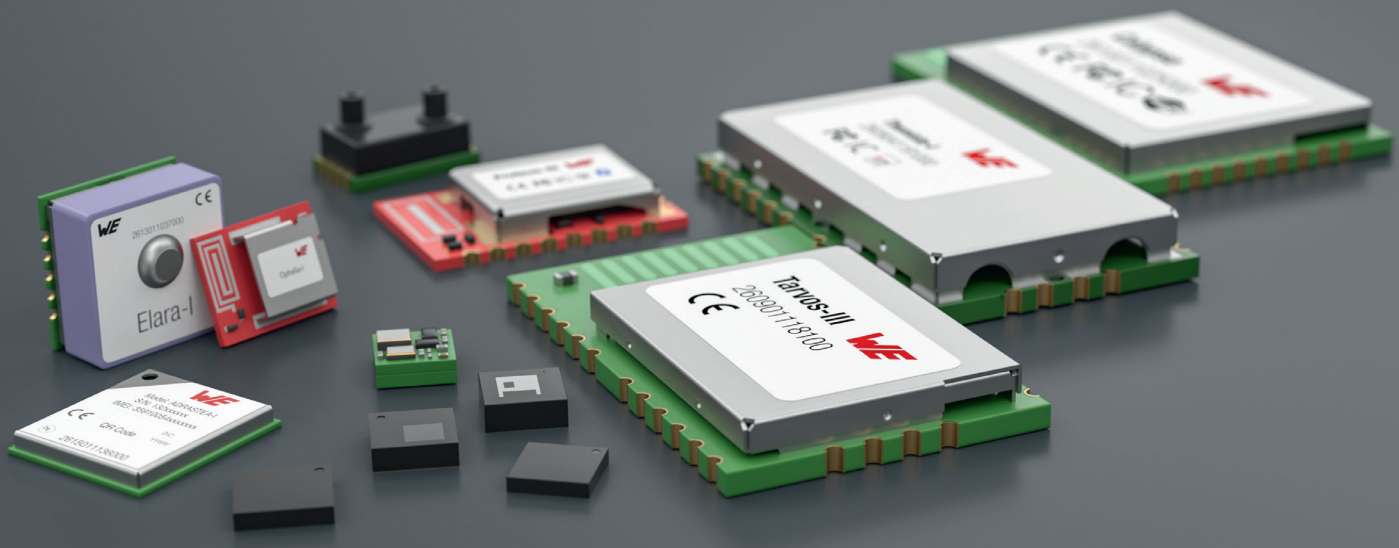

ANR006

Proteus High throughput mode

Version 1.5

July 19, 2023

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Revision history

Manual version	Notes	Date
1.0	• Initial version	November 2018
1.1	• Updated file name to new AppNote name structure. Updated important notes, legal notice & license terms chapters.	June 2019
1.2	• Updated address of Division Wireless Connectivity & Sensors location	October 2019
1.3	• Added new test results of Proteus-II and Proteus-III	February 2020
1.4	• Added new test results of the Proteus-III-SPI	February 2021
1.5	• Updated `Important notes`, meta data and document style	July 2023

Abbreviations

Abbreviation	Name	Description
Payload		The intended message in a frame / package.
RF	Radio frequency	Describes wireless transmission.
UART	Universal Asynchronous Receiver Transmitter	Allows the serial communication with the module.
[HEX] 0xhh	Hexadecimal	All numbers beginning with 0x are hexadecimal numbers. All other numbers are decimal, unless stated otherwise.

Contents

1 Introduction **4**
1.1 Compatibility and risks 5

2 Usage **6**
2.1 Maximum packet size 6
2.2 Bluetooth LE packet format 6
2.2.1 Example: Transmitting fragmented data 7

3 Throughput test **8**
3.1 Test conditions 8
3.2 Test procedure 8
3.3 Test results 9
3.3.1 Test setup 1: Via USB interface and PC 9
3.3.2 Test setup 2: Via micro controller 10

4 Important notes **12**

1 Introduction

The Proteus series is a product series of Bluetooth® modules based on the nRF52 Nordic Semiconductors SoC which presents various Bluetooth® LE and low power features.
Bluetooth® LE enabled devices allow to transmit/receive data in a short interval after the connection event. During this period, by default one data packet of up to 251 Bytes is transmitted from master to slave and one data packet from slave to master. After the transmission, the device goes to sleep until the next connection event occurs (see figure 1).

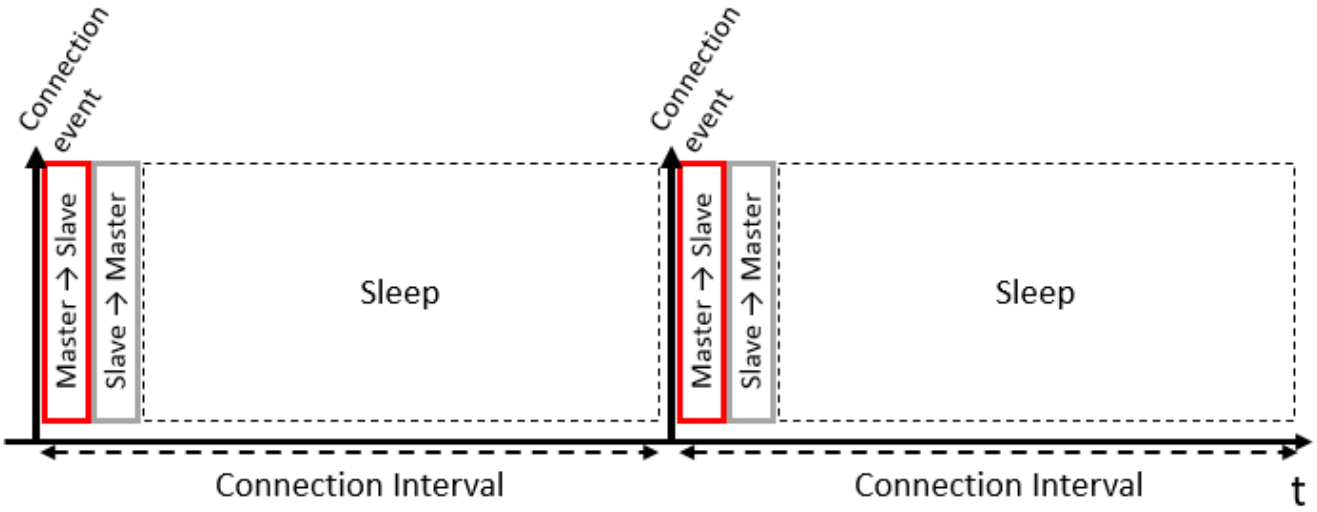

Figure 1: Data transmission with 1 packet per connection interval

Due to this procedure, the device is suited for low power applications, but it's default radio throughput is limited[1] to:

$$\frac{\text{Data per packet}}{\text{Connection interval}}[Byte/s] \tag{1}$$

To overcome this limitation, for Proteus-II and Proteus-III a new mode has been introduced. This so called "high throughput mode" enables to transmit/receive up to 4 Bluetooth® LE packets per connection interval (see figure 2). With this the throughput of a Bluetooth® LE connection can be increased, at the expense of a higher energy consumption.

[1] Please note that there are further conditions that slow down the throughput, like the UART speed and the µC processor speed.

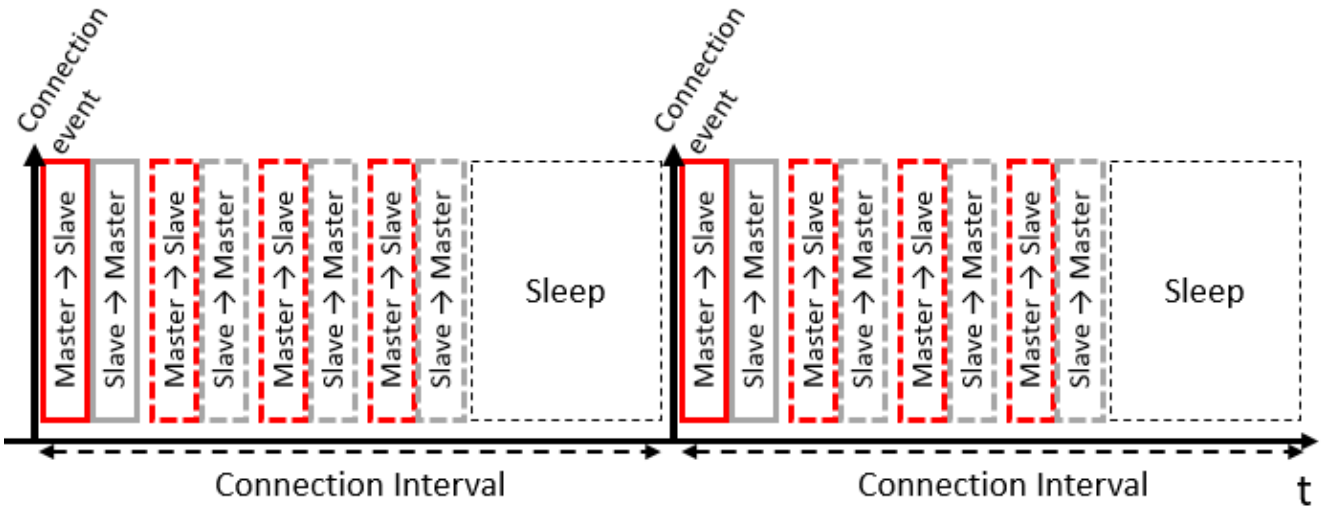

Figure 2: Improved data transmission with up to 4 packets per connection interval

1.1 Compatibility and risks

By default Bluetooth® LE enabled devices support the transmission of one data packet in a short period after the connection event. With upcoming new Bluetooth® LE devices an increasing number of data packets per connection interval is supported. Modern iOS devices support up to 4 packets, Android devices support up to 6 packets per connection interval:

Device	Number of packets per connection interval
Samsung Galaxy S8	4
Nexus 4, Nexus 6P	6
OnePlus 5	6
iPhone 6, 7, Xs	4

Table 1: Examples of Bluetooth® LE enabled devices supporting this feature

As only one packet per connection interval must be supported by a Bluetooth® LE connection, there is always a risk that the connection partner does not support this feature.

2 Usage

To enable the high throughput mode the corresponding bit in the `CFG_Flags` has to be set by using the `CMD_SET_REQ` command.

2.1 Maximum packet size

During the connection setup the module outputs a `CMD_CHANNELOPEN_RSP` message on the UART. This message contains the maximum payload of a standard packet (Φ_{ST}). Keeping this value in mind, the maximum packet to be transmitted via the `CMD_DATA_REQ` in high throughput mode is calculated as:

$$\Phi_{HTM} = (\Phi_{ST} - 2) \times 4[Bytes] \tag{2}$$

In case of the largest maximum transmission unit (MTU), that is supported by the Proteus modules, the size of the standard packet Φ_{ST} is 243 (0xF3) Bytes. Therefore, the largest packet size Φ_{HTM} in high throughput mode calculates as 964 Bytes.

2.2 Bluetooth LE packet format

The standard Bluetooth® LE packet of a Proteus module has the following format:

Bluetooth® LE Payload	
Header	Payload data
0x01	Φ_{ST} Bytes

Table 2: RF-packet format to transmit data

Using that frame format one packet is sent per connection interval. To increase throughput, the new frame format

Bluetooth® LE Payload			
Header	Sequence number	Fragment ID	Payload data
0x04	1 Byte	1 Byte	$\Phi_{ST} - 2$ Bytes

Table 3: RF-packet format for fragmented data of the high throughput mode

has been introduced in high throughput mode. Here up to 4 packets can be sent per connection interval. The sequence number and fragment ID are used to fragment and defragment the payload packet of Φ_{HTM} bytes into 4 packets, each of $(\Phi_{ST} - 2)$ bytes size.

Sequence number is a random number, that has to be the same for all fragments of a fragmented packet, but should differ for each packet.

Fragment ID is of the structure **0xAB** and will define the order of the fragments to be defragmented/combined again

A the first 4 bits define the current fragment number (starting with 1)

B the last 4 bits define the overall number of fragments

2.2.1 Example: Transmitting fragmented data

Let us assume that we have a fragment payload size of 5 Bytes, the random number has been chosen as 0x00 and we suppose to send data consisting of 12 Bytes 0xE1 - 0xEC.
Then 3 fragments of the following structure have to be transmitted:

Header	Sequence number	Fragment ID	Payload
0x04	0x00	0x13	0xE1 0xE2 0xE3 0xE4 0xE5

Table 4: Fragment 1

Header	Sequence number	Fragment ID	Payload
0x04	0x00	0x23	0xE6 0xE7 0xE8 0xE9 0xEA

Table 5: Fragment 2

Header	Sequence number	Fragment ID	Payload
0x04	0x00	0x33	0xEB 0xEC

Table 6: Fragment 3

When receiving fragmented data packets with header **0x04**, the fragments are combined and output by a `CMD_DATA_IND` message on the UART only if:

- all fragments have been received
- all fragments have the same overall number of fragments number (**B**, i.e. the last 4 bits of the **Fragment ID**)
- all fragments contain the same sequence number

In a case where at least one of this conditions is not satisfied the data will be discarded without further notice.

3 Throughput test

This chapter describes the throughput tests that have been performed. The following devices have been tested:

- Proteus-II in FW version 1.1.0
- Proteus-III in FW version 1.1.0
- Proteus-III SPI in FW version 1.2.0

3.1 Test conditions

To get the best throughput performance we choose the following settings of the Proteus module:

- The Proteus module operates in command mode.
- The UART/SPI must be set to maximum speed:
  - Proteus-II: The UART runs with baudrate 921600 Baud, 8n1, and uses the flow control pins *RTS* and *CTS*.
  - Proteus-III: The UART runs with baudrate 1000000 Baud, 8n1, and uses the flow control pins *RTS* and *CTS*.
  - Proteus-III SPI: The SPI runs with a datarate of 8 Mbit.
- The connection interval has been set to a fast value depending on the devices used (see chapter 3.3).
- Set the module to high throughput mode by setting the corresponding bit in the `CFG_Flags`.
- The 2 Mbit mode PHY is used for fastest radio data transmission.

If data is transmitted from one Proteus module to another, both modules must be configured in the same manner.

3.2 Test procedure

The test procedure is as follows:

1. Setup a Bluetooth connection between the two Bluetooth® LE enabled devices. We call them here device A to device B.
2. Check if large data packets are supported, by reading the maximum transmission unit (MTU) from the `CMD_CHANNELOPEN_RSP` message during connection setup. The maximum supported data packets Φ_{ST} are:
   a) Proteus to Proteus: Φ_{ST} = 243 (0xF3) Bytes

b) Proteus to Android device: Φ_{ST} = 243 (0xF3) Bytes

c) Proteus to iOS device: Φ_{ST} = 181 (0xB5) Bytes

3. Switch the PHY to 2 Mbit using the `CMD_PHYUPDATE_REQ` command.

4. Iterate over:

   a) Transmit a data packet of maximum size Φ_{HTM} Bytes using a `CMD_DATA_REQ` command. Φ_{HTM} calculates as:

      i. Proteus to Proteus: Φ_{HTM} = (243 - 3 + 1) × 4 = 964 Bytes

      ii. Proteus to Android device: Φ_{HTM} = (243 - 3 + 1) × 4 = 964 Bytes

      iii. Proteus to iOS device: Φ_{HTM} = (181 - 3 + 1) × 4 = 716 Bytes

   b) Wait for the transmission success message (`CMD_TXCOMPLETE_IND`) that is returned on the UART as result of the previous transmission request.

3.3 Test results

The throughput tests have been performed in two different test setups.

3.3.1 Test setup 1: Via USB interface and PC

The Proteus modules are connected via FTDI-chip and USB cable to a PC. A PC tool triggers the repeated data transmission. In this case the USB latencies slow down the data transmission.

Device A	Device B	Connection interval [ms]	Throughput [kBytes/s]
Proteus-II	Proteus-II	7.5 - 7.5	24.7
Proteus-II	Proteus-II	7.5 - 30	16
Proteus-II	Proteus-II	20 - 75	6.3
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-II	7.5 - 7.5	18.1
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-II	7.5 - 30	11.28
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-II	20 - 75	9.7
iPhone 7 (MN922 ZD/A, iOS 11.4)	Proteus-II	7.5 - 30	8.125
iPhone 7 (MN922 ZD/A, iOS 11.4)	Proteus-II	20 - 75	8.125

Table 7: Test results Proteus-II via USB

Device A	Device B	Connection interval [ms]	Throughput [kBytes/s]
Proteus-III	Proteus-III	7.5 - 7.5	21.3
Proteus-III	Proteus-III	7.5 - 30	15.9
Proteus-III	Proteus-III	20 - 75	6.3
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-III	7.5 - 7.5	18.9
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-III	7.5 - 30	8
Samsung Galaxy S8 (SM-G950F, Android 8.0)	Proteus-III	20 - 75	6.16
iPhone XS (iOS 12.2)	Proteus-III	7.5 - 30	8.7
iPhone XS (iOS 12.2)	Proteus-III	20 - 75	8.2

Table 8: Test results Proteus-III via USB

3.3.2 Test setup 2: Via micro controller

The Proteus modules' UARTs/SPIs are directly connected to a micro controller[2], that triggers the repeated data transmission.

Device A	Device B	Connection interval [ms]	Throughput [kBytes/s]
Proteus-II	Proteus-II	7.5 - 7.5	32.17
Proteus-II	Proteus-II	7.5 - 30	16.07
Proteus-II	Proteus-II	20 - 75	6.78
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-II	7.5 - 7.5	28.60
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-II	7.5 - 30	16.07
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-II	20 - 75	9.93
iPhone 7 (MN922 ZD/A, iOS 11.4)	Proteus-II	7.5 - 30	12.13
iPhone 7 (MN922 ZD/A, iOS 11.4)	Proteus-II	20 - 75	11.21

Table 9: Test results Proteus-II via micro controller

[2] In this test a STM32 on a NUCLEOL476RG has been used.

Device A	Device B	Connection interval [ms]	Throughput [kBytes/s]
Proteus-III	Proteus-III	7.5 - 7.5	42.9
Proteus-III	Proteus-III	7.5 - 30	16.43
Proteus-III	Proteus-III	20 - 75	6.75
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III	7.5 - 7.5	28.69
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III	7.5 - 30	16.07
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III	20 - 75	6.30
iPhone XS (iOS 12.2)	Proteus-III	8 - 30	11.95
iPhone XS (iOS 12.2)	Proteus-III	20 - 75	11.95

Table 10: Test results Proteus-III via micro controller

Device A	Device B	Connection interval [ms]	Throughput [kBytes/s]
Proteus-III-SPI	Proteus-III-SPI	7.5 - 7.5	64.3
Proteus-III-SPI	Proteus-III-SPI	7.5 - 30	16.07
Proteus-III-SPI	Proteus-III-SPI	20 - 75	12.85
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III-SPI	7.5 - 7.5	42.88
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III-SPI	7.5 - 30	16.07
Samsung Galaxy S8 (Model SM-G950F, Android 8.0)	Proteus-III-SPI	20 - 75	6.42

Table 11: Test results Proteus-III SPI via micro controller

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*. 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 customers 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. COSTUMER SHALL INDEMNIFY WE EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF THE COMPONENTS IN SUCH SAFETY-CRITICAL APPLICATIONS.

List of Figures

1 Data transmission with 1 packet per connection interval 4
2 Improved data transmission with up to 4 packets per connection interval 5

List of Tables

1 Examples of Bluetooth® LE enabled devices supporting this feature 5
2 RF-packet format to transmit data 6
3 RF-packet format for fragmented data of the high throughput mode 6
4 Fragment 1 7
5 Fragment 2 7
6 Fragment 3 7
7 Test results Proteus-II via USB 9
8 Test results Proteus-III via USB 10
9 Test results Proteus-II via micro controller 10
10 Test results Proteus-III via micro controller 11
11 Test results Proteus-III SPI via micro controller 11

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