REALTEK

Ameba-ZII DEV User Manual

Realtek Semiconductor Corp.
No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan
Tel.: +886-3-578-0211. Fax: +886-3-577-6047
www.realtek.com

DISCLAIMER

Please Read Carefully:

Realtek Semiconductor Corp., (Realtek) reserves the right to make corrections, enhancements, improvements and other changes to its products and services. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Reproduction of significant portions in Realtek data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Realtek is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions.

Buyers and others who are developing systems that incorporate Realtek products (collectively, "Customers") understand and agree that Customers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Customers have full and exclusive responsibility to assure the safety of Customers' applications and compliance of their applications (and of all Realtek products used in or for Customers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Customer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Customer agrees that prior to using or distributing any applications that include Realtek products, Customer will thoroughly test such applications and the functionality of such Realtek products as used in such applications.

Realtek's provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation kits, (collectively, "Resources") are intended to assist designers who are developing applications that incorporate Realtek products; by downloading, accessing or using Realtek's Resources in any way, Customer (individually or, if Customer is acting on behalf of a company, Customer's company) agrees to use any particular Realtek Resources solely for this purpose and subject to the terms of this Notice.

Realtek's provision of Realtek Resources does not expand or otherwise alter Realtek's applicable published warranties or warranty disclaimers for Realtek's products, and no additional obligations or liabilities arise from Realtek providing such Realtek Resources. Realtek reserves the right to make corrections, enhancements, improvements and other changes to its Realtek Resources. Realtek has not conducted any testing other than that specifically described in the published documentation for a particular Realtek Resource.

Customer is authorized to use, copy and modify any individual Realtek Resource only in connection with the development of applications that include the Realtek product(s) identified in such Realtek Resource. No other license, express or implied, by estoppel or otherwise to any other Realtek intellectual property right, and no license to any technology or intellectual property right of Realtek or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Realtek products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Realtek Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Realtek under the patents or other Realtek's intellectual property.

Realtek's Resources are provided "as is" and with all faults. Realtek disclaims all other warranties or representations, express or implied, regarding resources or use thereof, including but not limited to accuracy or completeness, title, any epidemic failure warranty and any implied warranties of merchantability, fitness for a particular purpose, and non-infringement of any third-party intellectual property rights. Realtek shall not be liable for and shall not defend or indemnify Customer against any claim, including but not limited to any infringement claim that related to or is based on any combination of products even if described in Realtek Resources or otherwise. In no event shall Realtek be liable for any actual, direct, special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of Realtek's Resources or use thereof, and regardless of whether Realtek has been advised of the possibility of such damages. Realtek is not responsible for any failure to meet such industry standard requirements.

Where Realtek specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Customers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any Realtek products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death. Such equipment includes, without limitation, all medical devices identified by the U.S.FDA as Class III devices and equivalent classifications outside the U.S.

Customers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Customers' own risk. Customers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Customer will fully indemnify Realtek and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT

Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.

Revision History

Revision	Release Date	Summary
0.1	2019/10/21	Initial draft

Application Note All information provided in this document is subject to legal disclaimers. © REALTEK 2018. All rights reserved.

4

Table of Contents

List of Figures

List of Table

1 Demo Board User Guide

1.1 PCB Layout Overview

RTL8720C embedded on Ameba-ZII DEV demo board, which consists of various I/O interfaces. For the details of the HDK, please contact us for further reference.
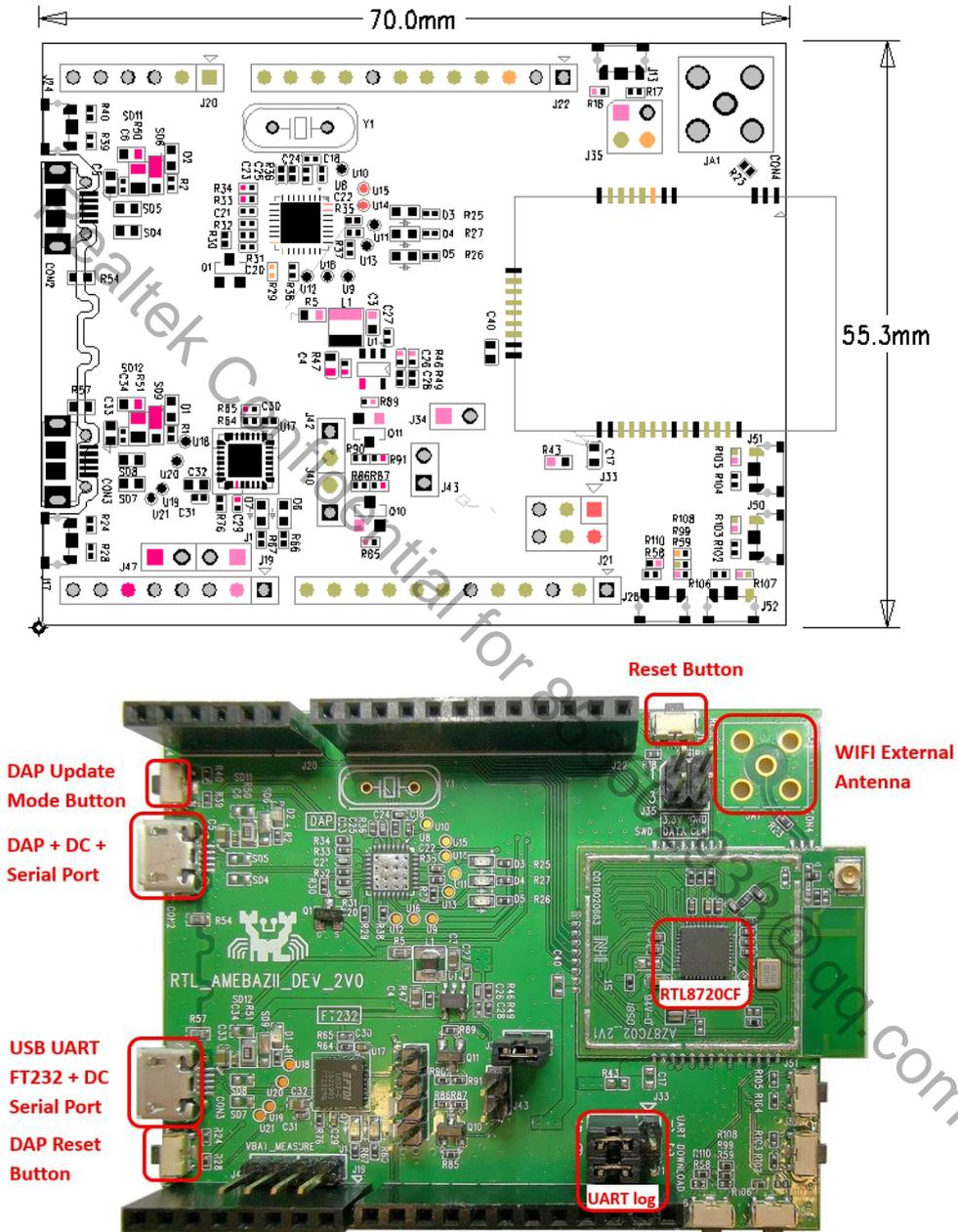
Figure 1-2 Ameba-ZII 2V0 Dev Board PCB Layout

1.2   Pin Mux Alternate Functions

1.2.1   Pin Mux Table

Pin Name	SPIC-Flash/SDIO	JTAG	UART	SPI/WL_LED/EXT_32K	I2C	PWM
GPIOA_0		JTAG_CLK	UART1_IN	EXT_32K		PWM[0]
GPIOA_1		JTAG_TMS	UART1_OUT	BT_LED		PWM[1]
GPIOA_2		JTAG_TDO	UART1_IN	SPI_CSn	I2C_SCL	PWM[2]
GPIOA_3		JTAG_TDI	UART1_OUT	SPI_SCL	I2C_SDA	PWM[3]
GPIOA_4		JTAG_TRST	UART1_CTS	SPI_MOSI		PWM[4]
GPIOA_5			UART1_RTS	SPI_MISO		PWM[5]
GPIOA_6						PWM[6]
GPIOA_7	SPI_M_CS			SPI_CSn		
GPIOA_8	SPI_M_CLK			SPI_SCL		
GPIOA_9	SPI_M_DATA[2]		UART0_RTS	SPI_MOSI		
GPIOA_10	SPI_M_DATA[1]		UART0_CTS	SPI_MISO		
GPIOA_11	SPI_M_DATA[0]		UART0_OUT		I2C_SCL	PWM[0]
GPIOA_12	SPI_M_DATA[3]		UART0_IN		I2C_SDA	PWM[1]
GPIOA_13			UART0_IN			PWM[7]
GPIOA_14	SDIO_INT		UART0_OUT			PWM[2]
GPIOA_15	SD_D[2]		UART2_IN	SPI_CSn	I2C_SCL	PWM[3]
GPIOA_16	SD_D[3]		UART2_OUT	SPI_SCL	I2C_SDA	PWM[4]
GPIOA_17	SD_CMD					PWM[5]
GPIOA_18	SD_CLK					PWM[6]
GPIOA_19	SD_D[0]		UART2_CTS	SPI_MOSI	I2C_SCL	PWM[7]
GPIOA_20	SD_D[1]		UART2_RTS	SPI_MISO	I2C_SDA	PWM[0]
GPIOA_21			UART2_IN		I2C_SCL	PWM[1]
GPIOA_22			UART2_OUT	LED_0	I2C_SDA	PWM[2]
GPIOA_23				LED_0		PWM[7]

Table 1-1 GPIOA Pin MUX: DEV_2V0 Board

Note: This table may not be up-to-date, please check the HDK and datasheet for more details.

1.2.2   Pin-Out Reference
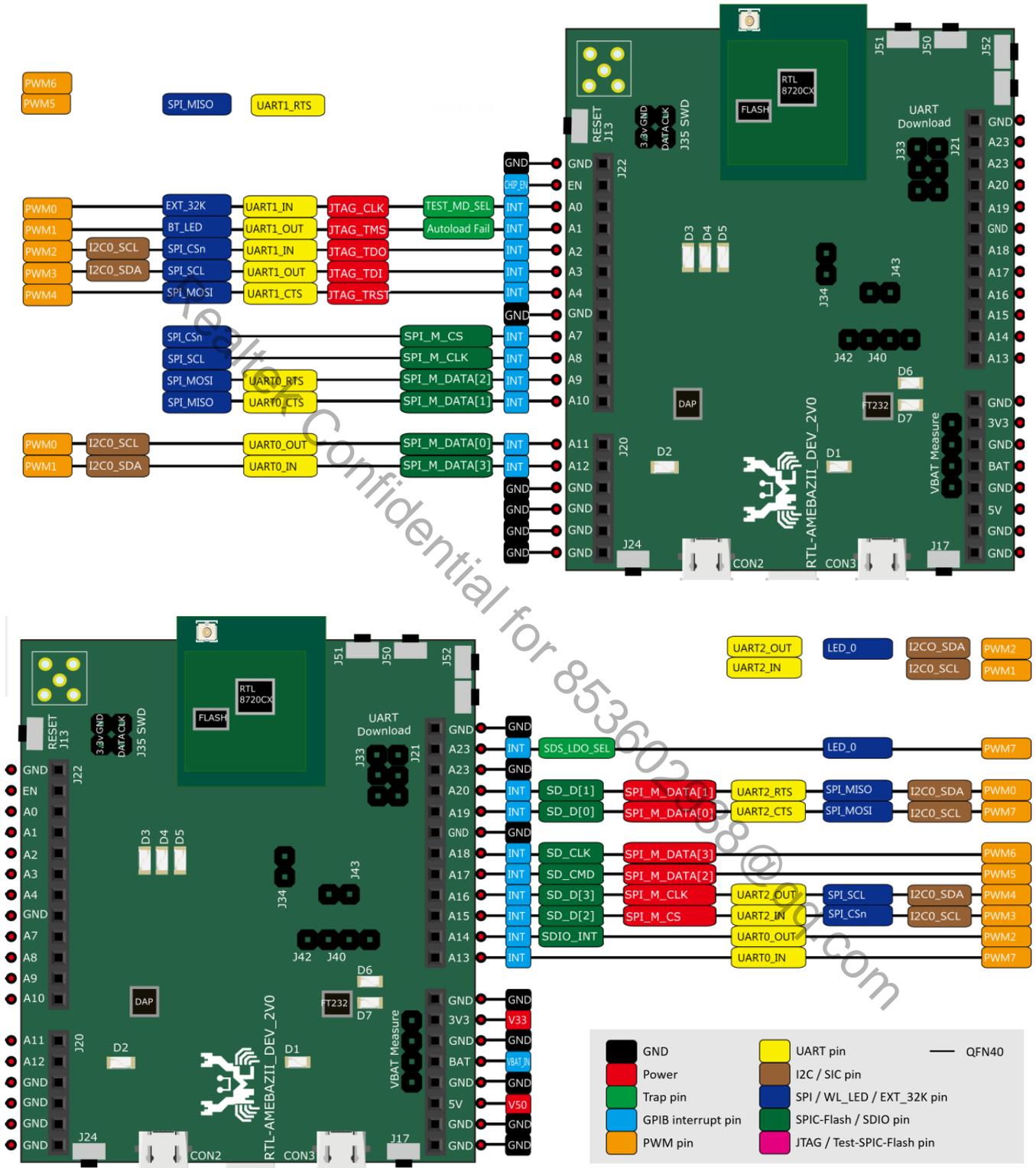
Figure 1-3 Pin Out Reference for DEV_2V0

2 SDK Build Environment Setup

2.1 Debugger Settings

To download code or debug on Ameba-ZII, user needs to make sure the debugger is setup properly.

Ameba-ZII supports **J-Link** for code download and entering debugger mode. The settings are described below.

2.1.1 J-Link Debugger

2.1.1.1 Connection

Ameba-ZII supports J-Link debugger. you need to connect the **Serial Wire Debug** (SWD) connector of Ameba-ZII to J-Link debugger as shown below and then connect J-Link to PC. You can refer to section 1.2.2 for SWD pin definitions.
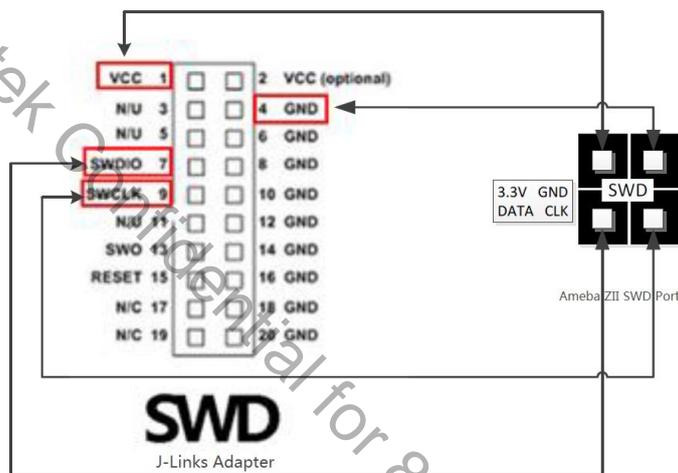
Figure 2-1 Connection between J-Link Adapter and Ameba-ZII SWD connector

Note:

1. *To be able to debugger Ameba-ZII which is powered by Cortex-M33, user needs a J-Link debugger with the latest hardware version (Check https://wiki.segger.com/Software_and_Hardware_Features_Overview for details).*

2.1.1.2 Setups on Windows OS

To be able to use J-Link debugger, you need to firstly install J-Link GDB server.

Please check http://www.segger.com and download "**J-Link Software and Documentation Pack**" (https://www.segger.com/downloads/jlink).

Note: To support TrustZone feature, it's better to download the **latest version** of J-Link Software. Version 6.40 is used to prepare this document.

The process of is as follows:

1. Install J-Link GDB server.

Please check http://www.segger.com and download "J-Link Software and Documentation Pack" (https://www.segger.com/downloads/jlink).

Figure 2-2 J-Link Setup Interface

2. Open installation location of 'JLink_V640' and run "**JLinkGDBServer.exe**" to check connection.

3. Make sure the configuration is fine and click 'OK'.
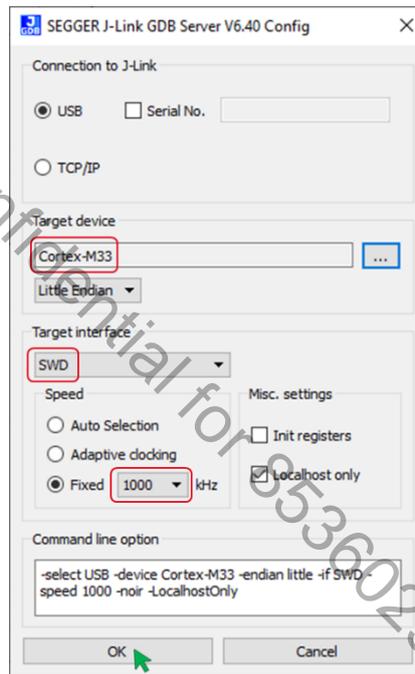
Figure 2-3 J-Link GDB server UI under Windows

4. Check if the below information is shown properly.
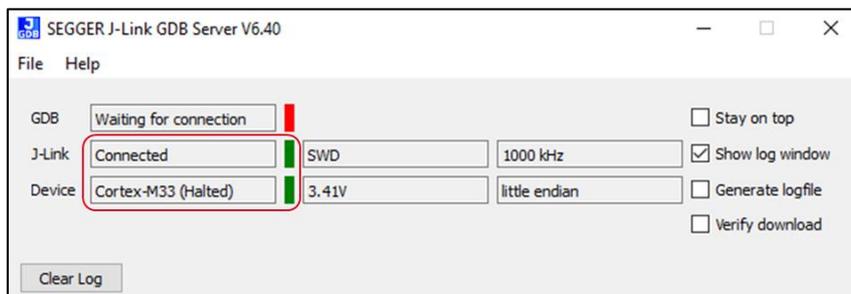
Figure 2-4 J-Link GDB server connect under Windows

Note: If J-Link GDB Server is unable to detect the device, try re-connecting the wires and re-open 'JLinkGDBServer.exe' may solve the problem.

2.2   Log UART Settings

To be able to start development with the demo board, Log UART must be connected properly.

Different versions of EVBs have different connections.

2.2.1   EVB v2.0

By default, UART2 (GPIOA_15 / GPIOA_16, c heck figure 1-3) is used as system log UART. User needs to connect jumpers to **J33** for **CON3 (FT232)** or **CON2 (DAP)**.

1)   Connection to log UART via **FT232 (CON3):**
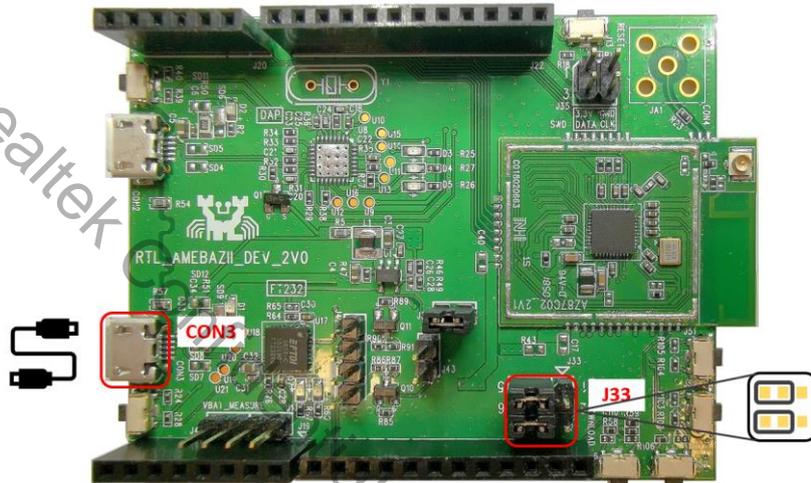
Figure 2-2 Log UART via FT232 on EVB V2.0

2.3   IAR Environment Setup

IAR IDE (integrated development environment) only supports Windows OS, this section is applicable for **Windows OS only**.

2.3.1   Install and Setup IAR IDE in Windows

IAR IDE provides the toolchain for Ameba-ZII. It allows users to write programs, compile and upload them to your board. Also, it supports step-by-step debug function.

User can visit the official website of IAR Embedded Workbench and install the IDE by following its instructions.

Note: Please use IAR version 8.30 or above.

2.3.2   IAR Project Introduction

2.3.2.1   Ignore Secure Project

Currently users can use **ignore secure mode**. 'project_is' (ignore secure) is the project without **TrustZone** configuration. This project is easier to develop and suit for first-time developer.

2.3.2.1.1 Compilation

1)   Open SDK/project/realtek_amebaz2_v0_example/EWARM-RELEASE/Project_is.eww.

2)   Confirm application_is in Work Space, right click application_is and choose "Rebuild All" to compile.

3)   Make sure there is no error after compile.

2.3.2.1.2 Generating Image Binary

After compile, the images partition.bin, bootloader.bin, firmware_is.bin and flash_is.bin can be seen in the EWARM-RELEASE\Debug\Exe.

1)   partition.bin stores partition table, recording the address of Boot image and firmware image;

2)   bootloader.bin is bootloader image;

3)   firmware_is.bin is application image;

4)   flash_is.bin links partition.bin, bootloader.bin and firmware_is.bin. Users need to choose flash_is.bin when downloading the image to board by PG Tool.

2.3.2.1.3 Download

After a successfully compilation and '**flash_is.bin**' is generated without error, user can either

1) Directly download the image binary on to demo board from IAR IDE (as below)
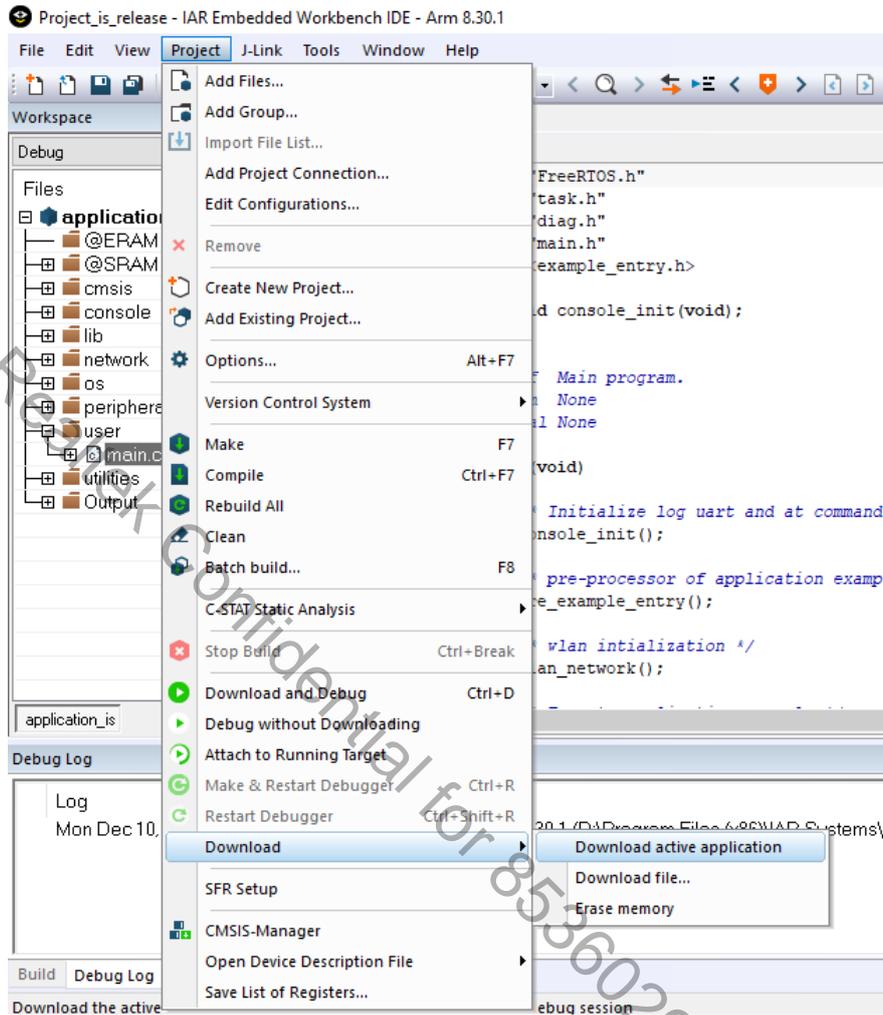
Figure 2-7 IAR download binary on flash

Note: Please make the project first when some code is modified before download the bin file on the board, otherwise the download will fail and below logs will be shown.
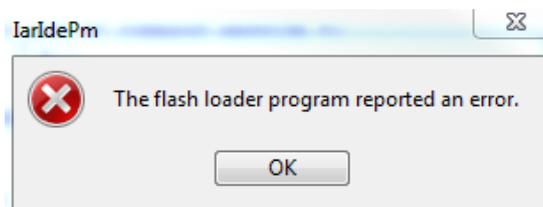
Figure 2-8 IAR download code on flash error message on IDE
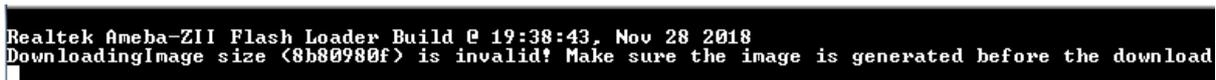
Figure 2-9 IAR download code on flash error message on Log UART

2) Or using the PG tool for Ameba-ZII (Will not be shown here, please check chapter 3 for details).

2.3.2.1.4 Compilation

1) Open SDK/project/realtek_amebaz2_v0_example/EWARM-RELEASE/Project_tz.eww.

2) Confirm '***application_ns'*** and '***application_s'*** are in Work Space.

3) Right click '***application_s'*** and click "**Rebuild All**" to compile '***application_s'*** first. If '***application_s'*** is compiled successfully, it will generate a file named '***application_s_import_lib.o'*** and the file will be put in "lib" folder of '***application_ns'***, shown in Figure 2-10.
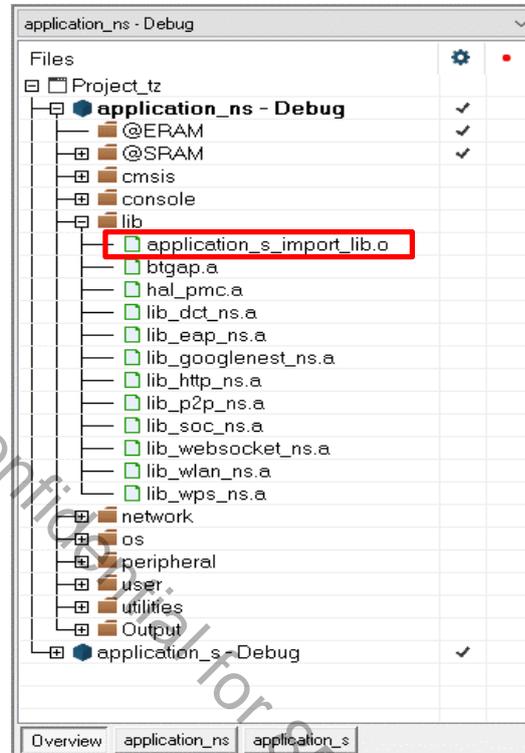
Figure 2-10 application_s compile result

4) Make sure '***application_s'*** is compiled successfully and the file '***application_s_import_lib.o'*** has been put under "lib" in '***application_ns'***.

5) Right click '***application_ns'*** and click "**Rebuild All**" to build '***application_ns'***.

6) Make sure the '***application_ns'*** is compiled successfully.

2.3.2.1.5 Generating image binary

After compile, the images *partition.bin, bootloader.bin, firmware_tz.bin* and *flash_tz.bin* can be seen in the EWARM-RELEASE\Debug\Exe.

1) ***partition.bin*** stores partition table, recording the address of Boot image and firmware image;

2) ***bootloader.bin*** is bootloader image;

3) ***firmware_tz.bin*** is application image;

4) ***flash_tz.bin*** links *partition.bin, bootloader.bin* and *firmware_tz.bin*. Users need to choose *flash_tz.bin* when downloading the image to board by PG Tool.

2.3.2.1.6 Download

After a successfully compilation and '***flash_tz.bin'*** is generated without error, user can either

1) Directly download the image binary on to demo board from IAR IDE (as below)
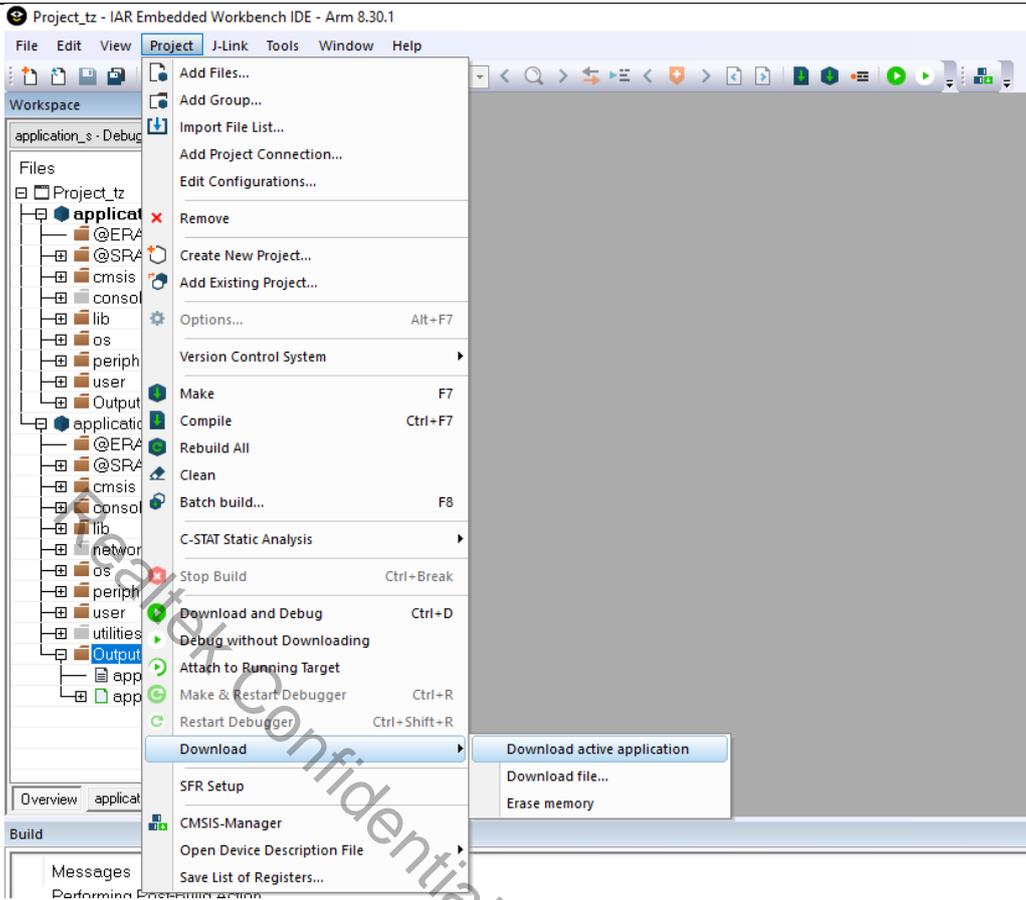
Figure 2-11 IAR download binary on flash

2) Or using the PG tool for Ameba-ZII (Will not be shown here, please check chapter 3 for details).

2.4 GCC Environment on Windows (Using Cygwin)

2.4.1 Install Cygwin

Cygwin a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows. It provides the GCC toolchain for Ameba-ZII.

User can visit the official website of Cygwin and install by following its instructions.

- During the Cygwin installation, please install "**math**" "bc: Arbitrary precision calculator language"

- During the Cygwin installation, please install "**devel**" "make: The GNU version of the 'make' utility"

Note: Please use **Cygwin 32bit**.

2.4.2 Building the Non-Trust Zone Project

2.4.2.1 Compile Project on Cygwin

1) Open "Cygwin Terminal"

2) Direct to compile path. Enter command "cd /SDK/project/realtek_amebaz2_v0_example/GCC-RELEASE"

3) Clean up pervious compilation files. Enter command "**make clean**"

4) Build all libraries and application. Enter command "**make all**"

5) Make sure there is no error after compile.

2.4.2.2 Generating Image Binary

After compile, the images partition.bin, bootloader.bin, firmware_is.bin and flash_is.bin can be seen in different folders of \GCC-RELEASE.

1) partition.bin stores partition table, recording the address of Boot image and firmware image; located at folder \GCC-RELEASE;

2) bootloader.bin is bootloader image; located at folder \GCC-RELEASE\bootloader\Debug\bin;

3) firmware_is.bin is application image; located at folder \GCC-RELEASE\application_is\Debug\bin;

4) flash_is.bin links partition.bin, bootloader.bin and firmware_is.bin. Located at folder \GCC-RELEASE\application_is\Debug\bin.

Users need to choose '**flash_is.bin**' when downloading the image to board by PG Tool.

2.4.2.3 Download

After a successfully compilation and '**flash_is.bin**' is generated without error, user can either

1) Directly download the image binary on to demo board from Cygwin (as below)

   Connect **SWD** to board and open "**JLinkGDBServer.exe**". Please refer to 2.2.1 Jlink for SWD connection.

   Enter command "make flash" at Cygwin.

2) Or using the PG tool for Ameba-ZII (Will not be shown here, please check chapter 3 for details).

2.5   GCC Environment on Ubuntu/Linux

2.5.1   Verify Device Connections

Once the JLink software is installed, the connections to the ubuntu machine of the device need to be verified.

1. Ensure that the JLink debugger is connected to the target and the USB device is connected to the Ubuntu/Linux machine.
2. Ensure that the micro-usb is connected to CON3 and plugged into the Ubuntu/Linux machine via USB in order to receive serial logs.
3. To verify if both devices i.e. the JLink device and the device serial port have been detected properly we can use the "lsusb" command to see list of devices as shown below:

4. As you can see above the SEGGER J-Link and the FTDI USB UART device have been successfully detected.

2.5.2   Compile and Generate Binaries

1. Open the Ubuntu/Linux terminal.
2. Direct to compile path. Enter command "cd /SDK /project/realtek_amebaz2_v0_example/GCC-RELEASE"
3. Clean up pervious compilation files. Enter command "make clean"
4. Build all libraries and application. Enter command "make all"
5. Once the build is successful, you should be able to see the success logs as shown below.

2.5.3   Download and Flash Binaries

There are in-built scripts in the makefile that initiate download and flashing of the software via JLink. In order to flash successfully, the JLinkGDBServer needs to be initiated manually by the user and successful connection needs to be ensured. The JLink GDB server must be active and connected to the target before any type of flash action is taken. In order to start the JLink GDB server, follow the steps in 2.2.1.2.

2.5.3.1 Initiate Flash Download

Once the JLink GDB server is set up as per the instructions given before, perform the following steps to initiate the flash download.

1. Proceed back to the previous terminal where the SDK was made, without closing the terminal from which GDB server is running

2. Run the command "make setup GDB_SERVER=jlink or pyocd" to select GDB Server.

3. Run the command "sudo make flash"

4. If the flash download is successful, the following log will be printed

```
Flash Download done, exist
A debugging session is active.

        Inferior 1 [Remote target] will be killed.

Quit anyway? (y or n) [answered Y; input not from terminal]
make[1]: Leaving directory '/home/parallels/sdk-ameba-v7.1a_rc4_gcc/project/realtek_amebaz2_v0_example/GCC-RELEASE'
parallels@ubuntu:~/sdk-ameba-v7.1a_rc4_gcc/project/realtek_amebaz2_v0_example/GCC-RELEASE$
```

2.5.3.2 Debug

After a successfully downloading, user can debug with pyOCD + DAPLink enabled HDK or using JLINKGDBServer + JLINK by following command
"make debug_tz"
Before using "make debug_tz", "make setup GDB_SERVER=jlink or pyocd" to select GDB Server is necessary.

3 Image Tool

3.1 Introduction

This chapter introduces how to use Image Tool to generate and download images. As show in Figure 3-1, Image Tool has two menu pages:

- Download: used as image download server to transmit images to Ameba through UART.

- Generate: contact individual images and generate a composite image.

Please download the '**PG Tool Release Package**' and browse the image tool document '**UM0503**'.

Note: If you need to download code via external uart, must use FT232 USB To UART dongle.
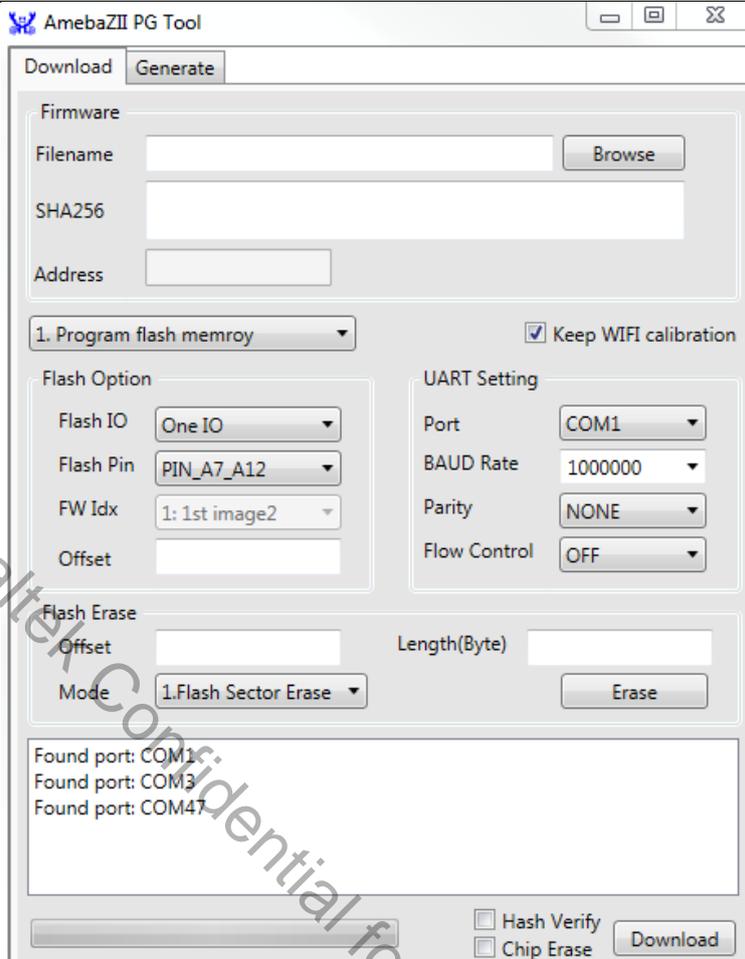
Figure 3-1 AmebaZII Image Tool UI

3.2 Environment Setup

3.2.1 Hardware Setup

3.2.1.1 EVB V2.0

User needs to connect CON3 to user's PC via a Micro USB cable. Add jumpers for J34 and J33 (J33 is for log UART which has two jumpers) if there is no connection.
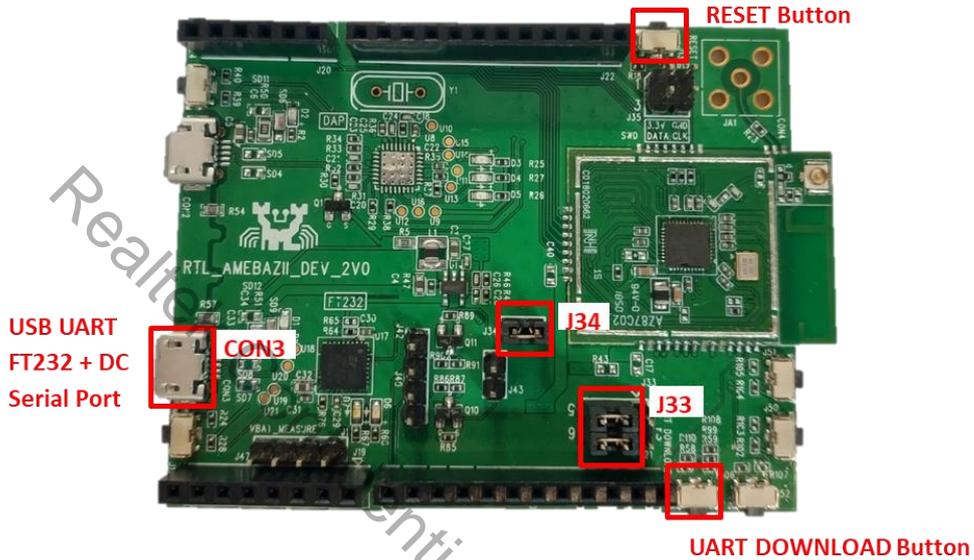
Figure 3-2 Ameba-ZII EVB V2.0 Hardware Setup

3.2.2 Software Setup

- Environment Requirements: EX. WinXP, Win 7 Above, Microsoft .NET Framework 3.5
- AmebaZII_PGTool_v1.0.1.exe

3.3   Image Download

User can download the image on demo board by following below steps:

1) Trigger Ameba-ZII chip enter UART download mode by
    a. For EVB V2.0, to enter UART download mode:
        Press and hold the UART DOWNLOAD button then press the RESET button and release both buttons. And make sure the log UART is connected properly (refer to section 2.3).
    b. Press RESET button and release, then below log should be shown on log UART console. (Please remember to disconnect the log UART before using Image Tool to download, because the tool will also need to connect to this log UART port)
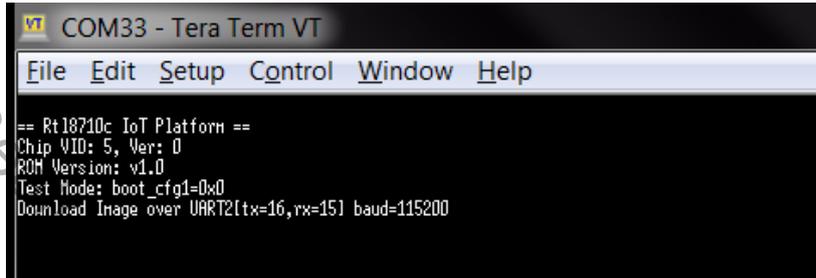
Figure 3-4 Ameba-ZII UART download mode

2) Choose the correct UART port (use rescan to update the port list)

3) "Browse" to choose the image to be downloaded (flash_xx.bin)

4) Choose "Mode" "1. Program flash memory"

5) Choose correct "Flash Pin" according to the IC part number

Flash Pin	IC part number
PIN_A7_A12	RTL8710CX/RTL8720CM
PIN_B6_B12	RTL8720CF

6) Click "Download" to start downloading image. While downloading, the status will be shown on the left bar.

Note: It's recommended to use the default settings unless user is familiar with them.