SEGGER Embedded Studio

User Guide

SUMMARY: SEGGER Embedded Studio (SES) is an Integrated Development Environment with project management tools and editor. This document provides the installation and configuration required to use SES for the Atmosic ATM2/ATM3/ATM33e/ATM34e Wireless SoC Series Software Development Kit (SDK).





SEGGER Embedded Studio User Guide July 3, 2024 4286-0139-0063

Atmosic[®]

Table of Contents

Acronyms and Abbreviations	4
1. Overview	5
1.1 Prerequisite	5
1.2 Compiler and Atmosic SDK support	6
2. ATM2/ATM3 Series	6
2.1 Set Up Tool Chain	6
2.1.1 Set up the Atmosic SDK	6
2.1.2 Install SEGGER Tools	7
2.2 Create SES Application Project	7
2.3 Compile Application	7
2.4 Program Firmware	8
2.4.1 Hardware Setup	8
2.4.2 Download Firmware and Flash NVDS	11
2.5 Debug	11
3. ATM33/e Series	12
3.1 Installation	12
3.1.1 Hardware Setup	12
3.1.2 Setup Driver and Environment Setting	12
3.2 Generate SES Project File and NVDS	15
3.3 Build and Download	15
3.3.1 Build	15
3.3.2 Download FW	15
3.4 Bootloader	16
3.5 Debugging	16
Reference Documents	19
Revision History	20

Atmosic

List of Figures

- Figure 1 Build Example
- Figure 2 Build Output with No Errors
- Figure 3 Tool Chain Directory
- Figure 4 Atmosic Interface Board
- Figure 5 Atmosic IO Adaptor Board
- Figure 6 Atmosic Interface Board with IO Adapter Board
- Figure 7 SES Debug Session Windows
- Figure 8 ATM33/e Evaluation Boards
- Figure 9 Toolchains Under SDK/tools Folder in macOS
- Figure 10 BULK Interface
- Figure 11 Uninstall BULK Interface
- Figure 12 Install J-Link Driver
- Figure 13 SES IDE GUI Compile
- Figure 14 Build Bootloader in The Pre-link Process
- Figure 15 Debug Mode
- Figure 16 Debug Mode breakpoint and break
- Figure 17 Debug Mode

List of Tables

Table 1 - ATM2/ATM3 and ATM33/e Evaluation Kits

Acronyms and Abbreviations

Acronyms	Definition
ATMx2xx	ATM2201 ATM2202 ATM2221 ATM2231 ATM2251 ATM3201 ATM3202 ATM3221 ATM3231
ATM33	ATM3330 ATM3325
ATM33e	ATM3330e
ATM33/e	ATM33/ATM33e
ATM34	ATM3405 ATM3425
ATM34e	ATM3430e
ATM34/e	ATM34/ATM34e
BD	Bluetooth Device
EVK	Evaluation Kit
IDE	Integrated Development Environment
NVDS	Non-Volatile Data Storage
OB	On Board
OTP	One Time Programmable
SDK	Software Development Kit
SES	SEGGER Embedded Studio



1. Overview

This document provides the installation and configuration required to use SES with the Atmosic SDK. SES is a cross-platform IDE, which supports different operating systems.

This document is applicable to:

- ATM2/ATM3 series
- ATM33/e series
- ATM34/e series

1.1 Prerequisite

- 1) Atmosic SDK 5.2 or later
- 2) ATM2/ATM3 Series Evaluation Kit or
- 3) ATM33/e Series Evaluation Kit or
- 4) ATM34/e Series Evaluation Kit

Table 1 lists the applicable Evaluation Kits.

EVK	SoC Package	SoC Part Number	Kit Part Number
Evaluation Kit for ATM2202	40-pin 5x5 mm QFN	ATM2202	ATMEVK-M2202-02
Evaluation Kit for ATM2221	64-pin 6x6 mm DR-QFN	ATM2221	ATMEVK-M2221-02
Evaluation Kit for ATM2251	37L WLCP	ATM2251	ATMEVK-M2251-01
Evaluation Kit for ATM3201	40-pin 5x5 mm QFN	ATM3201	ATMEVK-M3201-02
Evaluation Kit for ATM3202	40-pin 5x5 mm QFN	ATM3202	ATMEVK-M3202-02
Evaluation Kit for ATM3221	64-pin 6x6 mm DR-QFN	ATM3221	ATMEVK-M3221-02
Evaluation Kit for ATM3325	40-pin 5x5 mm QFN	ATM3325-5DCAQK	ATMEVK-3325-QK
Evaluation Kit for ATM3325 w/ Extended Storage	40-pin 5x5 mm QFN	ATM3325-5LCAQK	ATMEVK-3325-LQK
Evaluation Kit for ATM3330	56-pin 7x7 mm QFN	ATM3330-5DCAQN	ATMEVK-3330-QN
Evaluation Kit for ATM3330e	56-pin 7x7 mm QFN	ATM3330E-5DCAQN	ATMEVK-3330e-QN
Evaluation Kit for ATM3405	40-pin 5x5 mm QFN	ATM3405-2PCAQK	ATMEVK-3405-PQK

Atmosic

Evaluation Kit for ATM3425	40-pin 5x5 mm QFN	ATM3425-2PCAQK	ATMEVK-3425-PQK
Evaluation Kit for ATM3430e	56-pin 7x7 mm QFN	ATM3430E-2WCAQN	ATMEVK-3430e-WQN

Table 1 - ATM2/ATM3, ATM33/e and ATM34/e Evaluation Kits

- 5) SEGGER J-Link Plus debug probes with pin override support Note: Pin override is required for chip reset. Some J-Link models (e.g. J-Trace) do not support this feature and will generate an error during the target init process.
- 6) SEGGER Embedded Studio (version: V6.32 and older)
- 7) One of the following operating systems:
 - Windows 10, 11
 - macOS Monterey 12.5
 - Linux ubuntu 20.04.3 LTS

Refer to the <u>Reference Documents</u> section for a list of related documents, available on the Atmosic Support website. Please <u>submit a support request</u> for access credentials.

1.2 Compiler and Atmosic SDK support

• Compiler: Supports gcc toolchain only:

The SES projects work with the specific GNU gcc toolchain version (13.2.Rel1) only.

• Atmosic SDK functionality:

The SES projects support the building of the bootloader, example firmware, and NVDS data. For the other SDK makefile functions (such as program OTP, pull firmware/NVDS/OTP, show NVDS/OTP data, etc.), please use the MSYS2 console in the SDK.

For ATM33/e and ATM34/e, changing the J-Link OB driver for openocd use might be required if accessing devices with openocd.

2. ATM2/ATM3 Series

2.1 Set Up Tool Chain



2.1.1 Set up the Atmosic SDK

Follow the Atmosic SDK Quick Start Guide to set up the SDK. Enter the make BOARD=<chip> run_all command in the command line to ensure that GNU/GCC can build an Atmosic SDK example, then download the example image into the Atmosic Evaluation Board.

2.1.2 Install SEGGER Tools

Plug the SEGGER J-Link debug probes into your laptop or PC and install the most recent release of the SES. Please make a note of the SES installation destination path in the setup wizard. This path is required to put the J-Link flash loader file. Please also make sure the J-Link driver is installed correctly.

2.2 Create SES Application Project

Please refer to the IDE Auxiliary Flash Programming Tool User Guide to generate the project file with the IDE auxiliary tool. You will see a new folder named ses. This is the SES project file for this example.

2.3 Compile Application

Compile the application from the SES project

Open your project in SES
 SES projects are located in the ses subfolder of the example, such as

SDK_root\platform\atm2\ATM22xx-x1x\examples\BLE_adv\ses\example .emProject

2) Select Build>Build example

Build	Debug	Target	Tools	Window	Help	
🏭 Bui	ild exampl	e		F7		Ī
Rel	ouild exam	nple		Alt+F7		Γ
Cle	ean examp	le				L

Figure 1 - Build Example

Make sure that there are no build errors. The output should look similar to Figure 2:

Atmosic

Output		🖸 #4 🗙
Show: Transcript 🔹 🐐 Tasks	•	*
 Building 'example' from solution 'example' from solution 'example' be completed 24 Notes 	24 targets in 5. 4 targets/s	
☑ Post-build command ▹ Completed		
Build complete Completed		2 projects in 6.2 0 projects/s
FLASH1	RAM1	
16.8 KB of 480.0 KB used	3.5% 2.0 KB of 16.0 KB used	12.8%

Figure 2 - Build Output with No Errors

The GNU/GCC toolchain will use Atmosic SDK's path for SES by default. If users want to modify the toolchain path, users need to modify it in SES ID, see <u>Figure 3</u>. To get to this UI, select Project -> Options.

Solution 'example' Options									
↑↓ C ATMx	Search Options								
- Code	Option	Value							
Assembler									
Build	✓ ■ Build								
Code Analyzer	Output Directory	Output/\$(Configuration)/Exe							
Compiler	Intermediate Directory	Output/\$(Configuration)/Obi/\$(ProjectName)							
External Build	Executable File Name	\$(OutDir)/\$(ProjectName)\$(EXE)							
File	 Object File Name 	\$(IntDir)/\$(InputName)\$(OBJ)							
GCC Compiler	Project Macros								
General	Batch Build Configurations								
Linker	Tool Chain Directory	////tools/gcc-arm-none-eabi-8-2019-q3-update/bin/							
Preprocessor	Property Groups File	None							
Source Code	 Build Options Generic File Name 	None							
User Build Step									

Figure 3 - Tool Chain Directory

Users also need to modify the post-build script named postbuild.js in the ses folder.

2.4 Program Firmware (IDE Auxillary Flash Programming Tool)

2.4.1 Hardware Setup

Connect J-link debug probes through the Atmosic Interface Board (part of the ATM2/ATM3 evaluation kit) and Atmosic IO Adapter Board (ATMIAB).

J-Link debug probes will not provide power to the interface board. Please plug the USB cable into USB1 of the EVK to supply the power.

- 1) Connect the USB cable to USB1 of the EVK
- 2) Remove jumper for P32 on J3 header (J3 is Boot Mode connector, see item 5 in Figure 4)

Atmosic[®]

- 3) Remove jumper for SWD (see items 3 and 4 in Figure 4)
- 4) Connect Jlink adapter board (see item 1 in Figure 4)





Figure 4 - Atmosic Interface Board V3.x

The Atmosic I/O Adapter Board is shown in Figure 5.



Figure 5 - Atmosic I/O Adapter Board



Figure 6 shows the connection for the Atmosic interface board, I/O adapter board, and J-Link probes.



SEGGER Embedded Studio User Guide July 3, 2024 4286-0139-0063



Figure 6 - Atmosic Interface Board with I/O Adapter Board, J-Link Probes

Refer to the Install J-Link section of the **IDE Auxiliary Tool User Guide** for more information.

2.4.2 Download Firmware and Flash NVDS

After building successfully, please download FW and NVDS with the IDE Auxiliary Flash Programming tool. For more details, please refer to the IDE Auxiliary Flash Programming tool User Guide.

2.5 Debug

You must set up a J-Link debugging session. SES has an integrated debugger that can be to step through your application. Select Debug>Go

File Edit View Search Navigate	Project Build De	ebug Target Tools Window	Help			
Disassembly	n 💿 🗙	atm_adv.c gapm_task.h BLE_ad	tv.c	≥ ~ X	Globals	# 🖬 🗊 📾 🔛 🗙
main + 0xa 🕨 🖬 🗧	- ១០គេឆ្ -	int main()	• • • • • • • • •	ାଣା ଏ 💀 🤞 🖉	x ₂ x ₈ x ₁₃ x ₁₅ x ₁₆ x .	🗎 🗈 👻
10003300 .wo	ord 0x100033(^			^	Expression	Value
100036AE .wo	ord 0x100036/	 atm_asm_init_ 	table(S_TBL_IDX, s_tb	1, ARRAY_LEI	app_version	"\$Revision: App Version 0.0.0.9 \$"
000066B1 .wo	ord 0x0000661	 atm_asm_set_s 	<pre>state_op(S_TBL_IDX, S_</pre>	INIT, OP_EN	atmosic sdk ver	"\$Revision: SDK Version 4.1.0 \$"
100001FD .wo	ord 0x100001	 atm_asm_move(S_TBL_IDX, OP_MODULE_	INIT);	default adv create param	<array></array>
10001A4D .wo	ord 0x10001A	460			<pre>> default adv data0</pre>	"\f\tAtmosic-AD"
10001A71 .wo	ord 0x10001A	// Don't use	the app in the ROM		default adv start param	<array></array>
100036EE .wo	ord 0x1000361	· return RV_DON	iE;		default bt init param	<struct></struct>
BLE_adv.c 47	'0	}			default dev conf	<struct></struct>
1		/*			default scan data0	"\t\377"
* Verter		* user main()			<pre>> default set adv data</pre>	<array></array>
 vector replacemen 	τ	* Driver initial	ization and ren vec a	dditions	HFH Stopper	0x61934b1c
	ST/uson ann	*/	ización and rep_vec a	durcrons		
RV_APPM_INIT_ADD_LA	(si(user_appr	int main(void)				
10000570 B310 pds	$n_1^{1,4}, 1^{1,7}$	470 {				
10000572 4505 Idr	r0 = 0x200	/*				
10000576 4806 ldr	r3, =0x000	* Vector rep	lacement			
10000578 4798 blx	r3	*/				
BLE adv.c 47	5	 RV_APPM_INIT_ 	ADD_LAST(user_appm_in	it);		
ATM LOG(D, "user ma	in() done")					
•1000057A 4906 ldr	r1, =0x1000	• 476 ATM_LOG(D, "u	<pre>user_main() done");</pre>			
1000057C 4806 ldr	r0, =0x100	 return 0; 				
1000057E 4B07 ldr	r3, =0x0000	}				
10000580 4798 blx	r3					
BLE_adv.c 47	7					
return 0:	~			~		
<	>	<		>	<	>
References	<u>∧</u> ×				Registers 1	🖬 🗙
Search within results		Output a a a		🗠 🗡		B D Count Desisters
Dealers		Show: Target 🔹 🍾	Tasks 🔹 🖉 🕨 🐢	· ·	Groups 🗢 🗢 🖃 🖓 🖃	Search Registers
керіасе		Preparing target for dov	F Call Address	File St	Name Value	^
Replace All < Prev Next >	Replace	 Completed 	0x1000057A	BLE_adv.c:476 0x		
		Erasing Chip	0xFFFFFFE	Ox	CPU - Current Context	R (A)
		Completed			🖬 r0 🛛 Oxaaaaaaaa	
		Downloading 'example. 20.2 K	(B in 1.3s		■r1 0x20000000	
		Download successful 15.4 K	(B/s		©r2 0x20014248	
		Downloading 'flash nvc 0.1 KE	3 in 0.2s		ur3 0x1000001	
		Download successful 0.4 KE	B/s		ur4 0x0000001	
No results					Ur5 0x0000013	
					OXTITITIT Ovffffffff	
					n10	
					n11 0x20004000	
			<	>	-12 0x0000000	~
					CortexM0 d	on J-Link 🔍 2 INS (No editor) 17:25 🖸

Figure 7 - SES Debug Session Windows

3. ATM33/e and ATM34/e Series

3.1 Installation

3.1.1 Hardware Setup

Connect the Micro-B USB cable to the ATM33/e Evaluation Board. See Figure 8.



Figure 8 - ATM33/e Evaluation Board

3.1.2 Setup Driver and Environment Setting

• Linux/macOS

1) Setup IDE environment

Replace [*ATM_SDK*] name with the actual installation folder name of Atmosic SDK

a) Install GNU Arm Embedded Toolchain, Version 13.2.rel1 to [*ATM_SDK*]\tools\



	< > arm-gnu-toolchain-13.2.Rel1-da	arwin-arm64 ፡፡፡ 등 ⊨		··· ~	Q Search	
Favorites	atmosic_sdk_5.5.1	commit_id	🗾 arm-gnu-toonone-eabi		■ .fseventsd →	
🔊 AirDrop	atmosic_sdk_6.0.0_alpha →	contrib	atm_hl_tools	->	13.2.Rel1-dai-manifest.txt	
② Recents	atmosic_sdk0_RC2.tar.xz	icenses	atm_isp		🗧 arm-none-eabi 🛛 🚽	
	atmosic_sdk0_RC3.tar.xz	📄 platform	🚞 atm_isp_python	->	🚞 bin 🛛 🔅	
	🚞 atmosic_sdk_600_Alpha4 🔅	README	🚞 iar	->	include →	
Documents	📄 atmosic_sdk_600_RC3 🛛 🚿	i tools	🚞 ide_prj_gen	->	🔲 lib 🛛 🖓	
Ownloads	atmosic_sdk_SDK600_RC2 >	version.h	🚞 ide_tool	->	ibexec >	
Locations.	FreeRTOSv202212.00		🚞 jlink	->	💼 share 🛛 🔅	
	howto.rtf		🚞 layout_info	->		
🖵 user-Mac mini	openocd		🚞 mcuboot_utils	->		
📮 atmosic-NAS 🔶	rf4ce_sdk_6.0.0_RC2.tar.xz		🚞 openocd	->		
® Network	🚞 tar 🛛 🔅		📄 pinmux	->		
	🚞 toolchain 🛛 🔅 🔅		README			
Tags			README.ISP			
● 紅色			README.macOS			
● 橙色		1	 README.Windows	11		

Please refer to README/README.macOS for the related version.

Figure 9 - Install Toolchain to SDK/tools Folder in macOS

2) Install SEGGER Embedded Studio

• Windows

- 1) Confirm J-Link driver is showing in Windows Device Manager. If the SDK RDI driver is installed, it would display as "BULK interface" and not the J-Link driver. Please remove the BULK driver with the below steps.
 - a) Find the BULK interface in the Device Manager



Figure 10 - BULK Interface

b) Uninstall the BULK interface device. Press the right mouse button and select uninstall.

Atmosic



Figure 11 - Uninstall BULK Interface

c) Click the Scan for hardware changes icon and the J-Link driver will be installed automatically

ᡖ Device Manager	_		\times					
File Action View Help								
🖗 Generic SuperSpeed USB Hub			^					
🏺 Generic SuperSpeed USB Hub								
🏺 Generic USB Hub								
🏺 Generic USB Hub								
🏺 Generic USB Hub								
🏺 Generic USB Hub								
🏺 Generic USB Hub			- 10					
Intel(R) USB 3.1 eXtensible Host Controller - 1.10 (Microsoft)	Intel(R) USB 3.1 eXtensible Host Controller - 1.10 (Microsoft)							
🏺 J-Link driver								
USB Composite Device								
USB Composite Device			~					

Figure 12 - Install J-Link Driver

2) Setup IDE environment

Windows Start -> Atmosic SDK, run Setup IDE Environment

- 3) Reboot the PC for system environment variable update.
- 4) Install SEGGER Embedded Studio



3.2 Generate SES Project File and NVDS

Please refer to the IDE Auxiliary Flash Programming tool User Guide to generate the project file with the IDE auxiliary tool. You will see a new folder named ses. This is the SES project file for this example.

NVDS bin file would be generated to the ses folder as generating SES project file. The NVDS bin file would not be updated as building examples. To change the BD address, please add USER_BD_ADDR:="xx xx xx xx xx xx xx" to generate the SES project.

3.3 Build and Download

3.3.1 Build

- 1) Example project (under ses folder)
- 2) Edit/Compile firmware in SES IDE, see Figure 13



Figure 13 - SES IDE GUI - Compile

3.3.2 Download FW

After building successfully, please download FW and NVDS with the IDE Auxiliary Flash Programming tool. For more details, please refer to the IDE Auxiliary Flash Programming Tool User Guide.

3.4 Bootloader

The pre-link process would generate the bootloader binary file as Figure 14.

```
•••
2> Pre-link command
2> ATM33 SES Build type:undefined
2> Prebuild in Windows system
2> rm -f tag_data/18-PROG_DELAY/default.tds
2> rm -f tag_data/*/*.bin
2> rm -f flash_nvds_resize.bin
2> rm -f flash_nvds.ihex
2> rm -f bootloader.{asm,bin,elf,hex,ihex,map}
2> rm -f *.gcno *.gcda
2> rm -f *.gcov
2> rm -f merged_bootloader.{asm,bin,elf,hex,ihex,map}
2> rm -f signed_bootloader.{asm,bin,elf,hex,ihex,map,ubin}
2>/c/Projects/atmosic_sdk/tools/gcc-arm-none-eabi-10.3-2021.07/bin/arm-none-eabi-gcc --
specs=nosys.specs --specs=nano.specs -Wl,--fatal-warnings,--warn-common -mthumb -march=armv8-
m.main+fp+dsp -mcpu=cortex-m33 -mfloat-abi=hard -mfp16-format=ieee -Wl,--gc-sections -Wl,-z,max-page-
size=2048 -L/c/Projects/atmosic_sdk/platform/atm33/ATM33xx-5/user -Wl,-
defsym=RRAM_PROGRAM_SIZE=0x7F800 -Wl,--defsym=__s_user_rram_size=0x5000 -Wl,--defsym=USER_SIZE=0 -Wl,-
-defsym=NVDS_START=583680 -Wl,--defsym=NVDS_SIZE=0x1000 -Xlinker -Map=bootloader.map -mcmse -Wl,--cmse-
implib -Wl,--out-implib=Secure_Functions.o -Tsecure.ld common.o bootloader.o retarget_uart.o
exception_handler_armv8mml.o sec_hw_cfg.o spi.o sec_jrnl.o arm_traceback.o sec_service.o -o
bootloader.elf
2>/c/Projects/atmosic_sdk/tools/gcc-arm-none-eabi-10.3-2021.07/bin/arm-none-eabi-objcopy -0 binary
bootloader.elf bootloader.bin
2>/c/Projects/atmosic_sdk/tools/gcc-arm-none-eabi-10.3-2021.07/bin/arm-none-eabi-objdump -d
bootloader.elf >bootloader.asm
```

Figure 14 - Build Bootloader in the Pre-link Process

3.5 Debugging

Please don't use the Debug->Go button for debugging. Pressing Debug ->Go in SES IDE, would trigger the program FW process. The FW would be updated unexpectedly. Please use Attach Debugger as per the below steps.

1) Select Target -> Attach Debugger

Atmosic[®]

File Edit View Search Na	vigate Pro	ject Build	Debug	Tar	get Tools V	Nindow H	elp	
Project Explorer			startup_	A 🏷	Connect J-Link		Ctrl+T, C	
ATMx3 Debug 🔻 🗆 🧰 🕈	0	1 V V	$\leftarrow \rightarrow$	×.	Disconnect		Ctrl+T, D	
Project Items	Code	Data+RO		1Se	Reconnect		Ctrl+T, E	
Solution 'BLE adv'	code	Dutarrito	i	I []≣	Attach Debugg	ger	Ctrl+T, H	
Project 'BLE_adv'	379.2K	13.2K	1	-	Reset		Ctrl+T. S	
in CMSIS				T≣	Download BLE	adv	Ctrl+T I	
🗀 app					Verify BLE adv		Ctrl+T V	
🔺 🚔 driver 35 files	[30.5K]	[5.4K]	#	•	Verify DEE_dov		Curri, v	
▷ 🗟 atm_ble.c	152	91	120		Erase All		Ctrl+T, K	
atm_bp_clock.c	1.2K	150	#	£	Upload Range.			
▷ 🗟 atm_mpu.c	396				Download File		•	
▷ 🗟 atm_plf_drv.c	124	44			Verify File		•	
▷ 🗟 atm_pm.c	724	166	125 #	_	Verify The			
▷ 🗟 ble_irq.c	152	31	#	, Þ≚	Start Cycle Cou	unter		
b la brwnout.c	354	134		垦	Pause Cycle Co	ounter		
b label{eq:states} brwnout_core.c	120		#	2	Zero Cycle Cou	unter	Ctrl+T, Z	
▷ 🗟 fake_flash.c	372	64	130	r, P	Target Connect	tion Propertie	26	set_loc
▷ 🗟 gadc.c	1.6K	516	150		EBUG TRACE	"Shub disa	abled");	
▷ 🗟 heap.c			#	endi	lf `		7 -	
▷ 🗟 hw_cfg.c	218	102						
hw_cfg_core.c	2.0K	88	#	ifde	+ CFG_EXT_F	LASH	().	
🖻 둳 interrupt.c	568	269	#	e tendi	f	sn_wakeup	60	
Et lass	100	10		CHOI				

BLE_adv - SEGGER Embedded Studio for ARM V5.42b (32-bit) - Non-Commercial License



2) Set a breakpoint and press break and continue

🏈 BL	E_adv - 1	SEGGER	Embedd	ed Studio fo	r ARM V5.4	42b (32-	bit) - Non-	Commer	cial Licens	se (Runn	ning)	
File	Edit	View	Search	Navigate	Project	Build	Debug	Target	Tools	Windo	ow Help	
Disas	sembly									(1)	< startu	p_ARMv8MBL.c user_init.c
Locat	ion								9∃ [∃ ç3	t d⊒ 🗸	-	
											120 125	<pre>#ifdef CFG_NO_SPE DEBUG_TRACE("No SPE"); sec_lockdown(); #if VERIFY_ATMWSTK if (Iverify_atmwstk()) { ASSERT_ERR(false); } #endif #endif // CFG_NO_SPE #ifdef LOCKDOWN_SHUB_DISABLE _UNUSED bool sec_s = sec_device_set_lockout(SEC_DEV_LOCKOUT_SHUB_ ASSERT_ERR(sec_s); DEBUG_TRACE("Shub disabled"); #endif #ifdef CFG_EXT_FLASH external flash wakeup();</pre>
												#endif
											• • 140	<pre>DEBUG_TRACE("SDK Version: %s", SDK_VERSION); DEBUG_TRACE("APP Version: %s", APP_VERSION);</pre>
											•	WRPR_CTRL_SET(CMSDK_PSEQ, WRPR_CTRL_CLK_ENABLE); { CMSDK_PSEQ->CTRL0 &= ~(PSEQ_CTRL0_PINPU_LATCH_OPEN_MASK PSEQ_CTRL0_GPIO_LATCH_OPEN_MASK PSEQ_CTRL0_SPI_LATCH_OPEN_MASK

Figure 16 - Debug Mode - breakpoint and break

Atmosic^{**}

🐼 R

File Edit	View Search Navigate Project Build Debug Target Tools Wind	dow Help
Disassembly		★ startup_ARMv8MBL.c user_init.c
main + 0xa	ão, ≣0, Ξ° → ■ <	✓ ♦ int main()
0001A02E 0001A030	2A01 cmp r2, #1 D908 bls 0x0001A04A — user_init.c - 97 return NVDS_LENGTH_OUT_OF_RANGE; } // Conv_data to output buffer	<pre>* #ifdef CFG_NO_SPE DEBUG_TRACE("No SPE"); 120 sec_lockdown(); #if VERIFY_ATMWSTK if (!verify_atmwstk()) { ASSERT ERR(false);</pre>
0001A032 0001A034	wmccpy(buf, user_params[1].data, data_len); 2232 movs r2, #0x32 8022 strh r2, [r4] user_init.c 102	<pre>} #endif #endif // CFG_NO_SPE #ified () CCCDCIN CUUD DICADUS</pre>
0001A036 0001A038 0001A03A	// Return bytes copied *lengthPtr = data_len; 2202 movs r2, #2 2000 movs r0, #0 700A strb r2, [r1]	<pre>#ITGET LOCKDOWN_SHOD_DISAGLE UNUSED bool sec_s = sec_device_set_lockout(SEC_DEV_LOCKOUT_SHUB_[ASSERT_ERR(sec_s); DEBUG_TRACE("Shub disabled"); #endif</pre>
0001A03C 0001A040	- user_init.c - 105 return NVDS_OK; F85D4804 pop.w {r4} 4770 bx lr - user_init.c - 106	<pre>#ifdef CFG_EXT_FLASH external_flash_wakeup(); #endif</pre>
0001A042 0001A046 0001A04A 0001A04A 0001A04E	<pre>} // Did not find it - fall back to NVDS USER_PARAM_DBG("get param %#x", param_id); return nvds_get8(param_id, lengthPtr, buf); F8D4804 pop.w {r4} F04EBDA8 b.w 0x00068B9A <nvds_get8> 2004 movs r0, #4 E7F6 b 0x0001A03C 0000 movs r0, r0 user_init.c - 112</nvds_get8></pre>	<pre> • 138 DEBUG_TRACE("SDK Version: %s", SDK_VERSION); • DEBUG_TRACE("APP Version: %s", APP_VERSION); 140 . WRPR_CTRL_SET(CMSDK_PSEQ, WRPR_CTRL_CLK_ENABLE); { CMSDK_PSEQ->CTRL0 &= ~(PSEQ_CTRL0_PINPU_LATCH_OPEN_MASK PSEQ_CTRL0_GPIO_LATCH_OPEN_MASK PSEQ_CTRL0_SPI_LATCH_OPEN_MASK PSEQ_CTRL0_PINSEL_LATCH_OPEN_MASK); } HDDP_CTRL_SET(CMEDK_BSEQ_URPR_CTRL_CLK_DTSARLE); </pre>
0001A050 0001A052 0001A054 0001A056	<pre>int main(void) { // Initialize MPU common_mpu_cfg(); B57F push {r0-r6, lr} 2501 movs r5, #1 2400 movs r4, #0 F000F8B1 bl 0x001A1BC <common_mpu_cfg> user init.c - 134 </common_mpu_cfg></pre>	<pre>www.crk_str(wbb_rseq, wkrk_crkt_clk_bisAble); 150 #ifdef CFG_EXT_FLASH external_flash_init(); cvt_flash_init(); cvt</pre>
	<pre>#ifdef CFG_EXT_FLASH external_flash_wakeup(); #endif DEBUG_TRACE("SDK Version: %s", SDK_VERSION);</pre>	Checking project status Project up to date
0001A05A 0001A05C 0001A05E	4923 ldr r1, =0x0006FBB3 4823 ldr r0, =0x0006FBB9 F002FBF1 bl 0x0001C844 <debug_trace> </debug_trace>	
 0001A062 0001A064 0001A066 	DEBUG_TRACE("APP Version: %s", APP_VERSION); 4923 ldr rl, =0x0006FBC9 4823 ldr r0, =0x0006FBD1 F002FBED bl 0x0001C844 <debug_trace> user_init.c 140</debug_trace>	
0001A06A 0001A06C 0001A06E	WRPR_CTRL_SET(CMSDK_PSEQ, WRPR_CTRL_CLK_ENABLE); 4A23 ldr r2, =0x40153000 4923 ldr r1, =0x40158000 6215 str r5, [r2, #32] 	

Figure 17 - Debug Mode

1



Reference Documents

Title	Document Number
ATM33/e Series Evaluation Kit User Guide	ATM33_e-UGEVK
ATM34/e Series Evaluation Kit User Guide	6441-0063-0011
Atmosic SDK User Guide	6844-xxxx-xxxx
EVK User's Guide for ATMx221	ATMx221-UG
EVK User's Guide for ATMx301/ATMx202	ATMx201-UG
IAR Workbench User Guide	4247-xxxx-xxxx
IDE Auxiliary Flash Programming Tool User Guide	4381-xxxx-xxxx
Keil MDK Configuration Guide	4252-xxxx-xxxx

Atmosic[®]

Revision History

Date	Version	Description
July 3, 2023	0.63	Updated for SDK 6.0.0.
July 19, 2023	0.62	Updated <u>Overview</u> , <u>Table 1 - ATM2/ATM3 and</u> <u>ATM33/e Evaluation Kits</u> , <u>Create SES application</u> <u>project</u> , <u>Download Firmware and Flash NVDS</u> , <u>Generate SES Project File and NVDS</u> , <u>Build and</u> <u>Download</u> , <u>Bootloader</u> , <u>Debugging sections</u> , <u>Figure 4 - Atmosic Interface Board V3.x</u> , <u>Figure 5 -</u> <u>Atmosic I/O Adapter Board</u> , <u>Figure 6 - Atmosic</u> <u>Interface Board with I/O Adapter Board</u> , J-Link <u>Probe</u> , <u>Figure 8 - ATM33/e Evaluation Board</u> , <u>Figure 14 - Build Bootloader in the Pre-link</u> <u>Process</u> . Added <u>Acronyms and Abbreviations</u> , <u>Hardware Setup</u> , <u>Driver and Environment Setting</u> <u>Setup</u> , <u>Reference Documents</u> sections.
June 24, 2022	0.61	Updated Overview item 4),
May 13, 2022	0.60	Initial version created for SDK 5.1 which supports ATM2/ATM3 and ATM33.

ATMOSIC TECHNOLOGIES – DISCLAIMER

This product document is intended to be a general informational aid and not a substitute for any literature or labeling accompanying your purchase of the Atmosic product. Atmosic reserves the right to amend its product literature at any time without notice and for any reason, including to improve product design or function. While Atmosic strives to make its documents accurate and current, Atmosic makes no warranty or representation that the information contained in this document is completely accurate, and Atmosic hereby disclaims (i) any and all liability for any errors or inaccuracies contained in any document or in any other product literature and any damages or lost profits resulting therefrom; (ii) any and all liability and responsibility for any action you take or fail to take based on the information contained in this document; and (iii) any and all implied warranties which may attach to this document, including warranties of fitness for particular purpose, non-infringement and merchantability. Consequently, you assume all risk in your use of this document, the Atmosic product, and in any action you take or fail to take based upon the information in this document. Any statements in this document in regard to the suitability of an Atmosic product for certain types of applications are based on Atmosic's general knowledge of typical requirements in generic applications and are not binding statements about the suitability of Atmosic products for any particular application. It is your responsibility as the customer to validate that a particular Atmosic product is suitable for use in a particular application. All content in this document is proprietary, copyrighted, and owned or licensed by Atmosic, and any unauthorized use of content or trademarks contained herein is strictly prohibited.

Copyright ©2022-2024 by Atmosic Technologies. All rights reserved. Atmosic logo is a registered trademark of Atmosic Technologies Inc. All other trademarks are the properties of their respective holders.

