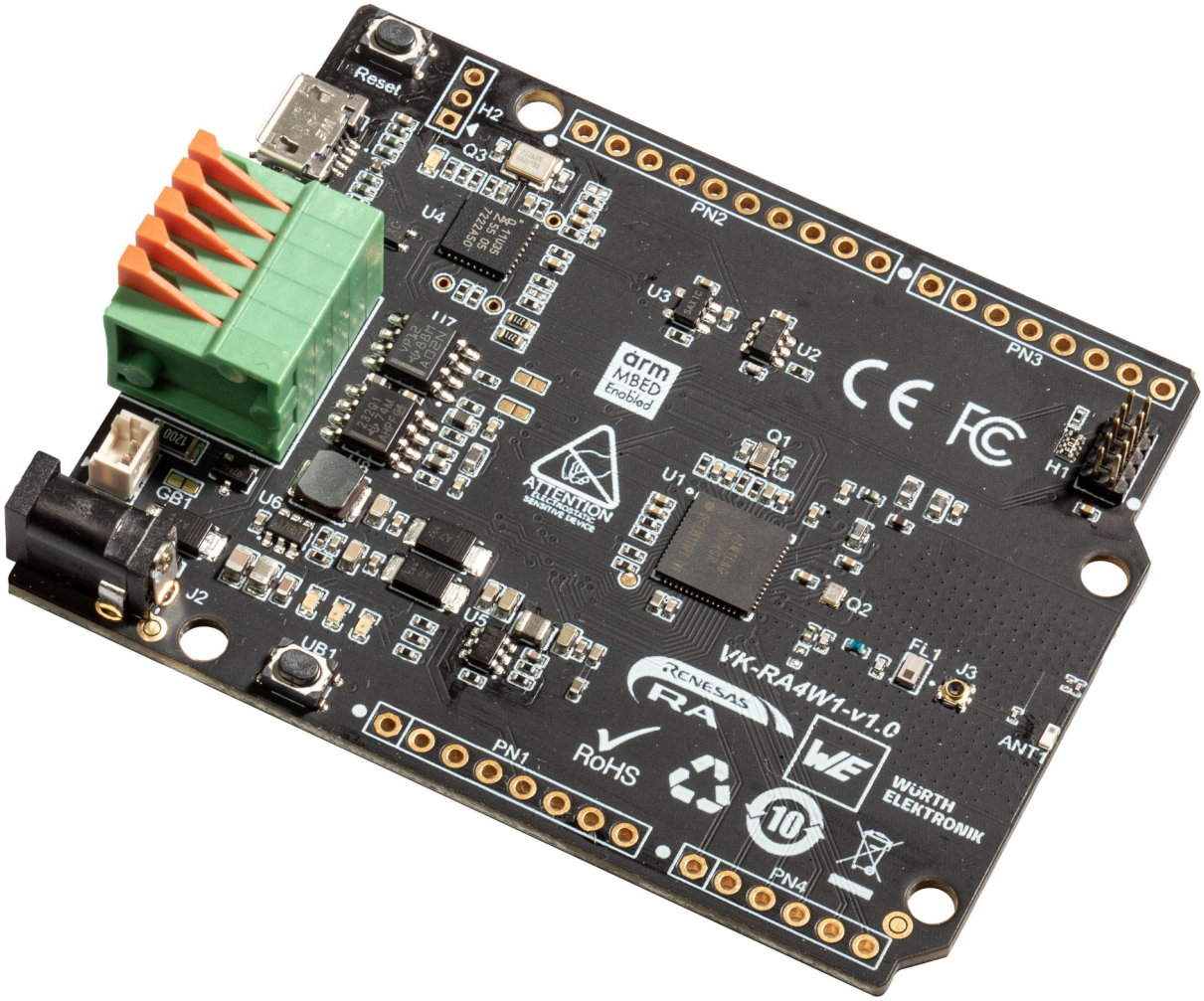
VK-RA4W1 FSP App Demo



VK-RA4W1 v1.0 Board

Content:

1. Introduction

VK-RA4W1 is development board, based on **Renesas RA4W1 ARM Cortex-M4 MCU**. The main purpose of this application is demonstration of board's hardware capabilities and main components workability (such as: USB, BlueTooth, LEDs, BTNs ets.)

2. Environment Setup

This demo is built with Renesas RA **F**lexible **S**oftware **P**ackage **3.7.0** and includes 1 project, setuped for **E²Studio**, **IAR** & **Keil**. It follows the folder structure below:

E²Studio (22.4.0)

```
Demo
+ -- E2
|     ` -- GATT
|           + …
|           ` -- Debug
|
+ - IAR
` - U5
```

IAR (9.20.1)

```
Demo
+ -- IAR
|     ` -- GATT
|           + …
|           ` -- Debug
|
+ - E2
` - U5
```

Keil uVision5 (5.32.0.0)

```
Demo
+ -- U5
|     ` -- GATT
|           + …
|           ` -- Objects
|
+ - E2
` - IAR
```

If you just want to see what demo actually does and you are not interested in compile & debug, use the precompiled hex files (located in `Demo\<IDE>\GATT\Debug(\Objects)\GATT.hex`)

3. Build

Look the Blinky project for more details about how to compile, debug & use IDE with the FSP.

4. Demo Apps

This application tests board's bluetooth hardware periphery and logs output on J1 µUSB connector (CDC: 115200,1,N,8). You can redirect the log through the J-Link debugger (J-Link RTT Viewer), but have to comment the line № **33** in file `common utils.h` :

(**`#include "SYSTEM_TTY/SYSTEM_TTY.h"`**).

4.1 GATT

This project is port of **ble_baremetal_ek_ra4w1** example (part of the official Renesas RA family ble sample applications). Its main goal is checking functionality of the bluetooth controller and its radio antenna ANT1. The cmd CLI console & the log system are removed from the original app, because the demo has log of its own (similar to the SEGGER's RTT Viewer). When launched, this demo turns VK-RA4W1 board in to GATT server, allowing the control of the **2** onboard LEDs (**R**ed & **Y**ellow) through the Renesas client (GATTbrowser). The client can also get a notification when user button UB1 is pressed or released. The control itself is happening through the LED Blink Rate Characteristic, which happens to be **8 bit** variable with the fallowing bit meaning:

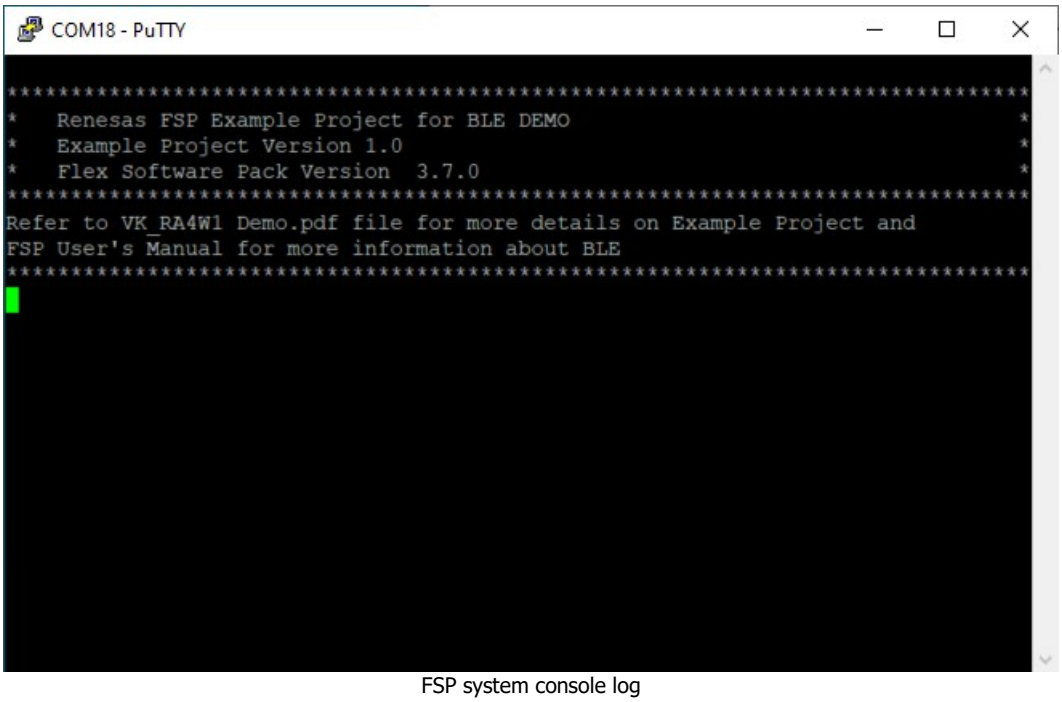
```
[ MSB 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 LSB ]
[ Blink Rate (50ms per bit) | Y | R     ]
```

Examples:

Writing 0x0**1** → turns **ON** the **R** LED Writing 0x0**9** → Toggles the **R** LED every **100 ms**

Writing 0x0**2** → turns **ON** the **Y** LED Writing 0x0**A** → Toggles the **Y** LED every **100 ms**

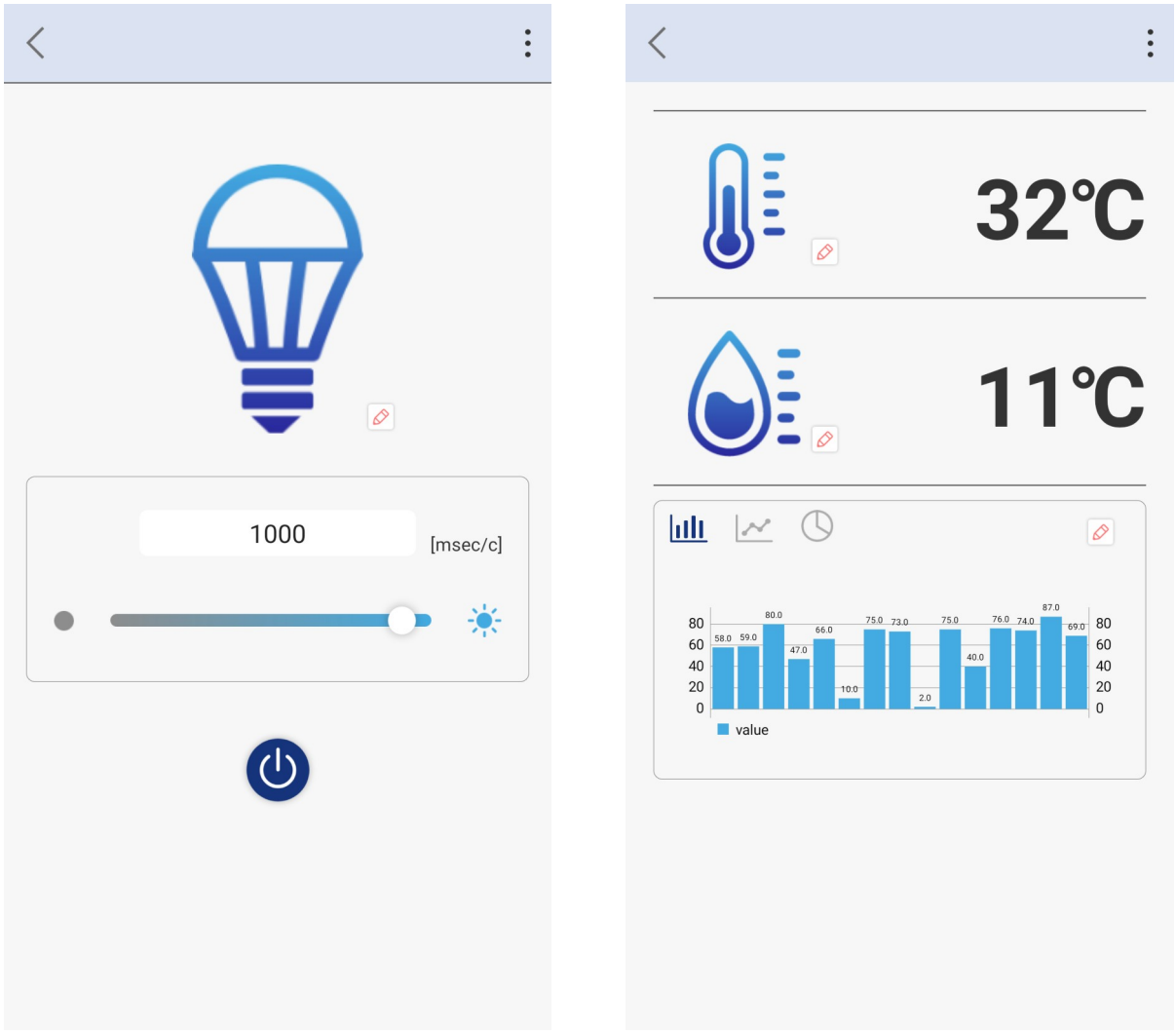
Writing 0x0**7** → Toggles the **R** & **Y** LEDs every **50 ms**



FSP system console log

4.2 TryBT

This project is slightly modified version of the GATT, but more user friendly, using TryBT Renesas client (available for Android & iOS). Instead of entering hex digits in the GATTbrowser, you can simply switch ON and OFF the blinking or slide the rate you want when LED **R**ed is on. Pressing the UB1, the app generates random data and populates its chart.



TryBT app interface

Hint:

In version 2.0 of the VK-RA4W1, there is humidity & temperature sensor onboard, you can actually modify TryBT firmware to get a real live sensor data. Of course to bypass random generation, you should touch the mobile app too, but this is not in the scope of this document.

Revision overview list

Revision number	Description changes
1.0	Initial

Vekatech Ltd.

63, Nestor Abadzhiev st.
4023 Plovdiv
Bulgaria
Tel.: +359 (0) 32 262362
info@vekatech.com

www.vekatech.com