

ANR030

NRFCONNECT - DEVELOPING A
CUSTOM FIRMWARE

VERSION 1.8

APRIL 14, 2026

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Contents

1 Abbreviations	2
2 Revision history	3
3 Introduction	4
4 Using nRF Connect SDK for firmware development	6
4.1 Choosing a Würth Elektronik eiSos board file	7
4.1.1 Known issues	8
4.1.2 Choosing a board with small memory	8
4.2 Overlay and configuration files	9
4.3 Connecting the Würth Elektronik eiSos hardware to the PC	10
5 Additional samples	11
5.1 Zephyr sensor node BLE	11
6 Certification of a Bluetooth custom firmware	12
7 References	13
8 Important notes	14

1 Abbreviations

Abbreviation	Name	Description
IDE	Integrated Development Environment	GUI and compiler to develop a firmware
RTOS	Real Time Operating System	Operating system for embedded devices
SDK	Software Development KIT	Source code and tools for firmware development
SWD	Serial Wire Debug	Protocol to debug a micro controller
VS Code	Visual Studio Code	PC tool developed by Microsoft

2 Revision history

Manual version	Notes	Date
1.0	<ul style="list-style-type: none"> Initial version 	March 2022
1.1	<ul style="list-style-type: none"> Removed description to manually add the Würth Elektronik eiSos board files to the Zephyr RTOS, as they are natively included In chapter <code>Certification of a Bluetooth custom firmware</code> link to the new application note about custom module certification 	October 2022
1.2	<ul style="list-style-type: none"> Updated <code>Important notes</code>, meta data and document style 	July 2023
1.3	<ul style="list-style-type: none"> Updated <code>Important notes</code> Added product <code>Ophelia-III/Proteus-III (nrf52840)</code> 	March 2024
1.4	<ul style="list-style-type: none"> Added product <code>Ophelia-IV/Proteus-IV (nrf54L15)</code> 	September 2025
1.5	<ul style="list-style-type: none"> Added chapter <code>Known issues</code> 	October 2025
1.6	<ul style="list-style-type: none"> Added support for nRFConnect SDK 3.1.1 	December 2025
1.7	<ul style="list-style-type: none"> Added chapter <code>Overlay and configuration files</code> 	March 2026
1.8	<ul style="list-style-type: none"> Added chapter <code>Additional samples</code> Added support for nRFConnect SDK 3.2.x 	April 2026

3 Introduction

This application note gives a short overview about the options to create a custom firmware for Würth Elektronik eiSos radio modules by using the hardware platform and the embedded nRF5 system on chip. It presents options on firmware development environments and accessories (like SDKs) for the use within the nRF5 ecosystem.

Supported Würth Elektronik eiSos radio modules:

Radio module	Article number	Radio chip
Proteus-I, Proteus-II	2608011x2xxxx	nRF52832
Proteus-III, Proteus-III-SPI, Ophelia-III	261101102xxxx	nRF52840
Proteus-e, Ophelia-I	261201102xxxx	nRF52805
Proteus-IV, Ophelia-IV	262101102xxxx	nRF54L15

Table 3: Integrated radio chips

The radio modules of the Proteus series contain a firmware, which allows to control the radio module via commands, that are sent by a connected micro processor via UART/SPI. In addition, they provide a Bluetooth® LE interface for data exchange with other Bluetooth® LE enabled devices.

The Ophelia variants contain no firmware and come with an empty flash memory.

In both cases, either using Ophelia or a Proteus radio modules, it may be necessary to develop an own firmware tailored to the applications needs.

For firmware development on base of Nordic semiconductor Bluetooth® LE chips, Nordic semiconductor provides two options:

1. The nRFConnect SDK [1] that uses the Zephyr RTOS [2] with the integrated Bluetooth® LE stack (SoftDevice Controller subsystem). There are regular releases (usually 4 major releases per year) of new versions of the nRF Connect SDK as its development and advancement is in the focus of Nordic Semiconductor and various other contributors.
2. The nRF5 SDK [3] that uses the so called "Soft devices" that integrate the Bluetooth® LE stack and the RTOS. There are various soft devices available, each providing a different function set and compatibility to different radio chips.

Soft device	Radio chips	Supported roles
S112	nRF52805, nRF52810, nRF52811, nRF52820, nRF52832	Peripheral
S132	nRF52832, nRF52810	Central and peripheral
S140	nRF52840	Central and peripheral
...

Table 4: Example: SoftDevice compatibility



The development of the nRF5 SDK was taken out of focus with major version 17. That means: Only bug fixes and maintenance releases are still done. This path will therefore not support Bluetooth® versions newer than 5.1.

As the nRF5 SDK won't be adapted to new Bluetooth® standards, Bluetooth® LE features or new Bluetooth® LE chip sets, we focus on the nRF Connect SDK in the remaining part of the document.

4 Using nRF Connect SDK for firmware development

As motivated above, in this chapter we consider the set-up of the development environment on base of the nRF Connect SDK.

The nRF Connect SDK uses CMake [4] and the resulting make files to compile the provided Bluetooth® stack (Zephyr RTOS) and the application source code. The IDE with best support and comfort is VS Code [5] equipped with the nRF Connect plug-in that provides all the features needed to develop a firmware on base of the Nordic Semiconductor radio chips.

To install and set-up nRF Connect using VS Code, Nordic Semiconductor provides a play list on youtube describing the work flow.



It's recommended to have administrator access and an internet connection on the PC where you install VS Code, nRF Connect and their respective dependencies, drivers and tools.

nRF Connect for VS Code tutorials:

https://www.youtube.com/playlist?list=PLx_tBuQ_KSqEt7NK-H7Lu78IT2OijwIMI

1. *Installation*

Use the nRF Connect tool to download all required sources, like Zephyr, from GitHub.

2. *Create an application*

Copy one of the various sample applications as a starting point for an own firmware development.

3. *Building an application & Exploring windows*

Bind your application to the hardware (for example Proteus-III EV-Board) and build the firmware. See also chapter 4.1 and 4.1.2 for details on selecting the Würth Elektronik eiSos radio module hardware platform.

4. *Flash an application to board*

Flash and run the firmware on the hardware. See also chapter 4.3 to connect the Würth Elektronik eiSos hardware to the PC.

5. *Debugging*

Debug the firmware on the hardware. See also chapter 4.3 to connect the Würth Elektronik eiSos hardware to the PC.

6. *Migrating from other IDEs to Microsoft VisualStudio Code ("VS Code")*

Less interesting for new developments: Firmware migration from previous supported IDEs to VS Code.

7. *Tips & Tricks*

Best practices when using nRF Connect SDK in VS Code.

4.1 Choosing a Würth Elektronik eiSos board file

In the step 3 "Building an application & Exploring windows" the underlying hardware is selected. By default, the nRF Connect SDK provides a bunch of supported boards. Würth Elektronik eiSos periodically adds the description of the available EV-Boards, like Proteus-III EV-Board, to the Zephyr RTOS. In nRF Connect SDK 2.0.0 or newer these boards are automatically available in the drop down menu of the board selection step.

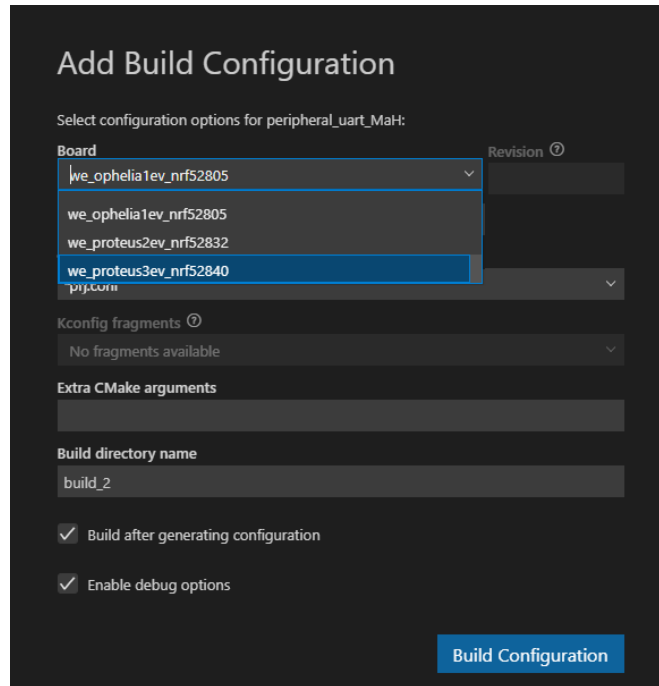


Figure 1: Select a Würth Elektronik eiSos board

These board files basically contain the radio module related pin-out information, such as UART pins, LEDs and buttons, as well as the use of the internal RC low frequency crystal, in case the radio module does not provide the external low frequency crystal on its PCB.

As several radio modules and the corresponding EV-Boards share the same hardware, the board name like "we_ophelia1ev_nrf52805" represents the hardware of several Würth Elektronik eiSos products. To choose the right board file for the purchased hardware, please refer to Table 5:

Board file	Product name	Article number
we_ophelia1ev/nrf52805	EV-Board of Proteus-e, Ophelia-I	261201XXXXXXX
we_proteus2ev/nrf52832	EV-Board of Proteus-I, Proteus-II	260801XXXXXXX
we_proteus3ev/nrf52840	EV-Board of Proteus-III, Proteus-III-SPI, Thyone-I, Ophelia-III	261101XXXXXXX
ophelia4ev/nrf54l15/cpuapp	EV-Board of Proteus-IV, Ophelia-IV	262101XXXXXXX, 262111XXXXXXX

Table 5: Choose the right board

4.1.1 Known issues

Due to the maintainance of the nRFConnect SDK by the Zephyr community and Nordic Semiconductor, there may be temporary limitations in the functions, that are solved in later versions of the nRFConnect SDK.

nRFConnect SDK version	Affected board	Issue
Only 3.1.x	ophelia4ev	The ophelia4ev/nrf54l15 does not provide the capabilities for MCUBOOT and partition manager feature, which are needed for the over the air update procedure. Solution: Please copy the extended board description from https://github.com/WurthElektronik/nRFConnectSDK/ and place it in the boards directory of the nRFConnect SDK installation (see corresponding the README-file).
3.1.x - 3.2.x ¹	ophelia4ev	The device tree of ophelia4ev/nrf54l15/cpuapp uses pins on P2 for UART30, however only P0 pins are supported. Solution: Please copy the extended board description from https://github.com/WurthElektronik/nRFConnectSDK/ and place it in the boards directory of the nRFConnect SDK installation (see corresponding the README-file).

Table 6: Known issues

4.1.2 Choosing a board with small memory

The Ophelia-I and Proteus-e use the Nordic Semiconductor chip nRF52805 [6] which provides only 192 KB flash and 24 KB RAM. In case the full feature set of the Zephyr Bluetooth® stack shall be used within this radio chip, the flash and RAM memory limitations are exceeded. Thus it is necessary to use the minimal build configuration² "prj_minimal.conf" (see figure 2) as a starter and add more features later depending on the need of the application and available memory resources.

¹This issue is fixed in Zephyr OS version 4.4.0.
²Project files are linked to the chosen example project. Most of them, but not all, provide the mentioned minimal project configuration file in addition to the standard project file.

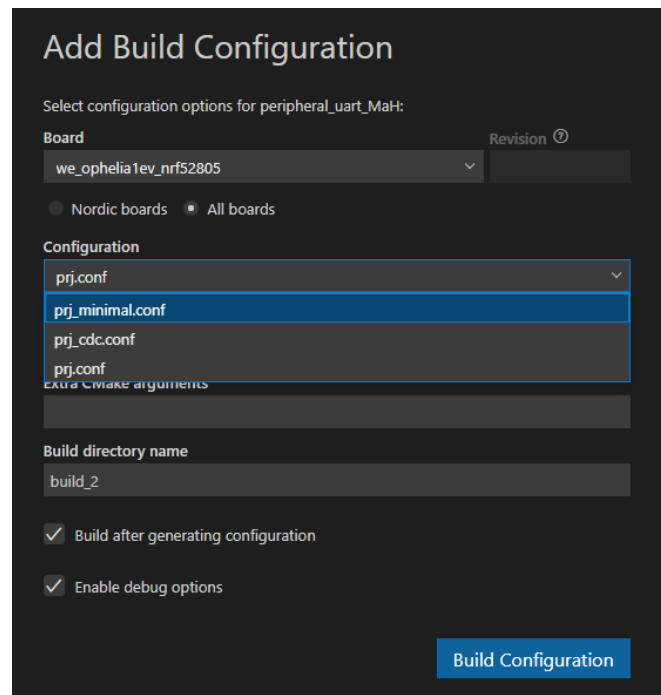


Figure 2: Choose minimal configuration

4.2 Overlay and configuration files

By default, the board definition is taken from the "...\\ncs\\x.x.x\\zephyr\\boards\\we" directory, which already contains a generic selection of GPIOs and peripherals (SPI, UART,...). Nevertheless, in most cases the chosen example project needs example-specific information of the underlying hardware. This is done via board-dependent overlay (.overlay) and config (.conf) files.

A good indication if these files are needed is, that they already exist for Nordic development kits of the same chipset. For Ophelia-IV, which is based on the nRF54L15, the nrf54l15dk_nrf54l15_cpuapp.overlay/.conf files exist in some of the examples. This leads to the conclusion that overlay- and config-files for Ophelia-IV must be created as well to add Ophelia-IV support to that example.

To do that a copy of the Nordic DK's files needs to be placed in the example directory under "boards", like:

```
"...\\zephyr\\samples\\bluetooth\\peripheral_dis\\boards"
```

The name of these files must match the board's name, like "ophelia4ev_nrf54l15_cpuapp.overlay" and "ophelia4ev_nrf54l15_cpuapp.conf" for the Ophelia-IV board.

If these files exist in the "boards" directory, they are automatically applied during the project compilation.

In most cases, the content of the Nordic DK's files, which is now also applied to our board, is sufficient to add example support to the underlying module hardware. Nevertheless, it is recommended to check the content for conflicting definitions.

Please note that the content in the overlay and config files strongly depend on the underlying version of nRFConnect SDK. In case the SDK is updated to a newer version, it must be checked if the content of the overlay and configuration file needs to be updated as well. Also here it is recommended to compare the content to the Nordic DK's files of the updated nRFConnect SDK.

4.3 Connecting the Würth Elektronik eiSos hardware to the PC

To flash and debug the firmware on the radio module, the corresponding EV-Board must be connected to the PC.

First of all, a USB-cable must be connected to the USB-plug on the EV-Board to power-source it. Next, the debug probe must be attached to the EV-Boards 2x10 pin debug connector. The considered radio chips provide a SWD (serial wire debug) interface that can be accessed to flash and debug the firmware. On modules with firmware (Proteus) you will need to perform the action "unprotect" before you will be able to access the module for your development. This action will perform a mass erase on the memory of the module and enable the access to SWD.



We recommend to use a SEGGER J-Link debug probe [7], like the *J-Link Plus* or *Ultra+*.

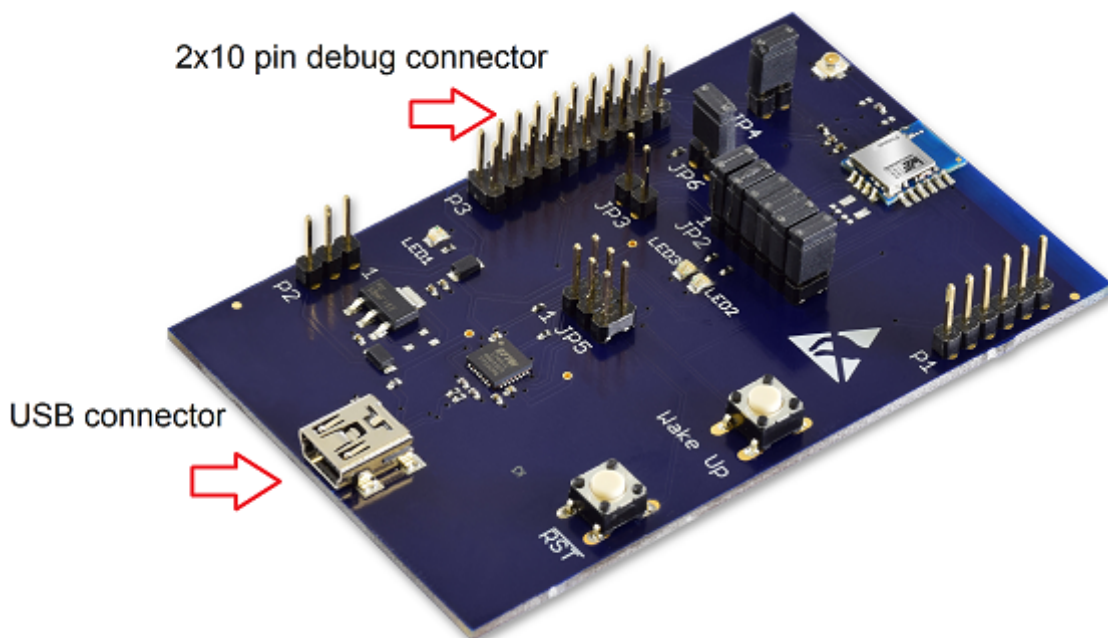


Figure 3: Example: EV-Board Proteus-II

For more detailed information about the EV-Boards, please refer to the corresponding EV-Board manual.

5 Additional samples

The nRFConnect SDK contains various samples which can be used as base for a custom firmware development.

In addition, Würth Elektronik eiSos provides samples in our own repository to showcase the use of the Würth Elektronik eiSos radio module and sensors:

<https://github.com/WurthElektronik/nRFConnectSDK/>

5.1 Zephyr sensor node BLE

The *Zephyr Sensor Node BLE sample* periodically reads sensor values via I²C and sends them via the Bluetooth[®] LE interface to the connected Bluetooth[®] peer (like a smartphone).



Figure 4: Setup

Depending on the connected sensors (*WSEN-TIDS*, *WSEN-PADS*, *WSEN-ITDS*, *WSEN-HIDS*), temperature, absolute pressure, acceleration and humidity values are transmitted. The sample is based on nRFConnect SDK 3.1.0, *Ophelia-III* and/or *Ophelia-IV* radio module.

The "WE-SensorBLE" smartphone app interpretes and displays the sensor values sent by the radio module. It is available for Android [8], iOS [9] and as source code [10].

6 Certification of a Bluetooth custom firmware

All Bluetooth® LE radio modules sold by Würth Elektronik eiSos have been tested to comply to various national regulations. In case the radio of the module containing the new custom firmware behaves like the original Würth Elektronik eiSos radio module, parts of the test results and certifications can be reused to certify the new custom radio module. For more details please refer to application note ANR031 [11].

In addition to that, the Bluetooth® listing must be considered. As the custom firmware has not been developed by Würth Elektronik eiSos, the module integrator can not use the "end device" Bluetooth® listing of the original product. To gain the Bluetooth® listing for the new custom product, please refer to the application note ANR027 [12]. This application note guides step by step through the listing process, in case a custom firmware shall be listed with the Bluetooth® SIG.

7 References

- [1] Nordic Semiconductor. nRF Connect SDK. <https://www.nordicsemi.com/Products/Development-software/nRF-Connect-SDK>.
- [2] Zephyr OS on GitHub. <https://github.com/zephyrproject-rtos/zephyr>.
- [3] Nordic Semiconductor. nRF5 SDK. <https://www.nordicsemi.com/Products/Development-software/nrf5-sdk>.
- [4] CMake. <https://cmake.org/overview/>.
- [5] Visual Studio Code. <https://code.visualstudio.com/>.
- [6] Nordic Semiconductor. Nordic nRF52805 resources. <https://www.nordicsemi.com/products/nrf52805>.
- [7] SEGGER J-Link debug probes. <https://www.segger.com/products/debug-probes/j-link/>.
- [8] Würth Elektronik. WE Sensor BLE app for Android. <https://play.google.com/store/apps/details?id=com.eisos.wesensorble>.
- [9] Würth Elektronik. WE Sensor BLE app for iOS. <https://apps.apple.com/de/app/we-sensorble/id1642507035>.
- [10] Würth Elektronik. Source code of WE Sensor BLE app (cross platform). <https://github.com/WurthElektronik/WE-SensorBLE>.
- [11] Würth Elektronik. Application note 31 - Certification of custom modules. <https://www.we-online.com/ANR031>.
- [12] Würth Elektronik. Application note 27 - Bluetooth listing guide. <http://www.we-online.com/ANR027>.

8 Important notes

The Application Note and its containing information ("Information") is based on Würth Elektronik eiSos GmbH & Co. KG and its subsidiaries and affiliates ("WE eiSos") knowledge and experience of typical requirements concerning these areas. It serves as general guidance and shall not be construed as a commitment for the suitability for customer applications by WE eiSos. While WE eiSos has used reasonable efforts to ensure the accuracy of the Information, WE eiSos does not guarantee that the Information is error-free, nor makes any other representation, warranty or guarantee that the Information is completely accurate or up-to-date. The Information is subject to change without notice. To the extent permitted by law, the Information shall not be reproduced or copied without WE eiSos' prior written permission. In any case, the Information, in full or in parts, may not be altered, falsified or distorted nor be used for any unauthorized purpose.

WE eiSos is not liable for application assistance of any kind. Customer may use WE eiSos' assistance and product recommendations for customer's applications and design. No oral or written Information given by WE eiSos or its distributors, agents or employees will operate to create any warranty or guarantee or vary any official documentation of the product e.g. data sheets and user manuals towards customer and customer shall not rely on any provided Information. THE INFORMATION IS PROVIDED "AS IS". CUSTOMER ACKNOWLEDGES THAT WE 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 A PURPOSE OR USAGE. WE 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 WE EISOS INFORMATION IS USED. INFORMATION PUBLISHED BY WE EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WE eiSos TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

The responsibility for the applicability and use of WE eiSos' components 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 and investigate, where appropriate, and decide whether the device with the specific characteristics described in the specification is valid and suitable for the respective customer application or not. The technical specifications are stated in the current data sheet and user manual of the component. Therefore the customers shall use the data sheets and user manuals and are cautioned to verify that they are current. The data sheets and user manuals can be downloaded at www.we-online.com. Customers shall strictly observe any product-specific notes, cautions and warnings. WE eiSos reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time without notice.

WE eiSos will in no case be liable for customer's use, or the results of the use, of the components or any accompanying written materials. IT IS CUSTOMER'S RESPONSIBILITY TO VERIFY THE RESULTS OF THE USE OF THIS INFORMATION IN IT'S OWN PARTICULAR ENGINEERING AND PRODUCT ENVIRONMENT AND CUSTOMER ASSUMES THE ENTIRE RISK OF DOING SO OR FAILING TO DO SO. IN NO CASE WILL WE EISOS BE LIABLE FOR CUSTOMER'S USE, OR THE RESULTS OF IT'S USE OF THE COMPONENTS OR ANY ACCOMPANYING WRITTEN MATERIAL IF CUSTOMER TRANSLATES, ALTERS, ARRANGES, TRANSFORMS, OR OTHERWISE MODIFIES THE INFORMATION IN ANY WAY, SHAPE OR FORM.

If customer determines that the components are valid and suitable for a particular design and wants to order the corresponding components, customer acknowledges to minimize the risk of loss and harm to individuals and bears the risk for failure leading to personal injury or death due to customer's usage of the components. The components have been designed and developed for usage in general electronic equipment only. The components are not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the components is reasonably expected to cause severe personal injury or death, unless WE eiSos and customer have executed an agreement specifically governing such use. Moreover WE eiSos components are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation, transportation signal, disaster prevention, medical, public information network etc. WE eiSos 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 component which is used in electrical circuits that require high safety and reliability functions or performance. CUSTOMER SHALL INDEMNIFY WE EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF THE COMPONENTS IN SUCH SAFETY-CRITICAL APPLICATIONS.

List of Figures

1	Select a Würth Elektronik eiSos board	7
2	Choose minimal configuration	9
3	Example: EV-Board Proteus-II	10
4	Setup	11

List of Tables

3	Integrated radio chips	4
4	Example: SoftDevice compatibility	4
5	Choose the right board	7
6	Known issues	8



Contact

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

Max-Eyth-Straße 1
74638 Waldenburg
Germany

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

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT