IAR Workbench

User Guide

SUMMARY: This document provides the installation and configuration required to use IAR Workbench IDE for the Atmosic Wireless SoC Series Software Development Kit (SDK) on the Windows platform.



Table of Contents

User Guide	1
Table of Contents	2
List of Figures	3
Acronyms and Abbreviations	3
1. Purpose	5
1.1 Prerequisites	5
2. ATM2/ATM3 Platform	5
2.1 Installation	5
2.2 Generate IAR Workspace	7
2.3 Build and Program	7
2.4 Set Flash Loader	9
2.5 Program Flash with NVDS	9
3. ATM33/e and ATM34/e Platform	12
3.1 Installation	12
3.2 Generate IAR Workspace	13
3.3 Build and Program Bootloader	13
3.4 Build Example and Program (without USE_MCUBOOT)	14
3.5 Program Image with NVDS (without USE_MCUBOOT)	17
Reference Documents	19
Revision History	20

List of Figures

- Figure 1 I-jet Installation Successful
- Figure 2 Interface Board V3.x/V4.x and I-jet Connections
- Figure 3 IAR Workbench GUI
- Figure 4 Debug Session
- Figure 5 Set Flash Loader
- Figure 6 Add flash_nvds.bin
- Figure 7 ATM33/e Evaluation Board
- Figure 8 IAR Workbench GUI
- Figure 9 Download and Debug
- Figure 10 Runtime Debug
- Figure 11 BLE_adv Example Log
- Figure 12 Program without NVDS (without USE_MCUBOOT)

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

EVB	Evaluation Board
EVK	Evaluation Kit
SDK	Software Development Kit
SoC	System-on-Chip



1. Purpose

This document provides the installation and configuration required to use the IAR Embedded Workbench for the Atmosic SDK on the Windows platform.

1.1 Prerequisites

- 1) Atmosic SDK v5.5 or later
- 2) Atmosic Evaluation Board for ATM3330e/ATM3330/ATM3325 devices or
- 3) Atmosic Evaluation Board for ATM3430e/ATM3425/ATM3405 devices or
- 4) Atmosic Evaluation Board for ATMx2xx devices and Interface Board v3.x
- 5) IAR Embedded Workbench v9.32.2.57414
 - a) Download IAR Workbench
- 6) I-jet Probe/debugger (for ATM2/ATM3 platform only)

2. ATM2/ATM3 Platform

2.1 Installation

- 1) I-jet Probe/debugger
 - a) Install IAR Embedded Workbench
 - b) Insert the I-jet device
 - c) After successful installation, IAR Systems USB Probes should display under Device Manager as shown in Figure 1



Figure 1 - I-jet Installation Successful

d) Pin Connection

- i) Atmosic Interface Board V3.x, V4.x or V5.x, see Figure 2
- ii) Remove JP1, JP2 and J3 jumpers on the interface board
- iii) Use the attached MIPI-10 JTAG cable to connect to the J-Link socket of the interface board
- e) Connect the interface board to the ATM2/ATM3 EVB (refer to the ATM2/ATM3 Evaluation Kit User Guide and Interface Board User Guide for more information)



Figure 2 - Interface Board V3.x/V4.x/V5.x and I-jet Connections

2) Flash Loader

Copy all files from <SDK installation folder>\tools\iar to <IAR installation folder>\arm\config\flashloader\Atmosic

See the Generate IAR Workspace section to generate the IAR Workspace.



2.2 Generate IAR Workspace

Please refer to the section on **How to Generate IDE Project** from the **IDE Auxiliary Flash Programming Tool User Guide** to generate IAR Workspace.

After generating the workspace successfully, it can be found in the following folder: <Atmosic SDK>\platform\atm2\ATM2xxx-xxx\example\xxx\iar <Atmosic SDK>\platform\atm3\ATM3xxx-xxx\example\xxx\iar

Open workspace file in IAR folder, i.e., example\xxx\iar\<xxx>.eww

Note: IAR for ATM_mcuboot example is not supported. (Only GCC is supported for this example). For more details please refer to the makefile of ATM_mcuboot.

2.3 Build and Program

- 1) Program Flash
 - a) Without NVDS for each example is the default configuration
 - b) With NVDS data for each example (refer to <u>Program Flash with NVDS</u> section)
- 2) Edit/Compile/Program/Debug Code at IAR IDE

BLE_adv - IAR Embedded Workbench IDE - Arm 8.50.1			Make
File Edit View Project I-jet Tools Window Help			Wate
1 0 🖬 🖨 📇 X 🛍 0 1 5 C 1		(Q	> 毎 曜 く 🔍 > 🕢 🖻 📘 🌒 🛊 🚺 🕨 🕌 🌉
Workspace	•	- a >	
ATMx2_Debug		v	Program and
Files	۰	•	Debug
🖃 🌒 BLE_adv - ATMx2_Debug	~		
i app			
+ 🖬 driver			
🕂 🖬 lib			
🖬 profiles			
-+ isrc			
-+ 🖬 user			
-+ 🖬 Output			

Figure 3 - IAR Workbench GUI

 Run time debugger, <u>Figure 4</u> shows the layout of the debugger session as an example

	2		4						
ace	₩ Ø :	X BLE_advac x		Disasembly			-		
L Debug		maing	+0	Goto		 Memory 	~ 🖪		
0.8	• •			Disass	embly				
BLE_adv - ATHx2_Debug	~	#ifdef CFG_DYN_ADV			0x550a:	0x2080	MOVS	R0, #128	
app 📕		<pre>tid_update_adv = sw_timer_alloc(update_adv_timer, NULL);</pre>			0x550c:	0x2301	MOVS	R3, #1	
E 📕 driver		- Hendit // GPG_DTM_ADV			0x550e:	0x0180	LSLS	R0, R0, #6	
🗄 📫 lib		adv_lock_hiber = atm_pm_alloc(PM_LOCK_HIBERNATE);			0x5510:	0x7223 0xf7fb_0xf8c3	BL	N3, [N4, #UX8]	
E lib_c		atm_pm_lock(adv_lock_hiber);			0x5516:	0x6823	LDR	R3, [R4]	
profiles		atm_pm_set_hib_restart_time(restart_time_csec);			0x5518:	0x1da0	ADDS	R0, R4, #6	
src		stm asm init table(S_TRL_IDX_s_tblARRAY_LEN(s_tbl));			0x551a: 0x551c:	0x4a04 0x685c	LDR.N LDR	R2, [PG, #0 R4, [R3, #0x4]	
H BLE_Adv.c		atm asm set state op(S TBL IDX, S INIT, OP END);			0x551e:	0x2102	MOVS	R1, 42	
user 🖉		atm_asm_move(S_TBL_IDX, OP_MODULE_INIT):			0x5520:	0x002b	MOVS	R3, R5	
🗉 🖬 Output		1/ Dealth was Alexander In Alex DOM			0x55221	0x47a0 0xe7ee	BLX B.N	0x5504	
		return RV DONE:			0x5526:	0x46c0	VOM	R8, R8	
					0x5528:	0x0bd8	LSRS	R0, R3, #15	
					0x552a: 0x552c+	0x2000	STRB	R5, [R4, R2]	
					0x552e:	0x0000	MOVS	RO, RO	
		* User_Nain() * Driver initialization and ren vec additions			0x5530:	0x4809	LDR.N	R0, [PC, 40	
		* 0/1101 - 11/2/2/01 und 100_100 udd12/010			0x5532:	0x6801	LOR	R1, [R0] p1 p1 #13	
		int main(void)			0x5536:	0xd300	BOC.N	0x553a	
					0x5538:	0£1dx0	WFI		
		* Vactor ranlacement			0x553a:	0x4808	LDR.N	R0, [PC, #0	
		- */			0x553e:	0x4a09	LDR.N	R2, [PC, #0	
		RV_APPM_INIT_ADD_LAST (user_appm_init);			0x5540:	0x4b09	LDR.N	R3, [PC, #0	
		ATM LOG (D. "unor main(), dono") :			0x5542: 0x5544	0x1a9b 0xc00o	SUBS	R3, R3, R2 p01 (P1=P3)	
		return 0:			0x5546:	0x3814	SUBS	R0, R0, #20	
					0x5548:	0x2180	MOVS	R1, #128	
					0x554a: 0x554c:	0x0609	STR	R1, R1, #24 R1, [R0]	
dv		¢	3 4		0x554e:	0x2100	MOVS	R1, #0	
2									
a									
e Dec 07, 2021 17:53:	1: Probe: V	Versions: JTAG-1.85 SWO-1.40 A2D-1.73 Stream-1.50 SigCom-2.44							
e Dec 07, 2021 17:53:	01: Emulatio	on layer version 4.61							
e Dec 07, 2021 17:53: e Dec 07, 2021 17:53:	<pre>/1: Notifica)1: Connecte</pre>	d DAP v1 on SWD. Detected DP TD=0xbb11477.							
e Dec 07, 2021 17:53:	01: Connecti	ing to TAP#0 DAP AHB-AP port 0 (IDR=0x4770021).							
e Dec 07, 2021 17:53:	01: Recogniz	ed CPUID-0x410cc200 Cortex-M0 r0p0 arch ARMv6-M							
e Dec 07, 2021 17:53: e Dec 07, 2021 17:53:	1: LosLevel	Reset(system, delay 200)							
e Dec 07, 2021 17:53:	01: CPU stat	us - IN RESET							
Dec 07, 2021 17:53: Dec 07, 2021 17:53:	02: Download	ing atmosic flash loader Alberta (Alberta) (Al	and and the second of a factor of the second						
ue Dec 07, 2021 17:53: ue Dec 07, 2021 17:53:	02: Loaded o 02: Target r	mougee: C:\Frogram Files (xee)\lue systems(Embedded Workbench e.4\arm(conlig(liash. reset	oader (winosie (achk_riash.ou)						
ue Dec 07, 2021 17:53:	03: Unloaded	i macro file: C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.4\arm\config\	lashloader\Atmosic\atmx_fla	sh.mac					
Tue Dec 07, 2021 17:53:	03: Download 03: 25037 kg	<pre>ied D:\sdk_repo\atmosic_sdk_precheck_iar\platform\atm2\ATM22xx-x1x\examples\BLE_adv' then deamlorded into ITING (0.16 Worker/core)</pre>	iarauto\Debug\Exe\BLE_adv.e.	lf to flas	h memory				
Tue Dec 07, 2021 17:53:	04: Loaded d	<pre>lebugee: D:\sdk repo\atmosic sdk precheck iar\platform\atm2\ATM22xx-xlx\examples\BL</pre>	2 adv\iarauto\Debug\Exe\BLE :	adv.elf					
Tue Dec 07, 2021 17:53:	04: LowLevel	Reset(software, delay 200)							
ue Dec 07, 2021 17:53: hue Dec 07, 2021 17:53:	14: LowLevel M: CEL stat	Reset (system, delay 200)							
Tue Dec 07, 2021 17:53:	04: Download	i completed.							
Tue Dec 07, 2021 17:53:	04: LowLevel	Reset(software, delay 200)							
Tue Dec 07, 2021 17:53: Tue Dec 07, 2021 17:53:	14: Target r 14: INFO: Co	reset anfiguring trace using 'Auto size limit=1002' patting							
Tue Dec 07, 2021 17:53:	04: INFO: Ca	innot measure current when I-jet is not powering the target.							

Figure 4 - Debug Session



2.4 Set Flash Loader

When creating a new IAR project, it needs to override the default .board file. The PATH is \$TOOLKIT_DIR\$\config\flashloader\Atmosic\atmx_flash.board. See <u>Figure 5</u>.

Options for node "BLE_adv"							×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Setup Ver Sup Use	Download ify download press downlo flash loader(Override def \$TOOLKIT_ Edit Perform mas	Images bad s) ault .board DIR\$\cont s erase be	Multicore d file fig\flashload	Extra Options	Plugins x_flas	Factory Settings
					OK	Cancel	

Figure 5 - Set Flash Loader

2.5 Program Flash with NVDS

Each IAR Workspace folder will include the flash_nvds.bin file.

To program NVDS, set flash_nvds.bin as an input and program the whole image. See <u>Figure 6</u>.

- 1) Go to the linker setting page: Project Options \rightarrow Linker \rightarrow Input
 - a) Keep symbols: NVDS
 - b) Raw binary file: \$PROJ_DIR\$\flash_nvds.bin
 - c) Symbol: NVDS, Section:.NVDS, Align:8
- 2) Rebuild the whole project
- 3) Program image

						Facto	ry Setting:
ieneral Options tatic Analysis untime Checking							
C/C++ Compiler	#define	Diagnostics	Checksu	ım	Encodings	s Extra	Options
Assembler	Config I	ibrary Input	Optimizat	ions	Advanced	Output	List
Output Converter	J	,					
Custom Build	Keep sym	bols: (one per line	e)				
Build Actions			,				
Linker	11003						
Debugger							
Simulator							
CADI							
CMSIS DAP							
GDB Server							
I-jet							
J-Link/J-Trace							
TI Stellaris							
Nu-Link							~
PE micro	- Raw bin	arv image					
SI-LINK	File:				Symbol:	Section:	Alian:
	\$PRO	DIR\$\flach_pvds	hin				Q
	φrito.				14705	.14005	Ŭ.
11 / 205	File:				Symbol:	Section:	Align:

Figure 6 - Add flash_nvds.bin

3. ATM33/e and ATM34/e Platform

3.1 Installation

 Connect the USB cable to the USB Connector on the ATM3330e/ATM3330/ATM3325/ATM3430e/ATM3425/ATM3405 Evaluation Board, see <u>Figure 7</u> as an example for ATM33/e. Confirm JLink CDC UART Port and J-Link driver are showing in the Windows device manager.



Figure 7 - ATM33/e Evaluation Board

2) Flash Loader

Copy all files from <SDK installation folder>\tools\iar to <IAR installation folder\arm\config\flashloader\Atmosic>

- 3) Refer to Generate IAR Workspace to generate IAR Workspace.
- 4) Two options to build and program firmware

Option 1: Generate IAR Workspace without USE_MCUBOOT

- Refer to <u>Build Example and Program (without US_MCUBOOT)</u> to build and program bootloader
- Refer to <u>Program Image without NVDS</u> and <u>Program Image with NVDS</u> to build and program application

Option 2: Generate IAR Workspace with USE_MCUBOOT



- Edit and build the project to generate application image
 - Note: if the bootloader was updated, then it needs to follow section
 3.3 to link right Secure_Functions.o.
- Refer to IDE Auxiliary Flash Programming Tool User Guide to program generated image.

3.2 Generate IAR Workspace

Please refer to the section on **How to Generate IDE Project** from the **IDE Auxiliary Flash Programming Tool User Guide** to generate IAR Workspace.

After generating successfully, the workspace can be found in the following folder: <Atmosic SDK>\platform\atm33\ATM33xx-5\example\xxx\iar OR <Atmosic SDK>\platform\atm34\ATM34xx-2\example\xxx\iar

Open workspace file in IAR folder, i.e., example\xxx\iar\<xxx>.eww

3.3 Build and Program Bootloader

Bootloader is an example to boot the system and it will generate an object file called Secure_Functions.o for applications to link and to use the secure entry functions. So it's important to know what Secure_Functions.o was programmed into the device. The application must link with proper Secure_Functions.o.

By default, the project will automatically use a script called prebuild.bat (located in the workspace folder) to generate a bootloader image in <Atmosic SDK>\platform\atm33\ATM33xx-5\example\bootloader\ and link it with the application.

Or for ATM34/e, <Atmosic SDK>\platform\atm34\ATM34xx-2\example\bootloader\.

If the bootloader is needed to build using IAR, please refer to the following steps to build and link the Secure_Functions.o.

IAR Workspace to Generate Bootloader

- 1) Run Setup IDE Environment and ensure the J-Link driver is showing in the Windows device manager under USB devices. Setup IDE Environment can be found at Start Menu -> AtmosicSDK.
- Generate IAR Workspace, (please refer to <u>Generate IAR Workspace</u>) and Open IAR Workspace at SDK\platform\atm33\ATM33xx-5\example\bootloader\iar\bootloader.eww or SDK\platform\atm34\ATM34xx-2\example\bootloader\iar\bootloader.eww
- 3) Build and Program bootloader.
- 4) Modify the link option in the application workspace.
 - a) Project -> Options -> Linker -> Extra Options
 - b) Replace \$PROJ_DIR\$\..\..bootloader\Secure_Functions.o with --> \$PROJ_DIR\$\..\..bootloader\iar\Debug\Exe\Secure_Functions.o

3.4 Build Example and Program (without USE_MCUBOOT)

- Run Setup IDE Environment and ensure the J-Link driver is showing in the Windows device manager under USB Serial Bus controllers. Setup IDE Environment can be found at Start Menu -> AtmosicSDK.
- 2) Edit/Compile/Program Code in IAR IDE, see Figure 8.





Figure 8 - IAR Workbench GUI

- 3) For the Program and Debug image, please refer to <u>Figure 12</u> for the programming image with NVDS
 - a) Press the Download and Debug button on the toolbar or (Project \rightarrow Download \rightarrow Download and Debug), see <u>Figure 9</u>.
 - b) Runtime debug, see <u>Figure 10</u>.

ATM/2 Delvia	program and debug		
MTMAS_Debug	18/** program and debug		
Files			7
B BLE_adv - ATMx3_Debut	Dug ≥ 3 · · · · · · · · · · · · · · · · · ·		
He lib c	5 .*		
He isrc	6 ** gerief-BlE-Advertising-Application 7 **		
BLE_adv.c	8 **Copyright*(C)*Atmosic*2020-2022		
	9 .*		
	11		
	13 + include (string, h)		
	15 #include "arch.h"		
	16 finclude "stat_ble.h" 17 finclude "stat_sen.b"		
	18 #include: "atm_adv.h"		
	19 finclude "star_adv_param.h"		
	20 #include "rvds.h"		
	22 #include "mvds_tag.h"		
	23 #include "bt_gav.h" 24 #include "att_gav.h"		
	25 #include:"ata_pm.h"		
	26 #include "Swythmer.h" 27 #include "Courtis.h"		
	28 🛱 #ifdef-Auto_Test		
	29 include "uart_stdout.h" 34 bandif		
BLE_adv		A	
Build			¥ 0 1
Messages	File		Line
BLE_adv-ATMx3_Debug			
medung project nodes			
Cleaning 46 files.			
Clean succeeded			
Build Debug Log			

Figure 9 - Download and Debug

Atmosic



Figure 10 - Runtime Debug

4) See the log of the debug port as shown in Figure 11.



```
•••
@000000b4 PSEQ STATUS=0x1
@000000ff RESET_SYNDROME=0x1
@00000156 PMU WKUP_DET = 0
@000001a5 boot_status = 1000000
@00000203 Cold boot
@0000023f Power on Reset
@000002ba pmu_init
@000002ec bp_freq is 16000000
@00000340 pmu_nonharv
@00000381 SDK Version: 5.3.0
@000003cd APP Version: 0.0.0.9
WARNING: OTA update not supported with this image
NVDS_RRAM: 4e 56 44 53 06 06 27 00 00 00 00 01 02 00 00 00 ...
@0000054d [ BLE_adv][D]: user_main() done
@0000060a rw ISRs configured
@0000066d [ BLE_adv][D]: ble_adv_init: NVDS tag for adv timeout param not
found. Using default
@00000781 rwip_init() done
@000007ca Entering main loop
             atm_adv][D]: Adv dur 0(unit:10ms) max_adv_evt 0 (timeout 0ms)
@00000899 [
@00000992
             BLE_adv][D]: adv_state = 2
             BLE_adv][D]: adv_state = 4
@00000a33
@00000ac6 [
             BLE_adv][D]: adv_state = 6
@00000b68
             atm_adv][D]: Adv0: ON
@00000be1 [
             BLE_adv][D]: adv_state = 9
```

Figure 11 - BLE_adv Example Log

3.5 Program Image with NVDS (without USE_MCUBOOT)

- 1) To program with NVDS, it needs to add the settings below. See Figure 12.
- 2) Steps for adding settings to program NVDS:
- 3) Go to the linker setting page: Project Options \rightarrow Linker \rightarrow Input
 - a) Keep symbols: NVDS
 - b) Raw binary file: \$PROJ_DIR\$\flash_nvds.bin
 - c) Symbol: NVDS, Section:.NVDS, Align:8
- Rebuild the whole project and the flash_nvds.bin will be linked into the final ELF file.

 \times

5) Program image

Options for node "BLE_adv"

Category:					Facto	ory Setting
Seneral Options						
Runtime Checking						
C/C++ Compiler	#define Di	agnostics	Checksum	Encoding	s Extra	Options
Assembler	Config Library	Input	Optimizations	Advance	d Output	List
Output Converter	comig Library		opartizations	Advance	u output	LISC
Custom Build	Keep symbols: (one per line	2)			
Build Actions			7			
Linker	NVDS					^
Debugger						
Simulator						
CADI						
CMSIS DAP						
CMSIS DAP GDB Server						
CMSIS DAP GDB Server I-jet						
CMSIS DAP GDB Server I-jet J-Link/J-Trace						
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris						
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link DE micro						Ŷ
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro	Raw binary im	age				~
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver	Raw binary im File:	age		Symbol:	Section:	Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	Raw binary im File:	age	bin	Symbol:	Section:	Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$	age \flash_nvds.	bin	Symbol: NVDS	Section: .NVDS	Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$ File:	age \flash_nvds.	bin	Symbol: NVDS Symbol:	Section: .NVDS Section:	Align: 8 Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$ File:	age \flash_nvds.	bin	Symbol: NVDS Symbol:	Section: .NVDS Section:	Align: 8 Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$ File:	age \flash_nvds.	bin	Symbol: NVDS Symbol:	Section: .NVDS Section:	Align: 8 Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$ File:	age \flash_nvds.	bin	Symbol: NVDS Symbol:	Section: .NVDS Section:	Align: 8 Align:
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Raw binary im File: \$PROJ_DIR\$ File:	age \flash_nvds.	bin	Symbol: NVDS Symbol:	Section: .NVDS Section:	Align: 8 Align:



Reference Documents

Title	Document Number
ATM2/ATM3 Evaluation Kit User Guide	ATM2_ATM3-UGEVK
ATM33/e Series Evaluation Kit User Guide	ATM33_e-UGEVK
ATM34/e Series Evaluation Kit User Guide	6441-xxxx-xxxx
IDE Auxiliary Flash Programming Tool User Guide	4381-xxxx-xxxx
Interface Board User Guide	ATMx221-UGIB



Revision History

Date	Version	Description
July 3, 2024	0.63	Updated for SDK 6.0.0.
August 30, 2023	0.62	Updated IAR Workbench version under <u>Prerequisites</u> ; ATM2/ATM3 Platform: <u>Installation</u> , <u>Generate IAR Workspace</u> , ATM33/e Platform: <u>Installation</u> , <u>Generate IAR Workspace</u> ,
March 17, 2023	0.61	Updated <u>Generate IAR workspace</u> , <u>Program Flash</u> with NVDS, <u>Installation</u> , <u>Build</u> , and <u>Program</u> <u>Bootloader</u> , <u>Figure 7 - ATM33/e Evaluation Board</u> , <u>Program Image with NVDS</u>
May 13, 2022	0.60	The initial version was created for SDK 5.1 which supports ATM2/ATM3 and ATM33 platforms.

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.

Atmosic[®]

Atmosic Technologies | 2130 Gold St. Suite 200 | San Jose CA, 95002 www.atmosic.com

> IAR Workbench User Guide July 3, 2024 4247-0136-0063