



TITANIA REFERENCE MANUAL

AMB3626 / 2607011111000

VERSION 4.1

MARCH 7, 2019

Revision history

| Manual version | FW version | HW version | Notes | Date |
|----------------|-------------|------------|---|---------------|
| 3.7 | 3.3 - 3.3.6 | 2.0 | <ul style="list-style-type: none"> Initial version | June 2017 |
| 4.0 | 3.3.6 | 2.0 | <ul style="list-style-type: none"> New corporate design and structure | November 2018 |
| 4.1 | 3.3.6 | 2.0 | <ul style="list-style-type: none"> Added chapter Reference design Added chapter Information for Ex Protection | February 2019 |

Abbreviations and abstract

| Abbreviation | Name | Description |
|---------------------------|-----------------------|--|
| ACK | Acknowledgement | Acknowledgement pattern confirming the reception of the transmitted data packet. |
| CS | Checksum | |
| DC | Duty cycle | Transmission time in relation of one hour. 1% means, channel is occupied for 36 seconds per hour. |
| FSE | Field Sales Engineer | Support and sales contact person responsible for limited sales area |
| 0xhh [HEX] | Hexadecimal | All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal. |
| HIGH | High signal level | |
| LOW | Low signal level | |
| LPM | Low power mode | Operation mode for reduced power consumption. |
| LRM | Long range mode | Tx mode increasing the RX sensitivity by using spreading and forward error correction |
| LSB | Least significant bit | |
| MSB | Most significant bit | |
| PL | Payload | The real, non-redundant information in a frame/packet. |
| RF | Radio frequency | Describes everything relating to the wireless transmission. |
| UART | | Universal Asynchronous Receiver Transmitter allows communicating with the module of a specific interface. |
| US | UserSettings | Any relation to a specific entry in the UserSettings is marked in a special font and can be found in the respective chapter. |
| VDD | Supply voltage | |
| Payload | | The real, non-redundant information in a frame/packet |
| User settings | | Any relation to a specific entry in the user settings is marked in a special font and can be found in the respective chapter |
| UART | | Universal Asynchronous Receiver Transmitter, allows to communicate with the module of a specific interface. |
| Duty cycle | | Transmission time in relation of one hour 1% means, channel is occupied for 36 seconds per hour. |
| Hexadecimal [HEX] 0xhh | | All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal. |

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 9 |
| 1.1 | Operational description | 9 |
| 1.2 | Block diagram | 9 |
| 1.3 | Ordering information | 10 |
| 2 | Electrical specifications | 11 |
| 2.1 | Recommended operating conditions | 11 |
| 2.2 | Absolute maximum ratings | 11 |
| 2.3 | Power consumption | 11 |
| 2.3.1 | Static | 12 |
| 2.4 | Radio characteristics | 12 |
| 2.5 | Pin characteristics | 12 |
| 2.6 | Integrated flash characteristics | 13 |
| 3 | Pinout | 14 |
| 4 | Quickstart | 17 |
| 4.1 | Minimal pin configuration | 17 |
| 4.2 | Power up | 17 |
| 4.3 | Quickstart example | 18 |
| 5 | Functional description | 21 |
| 5.1 | Operating modes | 21 |
| 5.1.1 | Switching from transparent to command mode | 21 |
| 5.1.2 | Switching from command to transparent mode | 21 |
| 5.1.3 | Transparent mode | 22 |
| 5.1.4 | /RTS signal, module busy | 22 |
| 5.1.5 | Command mode | 23 |
| 5.2 | Adopting parameters to fit your application | 23 |
| 5.3 | Device addressing and wireless monitoring | 23 |
| 6 | Host connection | 24 |
| 6.1 | Serial interface: UART | 24 |
| 7 | The command interface | 25 |
| 7.1 | Overview | 25 |
| 7.2 | Message overview | 26 |
| 7.3 | Data transfer & reception in the command mode | 28 |
| 7.3.1 | CMD_DATA_REQ | 28 |
| 7.3.1.1 | Example 1: | 28 |
| 7.3.2 | CMD_DATAEX_REQ | 29 |
| 7.3.2.1 | Example 1: | 29 |
| 7.3.3 | CMD_DATAEX_IND | 30 |
| 7.3.3.1 | Example 1: | 30 |
| 7.3.4 | CMD_DATARETRY_REQ | 31 |
| 7.4 | Requesting parameters and actions | 32 |
| 7.4.1 | CMD_FWRELEASE_REQ | 32 |
| 7.4.1.1 | Example 1: | 32 |

| | | |
|----------|---|-----------|
| 7.4.2 | CMD_SERIALNO_REQ | 32 |
| 7.4.3 | CMD_RESET_REQ | 33 |
| 7.4.4 | CMD_RSSI_REQ | 33 |
| 7.4.4.1 | Example 1: | 34 |
| 7.4.5 | CMD_ERRORFLAGS_REQ | 34 |
| 7.5 | Modification of volatile parameters | 36 |
| 7.5.1 | CMD_SET_MODE_REQ | 36 |
| 7.5.2 | Example 1: | 36 |
| 7.5.3 | CMD_SET_PAPOWER_REQ | 37 |
| 7.5.3.1 | Example 1: | 37 |
| 7.5.4 | CMD_SET_CHANNEL_REQ | 37 |
| 7.5.4.1 | Example 1: | 38 |
| 7.5.5 | CMD_SET_DESTNETID_REQ | 38 |
| 7.5.6 | CMD_SET_DESTADDR_REQ | 39 |
| 7.6 | Modification of non-volatile parameters | 40 |
| 7.6.1 | CMD_SET_REQ | 40 |
| 7.6.1.1 | Example 1: | 41 |
| 7.6.1.2 | Example 2: | 41 |
| 7.6.2 | CMD_GET_REQ | 42 |
| 7.6.2.1 | Example 1: | 42 |
| 7.6.3 | CMD_FACTORY_RESET_REQ | 43 |
| 8 | UserSettings - Module configuration values | 44 |
| 8.1 | Difference between volatile and non-volatile settings | 44 |
| 8.2 | Modifying the UserSettings | 44 |
| 8.3 | UART_PktMode | 46 |
| 8.3.1 | Example 1: | 47 |
| 8.3.2 | Example 2: | 47 |
| 8.4 | UART_PktSize | 48 |
| 8.4.1 | Example 1: | 48 |
| 8.4.2 | Example 2: | 48 |
| 8.5 | UART_RTSLimit | 49 |
| 8.5.1 | Example 1: | 49 |
| 8.5.2 | Example 2: | 49 |
| 8.6 | UART_ETXChar | 50 |
| 8.6.1 | Example 1: | 50 |
| 8.6.2 | Example 2: | 50 |
| 8.7 | UART_Timeout | 51 |
| 8.7.1 | Example 1: | 51 |
| 8.7.2 | Example 2: | 51 |
| 8.8 | UART_DIDelay | 52 |
| 8.8.1 | Example 1: | 52 |
| 8.8.2 | Example 2: | 52 |
| 8.9 | MAC_NumRetrys | 53 |
| 8.9.1 | Example 1: | 53 |
| 8.9.2 | Example 2: | 53 |
| 8.10 | MAC_AddrMode | 54 |
| 8.10.1 | Example 1: | 54 |
| 8.10.2 | Example 2: | 55 |

| | | |
|--------|---|----|
| 8.11 | MAC_NumRetrysCCA | 56 |
| 8.11.1 | Example 1: | 56 |
| 8.11.2 | Example 2: | 56 |
| 8.12 | MAC_CCARetryDelay | 57 |
| 8.12.1 | Example 1: | 57 |
| 8.12.2 | Example 2: | 57 |
| 8.13 | MAC_DefaultDestNetID | 58 |
| 8.13.1 | Example 1: | 58 |
| 8.13.2 | Example 2: | 58 |
| 8.14 | MAC_DefaultDestAddrLSB | 59 |
| 8.14.1 | Example 1: | 59 |
| 8.14.2 | Example 2: | 59 |
| 8.15 | MAC_DefaultSourceNetID | 60 |
| 8.15.1 | Example 1: | 60 |
| 8.15.2 | Example 2: | 60 |
| 8.16 | MAC_DefaultSourceAddrLSB | 61 |
| 8.16.1 | Example 1: | 61 |
| 8.16.2 | Example 2: | 61 |
| 8.17 | MAC_ACKTimeout | 62 |
| 8.17.1 | Example 1: | 62 |
| 8.17.2 | Example 2: | 62 |
| 8.18 | PHY_PAPower | 63 |
| 8.18.1 | Example 1: | 63 |
| 8.18.2 | Example 2: | 63 |
| 8.19 | PHY_DefaultChannel | 65 |
| 8.19.1 | Example 1: | 65 |
| 8.19.2 | Example 2: | 65 |
| 8.20 | PHY_CCAThr | 66 |
| 8.20.1 | Example 1: | 66 |
| 8.20.2 | Example 2: | 66 |
| 8.21 | OpMode | 67 |
| 8.21.1 | Example 1: | 67 |
| 8.21.2 | Example 2: | 67 |
| 8.22 | CfgFlags | 68 |
| 8.22.1 | Example 1: | 69 |
| 8.22.2 | Example 2: | 69 |
| 8.22.3 | UART_Baudrate: Configure the UART speed | 70 |
| 8.22.4 | Example 1: | 70 |
| 8.22.5 | Example 2: | 70 |
| 8.23 | UART_Databits | 71 |
| 8.23.1 | Example 1: | 71 |
| 8.23.2 | Example 2: | 71 |
| 8.24 | UART_Parity | 72 |
| 8.24.1 | Example 1: | 72 |
| 8.24.2 | Example 2: | 72 |
| 8.25 | UART_Stoppbits | 73 |
| 8.25.1 | Example 1: | 73 |
| 8.25.2 | Example 2: | 73 |
| 8.26 | RF_ConfigIndex | 74 |

| | |
|---|-----------|
| 8.27 RF_CCADisabled | 75 |
| 8.27.1 Example 1: | 75 |
| 8.27.2 Example 2: | 75 |
| 8.28 RF_CCACheckTime | 77 |
| 8.28.1 Example 1: | 77 |
| 8.28.2 Example 2: | 77 |
| 9 Timing parameters | 78 |
| 9.1 Reset behavior | 78 |
| 9.1.1 Power-on reset | 78 |
| 9.1.2 Reset via /RESET pin | 78 |
| 9.1.3 Reset as result of a serious error condition | 78 |
| 9.2 Latencies when leaving the LPM | 78 |
| 9.3 Latencies during data transfer / packet generation | 79 |
| 10 Radio parameters | 80 |
| 10.1 Channel assignment and requirements in Band D | 81 |
| 11 Battery powered operation | 82 |
| 11.1 Active mode | 82 |
| 11.2 Stand-by mode | 82 |
| 12 Custom firmware | 83 |
| 12.1 Custom configuration of standard firmware | 83 |
| 12.2 Customer specific firmware | 83 |
| 12.3 Customer firmware | 83 |
| 12.4 Contact for firmware requests | 84 |
| 13 Firmware update | 85 |
| 13.1 Update using UART interface | 85 |
| 13.2 Update using JTAG or Spy-Bi-Wire | 85 |
| 14 Firmware history | 86 |
| 15 Design in guide | 87 |
| 15.1 Advice for schematic and layout | 87 |
| 15.2 Dimensioning of the micro strip antenna line | 89 |
| 15.3 Antenna solutions | 90 |
| 15.3.1 Wire antenna | 90 |
| 15.3.2 Chip antenna | 91 |
| 15.3.3 PCB antenna | 91 |
| 15.3.4 Antennas provided by Würth Elektronik eiSos | 92 |
| 15.3.4.1 2600130011 - Helike - 169 MHz dipole antenna | 92 |
| 15.3.4.2 2600130041 - Herse - 434 MHz dipole antenna | 93 |
| 15.3.4.3 2600130081 - Hyperion-I - 868 MHz dipole antenna | 94 |
| 15.3.4.4 2600130082 - Hyperion-II - 868 MHz magnetic base antenna | 95 |
| 15.3.4.5 2600130021 - Himalia - 2.4 GHz dipole antenna | 96 |
| 16 Reference design | 97 |
| 16.1 Schematic | 98 |

| | |
|--|------------|
| 16.2 Layout | 100 |
| 17 Manufacturing information | 102 |
| 17.1 Moisture sensitivity level | 102 |
| 17.2 Soldering | 102 |
| 17.2.1 Reflow soldering | 102 |
| 17.2.2 Cleaning | 104 |
| 17.2.3 Other notations | 104 |
| 17.3 ESD handling | 104 |
| 17.4 Safety recommendations | 105 |
| 18 Physical dimensions | 106 |
| 18.1 Dimensions | 106 |
| 18.2 Weight | 106 |
| 18.3 Module drawing | 107 |
| 18.4 Footprint | 108 |
| 18.5 Antenna free area | 108 |
| 19 Marking | 109 |
| 19.1 Lot number | 109 |
| 19.2 General labeling information | 110 |
| 19.2.1 Example labels of Würth Elektronik eiSos products | 110 |
| 20 Information for Ex Protection | 111 |
| 21 Regulatory compliance information | 112 |
| 21.1 Important notice EU | 112 |
| 21.2 Conformity assessment of the final product | 112 |
| 21.3 Exemption clause | 112 |
| 21.4 EU Declaration of conformity | 113 |
| 22 Important information | 114 |
| 22.1 General customer responsibility | 114 |
| 22.2 Customer responsibility related to specific, in particular safety-relevant applications | 114 |
| 22.3 Best care and attention | 114 |
| 22.4 Customer support for product specifications | 114 |
| 22.5 Product improvements | 115 |
| 22.6 Product life cycle | 115 |
| 22.7 Property rights | 115 |
| 22.8 General terms and conditions | 115 |
| 23 Legal notice | 116 |
| 23.1 Exclusion of liability | 116 |
| 23.2 Suitability in customer applications | 116 |
| 23.3 Trademarks | 116 |
| 23.4 Usage restriction | 116 |

| | |
|---|------------|
| 24 License agreement for Würth Elektronik eiSos GmbH & Co. KG connectivity product firmware and software | 118 |
| 24.1 Limited license | 118 |
| 24.2 Usage and obligations | 118 |
| 24.3 Ownership | 119 |
| 24.4 Firmware update(s) | 119 |
| 24.5 Disclaimer of warranty | 119 |
| 24.6 Limitation of liability | 120 |
| 24.7 Applicable law and jurisdiction | 120 |
| 24.8 Severability clause | 120 |
| 24.9 Miscellaneous | 120 |

1 Introduction

1.1 Operational description

The Titania is a radio sub module for wireless communication between devices such as control systems, remote controls, sensors etc. It offers several radio configurations, address modes and relieves the host system of radio-specific tasks as

- checksum calculation
- address resolution
- repetition of addressed but unacknowledged telegrams (if enabled)

It can be deployed wherever the wireless exchange of data packets between two or more parties is required. A serial interface (UART) whose data rate can be adjusted is available for communicating with the host system.

1.2 Block diagram

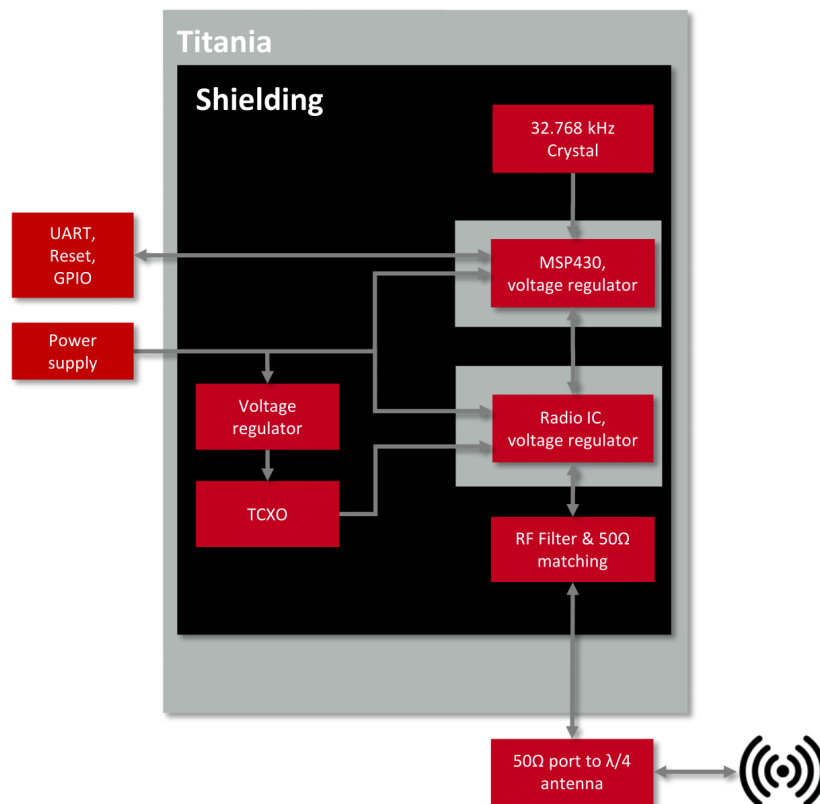


Figure 1: Block diagram

1.3 Ordering information

| WE order code | Former order code | Description |
|---------------|-------------------|---|
| 2607011111001 | AMB3626 | Proprietary radio module 169MHz with antenna pad |
| 2607011111000 | AMB3626-TR | Proprietary radio module 169MHz with antenna pad, Tape & Reel |
| 2607011111009 | AMB3626-DEV | Development kit for AMB3626 |
| 2607019211001 | AMB3626-EV | Evaluation kit for AMB3626 modules |
| 2607046211001 | AMB3665 | USB dongle with AMB3626 module, SMA antenna connector |

Table 1: Ordering information

2 Electrical specifications

As not otherwise stated measured on the evaluation board Titania -EV with $T = 25^{\circ}\text{C}$, $V_{CC} = 3.3\text{V}$ and $50\ \Omega$ conducted.

2.1 Recommended operating conditions

| Description | Min. | Typ. | Max. | Unit |
|------------------------------|------|------|------|--------------------|
| Ambient temperature | -40 | 25 | 85 | $^{\circ}\text{C}$ |
| Supply voltage (V_{DDS}) | 2.0 | 3.0 | 3.6 | V |

Table 2: Recommended operating conditions

2.2 Absolute maximum ratings

| Description | Min. | Typ. | Max. | Unit |
|----------------------------|------|------|------------------------|------|
| V_{CC} | -0.3 | | 3.9 | V |
| Level at any module IO pin | -0.3 | | $V_{CC}+0.3$, max 3.9 | V |
| Output RF level | | 15 | | dBm |
| Input RF level | | | 10 | dBm |

Table 3: Absolute maximum ratings

2.3 Power consumption



The transmit and receive currents are depending on the impedance matching, and therefore may vary depending on antenna selection and matching.



A stable power supply is indispensable to ensure valid operating conditions for the module. Releasing the */RESET* pin before the V_{CC} is stable may damage the module permanently.



A stable power supply and proper antenna design and matching is indispensable to achieve the stated radio characteristics.

2.3.1 Static

The stated current consumption is the sum of the CPU current and the radio current, depending on the active radio mode.

Conditions: Titania -EV with $T = 25^{\circ}\text{C}$, $V_{CC} = 3.3\text{V}$, terminated unused module pads and $50\ \Omega$ conducted.

| Description | Min | Typ. | Max | Unit |
|--|-----|------|-----|---------------|
| TX current consumption at max output power | 60 | 62 | 65 | mA |
| RX current consumption | 27 | 28 | 30 | mA |
| Low power (LPM3), radio off, UART via ACLK | | 10 | | μA |

Table 4: Power consumption

2.4 Radio characteristics

Conditions: Titania -EV, $T=25^{\circ}\text{C}$, $V_{CC}=3.0\text{V}$, terminated unused module pads, $50\ \Omega$ conducted

| Description | Min | Typ. | Max | Unit |
|------------------------|-------|------|---------|------|
| Max output power | -11 | +15 | +15 | dBm |
| Best input sensitivity | | -120 | | dBm |
| Frequencies | 169.4 | | 169.475 | MHz |

Table 5: Radio characteristics

2.5 Pin characteristics

| Property | Min | Typ. | Max | Unit |
|-------------------------------|-----|------|-----|------------------|
| GPIO maximum current per pin | | 2 | | mA |
| GPIO maximum current over all | | 48 | | mA |
| Internal pull-up resistor | 20 | 35 | 50 | $\text{k}\Omega$ |
| Internal Pull-down resistor | 20 | 35 | 50 | $\text{k}\Omega$ |

Table 6: Pin characteristics

2.6 Integrated flash characteristics

The integrated flash memory has only a limited number cycles. This number of flash cycles has a guaranteed min. value of 10000 and a typical value of 100000.

Hosts shall implement methods to prevent their applications from excessively using flash cycles. This can be achieved by always performing a `CMD_GET_REQ` before any `CMD_SET_REQ` and other command that uses flash cycles.

3 Pinout

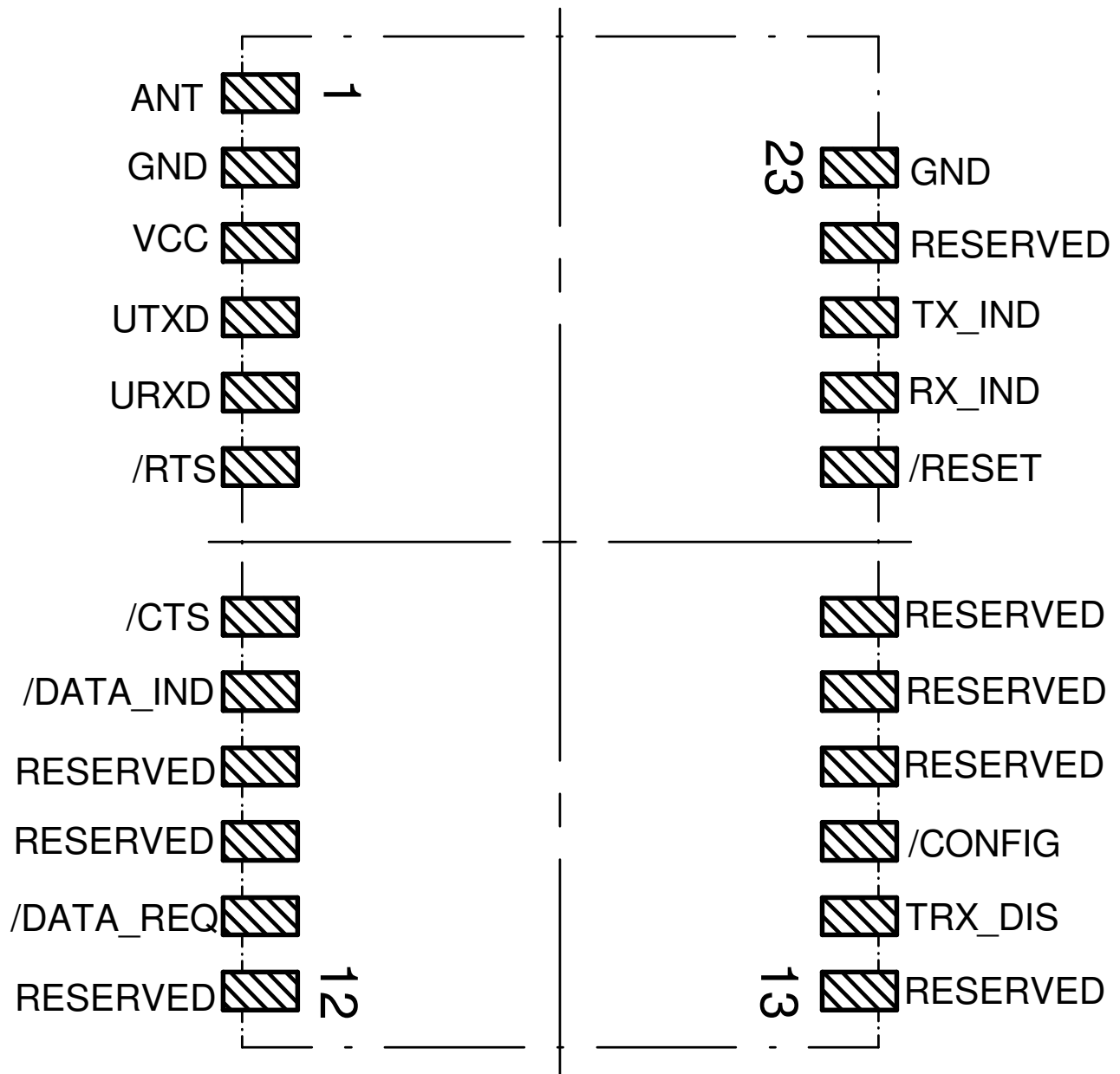


Figure 2: Pinout

| No | Designation | I/O | Description |
|----|----------------------|--------|--|
| 1 | <i>ANT</i> | I/O | Antenna connection, 50Ω |
| 2 | <i>GND</i> | Supply | Ground |
| 3 | <i>VCC</i> | Supply | Supply voltage |
| 4 | <i>UTXD</i> | Output | UART TX (module transmission) |
| 5 | <i>URXD</i> | Input | UART RX (module reception) |
| 6 | <i>/RTS</i> | Output | UART ready to send. HIGH level signalizes a busy module UART buffer. While HIGH, all received UART bytes will be discarded. |
| 7 | <i>/CTS</i> | Input | UART clear to send. HIGH signal indicates host busy. Disabled by default. If the pin function is disabled, connect it to GND. |
| 8 | <i>/DATA_IND</i> | Output | Packet receive indication. Goes LOW as soon as a valid packet with correct address is received via radio, stays LOW for an additional <code>UART_DIDelay</code> and remains LOW as long as the output via UART continues. |
| 9 | <i>RESERVED</i> | Output | Reserved. Do not connect. |
| 10 | <i>RESERVED</i> | Output | Reserved. Do not connect. |
| 11 | <i>/DATA_REQUEST</i> | Input | A falling edge triggers the wireless transmission of buffered data (transparent <code>OpMode</code> only). When enabled this pin uses an internal pull-up resistor. Can be disabled by <code>CfgFlags</code> . If the pin function is disabled, connect it to GND. |
| 12 | <i>RESERVED</i> | Output | Reserved. Do not connect. |

| No | Designation | I/O | Description |
|----|--------------------|--------|---|
| 13 | <i>RESERVED</i> | Output | Reserved. Do not connect. |
| 14 | <i>TRX_DISABLE</i> | Input | Switches the radio receive function off when a HIGH level is applied. The pin level must be hold LOW during the module boot up. Connect to GND if not needed. |
| 15 | <i>/CONFIG</i> | Input | Switch the module's OpMode on a falling edge detection. Can be disabled by CfgFlags. Connect to GND if function is disabled. |
| 16 | <i>RESERVED</i> | Output | Reserved. Do not connect. |
| 17 | <i>TEST</i> | Debug | Test Line. Do not connect. |
| 18 | <i>RESERVED</i> | Output | Reserved Do not connect. |
| 19 | <i>/RESET</i> | Input | Module internal pull-up, a rising edge will release module from reset, LOW will hold module in reset state. |
| 20 | <i>TX_INDICATE</i> | Output | Shows radio activity, HIGH on radio TX. Can be enabled by CfgFlags. |
| 21 | <i>RX_INDICATE</i> | Output | Shows radio activity, HIGH on radio RX. Can be enabled by CfgFlags. |
| 22 | <i>RESERVED</i> | Output | Reserved. Do not connect. |
| 23 | <i>GND</i> | Supply | Ground |

Table 7: Pinout

4 Quickstart

4.1 Minimal pin configuration

In the factory state, the modules are immediately ready for operation; the following pins are required in the minimal configuration: *VCC*, *GND*, *UTXD* and *URXD*.

If the module has to be connected to a PC, an adapter (TTL to RS-232 or TTL to USB) has to be used. The Titania -EV is suited for this.

In the default configuration all module inputs (*TRX_DISABLE* and */CONFIG*) are activated and must be connected as shown in table 7. If the function of the */DATA_REQUEST* pin is enabled (see chapter 8.22), this pin has an internal pull-up resistor.



If *TRX_DISABLE* is used by the host it must be set to *GND* during start-up / after reset till the module's start-up is completed. The module will wait for this pin to go to *GND* level before finishing its start-up procedure. The module's UART or function pins (such as */CONFIG*) will not be available until the start-up is finished.

4.2 Power up

Condition: $\Delta t > 2\text{ms}$.

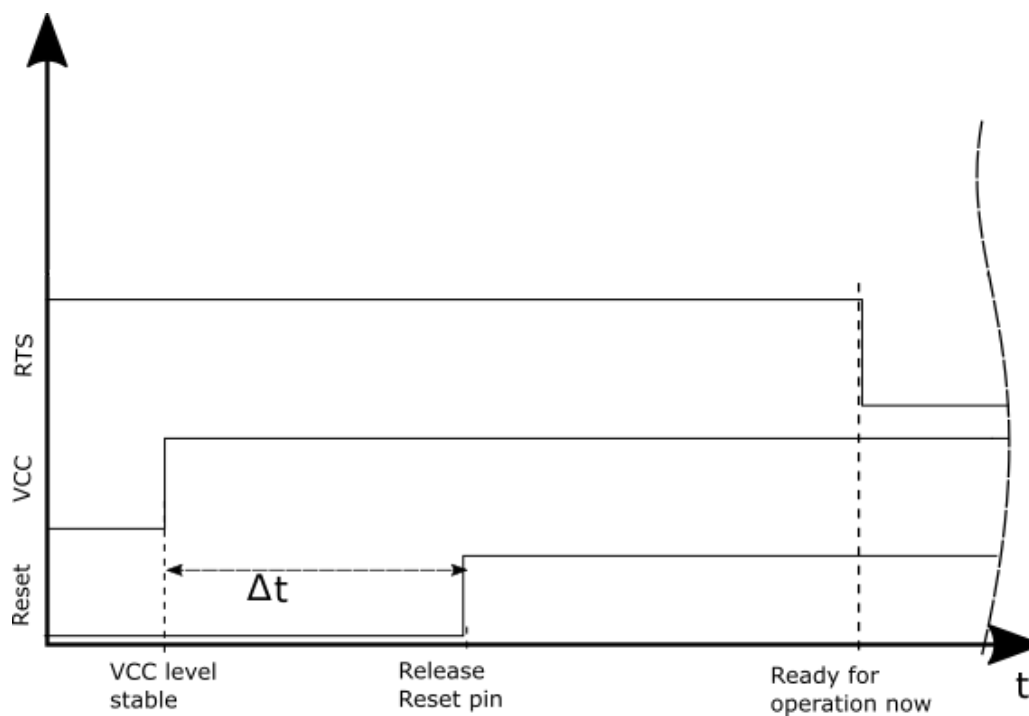


Figure 3: Power up

4.3 Quickstart example



In factory state the module is in transparent mode.

Sending and receiving: Hello World in transparent mode

Connect your pair of modules, EV-boards or USB-sticks with the PC as explained in chapter 4.1. Please make sure you have a minimum distance of 3 meters between the two modules or devices to avoid over modulation. When short distances are needed, you could reduce the `PHY_PAPower` to a minimum.

When the connection to the PC is done, please use a terminal tool of your choice. For convenience we assume you selected the tool *hterm*. Select the two corresponding COM ports and open them with a configuration of 9600 Baud, 8 Data bits, 1 Stop bit and Parity set to None.

Baud 9600 Data 8 Stop 1 Parity None

Enter the string "Hello World" into the input line of hterm and use the "ASend" button followed by pushing the "start" button to send the data once.

Type ASC Hello World! ASend

This data will be received by the second module and shows up as received data in the second hterm instance. You may send any string of size 1 to 120 characters from one module to the other.

Clear received [x] Ascii [x] Hex [] Dec [] Bin [] Save

Received Data

| 1 | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|--------------|---|----|----|----|----|----|----|
| Hello World! | | | | | | | |
| Hello World! | | | | | | | |
| Hello World! | | | | | | | |

You just used the so called "transparent mode" of the modules to send your data. The address mode that was used is "0". Thus all radio frames are broadcasts that can be received by anyone listening with an Titania in default settings. The frame you send was generated using the timeout method.

Besides the transparent mode, that is suited for transparent data transmission, the so called "command mode" allows both, the module configuration and the data transmission, using a predefined command interface (see chapter 7).

Sending and receiving: Hello World in command mode



Be sure that the module runs in command mode by default.

Connect your pair of modules, EV-boards or USB-sticks with the PC as explained in chapter 4.1. Please make sure you have a minimum distance of 3 meters between the two modules or devices to avoid over modulation. When short distances are needed, you could reduce the PHY_PAPower to a minimum.

A terminal program, for example hterm, is used to perform the communication over the COM ports. The two corresponding COM ports have to be selected and opened with a configuration of 9600 baud, 8 Data bits, 1 Stop bit and Parity set to none.

Baud Data Stop Parity

As soon as the module is ready for operation (at start-up or after a reset) a CMD_RESET_CNF message (0x02 0x45 0x01 0x00 0x46) is sent on the UART. Eventually the reset button has to be pushed (or CMD_RESET_REQ performed) to see this message.



If the CMD_RESET_CNF message did not occur after resetting, the module is not in command mode.

In factory state the default address mode is "0", which means that all radio frames are broadcasts that can be received by anyone listening with an Titania in default settings.

Transmitter

To send the string "Hello World", the corresponding CMD_DATA_REQ has to be inserted into the input line of hterm. The command CMD_DATA_REQ has the following structure:

| Start signal | Command | Payload Length | Payload | CS |
|--------------|---------|----------------|----------------|--------|
| 0x02 | 0x00 | 1 Byte | Payload length | 1 Byte |

In this case the payload has a length of 12 (0x0C) bytes and 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21 (Hello World!) is the payload data. The checksum CS is a XOR conjunction of all previous bytes, which in this case is 0x0F.

Using the "ASend" button followed by pushing the "start" button sends the data once.

Type: HEX 02 00 0C 48 65 6C 6C 6F 20 57 6F 72 6C 64 21 0F ASend

The sending module answers with a CMD_DATA_CNF to confirm that the request has been received.

Receiver

In factory state, the second module receiving this packet outputs a CMD_DATAEX_IND message which format is as follows:

| Start signal | Command | Payload Length + 1 | Payload | RSSI | CS |
|--------------|---------|--------------------|----------------|--------|--------|
| 0x02 | 0x81 | 1 Byte | Payload length | 1 Byte | 1 Byte |

Clear received ☒ Ascii ☒ Hex ☐ Dec ☐ Bin Save output

Received Data

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| □ | □ | ✓ | H | e | l | l | o | | W | o | r | l | d | ! | □ | V |
| 02 | 81 | 0D | 48 | 65 | 6C | 6C | 6F | 20 | 57 | 6F | 72 | 6C | 64 | 21 | D9 | 56 |

Thus the CMD_DATAEX_IND message informs us that we received a packet with payload of 0x0D (13) bytes. 12 byte of these are user payload (0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21 = Hello World!) and one byte is the RSSI value (here 0xD9, which is two's complement for -39 dBm) directly outputted behind the payload, before the checksum.

5 Functional description

5.1 Operating modes

The Titania can be used in the following operating modes:

1. Transparent mode (transparent data transmission)
2. Command mode (module configuration and data transmission using the predefined command interface)

The operating mode after power-up can be configured by means of the `OpMode` parameter. By default, the module operates in transparent mode. Starting in the command mode, the module responds with a `CMD_SET_MODE_CNF` telegram.



We highly recommended to only use Command mode.

5.1.1 Switching from transparent to command mode

The command mode can be entered by applying a falling edge on the `/CONFIG` pin. The detection of the falling edge on the `/CONFIG` pin can be disabled using the user setting `CfgFlags`.

The successful switchover is acknowledged by a `CMD_SET_MODE_CNF` telegram indicating command mode.

The switchover can only occur when no data is being received by wireless transmission or UART interface (approximately 100 µs after `/RTS` goes low and indicates readiness).

5.1.2 Switching from command to transparent mode

The transparent mode can be entered by applying a falling edge on the `/CONFIG` pin or by using the command `CMD_SET_MODE_REQ`. The detection of the falling edge on the `/CONFIG` pin can be disabled using the user setting `CfgFlags`.

The successful switchover is acknowledged by a `CMD_SET_MODE_CNF` telegram indicating transparent mode.

The switchover can only occur when no data is being received by wireless transmission or UART interface (approximately 100 µs after `/RTS` goes low and indicates readiness).



Recommendation:

Automatic switching to a specific mode can be realized by applying falling edges on the `/CONFIG` pin as long as the needed `CMD_SET_MODE_CNF` is returned:

- 0x02 0x44 0x01 0x10 0x57 telegram indicating command mode
- 0x02 0x44 0x01 0x00 0x47 telegram indicating transparent mode

5.1.3 Transparent mode

In this mode, data is received via the serial interface and initially buffered. As soon as a specific condition is met, the RF telegram is generated with a preamble, checksum, and address information (optional).

To initiate an RF transmission, several options are available, listed in table 8.

| Start Condition | Description | Dependent Usersettings |
|------------------------------|--|---|
| Timeout | Transmission starts if no new character is detected within a configurable time period after receiving a character via UART. The timeout is reset every time a new character is received. | UART_Timeout UART_PktMode |
| End-Of-Text-Character | Transmission begins when the preconfigured character is transmitted via UART. | UART_PktMode UART_ETXChar |
| Fixed Packet Size | Transmission starts when the preconfigured number of bytes is reached in the RX buffer of the UART. | UART_PktSize UART_RTSLimit UART_PktMode |
| /Data Request Pin | The transmission starts as soon as a falling edge is detected on the <code>/DATA_REQUEST</code> pin. | CfgFlags |

Table 8: Communication in transparent mode

The `UART_PktMode` parameter (see chapter 8.3) can be used to determine which of the listed combinations is to be used.

5.1.4 /RTS signal, module busy

`/RTS` signals a busy UART buffer of the module which means, when `/RTS` is set HIGH, all incoming UART bytes will be discarded.

`/RTS` is set when any of the events in the prior chapter has occurred.



We highly recommended to implement the module `/RTS` pin handling into the host to prevent data loss.

5.1.5 Command mode

This operating mode primarily serves module configuration. The module acts as a slave and can be fully controlled by an external host using the commands of the command interface (see chapter 8.28).

It can also be used for wireless transmission of payload data providing a feedback dependent on the transmission success.

5.2 Adopting parameters to fit your application

The non-volatile parameters (see chapter 8) can only be changed in the command mode by using the `CMD_SET_REQ` command. This command will need the following parameters:

- memory position of the parameter
- the new value that shall be applied to this parameter

Furthermore, there are volatile settings that can be accessed by explicit commands for each parameter. All available commands are introduced in chapter 7.

5.3 Device addressing and wireless monitoring

Settings like the module address can only be modified in the command mode. Thus we recommend to permanently operate in command mode by setting the user settings parameter `OpMode` to the value of `0x10` (16).

To use non-broadcast transmissions you need to adopt the following non-volatile settings:

- `MAC_AddrMode` (mode 1 or 2 should be used depending on the number of addresses you need)
- `MAC_DefaultSourceAddrLSB` as the local address for each device of your network, each member of the network will need an unique address. A value of 255 is invalid.
- `MAC_DefaultSourceNetID`, as the local network address for each device of your network, each member of the network will need an unique address. A value of 255 is invalid.

In command mode, the command `CMD_DATAEX_REQ`, that has the destination address as an own parameter, can be used to send your data to the specified address. A broadcast message can still be achieved when using `0xFF` (255) for both destination address LSB and destination net ID.

The address resolution can be disabled ("packet sniffer") with bit 7 in the `CfgFlags`. A module configured in this way will receive all data packets and forward them to the serial interface, regardless of the addressing mode. In sniffer mode, the module does not send any acknowledgment.

6 Host connection

6.1 Serial interface: UART

The configuration in factory state of the UART is 9600 baud with data format of 8 data bits, no parity and 1 stop bit ("8n1"). The baud rate of the UART can be configured by means of the UserSetting `UART_Baudrate`. The data format can be modified by the UserSettings `UART_Databits`, `UART_Parity` and `UART_Stopbits`.

Since the UART speed is derived from a digitally calibrated oscillator, this may result in variations of up to $\pm 2\%$. When the UART Clock is derived from the ACLK (i.e. 32768Hz crystal) of the module this variation can be up to $\pm 47\%$. Details to this behavior can be extracted from Ti's Family User's Guide for MSP430x5xx μ C family.

The output of characters on the serial interface runs with secondary priority. For this reason, short interruptions may occur between the outputs of individual successive bytes. The host must not implement too strict timeouts between two bytes to be able to receive packets that have interruptions in between.

For the direction "host to module" the host must respect byte-wise the line `/RTS`, which will indicate that the next byte of the packet can be received by the module.

7 The command interface

7.1 Overview

In the command mode, communication with the module occurs in form of predefined commands. These commands must be sent in telegrams according to the format described in table 9.

| | | | | |
|--------------|---------|-------------|------|----------|
| Start signal | Command | No. of data | Data | Checksum |
|--------------|---------|-------------|------|----------|

Table 9: Telegram format in the command mode

Start signal: 0x02 (1 byte)

Command: One of the predefined commands according to chapter 7.2 (1 byte)

No. of data: Specifies the number of data in the following field of variable length and is limited to 128 in order to prevent buffer overflow (1 byte). With appropriate commandos values > 128 can occur.

Data: Variable number of data or parameters (maximum 128 byte, payload plus 6 byte parameter, LSB first)

Checksum: Byte wise XOR combination of the preceding fields including the start signal, i.e. $0x02 \wedge \text{command} \wedge \text{no. of data} \wedge \text{data byte } 0 \dots$ (1 byte)

Using a specific command, data can also be sent via RF, i.e. the module can be operated entirely in the command mode. Only in this way quick channel changes, can be realized.

If no new signal is received for `UART_Timeout` milliseconds after receiving the STX signal, the unit will wait for a new start signal.

On each command follows a response from the module to the host.

7.2 Message overview

| Start signal | CMD | Message name | Short description |
|-----------------|------|-----------------------|---|
| Requests | | | |
| 0x02 | 0x00 | CMD_DATA_REQ | Send data to configured address |
| 0x02 | 0x01 | CMD_DATAEX_REQ | Send data to specific address |
| 0x02 | 0x02 | CMD_DATA_RETRY_REQ | Resend the transmission of the data submitted earlier |
| 0x02 | 0x04 | CMD_SET_MODE_REQ | Change into new operating mode |
| 0x02 | 0x05 | CMD_RESET_REQ | Reset module |
| 0x02 | 0x06 | CMD_SET_CHANNEL_REQ | Change the RF channel |
| 0x02 | 0x07 | CMD_SET_DESTNETID_REQ | Set the destination network ID |
| 0x02 | 0x08 | CMD_SET_DESTADDR_REQ | Set the destination address |
| 0x02 | 0x09 | CMD_SET_REQ | Change the user settings |
| 0x02 | 0x0A | CMD_GET_REQ | Read the user settings |
| 0x02 | 0x0B | CMD_SERIALNO_REQ | Request the individual serial number of the module |
| 0x02 | 0x0C | CMD_FWRELEASE_REQ | Request the firmware version of the module |
| 0x02 | 0x0D | CMD_RSSI_REQ | Request RSSI of last packet |
| 0x02 | 0x0E | CMD_ERRORFLAGS_REQ | Returns internal error states |
| 0x02 | 0x11 | CMD_SET_PAPOWER_REQ | Change the RF TX power |
| 0x02 | 0x12 | CMD_FACTORY_RESET_REQ | Perform a factory reset |

Table 10: Message overview - Part 1

| Start signal | CMD | Message name | Short description |
|----------------------|------|-----------------------|--|
| Confirmations | | | |
| 0x02 | 0x40 | CMD_DATA_CNF | Data has been sent |
| 0x02 | 0x42 | CMD_DATA_RETRY_CNF | Data has been resent |
| 0x02 | 0x44 | CMD_SET_MODE_CNF | Operating mode has been changed |
| 0x02 | 0x45 | CMD_RESET_CNF | Reset request received |
| 0x02 | 0x46 | CMD_SET_CHANNEL_CNF | Channel has been updated |
| 0x02 | 0x47 | CMD_SET_DESTNETID_CNF | Destination network ID has been updated |
| 0x02 | 0x48 | CMD_SET_DESTADDR_CNF | Destination address has been updated |
| 0x02 | 0x49 | CMD_SET_CNF | User settings have been updated |
| 0x02 | 0x4A | CMD_GET_CNF | Return the requested user setting values |
| 0x02 | 0x4B | CMD_SERIALNO_CNF | Serial number request received |
| 0x02 | 0x4C | CMD_FWRELEASE_CNF | Firmware version request received |
| 0x02 | 0x4D | CMD_RSSI_CNF | RSSI request received |
| 0x02 | 0x4E | CMD_ERRORFLAGS_CNF | Internal error states have been returned |
| 0x02 | 0x51 | CMD_SET_PAPOWER_CNF | RF TX power has been changed |
| 0x02 | 0x52 | CMD_FACTORY_RESET_CNF | Factory reset request received |
| Indications | | | |
| 0x02 | 0x81 | CMD_DATAEX_IND | Data has been received |

Table 11: Message overview - Part 2

7.3 Data transfer & reception in the command mode

This group of commands includes the commands that are used to either request a radio telegram to be send or indicates a received frame.

7.3.1 CMD_DATA_REQ

This command serves the simple data transfer in the command mode. Transmission takes place on the configured channel to the previously parameterized destination address. This command is especially suitable for transmission for a point-to-point connection. The number of payload data bytes is limited to 120.

Format:

| Start signal | Command | Payload length | Payload | CS |
|--------------|---------|----------------|----------------|--------|
| 0x02 | 0x00 | 1 Byte | Payload length | 1 Byte |

Response CMD_DATA_CNF:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x40 | 0x01 | 1 Byte | 1 Byte |

0x00: ACK received or not requested (MAC_NumRetrys is 0 or MAC_AddrMode is 0)

0x01: no ACK received

7.3.1.1 Example 1:

Transmit "Hello" (0x48 0x65 0x6C 0x6C 0x6F).

Command:

| Start signal | Command | Payload length | Payload | CS |
|--------------|---------|----------------|--------------------------|------|
| 0x02 | 0x00 | 0x05 | 0x48 0x65 0x6C 0x6C 0x6F | 0x45 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x40 | 0x01 | 0x00 | 0x43 |

Transmission of "Hello" was successful.

7.3.2 CMD_DATAEX_REQ

This command serves data transfer in a network with several parties. Both the channel to use and the destination address (depending on the parameterized addressing mode) are specified along with the command. The number of payload data bytes is limited to 120. The entered channel, destination network and destination address are loaded into the volatile runtime settings and thus kept until the system is reset.

Format in addressing mode 0:

| Start signal | Command | Payload length + 1 | Channel | Payload | CS |
|--------------|---------|--------------------|---------|----------------|--------|
| 0x02 | 0x01 | 1 Byte | 1 Byte | Payload length | 1 Byte |

Format in addressing mode 1:

| Start signal | Command | Payload length + 2 | Channel | Destination address | Payload | CS |
|--------------|---------|--------------------|---------|---------------------|----------------|--------|
| 0x02 | 0x01 | 1 Byte | 1 Byte | 1 Byte | Payload length | 1 Byte |

Format in addressing mode 2:

| Start signal | Command | Payload length + 3 | Channel | Destination network ID | Destination address | Payload | CS |
|--------------|---------|--------------------|---------|------------------------|---------------------|----------------|--------|
| 0x02 | 0x01 | 1 Byte | 1 Byte | 1 Byte | 1 Byte | Payload length | 1 Byte |

Response CMD_DATA_CNF:

| Start signal | CMD_DATA_REQ 0x40 | Length | Status | CS |
|--------------|---------------------|--------|--------|--------|
| 0x02 | 0x40 | 0x01 | 1 Byte | 1 Byte |

0x00: ACK received or not requested (MAC_NumRetrys is 0 or MAC_AddrMode is 0)

0x01: no ACK received

0x02: invalid channel selected

7.3.2.1 Example 1:

Transmit "Goodbye" in addressing mode 1 on channel 108 (0x6A) to address 5.
Command:

| Start signal | Command | Payload length + 2 | Channel | Destination address | Payload | CS |
|--------------|---------|--------------------|---------|---------------------|--|------|
| 0x02 | 0x01 | 0x09 | 0x6A | 0x05 | 0x47 0x6F 0x6F 0x64 0x62 0x79 0x65 | 0x38 |

Response:

| Start signal | CMD_DATA_REQ 0x40 | Length | Status | CS |
|--------------|---------------------|--------|--------|------|
| 0x02 | 0x40 | 0x01 | 0x00 | 0x43 |

"Goodbye" was successfully transmitted.

7.3.3 CMD_DATAEX_IND

This telegram indicates the reception of data bytes and represents the counterpart to the commands CMD_DATA_REQ and CMD_DATAEX_REQ. Apart from the RX field strength (RSSI value), this telegram also specifies the sender address (depending on the parameterized addressing mode).

Format in addressing mode 0:

| Start signal | Command | Payload length + 1 | Payload | RSSI | CS |
|--------------|---------|--------------------|----------------|--------|--------|
| 0x02 | 0x81 | 1 Byte | Payload length | 1 Byte | 1 Byte |

Format in addressing mode 1:

| Start signal | Command | Payload length + 2 | Sender address | Payload | RSSI | CS |
|--------------|---------|--------------------|----------------|----------------|--------|--------|
| 0x02 | 0x81 | 1 Byte | 1 Byte | Payload length | 1 Byte | 1 Byte |

Format in addressing mode 2:

| Start signal | Command | Payload length + 3 | Sender network ID | Sender address | Payload | RSSI | CS |
|--------------|---------|--------------------|-------------------|----------------|----------------|--------|--------|
| 0x02 | 0x81 | 1 Byte | 1 Byte | 1 Byte | Payload length | 1 Byte | 1 Byte |

7.3.3.1 Example 1:

Data was received in MAC_AddrMode 0:

| Start signal | Command | Payload length + 1 | Payload | RSSI | CS |
|--------------|---------|--------------------|---|------|------|
| 0x02 | 0x81 | 0x0D | 0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21 | 0xD9 | 0x56 |

The `CMD_DATAEX_IND` message informs us that we received a packet with payload of 0x0D (13) bytes. 12 byte of these are user payload (0x48 0x65 0x6C 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21 = Hello World!) and one byte is the RSSI value (here 0xD9, which is two's complement for -39 dBm) directly outputted behind the payload, before the checksum.

7.3.4 CMD_DATARETRY_REQ

This command resends the transmission of the data submitted earlier on with `CMD_DATA_REQ` or `CMD_DATAEX_REQ`. Thus, the data does not need to be transmitted again via UART.

The buffered data is lost as soon as new data is sent via UART or data is received via wireless transmission.

Format:

| Start signal | Command | 0x00 | CS |
|--------------|---------|------|--------|
| 0x02 | 0x02 | 0x00 | 1 Byte |

Response `CMD_DATARETRY_CNF`:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x42 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: ACK received, only possible if `MAC_NumRetrys` is not 0; or none is requested

0x01: no ACK received

0x03: no data available (e.g., overwritten by wireless data reception)

7.4 Requesting parameters and actions

This group includes all commands that will return read-only parameters or request actions in the module.

7.4.1 CMD_FWRELEASE_REQ

This command is used to request the firmware version of the module.

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0C | 0x00 | 0x0E |

Response CMD_FWRELEASE_CNF:

| Start signal | Command 0x40 | Length | Firmware version | CS |
|--------------|----------------|--------|------------------|--------|
| 0x02 | 0x4C | 0x03 | 3 Bytes | 1 Byte |

The main version number is returned first, followed by the secondary version number and the revision number.

7.4.1.1 Example 1:

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0C | 0x00 | 0x0E |

Response:

| Start signal | Command 0x40 | Length | Firmware version | CS |
|--------------|----------------|--------|------------------|------|
| 0x02 | 0x4C | 0x03 | 0x03 0x03 0x04 | 0x49 |

The firmware version of the module is 3.3.4.

7.4.2 CMD_SERIALNO_REQ

This command can be used to query the individual serial number of the module.

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0B | 0x00 | 0x09 |

Response CMD_SERIALNO_CNF:

| Start signal | Command 0x40 | Length | Serial number | CS |
|--------------|----------------|--------|---------------|--------|
| 0x02 | 0x4B | 0x04 | 4 Bytes | 1 Byte |

For the serial number, the most significant byte (MSB), which identifies the product (product ID), is returned first.

7.4.3 CMD_RESET_REQ

This command triggers a software reset of the module. The reset is performed after the acknowledgement is transmitted.

Format:

| Start signal | Command | 0x00 | CS |
|--------------|---------|------|------|
| 0x02 | 0x05 | 0x00 | 0x07 |

Response CMD_RESET_CNF:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x45 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: success

7.4.4 CMD_RSSI_REQ

This command returns the RX level of the last received packet determined by the transceiver IC in the form of a signed two's complement.

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0D | 0x00 | 0x0F |

Response CMD_RSSI_CNF:

| Start signal | Command 0x40 | Length | RX level | CS |
|--------------|----------------|--------|----------|--------|
| 0x02 | 0x4D | 0x01 | 1 Byte | 1 Byte |

The delivered RSSI delivery is in two's complement.

The relation between the calculated value and the physical RX level in dBm is not linear across the entire operating range but can be estimated as linear in the range from -110 to -30 dBm.

7.4.4.1 Example 1:

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0D | 0x00 | 0x0F |

Response:

| Start signal | Command 0x40 | Length | RX level | CS |
|--------------|----------------|--------|----------|------|
| 0x02 | 0x4D | 0x01 | 0xBD | 0xF3 |

The value obtained in this way delivers the RX level in dBm as follows:

$$0xDB_{hex} = 10111101_{bin} \Rightarrow 1 \times -128 + 0 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = -67dBm$$

7.4.5 CMD_ERRORFLAGS_REQ

This command returns internal error states.

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x0E | 0x00 | 0x0C |

Response CMD_ERRORFLAGS_CNF:

| Start signal | Command 0x40 | Length | Error flags | CS |
|--------------|----------------|--------|-------------|--------|
| 0x02 | 0x4E | 0x02 | 2 Bytes | 1 Byte |

The value of "0" returned by the error flag implies that no error has occurred. The value is reset either after a query or by a reset.

The meaning of the error flags is not described in detail in this context.

7.5 Modification of volatile parameters

This group contains all functions that will modify runtime settings while the module is running. These settings are all volatile and will be reset to defaults on a reset of the module.

7.5.1 CMD_SET_MODE_REQ

This command is used to toggle the operating mode, e.g. to exit the command mode. The new operating mode is loaded into the volatile runtime settings. This and all other commands can be used in command mode only.

The following operating modes are defined:

- Transparent mode: 0x00
- Command mode: 0x10

Format:

| Start signal | Command | Length | Desired operating mode | CS |
|--------------|---------|--------|------------------------|------|
| 0x02 | 0x04 | 0x01 | 0x00 | 0x07 |

Response CMD_SET_MODE_CNF:

| Start signal | Command 0x40 | Length | Newly configured operating mode | CS |
|--------------|----------------|--------|---------------------------------|--------|
| 0x02 | 0x44 | 0x01 | 1 Byte | 1 Byte |

7.5.2 Example 1:

Enter the transparent mode.

Format:

| Start signal | Command | Length | Desired operating mode | CS |
|--------------|---------|--------|------------------------|------|
| 0x02 | 0x04 | 0x01 | 0x00 | 0x07 |

Response:

| Start signal | Command 0x40 | Length | Newly configured operating mode | CS |
|--------------|----------------|--------|---------------------------------|------|
| 0x02 | 0x44 | 0x01 | 0x00 | 0x47 |

The operating mode has been successfully changed to transparent mode.

7.5.3 CMD_SET_PAPOWER_REQ

This command is used to set the RF TX-power. Unlike the user settings parameter `PHY_PAPower`, this is a volatile runtime parameter, but it is handled in the same way. Thus see section 8.18 for more information.

The power value is entered as a complement on two.

Format:

| Start signal | Command | Length | Power | CS |
|--------------|---------|--------|--------|--------|
| 0x02 | 0x11 | 0x01 | 1 Byte | 1 Byte |

Response `CMD_SET_PAPOWER_CNF`:

| Start signal | Command 0x40 | Length | Power | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x51 | 0x01 | 1 Byte | 1 Byte |

7.5.3.1 Example 1:

Set the power to 14dBm.

Format:

| Start signal | Command | Length | Power | CS |
|--------------|---------|--------|-------|------|
| 0x02 | 0x11 | 0x01 | 0x0E | 0x1C |

Response:

| Start signal | Command 0x40 | Length | Power | CS |
|--------------|----------------|--------|-------|------|
| 0x02 | 0x51 | 0x01 | 0x0E | 0x5C |

The new RF power is 14dBm.

7.5.4 CMD_SET_CHANNEL_REQ

This command is used to select the radio channel. Unlike the user settings parameter `PHY_DefaultChannel`, this is a volatile runtime parameter.

Format:

| Start signal | Command | Length | Channel | CS |
|--------------|---------|--------|---------|--------|
| 0x02 | 0x06 | 0x01 | 1 Byte | 1 Byte |

Response CMD_SET_CHANNEL_CNF:

| Start signal | Command 0x40 | Length | Channel | CS |
|--------------|----------------|--------|---------|--------|
| 0x02 | 0x46 | 0x01 | 1 Byte | 1 Byte |

7.5.4.1 Example 1:

Set the channel to 108.

Format:

| Start signal | Command | Length | Channel | CS |
|--------------|---------|--------|---------|------|
| 0x02 | 0x06 | 0x01 | 0x6C | 0x69 |

Response:

| Start signal | Command 0x40 | Length | Channel | CS |
|--------------|----------------|--------|---------|------|
| 0x02 | 0x46 | 0x01 | 0x6C | 0x29 |

The new channel is 108.

7.5.5 CMD_SET_DESTNETID_REQ

This command serves to configure the destination network ID in addressing mode 2. Unlike the user settings parameter MAC_DefaultDestNetID, this is a volatile runtime parameter.

Format:

| Start signal | Command | Length | Destination network ID | CS |
|--------------|---------|--------|------------------------|--------|
| 0x02 | 0x07 | 0x01 | 1 Byte | 1 Byte |

Response CMD_SET_DESTNETID_CNF:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x47 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: success

7.5.6 CMD_SET_DESTADDR_REQ

This command serves to configure the destination address in addressing modes 1 and 2. Unlike the user settings parameter `MAC_DefaultDestAddrLSB`, this is a volatile runtime parameter.

Format:

| Start signal | Command | Length | Destination address | CS |
|--------------|---------|--------|---------------------|--------|
| 0x02 | 0x08 | 0x01 | 1 Byte | 1 Byte |

Response `CMD_SET_DESTADDR_CNF`:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x48 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: success

7.6 Modification of non-volatile parameters

The non-volatile parameters are also called user settings and are stored in a special flash location.

7.6.1 CMD_SET_REQ

This command enables direct manipulation of the parameters in the module's non-volatile user settings. The respective parameters are accessed by means of the memory positions described in chapter 8.

You can modify individual or multiple consecutive parameters in the memory at the same time. The sum of memory position and forwarded data has to be less than the total size of the user settings (however a max. of 128 Bytes). Otherwise the package is not acknowledged.

The module always makes a local copy of the user settings, then the new values are copied into the respective memory area and finally the complete user settings are rewritten.

Parameters of 2 or more bytes have to be transferred with the LSB first.



The changed parameters only take effect after a restart of the module. This can be done by a CMD_RESET_REQ.



The validity of the specified parameters is not verified. Incorrect values can result in device malfunction!



To save the parameters in the flash memory of the module, the particular memory segment must first be flushed entirely and then restored from RAM. If a reset occurs during this procedure (e.g. due to supply voltage fluctuations), the entire memory area may be destroyed.

In this case, the module may no longer be operable, which means that the firmware must be re-installed via "ACC V3", in which user settings can also be configured and verified.



Recommendation: First verify the configuration of the module with CMD_GET_REQ and only write if required.

Format:

| Start signal | Command | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|--------|
| 0x02 | 0x09 | 1 Byte | 1 Byte | 1 Byte | Length | 1 Byte |

Response CMD_SET_CNF:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x49 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: Request successfully received and processed

0x01: invalid memory position (write access to unauthorized area > 127 / 0xFF)

0x02: invalid number of bytes to be written (write access to unauthorized area > 0xFF)

7.6.1.1 Example 1:

Setting the number of wireless retries to 5 (parameter MAC_NumRetrys, memory position 20).

Command:

| Start signal | Command | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x14 | 0x01 | 0x05 | 0x18 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

7.6.1.2 Example 2:

Setting parameter UART_Baudrate on 115200 baud (memory position 80 and length 4).

Command:

| Start signal | Command | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|---------------------|------|
| 0x02 | 0x09 | 0x06 | 0x50 | 0x04 | 0x00 0xC2 0x01 0x00 | 0x9A |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

7.6.2 CMD_GET_REQ

This command can be used to query individual or multiple user settings parameters. The requested number of bytes from the specified memory position are returned.

You can query individual or multiple consecutive parameters in the memory at the same time. The sum of the memory position and requested data must not be more than the total size of the user-settings (however a max. of 128 Bytes). Otherwise no data will be returned.

Parameters of 2 or more bytes will be transmitted LSB first.

Format:

| Start signal | Command | Length | Memory position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|--------|
| 0x02 | 0x0A | 0x02 | 1 Byte | 1 Byte | 1 Byte |

Response CMD_GET_CNF:

| Start signal | Command 0x40 | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|--------------|--------|
| 0x02 | 0x4A | 1 Byte | 1 Byte | 1 Byte | Length Bytes | 1 Byte |

Read access to the memory area outside the user settings is blocked.

7.6.2.1 Example 1:

Read out the parameter `UART_Baudrate` (memory position 80 and length 4).

Command:

| Start signal | Command | Length | Memory position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x50 | 0x04 | 0x5E |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|---------------------|------|
| 0x02 | 0x4A | 0x06 | 0x50 | 0x04 | 0x00 0xC2 0x01 0x00 | 0xDB |

Read out the `UART_Baudrate` as 0x00 0x01 0xC2 0x00 (115200) Baud.

7.6.3 CMD_FACTORY_RESET_REQ

This command restores the default user settings of the module. If this was successful, a software reset of the module is executed additionally. The reset is performed after the acknowledgement is transmitted.

Format:

| Start signal | Command | Length | CS |
|--------------|---------|--------|------|
| 0x02 | 0x12 | 0x00 | 0x10 |

Response `CMD_FACTORY_RESET_CNF`:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|--------|
| 0x02 | 0x52 | 0x01 | 1 Byte | 1 Byte |

Status:

0x00: Request successfully received and processed

0x01: Request not successful

8 UserSettings - Module configuration values

8.1 Difference between volatile and non-volatile settings

The so-called UserSettings are stored permanently into the internal flash of the module. At start-up, these UserSettings are loaded as start values into the volatile settings ("RuntimeSettings"). Some of the RuntimeSettings can be modified by special commands (see chapter 7.5). These RuntimeSettings are lost and replaced by the UserSettings content when the module is restarted.



See chapters 7.5 and 7.6 for methods to change volatile and/or non-volatile settings.

The non-volatile UserSettings can be modified by means of specific commands in the configuration mode (CMD_SET_REQ) of the module. These parameters are stored permanently in the module's flash memory. All settings are described on the following pages. After changing those parameters, a reset will be necessary to make use of the new settings.



The validity of the specified parameters given with a CMD_SET_REQ is not verified. Incorrect values can result in device malfunction and may even result in the need of re-flashing the entire module firmware!

8.2 Modifying the UserSettings

The following chapters will give examples for the modification for many parameters using the commands CMD_SET_REQ and CMD_GET_REQ. The PC software ACC (version 3.4.3 or newer) can also be used to change non-volatile parameters.

All multiple byte parameters shall be used LSB first if not noted otherwise in the parameter specific section.

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------------|---|--------------------|---------------|-----------------|--------|
| UART_PktMode | Selects the packet generation method | 0, 1 | 0 | 5 | 1 |
| UART_PktSize | Number of characters for transmission start with set packet size | 1 - 120 | 120 | 7 | 1 |
| UART_RTSLimit | Number of received characters after which /RTS responds | 1 - 120 | 100 | 8 | 1 |
| UART_ETXChar | End-of-text character used to mark data packets, reception of this character triggers wireless transmission | 0 - 255 | 10 | 9 | 1 |
| UART_Timeout | Timeout after the last character before the data received via UART are transmitted via wireless transmission [ms] | 2 - 65535 | 5 | 12 | 2 |
| UART_DIDelay | Delay between signal by pin /DATA_INDICATION and beginning of output by UART [ms] | 0 - 65535 | 0 | 14 | 2 |
| MAC_NumRetrys | Number of wireless retries | 0 - 255 | 0 | 20 | 1 |
| MAC_AddrMode | Addressing mode to be used | 0 - 2 | 0 | 21 | 1 |
| MAC_NumRetrysCCA | Number of wireless retries for the CCA | 0 - 255 | 5 | 22 | 1 |
| MAC_CCARetryDelay | Delay for the retry of the CCA | 5 - 255 | 20 | 23 | 1 |

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------------------|--------------------------------|--------------------|---------------|-----------------|--------|
| MAC_DefaultDestNetID | Default destination network ID | 0 - 255 | 0 | 24 | 1 |
| MAC_DefaultDestAddrLSB | Default destination address | 0 - 255 | 0 | 25 | 1 |
| MAC_DefaultSourceNetID | Own/Local network ID | 0 - 254 | 0 | 28 | 1 |
| MAC_DefaultSourceAddrLSB | Own/Local address | 0 - 254 | 0 | 29 | 1 |

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------------|--|--------------------|---------------|-----------------|--------|
| MAC_ACKTimeout | Waiting time for wireless acknowledgement [ms] | 5 - 65535 | 170 | 32 | 2 |
| PHY_PAPower | Output power [dBm], value range depends on RF configuration complement on two | -11 - 15 | 14 | 41 | 1 |
| PHY_DefaultChannel | Utilized wireless channel after reset, value range depends on RF configuration | 0 - 4 | 2 | 42 | 1 |
| PHY_CCAThr | Threshold used for CCA complement on two | 0 - 102 | | 43 | 1 |
| OpMode | Operating mode | 0, 16 | 0 | 60 | 1 |
| CfgFlags | Flags for setting various properties see chapter 8.22 | 0 - 65535 | 512 | 72 | 2 |
| UART_Baudrate | Symbol rate of the UART | 1200 - 115200 | 9600 | 80 | 4 |
| UART_Databits | Number of data bits | 7, 8 | 8 | 84 | 1 |
| UART_Parity | Parity | 0 - 2 | 0 | 85 | 1 |
| UART_Stoppbits | Stop bits | 1, 2 | 1 | 86 | 1 |
| RF_ConfigIndex | Configuration index | 0 - 3 | 2 | 92 | 1 |
| RF_CCADisabled | Clear channel assessment | 0, 1 | 1 | 93 | 1 |
| RF_CCACheckTime | Observation time for CCA [ms] | 0 - 60000 | 5 | 94 | 2 |

Table 12: Overview of non-volatile user-settings

8.3 UART_PktMode

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|--------------------------------------|--------------------|---------------|-----------------|--------|
| UART_PktMode | Selects the packet generation method | 0, 1 | 0 | 5 | 1 |

Selects the packet mode used for generating packets for the transparent operating mode. In command mode the packet end is defined by the length information in the packet header. Only used in `OpMode = transparent`.

Two modes have been implemented:

Mode 0: Transmission starts when the timeout defined with `UART_Timeout` has been reached or the packet has reached size `UART_PktSize`.

Mode 1: Transmission starts when the character defined with `UART_ETXChar` has been detected or the packet has reached size `UART_PktSize`. The `UART_ETXChar` will be sent too.

8.3.1 Example 1:

Set the parameter `UART_PktMode` to 0 (which means the transmission starts when the defined packet size or timeout has been reached).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x05 | 0x01 | 0x00 | 0x0C |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.3.2 Example 2:

Read the `UART_PktMode` from memory position 5 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x05 | 0x01 | 0x0E |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x05 | 0x01 | 0x00 | 0x4F |

Successfully read the value of `UART_PktMode` as 0.

8.4 UART_PktSize

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|--|--------------------|---------------|-----------------|--------|
| UART_PktSize | Number of characters for transmission start with set packet size | 1 - 120 | 120 | 7 | 1 |

Maximum number of bytes after which the wireless transmission of the data received via UART starts. Used in packet mode 0 as well as in packet mode 1. Maximum is 128 due to buffer size.

Not used in command mode.

8.4.1 Example 1:

Set the parameter UART_PktSize to 120 (0x78).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x07 | 0x01 | 0x78 | 0x76 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.4.2 Example 2:

Read the UART_PktSize from memory position 7 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x07 | 0x01 | 0x0C |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x07 | 0x01 | 0x80 | 0xCD |

Successfully read the value of UART_PktSize as 128 (0x80).

8.5 UART_RTSLimit

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|---------------|---|--------------------|---------------|-----------------|--------|
| UART_RTSLimit | Number of received characters after which /RTS responds | 1 - 120 | 100 | 8 | 1 |

Number of bytes after which the host system is prompted to interrupt the data transfer over /RTS. This is necessary, because depending on the host system, an immediate response to the /RTS signal may not take place (UART FIFO).

Not used in command mode.

8.5.1 Example 1:

Set the parameter UART_RTSLimit to 120 (0x78)

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x08 | 0x01 | 0x78 | 0x79 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.5.2 Example 2:

Read the UART_RTSLimit from memory position 8 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x08 | 0x01 | 0x03 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x08 | 0x01 | 0x70 | 0x32 |

Successfully read the value of UART_RTSLimit as 112 (0x70).

8.6 UART_ETXChar

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|---|--------------------|---------------|-----------------|--------|
| UART_ETXChar | End-of-text character used to mark data packets, reception of this character triggers wireless transmission | 0 - 255 | 10 | 9 | 1 |

End-of-text character that triggers the transmission of the data received via UART. Only used in packet mode 1 and in `OpMode = transparent`. During the wireless transmission, the ETX character is treated like a normal character.

Not used in the command mode.

8.6.1 Example 1:

Set the parameter `UART_ETXChar` to 13.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x09 | 0x01 | 0x0D | 0x0D |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.6.2 Example 2:

Read the `UART_ETXChar` from memory position 9 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x09 | 0x01 | 0x02 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x09 | 0x01 | 0x0A | 0x49 |

Successfully read the value of `UART_ETXChar` as 10.

8.7 UART_Timeout

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|---|--------------------|---------------|-----------------|--------|
| UART_Timeout | Timeout for radio send data trigger (after the last received UART character) [ms] | 2 - 65535 | 5 | 12 | 2 |

The timeout defines the delay in milliseconds in transparent mode after the last character has been received by the UART before the wireless transmission starts. Only used in packet mode 0. The value should be chosen appropriate to the UART data rate. Only used in OpMode = transparent.

8.7.1 Example 1:

Set the parameter UART_Timeout to 10.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x04 | 0x0C | 0x02 | 0x0A 0x00 | 0x0B |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.7.2 Example 2:

Read the UART_Timeout from memory position 12 and length 2.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x0C | 0x02 | 0x04 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x04 | 0x0C | 0x02 | 0x05 0x00 | 0x47 |

Successfully read the value of UART_Timeout as 5 ms. 5

8.8 UART_DIDelay

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|---|--------------------|---------------|-----------------|--------|
| UART_DIDelay | Delay between signal by pin <i>/DATA_INDICATION</i> and beginning of UART transmission to host [ms] | 0 - 65535 | 0 | 14 | 2 |

This parameter determines the delay in milliseconds between the indication of incoming RF data by the */DATA_INDICATION* pin and the output of the data on UART. This delay can be used to alert a sleeping host system to prepare for the reception of data.

8.8.1 Example 1:

Set the parameter UART_DIDelay to 5.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x04 | 0x0E | 0x02 | 0x05 0x00 | 0x06 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.8.2 Example 2:

Read the UART_DIDelay from memory position 14 and length 2.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x0E | 0x02 | 0x06 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x04 | 0x0E | 0x02 | 0x00 0x00 | 0x40 |

Successfully read the value of UART_DIDelay as 0 ms.

8.9 MAC_NumRetrys

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|---------------|----------------------------|--------------------|---------------|-----------------|--------|
| MAC_NumRetrys | Number of wireless retries | 0 - 255 | 0 | 20 | 1 |

Determines the maximum number of wireless transmission retries. If this parameter is set to a value other than 0, the receiver module will automatically be prompted to send a wireless acknowledgement ("ACK"). Please note that sending acknowledgements additionally increases the traffic.

According to *ESTI EN 301 391*, the value for MAC_NumRetrys should be 5 at most.

8.9.1 Example 1:

Setting the number of wireless retries to 5 (parameter MAC_NumRetrys).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x14 | 0x01 | 0x05 | 0x18 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.9.2 Example 2:

Read the MAC_NumRetrys from memory position 20 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x14 | 0x01 | 0x1F |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x14 | 0x01 | 0x00 | 0x5E |

Successfully read the value of MAC_NumRetrys as 0.

8.10 MAC_AddrMode

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------|------------------------|--------------------|---------------|-----------------|--------|
| MAC_AddrMode | Addressing mode to use | 0 - 2 | 0 | 21 | 1 |

Addressing mode selection. The following modes have been implemented:

- No addressing (mode 0): Each module receives the transmitted RF telegram and delivers the received data to the host system via UART. No address information is transmitted in the radio telegram.
- 1-byte address (mode 1): The receiving module only delivers the data to the host system via UART if the destination address configured at the sender (MAC_DefaultDestAddrLSB) corresponds to the source address (MAC_DefaultSourceAddrLSB) or the destination address 255 (broadcast) was specified. Both the destination address and the source address are transmitted in the wireless telegram (total = 2 bytes).
- 2-bytes address (mode 2): The receiving module only delivers the data to the host system via UART if both the destination network ID and the destination address correspond to the source addresses (MAC_DefaultSourceNetID and MAC_DefaultSourceAddrLSB) or the destination address 255 (broadcast) was specified. A total of 4 bytes of address information are transmitted in the wireless telegram.



Caution: In addressing mode 0, the use of wireless acknowledgement may cause problems if several wireless modules are addressed simultaneously. Therefore no ACK is requested when using addressing mode 0.



The receiver and transmitter modules must operate in the same addressing mode! Otherwise the receiver cannot decrypt the data packet sent and thus the packet is discarded!

8.10.1 Example 1:

Setting the MAC_AddrMode to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x15 | 0x01 | 0x01 | 0x1D |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.10.2 Example 2:

Read the MAC_AddrMode from memory position 21 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x15 | 0x01 | 0x1E |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x15 | 0x01 | 0x01 | 0x5E |

Successfully read the value of MAC_AddrMode as 1.

8.11 MAC_NumRetrysCCA

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|------------------|--|--------------------|---------------|-----------------|--------|
| MAC_NumRetrysCCA | Number of wireless retries for the CCA | 0 - 255 | 5 | 22 | 1 |

Determines the maximum number of retries, the module is checking for a clear channel before wireless transmission (CCA). The user must adopt all relevant CCA values to the system design and environment.

8.11.1 Example 1:

Set the parameter MAC_NumRetrysCCA to 2.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x16 | 0x01 | 0x02 | 0x1D |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.11.2 Example 2:

Read the MAC_NumRetrysCCA from memory position 22 & length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x16 | 0x01 | 0x1D |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x16 | 0x01 | 0x05 | 0x59 |

Successfully read the value of MAC_NumRetrysCCA as 5.

8.12 MAC_CCARetryDelay

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------------|--------------------------------|--------------------|---------------|-----------------|--------|
| MAC_CCARetryDelay | Delay for the retry of the CCA | 5 - 255 | 20 | 23 | 1 |

Determines the delay the module waits for wireless transmission after a busy channel was detected. The user must adopt all relevant CCA values to the system design and environment.

8.12.1 Example 1:

Set the parameter MAC_CCARetryDelay to 50 ms.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x17 | 0x01 | 0x32 | 0x2C |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.12.2 Example 2:

Read the MAC_CCARetryDelay from memory position 23 & length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x17 | 0x01 | 0x1C |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x17 | 0x01 | 0x14 | 0x49 |

Successfully read the value of MAC_CCARetryDelay as 20 ms.

8.13 MAC_DefaultDestNetID

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|----------------------|--------------------------------|--------------------|---------------|-----------------|--------|
| MAC_DefaultDestNetID | Default destination network ID | 0 - 255 | 0 | 24 | 1 |

Destination network address which is used in addressing mode 2. Can be modified with the command CMD_SET_DESTNETID_REQ at runtime (volatile). If the special broadcast ID and the broadcast address are set to 255, the packets will be received by all network participants.

8.13.1 Example 1:

Set the parameter MAC_DefaultDestNetID to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x18 | 0x01 | 0x01 | 0x10 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.13.2 Example 2:

Read MAC_DefaultDestNetID from memory position 24, length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x18 | 0x01 | 0x13 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x18 | 0x01 | 0x00 | 0x52 |

Successfully read the value of MAC_DefaultDestNetID as 0.

8.14 MAC_DefaultDestAddrLSB

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|------------------------|-----------------------------------|--------------------|---------------|-----------------|--------|
| MAC_DefaultDestAddrLSB | Default destination address (LSB) | 0 - 255 | 0 | 25 | 1 |

Least significant byte of the destination address which is used in addressing modes 1 and 2. Can be modified with the command CMD_SET_DESTADDR_REQ at runtime (volatile). If the special broadcast address 255 is used (in the case of addressing mode 2, broadcast ID is also 255), the packets will be received by all network participants.

8.14.1 Example 1:

Set the parameter MAC_DefaultDestAddrLSB to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x19 | 0x01 | 0x01 | 0x11 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.14.2 Example 2:

Read the MAC_DefaultDestNetID from memory position 24 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x18 | 0x01 | 0x13 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x18 | 0x01 | 0x00 | 0x52 |

Successfully read the value of MAC_DefaultDestNetID as 0.

8.15 MAC_DefaultSourceNetID

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|------------------------|----------------------|--------------------|---------------|-----------------|--------|
| MAC_DefaultSourceNetID | Own/Local network ID | 0 - 254 | 0 | 28 | 1 |

Source network ID to be used in addressing mode 2.

8.15.1 Example 1:

Set the parameter MAC_DefaultDestNetID to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x1C | 0x01 | 0x01 | 0x14 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.15.2 Example 2:

Read the MAC_DefaultSourceNetID from memory position 28, length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x1C | 0x01 | 0x17 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x1C | 0x01 | 0x00 | 0x56 |

Successfully read the value of MAC_DefaultSourceNetID as 0.

8.16 MAC_DefaultSourceAddrLSB

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------------------|-------------------------|--------------------|---------------|-----------------|--------|
| MAC_DefaultSourceAddrLSB | Own/Local address (LSB) | 0 - 254 | 0 | 29 | 1 |

Source device address to be used in addressing modes 1 and 2.

8.16.1 Example 1:

Set the parameter MAC_DefaultSourceAddrLSB to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x1D | 0x01 | 0x01 | 0x15 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.16.2 Example 2:

Read MAC_DefaultSourceAddrLSB from memory position 29, length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x1D | 0x01 | 0x16 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x1D | 0x01 | 0x00 | 0x57 |

Successfully read the value of MAC_DefaultSourceAddrLSB as 0.

8.17 MAC_ACKTimeout

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|----------------|--|--------------------|---------------|-----------------|--------|
| MAC_ACKTimeout | Waiting time for wireless acknowledgement [ms] | 5 - 65535 | 170 | 32 | 2 |

Time to wait for a RF acknowledgement before a RF retry is triggered.

8.17.1 Example 1:

Set the parameter MAC_ACKTimeout to 15.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x04 | 0x20 | 0x02 | 0x0F 0x00 | 0x22 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed.

8.17.2 Example 2:

Read the MAC_ACKTimeout from memory position 32 and length 2.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x20 | 0x02 | 0x28 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x04 | 0x20 | 0x02 | 0x0A 0x00 | 0x64 |

Successfully read the value of MAC_ACKTimeout as 10 ms.

8.18 PHY_PAPower

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------|---|--------------------|---------------|-----------------|--------|
| PHY_PAPower | Output power [dBm], value range depends on RF configuration complement on two | -11 - 15 | 14 | 41 | 1 |

Parameter for the RF output power of the module. The maximum permissible output depends on the used RF configurations.

The RF chip only supports discrete values. Mapping to the next possible PHY_PAPower value is done by the module. The next smaller PHY_PAPower value is always chosen when the transferred value is not possible. The step distance equals 1 dB.

The Usersettings PHY_PAPower is entered as a complement on two.



The statutory regulations for the maximum power output have to be adhered to.

8.18.1 Example 1:

Using the value 15 dBm with parameter PHY_PAPower.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x29 | 0x01 | 0x0F | 0x2F |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.18.2 Example 2:

Read the PHY_PAPower from memory position 41 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x29 | 0x01 | 0x22 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x29 | 0x01 | 0x0F | 0x6C |

Successfully read the value of PHY_PAPower as 15 dBm.

8.19 PHY_DefaultChannel

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|--------------------|--|--------------------|---------------|-----------------|--------|
| PHY_DefaultChannel | Utilized wireless channel after reset, value range depends on RF configuration | 0 - 4 | 2 | 42 | 1 |

Determines the wireless channel of the module to be used after a reset.

$$PHYDefaultChannel = \frac{Frequency_{RF} - 169.4125MHz}{0.0125MHz} \quad (1)$$

8.19.1 Example 1:

Using the channel 100 as PHY_DefaultChannel.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x2A | 0x01 | 0x64 | 0x47 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed.

8.19.2 Example 2:

Read the PHY_DefaultChannel from memory position 42 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x2A | 0x01 | 0x21 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x2A | 0x01 | 0x64 | 0x04 |

Successfully read the value of PHY_DefaultChannel as 100.

8.20 PHY_CCAThr

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-----------------------------|--|--------------------|---------------|-----------------|--------|
| PHY_CCAThr CCA threshold | Threshold used for CCA complement on two | 0 - 102 | | 43 | 1 |

This parameters sets the threshold at which a signal is considered as such. The user must adopt all relevant CCA values to the system design and environment. The calculation for the user settings is as follow:

$$\text{Threshold}_{dBm} = \text{PHY_CCAThr} - 102 \text{ dBm}$$

8.20.1 Example 1:

Using the value 25 (which means -87dBm) as PHY_CCAThr.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x2B | 0x01 | 0x19 | 0x3B |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed.

8.20.2 Example 2:

Read the PHY_CCAThr from memory position 43 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x2B | 0x01 | 0x20 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x2B | 0x01 | 0x16 | 0x77 |

Successfully read the value of PHY_CCAThr as 22 (which means -80 dBm).

8.21 OpMode

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------|----------------|--------------------|---------------|-----------------|--------|
| OpMode | Operating mode | 0, 16 | 0 | 60 | 1 |

Choose between operating modes. Can be selected between mode 0 (transparent data transfer) and mode 16 (command mode).

8.21.1 Example 1:

Set the parameter OpMode to 16 (command mode).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x3C | 0x01 | 0x10 | 0x25 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.21.2 Example 2:

Read the OpMode from memory position 60 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x3C | 0x01 | 0x37 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x3C | 0x01 | 0x00 | 0x76 |

Successfully read the value of OpMode as 0 (transparent mode).

8.22 CfgFlags

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------|---|--------------------|---------------|-----------------|--------|
| CfgFlags | Flags for setting various properties see chapter 8.22 | 0 - 65535 | 512 | 72 | 2 |

16-bit field in which the use of individual pins or signals can be disabled. Table 13 represents a description of the respective flags.

To use multiple settings, add the bit numbers and choose the result as value for CfgFlags.

By default, CfgFlags for modules are 0x0200 (LED's enabled). For the USB-Sticks the default CfgFlags are 0x021F (LED's enabled, *Config*, *Data_Request* & *TRX_Disable* pins disabled).

| Bit no. | Description |
|------------|---|
| 0 (0x0001) | Setting this bit disables the <i>/CONFIG</i> pin. Thus the unit can no longer be switched to the command mode via this pin. |
| 1 (0x0002) | Setting this bit disables the <i>/DATA_REQUEST</i> pin. |
| 2 (0x0004) | Reserved |
| 3 (0x0008) | Setting this bit, disables handling of the status of the <i>TRX_DISABLE</i> pin. Hence, the module can no longer be set to the various power-saving modes via this pin. |
| 4 (0x0010) | Setting this bit, enables a different behavior of the <i>TRX_DISABLE</i> pin. If this bit is set and the <i>TRX_DISABLE</i> pin is set, then additionally to the RF-chip, the UART is powered down. In this case, furthermore the chip is powered down to LPM3 and the <i>CONFIG</i> pin is disabled. The needed wakeup time after releasing the <i>TRX_DISABLE</i> pin is < 1ms. |
| 5 (0x0020) | Any character will be accepted as valid checksum in the command mode if this bit is set. |
| 6 (0x0040) | Setting this bit, disables the pull down of the pin <i>TRX_DISABLE</i> . |
| 7 (0x0080) | The address will not be resolved if this bit is set. The particular module can be used as packet sniffer to monitor a wireless link. No ACK is sent. |
| 8 (0x0100) | Setting this bit enables the <i>/CTS</i> flow control pin. |
| 9 (0x0200) | Setting this bit enables the outputs for RF activity, (e.g. for LEDs). |
| 10 to 15 | Reserved |

Table 13: Configuration flags



Warning: If both bit 0 and bit 2 are set, the module can no longer be set to the command mode.



This parameter set consisting of two bytes has to be transferred LSB first. That means, first the byte with bits 0 ... 7, then the byte with bits 8 ... 15.

8.22.1 Example 1:

Set the `CfgFlags` to 0.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x04 | 0x48 | 0x02 | 0x00 0x00 | 0x45 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.22.2 Example 2:

Read the `CfgFlags` from memory position 72 and length 2.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x48 | 0x02 | 0x40 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x04 | 0x48 | 0x02 | 0x00 0x02 | 0x04 |

Successfully read the value of `CfgFlags` as 512 (LEDs are enabled).

8.22.3 UART_Baudrate: Configure the UART speed

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|---------------|-------------------------|--------------------|---------------|-----------------|--------|
| UART_Baudrate | Symbol rate of the UART | 1200 - 115200 | 9600 | 80 | 4 |

A 32 bit field, that contains the symbol rate for the communication interface (in LSB first notation). Symbol rates up to 115200 baud are supported. Default symbol rate is 9600 baud. Please note that for baud rates higher than 9600 baud the LPM has a higher energy consumption.

8.22.4 Example 1:

Set the UART_Baudrate to 115200 Baud. Take decimal 115200, convert it to MSB first hex notation 0x0001c200 and then apply the LSB first byte order: 0x00C20100.

Command:

| Start signal | Command | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|---------------------|------|
| 0x02 | 0x09 | 0x06 | 0x50 | 0x04 | 0x00 0xC2 0x01 0x00 | 0x9A |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed.

8.22.5 Example 2:

Read the UART_Baudrate from memory position 80 and length 4.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x50 | 0x04 | 0x5E |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|---------------------|------|
| 0x02 | 0x4A | 0x06 | 0x50 | 0x04 | 0x00 0xC2 0x01 0x00 | 0xD9 |

Successfully read the value of UART_Baudrate as 115200 Baud.

8.23 UART_Databits

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|---------------|---------------------|--------------------|---------------|-----------------|--------|
| UART_Databits | Number of data bits | 7, 8 | 8 | 84 | 1 |

An 8 bit field that contains the number of data bits on the communication interface. Supported values are 7 and 8.



It is highly recommended to use 8 databits.

8.23.1 Example 1:

Set the parameter UART_Databits to 8.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x54 | 0x01 | 0x08 | 0x55 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.23.2 Example 2:

Read the UART_Databits from memory position 84 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x54 | 0x01 | 0x5F |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x54 | 0x01 | 0x08 | 0x16 |

Successfully read the value of UART_Databits as 8.

8.24 UART_Parity

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-------------|---------|--------------------|---------------|-----------------|--------|
| UART_Parity | Parity | 0, 1, 2 | 0 | 85 | 1 |

| Parity mode | UART_Parity value |
|-----------------|-------------------|
| 'n' no parity | 0 |
| 'e' even parity | 1 |
| 'o' odd parity | 2 |

An 8 bit field that contains the parity mode for the communication interface.

8.24.1 Example 1:

Set the parameter UART_Parity to 0 (no parity).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x55 | 0x01 | 0x00 | 0x5C |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.24.2 Example 2:

Read the UART_Parity from memory position 85 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x55 | 0x01 | 0x5E |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x55 | 0x01 | 0x00 | 0x1F |

Successfully read the value of UART_Parity as 0 ('n' no parity).

8.25 UART_Stoppbits

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|----------------|-----------|--------------------|---------------|-----------------|--------|
| UART_Stoppbits | Stop bits | 1, 2 | 1 | 86 | 1 |

An 8 bit field that contains the number of stop bits for the communication interface. Supported are 1 and 2 stop bits.



It is highly recommended to use 1 stop bit.

8.25.1 Example 1:

Set the parameter UART_Stoppbits to 1.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x56 | 0x01 | 0x01 | 0x5E |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.25.2 Example 2:

Read the UART_Stoppbits from memory position 86 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x56 | 0x01 | 0x5D |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x56 | 0x01 | 0x01 | 0x1D |

Successfully read the value of UART_Stoppbits as 1.

8.26 RF_ConfigIndex

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|----------------|---------------------|--------------------|---------------|-----------------|--------|
| RF_ConfigIndex | Configuration index | 0 - 3 | 2 | 92 | 1 |

An 8 bit field that addresses the applied RF configuration, see Table 14

8.27 RF_CCADisabled



Due to backwards compatibility this CCA method is not suited for being used as Polite Spectrum Access (PSA) as introduced by Norms of the European Radio Equipment Directive (RED). If you need PSA in your system a customer specific firmware is required.

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|----------------|--------------------------|--------------------|---------------|-----------------|--------|
| RF_CCADisabled | Clear channel assessment | 0, 1 | 1 | 93 | 1 |

An 8 bit field that disables the channel access via clear channel assessment. The default value is 1 and means CCA is disabled.

When enabled each CCA parameter must be adopted to the customer's end system and environment.

8.27.1 Example 1:

Set the parameter RF_CCADisabled to 0 (i.e. CCA is enabled).

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x03 | 0x5D | 0x01 | 0x00 | 0x54 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed

8.27.2 Example 2:

Read the parameter RF_CCADisabled from memory position 93 and length 1.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x5D | 0x01 | 0x56 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x03 | 0x5D | 0x01 | 0x00 | 0x17 |

Successfully read the value of `RF_CCADisabled` as 0 (which means CCA is enabled).

8.28 RF_CCACheckTime

| Designation | Summary | Permissible values | Default value | Memory position | Length |
|-----------------|-----------------------|--------------------|---------------|-----------------|--------|
| RF_CCACheckTime | Observation time [ms] | 0 - 60000 | 5 | 94 | 2 |

A 16 bit field that contains the time in milliseconds for which the channel with activated CCA has to be observed and identified as free before channel access can take place.

8.28.1 Example 1:

Set the RF_CCACheckTime to 10ms.

Command:

| Start signal | Command | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|---------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x09 | 0x04 | 0x5E | 0x02 | 0x0A 0x00 | 0x59 |

Response:

| Start signal | Command 0x40 | Length | Status | CS |
|--------------|----------------|--------|--------|------|
| 0x02 | 0x49 | 0x01 | 0x00 | 0x4A |

Request successfully received and processed.

8.28.2 Example 2:

Read the RF_CCACheckTime from memory position 94 and length 2.

Command:

| Start signal | Command | Length | Memory Position | Amount of Bytes | CS |
|--------------|---------|--------|-----------------|-----------------|------|
| 0x02 | 0x0A | 0x02 | 0x5E | 0x02 | 0x56 |

Response:

| Start signal | Command 0x40 | Length + 2 | Memory Position | Length | Parameter | CS |
|--------------|----------------|------------|-----------------|--------|-----------|------|
| 0x02 | 0x4A | 0x04 | 0x5E | 0x02 | 0x05 0x00 | 0x15 |

Successfully read the value of RF_CCACheckTime as 5 ms.

9 Timing parameters

9.1 Reset behavior

Following a reset, a stable (>5ms) LOW on the */RTS* pin signals that the module is ready for operation. This level is however only valid, after the delay required for the internal initialization of the processor (a few μ s).

9.1.1 Power-on reset

Hold the */RESET* LOW. After applying the supply voltage, waiting till the supply voltage level is stable and releasing the */RESET* pin to HIGH, the time until the module is ready for operation can last up to 1s. And will be indicated by a stable (>5ms) LOW level of the */RTS* module signal.

9.1.2 Reset via */RESET* pin

To force a module restart by means of the */RESET* pin, it must first be drawn to LOW for at least 10 ms before a rising edge will trigger a module restart.

After the pin is released, */RTS* will switch to high after 100 μ s at the latest. Since the start-up time for the clock quartz does not apply in this case, the time until the module is ready for operation is reduced to a couple of ms. During this time, the processor clock-rate will be calibrated, which takes anyway between 2 and 20 ms depending on the supply voltage and temperature.

Recommended procedure:

After the */RESET* pin is released, wait for the stable LOW level on the */RTS* pin. Subsequently, additional 100 μ s are required until the system is ready.

9.1.3 Reset as result of a serious error condition

If the module runs in a serious error condition, a software reset is executed. In this case, the module starts up automatically and can be used again. The volatile runtime settings are reset to default, see chapter 7.

9.2 Latencies when leaving the LPM

The module enters a LPM as soon as no data-transmission request is received via serial and RF interface.

If the device returns from such a mode, all internal settings like the channel calibration values and noise levels have been retained, such that the module is ready after a few μ s. Also here a low signal at the */RTS* indicates that the module is ready for operation.

9.3 Latencies during data transfer / packet generation

The data transfer is always buffered, i.e. data received via UART is buffered in the module until a specific event occurs. Subsequently, the UART reception is interrupted (flow control with */RTS* signal), and the payload data is passed to the internal memory of the wireless transceiver (FIFO).

The optional channel access method (CCA) adds additional latency. This can be configured in the user settings. The default access method (ALOHA) does not add additional latency.

The wireless transmission starts as soon as the first data is available in the transceiver memory. During the continuous wireless transmission the remaining payload data is transmitted byte by byte.

On the receiver side, the FIFO is read as soon as an incoming packet is detected.

If the module detects a packet that requires an ACK, the ACK is sent directly after the packet reception. The channel access method is always deactivated for ACKs.

In combination with a suitable packet generation method, this procedure enables the minimisation of the latencies resulting from buffering.

According to ETSI EN Norms the time for one packet (including the respective acknowledge) should not exceed 100 ms. For slow data rates the packet size has to be reduced respectively by the user.

10 Radio parameters

The RF parameters are configured with the `RF_ConfigIndex` as well as with `PHY_DefaultChannel` and `PHY_PAPower`.

Furthermore, the volatile runtime parameters can be configured over the respective command during runtime. This leads to the adoption of this parameters with the next packet transmit or receive.



The parameters must be chosen with prudence to reach good functionality and compliance to active local regulatory (e.g. EN 300 220 in Europe).



The module is tested through an accredited test lab to conform to the requirements of the RED. The module complies to receiver category 2.

That means at 50Ω conducted on the corresponding evaluation board all requirements are met. Decisive for the end product is the effective radiated power. The power depends on the selected antenna, the wiring to the antenna, the quality of the power supply, possible disturbances etc., and should therefore be tested.

An important aspect to comply with the radio regulatory is to adhere to the requirements of the duty cycle. The duty cycle is the ratio expressed as a percentage of the cumulative duration of transmission $T_{on,cum}$ within an observation time interval of T_{obs} . $DC = (T_{on,cum}/T_{obs})F_{obs}$ on an observation bandwidth F_{obs} . Unless otherwise specified, T_{obs} is 1 hour and the observation bandwidth F_{obs} is the operational frequency band.



There are no mechanisms for constraining the duty cycle in the firmware. The customer is fully responsible for the compliance of the duty cycle.

| RF_ConfigIndex | Data rate (gross) [kBaud] | Modulation | Wideband WB/ Narrowband NB |
|----------------|------------------------------|------------|-------------------------------|
| 0 | 1.2 | 2-GFSK | NB |
| 1 | 2.4 | 2-GFSK | NB |
| 2 | 9.6 | 2-GFSK | WB |
| 3 ¹ | 19.2 | 4-GFSK | WB |
| 4 | 25 | 2-GFSK | WB |

Table 14: RF parameters

¹ Profile 3 is obsolete, not pre-certified and will be removed in future releases.

10.1 Channel assignment and requirements in Band D



The hereafter shown channels were tested at 50Ω tethered to satisfy the requirements of the EN 300 220. Other channels are not allowed to be chosen. Determining for the legitimacy is the finally effective radiated power, that is dependent on the utilized antenna and the environmental conditions. So the radiated power should be verified.

| Band | Channel No. | Frequency [MHz] | 1.2 kBaud | 2.4 kBaud | 9.6 kBaud | 25 kBaud |
|---|-------------|-----------------|-----------|-----------|-----------|----------|
| OCW | | | 12.5 kHz | 12.5 kHz | 50 kHz | 50 kHz |
| Band D: 169.400 MHz - 169.475 MHz output power ≤ 27 dBm $\leq 1\%$ DC | Border | 169.4000 | no | no | no | no |
| | 0 | 169.4125 | yes | yes | no | no |
| | 1 | 169.4250 | yes | yes | * | * |
| | 2 | 169.4375 | yes | yes | yes | yes |
| | 3 | 169.4500 | yes | * | no | * |
| | 4 | 169.4625 | yes | yes | no | no |
| | Border | 169.4750 | no | no | no | no |

Table 15: Channel overview, narrow band in 12.5 kHz channel spacing, predefined channel is bolded

"yes" means that the channel in general is allowed, "no" means that using the channel would not conform to EN 300 220.

There is an additional less strict duty cycle for metering devices $\leq 10\%$.

"" means, that the channel in general is allowed, but the above mentioned channel spacing must be fulfilled

11 Battery powered operation

The *TRX_DISABLE* pin can set the module to one of two different modes of operation.

11.1 Active mode

When *TRX_DISABLE* is low, the module is permanently ready to receive and forward data via UART or wireless transmission. The module will switch to one of the internal LPM after having processed any pending data transmission, i.e. */RTS* must be low.

11.2 Stand-by mode

When *TRX_DISABLE* is high, the operation of the module's transceiver is disabled. Wireless reception is not possible, but transmission of data is possible. The module will switch to one of the internal LPM as long as no data will be transmitted. A UART data rate of more than 9600 baud will result in a higher current consumption.



The *CfgFlags* Bit 4 can modify this behavior (see chapter 8.22).

12 Custom firmware

12.1 Custom configuration of standard firmware

The configuration of standard firmware includes adoption of the non-volatile User settings (see chapter 8) to customer requirements and creating a customized product on base of the standard product with a unique ordering number for a specific customer that needs this configuration.

For example if the UART baud rate shall be changed from the default value to another value. This variant will result in a customer exclusive module with a unique ordering number. This will also fix the firmware version to a specific and customer tested version and thus results in a customer exclusive module with a unique ordering number.

Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.

12.2 Customer specific firmware

A customer specific firmware may include "Custom configuration of standard firmware" plus additional options or functions and tasks that are customer specific and not part of the standard firmware.

Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.

This also results in a customer exclusive module with a unique ordering number.

An example for this level of customization are functions like host-less operation where the module will perform data generation (e.g. by reading a SPI or I²C sensor) and cyclic transmission of this data to a data collector while sleeping or being passive most of the time.

Also replacing UART with SPI as host communication interface is classified such a custom specific option.

Certification critical changes need to be re-evaluated by an external qualified measurement laboratory. These critical changes may occur when e.g. changing radio parameters, the channel access method, the duty-cycle or in case of various other functions and options possibly used or changed by a customer specific firmware.

12.3 Customer firmware

A customer firmware is a firmware written and tested by the customer himself or a 3rd party as a customer representative specifically for the hardware platform provided by a module.

This customer firmware (e.g. in form of a Intel hex file) will be implemented into the module's production process at our production side.

This also results in a customer exclusive module with a unique ordering number.

The additional information needed for this type of customer firmware, such as hardware specific details and details towards the development of such firmware are not available for the public and can only be made available to qualified customers.



The qualification(s) and certification(s) of the standard firmware cannot be applied to this customer firmware solution without a review and verification.

12.4 Contact for firmware requests

Please contact your local field sales engineer (FSE) or wireless-sales@we-online.com for quotes regarding this topics.

13 Firmware update



We highly recommend to have pads/connectors for realizing these (external) uart connection on any customer PCB.

13.1 Update using UART interface

As long as a firmware is running on the module the module can be updated with the PC utility "AMBER Config Center" (ACC V3) via the serial interface.

If the module is not directly connected to a PC, the UART should be made accessible, e.g. by means of suitable connectors. Only the *UTDX*, *URXD* and *GND* signals are needed for this connection. An adapter is required for a PC connection (e.g. the FTDI TTL-323R-3V3 UART to USB converter).

The */RESET* signal shall be connectable to GND for performing a reset of the module (e.g. using a push-button which pulls to *GND* when pressed)

13.2 Update using JTAG or Spy-Bi-Wire

Using one of this two interface options allows performing a fail-safe firmware update even in case of a broken firmware or malconfiguration.

The user needs hardware and software tools to be able to perform this procedure. In detail those are:

- Flash adapter for MSP430 μ C's (e.g. from TI, Elprotronic or Olimex), caution: not every adapter supports both described connection methods. Recommended adapter: "Elprotronic Flash Pro 430"
- In case of SPY-Bi-Wire a dedicated connector with some passive parts is needed (see the documentation of the flash adapter you use) this connector may vary from μ C to μ C
- In general */Reset*, *GND* and *VCC* are needed for such connections
- JTAG is supported through the module's pads 6(J.0), 7(J.1), 8(J.2), 18(J.3)
- SPY-Bi-wire is supported through pad 17 (TEST pin of the μ C)

The manual of the EV-Board gives an example of a JTAG connection with a 2*7 pin connector for the MSP430F2xxx and MSP430F5xxx Platforms.

14 Firmware history

Version 3.3 "Production"

- First release

Version 3.3.6 "Production"

- Minor updates
- RF profile 4 added as replacement for profile 3. (better data rate + better range)

Version 3.5.0 "Production"

- Code Maintenance
- Adopted to new version numbering requirement
- Unused module internal only pin termination changed.
- SPI line handling towards radio IC adopted to new method.

15 Design in guide

15.1 Advice for schematic and layout

For users with less RF experience it is advisable to closely copy the relating evaluation board with respect to schematic and layout, as it is a proven design. The layout should be conducted with particular care, because even small deficiencies could affect the radio performance and its range or even the conformity.

The following general advice should be taken into consideration:

- A clean, stable power supply is strongly recommended. Interference, especially oscillation can severely restrain range and conformity.
- Variations in voltage level should be avoided.
- LDOs, properly designed in, usually deliver a proper regulated voltage.
- Blocking capacitors and a ferrite bead in the power supply line can be included to filter and smoothen the supply voltage when necessary.



No fixed values can be recommended, as these depend on the circumstances of the application (main power source, interferences etc.).



Frequently switching the module on and off, especially with a slowly changing voltage level of the power supply, can lead to erratic behavior, in rare cases even as far as damaging the module or the firmware. The use of an external reset IC can solve this matter and shall be considered especially in battery operated scenarios.

- Elements for ESD protection should be placed on all pins that are accessible from the outside and should be placed close to the accessible area. For example, the RF-pin is accessible when using an external antenna and should be protected.
- ESD protection for the antenna connection must be chosen such as to have a minimum effect on the RF signal. For example, a protection diode with low capacitance such as the LXES15AAA1-100 or a 68 nH air-core coil connecting the RF-line to ground give good results.
- Placeholders for optional antenna matching or additional filtering are recommended.
- The antenna path should be kept as short as possible.



Again, no fixed values can be recommended, as they depend on the influencing circumstances of the application (antenna, interferences etc.).

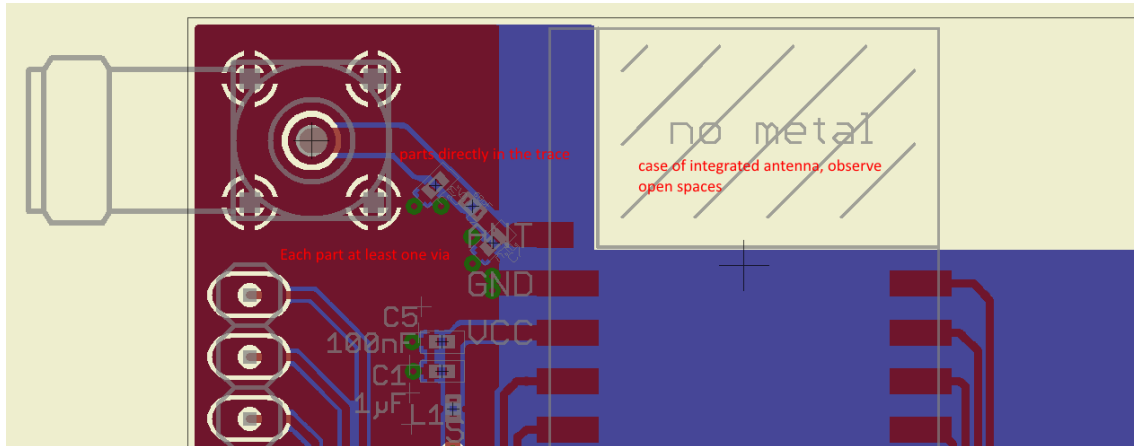


Figure 4: Layout

- To avoid the risk of short circuits and interference there should be no routing underneath the module on the top layer of the baseboard.
- On the second layer, a ground plane is recommended, to provide good grounding and shielding to any following layers and application environment.
- In case of integrated antennas it is required to have areas free from ground. This area should be copied from the evaluation board.
- The area with the integrated antenna must overlap with the carrier board and should not protrude, as it is matched to sitting directly on top of a PCB.
- Modules with integrated antennas should be placed with the antenna at the edge of the main board. It should not be placed in the middle of the main board or far away from the edge. This is to avoid tracks beside the antenna.
- Filter and blocking capacitors should be placed directly in the tracks without stubs, to achieve the best effect.
- Antenna matching elements should be placed close to the antenna / connector, blocking capacitors close to the module.
- Ground connections for the module and the capacitors should be kept as short as possible and with at least one separate through hole connection to the ground layer.
- ESD protection elements should be placed as close as possible to the exposed areas.

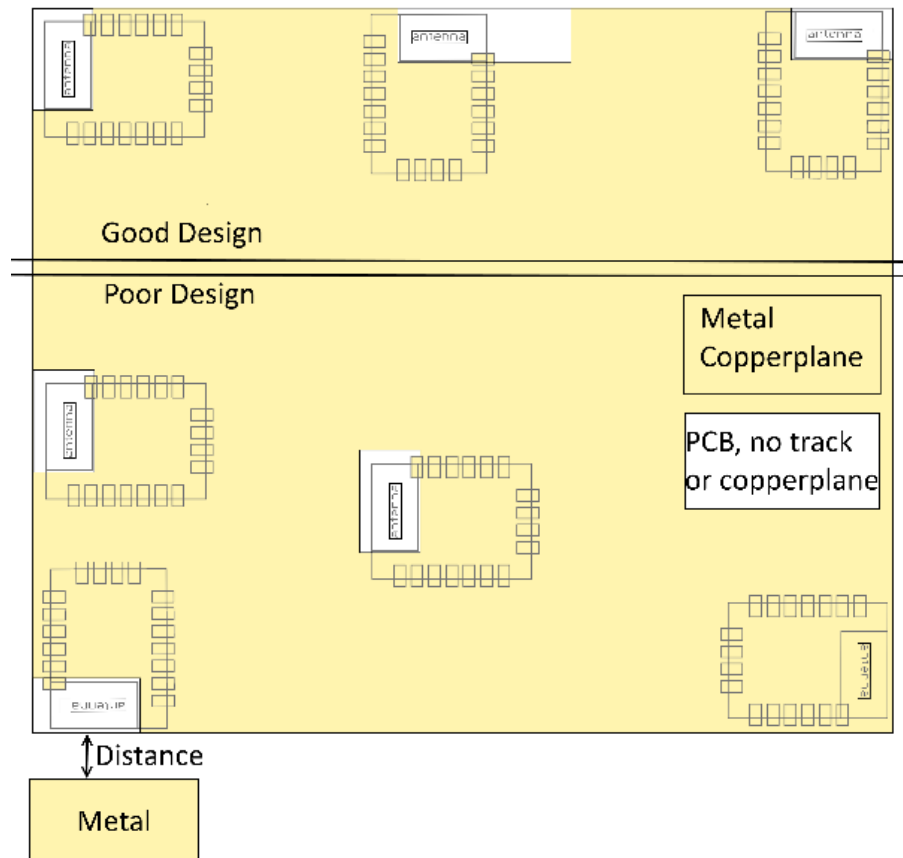


Figure 5: Placement of the module with integrated antenna

15.2 Dimensioning of the micro strip antenna line

The antenna track has to be designed as a 50Ω feed line. The width W for a micro strip can

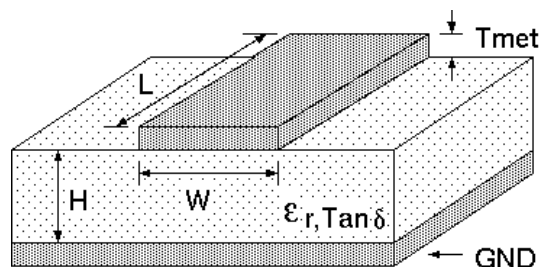


Figure 6: Dimensioning the antenna feed line as micro strip

be calculated using the following equation:

$$W = 1.25 \times \left(\frac{5.98 \times H}{e^{\frac{50 \times \sqrt{\epsilon_r + 1.41}}{87}}} - T_{met} \right) \quad (2)$$

Example:

A FR4 material with $\epsilon_r = 4.3$, a height $H = 1000 \mu\text{m}$ and a copper thickness of $T_{met} = 18 \mu\text{m}$

will lead to a trace width of $W \sim 1.9$ mm. To ease the calculation of the micro strip line (or e.g. a coplanar) many calculators can be found in the internet.

- As rule of thumb a distance of about $3 \times W$ should be observed between the micro strip and other traces / ground.
- The micro strip refers to ground, therefore there has to be the ground plane underneath the trace.
- Keep the feeding line as short as possible.

15.3 Antenna solutions

There exist several kinds of antennas, which are optimized for different needs. Chip antennas are optimized for minimal size requirements but at the expense of range, PCB antennas are optimized for minimal costs, and are generally a compromise between size and range. Both usually fit inside a housing.

Range optimization in general is at the expense of space. Antennas that are bigger in size, so that they would probably not fit in a small housing, are usually equipped with a RF connector. A benefit of this connector may be to use it to lead the RF signal through a metal plate (e.g. metal housing, cabinet).

As a rule of thumb a minimum distance of $\lambda/10$ (which is 3.5 cm @ 868 MHz and 1.2 cm @ 2.44 GHz) from the antenna to any other metal should be kept. Metal placed further away will not directly influence the behavior of the antenna, but will anyway produce shadowing.



Keep the antenna away from large metal objects as far as possible to avoid electromagnetic field blocking.



The choice of antenna might have influence on the safety requirements.

In the following chapters, some special types of antenna are described.

15.3.1 Wire antenna

An effective antenna is a $\lambda/4$ radiator with a suiting ground plane. The simplest realization is a piece of wire. It's length is depending on the used radio frequency, so for example 8.6 cm 868.0 MHz and 3.1 cm for 2.440 GHz as frequency. This radiator needs a ground plane at its feeding point. Ideally, it is placed vertically in the middle of the ground plane. As this is often not possible because of space requirements, a suitable compromise is to bend the wire away from the PCB respective to the ground plane. The $\lambda/4$ radiator has approximately 40 Ω input impedance, therefore matching is not required.

15.3.2 Chip antenna

There are many chip antennas from various manufacturers. The benefit of a chip antenna is obviously the minimal space required and reasonable costs. However, this is often at the expense of range. For the chip antennas, reference designs should be followed as closely as possible, because only in this constellation can the stated performance be achieved.

15.3.3 PCB antenna

PCB antenna designs can be very different. The special attention can be on the miniaturization or on the performance. The benefits of the PCB antenna are their small / not existing (if PCB space is available) costs, however the evaluation of a PCB antenna holds more risk of failure than the use of a finished antenna. Most PCB antenna designs are a compromise of range and space between chip antennas and connector antennas.

15.3.4 Antennas provided by Würth Elektronik eiSos

15.3.4.1 2600130011 - Helike - 169 MHz dipole antenna



Figure 7: 169 MHz dipole-antenna

| Specification | Value |
|--|------------|
| Frequency range [MHz] | 169 |
| Impedance [Ω] | 50 |
| VSWR | ≤ 2.1 |
| Gain [dBi] | 1 |
| Dimensions (L x d) [mm] | 320 x 15 |
| Weight [g] | 42 |
| Connector | SMA plug |
| Operating Temp. [$^{\circ}\text{C}$] | -40 – +85 |



This antenna requires a ground plane which will influence the electrical parameters.

15.3.4.2 2600130041 - Herse - 434 MHz dipole antenna



Figure 8: 434 MHz dipole-antenna

| Specification | Value |
|--|------------|
| Frequency range [MHz] | 433 |
| Impedance [Ω] | 50 |
| VSWR | ≤ 1.5 |
| Polarization | Vertical |
| Radiation | Omni |
| Gain [dBi] | 0 |
| Antenna Cover | TPEE |
| Dimensions (L x d) [mm] | 90 x 12 |
| Weight [g] | 9.6 |
| Connector | SMA plug |
| Operating Temp. [$^{\circ}\text{C}$] | -40 – +80 |



This antenna requires a ground plane which will influence the electrical parameters.

15.3.4.3 2600130081 - Hyperion-I - 868 MHz dipole antenna

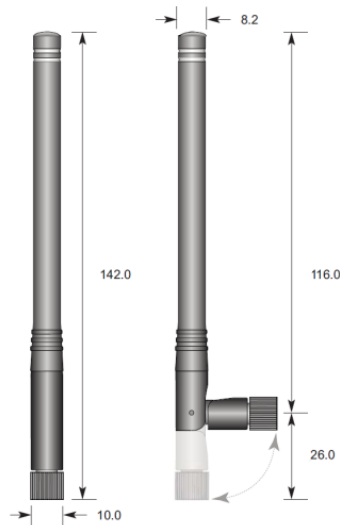


Figure 9: 868 MHz dipole-antenna

Ideally suited for applications where no ground plane is available.



The 2600130081 antenna can be also used for 902MHz - 928MHz range.

| Specification | Value |
|--|------------|
| Center frequency [MHz] | 868 |
| Frequency range [MHz] | 853 – 883 |
| Wavelength | 0.5 wave |
| VSWR | ≤ 2.0 |
| Impedance [Ω] | 50 |
| Connector | SMA (Male) |
| Dimensions (L x d) [mm] | 142 x 10 |
| Peak gain [dBi] | -2.3 |
| Operating temp. [$^{\circ}\text{C}$] | -30 – +80 |

15.3.4.4 2600130082 - Hyperion-II - 868 MHz magnetic base antenna

Well suited for applications where the RF is lead through a metal wall that could serve as ground plane to the antenna.

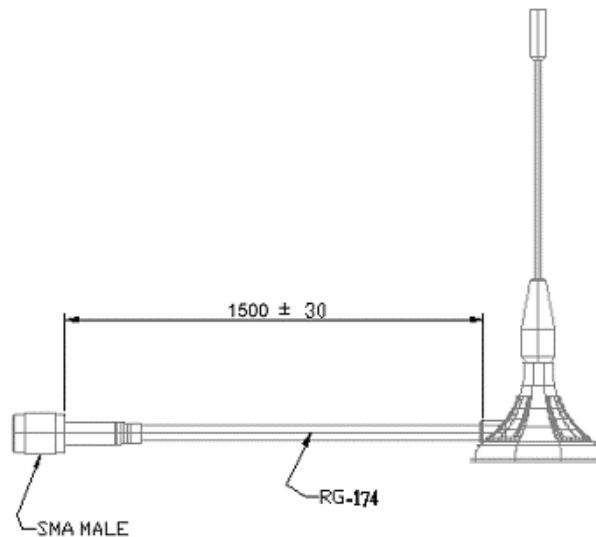


Figure 10: 868 MHz magnet foot antenna with 1.5 m antenna cable



The 2600130082 is a kind of $\lambda/4$ radiator and therefore needs a ground plane at the feeding point.

| Specification | Value |
|--|------------|
| Frequency range [MHz] | 824 – 894 |
| VSWR | ≤ 2.0 |
| Polarisation | Vertical |
| Impedance [Ω] | 50 ± 5 |
| Connector | SMA (Male) |
| Dimensions (L x d) [mm] | 89.8 x 27 |
| Weight [g] | 50 ± 5 |
| Operating temp. [$^{\circ}\text{C}$] | -30 – +60 |

15.3.4.5 2600130021 - Himalia - 2.4 GHz dipole antenna



Figure 11: 2.4 GHz dipole-antenna

Due to the fact, that the antenna has dipole topology there is no need for an additional ground plane. Nevertheless the specification was measured edge mounted and 90° bent on a 100 x 100 mm ground plane.

| Specification | Value |
|--|------------------|
| Frequency range [GHz] | 2.4 – 2.5 |
| Impedance [Ω] | 50 |
| VSWR | $\leq 2:1$ |
| Polarization | Linear |
| Radiation | Omni-Directional |
| Peak Gain [dBi] | 2.8 |
| Average Gain [dBi] | -0.6 |
| Efficiency | 85 % |
| Dimensions (L x d) [mm] | 83.1 x 10 |
| Weight [g] | 7.4 |
| Connector | SMA plug |
| Operating temp. [$^{\circ}\text{C}$] | -40 – +80 |

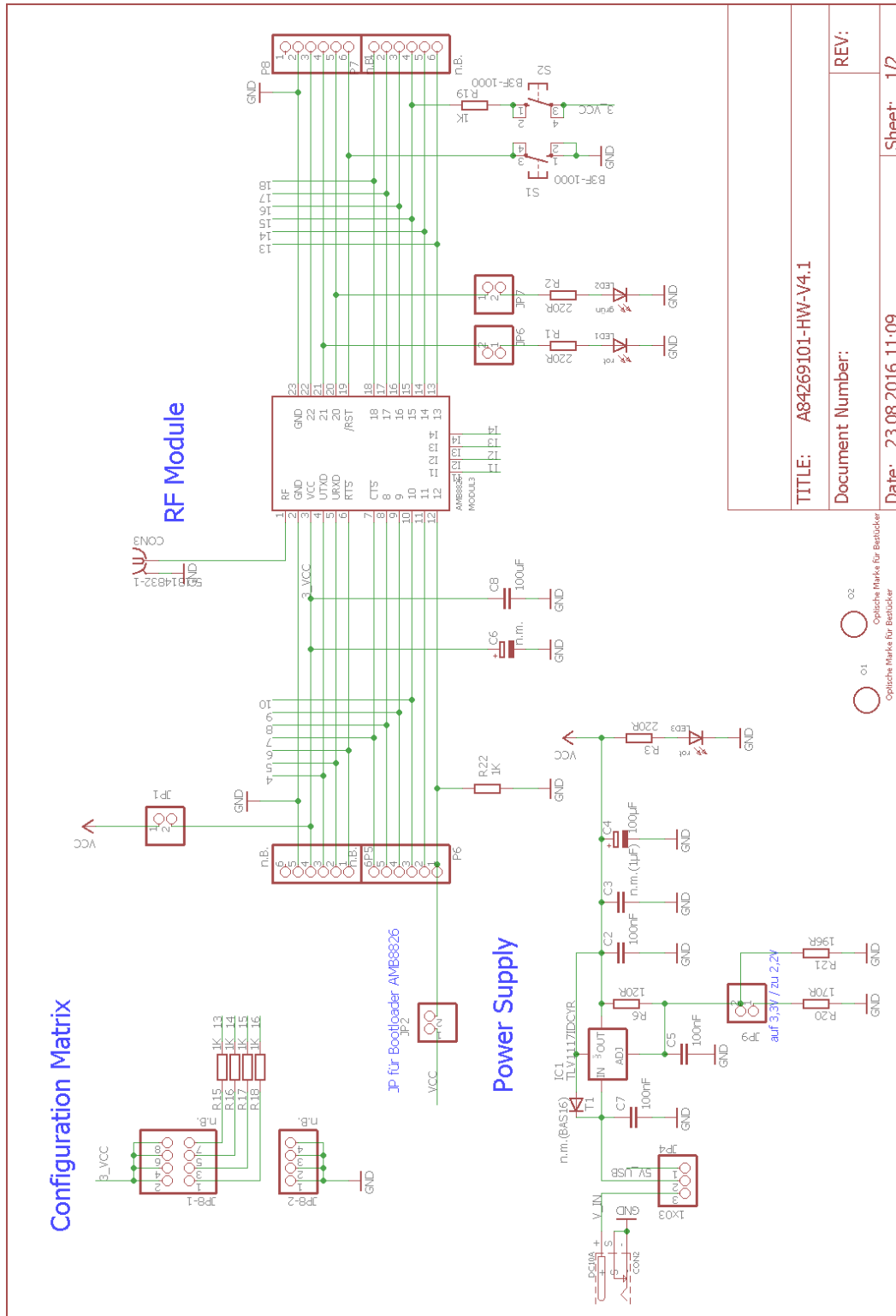
16 Reference design

Titania was tested and certified on the corresponding Titania evaluation board. For the compliance with the EU directive 2014/53/EU Annex I, the evaluation board serves as reference design.

This is no discrepancy due to the fact that the evaluation board itself does not fall within the scope of the EU directive 2014/53/EU Annex I as the module is tested on the evaluation board, which is also the recommended use.

Further information concerning the use of the evaluation board can be found in the manual of the Titania evaluation board.

16.1 Schematic



16.2 Layout

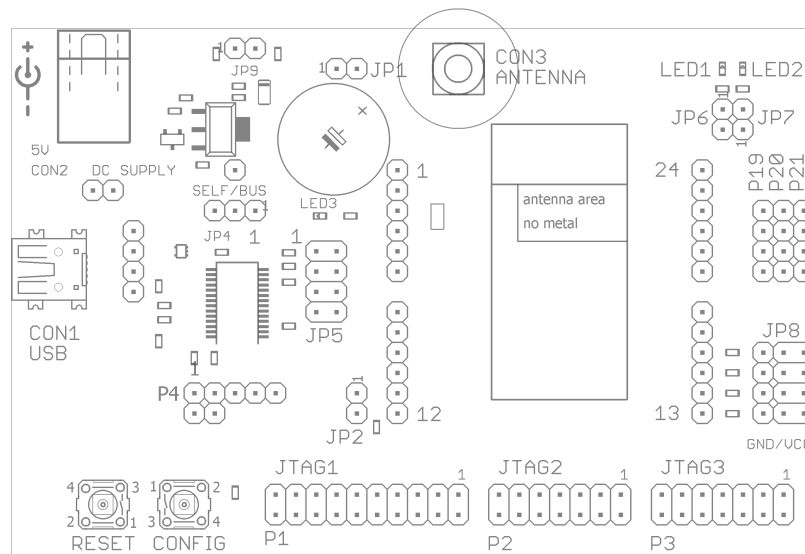


Figure 12: Assembly diagram

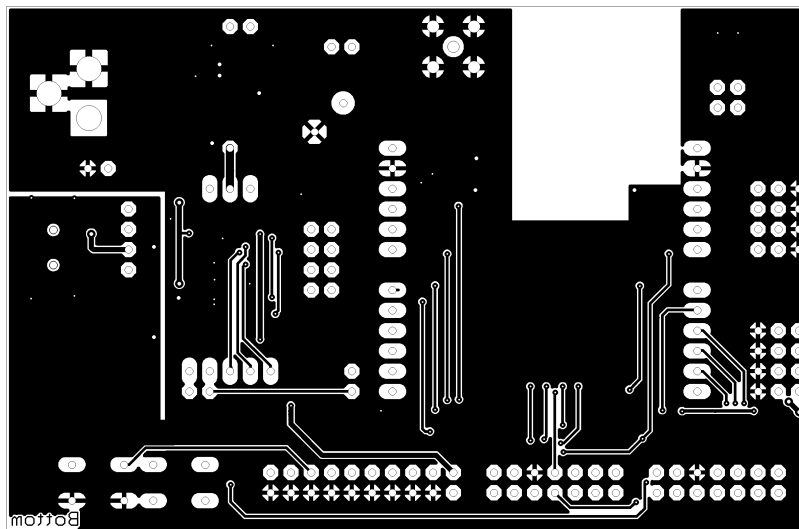
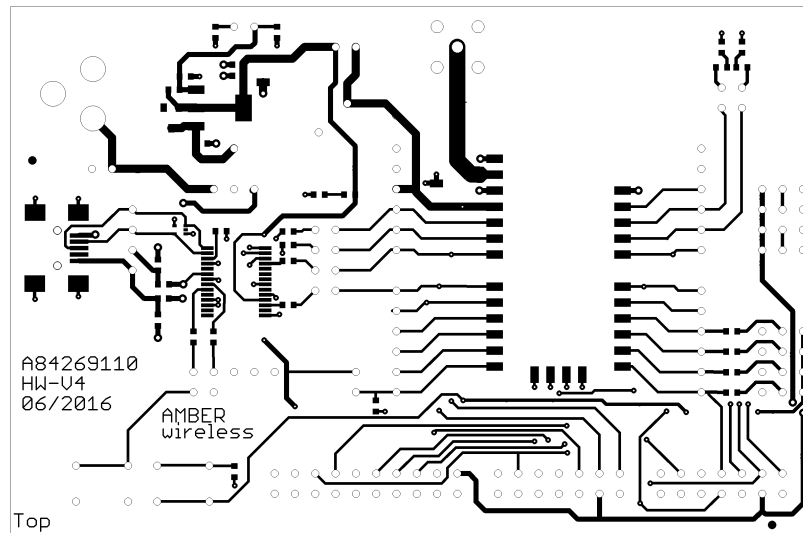


Figure 13: Top and Bottom Layer

17 Manufacturing information

17.1 Moisture sensitivity level

This wireless connectivity product is categorized as JEDEC Moisture Sensitivity Level 3 (MSL3), which requires special handling.

More information regarding the MSL requirements can be found in the IPC/JEDEC J-STD-020 standard on www.jedec.org.

More information about the handling, picking, shipping and the usage of moisture/reflow and/or process sensitive products can be found in the IPC/JEDEC J-STD-033 standard on www.jedec.org.

17.2 Soldering

17.2.1 Reflow soldering

Attention must be paid on the thickness of the solder resist between the host PCB top side and the modules bottom side. Only lead-free assembly is recommended according to JEDEC J-STD020.

| Profile feature | | Value |
|--|--------------|--------------------|
| Preheat temperature Min | $T_{S\ Min}$ | 150 °C |
| Preheat temperature Max | $T_{S\ Max}$ | 200 °C |
| Preheat time from $T_{S\ Min}$ to $T_{S\ Max}$ | t_S | 60 - 120 seconds |
| Ramp-up rate (T_L to T_P) | | 3 °C / second max. |
| Liquidous temperature | T_L | 217 °C |
| Time t_L maintained above T_L | t_L | 60 - 150 seconds |
| Peak package body temperature | T_P | see table below |
| Time within 5 °C of actual peak temperature | t_P | 20 - 30 seconds |
| Ramp-down Rate (T_P to T_L) | | 6 °C / second max. |
| Time 20 °C to T_P | | 8 minutes max. |

Table 16: Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E

| Package thickness | Volume mm ³ <350 | Volume mm ³ 350-2000 | Volume mm ³ >2000 |
|-------------------|--------------------------------|------------------------------------|---------------------------------|
| < 1.6mm | 260 °C | 260 °C | 260 °C |
| 1.6mm - 2.5mm | 260 °C | 250 °C | 245 °C |
| > 2.5mm | 250 °C | 245 °C | 245 °C |

Table 17: Package classification reflow temperature, PB-free assembly, Note: refer to IPC/-JEDEC J-STD-020E

It is recommended to solder this module on the last reflow cycle of the PCB. For solder paste use a LFM-48W or Indium based SAC 305 alloy (Sn 96.5 / Ag 3.0 / Cu 0.5 / Indium 8.9HF / Type 3 / 89%) type 3 or higher.

The reflow profile must be adjusted based on the thermal mass of the entire populated PCB, heat transfer efficiency of the reflow oven and the specific type of solder paste used. Based on the specific process and PCB layout the optimal soldering profile must be adjusted and verified. Other soldering methods (e.g. vapor phase) have not been verified and have to be validated by the customer at their own risk. Rework is not recommended.

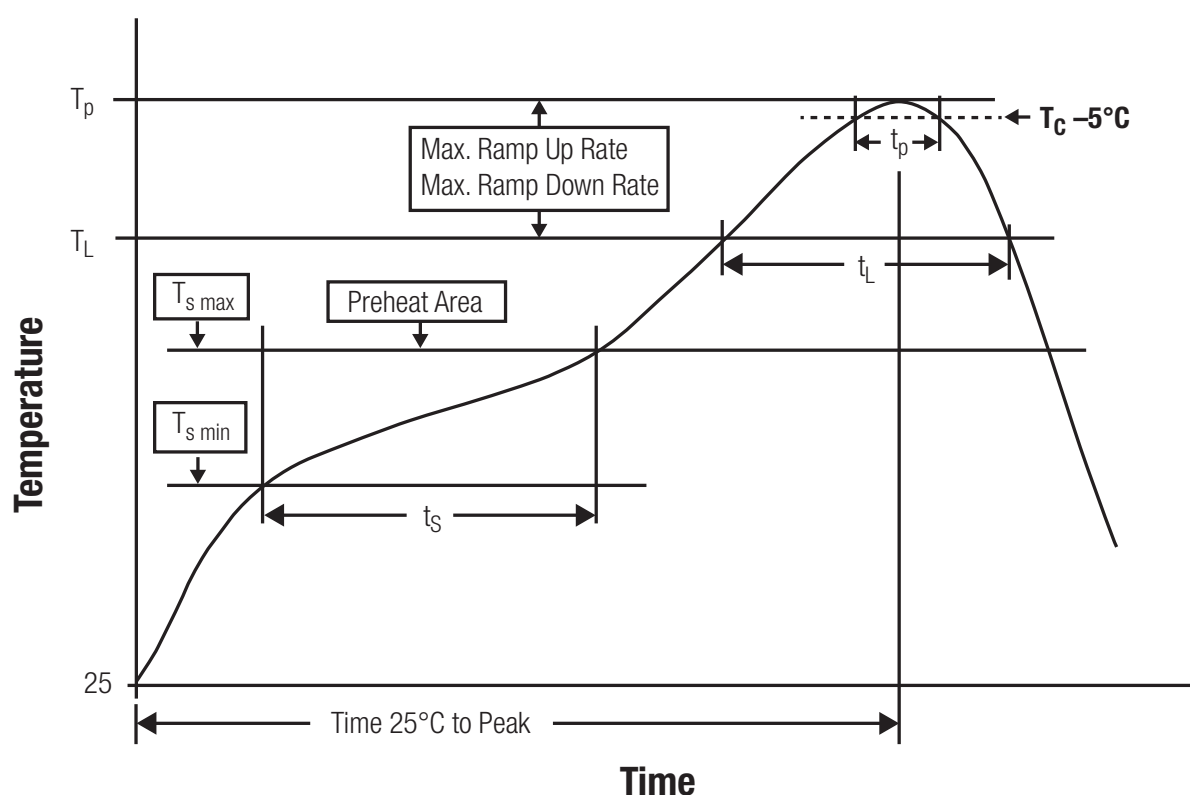


Figure 14: Reflow soldering profile

After reflow soldering, visually inspect the board to confirm proper alignment

17.2.2 Cleaning

Do not clean the product. Any residue cannot be easily removed by washing. Use a "no clean" soldering paste and do not clean the board after soldering.

- Do not clean the product with water. Capillary effects can draw water into the gap between the host PCB and the module, absorbing water underneath it. If water is trapped inside, it may short-circuit adjoining pads. The water may also destroy the label and ink-jet printed text on it.
- Cleaning processes using alcohol or other organic solvents may draw solder flux residues into the housing, which won't be detected in a post-wash inspection. The solvent may also destroy the label and ink-jet printed text on it.
- Do not use ultrasonic cleaning as it will permanently damage the part, particularly the crystal oscillators.

17.2.3 Other notations

- Conformal coating of the product will result in the loss of warranty. The RF shields will not protect the part from low-viscosity coatings.
- Do not attempt to improve the grounding by forming metal strips directly to the EMI covers or soldering on ground cables, as it may damage the part and will void the warranty.
- Always solder every pad to the host PCB even if some are unused, to improve the mechanical strength of the module.
- The part is sensitive to ultrasonic waves, as such do not use ultrasonic cleaning, welding or other processing. Any ultrasonic processing will void the warranty.

17.3 ESD handling

This product is highly sensitive to electrostatic discharge (ESD). As such, always use proper ESD precautions when handling. Make sure to handle the part properly throughout all stages of production, including on the host PCB where the module is installed. For ESD ratings, refer to the module series' maximum ESD section. For more information, refer to the relevant chapter 2. Failing to follow the aforementioned recommendations can result in severe damage to the part.

- the first contact point when handling the PCB is always between the local GND and the host PCB GND, unless there is a galvanic coupling between the local GND (for example work table) and the host PCB GND.
- Before assembling an antenna patch, connect the grounds.
- While handling the RF pin, avoid contact with any charged capacitors and be careful when contacting any materials that can develop charges (for example coaxial cable with around 50-80 pF/m, patch antenna with around 10 pF, soldering iron etc.)

- Do not touch any exposed area of the antenna to avoid electrostatic discharge. Do not let the antenna area be touched in a non ESD-safe manner.
- When soldering, use an ESD-safe soldering iron.

17.4 Safety recommendations

It is your duty to ensure that the product is allowed to be used in the destination country and within the required environment. Usage of the product can be dangerous and must be tested and verified by the end user. Be especially careful of:

- Use in areas with risk of explosion (for example oil refineries, gas stations).
- Use in areas such as airports, aircraft, hospitals, etc., where the product may interfere with other electronic components.

It is the customer's responsibility to ensure compliance with all applicable legal, regulatory and safety-related requirements as well as applicable environmental regulations. Disassembling the product is not allowed. Evidence of tampering will void the warranty.

- Compliance with the instructions in the product manual is recommended for correct product set-up.
- The product must be provided with a consolidated voltage source. The wiring must meet all applicable fire and security prevention standards.
- Handle with care. Avoid touching the pins as there could be ESD damage.

Be careful when working with any external components. When in doubt consult the technical documentation and relevant standards. Always use an antenna with the proper characteristics.



Since the module itself is not fused the voltage supply shall be fed from a limited power source according to EN 62368-1 class PS1.



Modules with high output power of up to 500mW, as for example the Thebe family, generate a high amount of warmth while transmitting. The manufacturer of the end device must take care of potentially necessary actions for his application.

18 Physical dimensions

18.1 Dimensions

| Dimensions |
|------------------|
| 17 x 27 x 3.8 mm |

Table 18: Dimensions

18.2 Weight

| Weight |
|--------|
| 3g |

Table 19: Weight

18.3 Module drawing

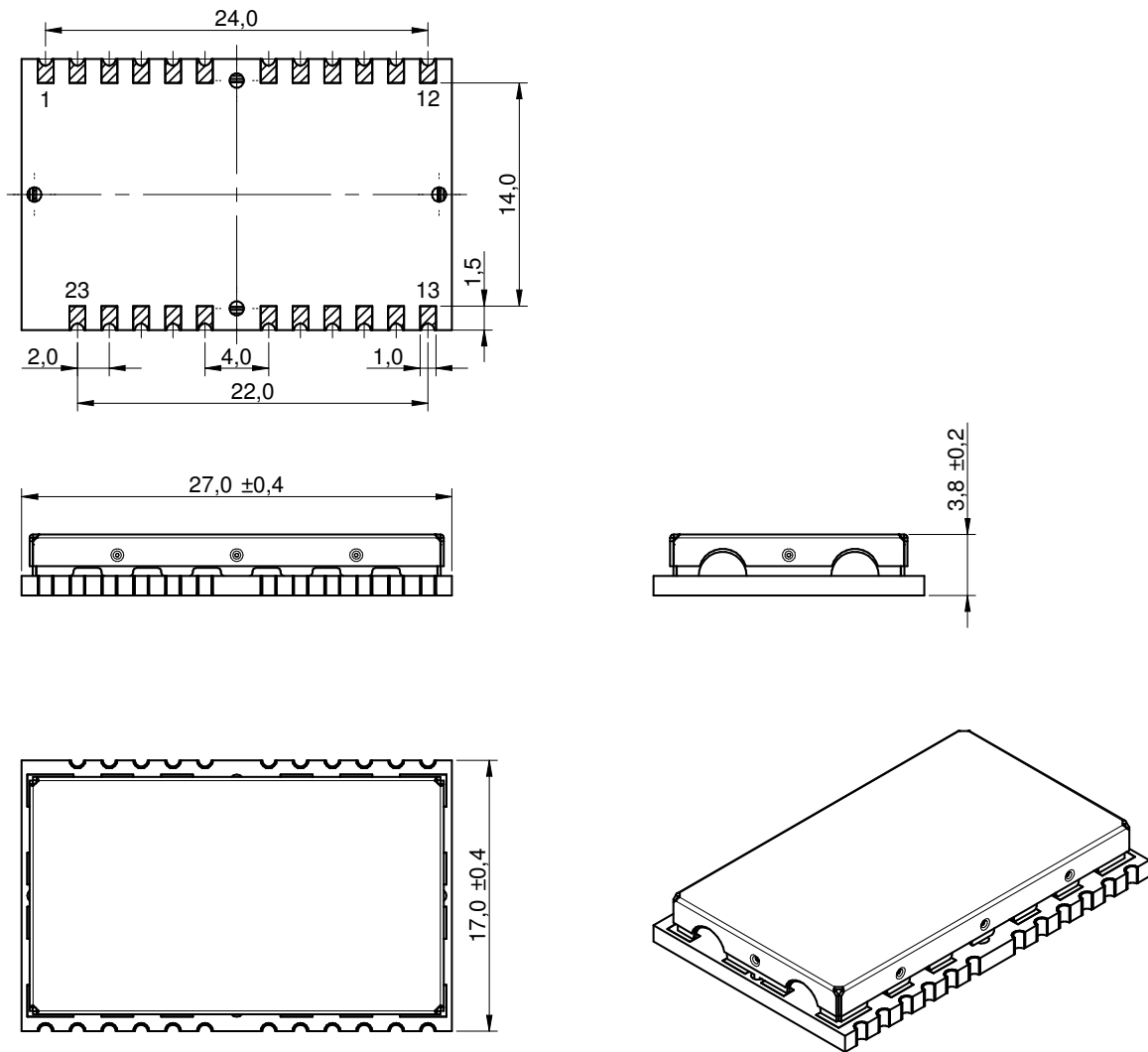


Figure 15: Module dimensions [mm]

18.4 Footprint

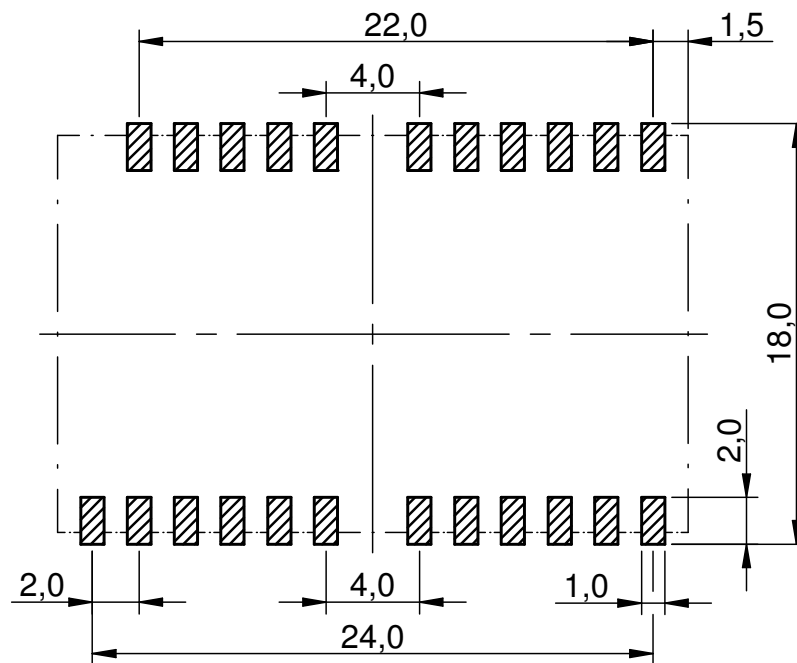


Figure 16: Footprint and dimensions [mm]

18.5 Antenna free area

To avoid influence and mismatching of the antenna the recommended free area around the antenna should be maintained. As rule of thumb a minimum distance of metal parts to the antenna of $\lambda/10$ should be kept (see figure 16). Even though metal parts would influence the characteristic of the antenna, but the direct influence and matching keep an acceptable level.

19 Marking

19.1 Lot number

The 15 digit lot number is printed in numerical digits as well as in form of a machine readable bar code. It is divided into 5 blocks as shown in the following picture and can be translated according to the following table.

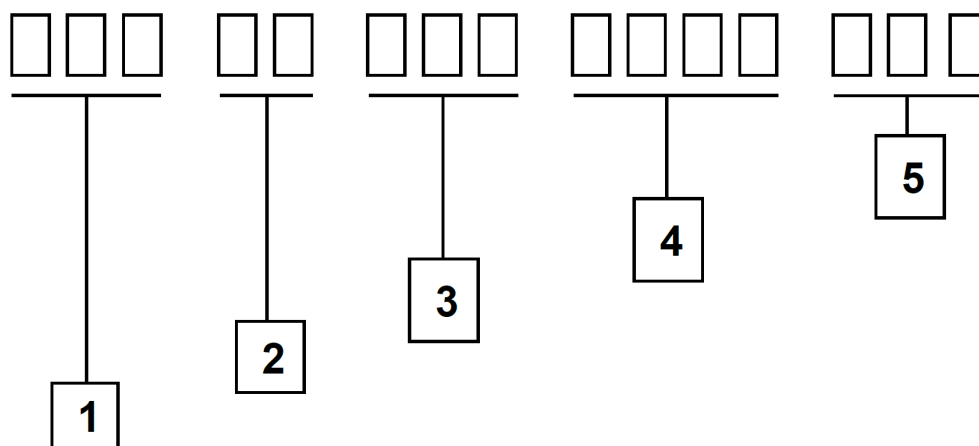


Figure 17: Lot number structure

| Block | Information | Example(s) |
|-------|----------------------------|---|
| 1 | eiSos internal, 3 digits | 439 |
| 2 | eiSos internal, 2 digits | 01 |
| 3 | Hardware version, 3 digits | V2.4 = 024, V12.2 = 122 |
| 4 | Date code, 4 digits | 1703 = week 03 in year 2017, 1816 = week 16 in year 2018 |
| 5 | Firmware version, 3 digits | V3.2 = 302, V5.13 = 513 |

Table 20: Lot number details

As the user can perform a firmware update the printed lot number only shows the factory delivery state. The currently installed firmware can be requested from the module using the corresponding product specific command. The firmware version as well as the hardware version are restricted to show only major and minor version not the patch identifier.

19.2 General labeling information

The module labels may include the following fields:

- Manufacturer identification WE, Würth Elektronik or Würth Elektronik eiSos
- WE Order Code and/or article alias
- Serial number or MAC address
- Certification identifiers (CE, FCC ID, IC, ARIB,...)
- Barcode or 2D code containing the serial number or MAC address

The serial number includes the product ID (PID) and an unique 6 digit number. The first 1 to 3 digits represent the PID, then the "." marks the start of the 6 digit counter to create a unique product marking.

In case of small labels, the 3 byte manufacturer identifier (0x0018DA) of the MAC address is not printed on the labels. The 3 byte counter printed on the label can be used with this 0018DA to produce the full MAC address by appending the counter after the manufacturer identifier.

19.2.1 Example labels of Würth Elektronik eiSos products

2603011021001 C E
FCC ID: R7TAMB2220
IC: 5136A-AMB2220
SN: 107.002005

AMB2621
SN: 0A6495
FCCID: C E
R7TAMB2621

 **WE** C E
SN: 116.002641
2609011081001
AMB8826

Würth Elektronik
FCC ID: R7TAMB9826
IC: 5136A-AMB9826
SN: 124.000323
2609011091001

20 Information for Ex Protection

In case the end product should be used in Ex protection areas the following information can be used:

- The module itself is unfused.
- The maximum output power of the module is 15dBm.
- The total amount of capacitance of all capacitors is 1.607 μ F.
- The total amount of inductance of all inductors is 677nH.

21 Regulatory compliance information

21.1 Important notice EU

The use of RF frequencies is limited by national regulations. The Titania has been designed to comply with the R&TTE directive 1999/5/EC and the RED directive 2014/53/EU of the European Union (EU).

The Titania can be operated without notification and free of charge in the area of the European Union. However, according to the R&TTE / RED directive, restrictions (e.g. in terms of duty cycle or maximum allowed RF power) may apply.

21.2 Conformity assessment of the final product

The Titania is a subassembly. It is designed to be embedded into other products (products incorporating the Titania are henceforward referred to as "final products").

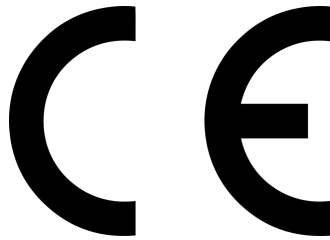
It is the responsibility of the manufacturer of the final product to ensure that the final product is in compliance with the essential requirements of the underlying national radio regulations. The conformity assessment of the subassembly Titania carried out by Würth Elektronik eiSos does not replace the required conformity assessment of the final product.

21.3 Exemption clause

Relevant regulation requirements are subject to change. Würth Elektronik eiSos does not guarantee the accuracy of the before mentioned information. Directives, technical standards, procedural descriptions and the like may be interpreted differently by the national authorities. Equally, the national laws and restrictions may vary with the country. In case of doubt or uncertainty, we recommend that you consult with the authorities or official certification organizations of the relevant countries. Würth Elektronik eiSos is exempt from any responsibilities or liabilities related to regulatory compliance.

Notwithstanding the above, Würth Elektronik eiSos makes no representations and warranties of any kind related to their accuracy, correctness, completeness and/or usability for customer applications. No responsibility is assumed for inaccuracies or incompleteness.

21.4 EU Declaration of conformity



EU DECLARATION OF CONFORMITY

Radio equipment: 2607011111000 & 2607046211001

The manufacturer: Würth Elektronik eiSos GmbH & Co. KG
Max-Eyth-Straße 1
74638 Waldenburg

This declaration of conformity is issued under the sole responsibility of the manufacturer.

Object of the declaration: 2607011111000 & 2607046211001

The object of the declaration described above is in conformity with the relevant Union harmonisation legislation: Directive 2014/53/EU and 2011/65/EU.

Following harmonised norms or technical specifications have been applied:

EN 300 220-1 V3.1.1 (2017-02)
EN 300 220-2 V3.1.1 (2017-02)
EN 301 489-1 V2.2.0 (Draft)
EN 301 489-3 V2.1.1 (Final draft)
EN 62311:2008
EN 60950-1: 2006 + A11: 2009 + A1 2010 + A12: 2011 + A2: 2013

i.A. G. Exler

Trier, 6th of November 2018
Place and date of issue

22 Important information

The following conditions apply to all goods within the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG :

22.1 General customer responsibility

Some goods within the product range of Würth Elektronik eiSos GmbH & Co. KG contain statements regarding general suitability for certain application areas. These statements about suitability are based on our knowledge and experience of typical requirements concerning the areas, serve as general guidance and cannot be estimated as binding statements about the suitability for a customer application. The responsibility for the applicability and use in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to the customer to evaluate, where appropriate to investigate and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for the respective customer application or not. Accordingly, the customer is cautioned to verify that the documentation is current before placing orders.

22.2 Customer responsibility related to specific, in particular safety-relevant applications

It has to be clearly pointed out that the possibility of a malfunction of electronic components or failure before the end of the usual lifetime cannot be completely eliminated in the current state of the art, even if the products are operated within the range of the specifications. The same statement is valid for all software and firmware parts contained in or used with or for products in the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG . In certain customer applications requiring a high level of safety and especially in customer applications in which the malfunction or failure of an electronic component could endanger human life or health, it must be ensured by most advanced technological aid of suitable design of the customer application that no injury or damage is caused to third parties in the event of malfunction or failure of an electronic component.

22.3 Best care and attention

Any product-specific datasheets, manuals, application notes, PCN's, warnings and cautions must be strictly observed in the most recent versions and matching to the products firmware revisions. This documents can be downloaded from the product specific sections on the wireless connectivity homepage.

22.4 Customer support for product specifications

Some products within the product range may contain substances, which are subject to restrictions in certain jurisdictions in order to serve specific technical requirements. Necessary information is available on request. In this case, the field sales engineer or the internal sales person in charge should be contacted who will be happy to support in this matter.

22.5 Product improvements

Due to constant product improvement, product specifications may change from time to time. As a standard reporting procedure of the Product Change Notification (PCN) according to the JEDEC-Standard, we inform about major changes in hard- or firmware. In case of further queries regarding the PCN, the field sales engineer, the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 22.1 and 22.2 remains unaffected.

22.6 Product life cycle

Due to technical progress and economical evaluation we also reserve the right to discontinue production and delivery of products. As a standard reporting procedure of the Product Termination Notification (PTN) according to the JEDEC-Standard we will inform at an early stage about inevitable product discontinuance. According to this, we cannot ensure that all products within our product range will always be available. Therefore, it needs to be verified with the field sales engineer or the internal sales person in charge about the current product availability expectancy before or when the product for application design-in disposal is considered. The approach named above does not apply in the case of individual agreements deviating from the foregoing for customer-specific products.

22.7 Property rights

All the rights for contractual products produced by Würth Elektronik eiSos GmbH & Co. KG on the basis of ideas, development contracts as well as models or templates that are subject to copyright, patent or commercial protection supplied to the customer will remain with Würth Elektronik eiSos GmbH & Co. KG. Würth Elektronik eiSos GmbH & Co. KG does not warrant or represent that any license, either expressed or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, application, or process in which Würth Elektronik eiSos GmbH & Co. KG components or services are used.

22.8 General terms and conditions

Unless otherwise agreed in individual contracts, all orders are subject to the current version of the "General Terms and Conditions of Würth Elektronik eiSos Group", last version available at www.we-online.com.

23 Legal notice

23.1 Exclusion of liability

Würth Elektronik eiSos GmbH & Co. KG considers the information in this document to be correct at the time of publication. However, Würth Elektronik eiSos GmbH & Co. KG reserves the right to modify the information such as technical specifications or functions of its products or discontinue the production of these products or the support of one of these products without any written announcement or notification to customers. The customer must make sure that the information used corresponds to the latest published information. Würth Elektronik eiSos GmbH & Co. KG does not assume any liability for the use of its products. Würth Elektronik eiSos GmbH & Co. KG does not grant licenses for its patent rights or for any other of its intellectual property rights or third-party rights. Notwithstanding anything above, Würth Elektronik eiSos GmbH & Co. KG makes no representations and/or warranties of any kind for the provided information related to their accuracy, correctness, completeness, usage of the products and/or usability for customer applications. Information published by Würth Elektronik eiSos GmbH & Co. KG regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof.

23.2 Suitability in customer applications

The customer bears the responsibility for compliance of systems or units, in which Würth Elektronik eiSos GmbH & Co. KG products are integrated, with applicable legal regulations. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Würth Elektronik eiSos GmbH & Co. KG components in its applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos GmbH & Co. KG. Customer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences lessen the likelihood of failures that might cause harm and take appropriate remedial actions. The customer will fully indemnify Würth Elektronik eiSos GmbH & Co. KG and its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

23.3 Trademarks

AMBER wireless is a registered trademark of Würth Elektronik eiSos GmbH & Co. KG. All other trademarks, registered trademarks, and product names are the exclusive property of the respective owners.

23.4 Usage restriction

Würth Elektronik eiSos GmbH & Co. KG products have been designed and developed for usage in general electronic equipment only. This product is not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the product is reasonably expected to cause severe personal injury or death, unless the parties have executed an agreement specifically governing such use. Moreover,

Würth Elektronik eiSos GmbH & Co. KG products are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. . Würth Elektronik eiSos GmbH & Co. KG must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every electronic component, which is used in electrical circuits that require high safety and reliability function or performance. By using Würth Elektronik eiSos GmbH & Co. KG products, the customer agrees to these terms and conditions.

24 License agreement for Würth Elektronik eiSos GmbH & Co. KG connectivity product firmware and software

Agreement between You and Würth Elektronik eiSos GmbH & Co. KG

The following terms of this license agreement for the usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity product firmware are a legal agreement between you and Würth Elektronik eiSos GmbH & Co. KG and/or its subsidiaries and affiliates (collectively, "Würth Elektronik eiSos "). You hereby agree that this license agreement is applicable to the product and the incorporated software and firmware (collectively, "Firmware") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form.

The Firmware included in any Würth Elektronik eiSos wireless connectivity product is purchased to you on the condition that you accept the terms and conditions of this license agreement. You agree to comply with all provisions under this license agreement.

24.1 Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the Firmware under the conditions that will be set forth in this license agreement. You are free to use the provided Firmware only in connection with one of the products from Würth Elektronik eiSos to the extent described in this license agreement. You are not entitled to change or alter the provided Firmware.

You are not entitled to transfer the Firmware in any form to third parties without prior written consent of Würth Elektronik eiSos .

You are not allowed to reproduce, translate, reverse engineer, read out, decompile, disassemble or create derivative works of the incorporated Firmware in whole or in part.

No more extensive rights to use and exploit the Firmware granted to you.

24.2 Usage and obligations

The responsibility for the applicability and use of the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to you to evaluate and investigate, where appropriate, and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for your respective application or not.

You are responsible for using the Würth Elektronik eiSos Product with the incorporated Firmware in compliance with all applicable product liability and product safety laws. You acknowledge to minimize the risk of loss and harm to individuals and bear the risk for failure leading to personal injury or death due to your usage of the product.

Würth Elektronik eiSos ' products with the incorporated Firmware are not authorized for use in safety-critical applications, or where a failure of the product is reasonably expected to cause severe personal injury or death. Moreover, Würth Elektronik eiSos ' products with the incorporated Firmware are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train

control, ship control), transportation signal, disaster prevention, medical, public information network etc. You shall inform Würth Elektronik eiSos about the intent of such usage before design-in stage. In certain customer applications requiring a very high level of safety and in which the malfunction or failure of an electronic component could endanger human life or health, you must ensure to have all necessary expertise in the safety and regulatory ramifications of your applications. You acknowledge and agree that you are solely responsible for all legal, regulatory and safety-related requirements concerning your products and any use of Würth Elektronik eiSos' products with the incorporated Firmware in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos. **YOU SHALL INDEMNIFY WÜRTH ELEKTRONIK EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF WÜRTH ELEKTRONIK EISOS' PRODUCTS WITH THE INCORPORATED FIRMWARE IN SUCH SAFETY-CRITICAL APPLICATIONS.**

24.3 Ownership

The incorporated Firmware created by Würth Elektronik eiSos is and will remain the exclusive property of Würth Elektronik eiSos.

24.4 Firmware update(s)

You have the opportunity to request the current and actual firmware for a bought wireless connectivity Product within the time of warranty. However, Würth Elektronik eiSos has no obligation to update a modules firmware in their production facilities, but can offer this as a service on request. The upload of firmware updates falls within your responsibility, e.g. via ACC or another software for firmware updates. Firmware updates will not be communicated automatically. It is within your responsibility to check the current version of a firmware in the latest version of the product manual on our website. The revision table in the product manual provides all necessary information about firmware updates. There is no right to be provided with binary files, so called "firmware images", those could be flashed through JTAG, SWD, Spi-Bi-Wire, SPI or similar interfaces.

24.5 Disclaimer of warranty

THE FIRMWARE IS PROVIDED "AS IS". YOU ACKNOWLEDGE THAT WÜRTH ELEKTRONIK EISOS MAKES NO REPRESENTATIONS AND WARRANTIES OF ANY KIND RELATED TO, BUT NOT LIMITED TO THE NON-INFRINGEMENT OF THIRD PARTIES' INTELLECTUAL PROPERTY RIGHTS OR THE MERCHANTABILITY OR FITNESS FOR YOUR INTENDED PURPOSE OR USAGE. WÜRTH ELEKTRONIK EISOS DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS IN WHICH THE WÜRTH ELEKTRONIK EISOS' PRODUCT WITH THE INCORPORATED FIRMWARE IS USED. INFORMATION PUBLISHED BY WÜRTH ELEKTRONIK EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WÜRTH ELEKTRONIK EISOS TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

24.6 Limitation of liability

Any liability not expressly provided by Würth Elektronik eiSos shall be disclaimed. You agree to hold us harmless from any third-party claims related to your usage of the Würth Elektronik eiSos ' products with the incorporated Firmware. Würth Elektronik eiSos disclaims any liability for any alteration, development created by you or your customers as well as for any combination with other products.

24.7 Applicable law and jurisdiction

Applicable law to this license agreement shall be the laws of the Federal Republic of Germany. Any dispute, claim or controversy arising out of or relating to this license agreement shall be resolved and finally settled by the court competent for the location of Würth Elektronik eiSos ' registered office.

24.8 Severability clause

If a provision of this license agreement is or becomes invalid, unenforceable or null and void, this shall not affect the remaining provisions of the agreement. The parties shall replace any such provisions with new valid provisions that most closely approximate the purpose of the agreement.

24.9 Miscellaneous

This license agreement constitutes the entire understanding and merges all prior discussions between the parties relating to this license agreement.

No ancillary verbal agreements have been made and no such agreements shall be valid. Any additions and amendments to this license agreement shall require the written form in order to be binding.

We recommend you to be updated about the status of new firmware, which is available on our website or in our data sheet, and to implement new firmware in your device where appropriate. In case only firmware is provided, we expressly exclude the automatic receipt of PCN information. Thus, new firmware will also not be provided automatically.

By ordering a wireless connectivity Product, you accept this license agreement in all terms.

List of Figures

| | | |
|----|--|-----|
| 1 | Block diagram | 9 |
| 2 | Pinout | 14 |
| 3 | Power up | 17 |
| 4 | Layout | 88 |
| 5 | Placement of the module with integrated antenna | 89 |
| 6 | Dimensioning the antenna feed line as micro strip | 89 |
| 7 | 169 MHz dipole-antenna | 92 |
| 8 | 434 MHz dipole-antenna | 93 |
| 9 | 868 MHz dipole-antenna | 94 |
| 10 | 868 MHz magnet foot antenna with 1.5 m antenna cable | 95 |
| 11 | 2.4 GHz dipole-antenna | 96 |
| 12 | Assembly diagram | 100 |
| 13 | Top and Bottom Layer | 101 |
| 14 | Reflow soldering profile | 103 |
| 15 | Module dimensions [mm] | 107 |
| 16 | Footprint and dimensions [mm] | 108 |
| 17 | Lot number structure | 109 |

List of Tables

| | | |
|----|--|-----|
| 1 | Ordering information | 10 |
| 2 | Recommended operating conditions | 11 |
| 3 | Absolute maximum ratings | 11 |
| 4 | Power consumption | 12 |
| 5 | Radio characteristics | 12 |
| 6 | Pin characteristics | 12 |
| 7 | Pinout | 16 |
| 8 | Communication in transparent mode | 22 |
| 9 | Telegram format in the command mode | 25 |
| 10 | Message overview - Part 1 | 26 |
| 11 | Message overview - Part 2 | 27 |
| 12 | Overview of non-volatile user-settings | 46 |
| 13 | Configuration flags | 68 |
| 14 | RF parameters | 80 |
| 15 | Channel overview, narrow band in 12.5 kHz channel spacing, predefined channel is bolded | 81 |
| 16 | Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E | 102 |
| 17 | Package classification reflow temperature, PB-free assembly, Note: refer to IPC/JEDEC J-STD-020E | 103 |
| 18 | Dimensions | 106 |
| 19 | Weight | 106 |
| 20 | Lot number details | 109 |



more than you expect



**Internet
of Things**



**Monitoring
& Control**



**Automated Meter
Reading**

Contact:

Würth Elektronik eiSos GmbH & Co. KG
Division Wireless Connectivity & Sensors

Rudi-Schillings-Str. 31
54296 Trier
Germany

Tel.: +49 651 99355-0
Fax.: +49 651 99355-69
www.we-online.com/wireless-connectivity

