

Ameba-ZII DEV User Manual



Realtek Semiconductor Corp.

Midential For OSSOC 2000 re P No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan 99. COM Tel.: +886-3-578-0211. Fax: +886-3-577-6047 www.realtek.com



COPYRIGHT

©2018 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

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 noncompliance 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.

ree ation of the source of the

Revision History

Revision	Release Date	Summary
0.1	2019/10/21	Initial draft
	P	
	9/2	
	°/	
	(
		Ŭ,
		20
		(\mathcal{O})
		<i>Y</i> ₀
		· Com

Table of Contents

COPYRIGHT	2						
DISCLAIMER	2						
TRADEMARKS							
USING THIS DOCUMENT							
Revision History	4						
Table of Contents	5						
List of Figures	7						
List of Table	8						
1 Demo Board User Guide	9						
1.1 PCB Layout Overview	9						
1.2 Pin Mux Alternate Functions	10						
1.2.1 Pin Mux Table							
1.2.2 Pin-Out Reference	11						
2 SDK Build Environment Setup	12						
2.1 Debugger Settings	12						
2.1.1 J-Link Debugger	12						
2.2 Log UART Settings	14						
2.2.1 EVB v2.0	14						
2.3 IAR Environment Setup	15						
2.3.1 Install and Setup IAR IDE in Windows	15						
2.3.2 IAR Project Introduction							
2.4 GCC Environment on Windows (Using Cygwin)							
2.4.1 Install Cygwin							
2.4.2 Building the Non-Trust Zone Project	19						
2.5 GCC Environment on Ubuntu/Linux	20						
2.5.1 Verify Device Connections	20						
2.5.2 Compile and Generate Binaries	20						
2.5.3 Download and Flash Binaries	20						
3 Image Tool	21						
3.1 Introduction	21						
3.2 Environment Setup							
Application roote All information provided in this document is subject to legal discialmers.	SEALTEN 2018. All rights reserved.						

ÄKÄREALTEK

Ameba-ZII Application Note

3.2	.1	Hardware Setup	23
3.2	.2	Software Setup	23
3.3	Ima	ge Download	24

Realter Confidential for 853602938 Confidential for 853602938 Confidential for 853602938 Confidential for 853602938 Confidence on the format of the format o

KEALTEK

List of Figures

Figure 1-1 Top View of Ameba-ZII 2V0 Dev Board	9
Figure 1-2 Ameba-ZII 2V0 Dev Board PCB Layout	9
Figure 1-3 Pin Out Reference for DEV_2V0	11
Figure 3-1 Connection between J-Link Adapterand Ameba-ZII SWD connector	12
Figure 3-4 Log UART via FT232 on EVB V2.0	14
Figure 4-1 AmebaZII Image Tool UI	22
Figure 4-2 Ameba-ZII EVB V2.0 Hardware Setup	23

21 Realies Confidential for ossociasose equ.com

List of Table

Table 1-1 GPIOA Pin MUX: DEV_2V0 Board1	0
---	---

Realtes Confidential for ossologoge equ.com

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

REALTEK

1.2 Pin Mux Alternate Functions

1.2.1 Pin Mux Table

Pin Name	SPIC-Flash/SDIO	JTAG	UART	SPI/WL_LED/EXT_32K	12C	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	2/4	JTAG_TRST	UART1_CTS	SPI_MOSI		PWM[4]
GPIOA_5	64		UART1_RTS	SPI_MISO		PWM[5]
GPIOA_6	· C					PWM[6]
GPIOA_7	SPI_M_CS	<i>r</i> .		SPI_CSn		
GPIOA_8	SPI_M_CLK	00		SPI_SCL		
GPIOA_9	SPI_M_DATA[2]		UARTO_RTS	SPI_MOSI		
GPIOA_10	SPI_M_DATA[1]		UARTO_CTS	SPI_MISO		
GPIOA_11	SPI_M_DATA[0]		UARTO_OUT	P	I2C_SCL	PWM[0]
GPIOA_12	SPI_M_DATA[3]		UARTO_IN	U S	I2C_SDA	PWM[1]
GPIOA_13			UART0_IN	000		PWM[7]
GPIOA_14	SDIO_INT		UART0_OUT	` <i>O</i> ??		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				7.0	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

SDK Build Environment Setup 2

Debugger Settings 2.1

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 Adapterand Ameba-ZII SWD connector

Note:

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

2.1.1.2 Setups on Windows OS

@99.con 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.seqger.com and download "J-Link Software and Documentation Pack" (https://www.segger.com/downloads/jlink).

REALTEK

SEGGER - J-Link V6.40 S	etup	– 🗆 ×
		Welcome to SEGGER - J-Link V6.40 Setup Setup will guide you through the installation of SEGGER -
Embedded Studio Powerful C/C++ IDE available for • Windows • Uinows • Uinows • Mac OS X	SEGGER	L'is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer. Click Next to continue.
Download Trial		Next > Cancel

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'.

	SEGGER J-Link GDB Server V6.40 Config	×
4/2	Connection to J-Link	
×	USB Serial No.	
C		
	Target device	
	Cortex-M33	
	Turnet institution	
	SWD •	
	Speed Misc. settings	
	O Auto Selection	
	Adaptive docking Fixed 1000 KHz	
	Command line option	
	-select USB -device Cortex-M33 -endian little -if SWD speed 1000 -noir -LocalhostOnly	2
		50
	OK Cancel	
	Figure 2-3 J-Link GDB server UI under	Windows
		40
Check if the below information is shown	properly.	7. C
		$\sim 0_{\rm A}$
SEGGER J-Link GDB Ser	ver V6.40	- x 7
File Help		
GDB Waiting for conner	tion	Stay on top
J-Link Connected	SWD 1000 kHz	Snow log window
Device Cortex-M33 (Halte	a) [3.41V] little endian	
Clear Log		

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.

4.

REALTEK

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):



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)

Project_is_release	e - IA	R Embedded Workbench IDE - Arm 8.30.1		
File Edit View	Proj	ject J-Link Tools Window Help		
i 🗅 🗅 🔛 🕋	G	Add Files		- < Q, > ⇆ 🖻 < 📮 > 🕢 🖻
Workspace		Add Group	- 1	
Debug	[+]	Import File List	- 1	
Filos		Add Project Connection	1	FreeRTOS.h"
E annlication		Edit Configurations		task.h"
E ■ ■ @ERAM	×	Remove	-	main.h"
- 🖃 🛋 @SRAM				example_entry.h>
	U	Create New Project		d concole init(moid).
⊢⊞ ■ console	0	Add Existing Project		a console_init (Vola);
🕂 🕀 🖬 network	\$	Options Alt+F	7	Main program
P+⊞ ■ OS		Version Control System	► a	None
	-		- 2.	l None
🛏 🗗 🖬 main.c		Make F	7	woid)
E i utilities		Compile Ctrl+F	7	volu)
Oatput	9	Rebuild All	e	Initialize log uart and at command
	1	Clean	01	nsole_init();
	9	Batch build F	в	pre-processor of application exampl
		C-STAT Static Analysis	•	<pre>e_example_entry();</pre>
	8	Stop Build Ctrl+Break	k ^e	wlan intialization */
	0	Download and Bebug	, '	an_network();
application is	Υ.	Debug without Downloading	1	
	Å	Attach to Rupping Tarret		
Debug Log	ě	Make & Bestart Debugger		
Log	č	Bartart Dabuagar		
Mon Dec 10,	č	Download),	20 1 (DADroaron Files & 20) UAD Custems E
	_		Ó	Download file
		SFR Setup	사	Frace memory
	1	CMSIS-Manager	Y	Endse mellioly
		Open Device Description File		~6'
Build Debug Log		Save List of Registers		-02
Download the active	_			ebug session

Figure 2-7 IAR download binary on flash

ۍ ک 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

Realtek Ameba-ZII Flash Loader Build @ 19:38:43, Nov 28 2018 DownloadingImage size (8b80980f) is invalid! Make sure the image is generated before the download

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).

KEALTEK

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)



	Project_tz - IAR	Embe	dded Workbench IDE - Arm 8.30.1		
	File Edit View	Pro	ject J-Link Tools Window Help		
	1 🗅 🔛 🕋		Add Files	< Q > ⇆ म्म < 📮 > 🛽	a 🗈 i 🖪 🐠 🛥 i 🖸 🔹 📜 🏭 📜
	Workspace		Add Group		
	application_s - Debug	[+]	Import File List		
	Files		Add Project Connection		
	🛛 🗖 Project_tz		Edit Configurations		
		×	Remove		
		0	Create New Project		
	-⊞ infections	7	Add Existing Project		
		ф	Options Alt+F7		
	-⊞ i≡ os		Version Control System		
	user ∎	•	Make F7		
	U L 🖂 💼 Output	ň	Compile Ctrl+F7		
			Rebuild All		
	–⊞ ≣ @SR/	4	Clean		
	⊢⊞ ∎ cmsis	Ð	Batch build F8		
	-⊞ ∎ lib		C-STAT Static Analysis		
	l ⊢⊞ networ		Stop Build Ctrl=Break		
	🗕 🗄 🖬 periph	Z			
	user → ⊕ ■ user		Download and Debug Ctri+D		
			Attach to Rupping Target		
	⊢ lapp ⊡ ⊡ Dapr	G	Make & Restart Debugger Ctrl+R		
		c	Restart Debugger Ctrl+Shift+R		
			Download	Download active application	
			SED Catura	Download file	
	Overview applicat	_	Sin Setup	Erase memory	
	Build		CMSIS-Manager	,	
	Messarres		Open Device Description File		
	Performing	-1107	-Build Action	,	
			Figure 2-11 IAR dov	wnload binary on flash	
				0.	
2) Or using	g the PG tool	for	Ameba-ZII (Will not be shown	here, please check cl	napter 3 for details).
				0.5	
				US I	
				00	
				4	
					9
					U.
					(\mathcal{O})
					40
					4.0
					C
					· · · · · · · · · · · · · · · · · · ·
					*

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;
- 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).

ÄREALTEK

GCC Environment on Ubuntu/Linux 2.5

Verify Device Connections 2.5.1

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

- Ensure that the JLink debugger is connected to the target and the USB device is connected to the Ubuntu/Linux 1. 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 "Isusb" command to see list of devices as shown below:

para	allel	ls@ubunt	tu:~\$	lsi	Jsb	
Bus	881	Device	089:	ID	1366:0101	SEGGER J-Link PLUS
Bus	001	Device	005:	ID	203a:fffa	
Bus	001	Device	004:	ID	203a:fffa	
Bus	881	Device	003:	ID	203a:fffa	
Bus	001	Device	002:	ID	203a:fff9	
Bus	001	Device	001:	ID	1d6b:0002	Linux Foundation 2.0 root hub
Bus	884	Device	001:	ID	1d6b:0003	Linux Foundation 3.0 root hub
Bus	003	Device	001:	ID	1d6b:0002	Linux Foundation 2.0 root hub
Bus	002	Device	002:	ID	0403:6001	Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC
Bus	882	Device	001:	ID	1d6b:0001	Linux Foundation 1.1 root hub
para	alle	ls@ubun1	tu:~\$			
_	_					

pd t. Milal for 85360 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"
- Once the build is successful, you should be able to see the success logs as shown below. 5



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.

REALTEK

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

lash Download done, exist
A debugging session is active.
Inferior 1 [Remote target] will be killed.
uit 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.

Image Tool 3

Introduction 3.1

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.

	🙀 AmebaZII 🕯	PG Tool			23		
	Download (Generate					
	- Firmware -						
	Filename			Browse			
	SHA256						
	Address						
	1. Program f	flash memroy 🔹		Keep WIFI calib	oration		
	- Flash Optio	n	UART Setting				
	Flash IO	One IO 🔹	Port	COM1	•		
	Flash Pin	PIN_A7_A12 •	BAUD Rate	1000000	•		
P	FW Idx	1: 1st image2 🔹	Parity	NONE	•		
'0	Offset		Flow Control	OFF	•		
	Flash Erase						
	Offset		Length(Byte)				
	Mode	1.Flash Sector Erase 🔻		Erase			
	Found port:	COM1C.					
	Found port:	COM3					
	Found port:	COM47~					
			Hash \	Verify			
	Chip Erase Download						
		Figure 3-1 Ameba	ZII Image Tool L	I			
			Č,				
			6				
				5			
				· Up			
				6	3		
				(0		
					YQ.		
						2	

KAREALTEK

Environment Setup 3.2

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.



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

3.2.2

Image Download 3.3

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

- 1) Trigger Ameba-ZII chip enter UART download mode by
 - For EVB V2.0, to enter UART download mode: a. 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)

	M C	OM33	- Tera T	erm VT				
	<u>F</u> ile	<u>E</u> dit	<u>S</u> etup	C <u>o</u> ntrol	<u>W</u> indow	<u>H</u> elp		
Po	== Rt18 Chip VI ROH Ver Test Mo Dounloa	710c IoT D: 5, Ve sion: v1 de: boot d Image	Platform : r: 0 .0 _cfg1=0x0 over UART21	== (tx=16,rx=15)	baud=115200			
		- C						

Figure 3-4 Ameba-ZII UART download mode

- Choose the correct UART port (use rescan to update the port list) 2)
- "Browse" to choose the image to be downloaded (flash_xx.bin) 3)
- Choose "Mode" "1. Program flash memory" 4)
- Choose correct "Flash Pin" according to the IC part number 5)

Flash Pin	1C part number	
PIN_A7_A12	RTL8710CX/RTL8720CM	
PIN_B6_B12	RTL8720CF	

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

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