NXP Semiconductors
Application Notes

Document Number: AN13456

Rev. 1, 02/2023

S32G3 Boot Process

by: NXP Semiconductors

1. Introduction

S32G3 is a high-performance vehicle network processor, combining CAN/LIN/FlexRay networking with high data rate Ethernet networking.

The target application for S32G3 are:

- Central gateways and domain controllers connecting various networks and translating their protocols
- Firmware Over-The-Air (FOTA) masters controlling secure software image downloads and their distribution to the ECUs in the network
- Secure key management
- Smart antennas
- High-performance central compute nodes

Contents

2. Key terms

- **Boot Core** – HSE_M7 the only CPU available when the hardware reset sequence completes
- **BootROM** – Firmware available in HSE ROM area that is executed by HSE_M7 after reset.
- **Non Secure boot** – BootROM passes control to the user application residing outside HSE subsystem.
- **Secure boot** – BootROM passes control to the HSE firmware running on HSE_M7. BOOT_SEQ bit in the IVT Boot configuration word defines if Secure boot is to be enabled.
- **Boot mode pins (BMODE1 and BMODE2)** – These pins are available to be configured by user to select the desired boot mode. These are the first user inputs read by BootROM.
- **Serial boot** - This boot mode allows application code to be downloaded to SRAM using LIN, CAN or Ethernet.
- **RCON** - Boot configurations defined outside of the chip. RCON can be parallel (using 32 dedicated IOs) or serial (using I2C based EEPROM).
- **External Flash** – Non Volatile memory connected on QSPI, SD or MMC interface.
- **IVT (Image Vector Table)** - The Image Vector Table (IVT) is the first image that the BootROM reads from the boot device. The IVT contains the required data components like image entry point, pointer to Device Configuration Data (DCD) and other pointers used by the BootROM during the boot process. The location of the IVT is the only fixed requirement by BootROM.
- **CM7_0 and A53_0** – First application cores to be enabled by BootROM on successful completion. Either of CM7_0 or A53_0 are selected based on BOOT_TARGET configuration in IVT Boot Configuration Word.

3. Boot differences between S32G3 and S32G2

Table 1 Boot features difference summary

Boot Feature	S32G2x	S32G3x
UART Baud Rates (for different source clock)	XOSC (20 MHz) - 24000 XOSC (40 MHz) - 48000 FIRC (48 MHz) - 48000	XOSC (20 MHz) - 57600 XOSC (40 MHz) - 115200 FIRC (48 MHz) - 115200
IVT Life-Cycle Configuration Word	1-bit used for each Life Cycle	4-bit pattern used for each Life-Cycle
QSPI Boot Initial Phase Frequency	40 MHz	30 MHz
Self-Test, DCD, App	No mechanism to identify whether	SRC_GPR_TOP_REG_28 can be

Boot Feature	S32G2x	S32G3x
Image identification (Primary/Backup)	primary or backup image was used in current boot cycle	used to identify if primary or backup image is used in current boot cycle
QSPI POR Delay	QSPI POR Delay applied by default only at POR. HSE service used to control POR delay across subsequent boot cycles	QSPI POR Delay applied by default only at POR. Application core can use SRC_GPR_TOP_REG_29 to control the POR Delay on subsequent boot cycles
Reserved SRAM region for boot via µSDHC interface	SRAM region 0x343FF000 – 0x34400000 is used for ADMA descriptors.	SRAM region 0x34000000 – 0x34002000 is used for ADMA descriptors

4. Setting up the first Boot

For a virgin device, the device can be forced to enter Serial boot mode using BMODE pins. In case boot from an external flash is selected using BMODE and no image is present in the external flash, the device enters into serial boot mode after eight resets.

Refer to Serial Boot for further details.

Key steps for booting up the device:

1. Ensure power supply on the board
2. Ensure correct BMODE settings
3. Lauterbach (LTB) and T32 support for S32G3

Steps to load an application and execute directly from SRAM using LTB

1. Example T32 Script to attach to S32G3 CM7_0 is provided with the T32 patch for S32G3
2. Custom application, use data.load.elf <File_name.elf> from the LTB window

   In this case BootROM is still polling the serial interfaces for a valid image through serial boot. BootROM may interfere with application UART/CAN transmission.

Steps if Serial boot is required:

1. Application binary to download
2. Tool to transfer binary on Serial (UART/CAN) port.
3. Hardware bus connection for the serial interface to target board

Steps if required to boot from QSPI:

1. Prepare binary image using S32DS IVT configurator
2. Use S32DS flash tool to program the QSPI flash. The tool is available with S32DS installation at directory - <installation_directory> \S32DS.3.4\S32DS\tools\S32FlashTool\GUI\s32ft.exe

3. Select the target, Algorithm as QSPI flash component on hardware and COM port as per individual machine configuration
4. Follow the instructions available with S32DS flash tool (S32_Flash_Tool_User_Guide.pdf in the S32DS installation directory) for further steps
5. Configure RCON as per application requirement, example recommendation in attached Boot Config Word setting
6. Configure BMODE to ‘10 - Boot from external memory using RCON configurations’ (See Table 2 below for further details)
7. The device will boot from QSPI after the next reset

Steps if required to boot from SD/MMC:

1. Prepare binary image using S32DS IVT configurator
2. Use S32DS flash tool to program the SD/MMC. The tool is available with S32DS installation at directory <installation_directory> \S32DS.3.4\S32DS\tools\S32FlashTool\GUI\s32ft.exe
3. Select the target, Algorithm as SD/MMC component on hardware and COM port as per individual machine configuration
4. Follow the instructions available with S32DS flash tool for further steps
5. Alternatively, a utility application like Win32DiskImager can also be used to format and write raw binary into the SD card. Plug in the SD card into the target board
6. Configure BMODE to ‘10 - Boot from external memory using RCON configurations’ (See Table 2 below for further details)
7. Configure RCON as per recommendation in Boot Config Word setting
8. The device will boot from SD/MMC after the next reset

4.1. Bring up considerations

S32G3 boot architecture is designed in such a way that it always expects the device to be in serial boot mode when none of the NVM devices have a verified application in it. Once in serial boot mode, the device expects to receive the application on UART/CAN interface as defined in the Serial boot.

During development phases, it is always recommended to configure the BMODE pins to put device in serial boot mode when erasing/programming the external NVM. User must always verify external flash for the availability of right content.

In a production line scenario, for a virgin device the FUSE_SEL would be available in reset state (depicted by 0 in the Table 2) and setting BMODE pins to serial boot would allow downloading the complete application image. Flash tools working on serial boot mode and provided with S32 Design Studio (S32DS) can also be used. As a next step, the FUSE_SEL should be blown such that the boot configurations are picked up from BOOT_CFG fuses rather than external RCON. After FUSE_SEL is set to 1, the BMODE configurations earlier depicting serial boot now automatically configures the device to boot from external NVM, refer to Table 2. The type of NVM is configured based on the BOOT_CFG1 fuse word.

5. Boot prerequisites

5.1. Power up

User must ensure that all power supplies required for operation of S32G3 are up in the sequence as mentioned in the S32G3 Data Sheet. Also, the supply limits for all power pins mentioned in the device Data Sheet must be respected at all times.

5.2. BMODE configuration

BMODE1 and BMODE2 pins are the first user inputs required by the device to configure the BootROM into correct state. These pins are sampled by the BootROM to decide the boot configuration of the device.

For the first unprogrammed samples, it is recommended to set BMODE pins for serial boot operation. Setting the boot mode to serial configuration also ensures that the HSE_M7 SWT does not timeout. BootROM disables the SWT in Serial boot mode for device lifecycle CUST_DEL. For lifecycle later than CUST_DEL a timeout of 60 seconds is applicable. Serial boot is only available if the DIS_SER_BOOT fuse bit is not blown.

Table 2 Supported Boot configuration

FUSE_SEL	BMODE1	BMODE2	Boot Mode	Available Boot Interfaces	Use Case
0	0	0	**Serial Boot**, XOSC configured in differential bypass mode*	Serial communication interfaces: LIN (UART) and CAN	
	0	1	**Serial Boot**, XOSC in Crystal mode or Bypass mode	Serial communication interfaces: LIN (UART) and CAN	Production config for virgin devices
	1	0	Boot from external memory using **RCON configurations**	Memory interfaces: QSPI, SD card, MMC, eMMC	Development config
1	0	X	Boot from external memory using **Fuse configuration**	Memory interfaces: QSPI, SD card, MMC, eMMC	Production config for programmed devices
	1	0	**Serial Boot**	Serial communication interfaces: LIN (UART) and CAN	Debug of programmed device
X	1	1	Reserved		

NOTE

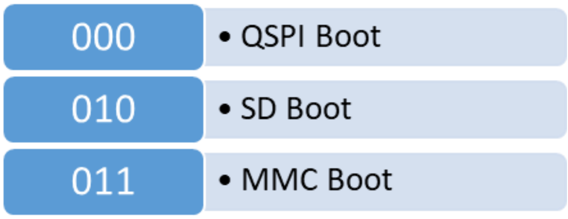
XOSC in differential bypass mode is not a supported configuration in S32G3 devices. Refer to FXOSC chapter in S32G3 Reference Manual for further details.

5.3. FUSE configuration

Fuse map table with details on all available fuses is attached with the S32G3 Reference Manual.

The fuse map table shows three BOOT_CFG_LOCK fuses. BootROM uses these three fuses to configure it's operations. Of these BOOT_CFG1 is the fuse that essentially needs to be configured before the device is powered on with FUSE_SEL fuse blown. RCON settings must be seen as a development time substitute for BOOT_CFG1 fuse only. The configuration of RCON pads on the hardware should also be done in accordance with the bit field description of BOOT_CFG1 fuse.

Bits 7-5 in the BOOT_CFG1 are the external memory selector pins.



000	• QSPI Boot
010	• SD Boot
011	• MMC Boot

Figure 1 BOOT_CFG1 Memory Selector bits

The rest of the bits in the three boot configuration fuses are used for configuration of external memory interface or configuration of serial boot supporting peripherals.

5.4. RCON configuration

During development phase, developer may need to try different boot configurations before the final boot strategy is selected. E.g., the QSPI flash may not be programmed on the first day and setting boot configuration to QSPI flash may result in no action from BootROM and subsequent resets. Thus a mechanism may be required wherein user can download their application directly to internal SRAM. Serial boot can be used in such cases which may later be required to be modified to boot from an external NVM (e.g., QSPI flash or SD card) at a later stage. In such scenario, using Fuses may not be a viable options since a fuse once blown cannot be reverted or changed. For such cases, RCONs are provided that can be used in lieu of the boot fuses. The 32 bits of fuses are mimicked with 32 parallel GPIO lines (parallel RCON) or through a serial EEPROM (serial RCON), that are sampled during the Functional Reset sequence.

Parallel RCONs can be used by connecting 32 DIP switches on the 32 RCON I/Os and can be set to different values as per the development requirement.

To use serial RCON, RCON8 pin must be pulled HIGH. The BootROM then assumes that a serial EEPROM is connected on the I2C port (RCON8 and RCON9). The 32 reset configuration bits are then fetched from the external EEPROM (first 4 bytes from offset 0x0 of the device) as part of the Boot process.

The address of the EEPROM device should be set as 0xA0.

FUSE_SEL fuse must be blown in case S32G3 is required to be forced to boot using Fuses. The BOOT_CFGn fuses must be blown prior to setting the FUSE_SEL fuse.

5.5. Setting up the external memory for Boot

Any external NVM memory interface selected by means of Fuses/RCON, should have the images programmed in a specific way for the BootROM to be able to process the image. The programmed image must have the following sections:

- Image Vector Table (IVT)
- Device Configuration Data (DCD)
- Self-Test DCD
- HSE_H Firmware
- Application Boot Code Image

S32DS also comes with configurator tools that can be directly used to generate these images. For further information on the same, please check the help manuals in S32DS.

5.5.1. Image Vector Table (IVT)

The Image Vector Table or the IVT is the first image read by BootROM from the boot device. As such there is a precise requirement to place the IVT in a predefined location of the external memory used/programmed. For QSPI boot, the BootROM expects the IVT to be placed at the 0h location of external memory. For SD/MMC/eMMC boot, the IVT should be placed at 0x1000 offset. Pointers to all other requirements/images is then parsed by the BootROM from the IVT. The IVT table structure is shown below.

Table 3 IVT image structure

Address Offset	Size (in Bytes)	Name	Comment
0x0	4	IVT Header	Header showing the start of IVT
0x04	4	Reserved	Reserved
0x08	4	Self-Test DCD pointer	Pointer to the start of the configuration data used for BIST
0x0C	4	Self-Test DCD Backup pointer	Pointer to the start of the backup configuration data used for BIST
0x10	4	DCD Pointer	Pointer to the start of DCD configuration data
0x14	4	DCD Backup pointer	Pointer to the start of backup DCD configuration data
0x18	4	HSE FW Flash Start pointer	Pointer to the start of the HSE_H firmware in flash memory
0x1C	4	HSE FW Backup Flash Start pointer	Pointer to the start of the backup HSE_H firmware in flash memory
0x20	4	Application Flash Start pointer	Pointer to the start of the application boot code in flash memory
0x24	4	Application Backup Flash Start pointer	Pointer to the start of the backup application boot code in flash memory

Address Offset	Size (in Bytes)	Name	Comment
0x28	4	Boot Configuration Word	Configuration data used to select the boot configuration
0x2C	4	Life-Cycle Configuration Word	Configuration data used for advancing Life-Cycle
0x30	4	Reserved	Reserved
0x34	36	Reserved for HSE_H firmware	Defined by HSE_H firmware specification
0x58	140	Reserved	Reserved
0xE4	12	Random Initialization Vector	Random Value of IV used in AES-GCM operation used for IVT authentication
0xF0	16	Galois Message Authentication Code (GMAC)	GMAC of first 240 bytes of IVT image structure

The **IVT image Header** is a fixed value (0xD1010060 – shown here in Big Endian format) for S32G3 and is also shown in the device RM.

The **IVT Boot Configuration Word** can be used to select:

i. The boot core - either M7_0 or A53_0 as the boot core (bits 0-1),

ii. Enable/Disable boot target watchdog (bit2) (SWT0 can be only configured)

iii. Secure boot mode - configures between the secure and non-secure boot mode (bit 3)

Table 3 IVT Boot Configuration word

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												BOOT_SEQ	SWT	BOOT_TARGET	

The **IVT Life-cycle configuration word** can be used to advance the lifecycle of the device to OEM_PROD (bits 3-0) or IN_FIELD (bit 7-4). Life cycle once advanced cannot be reverted back.

Table 4 IVT Life-cycle configuration word

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IN_FIELD				OEM_PROD			

BootROM supports concept of backup image for Self-test DCD, DCD, Application and HSE_FW. For each type of program image, BootROM first tries to execute its primary image if IVT points to a valid location. Only 0x0 is considered an invalid location. If primary image points to 0x0 or primary image header does not match with image required header or authentication fails, wherever applicable, BootROM tries to execute backup image pointed by IVT.

5.5.2. Self-test DCD

BootROM executes Self-test only when coming out of POR, and if no previous self-test has been run (i.e. ST_DONE is not set in RGM). For Self-test to be run by BootROM, the self-test DCD section in IVT must also be programmed. Self-test DCD follows the same structure as other DCD defined in next section.

Table 5 Self-test Header

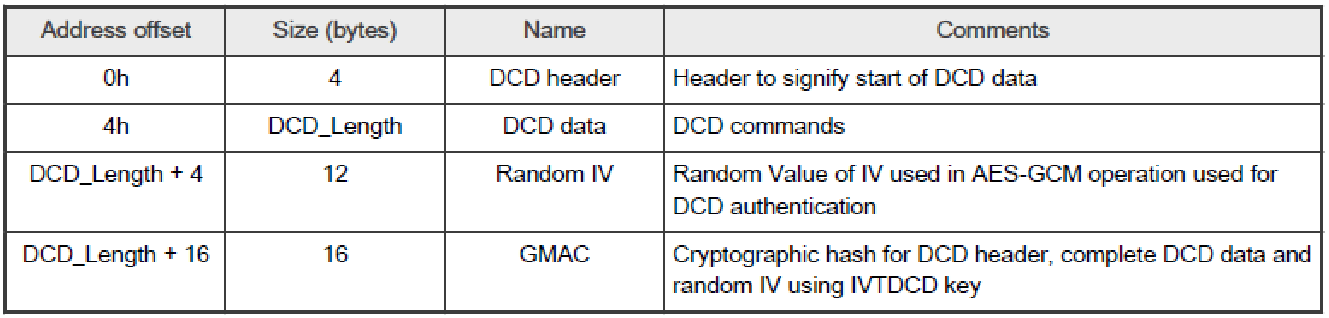
Byte 0	Byte 1	Byte 2	Byte 3
TAG = D3h	Length		Version = 60h

Execution of Self-test is always followed by a reset. The recommended Self-test DCD is provided as a binary with Safety Software Framework (SAF) release.

5.5.3. Device Configuration Data (DCD)

DCD is the configuration information contained in the DCD Image, that the BootROM interprets to configure various peripherals on the device. DCDs can be used to configure any peripheral before the control is handed over to application. The maximum size of DCD image can be 8192 bytes.

Table 6 DCD Image structure



Address offset	Size (bytes)	Name	Comments
0h	4	DCD header	Header to signify start of DCD data
4h	DCD_Length	DCD data	DCD commands
DCD_Length + 4	12	Random IV	Random Value of IV used in AES-GCM operation used for DCD authentication
DCD_Length + 16	16	GMAC	Cryptographic hash for DCD header, complete DCD data and random IV using IVTDCD key

The header structure for DCD is as follows:

Table 7 DCD header

Byte 0	Byte 1	Byte 2	Byte 3
TAG = D2h	Length		Version = 60h

5.5.4. Application Boot Code Image

For nonsecure boot configuration, the application image pointed to by IVT should comply with the below structure.

Table 8 Application boot image structure

Address Offset	Size (in Bytes)	Name	Comment
0x0	4	Image header	Header to signify start of application image.
0x04	4	RAM Start pointer	Pointer to the RAM location BootROM uses to copy the code.
0x08	4	RAM entry pointer	Pointer to start of code execution. This pointer should be within the section of SRAM where the application image is downloaded.
0x0C	4	Code length	Length of code section of the image.
0x10	48	Reserved	
0x40	Code_length	Code	Code can be any size up to the max size of System SRAM.

The Image header should be as shown below:

Table 9 Application header

Byte 0	Byte 1	Byte 2	Byte 3
TAG = D5h	Reserved		Version = 60h

6. BootROM

BootROM is the first software that runs in S32G3. It is placed in an internal ROM (accessible only to HSE_M7) and is executed by HSE_M7. After the execution of the BootROM, the control is passed either to HSE FW (in case of secure boot) or to customer application (in case of non-secure boot).

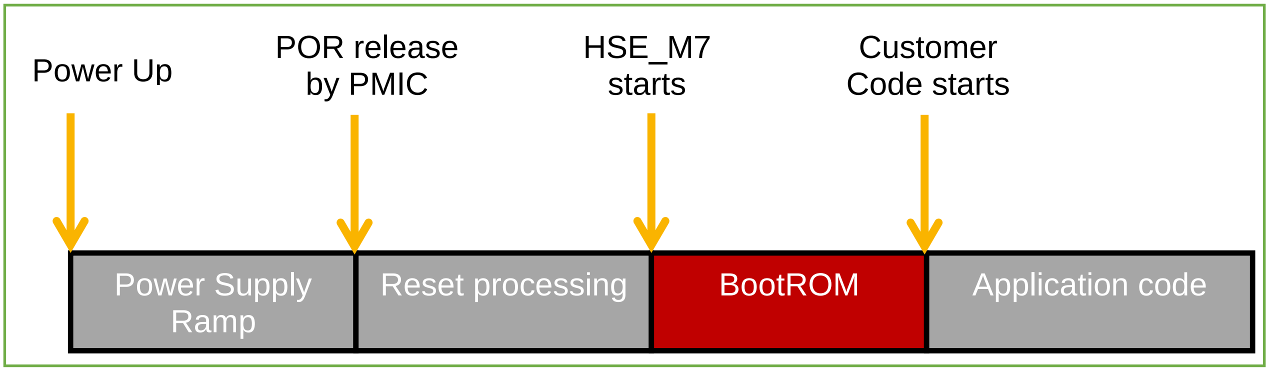


Figure 2 Non Secure boot flow

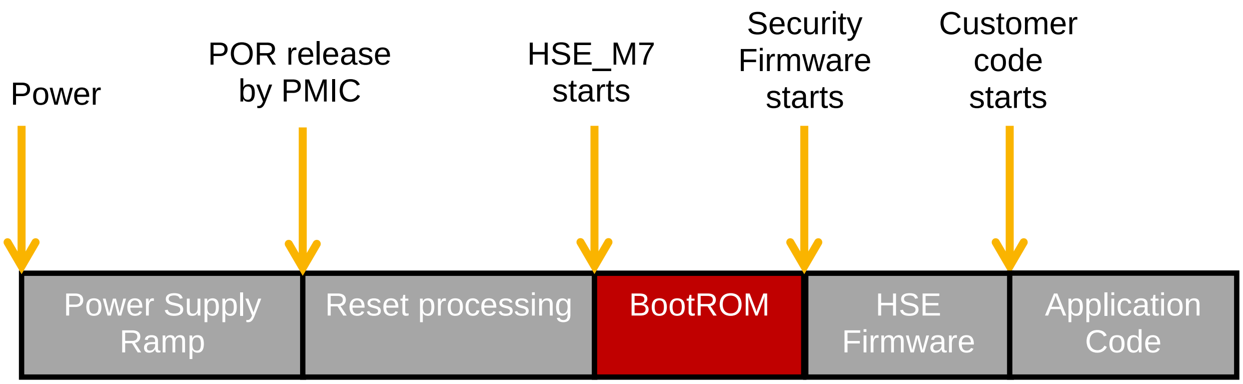


Figure 3 Secure boot flow

BootROM relies on user inputs in the form of BMODE pins, RCON and Fuse configurations to decide and enable the appropriate boot mode.

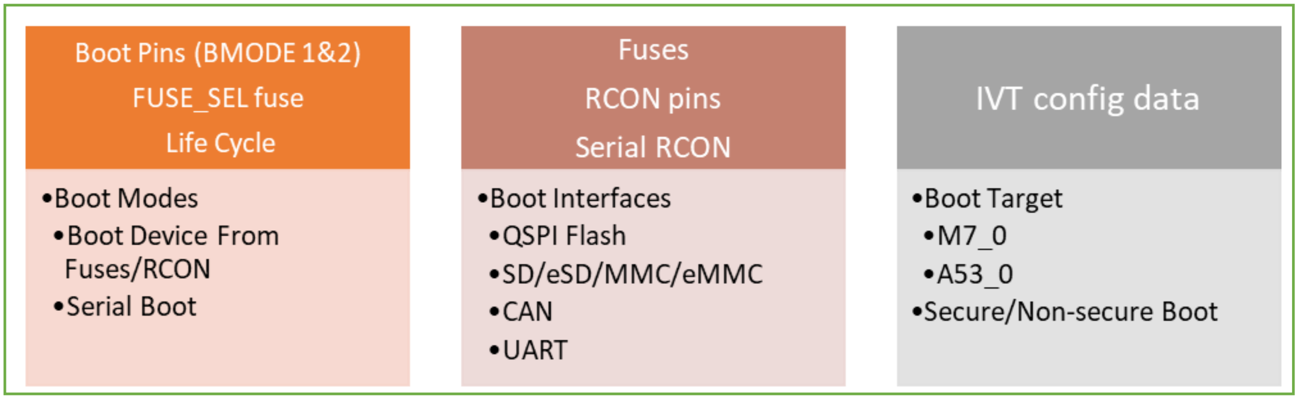


Figure 4 Boot selection Target, Interfaces and Mode

6.1. BootROM clock configuration

FIRC is the default system clock after Reset. BootROM configures CORE_PLL DFS (targeting 400 MHz) over FIRC. In case PLL lock is not achieved, BootROM continues with FIRC as the system clock.

In case of non-secure boot, BootROM configures the system clock to FIRC before handing the control to the application code.

In case of secure boot, BootROM passes control to HSE FW with system clock as CORE_PLL DFS. The HSE FW does not change any clock settings before handing over the control to application.

Refer to S32G3_BOOT_Settings.xlsx attached with the device RM for more details on clock and other peripheral configurations touched by BootROM.

6.2. Image authentication and decryption during Secure boot

BootROM has the responsibility to authenticate, decrypt and load HSM Firmware when performing a secure boot operation. The authentication scheme followed by BootROM to accomplish secure boot is shown in the following table.

Table 10 Authentication Scheme

Image	Authentication Scheme
IVT	AES-256 GCM
DCD	AES-256 GCM
Self-Test DCD	AES-256 GCM
Serial Application Image	RSA Signature Verification
HSE Firmware Image	AES-256 GCM

6.3. BootROM fail safe

Any kind of exception occurring during boot phase would result in BootROM issuing a functional reset. If the number of successive functional reset get to a value of 8 or beyond, BootROM enters Serial Boot mode.

Care must be taken by the application code to monitor and clear the reset sources issued by the application. Failure to do so may result in BootROM incorrectly identifying these resets as ones generated during boot phase and may result in BootROM forcing the system to Secure boot mode even when the application boot was possible.

6.4. BootROM at STANDBY exit

BootROM supports booting from STANDBY IVT or full boot at every STANDBY mode exit. The decision to boot either from STANDBY IVT or full boot is taken based on the wakeup source and its configuration in WKPU_WBMSR register. Full boot configuration is similar to device coming out of Reset.

STANDBY IVT has the same format as full boot/Reset IVT except that it’s placed at the first location of STANDBY RAM, i.e. 0x24000000. Only DCD and application boot are supported, and no data fetch from external NVM is supported. Also, secure boot is not supported in case of STANDBY IVT. All pointers in STANDBY IVT must point to addresses within STANDBY RAM.

In case more than one wakeup sources are latched, full boot is performed if any of the wakeups is configured for full boot.

7. Boot from external NVM

7.1. Required hardware configuration

Boot mode and RCON pins need to be configured for BootROM to understand and configure the required boot device.

Table 11 Boot Mode configuration

	FUSE_SEL	BMODE0	BMODE1	Boot Mode
Development phase	0	1	0	Boot from external memory using RCON configuration
Production phase	1	0	x	Boot from external memory using Fuse config

Configuration required on S32G-PROCEVB-S for boot from external NVM (SD/eMMC/QSPI):

1. SW14 -> 1 ON, SW14 -> 2 OFF
2. SW15 -> 1 OFF SW15- > 2 OFF

BOOT_CFG1 (using RCON or fuse depending on FUSE_SEL), BOOT_CFG2 and BOOT_CFG3 example settings are provided in the attachment.

7.2. QSPI Boot

BootROM performs QSPI controller configuration in two phases:

1) Initial configuration phase: Configure QSPI controller at 30 MHz in 1-bit mode for Quad and Octal flash and 8-bit mode for Hyperflash. Read user-provided Flash reconfiguration data from offset 0x200 of the flash.
2) Final configuration phase: Program QSPI clock source and QSPI controller as per details provided in reconfiguration data. Example reconfiguration binaries for some external flash devices is provided with the latest S32DS release at path (installation directory)\eclipse\mcu_data\processors\S32G274A_Rev2\PlatformSDK_S32G_2020_12\quadspi\default_boot_images. Same reconfiguration parameters can be used on S32G2 and S32G3.

7.2.1. Clock configuration

In Initial configuration phase, BootROM clocking scheme for QSPI operations depends upon PLL lock status. BootROM tries to lock the PERIPH PLL and PERIPH DFS1 for generating QSPI_1X_CLK of 30 MHz. If PLL-DFS are locked successfully at 30 MHz then clock is switched to PLL otherwise, QSPI clock is derived from FIRC, resulting QSPI_1X_CLK as FIRC/2.

In Final configuration phase, QSPI read operations clock requirements are derived by clock provided by user in QSPI reconfiguration data. In case of failure to generate the required clock, BootROM skips the configuration provided and continues read operations with default configuration and QSPI clock derived from FIRC.

BootROM sets a timeout of 500ms when downloading the application image. User must ensure that the first application image can be downloaded within this time. The image size would depend on the configurations used. For example, when configured in 40 MHz and 1x SDR mode, only application images with length less than 2 MB can be successfully downloaded by BootROM.

7.3. SD/MMC Boot

Only 3.3 V High Speed and Default Speed data rate operations are supported while booting from SD cards. BootROM does not support switching to 1.8 V for SDR modes.

NOTE

For application boot via the µSDHC interface, when BOOT_SEQ is set as 0, the RAM start pointer for the application should not point between 34008000h to 34008200. This address range is used by BootROM for internal operation during boot via the µSDHC interface. BootROM also uses 8 KB of SRAM memory starting at 34000000h for ADMA descriptors in case of µSDHC boot. The Application boot image header should not point to this location in case of µSDHC boot.

7.3.1. Clock configuration

BootROM tries to lock the Peripheral PLL-DFS and in case of success, switches the SDHC_CLK from FIRC to PLL.

7.3.2. µSDHC supported data rates

Table 12 µSDHC supported data rates

Speed Modes	Max Baud Rate	Clock Speed	Signal Voltage
Identification Speed	NA	~380 KHz	3V3
Default Speed	12.5 MB/s	25 MHz	3V3
High Speed	25 MB/s	50 MHz	3V3

BOOT_CFG1[19] controls selection of Default Speed or High Speed mode for SD interface.

7.3.3. MMC supported data rates

Table 13 MMC supported data rates

Speed Modes	Max Baud Rate	Clock Speed	Bus Width
Normal Speed	25 MB/s	25 MHz	1, 4, 8 bits
High Speed	50 MB/s	50 MHz	1, 4, 8 bits
High Speed DDR	100 MB/s	50 MHz	4, 8 bits

BOOT_CFG1[22:19] controls selection of Speed modes when MMC/eMMC card is used.

7.4. RCON/Boot_CFG settings

The example recommendation of RCONs for EVB is provided as an attachment to this document. Application developers must adapt these recommendations for their hardware.

8. Serial Boot

Serial Boot mode enables application code to be downloaded to SRAM through serial peripherals. The two primary use cases for serial boot are:

- End-of-line flash memory and fuse programming
- Recovery of chips that fail to boot correctly (nonresponsive modules)

This feature can be disabled by blowing fuse DIS_SER_BOOT.

Serial Boot mode is entered via the BMODE input pins. The device also enters serial boot mode if the Functional reset counter reaches a value >= 8. The functional reset may be result of any of the errors encountered during BootROM execution for example, software programming errors during development, flash memory failure, PCB failure during production, timeout due to failure of hardware modules, authentication failure during secure boot, exceptions during BootROM execution, unavailability of a valid boot image in the selected NVM and so on.

In Serial Boot mode, BootROM continuously polls for activity on any of the available interfaces:

- CAN
- LIN (UART)

In CUST_DEL life cycle, BootROM does not configure any watchdog when polling on serial interfaces. However for life cycles OEM_PROD and IN_FIELD, HSE SWT is activated with a timeout of 60 seconds.

The application should also specially take care that once serial download is complete, BootROM enables the SWT_0 with its default configuration. It is now the responsibility of the application to take care of servicing the watchdog.

Refer to S32G3 Reference Manual for further details.

9. Frequently Asked Questions (FAQ)

Ques. Which QSPI flash are supported?

Ans. S32G3 family of devices support all standard QSPI flash devices. Specific requirement from the boot side is that the Octal/Quad flash being used must be available in 1x mode after reset. This is the configuration used by BootROM for initial communication over the QSPI interface. This also means that the RESET_B line of S32G3 should be connected to flash reset to guarantee availability of flash in 1x mode at every boot.

Ques. Are QSPI reconfiguration parameters a mandatory requirement for QSPI boot?

Ans. No, the reconfiguration parameters are not mandatory and the device can perform a slower boot without these. The reconfiguration parameters are required to optimize the QSPI interface as per the external flash device. BootROM starts the QSPI accesses at 30MHz, 1x SDR mode which would be very slow for most of the flash devices. Reconfiguration parameters can be used in such cases to provide configuration inputs to BootROM which help fasten up the interface.

Ques. The default speed of QSPI interface has changed from 40 Mhz in S32G2 to 30 Mhz in S32G3. What is the impact?

Ans. This change was done to accommodate more range of flash devices. It does not have a materialistic impact on the boot numbers for most application use cases. This is because the default speed is used by BootROM only to read the reconfiguration parameters from the flash. Thereafter the QSPI interface is reconfigured as per application requirement. Hence from the overall boot time perspective, this would be a miniscule factor.

Ques. I have programmed the QSPI flash correctly and have also verified the contents by same mechanism as used in flashing. But the device still does not boot, why?

Ans. Check the RCON configurations to ensure they are aligned with the external flash component used. Also reverify the orientation/endianness of the data written in flash. Ensure that byte swap is not enabled when writing to the flash.

Ques. How can one get the reconfiguration parameters for their flash?

Ans. NXP provides example reconfiguration parameters with the S32 Design Studio release. These can be found at path – (installation path)\S32DS.3.4\eclipse\mcu_data\processors\S32G274A_Rev2\PlatformSDK_S32XX_2021_05\quadspi\default_boot_images

The processor name and SDK/RTD version may change depending on the latest versions being used. The user can take these as a reference and create the parameters for their application specific use cases. NXP provides these scripts for reference only.

Ques. Where can I get more details on Secure boot?

Ans. Secure boot is covered in HSE manuals.

Ques. What are the most common use cases of DCD during boot?

Ans. This would be a very application specific requirement. However some of the commonly seen use case include initializing the complete SRAM area using DCD.

Ques. How can the parameter QuadSPI POR delay in RCON be used for QSPI boot?

Ans. The POR delay parameter is to allow external flash to stabilize after power ON. Generally this parameter is specified as tVSL in the flash component Data Sheet. Usually the parameter is well within the limits of Boot delays (~3.5ms – 4ms) and hence no separate consideration is required. But in case, a flash device has an unusually large delay requirement than it should be provided via RCON. It should also be noted that the boot delay provided above is from the S32G3 reset deasserting. Any implication of using an external regulator to supply the flash should be taken up during this analysis.

Ques. Are any FXOSC related parameters in RCON required to be programmed during external NVM boot.

Ans. BootROM uses FIRC based PLL for all NVM boot modes. FXOSC is only configured by BootROM during serial boot. Hence the FXOSC settings are not required to be provided during external NVM boot.

Ques. When QSPI interface is reconfigured to 200 MHz, and the interface clock is observed it seems to be less than 200 MHz. What can be going wrong?

Ans. BootROM configures PLL over FIRC. To accommodate for any FIRC variations (+/- 5%) and remove any possibility of a overshoot in terms of allowed PLL spec, all PLL calculations done by BootROM are based on the upper end of FIRC value (48+5%). As such in typical scenarios a variance of approx. 5% can be observed on the final generated clock. This may extend to approx. 10% in case the FIRC for a particular sample is towards the lower end of FIRC spec (48 – 5%).

10. References

- S32G3HDG – S32G3 Hardware Design Guidelines
- S32G3 Reference Manual
- S32G3 Data Sheet
- S32G-VNP-PROCEVB Schematic
- S32G-VNP-PROCEVB3 Schematic

11. Revision history

This section documents the changes done in this document.

Revision No.	**Release Date**	**Changes**
0	11/2021	• Initial release
1	02/2023	• Updated the Table3. • Updated the Table6.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an “as is” and “with all faults” basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer’s exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Suitability for use in automotive and/or industrial applications — This NXP product has been qualified for use in automotive and/or industrial applications. It has been developed in accordance with ISO 26262 respectively IEC 61508, and has been ASIL- respectively SIL-classified accordingly. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as “Critical Applications”), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys’ fees) that NXP may incur related to customer’s incorporation of any product in a Critical Application.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

Altivec — is a trademark of NXP B.V.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

CoolFlux — is a trademark of NXP B.V.

CoolFlux DSP — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

Embrace — is a trademark of NXP B.V.

Freescale — is a trademark of NXP B.V.

GreenChip — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

JCOP — is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

MagniV — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MCCI — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MIFARE Classic — is a trademark of NXP B.V.

MIFARE FleX — is a trademark of NXP B.V.

MIFARE4Mobile — is a trademark of NXP B.V.

MIFARE Plus — is a trademark of NXP B.V.

MIFARE Ultralight — is a trademark of NXP B.V.

MiGLO — is a trademark of NXP B.V.

MOBILEGT — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

NXP SECURE CONNECTIONS FOR A SMARTER WORLD — is a trademark of NXP B.V.

PEG — is a trademark of NXP B.V.

Plus X — is a trademark of NXP B.V.

POR — is a trademark of NXP B.V.

PowerQUICC — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

QorIQ Qonverge — is a trademark of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

SmartLX — is a trademark of NXP B.V.

SmartMX — is a trademark of NXP B.V.

StarCore — is a trademark of NXP B.V.

Symphony — is a trademark of NXP B.V.

Synopsys & Designware — are registered trademarks of Synopsys, Inc.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

TriMedia — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

Vybrid — is a trademark of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 02/2023
Document identifier: AN13456