

AN1497: SiWx917 SoC SWD Algorithm Programmer

This application note describes a Serial Wire Debug programmer can program the internal flash of an SiWx917 SoC.

KEY POINTS

- Overview Hardware and Software
- Programming the SiWx917 SoC with JFlash
- SiWx917 SoC Generic Flash APIs



Table of Contents

1.	Hardware Overview		3
	1.1 Hardware Platform. . <td></td> <td>3 3 3</td>		3 3 3
2.	Software Overview		4
3.	Flash Loader – Example		5
	3.1 Prerequisites		5
	3.1.1 Hardware	•	5 5
	3.2 Setup		5
	3.3 Programming the SiWx917 SoC with JFlash		6
	3.3.1 Generate .ELF File in Keil IDE	•	7 10
	3.3.3 Copy the .ELF file to the J-Link Folder.		11
	3.3.4 Erase Chip - SiWx917	-	12 17
4.	SiWx917 Generic Flash APIs		22
	4.1 Initialization	.2	22
	4.2 Program the Device		22
	4.2.1 ProgramPage API .		23 27

1. Hardware Overview

1.1 Hardware Platform

The flash programmer supports two hardware topologies: one which is based on the SiWx917 Pro Kit and the other which is based on a custom hardware design. These are described in the following sections.

1.1.1 SiWx917 Pro Kit

The SiWx917 Pro Kit consists of a radio board and a mainboard.

- The radio board supports on-chip antenna and micro-coaxial connector for conducted RF measurements or external antenna connection
- · Virtual COM port
- SEGGER J-link on-board debugger
- Breakout pads for Wireless SoC I/O

The flash programmer uses the SEGGER J-link on-board debugger for communicating with the SiWx917 SoC.

1.1.2 Customized Hardware

Alternatively, the SiWx917 SoC device can be used on a custom hardware platform to implement the programmer.

The SWDIO and SWCLK pins are used by SWD.

• The SWD pins can be used to communicate with the SWD Programmer.

2. Software Overview

This section covers the software drivers required for the programmer.

The generic flash programmer executes the flashing operation via the Network Wireless Processor (NWP) boot loader.

Description	Common Flash
Flash Memory	A single flash memory is shared between M4 (Host MCU) and NWP, and they access this flash memory over dedicated QSPI controllers.
Memory Access	An arbiter placed between the QSPIs and memory helps in arbitration for flash access.
Erase Operation	Uses the bootloader-based erase chip operation

The flash memory is shared between the M4 and NWP. Only the NWP can perform low-level operations on the flash memory, and the M4 can only support flash read operations. For low-level operations such as erase and write operations, the M4 can request the NWP bootloader to do the operations.

The steps required for flash programming are as follows.

Programming Stage	Description	APIs
Initialization	Device description needs to be provided in this stage.	In FlashDev.c file, we provided the device details in the "struct FlashDevice_t"
Program	Program the flash.	 In FlashPrg.c file, we used the following two SiWx917 APIs in the ProgramPage() API. rsi_device_init(uint8_t select_option) rsi_bl_upgrade_firmware(uint8_t *firmware_image, uint32_t fw_image_size, uint8_t flags)
Erase	Erases the entire flash	In FlashPrg.c, we used the following SiWx917 API in the Erase- Chip() API. • rsi_device_init(uint8_t select_option)

Note: The SiWx917 APIs are explained in detail in Section 4. SiWx917 Generic Flash APIs.

3. Flash Loader – Example

This section shows the steps on how to use the SEGGER template files to program the SiWx917 using the SEGGER JFlash.

Note: The SiWx917 SoC is not SEGGER Licensed, we included the Device name locally in JLinkDevices.xml file to shows this example.

3.1 Prerequisites

The following hardware and software are required for programming the device.

3.1.1 Hardware

- SiWG917 SoC Kit (BRD4338A radio board + 4002A base board)
- · USB to Type C cable for powering the kit and flashing the application
- · A PC with USB ports
- SEGGER J-Trace Pro with power cable
- · J-Link Adapter CortexM (You would get this with the J-Trace Pro when purchased)
- 20-position socket to socket connector cable

3.1.2 Software

- J-Link software (This example is verified with the version V7.96f.)
- Blinky example binary file can be downloaded here
- Source Code for Flash Loader Download here
- Keil IDE (This example is verified with MDK version 5.29.0.0)
- · Simplicity Commander: Download here

3.2 Setup

The image below illustrates the setup.



- The SiWx917 and the J-Trace are connected to a PC.
- The SiWx917 debug pins (on the 4002A board) and J-Trace J-Link Adapter CortexM Target pins are connected using a socket-tosocket connector.
- Make sure the J-Trace Target power light is ON after connecting the socket-to-socket connector.

3.3 Programming the SiWx917 SoC with JFlash

To program the SiWx917 SoC using the JFlash, you need a .elf file which is generated in the Keil IDE by compiling the Si917_Flash-loader project.

The flash programming involves the following steps, which are explained in detail in the sub-sections.

- 1. Generate the .ELF file in Keil IDE.
- 2. Add the SiWx917 device to SEGGER devices (locally).
- 3. Copy the .ELF file to the J-Link folder.
- 4. Erase the SiWx917 chip.
- 5. Program the SiWx917 using the JFlash.

3.3.1 Generate .ELF File in Keil IDE

The following steps explain how to generate the Si917_Common_Flash.elf file.

- 1. Download the SiWx917_Source_Code mentioned in the Section 3.1.2 Software.
- 2. Unzip the SiWx917_Source_Code. It should be as below.

📙 🗹 📜 🔻 SiWx917_Sou	irce_Code				- 🗆	\times
File Home Share	View					~ ?
Pin to Quick Copy Paste 2	ut opy path aste shortcut	Move to * X Delete *	New folder	Properties	Select all	ne ection
Clipboard		Organize	New	Open	Select	
← → ∽ ↑ 📜 « Dow	nloads ≻ Si	Wx917_Source_Code	~ Ū	Search SiWx9	17_Source_Co	de 🔎
E Pictures	*^ [Name		Date modifi	ed	Туре
📜 JLink		Si917_Flashloader		5/1/2024 9:2	29 AM	File folde
🔷 OneDrive - Silicon Labo	ratori					
🧢 This PC						
🧊 3D Objects						
E. Desktop						
Documents						
🖶 Downloads						
👌 Music						
Pictures						
🔣 Videos						
🔩 OSDisk (C:)						
A	~ <					>
1 item						:

3. Got the Keil project path: [Downloaded_Path]\SiWx917_Source_Code\Si917_Flashloader. Open the Si917_Flashloader.uvprojx Keil project file which will open the Keil IDE with this project file.

I P I + Si917_Flashloader			×
Pin to Quick Copy Paste short Clipboard	cut Move Copy to to to Copy Organize	New item Easy access New Oper	Open • Select all Edit Select none History Invert selection Select Select
\leftarrow \rightarrow \checkmark \uparrow 📜 $<$ Downloads :	SiWx917_Source_Code > Si917_Flashloader	マ ひ Search	n Si917_Flashloader 🛛 🔎
Quick access Desktop Downloads Documents Pictures JLink OneDrive - Silicon Laboratori This PC 3D Objects Desktop Documents Decuments	Name driver Listings Setup Src Debug_SRAM0_0x00.ini JLinkDevices.xml Si917_Flashloader.uvguix.simanda Si917_Flashloader.uvprojx Target.lin	Date modified 4/26/2024 2:48 PM 5/1/2024 9:22 AM 4/26/2024 2:48 PM 4/26/2024 2:48 PM 4/26/2024 9:12 AM 4/26/2024 9:12 AM 5/1/2024 9:22 AM 4/26/2024 9:12 AM	Type Size File folder File folder File folder File folder Gonfiguration setti Microsoft Edge HT SIMANDA File UVOPTX File UVOPTX File LIN File
Downloads	<		>
10 items 1 item selected 15.0 KB			
C.\Users\simanda\Downloads\SiWx91 File Edit View Project Flash De Project Plash De Project Common Project	7_Source_Code\Si917_Flashloader\Si917_Flashlo bug Peripherals Tools SVCS Window H C ← → P P P P P P P P P P P P P P P P P P	ader.uvprojx - µVision Help //= }	v 🗟 🌾 🍳 • 👁
Project: Si917_Flashloader			

4. Build the Project. This is generating the Si917_Common_Flash.elf file.

🔢 C:\Users\simanda\Downloads\SiWx917_Source_Code\Si917_Flashloader\Si917_Flashloader.uvprojx - μVision

File	Edit	View	Project	Flash	Debug	Peripherals	Tools	SVCS	Window	Help
	<u>i</u>	a 🥔	N	ew µVisio	n Project					
۲		🗄 🍅 🗄	N	ew Multi-l	Project Wo	orkspace				
Projec	t		0	oen Proje	ct					
	Proi	ect: Si91	CI	ose Proje	t					
	- (Commo	Ex	port						
_		Src	м	anage						
		ф-)) р		anage						
			Se	lect Devic	e for Targ	et				
		drive	Re	move Iter	n					
	T.	Start	<u> </u>	otions for	Target 'C	ommon_Flash	·			
			CI	ean Targe	ts					
			🛗 Bu	ild Target						
			🔛 Re	build all	target file	s				
			🗼 Ba	tch Build						
			🧼 Ba	tch Setup						
			🇐 Tr	anslate						
			🗒 St	op build						

5. Go to the path: [Downloaded_Path]\SiWx917_Source_Code\Si917_Flashloader\Objects. You should see the Si917_Common_Flash.elf file.

📕 🛃 = Objects				_		×
File Home Share View						~ ?
Pin to Quick Copy Paste shortcu	t Move Copy to	New item •	Properties ↓ Open ↓ ▷ Edit ↓ History	Select all Select none Invert selection		
Clipboard	Organize	New	Open	Select		
\leftarrow \rightarrow \checkmark \uparrow 📜 \ll Downloads $>$:	SiWx917_Source_Code > Si917_Flag	shloader > Objects	~ Ū	Search Objects		Ą
E Pictures 🖈 ^	Name I rsi_device_init_apis_tim.crt	Date modi 5/1/2024 9 5/1/2024 9	fied Type :41 AM CRF Fil :41 AM D File	Size	11 KB	^
 OneDrive - Silicon Laboratori 	rsi_device_init_apis_flm.o	5/1/2024 9 5/1/2024 9	41 AM O File	e	18 KB 19 KB	
This PC 3D Objects	rsi_device_init_flm.d	5/1/2024 9 5/1/2024 9	41 AM D File		1 KB 30 KB	
Desktop	<pre>rsi_hal_mcu_ioports_flm.crf rsi hal mcu ioports flm.d</pre>	5/1/2024 9 5/1/2024 9	41 AM CRF File	le	4 KB 1 KB	
Downloads Music	rsi_hal_mcu_ioports_flm.o Si917 Common Flash.build	5/1/2024 9 5/1/2024 9	41 AM O File	e HTML Do	12 KB 3 KB	
E Pictures	Si917_Common_Flash.elf	5/1/2024 9	41 AM ELF Fil	e	101 KB	
Videos	Si917_Common_Flash.hex	5/1/2024 9	41 AM HEX Fi	le	42 KB	
USDisk (C:)	Si917_Common_Flash.Inp	5/1/2024 9	41 AM LNP Fi	le	1 KB	
26 items 1 item selected 100 KB	Si917_Flashloader_Common	n_Flash.dep 5/1/2024 9	41 AM DEP Fi	le	6 KB	

3.3.2 Add SiWx917 Device to SEGGER Devices [Locally]

The following steps explain how to add the SiWx917 device to the SEGGER device locally.

- 1. Download and Install the SEGGER J-Link software mentioned in Section 3.1.2 Software.
- 2. Once installed, go to the path: [Installed_Path]\ SEGGER\Jlink. Example: Here it is installed in the path: (C:\Program Files (x86)\SEGGER\Jlink). Open the JLinkDevices.xml file with an editor of your choice.

📙 🛃 📜 = JLink					- 🗆	\times
File Home Share View						~ ?
Pin to Quick Copy Paste Shortcut	Move to Copy	New item •	Properties	Select all	ion	
Clipboard	Organize	New	Open	Select		
\leftarrow \rightarrow \checkmark \uparrow 📜 \Rightarrow This PC \Rightarrow OSDis	sk (C:) > Program Files (x86) > SE	GGER > JLink	~ Ū	Search JLink		<i>م</i>
🔄 Pictures 🛛 🖈 ^	Name	Date modi	fied Type		Size	^
📕 JLink	🔜 JLinkConfig.exe	5/19/2022	6:56 PM Appl	ication	563 KB	
	JLinkControlPanel.html	5/18/2022	5:53 PM Chro	me HTML Do	3 KB	
OneDrive - Silicon Laboratori	JLinkDevices.xml	5/1/2024 9	:19 AM Micro	osoft Edge HT	143 KB	
🧢 This PC	🔜 JLinkDLLUpdater.exe	5/19/2022	6:56 PM Appl	ication	139 KB	
3D Objects	JLinkGDBServer.exe	5/19/2022	6:56 PM Appl	ication	498 KB	
🔜 Desktop	JLinkGDBServerCL.exe	5/19/2022	6:56 PM Appl	ication	394 KB	
Documents	🔜 JLinkGUIServer.exe	5/19/2022	6:56 PM Appl	ication	307 KB	
Downloads	🔜 JLinkLicenseManager.exe	5/19/2022	6:56 PM Appl	ication	196 KB	
Music	JLinkRDI.dll	5/19/2022	6:56 PM Appl	ication extens	380 KB	
	🔡 JLinkRDIConfig.exe	5/19/2022	6:56 PM Appl	ication	125 KB	
Pictures	🔜 JLinkRegistration.exe	5/19/2022	6:56 PM Appl	ication	515 KB	
Tideos	🔜 JLinkRemoteServer.exe	5/19/2022	6:56 PM Appl	ication	470 KB	
CSDisk (C:)	🔜 JLinkRemoteServerCL.exe	5/19/2022	6:56 PM Appl	ication	370 KB	
×	🔜 JLinkRTTClient.exe	5/19/2022	IdaA M9 6:56 PM	ication	144 KB	×

3. In the JLinkDevices.xml file, go to the end of the file and add the following text and then save the file as shown below.

```
<Device>
<ChipInfo Vendor="Silicon Labs" Name="Si917" Core="JLINK_CORE_CORTEX_M4" WorkRAMAddr="0x00000000"
WorkRAMSize="0x00020000" />
<FlashBankInfo Name="QSPI Flash" BaseAddr="0x08000000" MaxSize="0x00800000"
Loader="Si917_Common_Flash.elf" LoaderType="FLASH_ALGO_TYPE_OPEN" />
</Device>
```



3.3.3 Copy the .ELF file to the J-Link Folder

1. Copy the Si917_Common_Flash.elf file generated in Step 5 in Section 3.3.1 Generate .ELF File in Keil IDE as shown below.

	📕 🗹 📕 🖛	Objects				_	
	File Home	Share View					~ ?
	Pin to Quick access	Paste Cut Paste Paste shor	tcut Move Copy to to to Delete Rename	New item ▼ ↑ New folder	Properties	Open → Se Edit Se History In	elect all elect none vert selection
	C	lipboard	Organize	New	Open		Select
Copy (Ctrl+C)	✓ ↑	📜 « SiWx917_Sou	rce_Code > Si917_Flashloader > Ob	jects 🗸	ර Search	Objects	م
Copy the selected items Clipboard.	to the cumen	ts 🖈 ^	□ Name	Date modi	fied	Туре	Size ^
	Pictures	*		5/1/2024 5		OFile	
	📕 JLink		rsi_device_init_apis_im.o	5/1/2024 9		O File	
			rsi_device_init_fim.cn	5/1/2024 9		CRF File	
	OneDrive -	Silicon Laboratori	rsi_device_init_fim.d	5/1/2024 9		D File	
	🧢 This PC		rsi_device_init_fim.o	5/1/2024 9		O File	
	🗊 3D Objec	ts	rsi_hal_mcu_loports_film.crf	5/1/2024 9		CRF FIIe	
			rsi_hai_mcu_ioports_fim.d	5/1/2024 9	:41 AM	D File	
			rsi_hal_mcu_ioports_fim.o	5/1/2024 9	:41 AM	O File	
	Documen	its	Si917_Common_Flash.build_	log.htm 5/1/2024 9	:41 AM	Chrome HTM	IL Do
	🔶 Downloa	ds	Si917_Common_Flash.elf	5/1/2024 9	:41 AM	ELF File	
	🎝 Music		Si917_Common_Flash.hex	5/1/2024 9	:41 AM	HEX File	
	E Pictures		Si917_Common_Flash.Inp	5/1/2024 9	:41 AM	LNP File	
	Videos		Si917_Flashloader_Common	_Flash.dep 5/1/2024 9	:41 AM	DEP File	\checkmark
	- 0	~	<				>
	26 items 1 ite	m selected 100 KB					

2. Paste the copied Si917_Common_Flash.elf file in the path: [Installed_Path]/SEGGER/Jlink as shown below.

📕 🛃 📜 🖛 JLin	ık				_	
File Home	Share View					~ ?
Pin to Quick Copy Pa access	Cut Copy path Copy path	Move Copy to to t	New item •	Properties ▼ History	Select all Select none Invert selection	
Clipb	oard	Organize	New	Open	Select	
← → ∽ ↑ 📕	> This PC > OSDis	k (C:) > Program Files (x86) > SE	EGGER > JLink	マ ひ Search	h JLink	Ą
E Pictures	🖈 ^ 🗌 Name	ANKS I KYTX.exe	Date modified 5/19/2022 6:56 PM	Type Application	Size 159 KB	^
loneDrive - Sili	con l	ikSWOViewerCL.exe	5/19/2022 6:56 PM	Application	156 KB	
This PC 3D Objects	JMe الملك الملك ال	em.exe n.exe	5/19/2022 6:56 PM 5/19/2022 6:56 PM	Application Application	405 KB 261 KB	
Desktop	JSco III JTA	ope.exe GLoad.exe	5/19/2022 6:56 PM 5/19/2022 6:56 PM	Application Application	429 KB 178 KB	
Documents	🔊 msv	vcp100.dll vcr100.dll	5/18/2022 6:02 PM 5/18/2022 6:02 PM	Application exten	s 412 KB s 756 KB	
👌 Music	S QtC	Core4.dll	5/18/2022 7:22 PM	Application exten	s 2,498 KB	
Pictures	QtG	Gui4.dll	5/18/2022 7:22 PM	Application exten	s 7,814 KB	
📑 Videos	🗹 🗋 Si9	17_Common_Flash.elf	5/1/2024 9:41 AM	ELF File	101 KB	
😍 OSDisk (C:)	SW	OAnalyzer.exe	5/19/2022 6:56 PM	Application	77 KB	
46 items 1 item se	elected 100 KB	nstall.exe	5/19/2022 6:57 PM	Application	199 KB	▼

3.3.4 Erase Chip - SiWx917

To confirm an example is loaded through the flash loader, first, erase the SiWx917 chip and keep it in IN mode using the Simplicity Commander by following the steps below.

- 1. Download and Install Simplicity Commander mentioned in Section 3.1.2 Software.
- 2. Go to the path: [Installed_Path]\ Simplicity Commander. Open Terminal and give the command: commander device masserase --device si917

Pin to Quick Copy Paste	Cut Copy path Paste shortcut Ut Copy path Copy path Copy path Copy to * Copy to *	New item ▼ ↑ New folder	Properties	Select all Select none Invert selection	
Clipboard	Organize	New	Open	Select	
← → ↓ ŀ <mark>•</mark> « co	mmander_winsz_xo4_101001048 > simplic	ity commander	✓ O Search S	implicity comman	der <i>p</i>
🖹 Documents 🛛 🖈 🔿	Name	Date modified	Туре	Size	
📰 Pictures 🛛 🖈	bearer	7/3/2024 6:17 PM	File folder		
OneDrive - Silicon I	iconengines	7/3/2024 6:17 PM	File folder		
	imageformats	7/3/2024 6:17 PM	File folder		
🧢 This PC	platforms	7/3/2024 6:17 PM	File folder		
3D Objects	resources	7/3/2024 6:17 PM	File folder		
E Desktop	📜 styles	7/3/2024 6:17 PM	File folder		
Documents	tokens	7/3/2024 6:17 PM	File folder		
Downloads	apack-link.json	7/3/2024 6:17 PM	JSON File	1 KB	
Music	changelog.txt	7/3/2024 6:17 PM	Text Document	61 KB	
Pictures	💅 commander.exe	7/3/2024 6:17 PM	Application	4,983 KB	
Videos	commander_apack.json	7/3/2024 6:17 PM	JSON File	4 KB	
SDick (C)	D3Dcompiler_47.dll	7/3/2024 6:17 PM	Application extens.	. 4,072 KB	
USDISK (C.)	edge.exe	7/3/2024 6:17 PM	Application	237 KB	
 Network 36 items 	edge_apack.json	7/3/2024 6:17 PM	JSON File	2 KB	
C:\Windows\System3	2\cmd.exe			- 0	×
crosoft Windows [\) Microsoft Corpor	Version 10.0.19045.4529] ration. All rights reserved.				
\Commander_win32_>	x64_1v16p10b1648\Simplicity Con	mmander>commander	device masseras	edevice s	;i917

- 3. Go to the path: [Installed_Path]\ Simplicity Commander. Double click the commander.exe.
- 4. Click on Kit, change the **Debug Mode** from MCU to IN.

ıg Interface SWD 🗸 🗸	10000 kHz V Device SiWG	917M111MGTBA		Reload Tal
Update Kit Installation package				
				Browse Install Package
Kit Information Kit: Firmware version: JLink serial number: VCOM port:	SIWx917 WI-Fi 6 and Bluetooth L	E 8MB Flash Pro-Kit PK6031A Rev. A00 1v5p0b240 440325114 COM6	Network Information IP Address: Gateway: DNS Server: MAC Address:	0.0.0.0/0 0.0.0.0 0.0.0.0
Debug Mode: Board Information	Ν	ICU V OFF MCU		Edit
Wireless Pro Kit Ma	sinboard	OUT MINI	BRD4002A Rev. A06 - 3	S/N: 234119004
SiWG917 Single-ba	nd Wi-Fi and BLE 8MB Flash Radi	o Board	BRD4338A Rev. A12 -	S/N: 234600133
	g Interface SWD Update Kit Installation package Kit Information Kit: Firmware version: JLink serial number: VCOM port: Nickname: Debug Mode: Board Information Wireless Pro Kit Ma SiWG917 Single-ba	g Interface SWD V 10000 kHz Device SiWG Update Kit Installation package Kit Information Kit: SiWx917 Wi-Fi 6 and Bluetooth L Firmware version: JLink serial number: VCOM port: Nickname: Debug Mode: Board Information Wireless Pro Kit Mainboard SiWG917 Single-band Wi-Fi and BLE 8MB Flash Radi	g Interface SWD V 10000 kHz V Device SiWG917M111MGTBA	g Interface SWD V 10000 kHz V Device SIWG917M111MGTBA

5. Upon changing the Debug Mode, the Device will be shown as ${\bf Cortex}~{\bf M4}$.

	_ Undate Kit			
	Installation package			
•				Browse
Kit				Install Package
	Kit Information		Network Information	
	Kit:	SiWx917 Wi-Fi 6 and Bluetooth LE 8MB Flash Pro-Kit	IP Address:	0.0.0/0
Device Info		PK6031A Rev. A00	Gateway:	0.0.0.0
Jevice Into	Firmware version:	1v5p0b240	DNS Server:	0.0.0
=	JLink serial number:	440325114	MAC Address:	
	VCOM port:	COM6		
/	Nickname:			
Flash	Debug Mode:	IN ~		Edit
	Board Information		3	
>_				
	Wireless Pro Kit Ma	ainboard	BRD40024 Rev. 406 -	S/N: 234110004
Console			51510027(100, 100	5/11.251115001
	SiWG917 Single-ba	and Wi-Fi and BLE 8MB Flash Radio Board		
			BRD4338A Rev. A12 -	S/N: 234600133

3.3.5 Program Using the JFlash

This section provides the steps to program the SiWx917 SoC using the JFlash.

1. Go to the Path: [Installed_Path]\SEGGER\Jlink. You will find JFlash.exe in it.

📕 🗹 📕 =	Manage	JLink			-	
File Home Share	View Application Tools					~ ?
Pin to Quick Copy Paste	Cut Copy path Paste shortcut	Delete Rename	New item • New folder	Properties	Select all Select none Invert selection	
Clipboard	Org	ganize	New	Open	Select	
\leftarrow \rightarrow \checkmark \uparrow 📜 \Rightarrow This	s PC > OSDisk (C:) > Progra	m Files (x86) → S	EGGER > JLink	∨ Ü Sea	arch JLink	Ą
📰 Pictures 🛛 🖈 ^	Name ^		Date modified	Туре	Size	^
📕 JLink	Devices		12/12/2022 7:02 PM	File folder		
OneDrive - Silicon I	Doc		12/12/2022 7:02 PM	File folder		
	ETC		12/12/2022 7:02 PM	File folder		
S This PC	Firmwares		12/12/2022 7:02 PM	File folder		
3D Objects	GDBServer		12/12/2022 7:02 PM	File folder		
Desktop	📕 RDDI		12/12/2022 7:02 PM	File folder		
Documents	Samples		12/12/2022 7:02 PM	File folder		
Downloads	USBDriver		12/12/2022 7:02 PM	File folder		
Music	JFlash.exe		5/19/2022 6:56 PM	Application	1,410 KB	
Dictures	🔜 JFlashLite.exe		5/19/2022 6:56 PM	Application	234 KB	
Fictures	JFlashSPI.exe		5/19/2022 6:56 PM	Application	1,129 KB	
videos	JFlashSPI_CL.exe		5/19/2022 6:56 PM	Application	311 KB	
USDisk (C:)	🔜 JLink.exe		5/19/2022 6:56 PM	Application	318 KB	
46 items 1 item selected	1.37 MB					·

2. Double click on JFlash.exe. This will open the SEGGER JFlash.



3. Select "Create new project" and click on Start J-Flash.

Relcor	ne to J-Flash			\times	a f	il
Please selec	ct one of the followi	ng start	options:		··· ·	
Open re	cent project:					
Create	new project					
Do not s	show this message a	again.	Start J-Flas	h		

4. Click on the "..." under the Target device. You will be re-directed to Target Device Settings. Search for Si917 under Device column and select it. Next, click on OK.

Create New Project Target device Ple Little Endian Target interface SWD 4000	× 1 × im kHz h	File her	SGER J-Flash V7.66a (di] areq Options information - Value -	View Help Target Device Settings Selected Device: 5917 Manufacturer Devi Selecton Labe Set17 Selecton Labe	ce Core Cortex-Mi	Little Endian • Core #0 • NumCores Flash: • Filter 1 #107 2	e -	×
		Applica - ٦-FT	ation log started ach V7 66a (l-Flach cd			OK Cancel		
	🔝 Target Device Se	ettings			×			
	Selected Device: Si917			Little Endian 💌	Core #0 ·			
	Manufacturer	Device	Core	NumCores	Flash :			
	Silicon Labs	 Si917 Si917 	Cortex-M4	✓ Filter	Filter			
				3	re			

5. In the Create New Project Window, click on OK.

	Create New Project	×	1
J FLASH	Target device		× a fi
Ple	Silicon Labs Si917		
0	Little Endian	~	
۲	Target interface Speed		
	SWD • 4000	∨ kHz	h
		ОК	٢.

6. Next, click on **Target** \rightarrow **Connect**.

J HASH S	EGGER	J-Flash V	7.66a - [*]	-		\times
<u>F</u> ile	<u>E</u> dit	<u>T</u> arget	<u>O</u> ptions <u>V</u> iew <u>H</u> elp			
Projec	t inform	<u>C</u> onr	nect			
Settin	ng	<u>D</u> iso	onnect			
[-]G	eneral	Test				
	Proj	Test		_		
	Host	Prod	uction Programming F7			
[-] TI	L F	<u>M</u> an	ual Programming			
	Туре		SWD	_		
	Init	. speed	4000 kHz			
	Speed	t	4000 kHz			
L-1	arget					
	MCU		Silicon Labs Si917			
	Core		Cortex-M4	Drag & Drop data file here		
	Endia	an	Little	5 1		
	Check	< core I	D No			
	Use 1	target R	AM 128 KB @ 0x0			
[+] F.	Lashbar	nk No. 0				
1						

7. In the **Emulator selection** pop-up window, select the J-Trace S/N. You can see the S/N on the backside of the J-Trace. Select the correct SN in the Emulator selection window. (Example: In the image below, the J-Trace S/N is 933001235, so SN 933001235 is selected). Next, click on **OK**.

SEGGER J-Link	/7.66	a - Emulator selection	\times
	Plea	se select the emulator you want to connect to:	
	#	USB Identification	
A	0	SN 440325114	1
plinte	1	SN 933001235	
		2 ОК	Cancel

8. Upon successful connection, you will see "Connected successfully" in the log window as shown below.

🔜 SEGGER J-Flash V7.66 <u>F</u> ile <u>E</u> dit <u>T</u> arget <u>O</u> pt	a-[*] ions <u>V</u> iew <u>H</u> elp				- 0	×
Project information	⊡ ×					
Setting	Value					
[-] General						
Project name						
Host connection	USB [Device 0]					
[-] TIF						
Type	SWD					
Init. speed	4000 kHz					
Speed	4000 kHz					
[-] Target						
MCU	Silicon Labs Si917					
Core	Cortex-M4		Drag & Drop data	a file here		
Endian	Little					
Check core ID	No					
Use target RAM	128 KB @ 0x0					
[+] Flashbank No. 0						
Log						Ð×
- [0][3]: E000000 CID - [0][4]: E0040000 CID - [0][5]: E0041000 CID - Executing init sequen - Initialized successf - Target interface spee	B105E00D PID 003BB001 I B105900D PID 000BB9A1 T B105900D PID 000BB925 E ce ully d: 4000 kHz (Fixed)	TM PIU TM				^
- Found 1 JTAG device.	Core ID: 0x2BA01477 (No v	ne)				~
Ready			Connected	Core Id: 0x2BA01477	Speed: 4000 k	Hz

9. Click on **File** → **Open data file...**, and then go to the path where you downloaded the Blinky Binary file mentioned in Section 3.1.2 Software. Next, select the sl_si91x_blinky_isp.bin file and click on **Open**.



10. Next, in the Enter start address pop-up window, give the Start address as **0x8201000** (This is the M4 MBR Start address). Then, click on **OK**.



11. Upon successful opening of the binary file, the JFlash screen will be shown as below.

SEGGER J-Flash V7.66	a - [*]																			_		×	
File Edit Target Opti	ions <u>V</u> iew <u>H</u> elp	C:\Ucorc\c	iman	da)	Dow	mlos	ode)	Dlin		Dinar	v cil	o) el	ci0	11v	hlinl	a, i	en k	vin @	09201000			-51	`
Cotting	Value	C. (03e13 (3	inan	ua (DOW	moa	ius i		ку I 20				_315		UIIII	~y_1	эр.г	in e	00201000			LP	
Setting	value	GO 10:							9			9 (4	1									
[-] General		0820_1000	01	00	00	00	0D	90	0D	90	00	8F	00	00	01	00	00	00			••		1
Project name		0820_1010	00	10	20	00	54	BO	78	06	00	00	00	00	00	00	00	00	T°x.	• • • • • •	••		
Host connection	USB [Device 0]	0820_1020	00	00	00	00	00	00	00	00	00	00	00	00	00	90	00	90			••		1
[-] TTE	. ,	0820 1040	A5	5A	00	00	00	20	20	08	00	00	00	80	00	00	00	00	¥Z				
Tune	CHD	0820 1050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
Type	SWD	0820_1060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
Init. speed	4000 kHz	0820_1070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
Speed	4000 kHz	0820_1080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		• • • • • •	••		
[-] Target		0820 1090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
MCU	Silicon Labs Si917	0820 1080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
Core	Contex-M4	0820 1000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
Core		0820_10D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
Endian	LITTIE	0820_10E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		• • • • • •	••		
Check core ID	No	0820_10F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		• • • • • •	••		
Use target RAM	128 KB @ 0x0	0820_1100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
[+] Flashbank No. 0		0820 1120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820 1130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820_1140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820_1150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
		0820_1160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		• • • • • •	••		
		0820_1170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
		0820 1190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820 11A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820_11B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00					
		0820_11C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		
		0820_11D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			••		1
Log																						Ð	×
 [0][5]: E0041000 CID E Executing init sequend Initialized successful Target interface speed Found 1 JTAG device. (Connected successfull) 	8105900D PID 000BB925 H ce Jlly 1: 4000 kHz (Fixed) Core ID: 0x2BA01477 (No /	ETM																					^
- Data file opened succe	ers\simanua\DOWnioads\n essfullv (36672 bvtes.	1 range. CR	y -11 5 of	dat	1_51 a =	0x1	_013 724F	EE59	_±s	p.oin RC of	y F fi	 le =	. 0×	1724	EE5	9)							Ϊ,
Ready									, .		0	Conn	ecte	d		- /	Со	re Id	: 0x2BA0147	7 Spee	d: 4000 kHz		

12. To flash the blinky application onto the SiWx917 device, click on **Target** \rightarrow **Manual Programming** \rightarrow **Program**.

Lot Extra gales Setting Connect IUSers\simanda\Downloads\Binky Binary File\sl_s91z_blinky_isp.bin @ 08201000 Setting Disconnect To: Image: Setting Image: Seting Image: Seting Image:	Eile		J-Fla Tarq	sh V7.66	a-[*] ions View Help																		-		×	
Setting Disconnect 1 Test 20,1000 01.00 00.00 <td< td=""><td>Project</td><td>inforn '</td><td>1</td><td>Connect</td><td></td><td>:\U</td><td>sers\si</td><td>mand</td><td>la\Do</td><td>wnio</td><td>ads\ </td><td>Blinky</td><td>/ Bii</td><td>nary</td><td>/ File</td><td>e\sl</td><td>si91</td><td>x b</td><td>olinky</td><td>isp</td><td>bin @</td><td>08201000</td><td></td><td></td><td>Ē</td><td>×</td></td<>	Project	inforn '	1	Connect		:\U	sers\si	mand	la\Do	wnio	ads\	Blinky	/ Bii	nary	/ File	e\sl	si91	x b	olinky	isp	bin @	08201000			Ē	×
[-] General Proj Host Jest 20 1000 01 00	Settin	g .		<u>D</u> isconn	ect	0 TO	:			_	~	16		1	2			3								
Host Broduction Programming 2 F7 I-] TIF Manual Programming Program Init. speed 4000 kHz Speed 4000 kHz I-] Target Check Blank MCU Silicon Labs Si917 Core Cortex-M4 Endian Little Check Core ID No Use target RMH 128 KB @ 0x0 [+] Flashbank No. 0 Verify Start Application F9 000 00 00 00 00 00 00 00 00 00 00 00 00	[-] Ge	neral Proj		<u>T</u> est		20	1000 1010 1020	01 00	00 00	00 00	0D 54	90 0 B0 7	D 9 8 0 0 0	90 96	00	8F 00	00	00	01 0		00	T°x.				^
I - J TIF Manual Programming Image: Chip 20 08 00 00 00 00 00 00 00 00 00 00 00 00	L 1	Host		<u>P</u> roducti	on Programming 2 F7	20	1030	00	00 00	00	00	00 0	0 0	00	00	00	00	00	OD 9	0 01	90					
Speed 4000 kHz Frase Sectors F3 00 00 00 00 00 00 00 00 00 00 00 00 00	1-111	Type Init.	spe	ed	SWD 4000 kHz	0	Unsec	e Chi cure (ip Chip		F2	2 0 0 0 0 0			00 00 00 00	00	00	00 00 00			00	#2		· · · · · · · · · · · · · · · · · · ·		
MCU Silicon Labs Si917 Erase Chip F4 Ou	[-] Ta	Speed rget			4000 kHz	ç	Erase	Sect	ors		F3	2 0		00	00	00	00	00								
Endlan Little Check core ID No Use target RAM 128 KB @ 0x0 [+] Flashbank No. 0 Million Start Application F9 00 00 00 00 00 00 00 00 00 00 00 00 00		MCU Core			Silicon Labs Si917 Cortex-M4		Erase <u>P</u> rogr	Chip am)	3	F4 F5			00	00	00 00 00	00	00 00 00	00 0 00 0 00 0	0 00	00			· · · · · ·		
Use target RAM 128 KB @ 0x0 Venty F8 00 00 00 00 00 00 00 00 00		Check	n cor	re ID	No	c c	Progr	am 8	k Veri	fy	F6			00	00	00	00	00	00 0 00 0 00 0	0 00 0 00	00 00					
Image: Display of the orgen of the	[+] F1	Use t ashban	arge k No	et RAM 5. 0	128 KB @ 0x0	0 0 0	<u>V</u> erity <u>R</u> ead <u>S</u> tart	back Appl	icatio	n	F8 • F9				00 00 00 00	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00	00 0 00 0 00 0 00 0	0 00 0 00 0 00 0 00	00 00 00 00 00			· · · · · · · · · · · · · · · · · · ·		
Log INR20 THUS 00 00 00 00 00 00 00 00 00 00 00 00 00						0820 0820 0820 0820 0820 0820 0820 0820	1150 1160 1170 1180 1190 11A0 11B0 11C0	00 00 00 00 00 00 00 00) 00 0 00 0 00 0 00 0 00 0 00 0 00 0 00	00 00 00 00 00 00 00 00	00 0 00 0 00 0 00 0 00 0 00 0 00 0			00 00 00 00 00 00 00 00	00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00	D 0 D 0 D 0 D 0 D 0 D 0 D 0 D 0 D 0	00 0 00 0 00 0 00 0 00 0 00 0		000000000000000000000000000000000000000			· · · · · · · · · · · · · · · · · · ·		~
<pre>- [0][5]: E0041000 CLD E1055000 P1D 000E0525 EIM - Executing init sequence Initialized successfully - Target interface speed: 4000 kHz (Fixed) - Found 1 JTAG device. Core ID: 0x2BA01477 (None) - Connected successfully Opening data file [c:\users\simmada\Downloads\Blinky Binary File\s1_si91x_blinky_isp.bin] Data file opened successfully (36672 bytes, 1 range, CRC of data = 0x1724FE59, CRC of file = 0x1724FE59)</pre>	Log					10820	TTDO	00	00 00) 00	00	00 0	0 0	0	00	00	00	00	00 0	0 00	00				Ð	×
	- [0] - Exe - Ir - Tar - Fou - Con Openir - Dat	[5]: E0 ecuting itiali: get int ind 1 J inected ig data a file	00410 init zed s terfa TAG o suco file oper	000 CID I t sequent successfu ace speed device. (cessfully e [C:\Use ned succe	sidosyddD PID 00088925 E ce ully 1: 4000 kHz (Fixed) Core ID: 0x28A01477 (Nor ers\simanda\Downloads\E essfully (36672 bytes,	one) Blinky 1 ran	Binary ge, CRC	File of e	e\sl_s data =	;i91× • 0x1	_bli 724F	nky_i E59,	sp. CRC	bin : of] fil	e =	0x1	724	FE59)							~

13. Upon successful programming, you will see "Target programmed successfully".

Project information	5 >	< C:\Users\si	imanda\Downloads	S\Blinky Bina	ry File\s	l_si91x_blin	ky_isp.bi	n @ 08201000		Ð	>
Setting	Value	Go To:		🖂 🛞 [4 🗋					
[-] General		0820 1000	01 00 00 00 01	0 90 0D 90	00 SF	00 00 01	00 00 0				1
Project name		0820_1010	00 10 20 00 54	BO 78 06	00 00	00 00 00	00 00 0	00T°x			1
Host connection	USB [Device 0]	0820_1020			00 00	00 00 00	00 00 0	00	•••••		
[-] TTF		0820 1040	A5 5A 00 00 00	20 20 08	00 00	00 80 00	00 00 0	00 ¥Z			
Type	SWD	0820_1050	00 00 00 00 00	0 00 00 00	00 00	00 00 00	00 00 0				
Tait and	4000 1-11-	0820_1060	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0		•••••		
init. speed	4000 KHZ	0820_1070			00 00	00 00 00	00 00 0		•••••		
Speed	4000 kHz	0820 1080			00 00	00 00 00	00 00 0	0			
-] Target		0820 10A0	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0	0			
MCU	Silicon Labs Si91	7 0820 10B0	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0				
Core	Cortex-M4	10000-1000					00 00 0				
Endian	Little	J-Flash V7	.66a			X	00 00 0	00	•••••		
Check core TD	No						00 00 0	0			
Use tanget RAM	139 KB A AvA	Target	programmed succes	fully - Compl	ated after	1 500 sec	00 00 0				
Use carget NAM	120 KD @ 0X0		programmed succes	orany compr		1.555 500	00 00 0				
+] Flashbank No. 0						01/	00 00 0	00	•••••		
						UK	00 00 0		•••••		
	L	0820 1150	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0	0			
		0820 1160	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0	0			
		0820 1170	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0	00			
		0820 1180	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0				
		0820_1190	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0				
		0820_11A0	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0				
		0820_11B0	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0				
		0820_1100	00 00 00 00 00	00 00 00	00 00	00 00 00	00 00 0		•••••		
1				1 00 00 00	00 00	00 00 00	00 00 0			-5	
Log											4
- Programming range 0x0	8201000 - 0x08208FFF	(8 Sectors,	32 KB)								
- Fod of flash programm	6209000 - 0X06209FFF	(I Sector, 4	+ KD)								
- Flash programming per	formed for 1 range (3	6864 bytes)									
- 0x8201000 - 0x8209FFF	(9 Sectors, 36 KB)	,									
- Start of restoring											
- End of restoring											
 larget programmed succ 	cesstully - Completed	atter 1.599 s	sec								

14. After programming successfully, you will see the LED0 on the board blinking continuously. This proves that the application is flashed through the JFlash and is running as expected.



4. SiWx917 Generic Flash APIs

This section explains how to configure and use flash APIs for the generic flash programmer.

The related software modules are found at SiWx917_SoC_Flash_Loader\Source_Code\Si917_Flashloader\Src. (This source code is provided in Section 3.1.2 Software.)

There are two main files:

- · FlashDev.c
- · FlashPrg.c

These two files are SEGGER template files. One can port these files to their system to make a flash programmer of their own.

The flash APIs provide complete low-level access to the flash memory and help to modify the flash memory contents. The APIs support flash memory erase, program, and read operations.

The code blocks (shown in the table below) play a major role in the flash programmer.

Code Block	Description
FlashOS.h	Contains all defines and prototypes of public functions
FlashDev.c	Flash device Description
FlashPrg.c	Implementation of RAM Code

4.1 Initialization

The initialization is done in the FlashDev.c file. The SEGGER template is used.

Though this is dummy, it is included for device description.

```
File
        : FlashDev.c
Purpose : Flash device Description Template
#include "FlashOS.h"
struct FlashDevice_t const FlashDevice __attribute__ ((section ("DevDscr"))) = {
                              // Algo version. Must be == 0x0101
 FLASH_DRV_VERS,
  { "Si917_Generic_Flash" },
                                   // Flash device name
  ONCHIP,
                                            // Flash device type. Must be == 1
  0x08000000,
                                          // Flash base address
  0x00800000,
                                          // Total flash device erase size in Bytes
  4096,
                                               // Page Size (Will be passed as <NumBytes> to ProgramPage().
                                                                A multiple of this is passed as <NumBytes> to
SEGGER_OPEN_Program() to
                                                       program more than 1 page in 1 RAMCode call, speeding up
programming).
                                                  // Reserved, should be 0
  0.
                                             // Flash erased value
  0xFF,
  50000000,
                                         // Program page timeout in ms
  50000000,
                                        // Erase sector timeout in ms
      0x00001000, 0x00000000 },
     0xffffffff, 0xffffffff }
                                  // Indicates the end of the flash sector layout. Must be present.
};
```

4.2 Program the Device

In FlashPrg.c, implementation of the RAM code template is done. The file contains SEGGER template APIs.

Among all the APIs, the **ProgramPage()** and **EraseChip()** play a major role in programming the common flash SiWx917 device.

4.2.1 ProgramPage API

{

The code block shown below explains the ProgramPage API.

```
*
   ProgramPage
*
* Function description: Programs one flash page.
* Parameters
*
     DestAddr
                 - Address to start programming on
    NumBytes - Number of bytes to program. Guaranteed to be == <FlashDevice.PageSize>
*
     pSrcBuff
                  - Pointer to data to be programmed
* Return value
     == 0 O.K.
*
     == 1 Error
* /
#define TA_RESET_ADDR
                           0x22000004
int ProgramPage(U32 DestAddr, U32 NumBytes, U8 *pSrcBuff)
 int32_t status = 1;
   static int32_t x
                     = 0, size;
   uint32_t Imageheader[HEADER_LENGTH];
 *(uint32_t *)(TA_RESET_ADDR) = 0x0; //put TA in reset
#ifdef DEBUG_OFL
*(volatile uint32_t *)0x24048624 |= (1<<5);
#endif
memset(Imageheader, '\0', HEADER_LENGTH);
  if (x == 0)
 {
     memcpy(Imageheader, pSrcBuff, HEADER_LENGTH);
    if (!board_ready)
         if((uint32_t)(Imageheader[0] & IMAGE_TYPE_MASK) == (uint32_t)TA_IMAGE)
          ł
             status = rsi_device_init(BURN_NWP_FW);
           else
          ł
             status = rsi_device_init(RSI_UPGRADE_IMAGE_I_FW);
      if (status != RSI_SUCCESS) {
        return status;
      }
    }
          size = Imageheader[2];
        size = (size) / CHUNK_SIZE;
    status = rsi_bl_upgrade_firmware(pSrcBuff, NumBytes, 1);
     size--;
     x = 1;
   #ifdef DEBUG_OFL
    *(volatile uint32_t *)0x24048624 &= ~(1<<5);
   #endif
    return status;
  if (size == 0) {
    status = rsi_bl_upgrade_firmware(pSrcBuff, NumBytes, 2);
     x = 0;
  } else {
    status = rsi_bl_upgrade_firmware(pSrcBuff, NumBytes, 0);
    size--;
#ifdef DEBUG_OFL
    *(volatile uint32_t *)0x24048624 &= ~(1<<5);
#endif
  return status;
}
```

Function Description

- · Mandatory function. Must be present to make OFL(Open Flash Loader) detected as valid.
- Programs flash. The block passed to this function is always a multiple of what is indicated as page size by FlashDevice.PageSize.
- This function can rely on only being called with destination addresses and NumBytes that are aligned to FlashDevice.PageSize.
- In this function, we call the boot loader APIs, to load the M4 and TA firmware for the common flash radio board.

API: rsi_device_init

Source File: rsi_device_init_apis_flm.c

Prototype:

int32_t rsi_device_init(uint8_t select_option);

Description

This API power cycles the module and sets the boot up option for module features. This API also initializes the module SPI.

Parameter

Parameter	Description
select_option	RSI_LOAD_IMAGE_I_FW : To load Firmware image
	RSI_LOAD_IMAGE_I_ACTIVE_LOW_FW : To load active low Firmware image. Active low firmware will generate active low interrupts to indicate that packets are pending on the module, instead of the default active high.
	RSI_UPGRADE_IMAGE_I_FW : To upgrade firmware file
	ERASE_COMMON_FLASH : To Erase the Common flash region.

Return Values

Value	Description
0	Successful execution of the command
Non Zero Value	-1: Failure

Example

if (!board_ready) {
<pre>status = rsi_device_init(RSI_UPGRADE_IMAGE_I_FW); if (status != RSI_SUCCESS) { return status; }</pre>

API: rsi_bl_select_option

Source File: rsi_device_init_flm.c

```
Prototype:
```

```
int16_t rsi_bl_select_option(uint8_t cmd);
```

Description

This API is used to send firmware load requests to TA or update default configurations.

Parameter

Parameter	Description
cmd	Type of configuration to be loaded

Return Values

Value	Description
0	Success
<0	Failure

Example

API: rsi_bl_upgrade_firmware

Source File: rsi_device_init_flm.c

Prototype:

int16_t rsi_bl_upgrade_firmware(uint8_t *firmware_image, uint32_t fw_image_size, uint8_t flags);

Description

- This API upgrades the firmware in the module device from the host. The firmware file is given in chunks to this API.
- Each chunk must be a multiple of 4096 bytes unless it is the last chunk.
- For the first chunk, set RSI_FW_START_OF_FILE in flags.
- For the last chunk set RSI_FW_END_OF_FILE in flags.

Parameters

Parameter	Description
firmware_image	This is a pointer to firmware image buffer
flags	1 - RSI_FW_START_OF_FILE
	2 - RSI_FW_END_OF_FILE
	Set flags to
	1 - If it is the first chunk
	2 - If it is last chunk,
	0 - For all other chunks
fw_image_size	This is the size of firmware image

Return Values

Value	Description
0	Successful execution of the command
Non Zero Value	-1: Failure

Example

rsi_bl_upgrade_firmware(fw_image, FW_IMG_SIZE, 1);

API: rsi_bootloader_instructions

Source File: rsi_device_init_flm.c

Prototype:

int16_t rsi_bootloader_instructions(uint8_t type, uint16_t *data);

Description

This API is used to send boot instructions to TA.

Parameters

Parameter	Description
type	Type of the instruction to perform
data	Pointer to data which is to be read/write

Return Values

Value	Description
0	Success
Non Zero Value	Failure

Example

rsi_bootloader_instructions(RSI_REG_READ, dat

4.2.2 EraseChip APIs

The below code block explains the EraseChip operation.

```
/*
         EraseChip
*
*
  Function description: Erases the entire flash.
*
*
  Return value
*
    == 0 O.K.
*
     == 1 Error
*/
#if SUPPORT_ERASE_CHIP
int EraseChip(void) {
    int status =RSI_SUCCESS;
       //Send the common flash erase command to the TA
        status = rsi_device_init(ERASE_COMMON_FLASH);
        if(status==RSI_SUCCESS)
            return RSI_OK;
        else
            return 0;
#endif
```

Function description:

• This function is used to Erases the entire flash for the common flash radio board.

API: rsi_device_init

Source File: rsi_device_init_apis_flm.c

Prototype :

```
int32_t rsi_device_init(uint8_t select_option);
```

Description

This API power cycles the module and sets the boot up option for module features. This API also initializes the module SPI.

Parameter

Parameter	Description
select_option	ERASE_COMMON_FLASH: To Erase the Common flash region.

Return Values

Value	Description
0	Successful execution of the command
Non Zero Value	-1: Failure

Example

```
status = rsi_device_init(ERASE_COMMON_FLASH);
    if(status==RSI_SUCCESS)
        return RSI_OK;
```





www.silabs.com/products



Quality www.silabs.com/quality



Support & Community www.silabs.com/community

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon La

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], Silabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals[®], WiSeConnect, n-Link, EZLink[®], EZRadio[®], EZRadio[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc. 400 West Cesar Chavez Austin, TX 78701 USA

www.silabs.com