# Release Notes

for S32G274A HSE Firmware 0.0.9.2

Rev. 1.0, 18-June-2021

# Contents

# Getting Started

**IMPORTANT NOTES:**

This is the Standard Package variant of the HSE Firmware for S32G274A, distributed for evaluation purposes only.

This version of the firmware can be used for evaluation of crypto services and **shall not be used in production**.

# Package content

This package contains the NXP S32G274A HSE Firmware 0.0.9.2:

- HSE Firmware: encrypted binary
- HSE Firmware interface files
- HSE Service API RM
- `HSE_FW_S32G274A_0.0.9.2_ReleaseNotes.pdf` – this file
- The license.txt EULA file and the `uninstall.exe` utility for removing the HSE FW binary

**NOTE:**

Demo Application is provided separately and contains details on how to provision HSE FW on virgin devices and demonstrates common use cases of its security features.

One can access via NXP DocStore (https://www.docstore.nxp.com) the following associated documentation:

- HSE Firmware Reference Manual

# Installation

Follow the install steps in the demo application.

If targeting the usage of AUTOSAR software stack in the application, it is recommended to install also the RTD crypto driver from S32XX AUTOSAR RTD package by following its installer steps.

# Release Details

This is the HSE Firmware 0.0.9.2 **hotfix** release for the S32G274A platform.

The provided example code shows how to setup and use the HSE FW and to perform basic crypto operations (refer to the documentation that comes with demo application). The examples show how to:

- Boot the demo-application (secure mode)
- Load the firmware
- Load the key(s)
- Perform crypto operations

This release was developed and tested using:

- Chip: PS32G274ABVUC-0P77B (Rev 2.0)
- Chip: PS32G274ABVU-IN92V (Rev 1.0)
- Motherboard: S32G-RV-PLATEVB (700-30081 REV A)
- Mini-Module: S32G-PROCEVB-S (700-32170 REV X3)

Standard HSE Firmware package contains the following configuration:

- 20 RAM keys, 40 NVM symmetric keys, 12 NVM asymmetric keys
- 4 SMR entries, 4 CR entries
- Support only for ECC-256bits and CURVE25519 curve (Montgomery and Twisted)
- Maximum key size limitations: HMAC - 512bits, ECC - max 256bits, RSA- max 2048bits
- SHA3 and IPSec services are not supported

**For Premium Package variant, the customers would need to purchase "premium S32G274A security parts" (to run Premium Package variant in production).**

**NOTE:** This is a preliminary release and has been tested on silicon samples **in CUST_DEL and OEM_PROD life cycles**. Limited testing was performed in IN_FIELD life cycle.

### Implemented Errata:

N/A

# Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP:

- S32G274A
- S32G254A
- S32G233A
- S32G234M

## Device Bricking scenario for HSE Firmware

- **S32G274A Rev 2.0**: No bricking scenario identified at this moment.
- **S32G274A Rev 1.0**:
    - There is an IVT_AUTH fuse bit. The purpose of this fuse bit is to enable BootROM authenticate the IVT in Life Cycle OEM_PROD or IN_FIELD. The signature of IVT is generated by HSE firmware on the device with device specific key. In case IVT_AUTH fuse bit is programmed, and the Life Cycle is advanced to OEM_PROD or IN_FIELD, BootROM cannot boot the HSE firmware and will issue a reset to the system. Upon multiple reset the device enters serial boot mode. One can program a new IVT in serial boot mode, but it cannot be signed again.
    - The reason it cannot be signed is that, since IVT_AUTH fuse bit is programmed BootROM always expects a signed IVT. Hence it cannot boot the HSE firmware if IVT is corrupted. The HSE firmware is needed to sign the IVT. This leads to device been bricked.

**Recommendation**

**Do not blow IVT_AUTH fuse bit**. This fuse bit is blown by HSE attribute ***HSE_ENABLE_BOOT_AUTH_ATTR_ID***.

## Security Aspects

Current release of HSE Firmware implements partial countermeasures against logical attacks (e.g. input parameter checking, address ranges). The code contained in this release was not subject to penetration testing / vulnerability attacks verification. As such, this version of the firmware can be used for evaluation of crypto services but not for robustness against attacks and **shall not be used in production**.

# Change Logs in 0.9.2 (hotfix)

## Fixed

- HSE reports a fatal error when multiple requests of different priority levels are issued concurrently. This impacts the parallel execution of a subset of the high priority services (service ID: *0xXXA5XXXX*) that are processed synchronously by HSE with a subset of the low priority services that are using the symmetric crypto accelerator. *(ASHF-3492)*

## API updates

- *HSE_SRV_ID_GET_RANDOM_NUM* macro value updated.

# Change Logs in 0.9.1 (hotfix)

## Removed

- The restriction on allowing an SMR to be updated, in advanced life cycles (OEM_PROD, IN_FIELD), only if it is already verified successfully. *(ASHF-3433)*

## Fixed

- Encrypted SMR not linked with CR table that have the periodic checks enabled are not decrypted when loaded in RAM during start-up. SMR that are loaded at run-time (linked with CR on-demand entries or not linked at all) can be used for periodic checks only without encryption. *(ASHF-3306)*
- OTFAD installation/update does not trigger a correct update of the SYS_IMG and publishing it in this case outputs corrupted content. *(ASHF-3434)*
- OTFAD module will not be enabled if ALL installed regions have *HSE_OTFAD_CTX_INACTIVE_ON_BOOT* flag set. *(ASHF-3322)*
- HSE goes to shutdown mode, becoming non-operational, if RNG initialization fails at start-up. *(ASHF-3435)*
- The key catalogs referenced in the Format Key Catalogs service request cannot be in the address range above first 3.5 GB. *(ASHF-3148)*
- The key generation service for RSA and ECC is not providing the correct public key. *(ASHF-3049)*
- Loading a SHE RAM Key using a key that has the DEBUG PROT flag set fails (returns key not available). *(ASHF-3309)*
- Export a SHE RAM Key using the Master ECU key that has the DEBUG PROT flag set fails (returns key not available). *(ASHF-3310)*
- SMR verify request using RSA authentication scheme returns INVALID_PARAM instead of VERIFY_FAILED when signature is not flashed. *(ASHF-2957)*

# Change Logs in 0.9.0

## Added

- Handling of tamper violations: clock monitoring (CMU), temperature sensor (TMU) and physical tamper.
- Service *HSE_SRV_ID_SIPHASH* (see *hseSipHashSrv_t*)
- Service *HSE_SRV_ID_XTS_AES_CIPHER* (see *hseXtsAesCipherSrv_t*)
- Service *HSE_SRV_ID_PREPARE_FOR_STANDBY*. This service must be called by the host before entering in stand-by mode.
- Service *HSE_SRV_ID_ON_DEMAND_CORE_RESET*. This service allows configuring CR entries and their associated SMR to be processed at run-time, on-demand.
- Event *HSE_WA_SMR_PERIODIC_CHECK_FAILED*. This event signals the host that a periodic check SMR failed (the verification failed).
- Register HSE GPR for tamper status. This register is updated by HSE when a tamper is configured. It can be read by the host to check what tampers are configured.
- Prevention of the accesses to shared memory of other cores via HSE (see `hseAttrAllMuMemRegions_t` attribute).
- Service versioning using a byte from the service ID to encode the version for each service.

## Updated

- Interface comments
- SHE keys catalog formatting must be configured for *HSE_KEY_OWNER_ANY* group owner.
- Secure Memory Regions (SMR):
  - Added support for encrypted SMR. The SMRs can be encrypted using GCM or CTR.
  - Removed the SMR verification method field (*hseSmrVerifMethod_t*).
  - The SMR periodic tick has been updated from *10ms* to *100ms* (at 400MHz frequency)
- Core Reset:
  - Updated Core Reset entries to include *PRE-BOOT*, *ALT_PRE_BOOT* and *POST-BOOT* SMR bitfields.
  - Included *hseCrStartOption_t* option: auto-start (automatically release the core from reset at start-up) or on-demand (the core boot is triggered on demand by another application core)
  - Added *HSE_CORE_RESET_RELEASE_ATTR_ID* attribute to configure the release-from-reset strategy:
    - all-at-once (cores are release all at once after all boot SMRs are verified);
    - one-by-one (cores are release one by one as soon as the boot SMR(s) verification passed for that core)
- Fast CMAC:
  - Use input and tag length in bits
  - Included *HSE_FAST_CMAC_MIN_TAG_BIT_LEN_ATTR_ID* attribute to configure the minimum tag bit-length that can be used for Fast CMAC verify / generate.

- HSE errors reported to HOST are divided into warnings and errors
- If VDD_EFUSE is connected to GROUND and a fuse needs to be written, the HSE FW returns an error.

## Removed

- All TDES support.
- IV Length parameter from symmetric cipher (see `hseSymCipherSrv_t`)
- `HSE_KDF_SP800_108_FEEDBACK` and `HSE_KDF_SP800_108_PIPELINE` KDF SP800 modes. Only Counter Mode remained supported.

# Change Logs in 0.8.5

## Added

- *HSE_HOST_PERIPH_CONFIG_DONE* event sent from host to HSE (writing the MUB_GCR register) to signal when the system and QSPI/SD/eMMC clock configuration were updated by application (see *hseHostEvent_t*)
- CRC32 service (see *hseCrc32Srv_t* service)
- On-the-fly AES decryption support (see *hseInstallOtfadContextSrv_t*, *hseActivateOtfadContextSrv_t*, *hseGetOtfadContextSrv_t*)
- Scatter-Gather support for RSA and ECDSA signature (refer to *hseSignSrv_t)*
- New HSE attributes (see *hseAttrId_t*):
    - HSE_AVAIL_ANTI_ROLLBACK_COUNTER_ATTR_ID – The anti-rollback counter updates left
    - HSE_FW_PARTITION_ATTR_ID – specifies the partition (primary or backup) used by BootROM to load the HSE Firmware (only for Rev2.0)
    - HSE_OTFAD_CTX_STATUS_ATTR_ID – returns the OTFAD context status

## Updated

- Interface comments
- Error events reported by HSE (see *hseError_t*). To clear the errors, the host must read the MUB_GSR register and write back the register value (W1C)
- Application header for Basic Secure Boot: “tag address” and “key type” fields were removed; “core ID” field not used; core booted is specified by BOOT_TARGET in IVT; the tag must be placed after the application code (see *hseAppHeader_t*, *hseBootDataImageSignSrv_t*)
- Import/Export key to support GCM and CCM: AAD, Tag and IV should be specified in the AEAD scheme, and the keyInfo (key proprieties) must be within AAD (see *hseImportKeySrv_t*, *hseExportKeySrv_t*)
- Publish SYS-IMG service: publish SYS-IMG in chunks was removed (see *hsePublishSysImageSrv_t*)
- TLS1.2 KDF to support KEY-EXCHANGE-PSK and KEY-EXCHANGE-ECDHE-PSK (see *hseKdfTLS12PrfScheme_t*)
- HMAC to support key sizes greater than hash block size (updated *hseMacSrv_t* service)
- Firmware Update service to return the total length of published HSE FW (see *hseFirmwareUpdateSrv_t*)
- Encrypt-then-MAC operation (see *hseAuthEncSrv_t* service) to support addition combination AES_CFB/OFB-THEN-HMAC.
- Erase service to delete the SHE keys only if system authorization was performed beforehand using MASTER_ECU key. Other keys (non-SHE keys) can be erased if the authorization operation was performed using any key type, including MASTER_ECU key (see *hseEraseKeySrv_t*)
- Load ECC service to save the ECC user-curve domain parameters in SYS-IMG (which needs to be published). The host needs Super User rights to be able to load an ECC user-

defined curve. The loaded ECC user-defined curves must have the private key size equal to or greater than 192 bits.

- SYS Authorization feature (see *hseSysAuthorizationReqSrv_t*) to perform the authorization using the *SHE MASTER_ECU_KEY*. SHE keys can be erased only if the host is authorized with MASTER_ECU_KEY.
- Core Reset (CR) / Secure Memory Regions (SMR) (see *hseSmrEntry_t*, *hseCrEntry_t*):
  - Added support for SMR periodic check
  - Updated CR to support an Alternative SMR(s) verification (called PRE_BOOT_ALT) if the PRE_BOOT SMR(s) verification fails.
  - Defined *HSE_SMR_VERIF_RUN_TIME_MASK* verification method used only for on-demand SMR verification
  - For unsecure boot (BOOT_SEQ=0), the SMRs are not loaded at boot time. Application can use the *hseSmrVerifySrv_t* service to load and verify the SMR (for validation purpose)
  - Updated the SHE Secure Boot used with SMR entry#0 (see *hseSmrEntryInstallSrv_t* comments)
  - Updated *hseSmrVerifySrv_t* service to:
    - loads and verifies the RUN_TIME SMR (if loaded before, HSE will perform only the verification)
    - PRE-BOOT, PRE-BOOT-ALT or POST-BOOT SMR can be verified on-demand only if:
      - it was loaded at boot-time (only verification in SRAM is performed)
      - or the *BOOT_SEQ = 0*: first call will trigger the load and verification; next calls will perform only the verification in SRAM.

## Removed

- *HSE_FLASH_PAGE_SIZE_ATTR_ID* attribute

# Known Issues

- SipHash is not functional with variant *HSE_SIPHASH_VARIANT_32* *(ASHF-2998)*
- The Physical Tamper Configuration is not functional when the Filter Duration is disabled (if configured as 0) *(ASHF-3298)*

# List of Limitations Existing in This Release

- VDD_EFUSE must be connected to *1.8V* in order to write the HSE fuse area. The fuses are written at the SYS-IMG or HSE FW update or thought Set Attribute service (e.g. life cycle, ADKP key, debug authorization method etc.). If *VDD_EFUSE* is connected to *GND*, the HSE fuse area can only be read (cannot be written)
- The translation of the HSE pink image into a blue image will always trigger a fuse counter update (when verified during start-up sequence)
- Limited tests have been performed on the *HSE_PHYSICAL_TAMPER_ATTR_ID* and *HSE_TEMP_SENSOR_VIO_CONFIG_ATTR_ID* attributes. Also, the CMU Tamper Configuration has been limited tested.
- Limited tests were performed on services using 40 bits addresses.
- During SD card booting, the last 4KB (*start_address = 0x347FF000, length = 0x1000*) and first 32KB (*start_address = 0x34000000, length = 0x8000*) from SRAM are used. These memory regions can be used by the application after a successful HSE initialization (HSE status shall be *HSE_STATUS_INIT_OK* for *BOOT_SEQ = 0* and *HSE_STATUS_BOOT_OK* for *BOOT_SEQ = 1*).
- Defining SMR that are checked periodically may impact HSE performance.
- The quality of random number generation (RNG) is guaranteed between 200MHz and 400MHz *XBAR_CLK*. If the *XBAR_CLK* is below 200Mhz, the required source of entropy is not ensured for the services that use the RNG module (e.g. asymmetric cryptographic services, random number generation service).

# List of Services Available

**NOTE:**

**All available HSE features/services** are also listed in the *hse_h_config.h* file (from HSE Interface). All other features not listed in the table below (or enabled in *hse_h_config.h* file) **are NOT supported**.

| Service Class | HSE Service ID | Description/Notes |
|---|---|---|
| **Administrative** | HSE_SRV_ID_SET_ATTR | **Set an HSE attribute**.<br>Attributes related to FUSE memory can be written only once (e.g. Debug Key) or can only be advanced (e.g. Life cycle). **Care must be taken.** |
| | HSE_SRV_ID_GET_ATTR | **Get an HSE attribute.** |
| | HSE_SRV_ID_CANCEL | **Cancel a one-pass or streaming service on a specific channel.**<br>An HSE service request can be cancelled if it is in the processing queue and NOT passed to the hardware to be executed. |
| | HSE_SRV_ID_FIRMWARE_UPDATE | **HSE firmware update** (generates the HSE FW blue image) |
| | HSE_SRV_ID_SYS_AUTH_REQ | **SYS Authorization request** used to be granted with CUST/OEM SuperUser rights |
| | HSE_SRV_ID_SYS_AUTH_RESP | **SYS Authorization response** (response to SYS Authorization Request) |
| | HSE_SRV_ID_BOOT_DATA_IMAGE_SIGN | **Generate the signature on IVT, DCD & SELF TEST images**. Also, signs the APP image for Basic Secure Boot (BSB). |
| | HSE_SRV_ID_BOOT_DATA_IMAGE_VERIFY | **Verify the signature on IVT, DCD & SELF TEST images.** Also, verifies the APP image for Basic Secure Boot (BSB). |
| | HSE_SRV_ID_IMPORT_EXPORT_STREAM_CTX | **Import and Export service for the crypto streaming context.** |
| | HSE_SRV_ID_PUBLISH_SYS_IMAGE | **Publish SYS-IMAGE file in System RAM.** |
| | HSE_SRV_ID_GET_SYS_IMAGE_SIZE | **Get SYS-IMAGE size.** |
| | HSE_SRV_ID_PUBLISH_LOAD_CNT_TBL | **Request to publish/load the NVM container for the Monotonic Counter table** |
| | HSE_SRV_ID_INSTALL_OTFAD_CTX | Install an On-The-Fly AES Decryption (OTFAD) context. |
| | HSE_SRV_ID_ACTIVATE_OTFAD_CTX | Activate on-demand OTFAD context |
| | HSE_SRV_ID_GET_OTFAD_CTX | Get OTFAD context information |
| | HSE_SRV_ID_PREPARE_FOR_STANDBY | Prepare HSE before system goes to Stand-by mode |
| **Key Management** | HSE_SRV_ID_LOAD_ECC_CURVE | **Load the domain parameters for a Weierstrass ECC curve.**<br>This service can be used to support additional Weierstrass ECC curves (which are not supported by default). The loaded ECC curve domain parameters are persistent. |
| | HSE_SRV_ID_FORMAT_KEY_CATALOGS | **Format key application key catalogs (RAM&NVM).** |

| Service Class | HSE Service ID | Description/Notes |
|---|---|---|
| | HSE_SRV_ID_ERASE_KEY | **Erase NVM/RAM key(s).**<br>Erase key service depends on authorization rights. One or multiple keys can be erased. |
| | HSE_SRV_ID_GET_KEY_INFO | **Get key proprieties** (flags). |
| | HSE_SRV_ID_IMPORT_KEY | **Import a key**.<br>Uses all algorithms supported by HSE firmware:<br>* Plain form or AES / RSA encrypted.<br>* MAC authenticated (refer to supported MAC algorithms) or RSA / ECDSA signed.<br>* Import key restrictions depends on sys authorization rights. The restrictions are described by the service in the interface. |
| | HSE_SRV_ID_EXPORT_KEY | **Export a key.**<br>Uses all algorithms supported by HSE firmware:<br>* Plain form (only public keys) or AES / RSA encrypted.<br>* MAC authenticated (refer to supported MAC algorithms) or RSA / ECDSA signed.<br>* Export key restrictions depends on authorization rights. The restrictions are described by the service in the interface. |
| | HSE_SRV_ID_KEY_GENERATE | **Request to generate a symmetric/asymmetric key.**<br>* Random symmetric key generation<br>* RSA and ECC key pair generation |
| | HSE_SRV_ID_DH_COMPUTE_SHARED_SECRET | **ECC Diffie-Hellman Compute Key (shared secret)**:<br>* SEC curves: SECP256R1<br>* Brainpool curves: BRAINPOOLP256R1<br>* Montgomery curve: CURVE25519<br>* 3 user-defined ECC curves (see Load ECC curve service) |
| | HSE_SRV_ID_KEY_DERIVE | **Perform a key derivation function:**<br>* NXP Generic KDF, Extract KDF, SP800_56C One Step, SP800_56C Two Step, SP800_108 (Only Counter Mode), ANS_X963, ISO/IEC 18033 KDF2, ISO/IEC 18033 KDF1, PBKDF2HMAC, HKDF, IKEV2, TLS12PRF |
| | HSE_SRV_ID_KEY_DERIVE_COPY | **Extract a key from the derived key material to a key slot.** |
| | HSE_SRV_ID_SHE_LOAD_KEY | **Load a SHE key** using the SHE memory update protocol. |
| | HSE_SRV_ID_SHE_LOAD_PLAIN_KEY | **Load the SHE RAM key** as plain text. |
| | HSE_SRV_ID_SHE_EXPORT_RAM_KEY | **Export the SHE RAM key.** |
| | HSE_SRV_ID_SHE_GET_ID | **Get UID as per SHE specification.** |
| | HSE_SRV_ID_SHE_BOOT_OK | The command is used to mark successful boot verification for later stages than CMD_SECURE_BOOT. For more details, see SHE specification |
| | HSE_SRV_ID_SHE_BOOT_FAILURE | The command will impose the same sanctions as if CMD_SECURE_BOOT would detect a failure but can be used during later stages of the boot process. For more details, see SHE specification. |

| Service Class | HSE Service ID | Description/Notes |
|---|---|---|
| **ROM Keys** | N/A | **Support for ROM keys (only AES keys).** |
| **Crypto** | HSE_SRV_ID_HASH | **Hash service (one-pass and streaming):**<br>* MD5<br>* SHA1<br>* SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256<br>* Miyaguchi-Preneel compression function (SHE specification support) |
| | HSE_SRV_ID_MAC | **Request to generate/verify a Message Authentication Code (MAC):**<br>* AES-CMAC, AES-GMAC, AES-XCBC-MAC<br>* HMAC_(MD5, SHA1, all SHA2) |
| | HSE_SRV_ID_FAST_CMAC | **Low latency, high performance CMAC generate/verify request** |
| | HSE_SRV_ID_SYM_CIPHER | **Symmetric encryption/decryption (one-pass and streaming):**<br>* AES-128/-192/-256: ECB, CBC, CTR, OFB, CFB |
| | HSE_SRV_ID_AEAD | **AEAD encryption/decryption**:<br>* AES-CCM-128/-192/-256 (one-pass, **no streaming support**)<br>* AES-GCM-128/-192/-256 (one-pass and streaming) |
| | HSE_SRV_ID_SIGN | **Request a Digital Signature Generation/Verification (one-pass and streaming):**<br>* RSASAA_PSS (1024, 2048, 3072, 4096)<br>* RSASAA_PKCS1-v1_5(1024, 2048, 3072, 4096)<br>* ECDSA (all supported ECC curves)<br>* EDDSA (for ED25519 curve) |
| | HSE_SRV_ID_RSA_CIPHER | **RSA encryption/decryption**:<br>* RSAES-PKCS1-v1_5 (1024, 2048, 3072, 4096)<br>* RSAES-OEAP (1024, 2048, 3072, 4096) |
| | HSE_SRV_ID_AUTHENC | **Combined Authenticated Encryption service:**<br>***AES_(ECB, CBC, CTR, CFB, OFB) -THEN- HMAC_ (MD5, SHA1, SHA2_224, SHA2_256, SHA2_384, SHA2_512) for "Encrypt-then-MAC"<br>*NULL cipher with all MAC algorithms (CMAC, GMAC, XCBC_MAC, HMAC(MD5, SHA1, SHA2)) |
| | HSE_SRV_ID_CRC32 | Computes CRC32 checksum. |
| | HSE_SRV_ID_SIPHASH | SipHash is optimized for fast processing speeds when used to authenticate small messages. (MACs) |
| | HSE_SRV_ID_XTS_AES_CIPHER | XTS AES encryption/decryption |
| **RNG** | HSE_SRV_ID_GET_RANDOM_NUM | **Get a random number.**<br>AIS31 and FIPS 140-2 compliant |
| **Counters** | HSE_SRV_ID_INCREMENT_COUNTER | **Incrementing volatile counters.**<br>The Counter table can be published and load as an encrypted and authenticated blob using HSE_SRV_ID_PUBLISH_LOAD_CNT_TBL service. |
| | HSE_SRV_ID_READ_COUNTER | **Read volatile counters.** |

| Service Class | HSE Service ID | Description/Notes |
|---|---|---|
| | | |
| **Advance Secure Booting (SMR/CR)** | HSE_SRV_ID_SMR_ENTRY_INSTALL | **Install a Secure Memory Region (SMR) table entry.** |
| | HSE_SRV_ID_SMR_VERIFY | **Verify (on demand) a Secure Memory Region (SMR) table entry.** |
| | HSE_SRV_ID_CORE_RESET_ENTRY_INSTALL | **Install a Core Reset (CR) table entry.** |
| | HSE_SRV_ID_ON_DEMAND_CORE_RESET | **On demand release a core from reset after loading and verification** |