



S32G2 SOFTWARE ENABLEMENT



This brochure describes the vast software that is available for the S32G2 vehicle network processors to help users build their application.

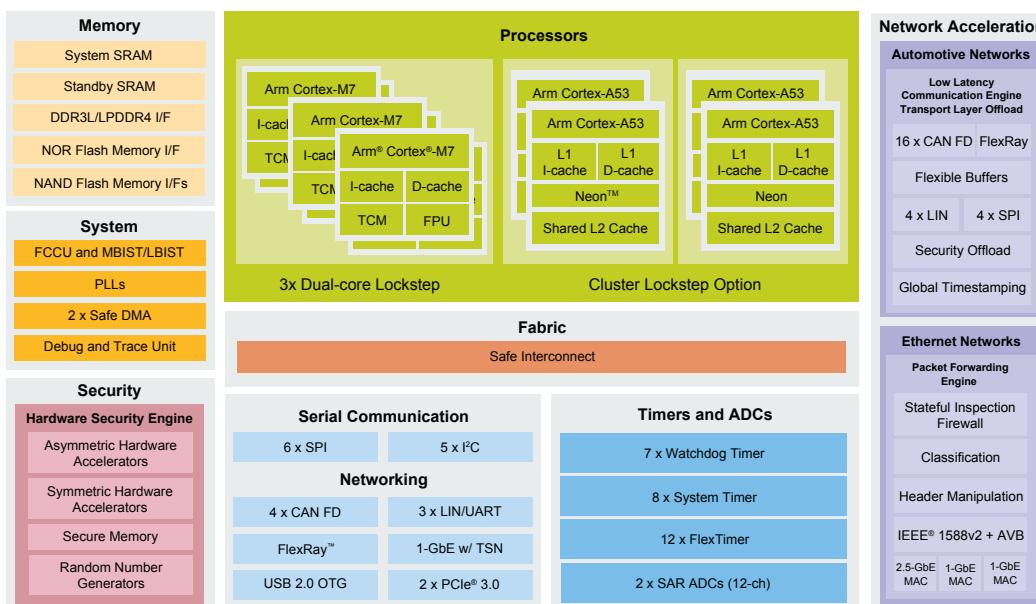
The S32G2 vehicle network processor combines ASIL D safety, hardware security, high-performance real-time and application processing, and network acceleration for service-oriented gateways, domain controllers and safety co-processors. Providing more than 10 times the performance and networking of NXP's previous family of automotive gateway devices, the versatile S32G2 processor is enabling the next generation of vehicle gateways and architectures.

The S32G2 features:

- Quad Arm® Cortex®-A53 cores with Arm Neon™ technology organized in two clusters of two cores with optional cluster lockstep for applications and services
- Triple Arm Cortex-M7 lockstep cores for real-time applications
- Low Latency Communication Engine (LLCE) for automotive networks acceleration

- Packet Forwarding Engine (PFE) for Ethernet networks acceleration
- Hardware Security Engine (HSE) for secure boot and accelerated security services
- Advanced functional safety hardware and software for ASIL D systems

S32G2 SOFTWARE AND FIRMWARE



S32G2 SOFTWARE SUPPORT

The software support offered to enable the features on the S32G2 processor can be split into 3 areas:

- Running on the Arm Cortex-M7
- Running on the Arm Cortex-A53
- Running on the accelerators

Arm Cortex-M7 cores

Real-time applications will run here on an OS like AUTOSAR and FreeRTOS. NXP provides full drivers for all peripherals and accelerators.

Arm Cortex-A53 cores

High-level operating systems like Linux will run here. Adaptive AUTOSAR can run on top of that. To aid development, NXP provides a Linux BSP for all peripherals including the accelerators.

Accelerators

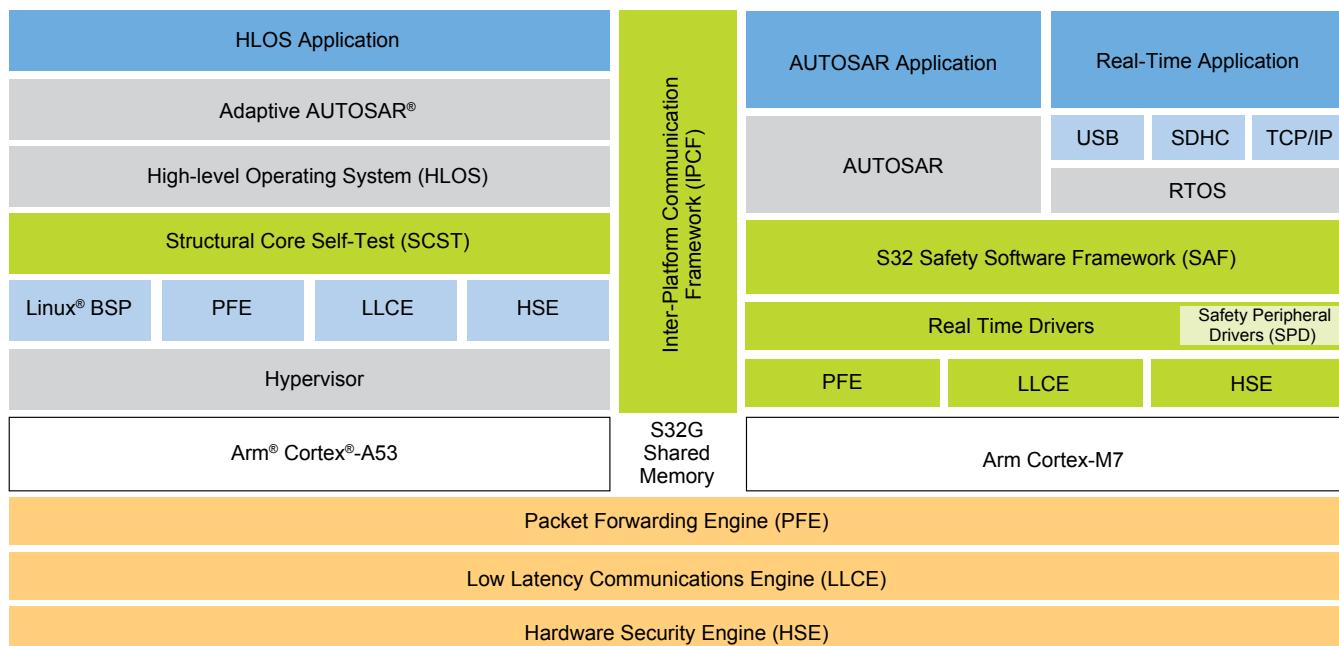
The Packet Forwarding Engine (PFE), Low Latency Communication Engine (LLCE) and Hardware Security Engine (HSE) all come with production quality firmware developed by NXP. This firmware runs on the cores within the hardware accelerators, performing the function for which they were specifically designed.

Applications running on the Arm Cortex-M7 cores and Arm Cortex-A53 cores communicate in an efficient and standard way using the Inter-Platform Communication Framework (IPCF) software package.

Real Time Drivers (RTD) can be configured using the configuration tool within the S32 Design Studio, or EB tresos, and any AUTOSAR-compliant configurator.

In addition, there is the S32 Safety Software Framework (SAF). This is premium software from NXP and is delivered as source code so it can run on the Arm Cortex-M7 or Cortex-A53 cores. However, in most automotive applications, one of the Arm Cortex-M7 cores is designated as the safety core running SAF. This software comprises software components for establishing safety foundation for the customer's safety applications. The components provide detection and reaction mechanisms for latent faults and single-point faults, enabling system ISO 26262 compliance.

S32G2 SOFTWARE SUPPORT*



* For S32G2 partner ecosystem go to page 6

SOFTWARE FROM NXP BROADLY FALLS INTO THREE CATEGORIES:

Reference software – reference-quality software to assist rapid development of your application, available free of charge.

Standard software – production-quality software for running in your application, available free of charge.

Premium software – production-quality software for running in your application, available for an additional charge.



S32G2 SOFTWARE OFFERINGS

S32G2 Reference Software	S32G2 Standard Software	S32G2 Premium Software
Linux BSP (Cortex-A53)	Real Time Drivers (RTD) (Cortex-M7, incl. EB tresos Studio)	S32G2 Security Contact sales for more information (NDA required)
FreeRTOS™ (Cortex-M7)	Safety Peripheral Drivers (SPD) (Cortex-M7)	Increased key count • IDPS • IPsec • Customization services
Integration Reference Examples (Arm® Cortex®-A53 and Cortex-M7)	Inter-Platform Communication Framework (Cortex-A53 and M7)	S32G2 Safety Available on NXP SafeAssure® site (NDA required)
USB Stack (Cortex-M7)	PFE Driver + Standard Firmware	• S32 Safety Software Framework (SAF) – Safety concept implementation – Cortex-A53 and M7 • Structural Core Self-Test (Cortex-A53)
TCP/IP Stack (Cortex-M7)	LLCE Driver + Firmware	
SDHC Stack (Cortex-M7)	HSE Standard Firmware	
S32G2 Board Diagnostic Tests		
S32G2 Tools		
S32G2 Design Studio & Config Tools		

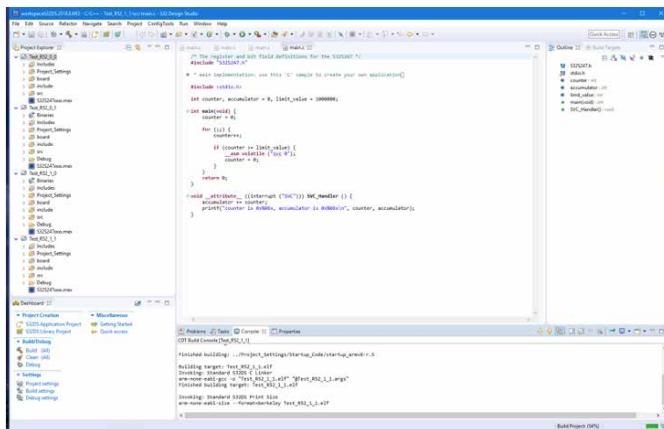
NXP SOFTWARE DEVELOPMENT TOOLS FOR S32G2

NXP provides the S32 Design Studio (S32DS), a complimentary integrated development environment (IDE) that enables editing, compiling and debugging of designs. S32DS offers designers a straightforward development tool with no code-size limitations, based on open-source software including Eclipse IDE, GNU Compiler Collection (GCC) and GNU Debugger (GDB).

S32DS supports GCC compiler/assembler/linker, but the S32G2 also supports GHS and Diab.

Integrated Development Environment – S32 Design Studio 3.x

- New Project Wizard
- Secure application debug support
- Secure Boot Support by S32 Flash Tool
- Support for real-time drivers



S32 Debugger/Trace and Profiling Tools

- Standard debug capabilities
- Scripting and logging
- Multicore debug/flash through JTAG
- USB/Ethernet probe connection (Remote/Local)
- Secure application debug support
- Trace and profiling on Cortex-A53 Cores

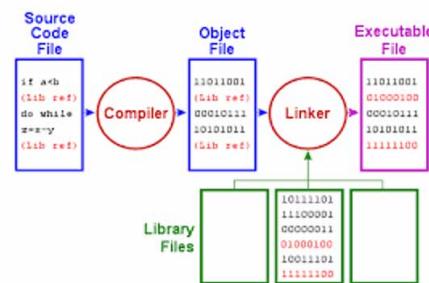


Within S32DS there is a suite of configuration tools giving the user an easy method of configuring the device:

- Pins
- Clocks
- Peripherals
- DCD
- IVT
- QuadSPI
- DDR

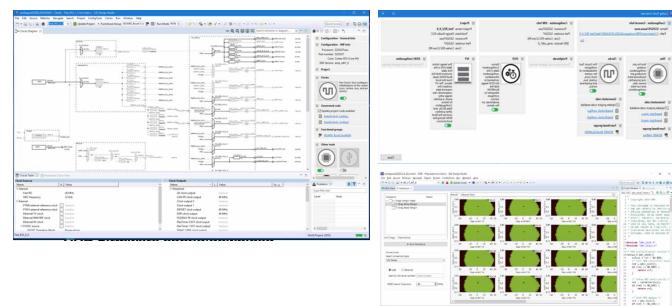
Code Generation Tools

- Compilers, assemblers, linkers
- NXP GCC-v6.3 & v9.2



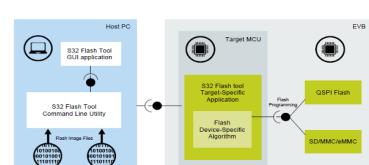
S32 Configuration Tools

- DDR configuration/QSPI configuration
- DDR firmware config/download
- DCD/IVT/clocks/pins/peripherals
- Support for real-time drivers



S32 Flash Tool

- GUI/Command line flash
- QSPI/SD/eMMC support
- UART/CAN/Ethernet
- Program IVT/DCD/self-test/HSE/application boot



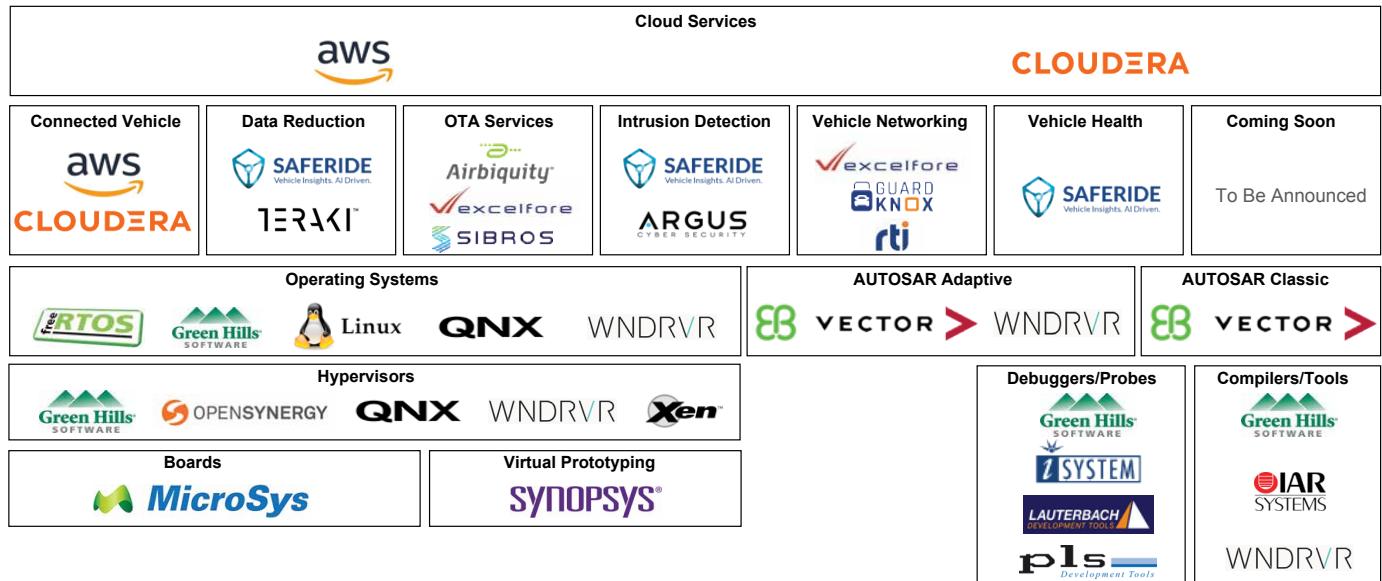
S32G2 ECOSYSTEM PARTNERS*

As well as the software and tools provided by NXP, there is a large ecosystem offering for the S32G2 family of processors.

The trusted partners of NXP provide expertise and technologies in their areas of specialty to help create complete solutions with S32G2 processors.

- Operating systems and hypervisors
- Development tools (compiles, debuggers, probes etc.)
- Application-level software covering many aspects, e.g., cloud services, OTA, networking, etc.

S32G2 PARTNER ECOSYSTEM*



* S32G2 partner list as of May 4, 2021. Check with partners for support details. Contact NXP Sales for future updates.





SOFTWARE DELIVERABLES AND QUALITY PROCESSES

The following table shows the main software packages available for the S32G2 processors.

For further information, click the link to the corresponding product brief.

The subsequent table explains the different software quality classifications.

Software Product	Description	Software Type	Delivery	Arm Core	Quality Class
Automotive SW-Linux BSP	S32G2 processor BSP provides a foundation software platform which contains various libraries and middleware, sample applications for S32G2 SoCs and includes the following components and features: <ul style="list-style-type: none"> • Arm® Trusted Firmware • U-Boot • Linux® Kernel • Yocto • ROOTFS • Drivers • IPCF • POSIX 	BSP	Source	A53	Class O
RTD	The Real Time Drivers (RTD) software product offers support for both AUTOSAR® and non-AUTOSAR applications. For AUTOSAR applications, a wide range of standard drivers and complex device drivers (CDDs) create a rich ecosystem. For non-AUTOSAR, the low-level drivers are also provided for highly optimized code. Support for Elektrobit tresos Studio (AUTOSAR) and S32CT (non-AUTOSAR) configurators is included.	Driver	Source	M7	Class A
SPD	Safety Peripheral Drivers (SPDs), a subset of RTDs, are specifically for the safety peripherals (FCCU, EIM, ERM, STCU, BIST, eMCEM).	Driver	Source	M7	Class A
IPCF	Inter-Platform Communication Framework (IPCF) enables applications running on multiple cores to communicate over various transport interfaces (shared memory, PCIe®, Ethernet, etc.)	Middleware	Source	A53/M7	Class A
LLCE	Low Latency Communication Engine (LLCE) controls the traditional automotive communication interfaces such as CAN, LIN, and FlexRay™. The LLCE can offload the host CPU from all interface-level tasks. <ul style="list-style-type: none"> • CAN, LIN, FR communication controller • Timestamping • CAN-to-CAN routing • CAN-to-Ethernet routing 	Firmware	Binary	-----	Class B
PFE	PFE is the Ethernet packet accelerator to offload core from an overwhelming level of network processing: Forwarding, NAT, VLAN, L2 bridge, IPsec and QoS, etc. in data plane.	Firmware	Binary	-----	Class B
HSE	This is firmware for the Hardware Security Engine (HSE) subsystem. It essentially serves the host (application cores) with a set of native security services.	Firmware	Encrypted Binary	-----	Class B

SOFTWARE QUALITY CLASSES

Class A

- Functional safety products — ISO 26262 compliant
- Products: SAF, RTD, IPCF

Class B

- SPICE-compliant products
- Products: LLCE, PFE, HSE

Class C

- SPICE tailored — CMMI compliant
- Products: System tools, IDE, integration example code

Class O

- Open-source software class
- Products: Linux® BSP, USB, SDHC & TCP/IP

Class D

- Demo/prototypes
- Products: demos, pre-EAR SW

 SafeAssure® and Automotive Qualified

 Automotive Qualified

 General Availability

LEGAL DISCLAIMERS

Disclaimer related to a project description/roadmap

The information given hereunder is non-binding and preliminary and provided without legal commitment whatsoever. The information may be subject to changes and amendments. As with any project, inherent uncertainties can lead to the termination or delay of the project at any time. NXP does not accept any liability with regard to the project description given hereunder nor to any project realization. Any project commitment is subject to conclusion of a separate duly signed contract.

Disclaimer for timelines/schedules

The dates provided herein are non-binding and preliminary and provided without legal commitment whatsoever. The timeline, and the assumptions underlying that timeline, are subject to change at any time. NXP does not accept any liability with regard to the dates provided. Any dates or other information provided by NXP are binding only upon conclusion of a written contract signed by customer and NXP.