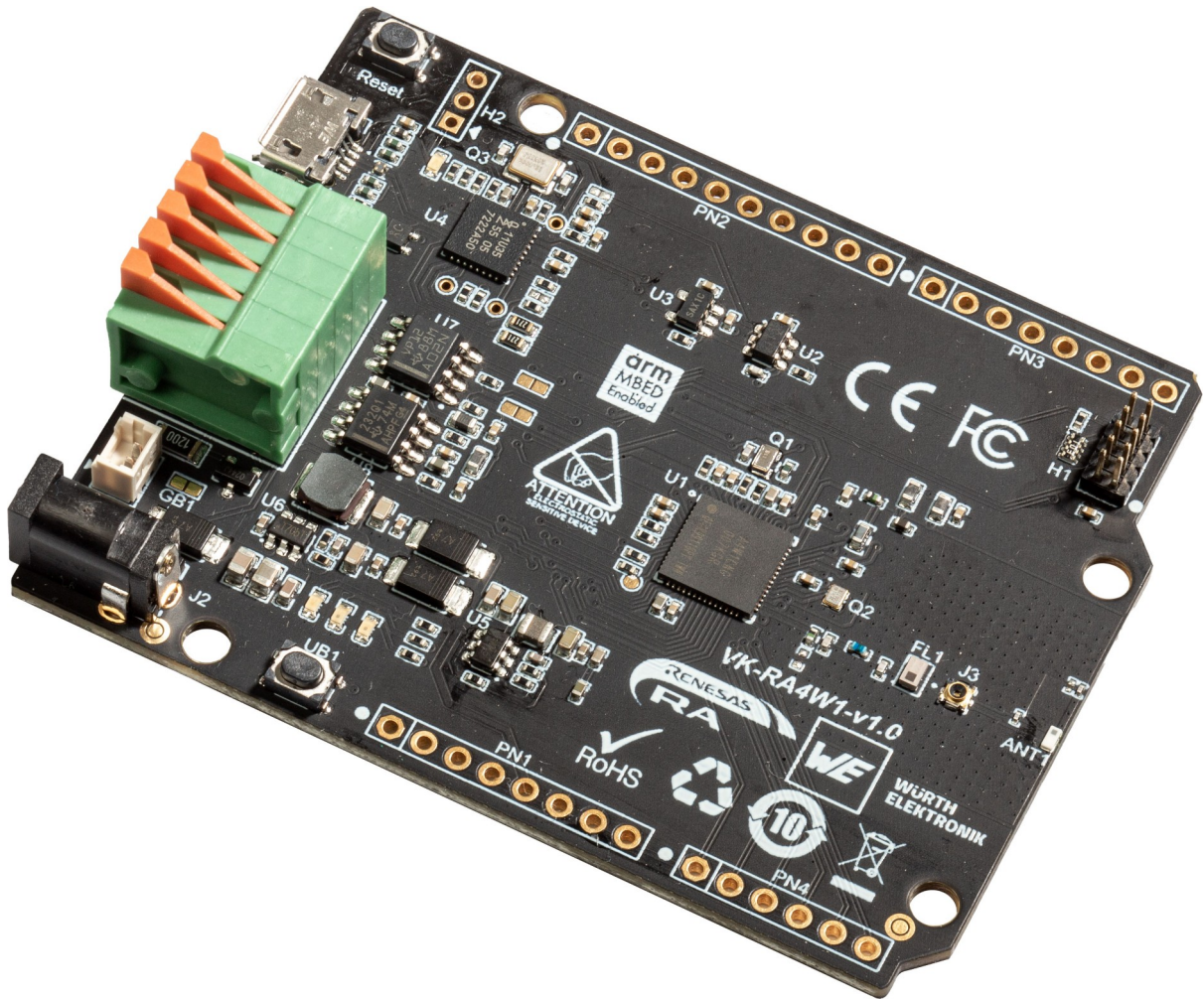


# VK-RA4W1 FSP Blinky



VK-RA4W1 v1.0 Board



# User's manual

---

## ***Content:***

<b>1. INTRODUCTION.....</b>	<b>3</b>
<b>2. ENVIRONMENT SETUP.....</b>	<b>3</b>
2.1 DRAG & DROP (COPY).....	3
2.2 DEBUG.....	4
<b>3. APP.....</b>	<b>5</b>



# User's manual

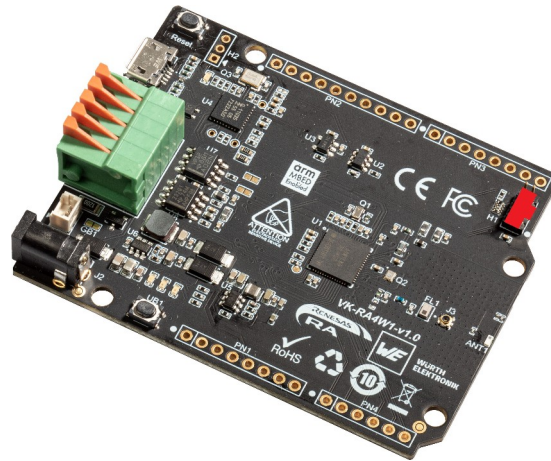
---

## 1. Introduction

[VK-RA4W1](#) is actually a development board, based on **Renesas RA Family** equipped with **ARM Cortex-M4 MCU**. The main purpose of this document is demonstration of drag & drop firmware programming and debugging capabilities of the board.

## 2. Environment Setup

Drag & drop function does not need anything. It is a built in function of the interface chip. You may need to do an extra drag & drop and stuff, if the interface chip happens to be empty or it's firmware is old (buggy) and needs upgrade, but it is unlikely. The board comes with a pre flashed interface chip, so it is all set and ready to test.



Debug connector

Debug function however, does require **CMSIS-DAP** support from the IDE of your choice (IAR & Keil). E<sup>2</sup>Studio is also an option, but requires external debugger (unfortunately it doesn't support CMSIS-DAP for now). There are 2 **FSP 3.7.0** projects (with different blink pattern), pre setupped for the 3 IDEs in question, so it is all set and ready to test.

### 2.1 **Drag & Drop (Copy)**

Connect the PC to the board's **μUSB** connector and wait appearance of a disk **VK-RA4W1** disk. The board's msd recognizes only \*.bin or \*.hex, so the firmware file should be one of those formats. After successful completion of the copping, the board will auto reset, which will allow the new firmware to run. If for some reason the copping fails, after the reset, the board's disk will contain a `FAIL.TXT` file, which will tell you what happened. You can test the copy function right away dropping the files `FSP\E2\Blink_I\Debug\Blink_I.hex` or `FSP\E2\Blink_II\Debug\Blink_II.hex`.



# User's manual

If disk with name **CRP DISABLD** appeared after connecting to the PC, this means, either interface chip's flash is empty or interface chip's update procedure is started (i.e. holding the reset button while powering the board through the  $\mu$ USB). Both cases require following attention:

- Find latest interface chip's firmware. Look for `????_LPC11U32_VK_RA4W1_0X0000.bin`.  
???? is version of the CMSIS-DAP firmware (at the time of writing of this manual it is **0254**)
- Open and see the content of the `CRP DISABLD` disk.
- Delete the file named `firmware.bin`.
- Drag & Drop or copy `????_LPC11U32_VK_RA4W1_0X0000.bin` to the `CRP DISABLD`.
- Wait for the copy operation to complete.
- Power cycle the board. The board will now enumerate and mount as `VK-RA4W1` disk.

## 2.2 Debug

There are 2 **BLINKY** projects, ready for debugging with 3 **IDE**, each of which follows the folder structure below:

<b>E<sup>2</sup>Studio</b> (22.4.0)	<b>IAR</b> (9.20.1)	<b>Keil uVision5</b> (5.32.0.0)
FSP	FSP	FSP
+ -- E2	+ -- IAR	+ -- U5
+ -- Blink_I	+ -- Blink_I	+ -- Blink_I
+ ...	+ ...	+ ...
\ -- Debug	+ -- Debug	\ -- Objects
	\ -- Release	
\ -- Blink_II	\ -- Blink_II	\ -- Blink_II
+ ...	+ ...	+ ...
+ - IAR \ -- Debug	+ - E2 + -- Debug	+ - E2 \ -- Objects
\ - U5	\ - U5 \ -- Release	\ - IAR

For **IAR** and **Keil** you need only USB cable connected to  $\mu$ USB and you are ready to debug. You also need to install [RA Smart Configurator](#) and have to integrate it in the **IDE**. Once done, you should launch **RASC** from the IDE menu bar, and hit **Generate Project Content** before the first build.

For **E<sup>2</sup>Studio** you need an external **JTAG** debugger connected to **H1** (the red rectangle). Pay attention to the connector orientation (tooth facing the edge of the board) as shown in the picture above (zoom it, if isn't noticeable). The **RASC** here is pre built in the studio, so you can directly hit build and every necessary file from the FSP will be added to the Blink project automatically.



# User's manual

## 3. App

After downloading `Blink_I` to the VK-RA4W1's flash and clean start (Reset), this program lights up the available board's LEDs in a specific pattern (running light). The board has 2 LEDs (red & yellow) and only 1 is ON for certain amount of time. By pressing the user BTN you can change the pattern like so: 100ms → 200ms → 400ms → 100ms → ... as can be seen it is cyclical. The app is built with Renesas RA Flexible Software Package **3.7.0** and uses **GPIO, External Interrupt & Timer** modules. It is bare metal, OSless, entirely interrupt driver program, leaving the main loop so empty, that the whole MCU time is all yours, just for your experiments and upgrades.

`Blink_II` is like `Blink_I`, but the pattern is pulsing light (i.e all LEDs are ON for a certain amount of time and then all are OFF for the same amount of time). The BTN cycle and everything else is the same.

### Revision overview list

Revision number	Description changes
<b>1.0</b>	Initial

Vekatech Ltd.

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

[www.vekatech.com](http://www.vekatech.com)