NXP Semiconductors
Application Notes

Document Number: AN13727
Rev. 1.1, 02/2023

Flashing Binaries to S32G-VNP-RDB3 Board

Contents

1. Introduction

This application note provides detailed procedures for flashing binary images to the S32G-VNP-RDB3 board. The S32G399A that is used on S32G-VNP-RDB3 supports two boot modes: Serial Boot mode and Boot from external flash memory (from QuadSPI flash, SD, or eMMC). In this document, we will show how to flash binary images to external flash and how to update the firmware of the ethernet switch SJA1110.

The descriptions in this document can help readers get familiar with the binary image programming of S32G-VNP-RDB3 board and provide a reference method that can be used on the customer's board.

2. Flashing binaries to external flash memory of S32G

2.1. Overview

In this section, the details of flashing binaries to external flash memories are described.

The BootROM of the S32G399A supports booting from external flash memory devices over the following interfaces:

- QuadSPI
- SD/MMC/eMMC via µSDHC interface

On the S32G-VNP-RDB3, both boot interfaces are supported. One 64MB octal flash memory MX25UW51245G is connected to the QuadSPI A interface. One 32GB eMMC device and one SD card slot are multiplexed and connected to µSDHC interface. Users can select to connect the SD card or eMMC to the S32G via a dip switch SW3. When SW3 is in the "ON" status, the S32G is connected to SD card. When SW3 is in the "OFF" status, the S32G is connected to the eMMC device.

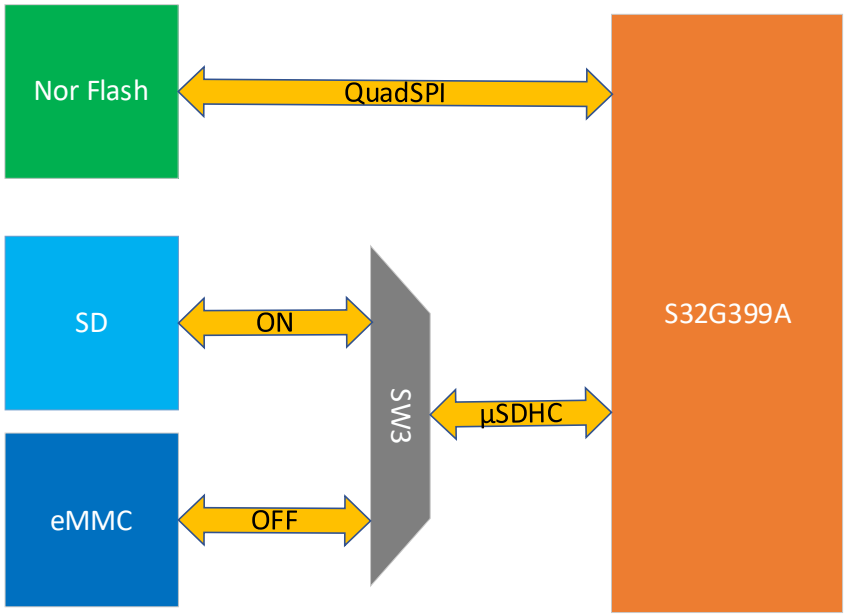


Figure 1. Diagram of boot interfaces from external flash

NOTE

At the time of writing, the descriptions of the BSP image bindings are applicable to NXP S32G BSP version 35.0. The structure of the image may change in subsequent versions of associated NXP Linux BSP releases.

ATF boot flow support enabled by default(BSP35.0) – U-boot can be used as BL33 only.

2.2. Flashing binaries with S32 Flash tool

There are two ways that user can use the flash tool:

First, the control button “Flash Image” can be found in “ConfigTool -> IVT” panel.

Second, use “S32 Flash Tool”. S32 Flash Tool is distributed with the S32 Design Studio installation package. User can find this tool under the installing folder of S32DS, for example, the default installation path is “C:\NXP\S32DS.3.5<or newer version>\S32DS\tools\ S32FlashTool”.

S32 Flash Tool is suitable for the first time assembled or manufactured board that does not include a pluggable SD card interface, in this scenario, the user can program the image to the external flash through the serial port.

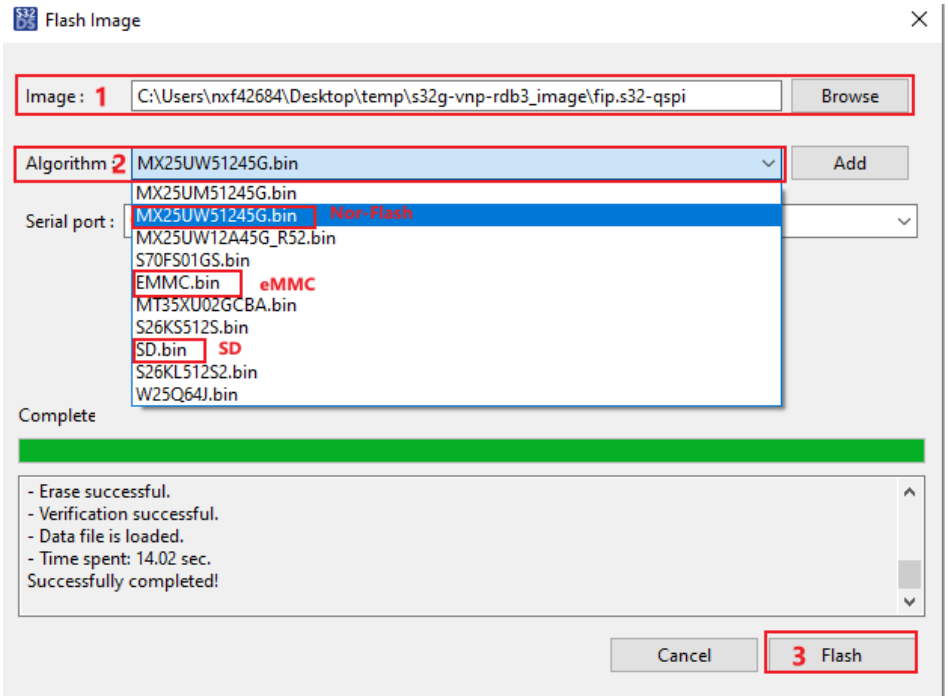
The S32 Flash Tool offers two ways to write to the flash, the detailed operating steps can be found in the "S32 Flash Tool User Guide".

- Using command line interface
- Using graphical user interface

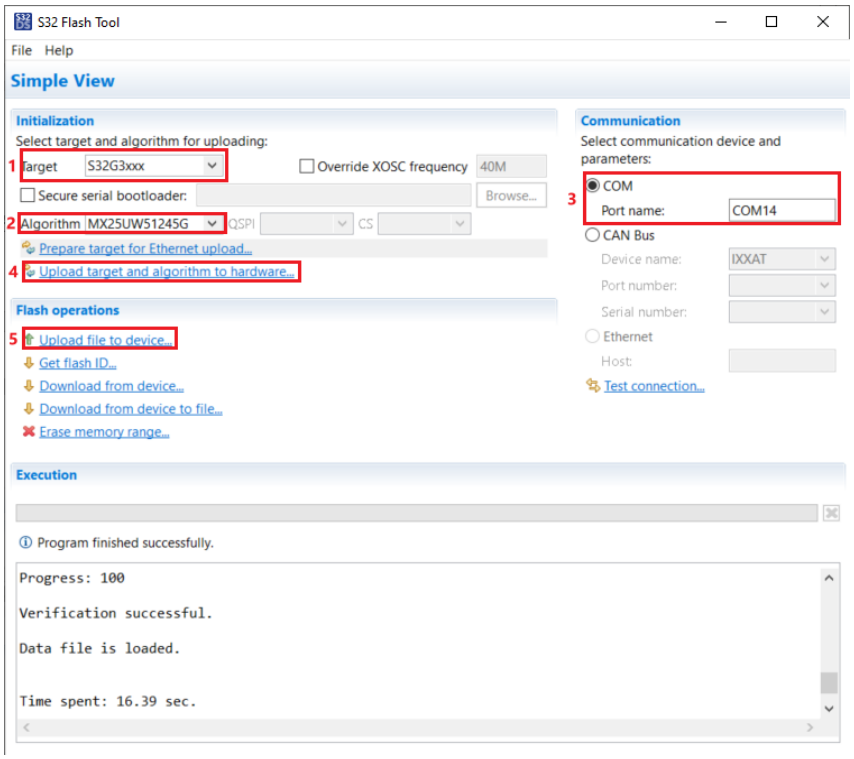
Step1. Connect UART0 to PC and set S32G-VNP-RDB3 to serial boot mode "Boot Mode Configuration".

Setp2. Flashing the image to the external memory.

Method 1. Flash Image from “ConfigTool -> IVT” panel.



Method 2. Using graphical user interface of S32 Flash tool.



Step3. Power off the board and configure the device to boot from the corresponding external flash "Boot Mode Configuration".

NOTE

The algorithm file used by QSPI-Flash on the S32G-VNP-RDB3 is MX25UW51245G.bin.

Please refer to the document ”S32G3_Fuse_Map_Tables.xlsx” in the S32G3 Reference Manual’s attachment for more information about the boot mode configuration.

2.3. Flashing binaries with U-Boot

In this section, we are going to use the pre-built binaries of Linux BSP35 and writing to the SD card via PC. Then boot from the SD card and program the image to eMMC and QSPI Flash through u-boot commands.

Compared with the serial downloading method of S32 Flash Tool, this method greatly reduces the time of programming large files such as the BSP image to external memory.

2.3.1. Prepare and write BSP binaries image to SD card

Step1. Insert the SD card to the Linux machine (eg: ubuntu) via SD card reader.

Step2. Identify the device node assigned to the SD card, enter the command:

```
ls /dev/sd*
```

```
/dev/sda  /dev/sda1  /dev/sdb  /dev/sdb1  /dev/sdb2
```

In this example it is assumed that the device assigned is /dev/sdb.

NOTE

Make sure the device node is correct for the SD card! Otherwise, it may damage your operating system or data or your PC.

Step3. Program the comprehensive Yocto Image " .sdcard" (after successfully building Yocto, look for build result in <build directory>/tmp/deploy/images/s32g399ardb3) to SD card.

```
sudo dd if=fsl-image-auto-s32g399ardb3.sdcard of=/dev/sdb bs=1M && sync
```

NOTE

Win32DiskImager can be used on the windows PC to write BSP image to SD card.

2.3.2. Flashing image into eMMC via u-boot

Control commands for SD/eMMC have been integrated in u-boot. User can get more information by entering the following command after u-boot starts on the board.

```
=> mmc help
mmc - MMC sub system

Usage:
mmc info - display info of the current MMC device
mmc read addr blk# cnt
mmc write addr blk# cnt
mmc erase blk# cnt
mmc rescan
mmc part - lists available partition on current mmc device
mmc dev [dev] [part] - show or set current mmc device [partition]
mmc list - lists available devices
mmc hwpartition [args...] - does hardware partitioning
  arguments (sizes in 512-byte blocks):
    [user [enh start cnt] [wrrel {on|off}]] - sets user data area attributes
    [gp1|gp2|gp3|gp4 cnt [enh] [wrrel {on|off}]] - general purpose partition
    [check|set|complete] - mode, complete set partitioning completed
  WARNING: Partitioning is a write-once setting once it is set to complete.
  Power cycling is required to initialize partitions after set to complete.
mmc setdsr <value> - set DSR register value
```

- **Write fip.s32-sdcard to eMMC:**

The below steps will load u-boot binary from the FAT32 partition of SD card.

Step1. Copy fip.s32-sdcard image from PC to the SD card's FAT32 partition. Alternatively, you can also load the image into DDR via the "tftp" command (Please refer to the section 2.4), and then write it to external memory.

Step2. Configure the switches to boot from the SD card "Boot Mode Configuration".

Step3. Power on the board and load the fip.s32-sdcard image into DDR.

```
=>fatload mmc 0:1 90000000 fip.s32-sdcard
1050080 bytes read in 50 ms (17.2 MiB/s)
```

Step4. Set SW3 to OFF, the S32G399A is connected to the eMMC card.

Step5. Write the image from DDR to eMMC

```
=>mmc rescan
=>mmc write 90000000 0 803
```

Calculate count of eMMC blocks needed for the loaded image:

i. cnt = filesize/512 + ((filesize%512 == 0) ? 0 : 1)

ii. Convert the value of cnt to hexadecimal

Eg: file size=1050080, 1050080/512=2050.93, cnt=2050+1=0x803

Step6. After setting the switches to boot from eMMC "Boot Mode Configuration", perform a power on reset of the board and verify.

■ Write fsl-image-auto-s32g399ardb3.sdcard to eMMC

The below steps will load the full fsl-image-auto-s32g399ardb3.sdcard image from ext3 partition, and write it to eMMC.

Step1. Insert the SD card into a Linux machine via SD card reader. And to create a new partition for SD card.

```
$ sudo fdisk /dev/sdb

Command (m for help): p
***
Device     Boot Start    End Sectors  Size Id Type
/dev/sdb1       8192  139263   131072   64M  c W95 FAT32 (LBA)
/dev/sdb2     139264 901119   761846  372M 83 Linux

Command (m for help): n
Partition type
  p   primary (2 primary, 0 extended, 2 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (3,4, default 3):
First sector (2048-60432383, default 2048): 901120
Last sector, +sectors or +size{K,M,G,T,P} (901120-60432383, default 60432383): +2G
```

```
Created a new partition 3 of type 'Linux' and of size 2 GiB.

***
Command (m for help): w
***
```

Format the newly created partition:

```
$sudo mkfs.ext3 -L temp /dev/sdb3
```

Step2. Copy fsl-image-auto-s32g399ardb3.sdcard from PC to SD card's new partition (Because the size of .sdcard is outside of default partition range).

Step3. Configure the switches to boot from the SD card "Boot Mode Configuration".

Step4. Power on the board and load the fsl-image-auto-s32g399ardb3.sdcard image into DDR.

```
=> mmc part

Partition Map for MMC device 0  --   Partition Type: DOS

Part    Start Sector    Num Sectors     UUID            Type
  1     8192            131072          036cb08f-01     0c
  2     139264          761856          036cb08f-02     83
  3     901120          4194304         036cb08f-03     83

=> ext4ls mmc 0:3
<DIR>       4096 .
<DIR>       4096 ..
<DIR>      16384 lost+found
       658505728 fsl-image-auto-s32g399ardb3.sdcard
=> ext4load mmc 0:3 90000000 fsl-image-auto-s32g399ardb3.sdcard
658505728 mmcbytes read in 24247 ms (22.9 MiB/s)
```

Step5. Set SW3 to OFF, the S32G is connected to the eMMC card.

Step6. Write the image from DDR to eMMC

```
=>mmc rescan
=>mmc write 90000000 0 13A000
```

Step7. After setting the switches to boot from eMMC "Boot Mode Configuration", perform a power on reset of the board and verify.

2.3.3. Flashing image into QSPI Flash via u-boot

Step1. Copy fsl-image-flash-s32g399ardb3.flashimage from PC to the SD card's new partition.

Step2. Configure the switches to boot from the SD card "Boot Mode Configuration".

Step3. Power on the board and load the fsl-image-flash-s32g399ardb3.flashimage into DDR.

```
=> mmc part

Partition Map for MMC device 0  --   Partition Type: DOS
```

```
Part    Start Sector    Num Sectors    UUID          Type
1       8192            131072         036cb08f-01   0c
 2      139264          761856         036cb08f-02   83
 3      901120          4194304        036cb08f-03   83

=> ext4ls mmc 0:3
<DIR>       4096 .
<DIR>       4096 ..
     658505728 fsl-image-auto-s32g399ardb3.sdcard
      67108864 fsl-image-flash-s32g399ardb3.flashimage
```

1. Prepare flash environment

```
=>run flashbootargs
```

2. Load QSPI Driver

```
=>sf probe 6:0
SF: Detected mx25uw51245g with page size 256 Bytes, erase size 64 KiB, total 64 MiB
```

3. Update u-boot parameters.

```
=>setenv image fsl-image-flash-s32g399ardb3.flashimage
```

4. Load image into DDR

```
=> setenv loadaddr 0x85000000
=>ext4load mmc 0:3 ${loadaddr} ${image}
67108864 bytes read in 2797 ms (22.9 MiB/s)
```

Step5. Write the image from DDR into QSPI Flash

```
=> sf erase ${uboot_flashaddr} +${filesize}
SF: 67108864 bytes @ 0x0 Erased: OK
=> sf write ${loadaddr} ${uboot_flashaddr} ${filesize}
device 0 whole chip
SF: 67108864 bytes @ 0x0 Written: OK
```

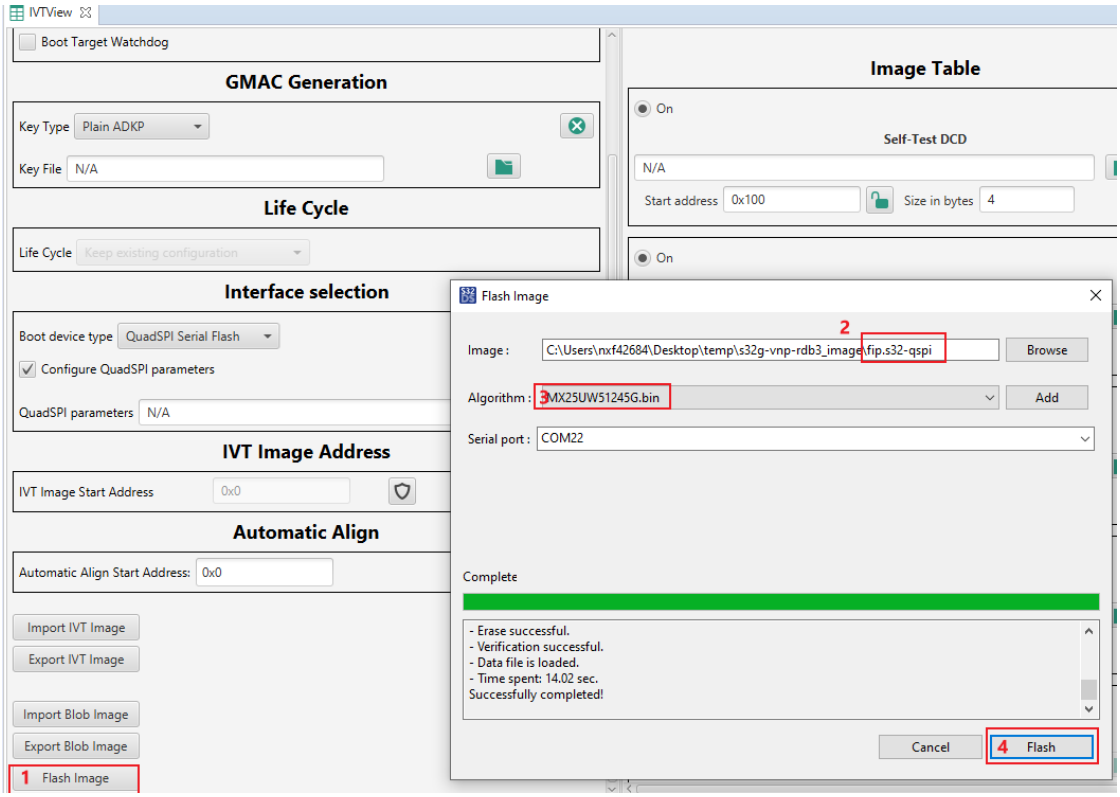
Step6. After setting the switches to boot from QSPI Flash "Boot Mode Configuration", perform a power on reset of the board and verify.

2.4. Flashing image with SD card-less

The following will demonstrate the process of programming the image “fip.s32-qspi” to QSPI-Flash with ”Flash Tool” and then using u-boot’s command “tftp” to flashing the image “fsl-image-auto-s32g399ardb3.sdcard” to eMMC.

Step1. Connect UART0 to PC and set S32G-VNP-RDB3 to serial boot mode "Boot Mode Configuration".

Setp2. Flashing the image “fip.s32-qspi” to QSPI-Flash.



Step3. Configure the switches to boot from the QSPI-Flash"Boot Mode Configuration".

Step4. Connected GMAC port and TFTP server via network cable.

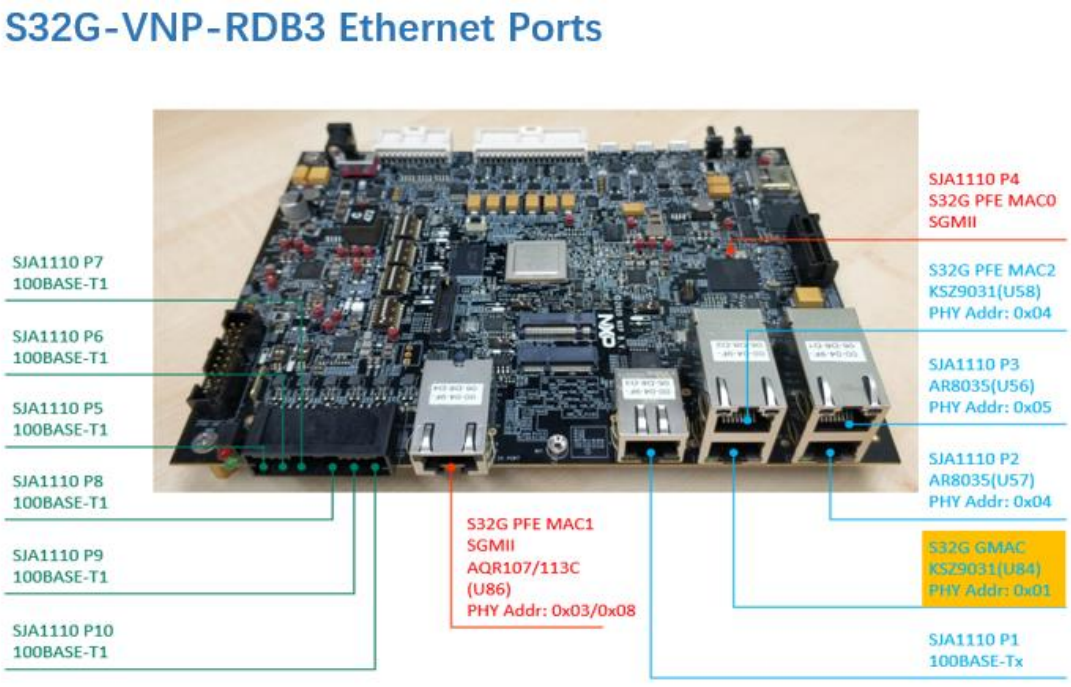


Figure 2. Diagram of S32G-VNP-RDB3 Ethernet ports

Step5. Power on the board and stops the program at the u-boot command line. Configure the TFTP client environment.

a) To load Image using TFTP it required to setup TFTP server and to do the TFTP client settings in u-boot to be initialized. The instructions to setup a TFTP server are outside the scope of this document.

b) Change the environment variable ethact to eth_eqos.

```
=> setenv ethact eth_eqos
```

c) Set Ip address of ipaddr and serverip

```
=> setenv ipaddr 10.193.248.207
=> setenv serverip 10.193.248.72
=> ping 10.193.248.72
Using eth_eqos device
host 10.193.248.72 is alive
```

Setp6. Loading image from TFTP server to DDR.

```
=>tftp 90000000 fsl-image-auto-s32g399ardb3.sdcard
          ###################################################################
          ...###############
          3.9 MiB/s
done
Bytes transferred = 658505728 (27400000 hex)
```

Step7. Set SW3 to OFF, the S32G is connected to the eMMC card.

Step8. Write the image from DDR to eMMC

```
=>mmc rescan
=>mmc write 90000000 13A000
```

Step9. After setting the switches to boot from eMMC "Boot Mode Configuration",perform a power on reset of the board and verify.

NOTE

The IP addresses are used for demo. User should change them according to the network they are using.

Different configurations lead to different sizes of images generated by BSP35, which are subject to your actual conditions.

3. Update firmware to peripheral devices of the RDB3

3.1. Overview

There are two peripheral devices on the board that need to have programmed firmware before they can work correctly. The switch SJA1110 and ethernet PHY AQR113C. Both are programmed with the latest firmware at the time of assembly.

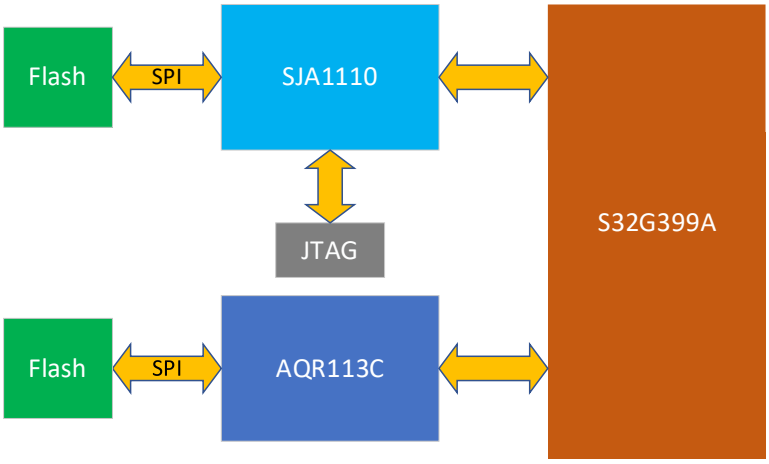


Figure3. Interface of SJA1110 and AQR113C diagram

NOTE

This chapter does not describe how to update the firmware of PHY AQR113C device. For more information about FW of AQR113C, please request from Aquantia https://portal.aquantia.com/user .

3.2. Quick start for SJA1110

NDA is required for the SJA1110 SDK activation code and some of the documentation mentioned below, and users need to apply for permission from sales/FAE/marketing teams.

3.2.1. Boot mode of SJA1110

The SJA1110 support different boot options, selectable via a jumper(J189).

Table 1. SJA1110 boot operations

Pin1-2	Pin3-4	Boot Option	NVM Device Type
OPEN	OPEN	NVM Boot	Reserved
OPEN	SHORT(default)		SPI Flash
SHORT	OPEN		Reserved
SHORT	SHORT	SDL Boot	N/A

Details on the boot options can be found in “UM11107 Software user manual for SJA1110”.

When using the internal microcontroller, it can boot from (cf. Bootloader chapter of UM11107):

- SJA1110 boots from flash (NVM boot)
- SPI Bootloader / SDL Boot

There is also the possibility to not use the internal microcontroller an only provide the switch core configuration (via SPI). Details on this can be found in Switch subsystem chapter of UM11107. For best experience with the S32G-VNP-RDB3 we suggest using the internal microcontroller of the SJA1110.

When using the internal microcontroller, the software image are typically generated using the SJA1110 SDK.

3.2.2. Install SJA1110 SDK

Most recent install instruction can be found in the SJA1110 documentation.

https://www.nxp.com/document/guide/get-started-with-the-sja1110-evm:GS-SJA1110-EVM

Install S32 Design Studio for Arm + SDK

1. Download and install S32 Design Studio S32DS 3.5 for ARM.
   a) Log in to nxp.com.
   b) Search: “S32DS 3.5”.
2. The license key for SDK activation is included in the box.
3. Download and install SDK.
   a) Enter license key when prompted.
   b) Download SJA1110 SDK file (SJA1110 SDK RTM v1.0.0 or newer).
   c) Open S32 Design Studio and click Help → Update Software.
   d) Select from Archive and browse to SDK-zip-file.
   e) Accept license and install.

Example Design for RDB3

The SJA1110 SDK comes with an ready to use example for the S32G-VNP-RDB3.

For more reference design information, please refer to the guide “S32G-VNP-RDB3 Reference Design – Ethernet Enablement Guide” on the www.nxp.com.

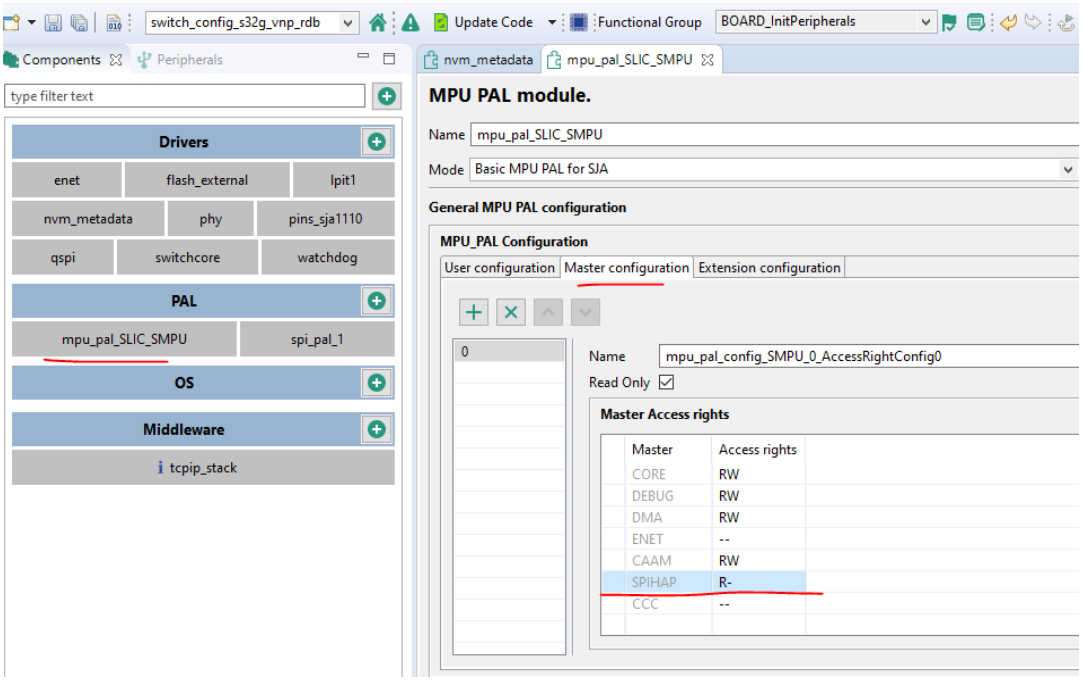
The example application configures the SJA1110 to operate as a simple L2 switch with the addition of an IP stack. The purpose of this design is to provide the user with an out-of-the box example application to enable the SJA1110 switch on the S32G-VNP-RDB3 board using SJA1110 SDK.

The following use cases are shown in this example application:

- Switch configuration (i.e. loading a static switch configuration) using the Ethernet Switch Core (SWITCH) driver.
- Initialization and management of PHYs using Ethernet PHY.
- Interaction between Ethernet Switch Core (SWITCH) and Ethernet PHY for auto-negotiation.

- lwIP stack integration.
- Periodic calling of main functions using Low Power Interrupt Timer (LPIT) driver.
- Protection of switch configuration access using SMPU using Memory Protection Unit Peripheral Abstraction Layer (MPU PAL).
- Supervision of execution with Software Watchdog Timer using Software Watchdog Timer (SWT).
- Firmware update via TFTP.

The document of the example can be found in the S32 SDK documentation at Examples and Demos section (<SDK_PATH>/doc/Start_Here.html).



3.2.3. Programming the example binary into flash through the JTAG

Step1. Connect debugger (J44, SJA1110 JTAG header) and power supply. Supported debugger, e.g:

i. Lauterbach base probe with Cortex-M debug probe,

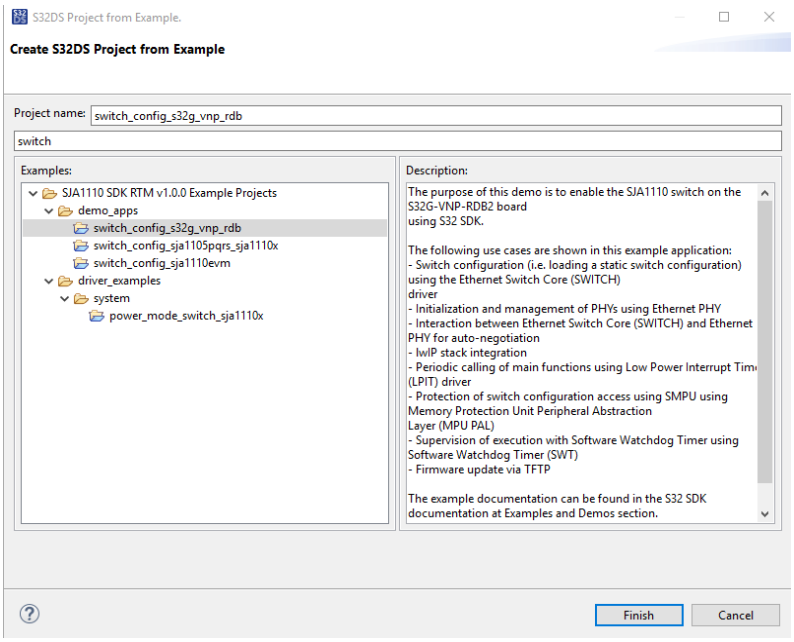
ii. Lauterbach uTrace for Cortex-M,

iii. Multilink Universal RevC/D, or

iv. Multilink Universal FX RevB/C.

Step2. Open S32DS.

Step3. Import Example Design. Click “New S32DS Project from Example”, then Select “switch_config_s32g_vnp_rdb” and click “Finish” .



Step4. Optional: Opening configuration views:

1. Click 'Open S32 Configuration'.
2. Click 'Peripherals' to view and adapt the switch related configuration.

Step5. Build the project:

1. Select the configuration to be built Flash (Debug_Flash) by left clicking on the downward arrow corresponding to the build button.
2. Wait for the build action to be completed before continuing to the next step.

Step6. Running the project:

1. Go to Run and select Debug Configurations. There will be four debug configurations for this project.
2. Select the desired debug configuration and click on Launch. Now the perspective will change to the Debug Perspective.
3. Use the controls to control the program flow.
4. Switch/board is now working according to example design.

More details and examples can be found in the SJA1110 Software user manual for SJA1110 (UM11107) and the S32SDK User Manual (SJA1110 EAR 0.9.0).

3.2.4. Programming a binary into flash through TFTP

If the running firmware of SJA1110 allows for it, the flash image can be updated via TFTP.

In the default example from the SDK, the SJA1110 firmware has a TFTP server running on address 192.168.0.200.

This IP can be changed via the SJA1110 SDK (e.g. to another IP, DHCP, or auto IP).

Step1. Prepare new image using the SJA1110 SDK (this typically generates a `flash_image.bin` file).

Step2. Connect to any port that can reach the SJA1110.

Step3. Send image via TFTP

=> tftp -i 192.168.0.200 put flash_image.bin flash.bin

4. References

Table2. The configuration of boot mode.

Switch	SD Boot Setting (default)	eMMC Boot Setting	NOR Flashing Boot Setting	Serial Boot Setting
SW3	ON	OFF	-	-
SW4	7-ON, REST-OFF	6,7-ON, REST-OFF	ALL-OFF	ALL-OFF
SW5	ALL-OFF	ALL-OFF	ALL-OFF	ALL-OFF
SW6	ALL-OFF	ALL-OFF	ALL-OFF	ALL-OFF
SW7	ALL-OFF	ALL-OFF	ALL-OFF	ALL-OFF
SW9	1-OFF, 2-OFF	1-OFF, 2-OFF	1-OFF, 2-OFF	1-OFF, 2-OFF
SW10	1-ON, 2-OFF	1-ON, 2-OFF	1-ON, 2-OFF	1-OFF, 2-OFF

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.

arm

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: AN13727