethernet interface. You can start the firmware update via the web server of the communication module.

Preconditions

- The communication module is connected to the PC via Ethernet.
- Exit all applications before you start the firmware update.

Procedure

1. Save the update file (e.g. RF182C_V_2_0_0.elf™), which you received from Siemens, in the desired directory.
2. Enter the IP address of the communication module in the address field of your browser. The web server of the communication module opens.
3. Click the "Durchsuchen" button.
4. Select update file (RF182C_V_2_0_0.elf).
5. Start the firmware update via the "Start Update" button.

7.3 Reader update

In preparation.
8.1 Diagnostics using LEDs

The following figure shows details of the LEDs of the RF182C.
**Table 8-1** Status LEDs for the RF182C

<table>
<thead>
<tr>
<th>LEDs</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>Lights up when the RF182C has completed start-up without errors.</td>
</tr>
<tr>
<td>24 V DC</td>
<td>Lights up when the 24 V supply voltage is connected to the RF182C.</td>
</tr>
<tr>
<td>ACT_1, ACT_2</td>
<td>Reserved</td>
</tr>
<tr>
<td>ERR_1, ERR_2</td>
<td>A flashing pattern indicates the last error to occur. These flashing patterns are described in Section Error messages (Page 71).</td>
</tr>
<tr>
<td>PRE_1, PRE_2</td>
<td>Indicates the presence of a tag/MDS.</td>
</tr>
<tr>
<td>RxD_1, RxD_2</td>
<td>Indicates live communication with the reader / SLG. May also indicate malfunctions on the reader / SLG.</td>
</tr>
</tbody>
</table>

**Table 8-2** LED display for Ethernet diagnosis

<table>
<thead>
<tr>
<th>BF</th>
<th>SF</th>
<th>Cause of error</th>
<th>Error handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>–</td>
<td>• Communication module is in start-up mode.</td>
<td>Check the Ethernet cable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No cable inserted</td>
<td></td>
</tr>
<tr>
<td>Flashes</td>
<td>On</td>
<td>• There is no TCP connection to the Ethernet controller.</td>
<td>• Check your Ethernet configuration using the PST tool.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No client has connected.</td>
<td></td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>• There is an error.</td>
<td>• Send a reset or comDevSetConfig command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The module has not received a reset or comDevSetConfig command on one channel.</td>
<td></td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>• Normal mode</td>
<td>–</td>
</tr>
</tbody>
</table>

= Status not relevant

**Table 8-3** LEDs on connection block

<table>
<thead>
<tr>
<th>Link (green)</th>
<th>Tx / Tx (yellow)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Off</td>
<td>No physical connection over Ethernet.</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>Physical connection over Ethernet, no data traffic</td>
</tr>
<tr>
<td>On</td>
<td>Flashes</td>
<td>Physical connection over Ethernet with data traffic</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>Temporary state following switch-on</td>
</tr>
</tbody>
</table>

The table is equally applicable to the left and right Ethernet connection.
Other communication module operating modes are indicated by the PRE, ERR, ACT, SF and ON LEDs:

<table>
<thead>
<tr>
<th>ON</th>
<th>BF</th>
<th>SF</th>
<th>PRE_1</th>
<th>ERR_1</th>
<th>ACT_1</th>
<th>PRE_2</th>
<th>ERR_2</th>
<th>ACT_2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Start-up active</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>LED test on start-up (start Ethernet)</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Internal fault</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>Checksum error of the firmware</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Slow flashing</td>
<td>Off</td>
<td>Slow flashing</td>
<td>Off</td>
<td>Firmware update (flashes with every described area)</td>
<td></td>
</tr>
<tr>
<td>On</td>
<td>Flashes</td>
<td>On</td>
<td>Off</td>
<td>Flashes 1 x acc. to table 8-1</td>
<td>Off</td>
<td>Off</td>
<td>Flashes 1 x acc. to table 8-1</td>
<td>Off</td>
<td>Run-up cannot be parameterized. No client has connected.</td>
</tr>
</tbody>
</table>
Diagnostics

8.1 Diagnostics using LEDs
Error messages

9.1  Response without error entry

Below the XML response without error entry is described:

XML response

<pre>
<reply>
 <resultCode>0000</resultCode>
 <Name of the output command, e.g. reset/>
</reply>
</pre>

9.2  Response with error entry

Below the XML response with error entry is described:

XML response

<pre>
<reply>
 <resultCode>Errorcode</resultCode>
 <Name of the output command, e.g. reset/>
</reply>
</pre>

The following table describes the possible error codes (resultCodes). The error codes are coded in 4 bytes.
## 9.3 Error messages of the RF182C

### Table 9-1  Error messages of the RF182C via the "resultCode" variable

<table>
<thead>
<tr>
<th>Error code</th>
<th>Flashing of ERR LED</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0000       | –                   | No error
|            |                     | Default value if everything is ok. |
| 0001       | 2x                  | Presence error: The MDS has moved out of the write/read device's transmission window. The MOBY command was executed only partially. |
|            |                     | Read command: No data is being transmitted to the client. |
|            |                     | Write command: The MDS which just left the field contains an incomplete data set. |
|            |                     | • Distance between write/read device and MDS not adhered to |
|            |                     | • Configuration error: The data set to be processed is too large (in dynamic mode) |
|            |                     | The next command is automatically executed on the next MDS. A read or write command is possible. |
|            |                     | • With timeout: No MDS in field |
| 0002       | 2x                  | Presence error: An MDS has passed by a write/read device without being processed by a MOBY command. |
|            |                     | This error message is not reported immediately. Instead, the communication module is waiting for the next command (read, write). This command is immediately replied to with this error. This means that a read or write command is not processed. The next command is executed normally again by the communication module. |
|            |                     | A RESET command from the client also resets this error state. |
|            |                     | Bit 2 is set in the reset parameter option_1, or a reset command was sent and there is no MDS in the transmission window. |
| 0003       | 3x                  | Error in the connection to the write/read device. Write/read device does not answer. |
|            |                     | • The cable between communication module and SLG is wired incorrectly or there is a cable break |
|            |                     | • The 24 V supply voltage is not connected or is not on or has failed briefly |
|            |                     | • Automatic fuse on the communication module has blown |
|            |                     | • Hardware defective |
|            |                     | • Another SLG is in the vicinity and is active |
|            |                     | • Interference injection on SLG line |
|            |                     | • Execute a RESET command after error correction |
| 0004       | 4x                  | Error in MDS's memory |
|            |                     | The MDS has never been write-accessed or has lost the contents of its memory due to battery failure. |
|            |                     | • Replace MDS (if battery bit is set). |
|            |                     | • Install MDS with the STG. |
|            |                     | • Reinitialize MSD (see Section "Command parameter settings"). |
## Error messages

### 9.3 Error messages of the RF182C

<table>
<thead>
<tr>
<th>Error code ASCII hex</th>
<th>Flashing of ERR LED</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0005                 | 5x                  | Unknown command  
The client is sending an uninterpretable command to the communication module.  
• Check the XML command  
• The MDS reported an address error |
| 0006                 | 6x                  | Field interference on write/read device  
The write/read device is receiving interference from its environment.  
• The distance between two write/read devices is too small and does not correspond to the configuration guidelines  
• The connecting cable to the write/read device is defective or too long or does not comply with the specification  
• MOBY U: MDS has left the field during communication.  
• MOBY U: Communication between write/read device and MDS was terminated by interference (e.g. person/foreign body moving between write/read device and MDS). |
| 0007                 | 7x                  | Too many transmit errors  
The MDS was not able to correctly receive the command or the write data from the communication module even after several attempts.  
• The MDS is positioned exactly on the boundary of the transmission window.  
• Data transmission to the MDS is being affected by external interference. |
| 0008                 | 8x                  | CRC sending error  
• The receiver monitor has detected at least one fault during transmission.  
  – Cause same as error 0006  
• MDS signaling CRC error frequently.  
  – The MDS is positioned exactly on the boundary of the write/read device.  
  – The hardware of the MDS and/or write/read device is defective. |
| 0009                 | 9x                  | Only during initialization: CRC error during acknowledgment receipt from MDS  
• Cause same as error 0006 |
| 000A                 | 10x                 | Only during initialization: MDS is unable to perform the initialization command.  
• MDS is defective. |
| 000B                 | 11x                 | MOBY U: Memory of MDS cannot be read correctly. |
| 000C                 | 12x                 | Memory of the MDS cannot be write-accessed.  
• Memory of the MDS is defective.  
• EEPROM MDS was written too frequently and has reached the end of its service life |
| 000D                 | 13x                 | Address error  
The address area of the MDS was exceeded.  
• Check the XML command  
• The MDS is not the right type.  
• RF300: Attempted write access to write-protected areas (OTP area) |
## Error messages

### 9.3 Error messages of the RF182C

<table>
<thead>
<tr>
<th>Error code</th>
<th>ASCII hex</th>
<th>Flashing of ERR LED 1°</th>
<th>Description</th>
</tr>
</thead>
</table>
| 000E       | 000E      | 14x                    | ECC error (only possible when ECC_mode = TRUE)  
  The data could not be read by the MDS.  
  - Data of the MDS have been lost (MDS defective).  
  - The MDS was not initialized with ECC driver.  
    - Initialize MDS  
    - MDS with EEPROM has reached the end of its service life. The data have been lost.  
    - Replace the MDS  
  - → The MDS was moved out of the transmission window while being write-accessed  
    - The MDS is not positioned correctly  
    - → Command to the communication module was issued incorrectly by user |
| 000F       | 000F      | 1x                     | Run-up message from a reader that is connected to the communication module  
  - Carry out a RESET |
| 0011       | 0011      |                        | Short circuit or overload of the 24 V outputs (error code, presence)  
  - The affected output is turned off.  
  - All outputs are turned off when total overload occurs  
  - A reset can only be performed by turning the 24 V voltage off and on again.  
  - Then start RESET |
| 0012       | 0012      | 18x                    | Internal communication module communication error.  
  - Connector contact problem on the communication module  
  - Defective communication module hardware  
    - Return communication module for repair  
  - Start RESET after error correction |
| 0013       | 0013      | 19x                    | The communication module/SLG U does not have enough buffer storage to store the command intermediated. |
| 0014       | 0014      | 20x                    | Internal communication module/SLG error.  
  - Program execution error on the communication module  
  - Turn power of communication module off and on again.  
  - Start RESET after error correction  
  - MOBY U: Watchdog error on write/read device |
| 0015       | 0015      | 21x                    | Wrong parameterization of the communication module/SLG  
  - Check the parameterization  
  - RESET command is parameterized incorrectly  
  - After a start-up, the communication module has still not received a RESET |
| 0016       | 0016      | 22x                    | The communication module is unable to process the command.  
  - Client command (e.g. READ) issued with too much user data |
| 0017       | 0017      | 23x                    | Communication error  
  - Check the client command which causes this error  
  - Start RESET after error correction |
| 0018       | 0018      |                        | An error has occurred which must be acknowledged with a RESET.  
  - The RESET command is faulty.  
  - Transmit RESET after error correction |
### Error messages of the RF182C

<table>
<thead>
<tr>
<th>Error code</th>
<th>Flashing of ERR LED</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0019       | 25x                 | Previous command is active or buffer overflow<br>The user sent a new command to the communication module although the last command was still active.  
- Active command can only be terminated with RESET.  
- Two client calls were parameterized with the same parameters  
- Start RESET after error correction  
- When command repetition is used, no data is fetched from the MDS. The data buffer on the communication module has overflowed. MDS data have been lost. |
| 001C       | 28x                 | The antenna of the write/read device is turned off. An MDS command to the communication module was started in this state.  
- Turn on the antenna with the command “antenna on/off.”  
- The antenna is turned on (off) and has received an additional turn-on (turn-off) command. |
| 001D       | –                   | More MDSes are in the transmission window than the SLG is capable of processing simultaneously.  
- Only 1 MDS can be processed at a time with a client |
| 001E       | 30x                 | Error when processing the function  
- Communication module defective: The communication module receives wrong data during a RESET |
| 001F       | –                   | Running command canceled by RESET  
- Communication with the MDS was terminated by RESET  
- This error can only be reported if there is a RESET |

1) The flashing ERR-LED can be implemented either on the communication module or on the reader.
### Error messages

#### 9.3 Error messages of the RF182C

<table>
<thead>
<tr>
<th>Error code</th>
<th>Flashing of ERR LED</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3214       | -                   | Fatal Error. Internal firmware error of the communication module  
|            |                     | - Turn the power supply of the communication module off and on again  
|            |                     | - Update the firmware to a new version, if applicable. |
| 3221       | -                   | Configuration required  
|            |                     | - Check configuration data or send it to communication module |
| 3222       | -                   | Conflicting configurations of the communication module.  
|            |                     | - The configuration module was already configured or is in default mode. |
| 3223       | -                   | Configuration faulty  
|            |                     | - Check configuration data to communication module |
| 3321       | -                   | Internal processing error  
|            |                     | - Turn the power supply of the communication module off and on again  
|            |                     | - Update the firmware to a new version, if applicable |
| 3322       | -                   | Buffer overflow in the communication module  
|            |                     | - The user is sending too many consecutive commands.  
|            |                     | - The communication module cannot transfer data to the user quickly enough or the user cannot receive the data quickly enough. |
| 3323       | -                   | Notification buffer overflow  
|            |                     | - Too many consecutive notification message frames that communicate results are received too quickly (presence).  
|            |                     | - The communication module cannot transfer data to the user quickly enough or the user cannot receive the data quickly enough. |
| 3324       | -                   | Alarm buffer overflow  
|            |                     | - Too many consecutive results are received too quickly via the alarm buffer (error messages).  
|            |                     | - The communication module cannot transfer data to the user quickly enough or the user cannot receive the data quickly enough. |
| 3325       | -                   | Internal data processing error in the communication module  
|            |                     | - Start RESET command. |
| 3326       | -                   | Error at the internal reader interface of the communication module  
|            |                     | - Start RESET command. |
| 3417       | -                   | Length error when receiving data via the TCP/IP connection.  
|            |                     | - Check if the TCP/IP data transmission over the line is executed properly |
| 3421       | -                   | Error in the configuration of the connection  
|            |                     | - Connection was not configured  
|            |                     | - Permissible number of connections was exceeded  
|            |                     | - Check and if necessary correct configuration. |
| 3422       | -                   | Error when sending data via the TCP/IP connection  
|            |                     | - Check if the TCP/IP data transmission over the line is executed properly |
| 3423       | -                   | Timeout when receiving data via the TCP/IP connection.  
<p>|            |                     | - Increase the data transmission rate of a command from the user side. |</p>
<table>
<thead>
<tr>
<th>Error code</th>
<th>Flashing of ERR LED</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3424       | -                   | Error when receiving the TCP channel  
• Connection was aborted  
• TCP/IP protocol error  
• Permissible number of connections at one port was exceeded  
• Check socket programming on the user side or find the fault with network analyzer (Wireshark). |
| 3425       | -                   | Receiving buffer overflow  
• Previous command is still active  
• Check if maximum possible number of commands was exceeded. |
| 3521       | -                   | Incorrect UID  
• The length of the UID is not correct  
• A UID is required that has not been transferred by the user  
• Check UID |
| 3523       | -                   | Conversion error  
• Check if individual characters of the user data of a command exceed the permissible range for ASCII hex. (Range 0 – 9 or A – F ). The number of characters must be an even number. |
| 3550       | -                   | Error in the XML structure of a command message frame.  
• Check command structure |

**Note**

**Error handling procedure**

If error messages occur, execute a reset command to eliminate the error.

If this procedure does not eliminate the error, disconnect the module from the power supply.
9.4 Diagnostics via Web server

9.4.1 Saving/reading of I&M data records

Via Identification&Maintenance data records you can store internal information on the module and retrieve it as required.

Save data records

Via the "Identification" menu in the "Settings" tab, various settings can be made and stored on the communication module. Use the "transmit and save" button to save the settings.
9.4 Diagnostics via Web server

Read data records

To read out the data records from the communication module, switch to the "Identification" menu in the "Readout" tab.

9.4.2 Communication status query

You can query the communication status of the RF182C communication module via the menu "Communication" in the Connection "tab". You can see the status of the user connection.
9.4.3 Event and message frame overview

Event overview

Via the "Diagnostic" menu in the "Events" tab, you can query the events of the module.

Message frame overview

Via the "Diagnostic" menu in the "Traffic" tab, you can query the last twenty message frames of the communication module.
Examples/applications

10.1 Basic principles of socket programming, exemplary in C

10.1.1 Socket programming requirements

Definition
A socket is the end point of a communication connection. All modern operating systems nowadays have a socket application interface. This software interface permits access to the TCP/IP stack on the operating system used.

Requirements

- An operating system/programming language must be used that supports network programs with sockets.
- Sockets are supported by the following operating systems: Microsoft Windows 95/98/ME/2000/XP/Vista, Linux, Unix. Other operating systems must be considered with regard to the usability of network programs.
- Sockets are used by many programming languages (e.g. C, C++, C#, Delphi, VB, Java). Other programming languages must be considered with regard to the usability of network programs.
10.1 Basic principles of socket programming, exemplary in C

10.1.2 Basic client/server principle

![Diagram of Basic client/server principle]

10.1.3 Important basic commands

<table>
<thead>
<tr>
<th>Commands/functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socket()</td>
<td>Initialization and parameter transfer of a new socket</td>
</tr>
<tr>
<td>Bind()</td>
<td>Assignment of a user (IP address and port number) to a socket</td>
</tr>
<tr>
<td>Lists()</td>
<td>Ready to connect</td>
</tr>
<tr>
<td>Connect()</td>
<td>Function for connecting the client to the server via TCP/IP</td>
</tr>
<tr>
<td>Accept()</td>
<td>Wait function of server until a client connects</td>
</tr>
<tr>
<td>Send()</td>
<td>Sending data</td>
</tr>
<tr>
<td>Recv()</td>
<td>Receiving data</td>
</tr>
<tr>
<td>Close()</td>
<td>Closing the socket after completing data transmission. When working with the RF182C, a Close usually only occurs when the system is shut down.</td>
</tr>
</tbody>
</table>

Note

No parameters or return values of the individual functions are listed here since they may differ in the different operating systems/programming languages.

The names of the functions are examples only, they may differ in the operating systems/programming languages. These function examples are limited to the basic principles of the client/server representation – only a partial example of the functions is shown.
10.1.4 Partial programming example of a client in C/Windows operating system

Under Windows, the header file "winsock.h" or the library "wsock32.lib" must also be integrated.

These must be initialized before calling the Windows sockets.

/*Init Windows Sockets*/

{
    WSADATA wsadata;
    if( WSAStartup( MAKEWORD( 1,1 ), &wsadata ) == 0 )
    {
        /*Initialization successful*/
    }
    else
    {
        /*Error during initialization*/
    }
}

/*Extract from main function*/

SOCKET Client;            // variable for socket handle
SOCKADDR_IN adr;          // variable for storing the target information
char caBuf[ 1500 ];       // the XML command or XML result is stored in this variable.
int nLen;                 // Length of the send/receive data

/*Initialization of target information prior to connecting*/

adr.sin_family = AF_INET; // Selection of the address family/Internet //connection-oriented
adr.sin_port =           // Assignment of port number:10001
adr.sin_addr.s_addr =    // Assignment of IP address: 192.168.0.100
/* Creation of a socket - function returns the handle from the socket
SOCK_STREAM - connection-oriented protocol TCP*/
Client = socket( AF_INET, SOCK_STREAM, 0 );
/*Connection establishment*/

if( connect(Client, (SOCKADDR*)&adr, sizeof( adr ) ) < 0)  
{    //Error has occurred
} else  
{    //Connection OK
}

/*Sending data*/
nLen = send( Client, caBuf, nLen, 0 );
if( nLen > 0 )  
{    // Data was sent successfully
} else  
{    // Error on sending
}

/*Receiving data*/
nLen = recv( Client, caBuf, sizeof( caBuf ) - 1, 0 );
if( nLen > 0 )  
{    // Data was received
    // The data can be read in the caBuf array
} else  
{    // Error occurred while receiving
    // Connection termination if applicable
}

/*Close connection*/
closesocket( Client );
10.2 RF182C user application

To permit user-friendly operation, there is an application available for the PC. The application is created in C# and can run direct on any Windows PC. You can load, modify, and expand the PC in your development environment. This application can be used as a basis for your application.

You can find the application on the RFID CD "Software&Documentation", Edition 2009 or later. On the user interface, follow the link "CM/ASM > RF182C > Demo". You can start the program directly.

Requirements

- Operating system: Windows XP
- Development environment: Microsoft Visual Studio 2008

Procedure

The application can be started direct via the user interface of the CD.

The sources of the application are stored in a zip file on the CD. You will find the file under "Data/Tools > Applications/RF182C".

Functions

This application offers the following functions:

- Establish connection
- Disconnect
- Notes on connection
- Input window for entering the commands
- Transfer of the commands to the RF182C communication module
- Display window for monitoring the communication process of the RF182C (acknowledgements and error messages)
10.2.1 User interface layout

1. IP address input window
2. Port number input window
3. Button for connecting or disconnecting
4. Note: Connected or not connected
5. Note: Presence or no presence
6. Check box: The output window scrolls automatically or does not scroll automatically
7. Output window for receive data
8. Button for sending data in the input window

Figure 10-2 User interface
10.2.2 Extracts example code of the user application in C#

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Web;

namespace RF182CApp
{
    public partial class MainForm : Format
    {
        // Socket to realize TCP/IP connection
        private Socket Connection = null;
        // connected (true) or not (false)
        private bool ConnectionState = false;
        // buffer for received data
        private String ReceiveBuffer = "";
        // Indicator for an error in asynchronous receive threads
        bool AsyncError = false;

        public MainForm()
        {
            InitializeComponent();
        }

        /* This routine is called when the window in initialized. */
        private void MainForm_Load(object sender, EventArgs e)
        {
            // We're not connected at the beginning
            SetConnectionState(false);
        }

        /* This routine is called when the Connect / Disconnect button
        * is clicked.
        */
        private void bConnect_Click(object sender, EventArgs e)
        {
            if (ConnectionState == true) // already connected -- close connection
            {
                /* Connection may be disposed already (i.e. if an error
                 * occurred before.
                 */
                if (Connection != null)
                {
                    // Close the socket
                    Connection.Close();
                }
            }
        }
    }
}
```
private void Enable(bool enableConnectionData, bool enableCommunicationData) {
    editIP.Enabled = enableConnectionData;
    editPort.Enabled = enableConnectionData;
    editMessage.Enabled = enableCommunicationData;
    buttonSend.Enabled = enableCommunicationData;
}

private void SetConnectionState(bool state) {
    if (state == true) {
        ConnectionState = true;
        // Visualization
        Enable(false, true);
        labelConnectionState.Text = "CONNECTED";
        labelConnectionState.BackColor = Color.Green;
        buttonConnect.Text = "Disconnect";
        // Start UpdateTimer
        UpdateTimer.Start();
    } else {
        ConnectionState = false;
        //Stop UpdateTimer
        UpdateTimer.Stop();
        //Now, we don't know anything about tags in the field
    }
}
SetTagDetectionState(TagDetectionState.UNDEFINED);
//There may be uncollected data in the buffer --> call ParseBuffer()
ParseBuffer();
// Visualization
Enable(true, false);
labelConnectionState.Text = "NOT CONNECTED";
labelConnectionState.BackColor = Color.Red;
buttonConnect.Text = "Connect";
AsyncError = false;
}
}

/* Connects a stream based TCP/IP socket as client. */
private bool Connect()
{
    try
    {
        //Collect port and IP from the window.
        int port = 0;
        if (Int32.TryParse(editPort.Text, out port) == false || port <= 0)
        {
            MessageBox.Show(editIP.Text + " is not a legal port");
            return false;
        }

        IPHostEntry hostEntry = null;
        // Get host related information.
        hostEntry = Dns.GetHostEntry(editIP.Text);

        // Loop through the AddressList to obtain the supported AddressFamily. This is to avoid
        // an exception that occurs when the host IP Address is not compatible with the address
        // family
        // (typical in the IPv6 case).
        for (IPAddress address in hostEntry.AddressList)
        {
            IPEndPoint ipe = new IPEndPoint(address, port);
            // stream based TCP/IP socket
            Socket tempSocket = new Socket(ipe.AddressFamily, SocketType.Stream, ProtocolType.Tcp);

            // the actual connect
            tempSocket.Connect(ipe);

            if (tempSocket.Connected)
            {
                //a connection was established successfully
                Connection = tempSocket;
                Connection.ReceiveTimeout = 25;
                break;
            }
        }
    }
    if (Connection == null) return false;
}
// Start asynchronous receive
ReceiveString s = new ReceiveString();
Connection.BeginReceive(s.buffer, 0, ReceiveString.BufferSize, 0, new AsyncCallback(ReceiveCallback), s);
    return true;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Connecting Failed");
    return false;
}

/* This routine is called if there is new data in the asynchronous receive available. The data is stored in ReceiveBuffer so that it can be collected by the synchronous timer UpdateTimer. */
private void ReceiveCallback(IAsyncResult res)
{
    try
    {
        if (ConnectionState == true)
        {
            // Collect data
            int size = Connection.EndReceive(res);
            ReceiveString s = (ReceiveString)res.AsyncState;
            // Without a lock we might cause race situations
            lock (ReceiveBuffer)
            {
                // Store data in buffer
                ReceiveBuffer += Encoding.ASCII.GetString(s.buffer, 0, size);
            }

            // Start new asynchronous receive
            ReceiveString rs = new ReceiveString();
            Connection.BeginReceive(rs.buffer, 0, ReceiveString.BufferSize, 0, new AsyncCallback(ReceiveCallback), rs);
        }
    }
    catch (Exception ex)
    {
        // An error occurred --> report it
        AsyncError = true;
        MessageBox.Show(ex.Message);
    }
}

// This routine is called when the Send button is clicked
private void buttonSend_Click(object sender, EventArgs e)
{
//only do something if there is a message specified
if (editMessage.Text != "")
{
    // convert to byte array
    byte[] buffer = Encoding.ASCII.GetBytes(editMessage.Text.ToCharArray());
    try
    {
        // Send!
        int count = Connection.Send(buffer);
        // Did we send everything?
        if (count != buffer.Length)
        {
            MessageBox.Show("Sending failed!");
            SetConnectionState(false);
        }
        //Show the message in the list.
        AppendOutMessage(editMessage.Text);
    }
    catch (Exception ex)
    {
        SetConnectionState(false);
        //An error occurred
        MessageBox.Show(ex.Message);
    }
    //Empty the editbox
    editMessage.Text = "";
}

/* This routine is called when UpdateTimer ticks. This happens
 * synchronous to the thread owning the dialog, hence we can
 * manipulate the list.
 * */
private void UpdateTimer_Tick(object sender, EventArgs e)
{
    if (AsyncError == true)
    {
        //An error occurred!
        SetConnectionState(false);
    }
    else
    {
        ParseBuffer();
    }
}

/* Parses ReceiveBuffer for complete XML telegrams and appends
 * them to the list.
 * */
private void ParseBuffer()
{
    // Without a lock we might cause race situations
// (ReceiveCallback is asynchronous)
lock (ReceiveBuffer)
{
    for (XMLTag tag = FirstTag(); tag != null; tag = FirstTag())
    {
        // Extract the parsed message and append it
        String message = ReceiveBuffer.Substring(tag.startIndex, tag.length);
        GetInMessage(message);
        // Remove parsed message from ReceiveBuffer
        ReceiveBuffer = ReceiveBuffer.Substring(tag.startIndex + tag.length);
        // See if we got information about detected tags
        if (message.Contains("<tagCount>"))
        {
            if (message.Contains("<tagCount>0000</tagCount>"))
            {
                // There are no tags in the field
                SetTagDetectionState(TagDetectionState.NO);
            }
            else
            {
                // There are tags in the field
                SetTagDetectionState(TagDetectionState.YES);
            }
        }
    }
}

// Looks for the first complete XML telegram in ReceiveBuffer.
private XMLTag FirstTag()
{
    // Is there a reply, a notification or an alarm
    int index1 = ReceiveBuffer.IndexOf("<reply>");
    int index2 = ReceiveBuffer.IndexOf("<notification>");
    int index3 = ReceiveBuffer.IndexOf("<alarm>");

    if (index1 == Int32.MaxValue && index2 == Int32.MaxValue && index3 == Int32.MaxValue)
    {
        // No XML tag found
        return null;
    }

    if (index1 == -1) index1 = Int32.MaxValue;
    if (index2 == -1) index2 = Int32.MaxValue;
    if (index3 == -1) index3 = Int32.MaxValue;

    // Assume that the first tag is an alarm
    String endTag="</alarm>";
    XMLTag tag = new XMLTag();
    tag.type = "alarm";
    tag.startIndex = index3;

    // See if this is true and change it if necessary
    if (index1 < index2 && index1 < index3)
    {
// first tag is a reply
endTag = "</reply>";
tag.type = "reply";
tag.startIndex = index1;
}
else if (index2 < index3)
{
// first tag is a notification
endTag = "</notification>";
tag.type = "notification";
tag.startIndex = index2;
}

// Is the complete message in the buffer?
int endIndex = ReceiveBuffer.IndexOf(endTag, tag.startIndex);
if (endIndex == -1) return null;
tag.length = endIndex - tag.startIndex + endTag.Length;
return tag;
}

// Appends an outgoing message to the list
private void AppendOutMessage(String message)
{
// Remove all CR and LF
message.Replace("\r", "");
message.Replace("\n", "");
// Add message
int index = listProcess.Items.Add("OUT: " + message);
// Scroll if desired
if (checkScroll.Checked == true)
{
    listProcess.SelectedIndex = index;
}
}

// Appends an incoming message to the list
private void AppendInMessage(String message)
{
// Functionality just lies AppendOutMessage
message.Replace("\r", "");
message.Replace("\n", "");
int index = listProcess.Items.Add("IN : " + message);
if (checkScroll.Checked == true)
{
    listProcess.SelectedIndex = index;
}
}

/* Call this function if a tag appeared / disappeared. */
private void SetTagDetectionState(TagDetectionState state)
{ 
    switch (state) 
    { 
        case TagDetectionState.YES:
            labelTagDetected.Text = "TAG(s) DETECTED";
            labelTagDetected.BackColor = Color.Green;
            break;
        case TagDetectionState.NO:
            labelTagDetected.Text = "NO TAG DETECTED";
            labelTagDetected.BackColor = Color.Red;
            break;
        case TagDetectionState.UNDEFINED:
            labelTagDetected.Text = "";
            labelTagDetected.BackColor = BackColor;  //Color of the dialog
            break;
    }
}

/* As long as no tag presence notification was send, we have no
* information whether there is a tag in the field or not.
* --> three states
* */
enum TagDetectionState
{
    YES, NO, UNDEFINED
}

// This class is for handling asynchronous communication processes
internal class ReceiveString
{
    public read-only static int BufferSize = 512;
    public byte[] buffer = new byte[BufferSize];
}

// This class describes a XML block in ReceiveBuffer
internal class XMLTag
{
    //Index of first character
    public int startIndex;
    //Block length
    public int length;
    // Type ( reply, notification or alarm, since only telegrams
    // from the ASM are handled within this structure)
    public String type;
}
10.2.3 Functions of the RF182C applications

The following screenshots show the different functions of the RF182C application:

Example of working with the application

1. First enter the IP address and port number in the corresponding entry fields.

![RF182C not connected](image)

Figure 10-3 RF182C not connected
2. Then click "Connect".

![Connected successfully](image)

The connection to the RF182C has been established successfully. The send window is empty after sending successfully.
3. Enter a RESET command in the input window. Click the "Send" button.

![RESET command](image)

Figure 10-5  RESET command

The RF182C communication module operates in default mode.
4. The next window shows that the RESET command has been sent successfully. The RF182C has sent an acknowledgement with the error code "0000" (everything OK). It is indicated that a tag has been detected in the antenna field of the reader (0001).

![Figure 10-6 Tag detected](image)
5. The next window shows that the tag has exited the reader’s antenna field again (0000)

![Figure 10-7 Tag exited the field](image)

10.3 Example application for a PLC according to DIN IEC 61131

In preparation.
### Technical data

#### Ethernet Interface to the user

<table>
<thead>
<tr>
<th>Principle</th>
<th>Ethernet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical medium</td>
<td>Ethernet (TCP/IP communication)</td>
</tr>
</tbody>
</table>
| Duty type          | • 10BASE-T Full or Half Duplex  
|                    | • 100BASE-TX Fast Ethernet Full or Half Duplex |
| Transmission rate  | 10/100 Mbit/s |
| Plug-in connection | M12, 4-pin, D coding |
| Maximum cable length | 100 m |
| Cable type         | STP Cat 5 |
| Autonegotiation    | Yes |
| Autocrossing       | Yes |
| Switch function    | Yes, internal |

#### Serial Interface to the reader/SLG

| Connector          | 2 x M12 coupler plugs, 8-pin |
| Max. cable length  | 1000 m, dependent on Reader/SLG
|                    | (2 m = standard length; for other standard cables and self-assembled cables, refer to Section Connection cables) |
| Connectable readers/SLGs | 2x readers/SLG of the RFID families RF300, RF600, MOBY D/U |

#### Software functions

| Tag/MDS addressing | Direct access via addresses |
| Commands           | Initialize tag, read data from tag, write data to tag, etc. |

#### Supply voltage

| Rated value | 24 V DC |
| Permissible range | 20 V to 30 V DC |
| Current consumption without reader / SLG | max. 500 mA; typ. 100 mA |
| Current consumption through reader connection | Each 500 mA |
| Maximum infeed current in the connection block M12, 7/8" | 1L = 6 A  
| | 2L = 8 A |
| Maximum infeed current in the push-pull connection block | Up to 40 °C: |
| | 1L = 12 A  
| | 2L = 12 A |
| | Up to 60 °C: |
| | 1L = 8 A  
| | 2L = 8 A |

| Galvanic isolation | Yes |
## Technical data

<table>
<thead>
<tr>
<th><strong>Ambient temperature</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>During operation</td>
<td>0 to +60 °C</td>
</tr>
<tr>
<td>Transport and storage</td>
<td>-40 to +70 °C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Dimensions (W x H x D) in mm</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Base unit only</td>
<td>60 x 210 x 30</td>
</tr>
<tr>
<td>Base unit with connection block M12, 7/8”</td>
<td>60 x 210 x 54</td>
</tr>
<tr>
<td>Base unit with push-pull connection block</td>
<td>60 x 216 x 100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Weight</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Base unit</td>
<td>Approx. 210 g</td>
</tr>
<tr>
<td>Connection block M12, 7/8”</td>
<td>Approx. 230 g</td>
</tr>
<tr>
<td>Push-pull connection block</td>
<td>Approx. 120 g</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Mechanical Environmental Conditions</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting position</td>
<td>All mounting positions are possible</td>
</tr>
<tr>
<td>Vibration during operation</td>
<td>According to IEC 61131-2:</td>
</tr>
<tr>
<td></td>
<td>0.75 mm (10Hz to 58 Hz)</td>
</tr>
<tr>
<td></td>
<td>10 g (58 Hz to 150 Hz)</td>
</tr>
<tr>
<td>Shock resistance, shock during operation</td>
<td>Acc. to IEC 61131-2:</td>
</tr>
<tr>
<td></td>
<td>30 g</td>
</tr>
<tr>
<td><strong>Degree of protection</strong></td>
<td>IP67</td>
</tr>
</tbody>
</table>

### MTBF (Mean Time Between Failures) in years

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Base unit</td>
<td>121</td>
</tr>
<tr>
<td>Connection block</td>
<td>1100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Approvals</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cULus (file E116536)</td>
<td></td>
</tr>
<tr>
<td>FCC Code of Federal Regulations, CFR 47, Part 15, Sections 15.107 and 15.109 (Class A)</td>
<td></td>
</tr>
</tbody>
</table>

1) All supply and signal voltages must be safety extra low voltage (SELV/PELV according to EN 60950)
24 V DC supply: Safety (electrical) isolation of low voltage (SELV / PELV acc. to EN 60950)  
2) The power supply must provide the current required (max. 500 mA) during brief power failures of ≤ 20 ms.  
3) A cable cross-section of 2.5 mm² is mandatory for an amperage > 8 A.
12.1 Dimension drawing for RF182C with fixing holes

Dimension drawing of an RF182C with bus connection block M12, 7/8” PN PN

Figure 12-1 Dimension drawing of an RF182C with bus connection block M12, 7/8”
Dimension drawings

12.1 Dimension drawing for RF182C with fixing holes

Dimension drawing of an RF182C with push-pull bus connection block

Figure 12-2 Dimension drawing of an RF182C with push-pull bus connection block
13.1 Routing of standard cables

Available cables

- Connecting cable RF300/RF600; MOBY D only 6GT2602-0AB10-0AX0
- Extension cable for all RFID systems

Maximum cable length

The RF182C can be operated with any reader/SLG configuration with a maximum cable length of 50 m. Longer connecting cables of up to 1000 m are possible in some instances. The current consumption of the connected reader/SLG must however be taken into account. You will find information in the relevant system manuals. Sequential arrangement of more than two sub-sections to form a long section of cable should be avoided due to the additional contact resistances.
### Pin assignment

#### Table 13-1 Connecting cable M12 ↔ Reader / SLG

<table>
<thead>
<tr>
<th>M12 connector (male)</th>
<th>Reader/SLG connector (female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2</td>
<td>2</td>
</tr>
<tr>
<td>2 5</td>
<td>5</td>
</tr>
<tr>
<td>3 3</td>
<td>3</td>
</tr>
<tr>
<td>4 4</td>
<td>4</td>
</tr>
<tr>
<td>5 6</td>
<td>6</td>
</tr>
<tr>
<td>6 1</td>
<td>1</td>
</tr>
<tr>
<td>7 –</td>
<td></td>
</tr>
<tr>
<td>8 7</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 13-2 Connecting cable / extension cable M12 <-> M12

<table>
<thead>
<tr>
<th>M12 connector (male)</th>
<th>M12 connector (female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>2 2</td>
<td>2</td>
</tr>
<tr>
<td>3 3</td>
<td>3</td>
</tr>
<tr>
<td>4 4</td>
<td>4</td>
</tr>
<tr>
<td>5 5</td>
<td>5</td>
</tr>
<tr>
<td>6 6</td>
<td>6</td>
</tr>
<tr>
<td>7 7</td>
<td>7</td>
</tr>
<tr>
<td>8 8</td>
<td>8</td>
</tr>
</tbody>
</table>

#### Table 13-3 Connecting cable M12 ↔ sub-D 9-pin

<table>
<thead>
<tr>
<th>M12 connector (male)</th>
<th>Sub-D connector (female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 –</td>
<td>1, 8</td>
</tr>
<tr>
<td>2 5</td>
<td></td>
</tr>
<tr>
<td>3 7</td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td></td>
</tr>
<tr>
<td>5 2</td>
<td></td>
</tr>
<tr>
<td>6 6</td>
<td></td>
</tr>
<tr>
<td>7 –</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Note: Reader/SLG with Sub-D connector must be supplied over an additional connector with 24 V DC.
13.2 Self-assembled cable

A reader/SLG connector plug with screw terminals is provided for users who want to individually pre-assemble their own cables (refer to the relevant system manual). Cables and reader/SLG connector plugs can be ordered from the Catalog *FS 10 Sensors for Production Automation*.

Cable structure

You will need cables of the following specifications for self-assembled cables:

- 7 x 0.25 mm²
- LiYC11Y 7 x 0.25

Connectors

M12 connectors can be obtained from the relevant specialist dealers (e.g. Binder).

Pin assignment

The pin assignment is listed in the following table.

Table 13-4 Pin assignment

<table>
<thead>
<tr>
<th>M12 connector (male)</th>
<th>Pin</th>
<th>Signal</th>
<th>Core color</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1L+ (+ 24 V)</td>
<td>Note data sheet provided by cable manufacturer</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>−RxD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0 V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>RxD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>TxD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>−TxD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Free</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Functional ground (PE)/shield</td>
<td></td>
</tr>
</tbody>
</table>
13.2 Self-assembled cable
## Ordering data

### RF182C ordering data and accessories

<table>
<thead>
<tr>
<th>RF182C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RF182C communication module max. 2 SLGs or readers can be connected</td>
<td>6GT2002-0JD10</td>
</tr>
<tr>
<td>Connection block M12, 7/8” (5-pole)</td>
<td>6GT2002-1JD00</td>
</tr>
<tr>
<td>Connection block M12, 7/8” (4-pole)</td>
<td>6GT2002-4JD00</td>
</tr>
<tr>
<td>Push-pull connection block, RJ45</td>
<td>6GT2002-2JD00</td>
</tr>
<tr>
<td>Labels 20 x 7 mm (1 pack = 340 items)</td>
<td>3RT1900-1SB20</td>
</tr>
</tbody>
</table>

#### Accessories for connection block M12, 7/8” (5-pole)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6XV1870-8Axxx</td>
<td>IE plug-in cable for PROFINET/Ethernet (pre-assembled trailing cable with two M12 connectors, 4-pin, code D)</td>
</tr>
<tr>
<td>6XV1822-5Bxxx</td>
<td>7/8”-plug-in cable for supply voltage (5 x 1.5 mm²) (pre-assembled trailing power cable with two 5-pin 7/8” connectors)</td>
</tr>
<tr>
<td>6XV1830-8AH10</td>
<td>Trailing power cable (5 x 1.5 mm²) (not pre-assembled; length min. 20 m, length max. 1000 m)</td>
</tr>
<tr>
<td>6GK1905-0FA00</td>
<td>Connector plug 7/8” for supply voltage; (1 pack = 5 items) with pin insert</td>
</tr>
<tr>
<td>6GK1905-0FB00</td>
<td>Connector plug 7/8” for supply voltage; (1 pack = 5 items) with socket insert</td>
</tr>
<tr>
<td>6GK1901-1BB10-2AA0</td>
<td>RJ45 plug-in cable with metal casing and FC connection system, 180° cable outlet; (1 pack = 1 item)</td>
</tr>
<tr>
<td>6GK1901-0DB10-6AA0</td>
<td>M12 plug-in cable with metal casing and fast connection system, 180° cable outlet (D coded) ; (1 pack = 1 item)</td>
</tr>
<tr>
<td>3RX9802-0AA0</td>
<td>M12 covering caps</td>
</tr>
<tr>
<td>6ES7194-3JA00-0AA0</td>
<td>Covering caps 7/8” (1 pack = 10 items)</td>
</tr>
<tr>
<td>6XV1840-2AH10</td>
<td>PROFINET/Ethernet standard cable 2x2, Type A, unassembled; minimum order quantity 20 m</td>
</tr>
</tbody>
</table>

#### Accessories for connection block M12, 7/8” (4-pole)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cable for supply voltage pre-assembled with 7/8” 4-pole connectors (only for 6GT2002-4JD00) Not available from Siemens</td>
</tr>
</tbody>
</table>
## Accessories for push-pull connection block

<table>
<thead>
<tr>
<th>Item Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trailing power cable (5 x 1.5 mm²) (not pre-assembled; length min. 20 m, length max. 1000 m)</td>
<td>6XV1830-8AH10</td>
</tr>
<tr>
<td>Push-pull cable connector for 1L+/2L+, not pre-assembled</td>
<td>6GK1907-0AB10-6AA0</td>
</tr>
<tr>
<td>Push-pull cable connector for RJ45, not pre-assembled</td>
<td>6GK1901-1BB10-6AA0</td>
</tr>
<tr>
<td>Caps for push-pull sockets (1L+/2L+), 5 items per package, 1 item</td>
<td>6ES7194-4JA50-0AA0</td>
</tr>
<tr>
<td>Caps for push-pull sockets RJ45, 5 items per package, 1 item</td>
<td>6ES7194-4JD50-0AA0</td>
</tr>
<tr>
<td>PROFINET/Ethernet standard cable 2x2, Type A, unassembled; minimum order quantity 20 m</td>
<td>6XV1840-2AH10</td>
</tr>
</tbody>
</table>

## Accessories for RFID

<table>
<thead>
<tr>
<th>Item Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLG cable MOBY U; 2 m</td>
<td>6GT2091-0FH20</td>
</tr>
<tr>
<td>SLG cable MOBY U; 5 m</td>
<td>6GT2091-0FH50</td>
</tr>
<tr>
<td>SLG cable MOBY D; 2 m</td>
<td>6GT2691-0FH20</td>
</tr>
<tr>
<td>Reader cable RF300, extension cable RF300 / RF600 / MOBY I / E / U / D; 2 m</td>
<td>6GT2891-0FH20</td>
</tr>
<tr>
<td>Reader cable RF300, extension cable RF300 / RF600 / MOBY U / D; 5 m</td>
<td>6GT2891-0FH50</td>
</tr>
<tr>
<td>Reader cable RF300, extension cable RF300 / RF600 / MOBY U / D; 10 m</td>
<td>6GT2891-0FN10</td>
</tr>
<tr>
<td>Reader cable RF300, extension cable RF300 / RF600 / MOBY U / D; 20 m</td>
<td>6GT2891-0FN20</td>
</tr>
<tr>
<td>Reader cable RF300, extension cable RF300 / RF600 / MOBY U / D; 50 m</td>
<td>6GT2891-0FN50</td>
</tr>
<tr>
<td>Reader cable for RF300; connector on the reader is angled; 2 m</td>
<td>6GT2891-0JH20</td>
</tr>
<tr>
<td>RFID CD &quot;Software&amp;Documentation&quot;</td>
<td>6GT2080-2AA10</td>
</tr>
</tbody>
</table>

1) These cables are available in different lengths. See Catalog IK PI for more details
Command and acknowledgement telegrams

In this Section, you will find detailed information on some commands mentioned in Section Communication interface (Page 49). Only those commands requiring a special coding of commands and results are described.

Note
Special information on telegram expansions for the RF620R/RF630R readers can be found in the Appendix of the "Configuration Manual RF620R/RF630R".
RESET

Command telegram to MOBY-ASM

<table>
<thead>
<tr>
<th>standby</th>
<th>Param</th>
<th>00</th>
<th>dill</th>
<th>multitag</th>
<th>fcon</th>
<th>ftim</th>
</tr>
</thead>
</table>

- Maximum number of tags being processed in parallel in the field; permissible values: 00 01 hex
- MOBY U:
  - field_ON_time (see input parameter)
  - BERO operating mode
  - 00 hex = without BEROs; no write/read device synchronization
  - 01 hex = field_ON_time switches the field off
  - 02 hex = 1st BERO switches the field on;
  - 2. BERO switches the field off
  - 03 hex = write/read device synchronization via cable connection activated (see manual for configuring, mounting and service for MOBY U)
- MOBY D, RF300 not used (00 hex)

- RF300:
  - distance_limiting (see input parameters)
  - 02 hex = 0.5 W ... 10 hex = 4 W (default) ... 26 hex = 10 W
- Option 1: MOBY RF300 only

- MOBY U:
  - distance_limiting (see input parameters)
  - 05; 0A; 0F; 14; 19; 1E; 23 hex = 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; 3.5 m
  - 85; 8A; 8F; 94; 99; 9E; A3 hex = dito with reduced output power
- MOBY D: HF rating (see input parameters)
can only be set in 0.25 W steps for SLG D10S
- 02 hex = 0.5 W ... 10 hex = 4 W (default) ... 26 hex = 10 W
- RF300:
  - distance_limiting only for RF380R change of output power
  - 02 hex = 0.5 W
  - 03 hex = 0.75 W
  - 04 hex = 1.0 W
  - 05 hex = 1.25 W

- Option 2: MOBY RF300 only

- Presence check and MDS control (MDS_control)
  - 000 = no presence check
  - 001 = no MDS control; presence check via firmware (default)
GetReaderStatus mode = 01

Result message frame
mode = 01

| 01 | SLG status |

The meaning of the SLG status is described in the following table.

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>S-Info</td>
<td>HW</td>
<td>HW-V</td>
<td>LD-V</td>
<td>FW</td>
<td>FW-V</td>
<td>TR</td>
<td>TR-V</td>
<td>INT</td>
<td>baud</td>
<td>res</td>
<td>res</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>res</td>
<td>dili</td>
<td>multi</td>
<td>fcon</td>
<td>fon</td>
<td>sync</td>
<td>ant</td>
<td>stand_by</td>
<td>MDS control</td>
<td></td>
</tr>
</tbody>
</table>

Parameter name | Comment
---|---
S-Info | Reader status mode information = 01
HW | Hardware type
HW-V | Hardware version
LD-V | Version of loader
FW | Firmware type
FW-V | Firmware version
TR | Driver type
TR-V | Version of driver
INT | Interface (RS 232/RS 422)
Baud | Baud rate
dili | Range-capacity setting
multi | Multitag reader
fcon | field\_ON\_control: BERO mode
fon | Field\_on\_time:
  - MOBY U: BERO time
  - MOBY D: Tag type
  - RF300-ISO: Tag type
sync | Semaphore control (synchronization with reader)
  (RF300: res)
ant | Status of antenna
stand\_by | Time of standby after command execution
  (RF300: res)
MDS control | Presence mode

For detailed information, please refer to the respective reader description.
GetReaderStatus mode = 02

Result message frame
mode = 02

<table>
<thead>
<tr>
<th>02</th>
<th>1st command</th>
<th>nth command</th>
</tr>
</thead>
</table>

Message frame headers of the commands that were executed last

00 hex ... 21 hex = max. 33 message frames are transferred

GetReaderStatus mode = 03

Result message frame
mode = 03

<table>
<thead>
<tr>
<th>03</th>
<th>1st error</th>
<th>nth error</th>
</tr>
</thead>
</table>

Indicates last errors that occurred

00 hex ... EB hex = max. 235 error messages

GetReaderStatus mode = 04

Result message frame
mode = 04

<table>
<thead>
<tr>
<th>04</th>
<th>1st MDS</th>
<th>nth MDS</th>
</tr>
</thead>
</table>

ID's of the MDS last detected in the transmission window.

00 hex ... 18 hex = max. 24 MDS ID's

GetReaderStatus mode = 05

Result message frame
mode = 05

<table>
<thead>
<tr>
<th>05</th>
<th>(to be defined)</th>
</tr>
</thead>
</table>
**GetReaderStatus mode = 06**

Result message frame  
mode = 06

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8 ... 26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>S-Info</td>
<td>FZP</td>
<td>ABZ</td>
<td>CFZ</td>
<td>SFZ</td>
<td>CRCFZ</td>
<td>BSTAT</td>
<td>ASMFZ</td>
<td>res.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-Info</td>
<td>Reader status mode information = 06</td>
</tr>
<tr>
<td>FZP</td>
<td>Error counter, passive (errors during idle time)</td>
</tr>
<tr>
<td>ABZ</td>
<td>Abort counter</td>
</tr>
<tr>
<td>CFZ</td>
<td>Code error counter</td>
</tr>
<tr>
<td>SFZ</td>
<td>Signature error counter</td>
</tr>
<tr>
<td>CRCFZ</td>
<td>CRC error counter</td>
</tr>
<tr>
<td>BSTAT</td>
<td>Current command status</td>
</tr>
<tr>
<td>ASMFZ</td>
<td>Interface error counter for ASM</td>
</tr>
<tr>
<td>res.</td>
<td>Spare</td>
</tr>
</tbody>
</table>

The meaning of the diagnostics data is described in the following table.
GetReaderStatus mode = 07

Result message frame
mode = 07

The meaning of the SLG status
is described in the following table.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-Info</td>
<td>Reader status mode information = 07</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware type</td>
</tr>
<tr>
<td>HW-V</td>
<td>Hardware version</td>
</tr>
<tr>
<td>res0</td>
<td>Reserved</td>
</tr>
<tr>
<td>FW</td>
<td>Firmware type</td>
</tr>
<tr>
<td>FW-V</td>
<td>Firmware version</td>
</tr>
<tr>
<td>TR</td>
<td>Driver type</td>
</tr>
<tr>
<td>H</td>
<td>hours</td>
</tr>
<tr>
<td>Sec</td>
<td>Seconds</td>
</tr>
<tr>
<td>res1</td>
<td>Reserved</td>
</tr>
<tr>
<td>SLG-V</td>
<td>Reader version</td>
</tr>
<tr>
<td>Baud</td>
<td>Baud rate</td>
</tr>
<tr>
<td>res2</td>
<td>Reserved</td>
</tr>
<tr>
<td>dili_SLG</td>
<td>Set transmission performance</td>
</tr>
<tr>
<td>multi</td>
<td>Multitag reader</td>
</tr>
<tr>
<td>field_on_control</td>
<td>Set communication type</td>
</tr>
<tr>
<td>field_on_time</td>
<td>Set channel</td>
</tr>
<tr>
<td>expert</td>
<td>Expert mode</td>
</tr>
<tr>
<td>ant</td>
<td>Status of antenna</td>
</tr>
<tr>
<td>scanning_time</td>
<td>Radio communication profile</td>
</tr>
<tr>
<td>MDS control</td>
<td>Presence mode</td>
</tr>
</tbody>
</table>

For detailed information, please refer to the respective reader description.
GetTagStatus mode = 00

Result message frame
mode = 00

Byte 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Param UID MDS_type sum_sub sum_search week Year battery ST

Parameter name | Comment
--- | ---
UID | UID (tag number, EPC ID)
MDS_type | MDS type
sum_sub | Sum of subframe access
sum_search | Sum of searchmode
week | Date of last sleep-time change (week of year)
Year | Date of last sleep-time change (year)
battery | Battery left (percentage)
ST | Set sleep-time value on MDS

For detailed information, please refer to the respective reader description.
**GetTagStatus mode = 01**

Result message frame
mode = 01

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10 ... 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>status</td>
<td>UID 1 4</td>
<td>UID 5 8</td>
<td>Lock_state</td>
<td>res.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Tag status mode information = 01</td>
</tr>
<tr>
<td>UID 1 4</td>
<td>UID (tag number, EPC ID)</td>
</tr>
<tr>
<td>UID 5 8</td>
<td></td>
</tr>
<tr>
<td>Lock_state</td>
<td>EEPROM write protection status</td>
</tr>
<tr>
<td>res.</td>
<td>Spare</td>
</tr>
</tbody>
</table>

For detailed information, please refer to the respective reader description.
GetTagStatus mode = 02

Result message frame mode = 02

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12 ... 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>status</td>
<td>UID 1 4</td>
<td>UID 5 8</td>
<td>LFD</td>
<td>FZP</td>
<td>FZA</td>
<td>res.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Tag status mode information = 02</td>
</tr>
<tr>
<td>UID 1 4</td>
<td>UID (tag number, EPC ID)</td>
</tr>
<tr>
<td>UID 5 8</td>
<td></td>
</tr>
<tr>
<td>LFD</td>
<td>Relationship between power flow density limit and actual measured value</td>
</tr>
<tr>
<td>FZP</td>
<td>Error counter, passive (errors during idle time)</td>
</tr>
<tr>
<td>FZA</td>
<td>Error counter, active (errors during communication)</td>
</tr>
<tr>
<td>ANWZ</td>
<td>Presence error</td>
</tr>
<tr>
<td>res.</td>
<td>Spare</td>
</tr>
</tbody>
</table>
GetTagStatus mode = 03

Result message frame
mode = 03

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>status</td>
<td>UID 1 4</td>
<td>UID 5 8</td>
<td>IC_version</td>
<td>Size</td>
<td>lock_state</td>
<td>block_size</td>
<td>nr_of_blocks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter name                  Comment
---                             ------
status                         Tag status mode information = 02
UID 1 4                        UID (tag number)
UID 5 8                        always 0
IC_version                     Chip version (for my-d = 00h)
Size                           Memory size in bytes
lock_state                     Lock status, OTP information: per block (4x4 bytes or 2x8 bytes) one bit (bit=1: block is locked)
br_of_blocks                   Block size of the transponder

For detailed information, please refer to the respective reader description.
GetTagStatus mode = 04

Result message frame
mode = 04

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>status</td>
<td>UID 1 4</td>
<td>UID 5 8</td>
<td>ant</td>
<td>RSSI</td>
<td>H</td>
<td>Min.</td>
<td>Sec</td>
<td>res</td>
<td>res1</td>
<td>res2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter name | Comment
--- | ---
status | Tag status mode information = 04
UID 1 4 | UID (tag number, EPC ID)
UID 5 8 |
ant | Antenna that has detected the MDS
RSSI | RSSI threshold value
H | hours
Min. | Minutes
Sec | Seconds
res | Spare
res1 | Spare
res2 | Spare

For detailed information, please refer to the respective reader description.
Addressing of the RFID tags

Address space of the MDS versions for MOBY U

Table B-1 Address space MOBY U

<table>
<thead>
<tr>
<th>Memory</th>
<th>Addressing</th>
<th>16-bit hexadecimal number</th>
<th>Integer number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 KB data memory</td>
<td>Start address</td>
<td>0000</td>
<td>+0</td>
</tr>
<tr>
<td></td>
<td>End address</td>
<td>07FF</td>
<td>+2047</td>
</tr>
<tr>
<td></td>
<td>Memory size</td>
<td>08 00</td>
<td>2048</td>
</tr>
</tbody>
</table>

Read OTP memory
(write access only possible once. The OTP memory of MOBY U can only be processed completely, i.e. the start address must always be specified with value FFF0 hex and the length with value 10 hex.)

<table>
<thead>
<tr>
<th>Start address</th>
<th>Length</th>
<th>Addressing</th>
<th>Integer number</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFF0</td>
<td>10</td>
<td>-16</td>
<td>+16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID No.: (4 fixed-coded bytes; can only be read with the MDS status command)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory</th>
<th>Addressing</th>
<th>16-bit hexadecimal number</th>
<th>Integer number</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 KB data memory</td>
<td>Start address</td>
<td>0000</td>
<td>+0</td>
</tr>
<tr>
<td></td>
<td>End address</td>
<td>7FFF</td>
<td>+32767</td>
</tr>
<tr>
<td></td>
<td>Memory size</td>
<td>80 00</td>
<td>32768</td>
</tr>
</tbody>
</table>

Read OTP memory (write access only possible once)*

<table>
<thead>
<tr>
<th>Start address</th>
<th>Length</th>
<th>Addressing</th>
<th>Integer number</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFF0</td>
<td>10</td>
<td>-16</td>
<td>+16</td>
</tr>
</tbody>
</table>

ID No.: (4 fixed-coded bytes; can only be read with the MDS status command)

Address space of the MDS versions for MOBY D

For address space of the RF300 transponders, see MOBY D System Manual (http://support.automation.siemens.com/WW/view/en/13628689/0/en), section "Mobile data storage units > Introduction".

Address space of the transponder versions for RF300

For address space of the RF300 transponders, see SIMATIC RF300 System Manual (http://support.automation.siemens.com/WW/view/en/21738946/0/en), section "RF300 transponders > Memory configuration of the RF300 tags" and section "ISO transponders > Memory configuration of the ISO tags".

RF182C communication module
Operating Instructions, 10/2010
Addressing of the RFID tags

Address space of the transponder versions for RF600

For address space of the RF600 transponders, see SIMATIC RF620R/RF630R Parameterization Manual (http://support.automation.siemens.com/WW/view/de/33287195/0/en), section "Examples/applications > Memory configuration".

Transfer scheme for hexadecimal tag data via XML

The addresses for memory addressing on the tag are hexadecimal. The data on the tag is also stored in hexadecimal format. However, with the read/write commands, the data is transferred as ASCII characters. Conversion from hex to ASCII hex and vice versa is executed automatically in the communication module. The following example shows the coding scheme:

Command to RF182C

```
<readTagData>
  <startAddress>0000</startAddress>
  <dataLength>0004</dataLength>
</readTagData>
```

Data in tag

<table>
<thead>
<tr>
<th>Address [hex]</th>
<th>Data [hex]</th>
<th>Data [ADC]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>4D</td>
<td>'M'</td>
</tr>
<tr>
<td>0001</td>
<td>4F</td>
<td>'O'</td>
</tr>
<tr>
<td>0002</td>
<td>42</td>
<td>'B'</td>
</tr>
<tr>
<td>0003</td>
<td>59</td>
<td>'Y'</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Result of the RF182C

```
<readTagData>
  <returnValue>
    <data>4D4F4259</data>
  </returnValue>
</readTagData>
```
Transfer scheme for hexadecimal tag data via XML

Optional: storage of the result data in the data block of a controller

<table>
<thead>
<tr>
<th>Address [dec]</th>
<th>Data [hex]</th>
<th>Data [ASC]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>24</td>
<td>,4”</td>
</tr>
<tr>
<td>N+1</td>
<td>44</td>
<td>'D'</td>
</tr>
<tr>
<td>N+2</td>
<td>34</td>
<td>,4”</td>
</tr>
<tr>
<td>N+3</td>
<td>46</td>
<td>'F'</td>
</tr>
<tr>
<td>N+4</td>
<td>34</td>
<td>,4”</td>
</tr>
<tr>
<td>N+5</td>
<td>32</td>
<td>,2”</td>
</tr>
<tr>
<td>N+6</td>
<td>35</td>
<td>,5”</td>
</tr>
<tr>
<td>N+7</td>
<td>39</td>
<td>,9”</td>
</tr>
</tbody>
</table>
Service & support

Technical support

The technical support specialists advise and assist customers by responding to their queries on the functions of our RFID products and how to work with them.

You can reach us worldwide Mon. to Fri. during office hours: 8 a.m. - 5 p.m. CET:

Telephone:  ++49 (0) 180 5050-222
Fax:  ++49 (0) 180 5050-223

Internet

Visit our site on the Internet at:
Support homepage (www.siemens.com/automation/service&support)

You can send a support query to:
Online support request form: (www.siemens.com/automation/support-request)

General information on new features of the RF182C communication module and an overview of our other identification systems can be found on the Internet at:
RFID homepage (www.siemens.com/simatic-sensors/rt)