# Revision history

<table>
<thead>
<tr>
<th>Manual version</th>
<th>SW version</th>
<th>Notes</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>3.0.0</td>
<td>• Initial version of this document</td>
<td>April 2019</td>
</tr>
<tr>
<td>1.2</td>
<td>3.0.0</td>
<td>• Updated file name to new AppNote name structure. Updated important notes, legal notice &amp; license terms chapters.</td>
<td>June 2019</td>
</tr>
<tr>
<td>1.3</td>
<td>3.1.0</td>
<td>• Updated supported modules</td>
<td>September 2019</td>
</tr>
</tbody>
</table>

* For SDK version history see chapter Software history
## Abbreviations and abstract

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Duty cycle</td>
<td>Active transmission time per hour expressed as percentage. 1% means, channel is occupied for 36 seconds per hour.</td>
</tr>
<tr>
<td>FSE</td>
<td>Field Sales Engineer</td>
<td>Support and sales contact person responsible for limited sales area</td>
</tr>
<tr>
<td>0xhh [HEX]</td>
<td>Hexadecimal</td>
<td>The prefix 0x indicates hexadecimal values. All other numbers are decimal values.</td>
</tr>
<tr>
<td>HIGH</td>
<td>High signal level</td>
<td></td>
</tr>
<tr>
<td>LOW</td>
<td>Low signal level</td>
<td></td>
</tr>
<tr>
<td>LPM</td>
<td>Low power mode</td>
<td>Operation mode with reduced energy consumption.</td>
</tr>
<tr>
<td>LRM</td>
<td>Long range mode</td>
<td>Tx mode increasing the RX sensitivity by using spreading and forward error correction</td>
</tr>
<tr>
<td>LSB</td>
<td>Least significant bit</td>
<td></td>
</tr>
<tr>
<td>MSB</td>
<td>Most significant bit</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>Payload</td>
<td>The real, non-redundant information in a frame/packet.</td>
</tr>
<tr>
<td>RF</td>
<td>Radio frequency</td>
<td>Describes everything relating to the wireless transmission.</td>
</tr>
<tr>
<td>SDK</td>
<td>Software development kit</td>
<td>Software code that implements the command interface of various Würth Elektronik eiSos products</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver Transmitter - a serial data transmission interface</td>
<td></td>
</tr>
<tr>
<td>VDD</td>
<td>Supply voltage</td>
<td></td>
</tr>
</tbody>
</table>
Contents

1 Introduction 4
  1.1 Motivation ........................................... 5

2 Wireless Connectivity SDK overview 7
  2.1 Content of the Wireless Connectivity SDK ............ 9

3 Host integration 10

4 Attaching Würth Elektronik eiSos radio modules / dongles to the Raspberry Pi 12
  4.1 Installation of the FTDI driver .......................... 12
  4.2 FAQ - Frequently asked questions ......................... 15
    4.2.1 The initialization function fails, what can I do? .... 15

5 Software history 16

6 Important notes 17
  6.1 General customer responsibility ........................ 17
  6.2 Customer responsibility related to specific, in particular safety-relevant applications ............................. 17
  6.3 Best care and attention ................................ 17
  6.4 Customer support for product specifications ............... 17
  6.5 Product improvements .................................. 18
  6.6 Product life cycle ..................................... 18
  6.7 Property rights ....................................... 18
  6.8 General terms and conditions ........................... 18

7 Legal notice 19
  7.1 Exclusion of liability ................................... 19
  7.2 Suitability in customer applications ...................... 19
  7.3 Trademarks ............................................. 19
  7.4 Usage restriction ....................................... 19

8 License terms 21
  8.1 Limited license ......................................... 21
  8.2 Usage and obligations ................................... 21
  8.3 Ownership ............................................. 22
  8.4 Firmware update(s) ..................................... 22
  8.5 Disclaimer of warranty .................................. 22
  8.6 Limitation of liability ................................... 23
  8.7 Applicable law and jurisdiction ........................ 23
  8.8 Severability clause ..................................... 23
  8.9 Miscellaneous .......................................... 23
1 Introduction

The Würth Elektronik eiSos wireless modules provide an easy to use radio interface to any embedded application. The module’s interface with the host processor of the embedded application via UART can be operated using a command interface.

The Wireless Connectivity SDK is a set of software tools that enable quick software integration of Würth Elektronik eiSos wireless modules into any of the most commonly used host processors. It consists of a collection of C-code developed on the Raspberry Pi 3 platform. It contains the drivers for radio modules as well as sample projects that use the UART and USB peripheral of Raspberry Pi 3 to communicate with the attached radio device.

Figure 1: Wireless connectivity SDK driver as part of the end product
1.1 Motivation

The aim of the Wireless Connectivity SDK is to minimize the effort required on customer side to enable his host MCU to communicate with Würth Elektronik eiSos radio modules. It contains the implementation of all available commands in pure C-code. In order to integrate any Würth Elektronik eiSos wireless module, the user has to simply port the corresponding C-code to his host processor. This significantly reduces the time needed for developing the software interface to the radio module.

Würth Elektronik eiSos products, like the 868MHz proprietary radio module Tarvos-III, use a so called command interface for configuration and operation tasks. This interface provides up to 30 commands that accomplish tasks like updating various device settings, transmit/receive data and putting the module into one of various low power modes.

There are Würth Elektronik eiSos wireless modules that can operate in transparent mode in addition to the standard command mode. When using the transparent mode, the device does not interpret the commands sent via UART. Please make sure that the connected radio device runs in command mode to use the Wireless Connectivity SDK.

The commands of such an interface can be divided into 3 categories:

1. Requests: The host requests the module to trigger any action, e.g. in case of the request `CMD_RESET_REQ` the host asks the module to perform a reset.

2. Confirmations: On each request the module answers with a confirmation message as a feedback on the requested operation status. In case of a `CMD_RESET_REQ`, the module answers with a `CMD_RESET_CNF` to tell the host whether the reset will be performed or not.

3. Indications and Responses: In case of special events, the module indicates the same spontaneously to the host. The `CMD_DATAEX_IND` indicates for example that data was received via radio.

The commands itself have the following format:

<table>
<thead>
<tr>
<th>Start byte</th>
<th>Command</th>
<th>Length</th>
<th>Payload</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x02</td>
<td>1 Byte</td>
<td>1 Byte</td>
<td>Length Bytes</td>
<td>1 Byte</td>
</tr>
</tbody>
</table>

Example: `CMD_DATA_REQ` of the Tarvos-III

The `CMD_DATA_REQ` has the command number 0x00. It serves a simple data transfer. The length field indicates the number of bytes to be transmitted via radio.

Format:

<table>
<thead>
<tr>
<th>Start byte</th>
<th>Command</th>
<th>Length</th>
<th>Payload</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x02</td>
<td>0x00</td>
<td>1 Byte</td>
<td>Length Byte</td>
<td>1 Byte</td>
</tr>
</tbody>
</table>

Sending "Hello World!"
Where we send 12 bytes (0x0C), which are "Hello World!" (0x48 0x65 0x6C 0x6F 0x20 0x57 0x6F 0x72 0x6C 0x64 0x21) and the resulting checksum is 0x0F.

To use the complete feature set of such a radio device, all available commands of the corresponding command interface have to be implemented on the custom host processor. This involves considerable effort for the user and this is exactly the reason why Würth Elektronik eiSos offers the Wireless Connectivity SDK.

The steps for porting are explained in more detail in chapter 3.
2 Wireless Connectivity SDK overview

The Wireless Connectivity SDK is developed on the Raspberry Pi 3 platform. It contains the radio module drivers as well as example projects demonstrating simple applications.

The radio modules supported by the different versions of the Wireless Connectivity SDK are shown in table 1.

The Evaluation board for a specific radio module can also be used with this SDK. As the Evaluation boards also include the FTDI UART to USB converter IC the "Plug" variant of the module driver package can also be used to interface the Evaluation boards.
<table>
<thead>
<tr>
<th>SDK version</th>
<th>Radio standard</th>
<th>Radio module &amp; usb dongle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>Proprietary 868 MHz</td>
<td>Tarvos-II, Tarvos-II Plug, Tarvos-III</td>
</tr>
<tr>
<td></td>
<td>Wireless M-BUS</td>
<td>Metis-II, Metis-II Plug, Metis-I, Metis-I Plug, Mimas-I, Mimas-I Plug</td>
</tr>
<tr>
<td>1.2.0</td>
<td>Proprietary 868 MHz</td>
<td>Tarvos-II, Tarvos-II Plug, Tarvos-III</td>
</tr>
<tr>
<td></td>
<td>Proprietary 2.4 GHz</td>
<td>Thalassa Plug</td>
</tr>
<tr>
<td></td>
<td>Wireless M-BUS</td>
<td>Metis-II, Metis-II Plug, Metis-I, Metis-I Plug, Mimas-I, Mimas-I Plug</td>
</tr>
<tr>
<td>1.6.3 &amp; 2.0.0</td>
<td>Bluetooth Low Energy</td>
<td>Proteus-I, Proteus-II</td>
</tr>
<tr>
<td></td>
<td>Proprietary 169 MHz</td>
<td>Titania</td>
</tr>
<tr>
<td></td>
<td>Proprietary 434 MHz</td>
<td>Thadeus</td>
</tr>
<tr>
<td></td>
<td>Proprietary 868 MHz</td>
<td>Tarvos-I, Tarvos-I Plug, Tarvos-II, Tarvos-II Plug, Tarvos-III, Tarvos-III Plug</td>
</tr>
<tr>
<td></td>
<td>Proprietary 915 MHz</td>
<td>Telesto-I, Telesto-II, Telesto-III, Telesto-III Plug</td>
</tr>
<tr>
<td></td>
<td>Proprietary 2.4 GHz</td>
<td>Triton, Thalassa, Thalassa Plug</td>
</tr>
<tr>
<td></td>
<td>Wireless M-BUS</td>
<td>Metis-II, Metis-II Plug, Metis-I, Metis-I Plug, Mimas-I, Mimas-I Plug</td>
</tr>
<tr>
<td>3.0.0</td>
<td>Bluetooth Low Energy</td>
<td>Proteus-I, Proteus-II</td>
</tr>
<tr>
<td></td>
<td>Proprietary 169 MHz</td>
<td>Titania</td>
</tr>
<tr>
<td></td>
<td>Proprietary 434 MHz</td>
<td>Thadeus</td>
</tr>
<tr>
<td></td>
<td>Proprietary 868 MHz</td>
<td>Tarvos-I, Tarvos-I Plug, Tarvos-II, Tarvos-II Plug, Tarvos-III, Tarvos-III Plug</td>
</tr>
<tr>
<td></td>
<td>Proprietary 915 MHz</td>
<td>Telesto-I, Telesto-II, Telesto-III, Telesto-III Plug</td>
</tr>
<tr>
<td></td>
<td>Proprietary 2.4 GHz</td>
<td>Triton, Thalassa, Thalassa Plug</td>
</tr>
<tr>
<td></td>
<td>Wi-Fi / WLAN</td>
<td>Calypso</td>
</tr>
<tr>
<td></td>
<td>Wireless M-BUS</td>
<td>Metis-II, Metis-II Plug, Metis-I, Metis-I Plug, Mimas-I, Mimas-I Plug</td>
</tr>
<tr>
<td>3.1.0</td>
<td>Bluetooth Low Energy</td>
<td>Proteus-I, Proteus-II</td>
</tr>
<tr>
<td></td>
<td>Proprietary 169 MHz</td>
<td>Titania</td>
</tr>
<tr>
<td></td>
<td>Proprietary 434 MHz</td>
<td>Thadeus</td>
</tr>
<tr>
<td></td>
<td>Proprietary 915 MHz</td>
<td>Telesto-I, Telesto-II, Telesto-III, Telesto-III Plug, Themisto-I</td>
</tr>
<tr>
<td></td>
<td>Proprietary 2.4 GHz</td>
<td>Triton, Thalassa, Thalassa Plug</td>
</tr>
<tr>
<td></td>
<td>Wi-Fi / WLAN</td>
<td>Calypso</td>
</tr>
<tr>
<td></td>
<td>Wireless M-BUS</td>
<td>Metis-II, Metis-II Plug, Metis-I, Metis-I Plug, Mimas-I, Mimas-I Plug</td>
</tr>
</tbody>
</table>

Table 1: Radio module support in the Wireless Connectivity SDK
2.1 Content of the Wireless Connectivity SDK

The Wireless Connectivity SDK is delivered as a zip-file.

Besides the various sample projects, there is a directory named drivers that contains the definition of the serial communication to the connected radio module. The subdirectory global contains all the shared functions as well as the definitions of the serial communication and GPIO interfaces of the underlying host.

The definition of the serial and USB interfaces of the Raspberry Pi is included.

In addition, each supported radio module has its own driver directory that contains the implementation of its command interface. Besides the definition of the commands, a thread is defined that checks for the confirmation and indication messages that are transmitted from the radio module to the host. Furthermore, functions that use specific radio module pins, like a pin reset or pin wake-up, are defined here.

```
/ 
  drivers ................................. Contains the code to be ported to custom hosts
    global
      _global.h.......................... Declares all functions to be defined on custom hosts
      _global.c............................ Implements shared functions
      _global_ftdi.c ................. UART and GPIO of the FTDI USB driver for RPi
      _global_serial.c.............. UART and GPIO of the serial interface of the RPi
    ...
    Triton............................... Command interface of the Triton module
      _Triton.h
      _Triton.c
    ...
    ThebeI............................... Command interface of the Thebe-I module
      _ThebeI.h
      _ThebeI.c
    ...
    Example_Triton........................ Demo project using Triton module
      _main.c
      _Example_Triton.cbp
    ...
    Example_ThebeI........................ Demo project using Thebe-I module
      _main.c
      _Example_ThebeI.cbp
    ...
```
3 Host integration

As described in chapter 2.1 the functions in the Wireless Connectivity SDK have been developed on the Raspberry Pi platform. To use all the features of the radio module, the module's drivers in the SDK have to be ported to the new custom platform.

In the following example, the steps involved in porting the drivers of the Bluetooth Low Energy 4.2 radio module, Proteus-I, to a custom platform is described.

- The directories `drivers/global` and `drivers/ProteusI` as well as the file `WE_common.h` have to be integrated to the custom project.

- In the function `InitDriver` of the file `ProteusI.c`, a thread is defined that listens to confirmation and indication messages that are transmitted from the attached radio module to the host processor.
  Case 1: Threads are supported, the `rxthread` has to be ported to the host’s thread system.
  Case 2: Threads are not supported, a state machine has to be created that periodically checks for incoming UART bytes.

```c
void *rx_thread()
{
    while(1)
    {
        /* wait for 1ms, then check if new RX data is available */
        delay (1);
        while (BytesAvailable())
        {
            /* interpret received byte */
            if (ReadByte(&readBuffer))
            {
                ...
            }
        }
    }
}
```

Code 1: Code snippet of the rxthread

- The file `global.h` declares the shared functions that deal with the serial interface as well as the usage of GPIOs for pin related functions.

```c
/* Switch pin to input/output high/low with/without pullup/pulldown
 * input:
 * − pin_number: number of pin
 * − inout: input or output
 * − pull: pullup, pulldown or no pull
 * − out: output level high or low
 * return: true, if success
 * false, otherwise
 */
extern bool SetPin( int pin_number, SetPin_InputOutput_t inout, SetPin_Pull_t pull,
                    SetPin_Out_t out);
```

```c
/* Open the serial interface
 * input:
 * − baudrate: baudrate of the interface
 * return: true, if success
 * false, otherwise
 */
```
Here the definition of these functions, depending on the custom host peripherals, has to be created by the user. The existing files `global_serial.c`, `global_ftdi.c` and `global.c` can be removed from the project as it contains the corresponding implementation for the Raspberry Pi.

After dealing with the `rxthread` and the definition of the functions declared in `global.h`, the driver is functional. The corresponding demo project can be considered as a basis for application development on the custom platform.
4 Attaching Würth Elektronik eiSos radio modules / dongles to the Raspberry Pi

For creating custom applications on the basis of the Raspberry Pi, connect the pins of the module to corresponding pins on the Raspberry Pi (power supply, ground, serial interface and other pins like reset). Please refer to figure 2 to get an overview of the pins of the Raspberry Pi used in the driver application examples.

![GPIO Connector](image)

Figure 2: Extended connector from Raspberry Pi

Alternatively, Würth Elektronik eiSos USB dongles such as the Tarvos-II Plug & Tarvos-III Plug can be directly connected to the USB-interface of the Raspberry Pi. If a USB dongle is to be used, the D2xx driver from FTDI has to be installed first (see section 4.1).

Depending on the chosen interface, the corresponding file (`global_serial.c` or `global_ftdi.c`), has to be included into the project.

Some products, such as Tarvos-II Plug, operate in transparent mode by default. In this case the device does not interpret the commands sent by the Wireless Connectivity SDK and thus communication fails. In this case first set the default operation mode to command mode, for example by using the Würth Elektronik eiSos software tool ACC.

4.1 Installation of the FTDI driver

To run a Würth Elektronik eiSos USB dongle such as the Tarvos-II Plug on the Raspberry Pi, the FTDI library is used and has to be installed.
To do so, first download the latest D2xx Linux driver for the ARMv6 hard-float architecture from https://www.ftdichip.com/Drivers/D2xx.htm.

To install the FTDI driver the following commands have to be run in the terminal. It is assumed that the driver is version 1.4.6 and saved to the ~/Downloads directory.

1. Go to the directory where the driver is saved
   ```
   cd ~/.Downloads
   ```

2. Unpack the gzip file
   ```
   tar -xvf libftd2xx-arm-v6-hf-1.4.6.tgz
   ```

3. Copy the needed header files to the system folder /usr/local/include
   ```
   sudo cp release/ftd2xx.h /usr/local/include
   sudo cp release/WinTypes.h /usr/local/include
   ```

4. Copy the libraries to the system folder /usr/local/lib and /usr/lib
   ```
   sudo cp release/build/lib* /usr/local/lib
   sudo cp release/build/lib* /usr/lib
   ```

5. As the D2xx driver is incompatible with the FTDI VCP driver in the Linux kernel, the kernel modules "ftdi_sio" and "usbserial" have to be unloaded. To do so, please run:
   ```
   sudo modprobe -r ftdi_sio
   sudo modprobe -r usbserial
   ```

6. Now the D2xx driver is ready for use and codeblocks has to be configured to use the library. Thus open the linker settings in Settings → Compiler → Linker Settings and add the library /usr/local/lib/libftd2xx.so to the Link libraries field.

7. In addition, the option -pthread has to be added to the Other linker options field (see figure 3). Close the linker settings again.
Figure 3: Adapted linker options

8. Now all supported USB dongles can be used.
4.2 FAQ - Frequently asked questions

4.2.1 The initialization function fails, what can I do?

The initialization function usually sets up the serial interface, performs a pin reset and waits for the module’s response. In case this fails, there are several possibilities:

- The module is not powered up. Please check the VCC and GND connection.
- The RESET line is not connected, thus no pin reset was applied.
- The UART RX and UART TX lines are not connected, thus the module response was not transmitted.
- The UART interface does not run well. Please check the UART settings and initialization.
- The connected module or USB dongle does not run in command mode and thus does not respond to a pin reset and/or command request. In this case, set the device to command mode, for example by using the software tool ACC.
- The rxthread function, waiting for module response, does not work correctly.
5 Software history

Version 1.0.0 "Engineering"
• Initial version of the SDK

Version 1.2.0 "Release"
• Added new products
• Updated driver structure to easily switch between serial and USB interface on Raspberry

Version 1.6.3/2.0.0 "Release"
• Added new products

Version 3.0.0 "Release"
• Added driver for Wi-Fi module Calypso and proprietary high power radio module Thebe-I
• Replaced old module names by new module names

Version 3.1.0 "Release"
• Added driver for proprietary high power radio module Thebe-II and Themisto-I
• Bugfix in reset function of Proteus-* drivers
• Fixed typos in function names and resulting bug in Calypso driver
6 Important notes

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

6.1 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.

6.2 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 sourcecode and firmware parts contained in or used with or for products in the wireless connectivity and sensor 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.

6.3 Best care and attention

Any product-specific data sheets, 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.

6.4 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.
6.5 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 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 6.1 and 6.2 remains unaffected. All wireless connectivity module driver software “wireless connectivity SDK” and its source codes as well as all PC software tools are not subject to the Product Change Notification information process.

6.6 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.

6.7 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.

6.8 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.
7 Legal notice

7.1 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.

7.2 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.

7.3 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.

7.4 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.
8 License terms

This License Terms will take effect upon the purchase and usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity products. You hereby agree that this license terms is applicable to the product and the incorporated software, firmware and source codes (collectively, "Software") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form.

The software 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 terms. You agree to comply with all provisions under this license terms.

8.1 Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the software and under the conditions that will be set forth in this license terms. You are free to use the provided Software only in connection with one of the products from Würth Elektronik eiSos to the extent described in this license terms. You are entitled to change or alter the source code for the sole purpose of creating an application embedding the Würth Elektronik eiSos wireless connectivity product. The transfer of the software to third parties is allowed to the sole extent that the software is used by such third parties in connection with our product or another hardware provided by Würth Elektronik eiSos under strict adherence of this license terms. Würth Elektronik eiSos will not assume any liability for the usage of the incorporated software and the source code. You are not entitled to transfer the source code in any form to third parties without prior written consent of Würth Elektronik eiSos.

You are not allowed to reproduce, translate, reverse engineer, decompile, disassemble or create derivative works of the incorporated Software and the source code in whole or in part. No more extensive rights to use and exploit the products are granted to you.

8.2 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 wireless connectivity 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.

8.3 Ownership

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

8.4 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.

8.5 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.
8.6 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, software and source code. 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.

8.7 Applicable law and jurisdiction

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

8.8 Severability clause

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

8.9 Miscellaneous

Würth Elektronik eiSos reserves the right at any time to change this terms at its own discretion. It is your responsibility to check at Würth Elektronik eiSos homepage for any updates. Your continued usage of the products will be deemed as the acceptance of the change. We recommend you to be updated about the status of new firmware and software, which is available on our website or in our data sheet and manual, and to implement new software in your device where appropriate.

By ordering a wireless connectivity product, you accept this license terms in all terms.
List of Figures

1  Wireless connectivity SDK driver as part of the end product . . . . . . . . . . . . 4
2  Extended connector from Raspberry Pi . . . . . . . . . . . . . . . . . . . . 12
3  Adapted linker options . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

List of Tables

1  Radio module support in the Wireless Connectivity SDK . . . . . . . . . . . . 8
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