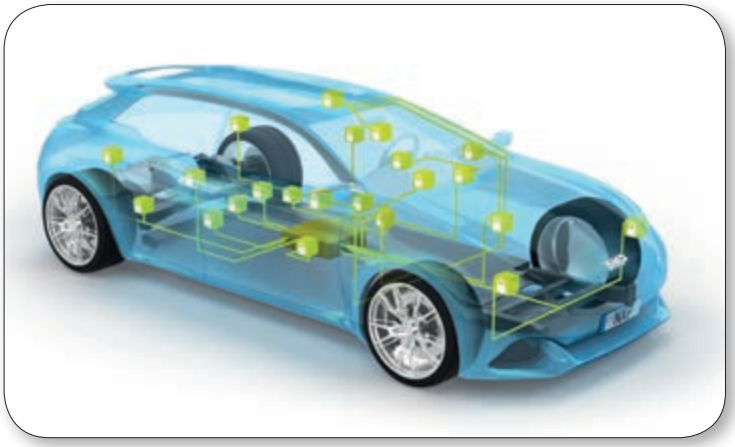
Microcontrollers combine safety and security with full-automotive software

By Danny Basler, NXP

This article describes in detail the features of a new MCU series specialized for automotive applications.



Carmakers are in a race to change, with in-car functions evolving as the market moves from motor vehicles towards mobility platforms. Even relatively static body applications like doors, steering wheels, seats, climate control and lighting systems are getting a makeover.

To help meet these challenges, NXP has introduced the S32K family of automotive microcontrollers (MCUs). The 32-bit ARM Cortex-based MCUs are the latest AEC-Q100 qualified, high-reliability products to address this segment. Leveraging the legacy of more than 60 years of experience and expertise, the S32K family offers new opportunities for end product differentiation in secure, connected vehicle edge nodes. Broad family scalability includes the following. Compatible MCU families with multiple performance, feature and pin-count options for fast, efficient end product platform evolution including a memory range from 128 KB to 2 MB, pin count from 32 to 176 pins, QFN, LQFP, MAPBGA packages, and IP compatibility across families. Next generation processing and peripherals – ARM Cortex-M class cores, CAN FD, robust hardware security, and low power consumption. Comprehensive software solution – automotive-grade Software Development Kit (SDK), S32 Design Studio IDE and third-party ecosystem support.

The S32K1 series consists of six MCU families spanning from 128 KB to 2 MB of flash memory. Use of area efficient 90nm Thin Film Storage (TFS) process technology allows a high level of on-chip feature integration while keeping MCU cost accessible to body node applications price points. Fast, low voltage transistors and distributed charge structure in the bit cell ensure fast access times and high immunity to leakage. MCUs operate from 2.7 to 5.5 V and feature a range of next-generation and market-proven digital and analog IP suitable for electrically harsh environments at up to 125°C ambient temperature.

Traditional body control MCU performance is being increasingly stretched. Interpreting complex sensor algorithms, managing high bandwidth communication peripherals and implementing advanced security features for stored and transmitted data are now common requirements in even modest ECUs. Performance headroom must also be considered for evolving software requirements such as the AUTOSAR standard, where a 10 to15% overhead is typical. The MCU series meets this need through use of the DSP enabled ARM Cortex-M4 and ultra-low power ARM Cortex-M0+ cores on the K14x and K11x families respectively. Operating frequency ranges from 48 to 112 MHz and is supplemented by a variety of proprietary acceleration technology. Extensive DMA support and crossbar switch technology streamline data throughput through reduced CPU loading, while on-chip I/D cache reduces memory access latencies. MCUs of this family also feature a bit manipulation engine (BME) that reduces code size and execution time by an average of 40% when performing bit-oriented math operations. With 100 ECUs now common in even mid-size cars, in-vehicle networking and end-of-line programming requirements are considerable and place an increasing demand on classic CAN 2.0 bus bandwidth. Layering security functions on top requires more memory and bus utilization to implement cryptographic functions. The MCUs include up to 3 ISO-compliant CAN FD modules which increase the data rate from 500 kb/s (typical) to 2 Mb/s in normal mode and up to 5 Mb/s in programming mode. A new message frame format expands the data field from 8 to 64 bytes reducing data overhead and increasing protocol efficiency. The MCUs can operate in dedicated CAN FD networks or mixed CAN 2.0/CAN FD ones where nodes are upgraded on a case-by-case basis.

NXP has been at the leading edge of automotive MCU security for many years with several generations of hardware peripherals in powertrain, safety and gateway MCUs. This capability now extends down to body

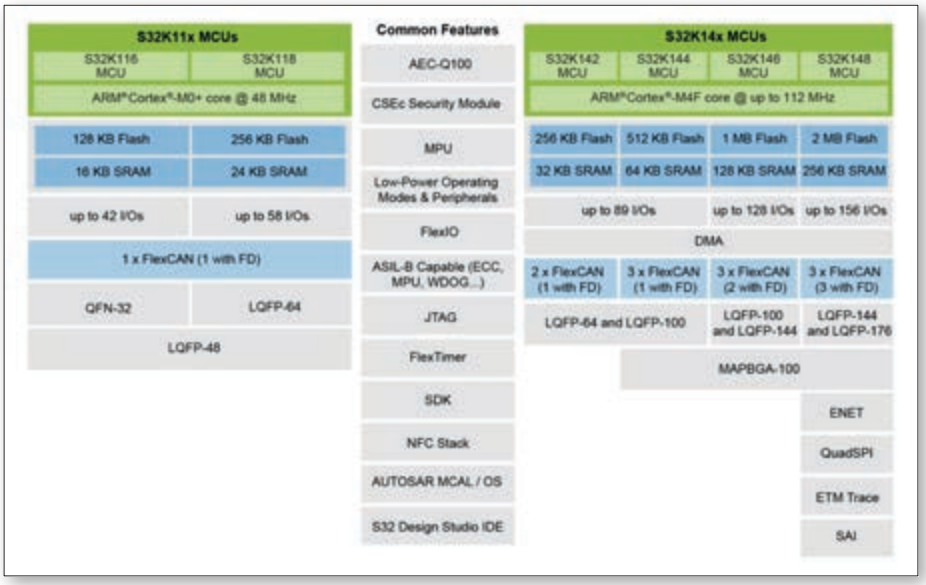


Figure 1. S32K1 MCU block diagram

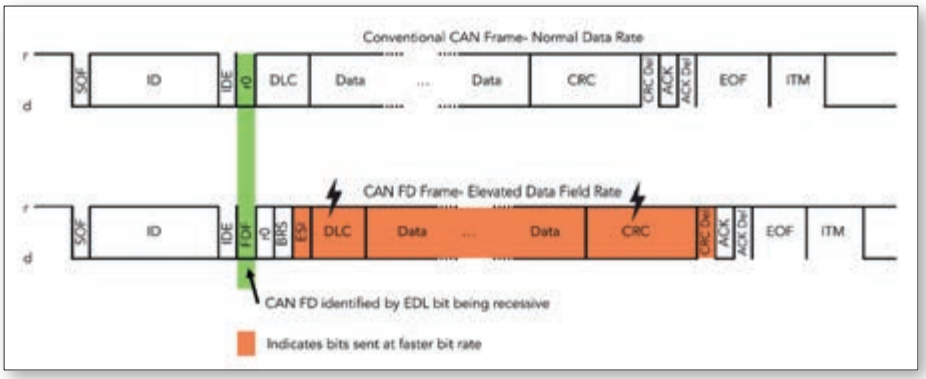


Figure 2. CAN FD frame

applications with the new Cryptographic Services Engine compressed (CSEc) which has been optimized for smaller memory MCUs. Compliant with the SHE (Secure Hardware Extension) standard, and EVITA-Low (E-safety Vehicle Intrusion Protected Applications) guidelines, the CSEc features a dedicated security co-processor and provides secure key storage, AES-128 encryption and decryption, and secure boot functions.

Using the internal 32-bit CSEc co-processor, firmware and a hardware-assisted AES-128 sub-block, the flash memory module enables encryption, decryption and Cipher-based Message Authentication Code (CMAC) algorithms for secure messaging. Two secure blocks of flash memory are pre-programmed with firmware, a unique identification number (UID) and a secret key (SK) - a random number whose value is never disclosed. The unique identifier (UID) is 120 bits long and programmed during manufacture ensuring no two MCUs contain identical keys. Furthermore, establishing the value of the keys in secure flash would require a huge effort and with each key being unique, even compromised keys would be useless for attacking other systems.

At system boot the CSEc core executes boot code from a dedicated ROM which then loads the firmware from secure flash into RAM and starts executing. This reduces the flash accesses by the CSEc's core on the crossbar thereby avoiding any impact on MCU system performance. If the secure boot code authentication fails, the application may offer a reduced level of functionality, or set a flag to deny use of the keys stored in secure flash. Use cases for the CSEc module are numerous and include mileage tamper prevention, component authentication and secure telematics.

The MCU family feature a range of low power architectural techniques, operating modes, and autonomous peripherals with low power functionality. At a foundational level the 90nm TFS process technology delivers a significant reduction in dynamic power vs. comparable technologies. Extending this is the option to de-rate the clock frequency during periods when maximum performance is not required. With the MCU clock tree accounting for most of the power consumption (up to 30% in some designs), clock gating is available for many peripherals while a low power boot feature can configure default clock settings to reduce losses during the boot phase. For the most extreme application power profiles, memories and peripherals can also be power-gated when inactive. The MCUs contain 3 active and 4 standby operating modes each accompanied by multiple wake-up sources. In all modes, all memory and register contents are maintained which simplifies software handling especially in AUTOSAR related applications. These are: RUN modes – high speed RUN

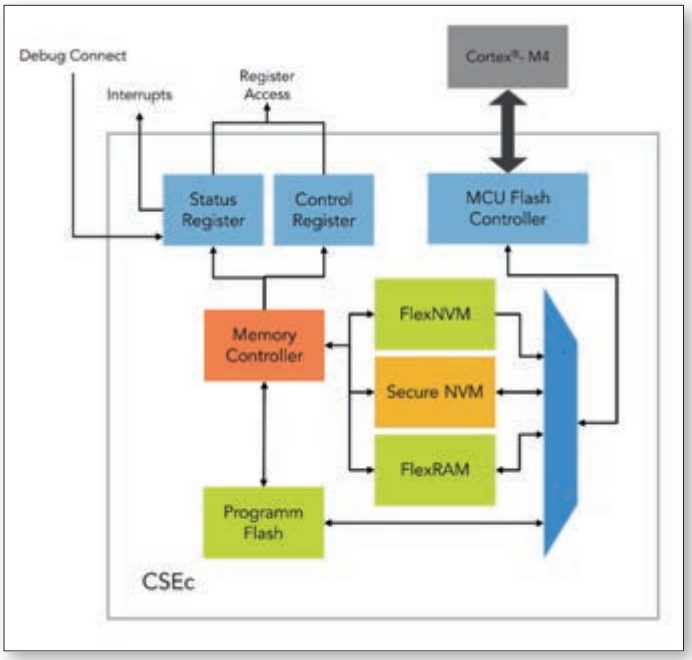


Figure 3. Cryptographic Services Engine compressed (CSEc) module

(HSRUN) mode allows over-drive conditions of up to 112 MHz whereas standard RUN mode limits the CPU clock to a maximum of 80 MHz. VLPR (very low power RUN) is new for automotive use cases and de-rates the core to 4 MHz while using a very low Idd for flash memory access. Here the internal regulator is placed in standby with full peripheral and low voltage detect functionality maintained. WAIT and very low power WAIT (VLPW) modes – similar to their equivalent run modes but the CPU is halted and flash memory and FlexMemory (EEPROM) programming is disabled. With interrupts enabled, the MCU can exit WAIT modes, perform the scheduled task and then quickly return to a low power state. This minimizes average power in applications that frequently toggle between active and reduced power states with savings of 30 to 60% achievable over RUN mode. STOP and VLPS – deep sleep modes where all I/O pins and several peripherals can function as wake-up sources. These have slightly longer wake-up times and are suited for applications where wake-up occurs infrequently. Current consumption at 25°C in VLPS mode starts from as low as 25µA with a recovery time of approximately 5µs.

Several of the analog, communication and timing peripherals can also perform basic timekeeping and monitoring tasks autonomously and in low power states. These include LPUARTs that can asynchronously transmit/receive LIN messages, and ADCs (1 Msps capable) that operate in low power states and activate when a threshold value has been reached. The DMA controller also allows data transfers in STOP/VLPS with the core inactive and minimal clocks enabled.

A typical body application use case would be fast sensor input measurement with power-up when pre-defined conditions have been met. This would be accomplished using the on-chip 128 kHz IRC (internal RC oscillator) to clock an API to wake the MCU every 10ms and toggle between VLPS and RUN mode. On power-up the MCU would switch to its 48 MHz IRC for fast execution. Conversely, in a slow sensor application the MCU would operate in VLPS using an 8 MHz IRC for reduced 4 MHz execution. When the sensor setting time has completed and data becomes available, the API would then change from VLPS to VLPR resulting in a much lower peak current. Another peripheral worth mentioning is the new FlexIO module. Highly configurable, it can emulate various communication peripherals – UART, I2C, SPI, and I2S – or generate 16-bit timers with support for trigger, reset, enable and disable conditions. Compared with GPIO software emulation schemes, the FlexIO module requires less CPU overhead and can operate in all power and debug modes.

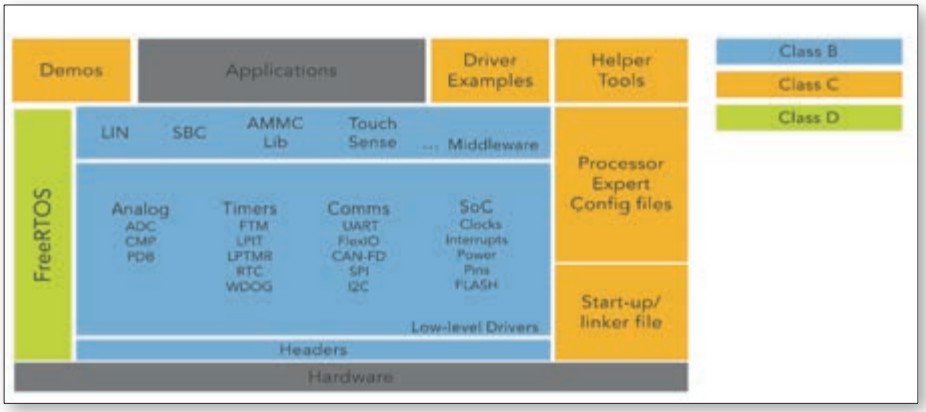


Figure 4. S32K software development kit

The MCUs are members of the NXP SafeAssure program and target the ASIL B integrity level with higher level compliance possible based on system level redundancies. Devices are designed in accordance with the ISO26262 standard with hardware and system level safety measures and comprehensive safety documentation. Features consist of error correcting code (ECC) on flash and RAM memories, a memory protection unit (MPU) for assigning secure access rights, internal and external watchdogs, a cyclic redundancy check (CRC) block and a structural core self-test library (SCST) for detecting permanent faults in the core. Detailed safety manual and FMEDA documentation is available to support system level certification.

The MCUs are supported by a range of automotive-grade development tools to speed and simplify the software development process. S32 Design Studio (S32 DS) is a free of charge, unlimited code size, Eclipse-based IDE with plug-in support. Using the Processor Expert tool included into S32 Design Studio, developers can configure the peripherals (internal and external) and software functionality using a simple GUI. The tool then generates highly optimized embedded C-code saving a huge amount of manual development effort.

For applications that do not require AUTOSAR support, an automotive-grade software development kit (SDK) is available. The SDK consists of free of charge peripheral drivers, the FreeRTOS operating system and application-specific middleware. MISRA 2012 and SPICE Level 3 compliant, the SDK comes pre-installed within the S32 DS IDE and is also compatible with third party compilers and debuggers including those from third party IAR Systems. The SDK plug-in for Design Studio greatly reduces the development effort, thanks to its GUI, where the developer can configure and drag and drop software functions to use the peripheral drivers.

With robust, high speed timers and analog peripherals, S32K MCUs are well suited to the growing number of in-car electric motor applications – fuel/oil/water pumps, HVAC systems and seats/mirrors. To assist developers a range of design tools are offered: embedded Automotive Math and Motor Control Library (AMMCLIB), Motor Control Toolbox for Matlab model-based design), and the FreeMASTER run-time debug monitor with Motor Control Application Tuner (MCAT) plug-in. ■