



## ANR035

ORTHOSIE-I - INTEGRATION IN  
RASPBERRY PI

VERSION 1.0

OCTOBER 1, 2024

**WÜRTH ELEKTRONIK** MORE THAN YOU EXPECT

## Revision history

Manual version	Notes	Date
1.0	<ul style="list-style-type: none"><li>Initial version</li></ul>	September 2024

## Abbreviations

Abbreviation	Name	Description
AP	Access Point	Connects WLAN devices to the network
HCI	Host Controller Interface	Layer of the Bluetooth® stack
OS	Operating System	
SDIO	Secure Digital Input Output	An interface for input or output devices
SPI	Serial Peripheral Interface	Allows the serial communication with the module
UART	Universal Asynchronous Receiver Transmitter	A serial data transmission interface
WLAN	Wireless Local Area Network	Most common form of WLAN is WiFi

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Setup</b>	<b>5</b>
2.1	Software setup . . . . .	6
2.1.1	Firmware setup . . . . .	6
2.1.2	Host software preparations . . . . .	7
2.1.3	Host software setup . . . . .	9
2.2	Hardware setup . . . . .	10
<b>3</b>	<b>References</b>	<b>13</b>
<b>4</b>	<b>Important notes</b>	<b>14</b>

# 1 Introduction

*esp-hosted* [1] by Espressif is a project, which provides a way to use ESP32-C3-based Orthosie-I [2] by Würth Elektronik eiSos as WiFi and Bluetooth® LE interface using Raspberry Pi as a host. The project supports multiple chips and communication interfaces. The provided functionality depends on the features of the used microchip. Refer to the datasheet [3] and errata [4] for supported functionalities.

This application note aims to guide through the official documentation and be used along side it. This document shows which parts are applicable and which steps have to be performed to use the supported Würth Elektronik eiSos radio products.

By means of the *esp-hosted* project the module can be used as 802.11 WLAN interface for WiFi connections and as HCI for Bluetooth® LE, allowing easy integration and development.

Some supported WiFi features are:

- 802.11b/g/n
- WiFi station
- Security modes: open, WPA, WPA2, WPA3

The *esp-hosted* project is compatible with the following Würth Elektronik eiSos radio products:

Radio module	Article number
Orthosie-I	2617011022000
Orthosie-I EV-Kit	2617029022001
Stephano-I	2617011025000
Stephano-I EV-Kit	2617029020001

Table 1: Integrated radio chips

While the Stephano-I hardware is supported, note that a special firmware by Espressif is required to use the *esp-hosted* project. Flashing Stephano-I with said firmware will replace its original firmware. The original firmware can not be restored. Therefore, using Orthosie-I is highly recommended.

## 2 Setup

This chapter will explain the steps required to set up the hardware and software on the radio module and Raspberry Pi.

The project comes in two variants to account for the variety of possible hosts. For Linux-based hosts, such as the Raspberry Pi running the official operating system, the esp-hosted Next Generation (ESP-Hosted-NG) shall be used. This document guides through the setup using an SPI connection for both WiFi and Bluetooth® LE.

Alternatively, WiFi over SPI and Bluetooth® LE over UART can be used. The setup of the UART interface is not described in this document. Refer to the official *esp-project documentation* [5] for instructions.

Note the use of SDIO is not supported with the ESP32-C3 chipset.

Interface	WLAN/WiFi support	Bluetooth® LE support
SPI	Yes	Yes
UART	No	Yes
SPI (WiFi) + UART (Bluetooth® LE)	Yes	Yes
SDIO	No	No

Table 2: Summary of supported transport interfaces

Check the original documentation [5] of the esp-hosted project for the latest version and in-depth instructions.



Read the *Espressif documentation* as well as issues in the connected git repository carefully. Würth Elektronik eiSos is neither responsible for nor can we influence the content of these documents. Any detected anomalies shall be directly reported to Espressif.



By installing Espressif firmware on the radio module, all radio certifications and declarations of Stephano-I become void and cannot be transferred to the resulting product without delta re-testing.

The setup was tested using a Raspberry-Pi 4 Model B, using the official Raspberry Pi OS with Linux kernel 6.1.21-v7. At the time of testing, the latest version of the master branch was used (commit 6728532) with firmware release 1.0.2.

## 2.1 Software setup

It is required to setup the software on both the module and the Raspberry Pi. The firmware for the module is pre-compiled and can be flashed as provided. The software for the Raspberry Pi host is available as source code and has to be compiled, as advised in the Espressif documentation.

### 2.1.1 Firmware setup

The firmware files are pre-compiled and provided, ready to flash, in the release section of the *esp-hosted project*. The *Espressif Flash Download Tool* can be used to flash the firmware to the module using the UART.



Flashing the firmware will remove any pre-installed firmware and calibration data from the module and can no longer be restored by the user.

After downloading and unpacking the released firmware, files for ESP32-C3 chips using SPI-only can be found in the folder: <esp-hosted-folder>/esp32c3/spi\_only

The following files have to be loaded into the flashing tool:

- bootloader.bin at address 0x00
- partition-table.bin at address 0x8000
- ota\_data\_initial.bin at address 0xd000
- network\_adapter.bin at address 0x10000

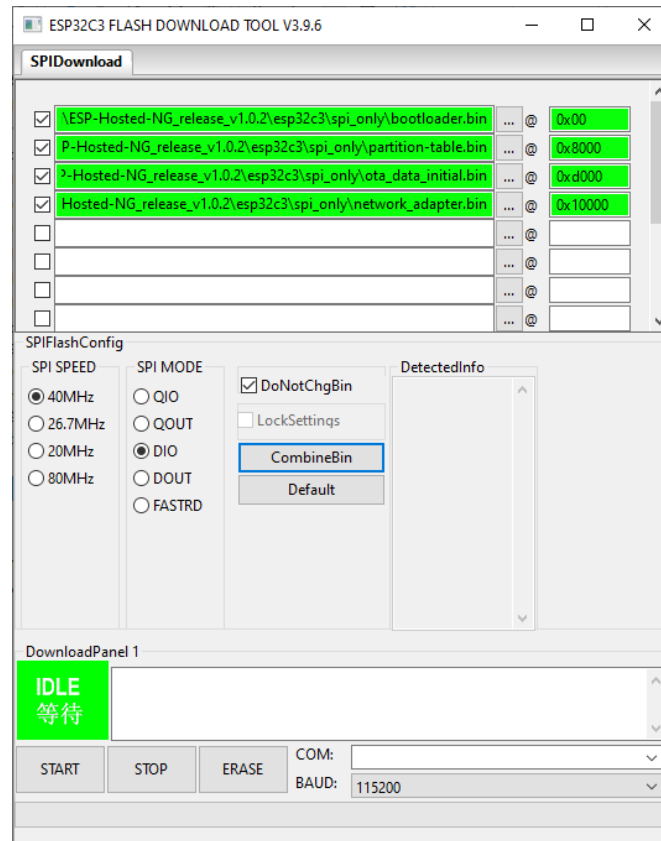


Figure 1: Load the necessary bin files into the flashing tool and flash the module.

### 2.1.2 Host software preparations

At the moment of writing, the following Raspberry Pi versions were supported as a host:

- Raspberry Pi 3 Model B
- Raspberry Pi 3 Model B+
- Raspberry Pi 4 Model B

Further Raspberry Pi models might be added in future by Espressif.



For other Linux platforms, the script provided by Espressif needs to be adapted to the host platform. This action is presented in the section *porting guide* within Espressif's documentation and will not be covered in this document.

To prepare the Raspberry Pi, install the required software and headers by performing the following commands in the command line:

Make sure the system is up to date:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```



Install the latest Linux header files and confirm correct installation:

```
$ sudo apt-get install raspberrypi-kernel-headers  
$ ls /lib/modules/$(uname -r)/build/
```

Install required libraries and tools:

```
$ sudo apt-get install git raspi-gpio  
$ sudo apt-get install bluetooth bluez bluez-tools rfkill bluez-firmware pi-bluetooth
```

To enable SPI and configure it for further use, adapt the /boot/config.txt file with the following content:

Enable SPI by adding or uncommenting the following lines:

```
dtparam=spi=on
```

Set SPI frequency by adding or uncommenting the following lines:

```
core_freq=250  
core_freq_min=250
```

Disable the integrated Bluetooth® interface of the Raspberry Pi by adding or uncommenting the following line:

```
dtoverlay=disable-bt
```

Save the file and reboot the Raspberry Pi to apply the changes.

### 2.1.3 Host software setup

Once the host system is up-to-date and the SPI interface is enabled, connect the module to the Raspberry Pi, as described in chapter 2.2. To setup, install and run the host software, perform the following steps:

Clone the esp-hosted project:

```
$ git clone https://github.com/espressif/esp-hosted.git
```

Connect the module to the Raspberry Pi as described in chapter 2.2. Then start the script which will compile and run the respective code.

```
$ cd esp_hosted/esp_hosted_ng/host/  
$ bash rpi_init.sh spi
```

When the script returns successfully, the virtual serial interface `/dev/wlanX`, usually `wlan1`, will be created, which can now be used as WLAN or WiFi interface.

For instructions to scan and connect to an AP, see the *Get Started chapter* of the official documentation. For instructions on how to use Bluetooth® LE, see chapter *Bluetooth/BLE Setup and Test Procedure*.

## 2.2 Hardware setup

The hardware in form of the radio modules Orthosie-I with the special firmware by Espressif will use the following pin-to-function mapping. Connect the module to the Raspberry Pi using the following configuration:

Function	Raspberry Pi Pin number	Modul Pin number	Orthosie-I/ESP32-C3 Pin name	Stephano-I Pin name
CS0	24	13	GPIO10	GPIO10
SCLK	23	8	GPIO6	U1RXD
MISO	21	17	GPIO2/BOOT_CTRL_0	BOOT_CTRL_0
MOSI	19	10	GPIO7	U1TXD
Handshake	15	3	GPIO3	GPIO3
Data Ready	13	6	GPIO4	U1RTS
Reset	31	4	CHIP_EN	RESET
VDD 3.3V	1	5	VDD	VDD
GND	25	B1-B6	GND	GND

Table 3: This table shows the pin names and how to connect all required pins from Raspberry Pi to the Orthosie-I.



It is recommended to add an external pull-up of 10 kOhm to the SPI's CS0 (Chip Select) to prevent unexpected behaviour due to a floating pin.



Due to the SPI clock speed, it is recommended to have the same length for all SPI lines (Clock, MOSI and MISO) with a maximum length of 10 cm.

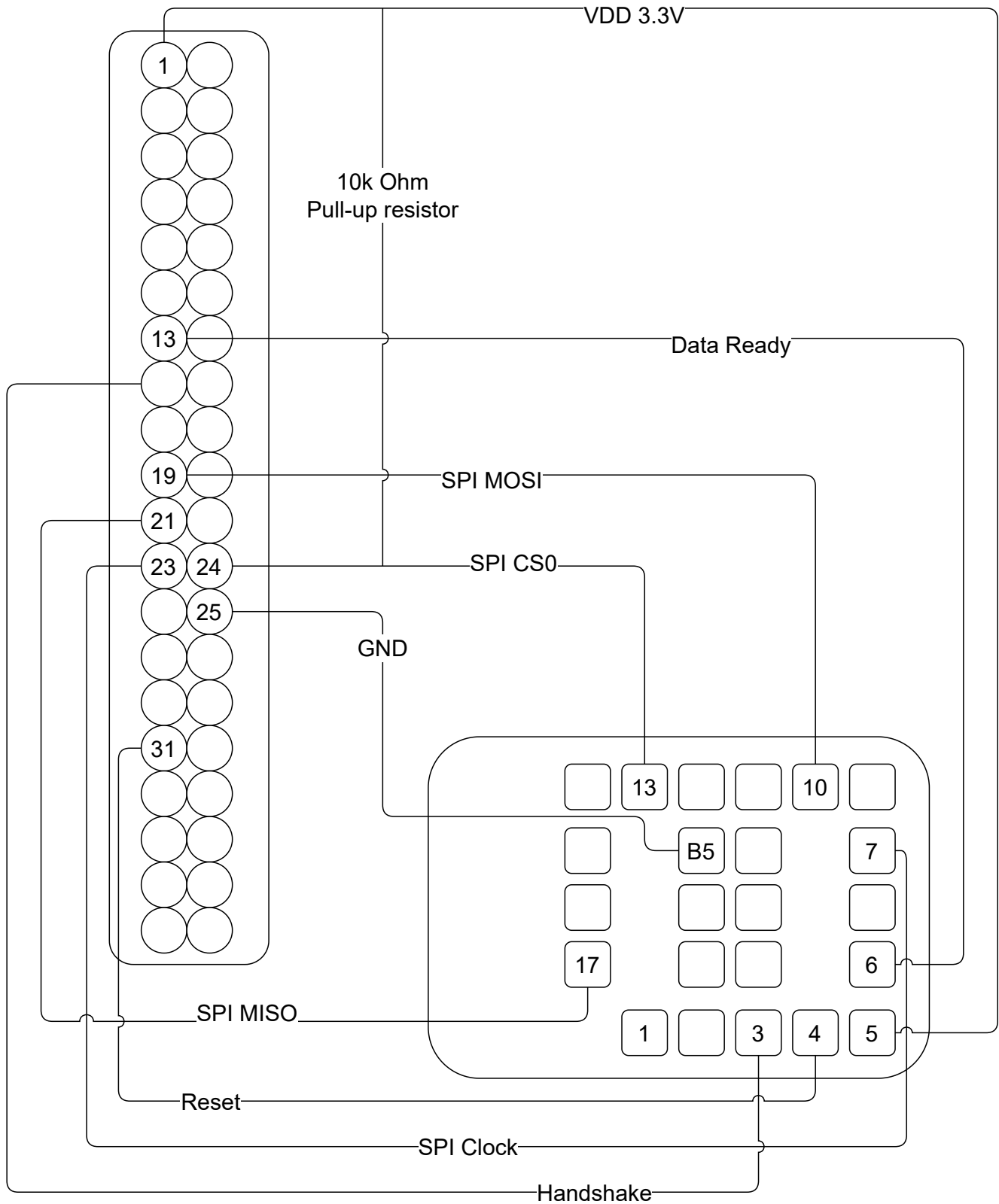


Figure 2: Block diagram of pin setup for SPI communication between Raspberry Pi and module.

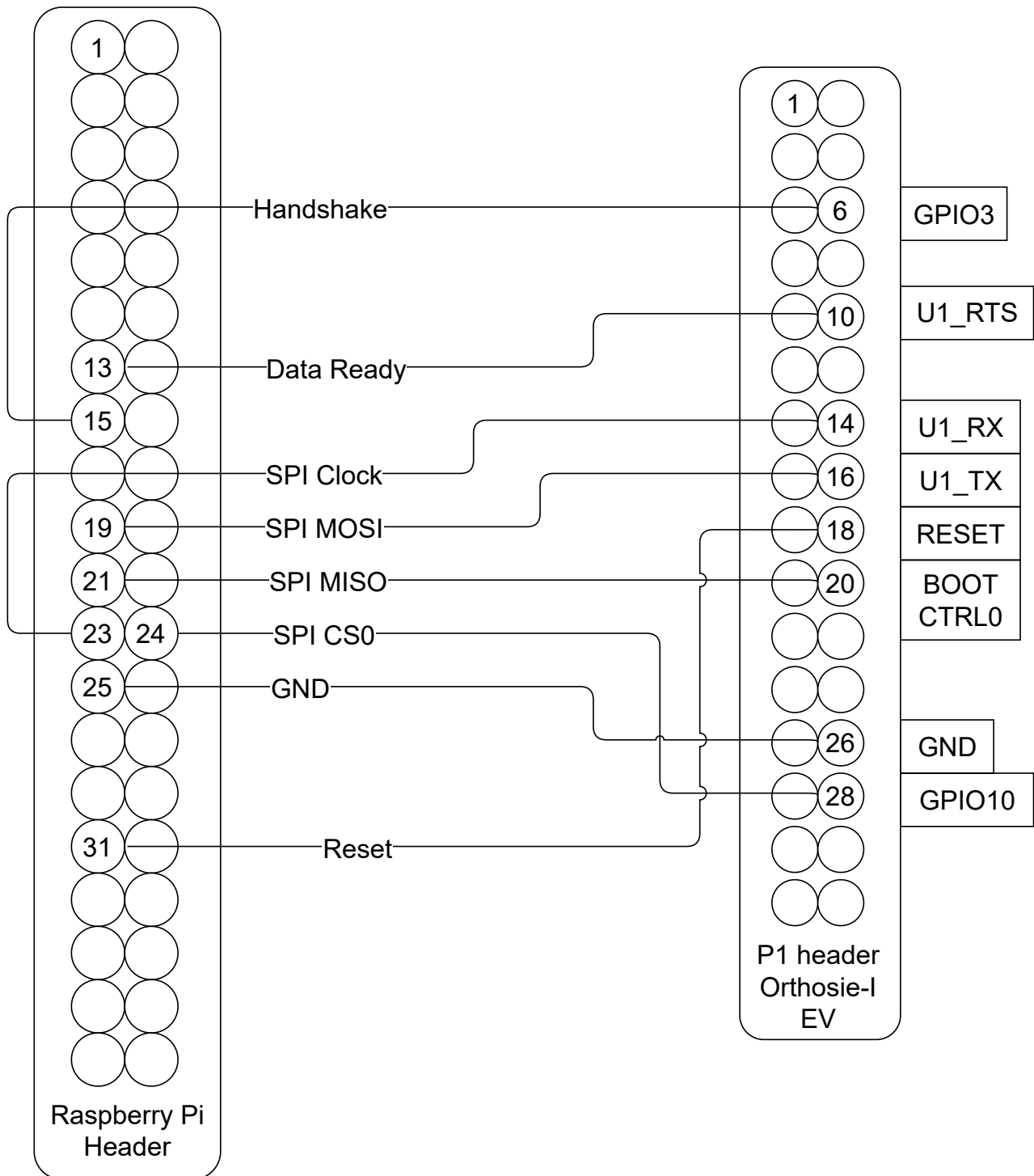


Figure 3: Block diagram of pin setup for SPI communication between Raspberry Pi and Orthosie-I EV-Board. The module pins are made available on the P1 header. No additional pull-up for SPI communication is required, as it is already integrated on the EV-Board. The EV-Board can be powered through the USB-port USB1 (not shown).

### 3 References

- [1] Espressif Systems. esp-hosted project on GitHub. <https://github.com/espressif/esp-hosted>.
- [2] Würth Elektronik. Orthosie-I user manual. <https://www.we-online.de/katalog/de/manual/2617011022000>.
- [3] Espressif Systems . Data sheet ESP32-C3 Wi-Fi And Bluetooth® SoC. [https://www.espressif.com/sites/default/files/documentation/esp32-c3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf).
- [4] Espressif Systems . Errata ESP32-C3 Wi-Fi And Bluetooth® SoC. [https://www.espressif.com/sites/default/files/documentation/esp32-c3\\_errata\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3_errata_en.pdf).
- [5] Espressif Systems. README of esp-hosted project on GitHub. [https://github.com/espressif/esp-hosted/blob/master/esp\\_hosted\\_ng/README.md](https://github.com/espressif/esp-hosted/blob/master/esp_hosted_ng/README.md).

## 4 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](http://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	Load the necessary bin files into the flashing tool and flash the module. . . . .	7
2	Block diagram of pin setup for SPI communication between Raspberry Pi and module. . . . .	11
3	Block diagram of pin setup for SPI communication between Raspberry Pi and Orthosie-I EV-Board. The module pins are made available on the P1 header. No additional pull-up for SPI communication is required, as it is already integrated on the EV-Board. The EV-Board can be powered through the USB-port USB1 (not shown). . . . .	12

## List of Tables

1	Integrated radio chips . . . . .	4
2	Summary of supported transport interfaces . . . . .	5
3	This table shows the pin names and how to connect all required pins from Raspberry Pi to the Orthosie-I. . . . .	10





**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](http://www.we-online.com/wireless-connectivity)

**WÜRTH ELEKTRONIK** MORE THAN YOU EXPECT