SBC-S32V234 QUICK START GUIDE (QSG)

Getting started instructions and a Guide to all Hardware, Software, Tools and Document resources Rev 6

www.nxp.com/SBC-S32V234







EXTERNAL USE

WHAT IS QUICK START GUIDE?

We at NXP continuously strive to provide an easy-to-use enablement package for our devices. This document serves as a launch-pad to navigate all essential resources that we provide to get started with S32V2 SoC and SBC-S32V234 board.

This document contains instructions on how to get started with the S32V2 vision processor, SBC-S32V234 board and associated software components.



SECURE CONNECTIONS FOR A SMARTER WORLD

Contents

- Overview of a complete Ecosystem
- Getting Started Instructions
 - Out-of-box experience: SBC-S32V234
 - Tutorials
- Resource Guide
 - Evaluation boards & Camera modules
 - OS and SDK
 - Drivers and other Software
 - IDE and Tools
 - Documentation
 - Support
 - Enablement Roadmap







GETTING STARTED



How To Get Started?

- This section will walk you through the steps of "how to get started?"
 - We will start with the simple out-of-box experience and move forward to gain understanding of different pieces of HW & SW one by one



Out-of-box experience: SBC-S32V234

The SBC-S32V234 is a low-cost development platform for the S32V2 vision processor.

SBC-S32V234 is a systemon-module concept developed by NXP partner, MicroSys Electronics GmbH. It is comprised of a Module board mounted on a Carrier board. The Module board contains S32V234 processor, memories and power regulators for the module board and the Carrier board contains all system specific i/o like camera connectors, ethernet port, display port, SD card slot, CAN ports etc. This design gives customers flexibility to design their own carrier board while still reusing the same module board.







Out-of-box experience: SBC-S32V234

2 of 4



Connect Power Supply and USB Cable

Connect board with 12V/2A DC power supply. Keep the board power off.

Connect one end of the USB cable to the PC and the other end to the micro-B connector on the SBC-S32V234 board. Allow the PC to automatically configure the USB drivers.

Install and Start Serial Terminal

1

2

3

4

Install and open any serial terminal (e.g. TeraTerm, Putty etc.). Select the port where board is connected and configure it to 115200 baud rate, 8 data bits, no parity and 1 stop bit.

Insert the SD card and Power on the Device

The box contains SD card pre-installed with Linux BSP*. Please make sure it is inserted into SD card slot. Power on the device by pressing Power Button. You can now see SBC-S32V234 booting U-boot followed by Linux in the terminal window. Username: "root", no password

Learn More About the SBC-S32V234

Access more documents at <u>nxp.com/SBC-S32V234</u> nxp.com/S32V

Note: if your computer is not detecting the USB-serial device, download the FTDI drivers from here.

* Linux BSP in SD card may not be latest. Follow steps on next 2 pages to download the latest BSP

Out-of-box experience: Linux BSP

NXP's SW team has developed a Linux Board Support Package(BSP) NXP periodically updates its SW so users are encouraged to check following locations for latest BSP

www.nxp.com/s32v >> Software & Tools >> "Automotive SW – Linux BSP"

		Automotive SW - Linux BSP ^{(REV} FLEXERA 1 KB SW32XX-LINUXBSP0	16) UPDATED	2018-03-27 11:58:00		
+	File Description	\$	File Size 🗘	File Name	\$	
+	Aquantia firmware for the	BlueBox LS2084A	192.3 KB	▲ Aquantia_binaries.tgz		
+	Benchmark results		103.7 KB	Auto_Linux_BSP_19.0_Benchmark.pdf		
+	GPU userspace libraries	and VTK tools	89.8 MB	binaries_gpu_userspace_vtk_6.2.4.zip		
1 +	Precompiled binaries		736.4 MB	➡ binaries_auto_linux_bsp19.0.tgz		Download precompiled Linux binaries from he
) +	Precompiled binaries for	BlueBox	480.6 MB	binaries_auto_linux_bsp19.0_bluebox.tgz		
) +	Precompiled binaries wit	h VSDK	814.7 MB	binaries_auto_linux_bsp19.0_vsdk.tgz		
) +	Quality package		2.9 MB	Auto_Linux_BSP_19.0_Quality_Package.zip		
) +	Quick start guide		201.6 KB	Auto_Linux_BSP_19.0_BlueBox_Quick_Start.pdf		
) +	Quick start guide		3.4 MB	Auto_Linux_BSP_19.0_Quick_Start.pdf		
) +	Release notes		110.4 KB	Auto_Linux_BSP_19.0_Release_Notes.pdf		
) +	Software content reports		4.8 MB	Auto_Linux_BSP_19.0_BOM.zip		
) +	User manual		3.7 MB	Auto_Linux_BSP_19.0_User_Manual.pdf		Download Linux user manual from here



Out-of-box experience: Linux BSP

How to Install the latest BSP onto an SD card?

- Download latest binaries as shown on previous page
- Extract the compressed file
- All SBC-S32V234 specific files can be found in folder: *binary_auto_linux_bspxx.x\s32v234sbc*

fsl-image-auto-s32v234sbc.sdcard	11/25/2018 10:15	SDCARD File	344,064 KB
fsl-image-auto-s32v234sbc.tar.gz	11/25/2018 10:15	GZ File	82,831 KB
🗋 Image	11/25/2018 10:15	File	7,897 KB
Image-s32v234sbc.dtb	11/25/2018 10:15	DTB File	21 KB
u-boot-s32v234sbc	11/25/2018 10:15	S32 File	304 KB

• Use the .sdcard image to flash sd-card using tools like "Win32 Disk Imager"



Learn: Linux BSP

- Two most important software components for S32V2 devices are Linux and Vision SDK
- Linux runs on the S32V2 vision processor and supports various NXP development boards
- NXP uses the Yocto project to create a custom Linux BSP
- Refer to the *Linux user manual* for more information on customizing Linux using Yocto
- To understand different components of Linux BSP please read: 1a_understand_Linux_BSP.pdf
 (download Quick Start Package to locate this file)



Learn: Vision SDK

- Vision SDK is the collection of libraries that can be used to create Vision specific programs
- Includes libraries for
 - ISP Image Signal Processor
 - APEX Image Cognition Processor
 - IOs MIPI-CSI2, display etc...
- Also includes useful utilities and 3rd party software to aid development
 - e.g. OpenCV, FFmpeg, Pthreads etc
- Go to next page for Vision SDK download instructions
- To understand different components of Linux BSP please read: 1b_understand_Vision_SDK.pdf (download Quick Start Package to locate this file)



Learn: Vision SDK

Download Vision SDK...

www.nxp.com/s32v >> Software & Tools >> "Vision SDK SW"



Linux BSP &

Vision SDK

Develop: S32 Design Studio IDE ISP and APEX Projects

13

Prepare: Flash Linux BSP that supports Vision SDK 1 of 3

- Vision SDK is an essential part for vision application development
- The Linux BSP that comes with the SBC-S32V234 and downloaded as shown earlier are generic BSPs
 - It does not support Vision SDK out-of-the box.
 - So, the user needs to include device drivers, apply device tree file patches and include shared libraries to create a Linux BSP that supports VSDK
- For ease of use, NXP provides the precompiled Linux BSP binaries with all VSDK specific modifications
 - Please follow the instructions on next couple of pages to re-flash the sd-card.
 - Alternatively, user can also follow the "section 17 ADAS/Vision Yocto Layer" of the Linux user manual to create these binaries from scratch using Yocto



Prepare: Flash Linux BSP that supports Vision SDK 2 of 3

Use Linux Machine for this task

1. Download the pre-compiled binaries: <u>www.nxp.com/s32v</u> >> Software & Tools >> "Vision SDK SW"





Prepare: Flash Linux BSP that supports Vision SDK 3 of 3

2. Unzip the file and go to **build_content\v234_linux_build\s32v234sbc**



3. You can find all components of Linux BSP in **boot** and **rootfs** folders

Image	
s32v234sbc.dtb	rootfs-sbc.tar.gz
📗 u-boot	

- 4. Please follow the steps in 2_prepare_sd_card.pdf document to flash these files into an sd-card
- 5. Insert this sd-card into the board and turn on power supply.
- observe that the system is booting...



Run: Examples available in the Linux

- There are many precompiled demo applications from Vision SDK that are available in home/root/vsdk folder.
- In this step, we will run one of the demos available on the sd-card.
 - Refer Vision SDK User Guide for demo specific information.
- Run the most simple examples...
 - ISP : isp_somyimx224_csi_dcu.elf
 user needs to connect S32V-SONYCAM to MIPI-A port and a display(default 1080p) via HDMI port
 - APEX: apex_add.elf
 - ISP+APEX: apex_isp_fast9.elf

user needs to connect S32V-SONYCAM to MIPI-A port and a display(default 1080p) via HDMI port

root@s32v234evb:~/ apex_add.elf apex_downsample_up apex_emulation_tes apex_face_detectio apex_factg_cv.elf apex_gauss5x5_cv.e apex_intstogram_cv. apex_indirect_inpu	Vusdk# ls psample_cv.elf st.elf pn_cv.elf elf elf ut_cv.elf	apex_integral_image_ apex_isp_face_detect apex_isp_fast9.elf apex_isp_ldw_cv.elf apex_isp_segmentatio apex_orb_cv.elf apex_orb_cv.elf apexcv.elf	_cv.elf ion_cv.elf on.elf	apexcv_pro_affine.elf apexcv_pro_aggcf.elf apexcv_pro_blockmatcl apexcv_pro_brief.elf apexcv_pro_canny.elf apexcv_pro_fast.elf apexcv_pro_fft_corn apexcv_pro_hog.elf	f hing.elf ers.elf	apexcv_pro_hough.elf apexcv_pro_image_pyrami apexcv_pro_image_pyrami apexcv_pro_laplacian_py apexcv_pro_lbp.elf apexcv_pro_lkpyramid.el apexcv_pro_lktracker.el apexcv_pro_orb.elf	d.elf d_multi.elf ramid.elf f f	apexcv_pro_remap.elf apexcv_pro_tmo.elf data encoded_stream.h264 gdc_cppunitapp.elf hello.elf isp_h264dec_single_stream isp_jpeg_4stream.elf	i: i: i: i: n.elf i	sp_ov10635_viu_dcu.elf sp_ov10640_quad.elf sp_sonyimx224_csi_dcu.elf sp_sonyimx224_default.elf sp_sonyimx224_h264enc.elf sp_sonyimx224_rgb_yuv_gs8.el sp_stereo_apexbm.elf sp_stereo_calib.elf	multi_thread.elf neon_eigen_add.elf neon_fftw_cv.elf neon_gauss3x3_cv.e web_server.elf f
17	Out-of- experie	box ence	Lea Linux B Vision	rn: SP & SDK	Flas th V	<i>Prepare:</i> sh Linux BSP at supports ⁄ision SDK		<i>Run:</i> Run Vision SDK demos		Develo S32 Design S ISP and APE	<i>p:</i> tudio IDE (Projects

Run: Compile and Run Demos from VSDK

- Vision SDK also comes with many software examples
 - Face Detection demo
 - Lane departure warning demo
 - CNN image classifier demo
 - Feature tracking demo,
 - Stereo disparity demo
 - Pedestrian detection demo
 - Audio Video Bridging demo
 - many more.....

(visit: vsdk_installation_dir/s32v234_sdk/demos)

- This step will show the user how to compile and run these demo examples on the board
- Here, we will compile a simple ISP demo and run it on the SBC-S32V234 board
- Please read document
 <u>3_compile_demo_example_tutorial_x</u>
 <u>x.pdf</u> in the quick start package for detailed instructions.



Develop: S32 Design Studio IDE

1 of 2

- Now that we have our device up and running, the next step is to start the application development.
- To aid the developer, NXP provides no-cost IDE called <u>S32 Design Studio</u> (S32DS-NG)
 - use S32 Design Studio v3.1 or greater
 - Note: Starting from S32DS v3.1, S32DS does not support any device by default. Device support can only be added by S32DS Extensions and Update tools.
 - Note: <u>S32 Design Studio for Vision</u> is now archived and no longer recommended for new developments
- Features
 - an Eclipse based IDE
 - S32DS Extensions and Update tool to include support for latest
 - Vision SDK
 - Peripheral SDK and FreeROTS
 - Toolchains for ARM, APEX and ISP cores
 - Graph tools for easy ISP and APEX code development
 - Demos
- Let us set-up S32DS and get started with it



Develop: S32 Design Studio IDE

- In this part we will download, install and learn S32 Design Studio
- Please follow the steps below...
 - 1. Download and install **S32DS v3.1** or later from <u>here</u>
 - 2. Open S32DS and create a new workspace
 - 3. Include S32V2xx support using S32DS Extensions and Updates tool
 - Go to Help >> S32DS Extensions and Update
 - Select S32 Design Studio Platform Package, S32 Design Studio Platform Tools package, Vision extension package for S32V2xx and S32V2xx development package
 - Click on Install/Update to install selected packages
 - S32DS is now ready for S32V2xx application development
 - 4. To learn S32DS, go to *Help >> Getting Started* and go through the **Quick Start Guide** document.
 - To learn more about the S32 Design Studio please refer to the S32 Design Studio User Guide at Help >> Getting Started >> Resources >> User Manuals
 - 5. Now let us import, compile and run built-in example
 - Navigate to Getting Started Help >> Getting Started >> Resources >> Tutorials page and follow instructions of following "HOW TO" documents, one-by-one
 - 1. ISP: Create An ISP Project From Example in S32DS.pdf
 - 2. EVB: Build a Project and Setup a Debug Configuration for debugging in S32 Design Studio.pdf
 - 3. EVB: S32V234 EVB Linux Static IP address configuration.pdf / S32V234 EVB Linux DHCP IP address setup.pdf
 - 4. DEBUGGING: Setup A Remote Linux Connection in S32DS S32V234.pdf
 - 6. Congratulation! You have just compiled and ran your first program using S32DS
- Explore more Documents, Videos and Tutorials available in S32DS to learn more about S32DS

20	Out-of-box experience	<i>Prepare:</i> Flash Linux BSP that supports Vision SDK	Run	<i>Run:</i> Vision SDK demos	Develop: S32 Design Studio IDE ISP and APEX Projects	
----	--------------------------	---	-----	------------------------------------	--	--

Develop: Create ISP and APEX projects using Visual Graph Tool

- After becoming an Expert with S32DS, let us move forward to develop an application that uses ISP and APEX
 - APEX and ISP visual graph tools provide easy to use graphical interface to develop vision pipeline
 - Code generated from APEX and ISP graph tools can be used in any application developed with or without IDE
- Please follow the Tutorials below to learn "ISP and APEX program development using visual graph tools?" (both tutorials can be found in quick start package and S32DS)

4a_ISP_graph_tool_in-depth_tutorial.pdf

4b_APEX_graph_tool_in-depth_tutorial.pdf





Fore more understating on different Vision SDK components please start with the following documents...

- 1. VisionSDK_UserGuide (available at VSDK download site and at s32v234_sdk\docs\vsdk)
- 2. ISP: SDI_Software_User_Guide (s32v234_sdk\docs\drivers)
- 3. ISP: Sequencer_Driver_Software_User_Guide (s32v234_sdk\docs\drivers)
- 4. APEX: APEX_Programming_Overview (s32v234_sdk\docs\apex)
- 5. APEX: UG-10267-03-16-ACF_User_Guide (s32v234_sdk\docs\apex\acf)



RESOURCE GUIDE



Hardware: SBC-S32V234

- List of necessary documents and files for SBC-S32V234
- Part number of Processor: FS32V234CMN1VUB

Documents	Description
SBC-S32V234 User Guide	This document describes all components and connections on EVB
Module Schematic	Schematic for module board that mounts S32V234 processor, memories, control MCU etc. Only available via special request directly made to MicroSys.
Carrier Schematic	Schematic for carrier board that incorporates all peripherals like Ethernet, MIPI connectors, CANs, LINs etc

Download: www.nxp.com/sbc-s32v234 >> DOCUMENTATION / SOFTWARE & TOOLS



Hardware: Camera Modules

• List of supported image sensors & camera modules



Camera Module	Documents*	Description
S32V-SONYCAM		MIPI based camera with Sony IMX224 sensor. Connects directly with MIPI ports of the carrier board
OV10640CSP-S32V		MIPI based camera with OmniVision 10640 sensor. Connects directly with MIPI ports of the carrier board
MXOV10635-S32V	For Sensor: Contact Sensor manufacturer For Module schematics: Contact Module manufacturer**	OmniVision 10635 sensor based camera with Maxim serializer. Connects to the MAX9286S32V234 de-serializer board with coaxial cable with FAKRA (Jack-Jack) connectors***. The packaging does not include cable.
/AX9286S32V234		Maxim 9286 de-serializer board. Connects directly with MIPI ports of carrier board and can connects with up to 4 remote camera. Connects to the MXOV10635-S32V camera with coaxial cable with FAKRA (Jack- Jack) connectors***. The packaging does not include cable.

Buy from NXP: www.nxp.com/sbc-s32v234 >> BUY/PARAMETRICS Buy from Distribution Partners: Arrow, Avnet, Digi-Key, E14, Future or Mouser Buy from Maxim: maximintegrated.com >> ORDER



*All support questions must be made to sensor or module manufacturer

**S32V-SONYCAM & MAX9286S32V234 schematics are also available at nxp.com/sbc-s23v234 >> SOFTWARE & TOOLS

***NXP does not sell cables. Cables should be bought from external venders.

Remote Camera Use Cases

• If user is intending to connect remote cameras with coaxial cables on LVDS lines using Maxim serializer and de-serializer, please order following parts

Use Case	Parts Needed
Single Camera Use Case	1x MXOV10635-S32V camera module 1x MAX9286S32V234 de-serializer board 1x Coaxial cables with FAKRA (Jack-Jack) connectors
Dual Camera Use Case	2x MXOV10635-S32V camera module 1x MAX9286S32V234 de-serializer board 2x Coaxial cables with FAKRA (Jack-Jack) connectors
Quad Camera/ Surround View Use Case	4x MXOV10635-S32V camera module 1x MAX9286S32V234 de-serializer board 4x Coaxial cables with FAKRA (Jack-Jack) connectors



Software: OS

List of supported Operating Systems

OS	Description
Linux BSP	Linux Board Support Package with fully configurable Yocto Distribution
AUTOSAR	AUTOSAR OS 4.2 is available only for Cortex-M4 core and MCAL 4.2 is available for both Cortex-M4 & Cortex-A53 cores
QNX BSP	Visit BlackBerry QNX for more information

Get Linux BSP & AUTOSAR: www.nxp.com/s32v >> SOFTWARE & TOOLS >> "Automotive SW - Linux BSP" Get QNX BSP: https://blackberry.qnx.com/





Software: NXP Software (SDK & others)

• List of NXP developed SDK and other important software

Software	Description
Vision SDK	Vision SDK contains necessary libraries and tools essential to the vision processing. It enables programming of on-chip vision accelerator IPs. Works with Linux BSP and QNX BSP.
AVB Video Listener	Driver to receive AVB streams from cameras connected through Ethernet
Structural Core Self Test Library	Library of optimized tests for M4 & A53 cores and Neon
MBIST Manager	Driver for dedicated Memory Built-in Self Test IP of SoC
sBoot	Tests the S32V2 configuration after startup, to check the configuration related safety requirements of the S32V2 Safety Manual
Automotive Math & Motor Control Library	A precompiled off-the-shelf software library with the building blocks for motor control and general mathematical applications.

Get all: www.nxp.com/s32v >> SOFTWARE & TOOLS



Tools

• List of supported NXP or 3rd party tools

ΤοοΙ	Description
IDEs	NXP S32 Design Studio(S32DS-NG v3.1 or higher) Elektrobit Tresos Studio, Green Hills MULTI
Yocto Project	An open source collaboration project that provides templates, tools and methods to help you create custom Linux-based systems for embedded products.
DDR Tool	S32V DDR Configuration and Validation Suite ALPHA 0.8.0
Compilers	GCC or other ARM compilers for ARM cores, Green Hills Compiler, NXP compiler for APEX2 accelerator (integrated in S32DS and Vision SDK), Synopsys compiler for APEX2 accelerator
Debuggers	P&E Multilink/Cyclone, Lauterbach TRACE32, Green Hills MULTI
Vivante SDK	Vivante SDK can be used with Linux OS for GPU programming

Get S32DS: www.nxp.com/s32ds Get DDR tool: www.nxp.com







G

PPI





Documents: SoC

• List of SoC specific documents

Document*	Description
Fact Sheet	High level information on S32V2 Vision processor
Data Sheet	S32V2 vision processor datasheet for Electrical and Physical characteristics
Reference Manual	Detailed manual for SoC IPs
Safety Manual	A guide for safety system developers to build safety-related systems using the safety mechanisms of the $s32v2$
Errata	List of errata for SoC with bug fixing solution
Application Notes	Instructions on how to use the features of the S32V2 vision processor
Security Documents & SW	Contact NXP

Get all: www.nxp.com/s32v >> DOCUMENTATION



Looking for Support? : <u>nxp.com/support</u>

NXP has dedicated teams of experts to handle support questions



- Technical Communities:
 - This device is only supported via "Support Requests" option.
 - However, for other devices/tools, we highly recommend to reach <u>NXP community</u> first, so that all users can benefit from questions you ask
 - Useful community link:
 - <u>S32 Design Studio</u>
- Support Requests:
 - To get technical support for this device, user should raise support tickets
 - Moreover, if you do not want to discuss special questions on community you can use this option for other devices/tools as well.
- Chat:
 - All non-technical questions can be answered by a live Chat
- 31 EXTERNAL USE





SECURE CONNECTIONS FOR A SMARTER WORLD

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP: <u>S32V234-EVB2</u>