# ANR030

nRFConnect - Developing a custom firmware

Version 1.2

July 19, 2023

**WÜRTH ELEKTRONIK** MORE THAN YOU EXPECT

# Revision history

| Manual version | Notes | Date |
|---|---|---|
| 1.0 | • Initial version | March 2022 |
| 1.1 | • Removed description to manually add the Würth Elektronik eiSos board files to the Zephyr RTOS, as they are natively included<br><br>• In chapter `Certification of a Bluetooth custom firmware` link to the new application note about custom module certification | October 2022 |
| 1.2 | • Updated `Important notes`, meta data and document style | July 2023 |

## 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. |

# Contents

# 1 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 | 261101102xxxx | nRF52840 |
| Proteus-e, Ophelia-I | 261201102xxxx | nRF52805 |

Table 1: 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 the so called ¨SPP-like¨ profile on the Bluetooth® LE interface for data exchange with other Bluetooth® LE enabled devices. The Ophelia-I contains no firmware and comes with an empty flash memory.

In both cases, either using Ophelia-I or a Proteus radio module, 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 [3] that uses the Zephyr RTOS [9] 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 [4] 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 2: 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.

# 2  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 [1] 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 [6] 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-H7Lu78lT2OijwIMl*

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 evaluation board) and build the firmware. See also chapter `2.1` and `2.2` 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 `2.3` to connect the Würth Elektronik eiSos hardware to the PC.

5. *Debugging*
   Debug the firmware on the hardware. See also chapter `2.3` 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.

## 2.1 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 evaluation boards, like Proteus-III evaluation board, to the Zephyr RTOS. In nRF Connect SDK 2.0.0 or newer these boards are automatically available in the drop down menu (see figure 1) of the board selection step.
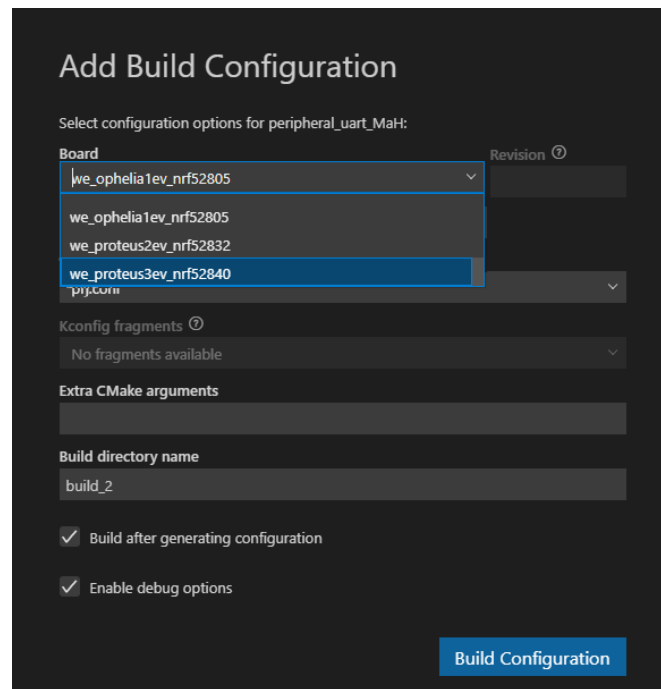


Figure 1: Select a Würth Elektronik eiSos board

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 evaluation 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 3:

| Board file | Product name | Article number |
|---|---|---|
| we_ophelia1ev_nrf52805 | Evaluation board of Proteus-e, Ophelia-I | 2612019XXXXXX |
| we_proteus2ev_nrf52832 | Evaluation board of Proteus-I, Proteus-II | 2608019XXXXXX |
| we_proteus3ev_nrf52840 | Evaluation board of Proteus-III, Proteus-III-SPI, Thyone-I | 2611019XXXXXX |

Table 3: Choose the right board

## 2.2 Choosing a board with small memory

The Ophelia-I and Proteus-e use the Nordic Semiconductor chip nRF52805 [2] 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 configuration[1] ¨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.
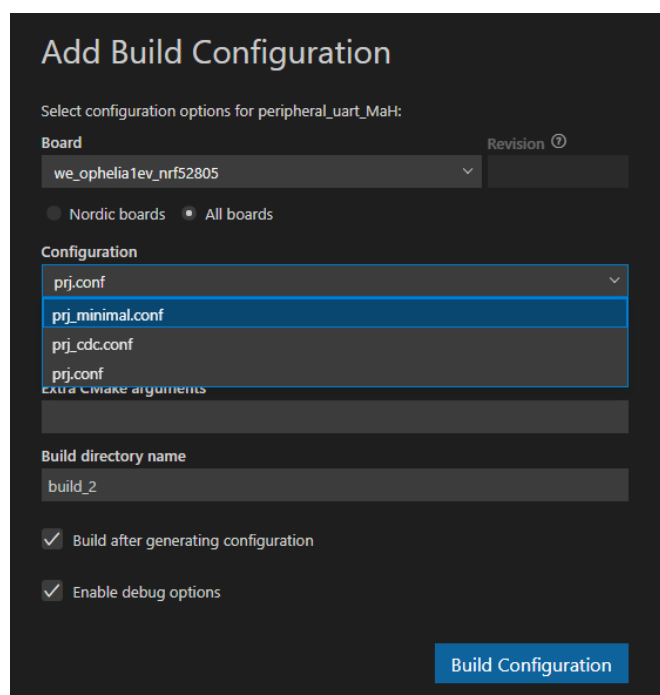


Figure 2: Choose minimal configuration

## 2.3 Connecting the Würth Elektronik eiSos hardware to the PC

To flash and debug the firmware on the radio module, the corresponding evaluation board must be connected to the PC.

First of all, a USB-cable must be connected to the USB-plug on the evaluation board to power-source it. Next, the debug probe must be attached to the evaluation 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.

---

[1]Project 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.

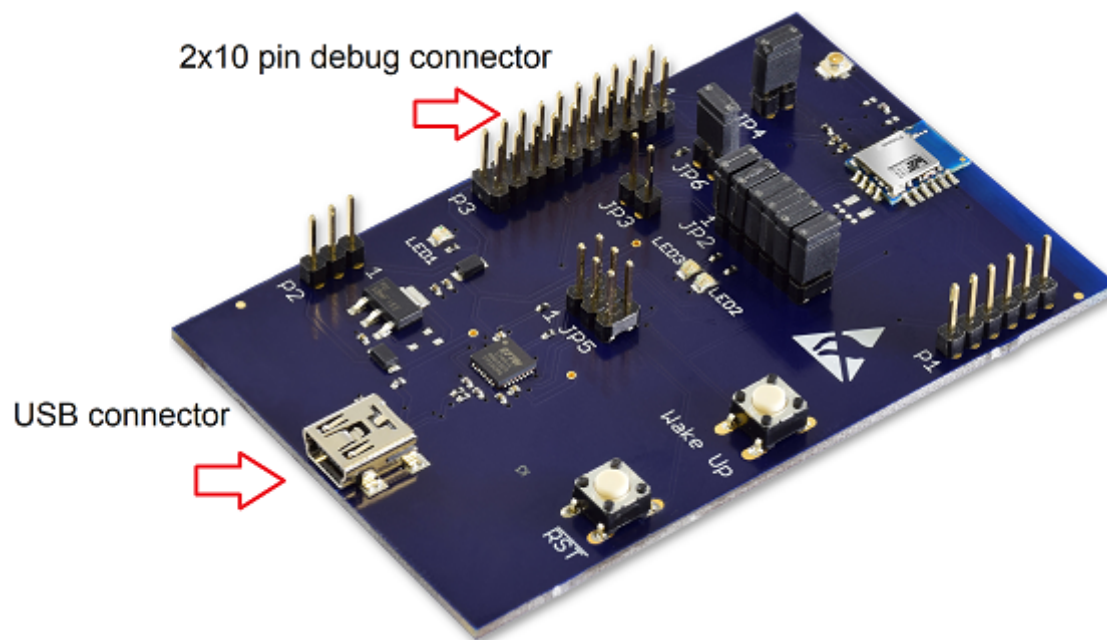> ⓘ We recommend to use a SEGGER J-Link debug probe [5], like the *J-Link Plus or Ultra+*.



Figure 3: Example: Evaluation board Proteus-II

For more detailed information about the evaluation boards, please refer to the corresponding evaluation board manual.

# 3 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 [8].

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 [7]. This application note guides step by step through the listing process, in case a custom firmware shall be listed with the Bluetooth® SIG.

# 4 References

[1] CMake. `https://cmake.org/overview/`.

[2] Nordic Semiconductor. Nordic nRF52805 resources. `https://www.nordicsemi.com/products/nrf52805`.

[3] Nordic Semiconductor. nRF Connect SDK. `https://www.nordicsemi.com/Products/Development-software/nRF-Connect-SDK`.

[4] Nordic Semiconductor. nRF5 SDK. `https://www.nordicsemi.com/Products/Development-software/nrf5-sdk`.

[5] SEGGER J-Link debug probes. `https://www.segger.com/products/debug-probes/j-link/`.

[6] Visual Studio Code. `https://code.visualstudio.com/`.

[7] Würth Elektronik. Application note 27 - Bluetooth listing guide. `http://www.we-online.com/ANR027`.

[8] Würth Elektronik. Application note "Certification of custom modules". `https://www.we-online.com/ANR031`.

[9] Zephyr OS on GitHub. `https://github.com/zephyrproject-rtos/zephyr`.

**WIRELESS CONNECTIVITY & SENSORS**

**ANR030 - nRFConnect - Developing a custom firmware**

WÜRTH
ELEKTRONIK
MORE THAN
YOU EXPECT

# 5 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

# List of Figures

# List of Tables

**WÜRTH ELEKTRONIK** MORE THAN YOU EXPECT