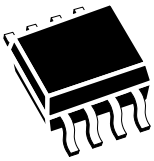
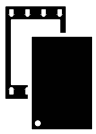
STSAFE-A120

Datasheet

Secure authentication companion device for consumables, accessories, and connected objects



SO8N 4 × 5 mm



UFDFPN8
2 × 3 mm

Product status
STSAFE-A120

Features

- Unique ID
- Authentication for:
  - Consumables and accessories anticloning
  - Connected objects secure connection and preattachment to clouds (Azure, AWS, and others)
  - Wireless chargers Qi 1.3 and Qi 2.0
  - Matter devices
  - Digital power supplies OCP M-CRPS
- Pairing and secure channel with host application processor
- Configurable secure storage
- Usage monitoring with secure counters
- Secure connection establishment with remote host including transport layer security (TLS 1.2 and TLS 1.3) handshake
- Signature verification service (secure boot and firmware upgrade)
- Secure storage in host nonvolatile memory based on wrapping and unwrapping of local host envelopes
- Data hashing
- Symmetric data encryption or decryption
- On-chip key pair generation

Cryptography and security features

- Advanced asymmetric cryptography
  - 5 Elliptic curve crytpography (ECC), nonvolatile private key slots + 1 ephemeral ECC key slot
  - Supported elliptic curves:
    - NIST P-256 P-384, P-521
    - Brainpool P-256 P-384, P-512
    - Edwards 25519
    - Curve25519
- Supported functionalities:
  - Digital signature generation and verification (ECDSA and EdDSA)
  - Diffie-Hellman shared secret establishment (ECDH)
- Advanced symmetric cryptography
  - 16 slots of symmetric cryptography with AES-128/256 CCM*, ECB, GCM, CMAC, and HKDF
- Pairing with host-applicative processor
  - AES 128-bit or 256-bit
- Local wrap/unwrap envelop key
  - 2 slots of keys with AES 128-bit or AES 256-bit
- Data hashing
  - SHA-2 with SHA-256, SHA-384, SHA-512
  - SHA-3 with SHA3-256, SHA3-384, SHA3-512

- Random number generator
  - Random number generator with NIST SP 800-90B compliant entropy source
- Latest generation of highly secure MCUs
  - Unique serial number on each die
  - CC EAL5+ AVA_VAN.5, and ALC_DVS.2 Common Criteria certified
  - Active shield
  - Monitoring of environmental parameters
  - Protection mechanism against fault injection
  - Protection against side-channel attacks

Hardware characteristics

- 16 Kbytes of configurable nonvolatile memory
  - 25 years of data retention at 25°C
  - 500 000 erase/write cycle endurance at 25°C
- 2.7 V to 5.5 V continuous supply voltages
- Operating temperature: -40°C to +105°C

Communication protocol

- I²C - bus slave interface
  - Up to 400 kbps transmission speed (Fast mode)
  - 7-bit addressing

Packages

- ECOPACK-compliant SO8N 8-lead plastic small outline package and UFDFPN 8-lead ultrathin profile fine pitch dual flat package.

1 Description

The STSAFE-A120 is a secure element, providing authentication and secure data management services to a local or remote host. It also provides cryptographic services like hashing, encryption, and decryption. It consists of a full turnkey solution with a secure operating system running on the latest generation of secure microcontrollers.

The STSAFE-A120 can be integrated in consumables, accessories, IoT (Internet of Things) devices, smart-home, smart-city and industrial applications, and consumer electronics devices.

LICENSED DPA COUNTERMEASURES™

1.1 Key function overview

Figure 1. Authentication to a remote server (connected device case)

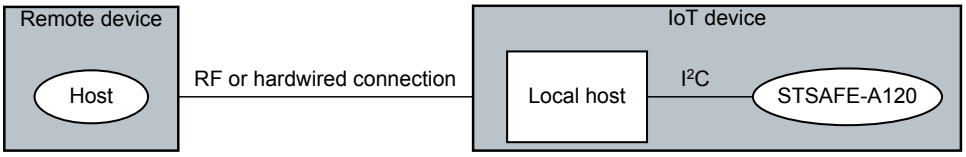
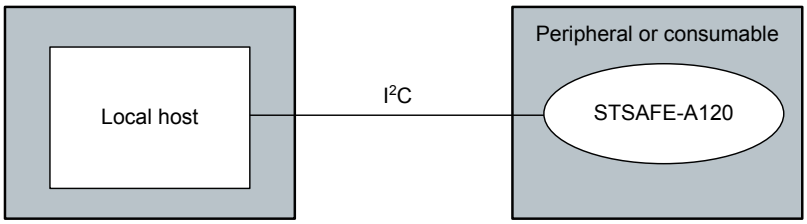


Figure 2. Authentication to a local host (consumable or peripheral case)



The STSAFE-A120 can be mounted on:

- A device that authenticates to a remote host (IoT device case), the local host being used as a pass-through to the remote server.
- A peripheral that authenticates to a local host, for example games, mobile accessories, or consumables.

The STSAFE-A120 secure element supports the following features:

- **Authentication**
  The STSAFE-A120 authentication service provides proof to a remote or local host that a certain peripheral or IoT is legitimate. An equipment manufacturer can thus ensure that only authentic peripherals (like accessories or consumables) can be used in conjunction with the original equipment. In the same way, a service provider can make sure that its service only operates with the appropriate IoT device.
  The authentication service utilizes the ECC cryptographic scheme with NIST P-256-bit, P-384-bit, P-521-bit curves, brainpool P-256-bit, P-384-bit, P-512-bit, and curve25519. It is also compliant with various standards that request object authentication such as CSA Matter, WPC, and OCP M-CRPS.
- **Secure storage**
  The STSAFE-A120 comes with 16 Kbytes of nonvolatile memory split into areas, whose read and write access rights can be configured to free access, local host access, or remote host access.
- **Secure one-way counters (peripheral life cycle and usage monitoring)**
  The STSAFE-A120 offers configurable one-way counters that allow the monitoring of disposable accessories or consumables. The number of counters available in the user NVM depends on the STSAFE-A120 personalization.
- **Pairing and secure channel with the host**
  The STSAFE-A120 allows a secure channel to be set up with the local host based on AES keys for command authorization, command data encryption, response data encryption, and response authentication. Typically, pairing between an STSAFE-A120 and its local host prevents the use of the STSAFE-A120 in a different device and protects the I²C line from eavesdropping of sensitive information.
- **Wrapping and unwrapping local envelopes**
  The STSAFE-A120 can be used to encrypt or decrypt data with one of its two local envelope keys. Typically, it can be used when the local host needs to store secrets like connectivity keys and credentials within its nonsecure data storage area.
- **Secure-channel key establishment (TLS)**
  The STSAFE-A120 assists the local host in establishing a secure connection between the device and a remote host (such as a cloud server or gateway). It assists the local host in establishing the session keys used to encrypt and decrypt data exchanges between the device and the remote host. This key establishment service relies on the computation of a shared secret based on the elliptic curve Diffie-Hellman schemes (ECDH and ECDHE) executed after the device has generated and exchanged ECC NIST, brainpool, or X25519 public keys with the server.
- **Entity authentication**
  With its public key slots, the STSAFE-A120 can authenticate a local or a remote host. Successful authentication by the STSAFE-A120 grants the local or remote host access to some authorized commands, or memory partitions.
- **Signature verification**
  The STSAFE-A120 can verify an elliptic curve digital signature algorithm (ECDSA) signature by using a public key provided by the local host. This mechanism can offload a local host application processor with no or limited cryptographic computing power. It is typically used to verify signatures of firmware in the context of secure boot or secure firmware update.
- **Symmetric keys**
  The STSAFE-A120 can be loaded with up to 16 symmetric keys for data encryption and decryption.
- **Data hashing**
  The STSAFE-A120 allows data hashing with SHA-2 (SHA-256, SHA-384, SHA-512), and SHA-3 (SHA3-256, SHA3-384, SHA3-512) algorithms.

1.2 STSAFE-A120 environment

The STSAFE-A120 comes with a host integration code tested with STMicroelectronics (ST) STM32 general-purpose MCUs. It can also be ported to a wide range of general-purpose microcontrollers or microprocessors. This integration code includes a command wrapper and examples for the most common generic use cases.

The STSAFE-A120 is available prepersonalized with generic data profiles for evaluation, prototyping, or production.

ST also offers and recommends secure provisioning services for key generation and storage of customer leaf certificates in a secure, certified environment.

The STSAFE-A120 are delivered with an ST CA certificate that allows authenticity verification of the leaf certificates present in each STSAFE-A120 device.

2 Product use cases

This section illustrates the many uses of an STSAFE-A120 device using asymmetric cryptography.

2.1 Authentication

This scenario illustrates the command flow where the STSAFE-A120 is mounted on a device that authenticates to a remote host (IoT device case), the local host being used as a pass-through to the remote server.

The scenario where the STSAFE-A120 is mounted on a peripheral that authenticates to a local host, for example games, mobile accessories or consumables, is exactly the same.

Command flow

1. Obtain the public key of the STSAFE-A120 chip in the host device:
   - *Command 1* is used to read the X509 public key certificate from the data partition of the STSAFE-A120 chip.
   - The host device verifies the X509 public key certificate with the CA public key (the host is responsible for getting a copy of this key). When the verification process succeeds, the host device has an authentic copy of the STSAFE-A120 public key that it is used later on for verification of the signature.
2. The host device generates a challenge and stores it for later use in the verification of the signature. The host device then computes a hash of this challenge and sends it to the STSAFE-A120 in *command 2* to fetch the signature that the STSAFE-A120 chip computed with its private key. The host device verifies the signature with the STSAFE-A120 public key (obtained in the first step of this scenario). When valid, the host knows that the peripheral or IoT device is authentic.

Figure 3. Example of IoT device authentication

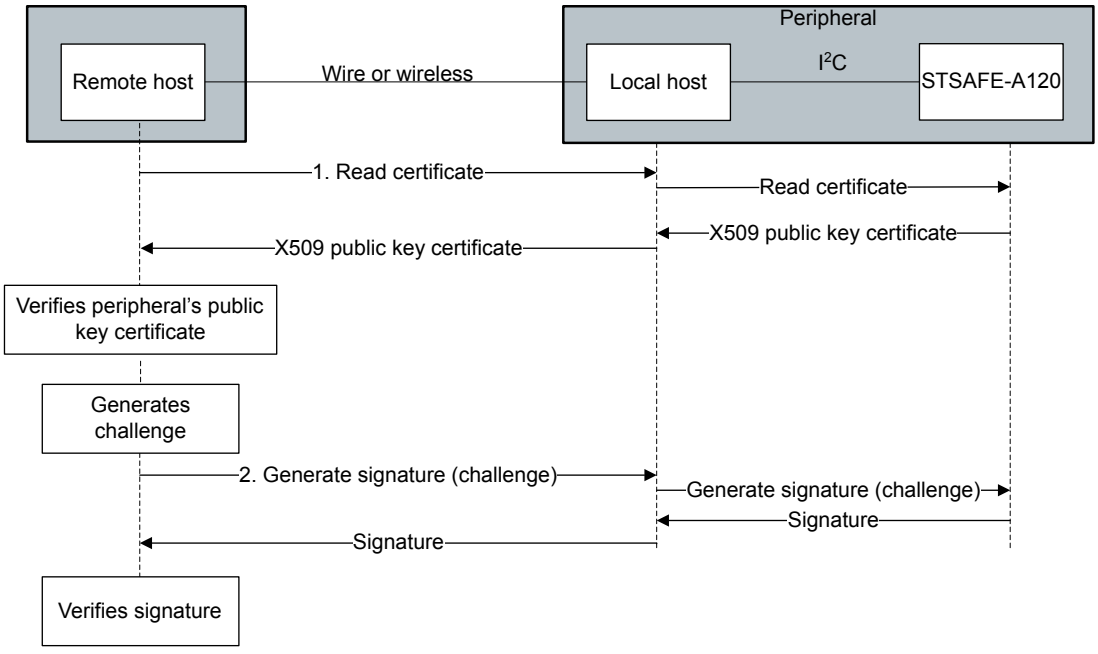
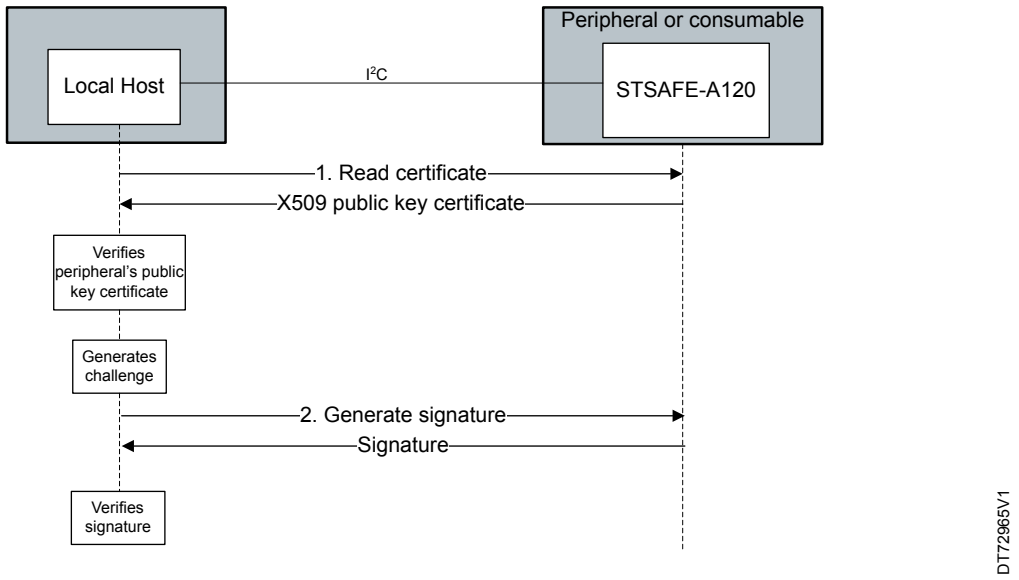


Figure 4. **Example of peripheral device authentication**



2.2 Applicative data storage

The STSAFE-A120 comes with 16 Kbytes of nonvolatile memory configurable by the customer for its application data storage. These 16 Kbytes must be partitioned into zones with the appropriate access rights.

Some of the zones can be configured into secure one-way counters (decrementing) zones with associated data space.

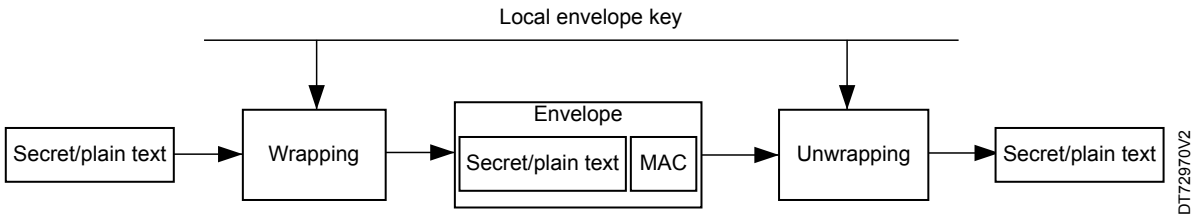
The access condition to the memory zones can be configured as well for each zone. The access condition can be:

- Free access
- Host secure channel access
- Entity authentication access
- Host secure channel and entity authentication access.

2.3 Local envelope wrapping/unwrapping

The STSAFE-A120 offers a data wrapping service (cyphering and signing data) into a secure envelope. This service is intended to be used by a host to store sensitive information in a non protected memory. This envelope can be unwrapped (unciphered and verified) at anytime by the STSAFE-A120.

Figure 5. **General principle of the wrapping/unwrapping key**



Wrapping is the mechanism used to protect a secret or plain text (like connectivity keys or credentials). The output of the wrapping is an envelope.

The envelope consists of the secret or plain text to be protected, encrypted with an AES key wrap algorithm. The algorithm uses a local envelope key for a local envelope. The envelope also contains the MAC of the encrypted key or plain text to authenticate the envelope.

Unwrapping is the mechanism used to decrypt the envelope and recover the secret or plain text.

The secret or the plain text can be sent to the STSAFE-A120 in the command data of the WRAP LOCAL ENVELOPE command. In the response, the STSAFE-A120 returns an envelope, which contains the encrypted secret or plain text and a MAC. Such an envelope is known as a local envelope.

The local host can use the UNWRAP LOCAL ENVELOPE command to retrieve the temporary secret or plain text. The wrapping and unwrapping processes utilize a key from one of the local envelope key slots and the AES key wrap algorithm.

Local envelope key slots

The STSAFE-A120 supports two local envelope key slots.

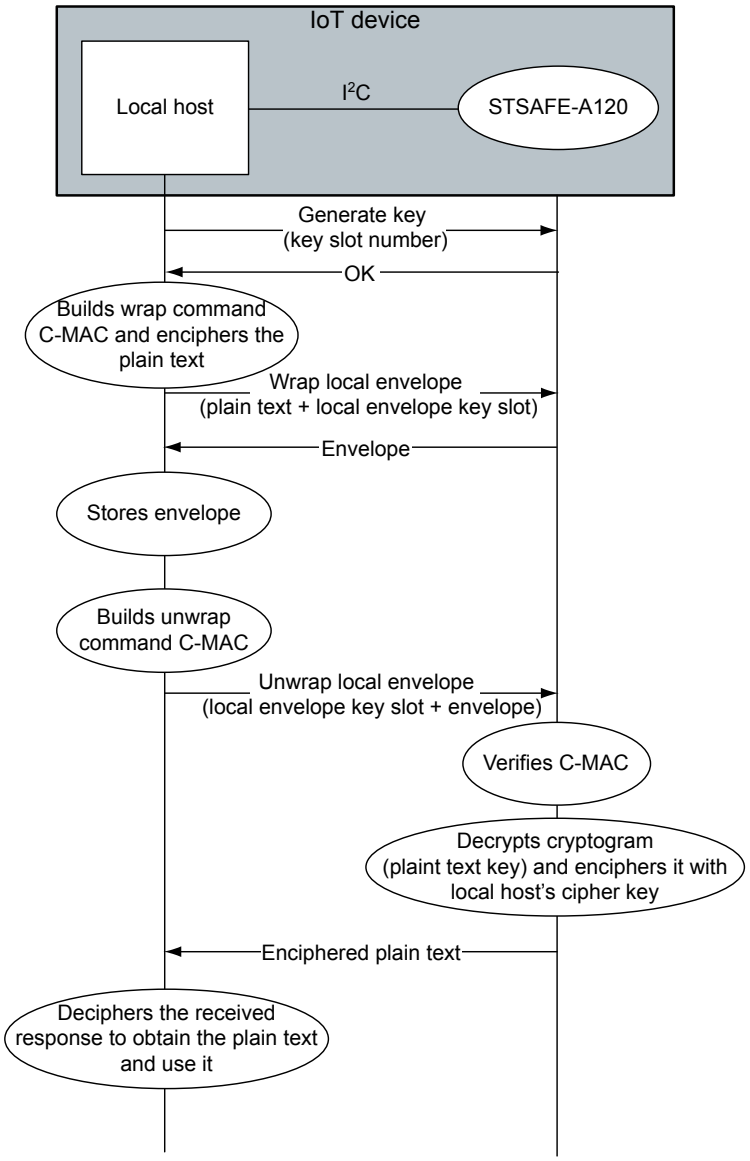
Each slot can store an AES-128- or AES-256-bit key that can be used for the wrapping and unwrapping of local envelopes.

A local envelope key is generated randomly by the STSAFE-A120 with the GENERATE KEY command, and it never leaves the STSAFE-A120.

Command flow

1. Generation of the local envelope key:
   - The local host queries the STSAFE-A120 to randomly generate a local envelope key in one of the two slots, using the GENERATE KEY command.
2. Wrapping of local envelope:
   - The local host builds the local envelope by using the STSAFE-A120 wrap local envelop command.
   - This command requires a local host's C-MAC, plain text data (usually a cryptographic key that needs to be encrypted with the host's cipher key) and the local envelope key slot number (the key to use for encryption of plain text data).
   - The response to the *wrap local envelope* command contains the envelope (plain text is encrypted using the *local envelope* key).
3. Unwrapping of local envelope by receiver:
   - The local host provides the local envelope to the STSAFE-A120 by using the *unwrap local envelop* command and the local envelope key slot.
   - The host must generate a local host's C-MAC with the command.
   - The STSAFE-A120 provides the envelope cryptogram (usually a cryptographic key) decrypted with the local envelope key) in its response.
   - The response is encrypted using the host's cipher key.
   - The host decrypts the response with the host's cipher key and obtains the decrypted envelope cryptogram.

Figure 6. **Wrap/unwrap local envelop command flow**



2.4 Key establishment for secure connection (TLS)

This use case shows how to generate the same shared secret in the local host and in the remote host without having to exchange it. The principle consists in generating two ECC ephemeral key pairs on both sides. Then, after exchanging the public keys of these two key pairs, the local host and the remote host run an ECDH scheme to calculate the shared secret.

The goal of this use case is to share a secret between the local host and the remote server using the elliptic curve Diffie-Hellman (ECDH) scheme with a static key in the STSAFE-A120. The STSAFE-A120 also supports ECDHE that utilizes an ephemeral key, but this is not illustrated.

The shared secret should further be derived to one or more session keys, but this is not illustrated here. The session keys can then be used in communication protocols like TLS for confidentiality, integrity, and authenticity of the data that are exchanged between the local host and the remote server. Below are some examples of data:

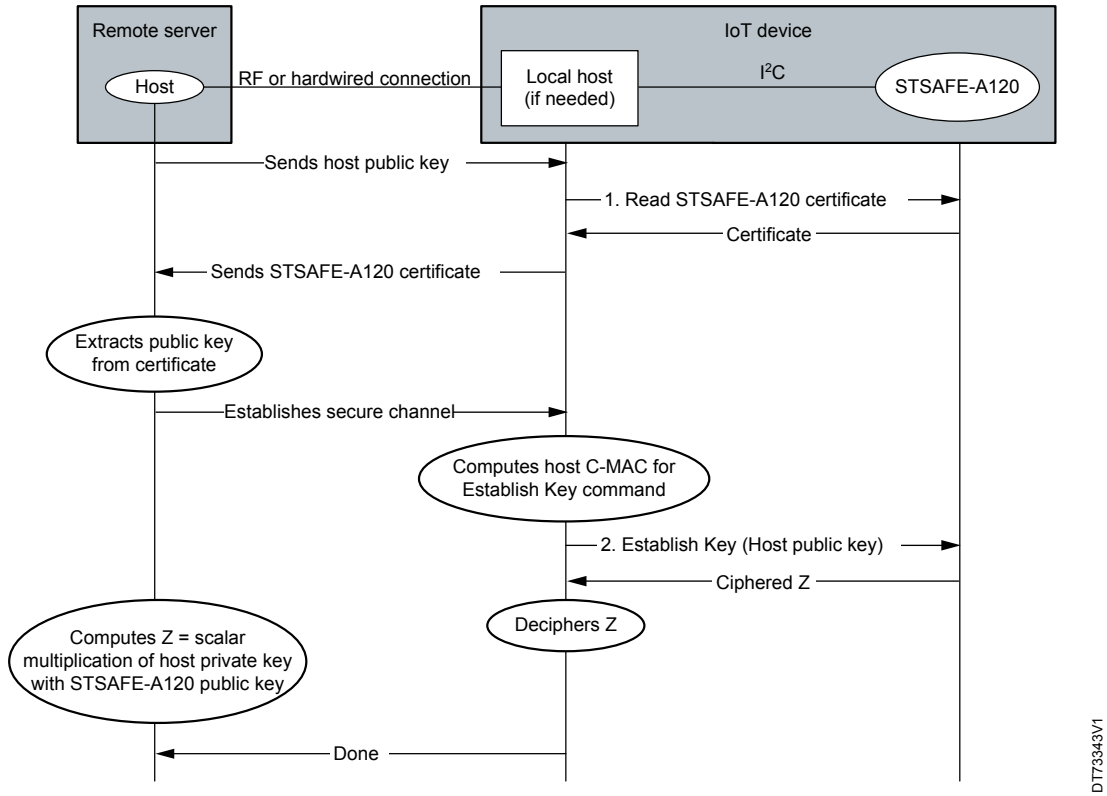
- From the local host to the remote server: power consumption of a smart meter, alarm of a fire sensor or blood pressure data of a health sensor.
- From the remote server to the local host: activating the recharge of the battery of an electric car, activating home appliances like air conditioning or water heaters, or pushing firmware upgrades to IoT devices.

Depending on the personalization profile, the *establish key* command needs to be MACed and its answer is encrypted to avoid eavesdropping on the shared secret. The scenario assumes that the local host has set up a host C-MAC and cipher keys as described in Section 3: Pairing with local host. It is also assumed that the local host knows the host C-MAC sequence counter; if not, it can send a *query* command to the STSAFE-A120.

Command flow

1. The remote host server sends its certificate to the local host. The local host extracts the public key and can optionally verify the validity of the certificate. In its response, the local host sends the STSAFE-A120 certificate.
2. The remote server verifies the STSAFE-A120 X.509 public key certificate with the CA public key (the host is responsible for getting this key). When the verification succeeds, the remote server has an authentic copy of the STSAFE-A120 public key.
3. The remote server then computes a shared secret (Z) by doing a scalar multiplication of the host's private key with the STSAFE-A120 public key.
4. The remote server requests the local host to establish a secure connection.
5. The local host computes the host's C-MAC for the *establish key* command
6. The local host sends the STSAFE-A120 an *establish key* command providing the remote host's public key appended with the previously computed host's C-MAC. The STSAFE-A120 does the same operation as the remote host server, and performs the scalar multiplication of its private key with the remote server's public key to compute the shared secret (Z). It then encrypts the response using the host's cipher key.
7. The local host reads the STSAFE-A120 answer and deciphers the shared secret (Z) with the locally stored host's cipher key.
8. The remote host server and the local host have a shared secret Z.

Figure 7. Key establishment command flow



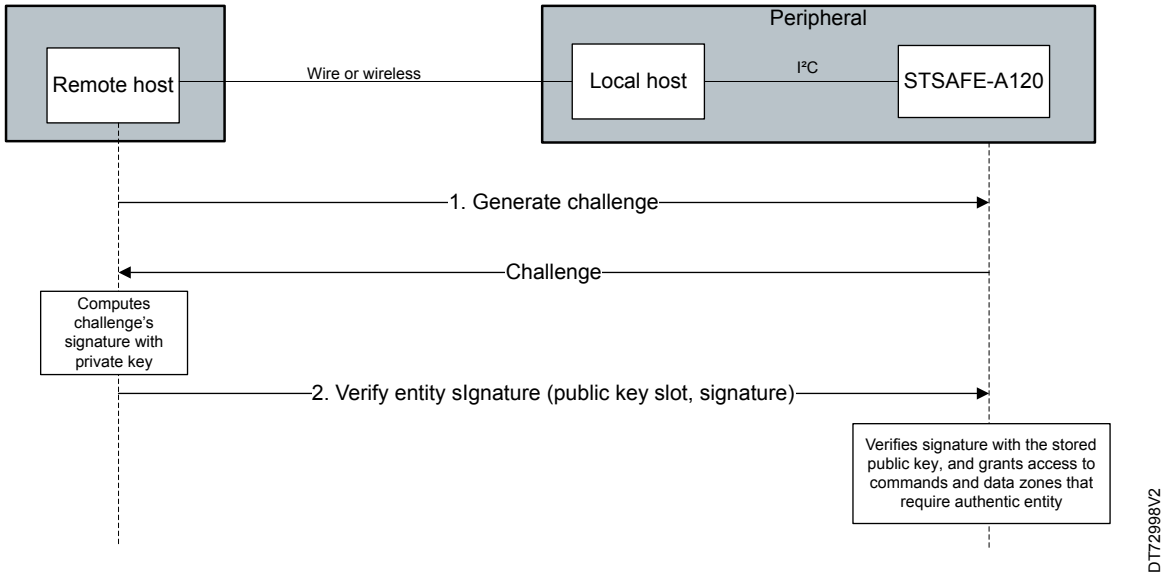
2.5 Entity authentication

An STSAFE-A120 device can authenticate an off-chip entity by verifying a digital signature generated by this entity with a private key. This is done over a challenge that was previously generated by the STSAFE-A120.

Once the entity has been authenticated, the STSAFE-A120 grants access to the commands and to the data partition zones that require the authentic entity status to be true.

This functionality requires that the STSAFE-A120 contains an authentic copy of the public key that corresponds to the private key used by the signature generation process of the off-chip entity.

Figure 8. Example of entity authentication



2.6 Public key signature verification

STSAFE-A120 offers signature verification services for host devices that does not support ECC. These services can be used to allow an IoT device to verify the authenticity of a remote server or of the local host firmware that is used at secure boot or loaded for a firmware update.

The *verify signature* command uses ECDSA or EdDSA with curves defined below:

- Curves for usage with ECDSA, EdDSA, and ECDH:
  - NIST P-256 P-384, P-521
  - Brainpool P-256 P-384, P-521
  - Edwards 25519
  - Curve25519

Signature verification with EdDSA is implemented as defined in RFC8032. The host must indicate in the command data whether the verification must be done with the pure Ed25519 variant in which case the host must give the message in the command data or with the prehashed Ed25519 variant in which case the host must itself compute the hash of the message and give the hash in the command data.

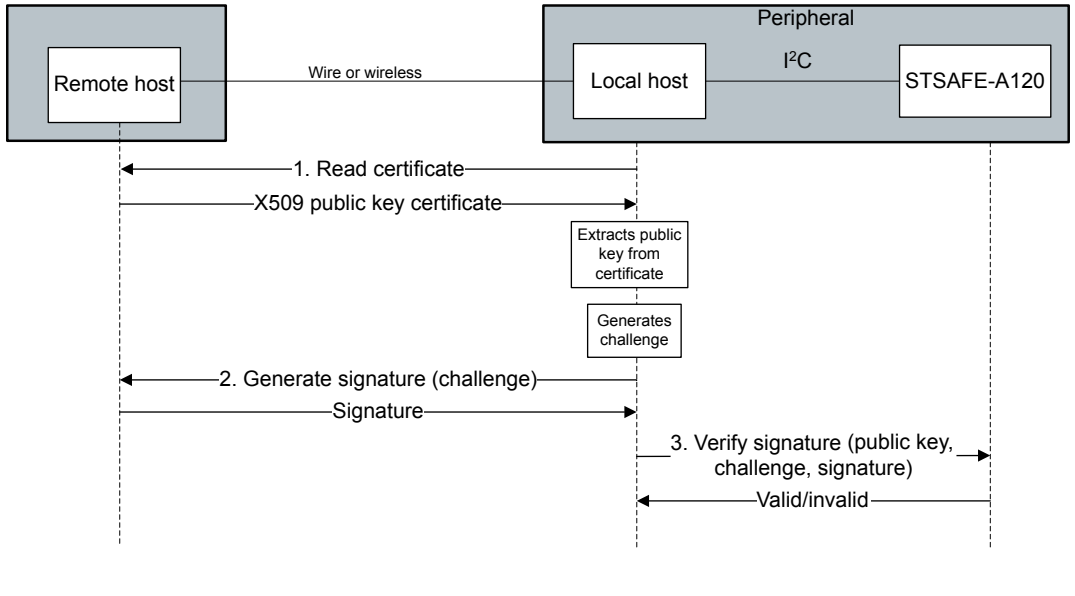
The host must always give the public key and the hash in the command data except for the pure Ed25519 signature verification where the message itself (not the hash) must be given in the command data.

One use case of the *verify signature* command is illustrated in the command flow and in the figure below where a local host authenticates a remote host using STSAFE-A120.

Command flow

1. The local host asks for the remote host public key certificate and the remote host responds.
2. The local host extracts the public key, generates a challenge (random number), and asks the remote host to sign this challenge. The remote host responds with the signature.
3. STSAFE-A120 verifies the challenge's signature using the extracted public key, and replies with "valid" or "invalid".

Figure 9. **Public key signature verification command flow**



2.7 Symmetric signature, verification, encryption, and decryption with keys from the symmetric key table

The STSAFE-A120 offers 16 slots for the storage of AES keys (AES-A128 or AES-256). These 16 slots are known as the symmetric key table.

Each of the keys can be configured independently to support one of the following modes of operations:

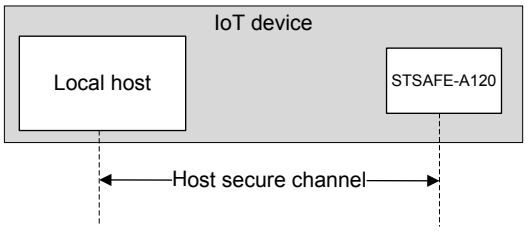
- Encryption and decryption
  - AES in CCM*/CTR mode
  - AES in CBC mode
  - AES in ECB mode
  - AES in GCM/GMAC mode
- Key derivation function (KDF)
  - HKDF key derivation with HMAC based on a SHA2 or SHA3 hash function
- MAC computation:
  - AES in conjunction with the CMAC construction
  - HMAC computation based on SHA2 or SHA3 hash function

The mode of operation of a key must be written with the WRITE_SYMMETRIC_KEY or the CONFIRM_SYMMETRIC_KEYS command. A key can only support one mode of operation at a time.

3 Pairing with local host

To protect and authenticate the data exchange between the STSAFE-A120 and the local host, a secure channel protocol using symmetric cryptography is set in place, namely the host secure-channel.

Figure 10. Host secure channel



The host secure-channel protocol constitutes the pairing. It is based on a set of four mechanisms using two symmetric keys, the so-called host keys. The host MAC key is used to compute and verify message authentication codes (MAC) for the commands (C-MAC) and respective responses (R-MAC). The host cypher key is used to encrypt the commands and decrypt their respective response to avoid eavesdropping.

The host MAC key and the host cipher key must be shared between the host and the STSAFE-A120.

The STSAFE-A120 supports AES-128 or AES-256 keys, and uses an incremental 32-bit C-MAC sequence counter (the V2 slot type) for host key storage.

The C-MAC sequence counter determines the limitation of the usage of the host keys.

The number of C-MAC operations is limited to 2^{32} - 1 operations. After that, the commands requiring a C-MAC will fail. There is no mechanism to reset the counter.

When the STSAFE-A120 receives an invalid host C-MAC, it increments a ratification counter called the host C-MAC ratification counter. When the counter reaches 50, the STSAFE-A120 refuses to use its host MAC key. Thus, commands requiring a host C-MAC are refused. When the STSAFE-A120 receives a valid host C-MAC and before the host MAC keys are blocked, it resets the host C-MAC ratification counter to 0.

The host keys can be provisioned with either:

- Plaintext: The keys are sent by the host in plaintext within the command
- Wrapped: The keys are sent wrapped with a working KEK (key encryption key) that is a one-time use key derived from a volatile base KEK established by an ECDHE process.

Command flow for plaintext host keys provisioning

This use case assumes that the slots are empty.

1. The local host requests the STSAFE-A120 to generate a 128-bit random to be used as the host C-MAC key.
2. The local host requests the STSAFE-A120 to generate a 128-bit random to be used as the host cipher key.
3. The local host sends the PUT ATTRIBUTE command for the “Host key slot” attribute, together with the two generated keys (forming a 256-bit payload).
4. The STSAFE-A120 stores the keys into their respective slots and returns a successful response.
5. The local host stores the host C-MAC and cipher keys to a secure area.

Figure 11. Host secure channel setup case for plaintext host keys provisioning

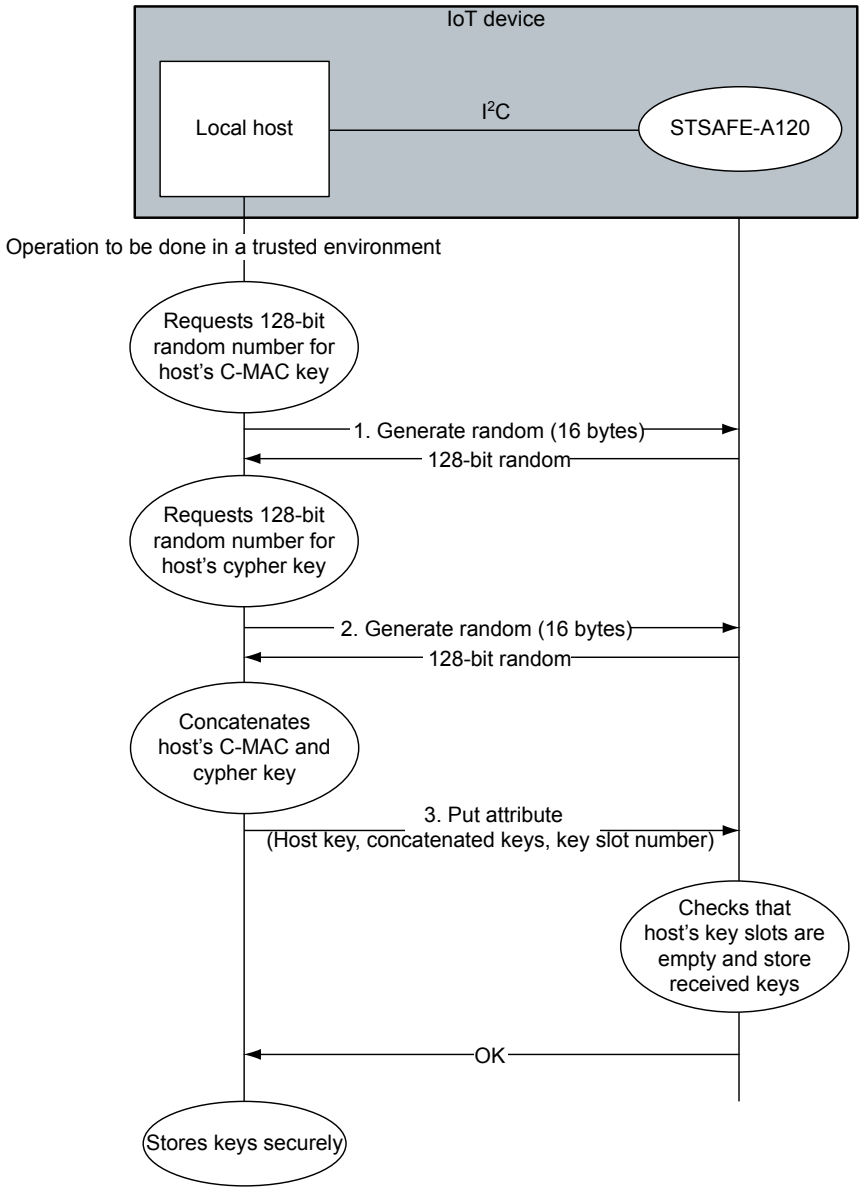
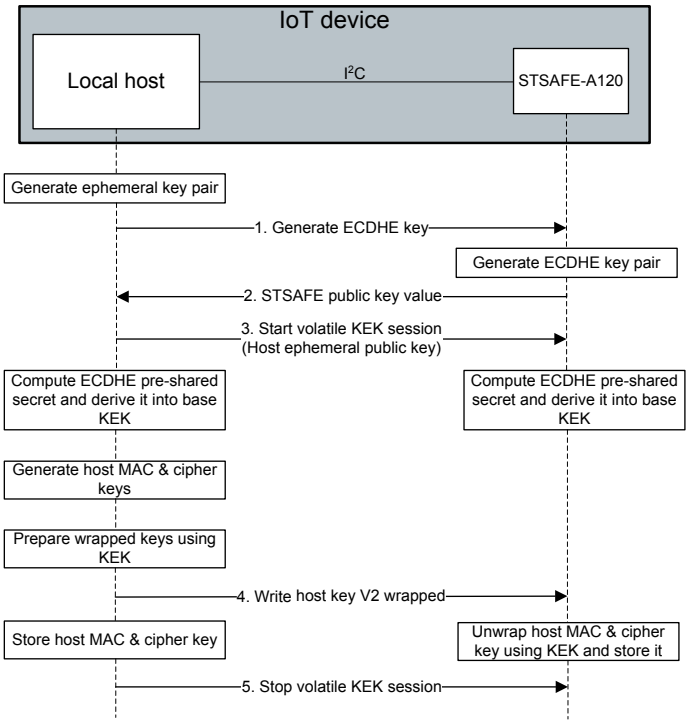


Figure 12. Host secure channel setup case for wrapped host keys provisioning



4 Command sets

4.1 General-purpose commands

Echo

Returns as a response the data that it received as command data.

Reset

Interrupts any ongoing session.

Put attribute

Used to put attributes in the STSAFE-A120 like low-power mode and I²C parameters.

4.2 Random generation

Generate random

Returns the requested number of random bytes.

Generate challenge

Returns a random byte string that can be signed by an off-chip entity to get authenticated by the STSAFE-A120 (i.e. entity authentication).

4.3 Data hashing commands

Start hash

Starts a hash computation over a message.

Process hash

Continues hash computation.

Finish hash

Ends a hash computation and returns the digest.

4.4 Private and public key commands

Write public key

Writes a public key in the generic public key table. Each slot can be written only once.

Generate signature

Signs a remote host challenge or a data hash with either an ECDSA or and EdDSA signature depending on the curve in the selected private key slot.

Verify signature

Used for message authentication. It verifies the remote host message signature with the provided public key.

Generate key pair

Generates a key pair.

Establish key

It is used to establish a shared secret (TLS 1.3 compliant) with a specific remote host using ECHE protocol.

Start signature session

Starts an asymmetric signature session. It notifies the STSAFE-A120 that a signature computation over the commands or responses sequences must be initiated.

Get signature

Returns the signature on the commands or responses sequence since the latest start of a signature session.

Verify entity signature

Verifies a signature generated by an off-chip entity over a challenge previously generated by the STSAFE-A120 with the **generate challenge** command.

Decompress public key

Supports the decompression of a NIST or brainpool public key available in compressed form with its X coordinate. It returns the Y coordinate of the public key.

4.5 Local envelope commands

Generate local envelope key

Generates local envelope key.

Wrap local envelope

This command is used to wrap data with a local key envelope using an AES key wrap algorithm.

Unwrap local envelope

This command is used to unwrap a local envelope with a local envelope key.

4.6 Data-partition commands

Decrement

Decrements the one-way counter in a counter zone. When the counter reaches zero, the command is refused.

Read

Reads data from a data partition zone. It reads the data starting from the specified offset within the zone and with the requested length. It checks the access conditions (for example the MAC) and only returns the data starting from the specified offset up to the zone boundary.

This command can also be used to change the read access conditions of the zone to a stricter value.

Update

Updates data in a zone. It checks if the written data is expected to exceed the zone boundary and if so, does not perform the operation. It also checks whether the access condition is satisfied (for example the MAC) and if not, does not perform the operation.

This command can also be used to change the update access conditions of the zone to a stricter value.

4.7 Symmetric key table commands

Write symmetric key plaintext

Supports the provisioning of a slot in the symmetric key table with a key in plaintext format

Write symmetric key wrapped

Supports the provisioning of a slot in the symmetric key table with a wrapped key.

Establish symmetric keys

Supports provisioning of new keys in the symmetric key table using the ECDHE protocol.

Confirm symmetric keys

Supports the provisioning and confirmation of new keys in the symmetric key table in combination with the **establish symmetric keys** command.

Derive keys

Supports the derivation of one or more output keys from an input key present in the symmetric key table or passed as a part of the command data.

Erase symmetric key slot

Erases the content of the specified key slot in the symmetric key table. This is required when the slot has been provisioned with the **derive keys** command and when the key slot is not locked.

Encrypt

Encrypts data with one of the AES keys in the symmetric key table.

Decrypt

Decrypts data with one of the AES keys in the symmetric key table.

Generate MAC

Signs data with one of the AES keys in the symmetric key table.

Verify MAC

Verifys data with one of the AES keys in the symmetric key table.

Start encrypt

Encrypts long messages of data in chunks with one of the AES keys in the symmetric key table. It must be used in combination with the **process encrypt** and the **finish encrypt** commands.

Process encrypt

Continues the encryption process. It can be used multiple consecutive times to encrypt several chunks of data with one of the AES keys in the symmetric key table. It must be used in combination with the **start encrypt** and the **finish encrypt** commands.

Finish encrypt

Ends the encryption process of long messages of data in chunks with one of the AES keys in the symmetric key table. It must be used in combination with the **start encrypt** and the **process encrypt** commands.

Start decrypt

Decrypts long messages of data in chunks with one of the AES keys in the symmetric key table. It must be used in combination with the **process decrypt** and the **finish decrypt** commands.

Process decrypt

Continues the decryption process. It can be used multiple consecutive times to decrypt several chunks of data with one of the AES keys in the symmetric key table. It must be used in combination with the **start decrypt** and the **finish decrypt** commands.

Finish decrypt

Ends the decryption process of long messages of data in chunks with one of the AES keys in the symmetric key table. It must be used in combination with the **start decrypt** and the **process decrypt** commands.

5 Standby mode

The STSAFE-A120 supports a standby mode, with a dedicated attribute. This attribute can be either set to none or to standby with the PUT_ATTRIBUTE[LOW POWER MODE] command.

When the STSAFE-A120 is in standby mode ($I_{CC-STDBY}$), it NACKs the incoming I²C START condition until $t_{WAKE-STDBY}$.

Figure 13. STSAFE-A120 wake up from STANDBY

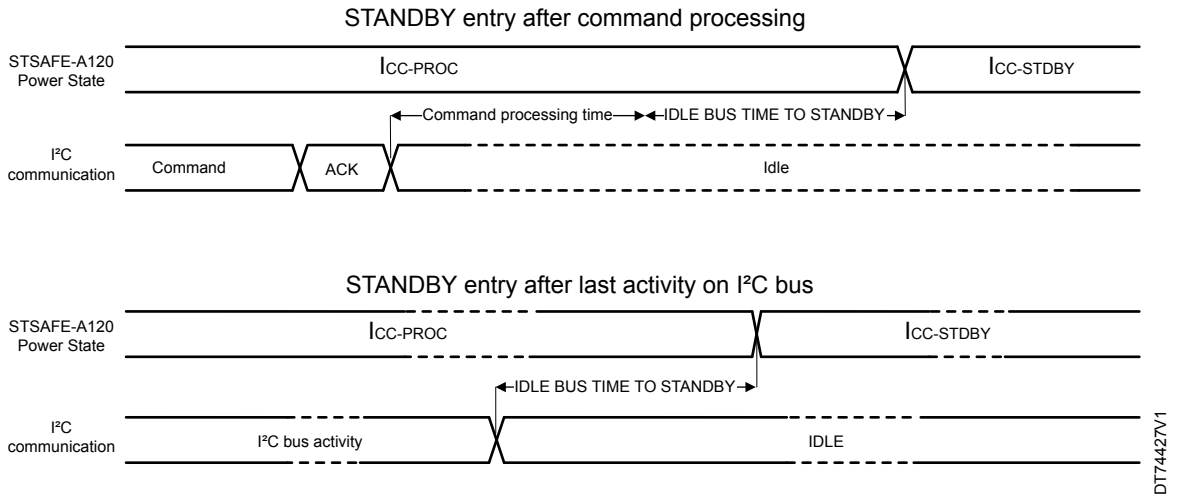
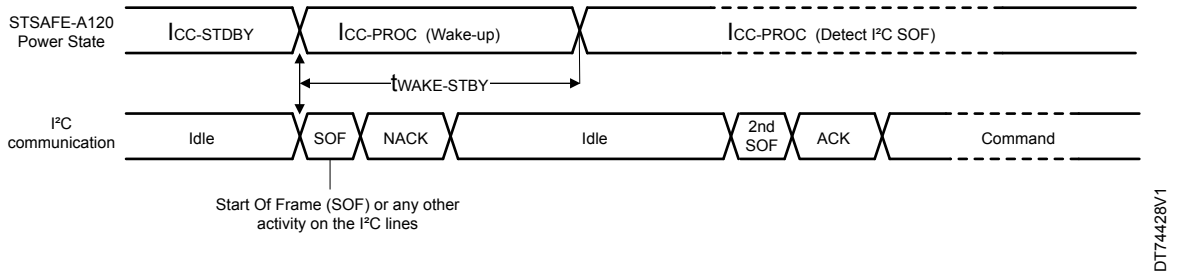


Table 1. STSAFE-A120 wake up time from standby

Name	Description	Min	Typ	Max	Units
$t_{WAKE-STDBY}$	Wake-up time from STANDBY	-	60	-	µs

If LOW POWER MODE is enabled, the STSAFE-A120 automatically enters in standby mode after IDLE BUS TIME TO STANDBY milliseconds following the scheme illustrated in Figure 14.

Figure 14. STSAFE-A120 standby entry



The polling time from the host shall be greater than $t_{WAKE-STDBY}$, and lower than IDLE BUS TIME TO STANDBY to avoid a dead loop.

IDLE BUS TIME TO STANDBY is adjustable, starting at 50 ms, and up to 1600 ms with 50 ms steps.

In case of power optimization, disabling the standby mode is not recommended when the bus is shared with other I²C devices.

6 Electrical integration

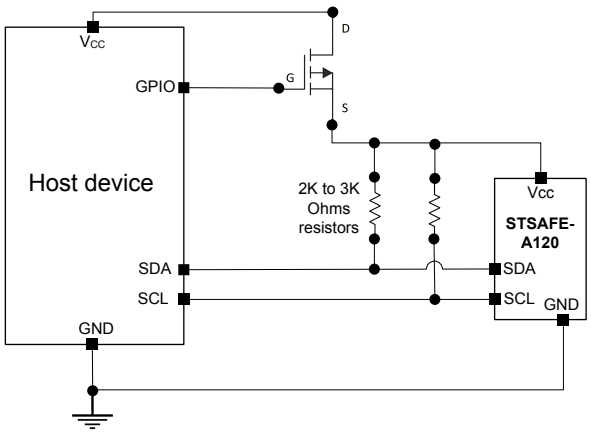
When the STSAFE-A120 is not used, it is possible to power off the device by controlling the V_{CC} pin. There are two possible ways to control or supply the STSAFE-A120 V_{CC} pin: through a transistor, or through the GPIO. These two methods are explained in the following sections.

6.1 V_{CC} control through a transistor

A transistor, controlled through GPIO can be used to control the STSAFE-A120 supply. By using this method, the STSAFE-A120 is powered directly from system V_{CC} domain.

I²C pull-up resistors must be connected on the same power domain than the STSAFE-A120 V_{CC} pin.

Figure 15. STSAFE-A120 V_{CC} pin control with a transistor



DT74401V1

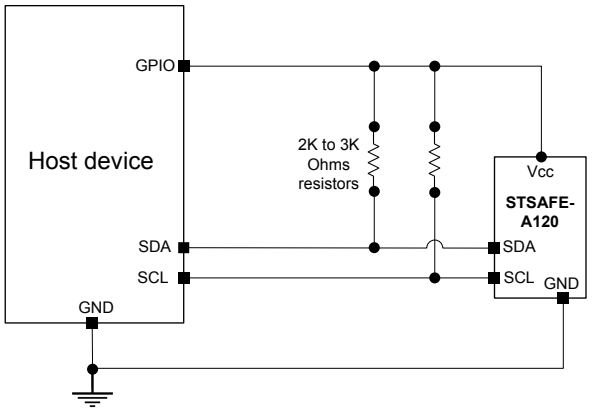
6.2 V_{CC} supply through a GPIO

If the STSAFE-A120 is on a dedicated bus, it can be powered through a host GPIO.

The following directions must be followed:

- Do not exceed the maximum source current of the GPIO
- Do not exceed the total acceptable current allowed by the host device. Refer to the electrical characteristics of the host device.

Figure 16. STSAFE-A120 V_{CC} pin supply through a GPIO



DT74402V1

7 Electrical characteristics

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristic tables that follow are derived from tests performed under the measurement conditions summarized in the relevant tables. Users should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

7.1 Absolute maximum ratings

Operation of STSAFE-A120 at ranges above the absolute maximum specifications may cause permanent device damage. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 2. Absolute maximum ratings

Name	Description	Conditions	Min	Typ	Max	Units
$V_{CC\ ABS}$	Absolute maximum power supply	Pins: VCC	-0.3	-	6.5	V
V_{IO}	Input or output voltage relative to ground	-	-0.3	-	VCC +0.3	V
V_{ESD}	Electrostatic discharge voltage according to ANSI/ESDA/ JEDEC JS-001	Human Body Model	-	6000	-	V
T_A	Ambient operating temperature	-	-25	-	85	°C
T_{STG}	Storage temperature	-	-40	-	125	°C
T_{LEAD}	Lead temperature during soldering(1)	-	-	-	260	°C

1. *SO8N and UFDFPN8 lead temperature during soldering shall be compliant with JEDEC Std J-STD-020D (for small body, Sn-Pb or Pb assembly), ST ECOPACK® 7191395 specification, and the European directive on restrictions on hazardous substances (ROHS directive 2011/65/EU, July 2011).*

7.2 Power supply

7.2.1 Power supply specifications

The table below provides the detailed description of the power requirements of STSAFE-A120.

Table 3. Power supply specifications

Name	Description	Conditions	Min	Typ	Max	Units
V_{POR}	Power on reset voltage	-	-	-	2.05	V
V_{CC}	Supply voltage	Considering a +/- 10% ripple on the V_{CC} line(1)	2.7	-	5.5	V
$V_{CC-HIPS}$	High power supply detection	-	5.7	6	-	V
$I_{CC-PROC}$	Supply current while processing a command	-	-	10	11.5	mA
$I_{CC-STDBY}$	Standby current	IO pulled up to VCC, TA = 25°C, 3 V	-	270	500	µA
		IO pulled up to VCC, TA = 25°C, 5 V	-	350		
$I_{CC-RESET}$	Supply current during Reset	RESET = 0 Reset pin held low. Hardware reset enabled. TA = 105°C, 3.3 V	-	0.8	1.5	mA

1. *The minimum supply voltage value of 2.7 V takes into account the ripple on the V_{CC} line of +/- 10%.*

   If there is no ripple on the V_{CC} line, the minimum supply voltage is 2.43 V and the maximum is limited by $V_{cc-HIPS}$ to 5.7 V .

7.2.2 Power-on and power-off sequences, and power supply glitch tolerance

The power-on sequence on STSAFE-A120 products must follow the requirements mentioned below:

- The $\overline{\text{RESET}}$ pin must not be tight to *high* prior to the V_{CC} power pin.
- The $\overline{\text{RESET}}$ pin must be tied to low prior to or simultaneously with the V_{CC} power pin.
- The voltage applied to the V_{CC} pin must be less than or equal to 0.3 V prior to starting a new power-on sequence.

For more information, refer to Figure 17. Power-on and reset sequence.

For security purposes, the STSAFE-A120 embeds detectors. When these are triggered, the STSAFE-A120 device enters the *reset* state until a power cycle or a reset event occurs.

It is recommended to use an application that is able to manage the $\overline{\text{RESET}}$ pin through a host GPIO to force reset upon alarm detection.

7.2.3 Reset pin (external reset)

The circuit is in reset state when the reset signal available on the $\overline{\text{RESET}}$ pin is at logical level '0'. If this signal is low for less than t_{WL}, it is not taken into account.

When the $\overline{\text{RESET}}$ pin is floating, an external reset is not available and the device remains in a reset state as the pin is connected to an internal weak pull-down.

When pin V_{CC} is tied high, if the $\overline{\text{RESET}}$ pin switches from high to low and then to high again, a warm reset occurs. For more information, refer to Figure 18. Warm reset sequence.

7.2.4 Power-on and reset sequence

Figure 17. Power-on and reset sequence

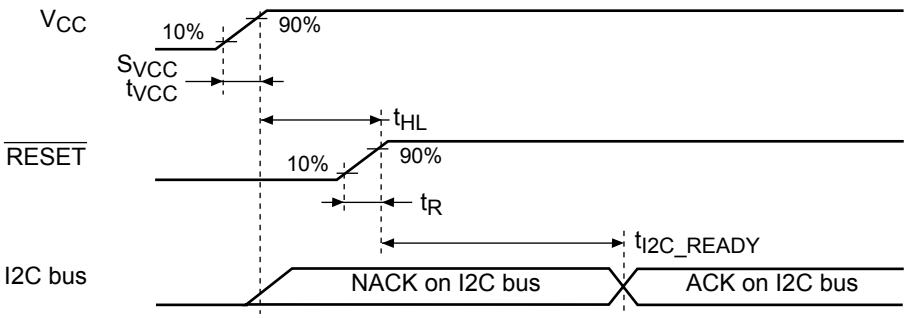


Figure 18. Warm reset sequence

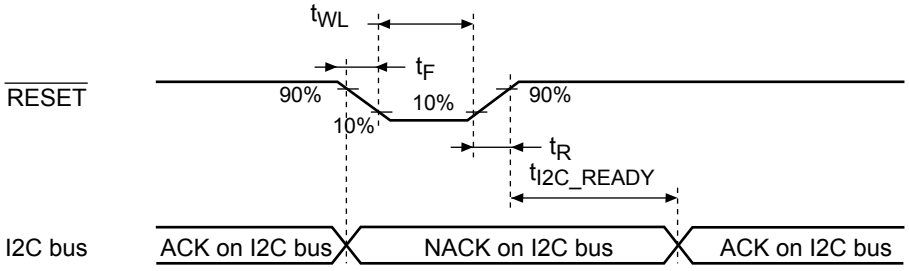


Table 4. Power-on and reset sequence timings

Name	Description	Conditions	Min	Typ	Max	Units
t_{HL}	Minimum time for reset active after power-up	-	0	-	-	µs
S_{VCC}	V_{CC} rising slope (from 10% to 90% of nominal value)	-	-	-	5	V/µs
t_{WL} Reset	Pulse width for reset(1)	-	5	-	-	µs
$t_{R/tF}$ Reset	Reset rise and fall time	$V_{CC} > V_{POR}$	-	-	1	µs
t_{I2C_READY}	Delay for STSAFE-A120 to accept I²C commands after a reset sequence.	-	-	4.5	20	ms

1. *Any low pulse (from 1 to 0, then 0 to 1) shorter than 5 µs is ignored.*

7.2.5 Power consumption optimization

When the STSAFE-A120 is not in use, it is possible to decrease its power consumption by removing the power supply properly.

This could be achieved by using a transistor to pilot the STSAFE-A120 power supply, or by using a GPIO able to provide an $I_{CC\text{-}PROC}$ current that respects the STSAFE-A120 powering conditions as illustrated in Section 6: Electrical integration.

Note: The $\overline{RESET}$ signal must remain low when power is removed.

7.3 DC characteristics

The following tables provide the detailed description of the DC operating conditions of STSAFE-A120 from 2.7 V to 5.5 V voltages.

Table 5. DC operating specifications and input parameters

Name	Description	Conditions	Min	Typ	Max	Units
V_{IH}	Input high voltage (CLK, RESET, I/O)	-	0.7 x Vcc	-	Vcc	V
V_{IL}	Input low voltage (CLK, RESET, I/O)	-	0	-	0.2 x Vcc	V
I_{IH}	Input high current, high impedance (SDA)	$0.7xV_{CC} < V_{IH} < V_{CC}$	-1	-	1	µA
	Input high current, high impedance (CLK)	$0.7xV_{CC} < V_{IH} < V_{CC}$	-100	-	200	nA
	Input high current, pull-down (RST)	$0.7xV_{CC} < V_{IH} < V_{CC}$	3	7	15	µA
I_{IL}	Input low current, high impedance (SDA)	$0V < V_{IL} < 0.2xV_{CC}$	-1	-	1	µA
	Input low current, high impedance (CLK)	$0V < V_{IL} < 0.2xV_{CC}$	-100	-	100	nA
	Input low current, pull-down (RST)	$0V < V_{IL} < 0.2xV_{CC}$	-1	1.5	10	µA
V_{OL}	Output low voltage (I/O)	IOL = 6 mA	-	-	460	mV
C_{IN1}	SCL input capacitance	V_{IN} = 0 to $V_{CC\ Max}$	-	-	30	pF
C_{IN2}	SDA input capacitance	V_{IN} = 0 to $V_{CC\ Max}$	-	-	30	pF

Note: $V_{CC\ MAX}$ is the maximum V_{CC} as defined in Table 3. Power supply specifications.

7.4 AC characteristics

Table 6. I²C operating conditions

Name	Description	Standard mode		Fast mode		Units
		Min.	Max.	Min.	Max.	
f_{SCL}	SCL frequency of subdevice: processor	-	100	-	400	kHz
$t_{HD;STA}$	Input low to clock low (start condition hold time)	4.0	-	0.6	-	µs
t_{LOW}	Low period of SCL clock	4.7	-	1.3	-	µs
t_{HIGH}	High period of SCL clock	4.0	-	0.6	-	µs
$t_{SU;STA}$	Clock high to input transition/setup time for a (repeated) start condition	4.7(1)	-	1.3(1)	-	µs
$t_{HD;DAT}$	Clock low to input transition	0(2)	-(3)	0(2)	(3)	µs
$t_{VD;DAT}$	Data valid time(4)	-	-	-	0.93(5)	µs
$t_{SU;DAT}$	Input transition to clock transition data setup time	250	-	100	-	ns
$t_{SU;STO}$	Clock high to input high (stop)	4.0	-	0.6	-	µs
t_R	Clock and data rise time on load capacitance of 30 pF	-	1000	20	300	ns
t_F	Clock and data fall time on load capacitance of 30 pF	-	300	20 x (VDD/5.5 V)	300	ns

1. *Repeated start not supported.*
2. *The device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the falling edge of SCL.*
3. *The maximum $t_{HD;DAT}$ could be 3.45 µs and 0.9 µs for standard mode and fast mode, but must be less than the maximum of $t_{VD;DAT}$ or $t_{VD;ACK}$ by a transition time. This maximum must only be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal. If the clock stretches the SCL signal, the data must be valid by the setup time before it releases the clock.*
4. *$t_{VD;DAT}$ = time for data signal from SCL LOW to SDA output (HIGH or LOW, depending on which one is worse).*
5. *The I²C specification value is 0.9 µs.*

Figure 19. AC clock and data timings

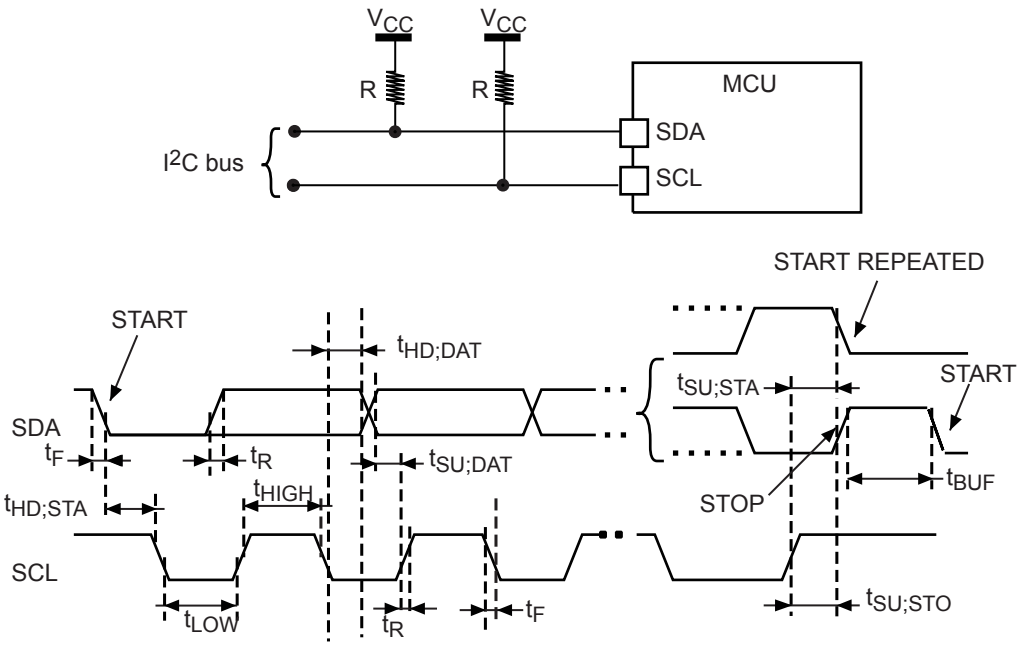


Table 7. AC measurement conditions

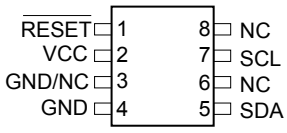
Description	Range	Units
Input rise and fall times	Max 10 ns	ns
Input pulse voltages	V_{IL} to V_{IH}	V
Input timing reference voltages	0.5 × V_{CC}	V
Output timing reference voltages	V_{OL} to V_{OH}	V

8 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK packages, depending on their level of environmental compliance. ECOPACK specifications, grade definitions and product status are available at: www.st.com. ECOPACK is an ST trademark.

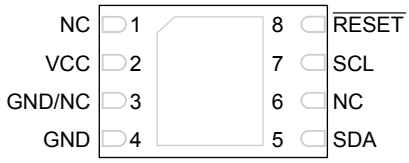
8.1 Pin descriptions

Figure 20. SO8N pinout - top view



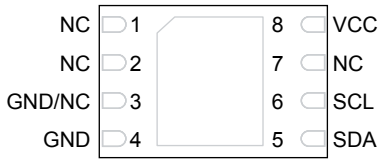
DT72963V2

Figure 21. UFDFPN8 pinout 1- top view



DT72964V2

Figure 22. UFDFPN8 pinout 2- top view



DT74422V1

Note: UPDFN8 pinout 2 does not have a reset pin. All references to the reset pin in this document do not apply to this pinout.

The table below provides the names and description of the four contacts on the STSAFE-A120 device. Details on each contact are provided later in this text.

Table 8. Signal descriptions

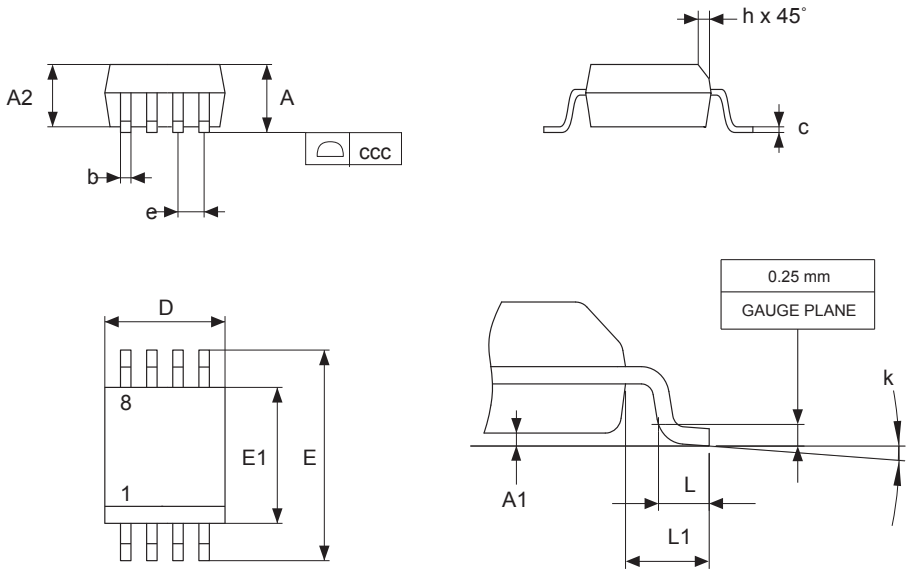
Signal	Name	Description
V_{CC}	Supply voltage	The supply voltage is supported for powering all internal STSAFE-A120 functions.
GND	Supply and signals ground	Ground reference pin for power and all I/O signals.
$\overline{\text{RESET}}$	Reset	This input signal is used to reset STSAFE-A120. The $\overline{\text{RESET}}$ pin is pull-down by default meaning that the device is reset if connected to ground or if the pin is floating. The device is active if the $\overline{\text{RESET}}$ pin is tied high.
SCL	Serial clock	This input signal is used to strobe all data in and out of STSAFE-A120. The I²C master drives the clock signal.
SDA	Serial data	This I/O signal is used to transfer data into and out of STSAFE-A120.

Signal	Name	Description
		The signal uses an open drain output configuration. An external pullup resistor is needed to “pull up” the output.
NC	-	Not connected internally.
GND/NC	-	Either connected to ground, or not connected at all.

8.2 SO8N package information

SO8N is an 8-lead, 4.9 × 6 mm, plastic small outline, 150 mils body width, package.

Figure 23. SO8N – Outline



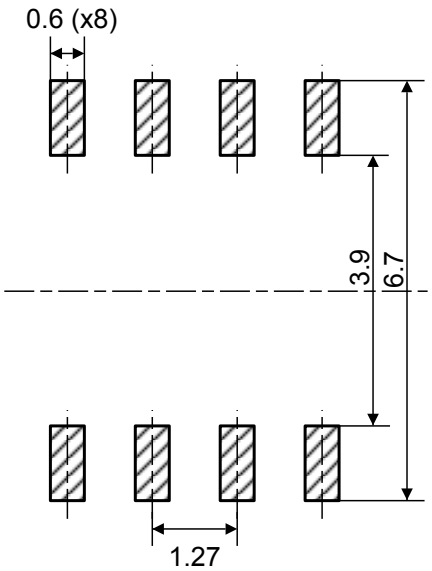
1. *Drawing is not to scale.*

Table 9. **SO8N – Mechanical data**

Symbol	millimeters			inches [1]		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	-	-	1.750	-	-	0.0689
A1	0.100	-	0.250	0.0039	-	0.0098
A2	1.250	-	-	0.0492	-	-
b	0.280	-	0.480	0.0110	-	0.0189
c	0.170	-	0.230	0.0067	-	0.0091
D	4.800	4.900	5.000	0.1890	0.1929	0.1969
E	5.800	6.000	6.200	0.2283	0.2362	0.2441
E1	3.800	3.900	4.000	0.1496	0.1535	0.1575
e	-	1.270	-	-	0.0500	-
h	0.250	-	0.500	0.0098	-	0.0197
k	0°	-	8°	0°	-	8°
L	0.400	-	1.270	0.0157	-	0.0500
L1	-	1.040	-	-	0.0409	-
ccc	-	-	0.100	-	-	0.0039

1. *Values in inches are converted from mm and rounded to four decimal digits.*

Figure 24. **SO8N - footprint example**

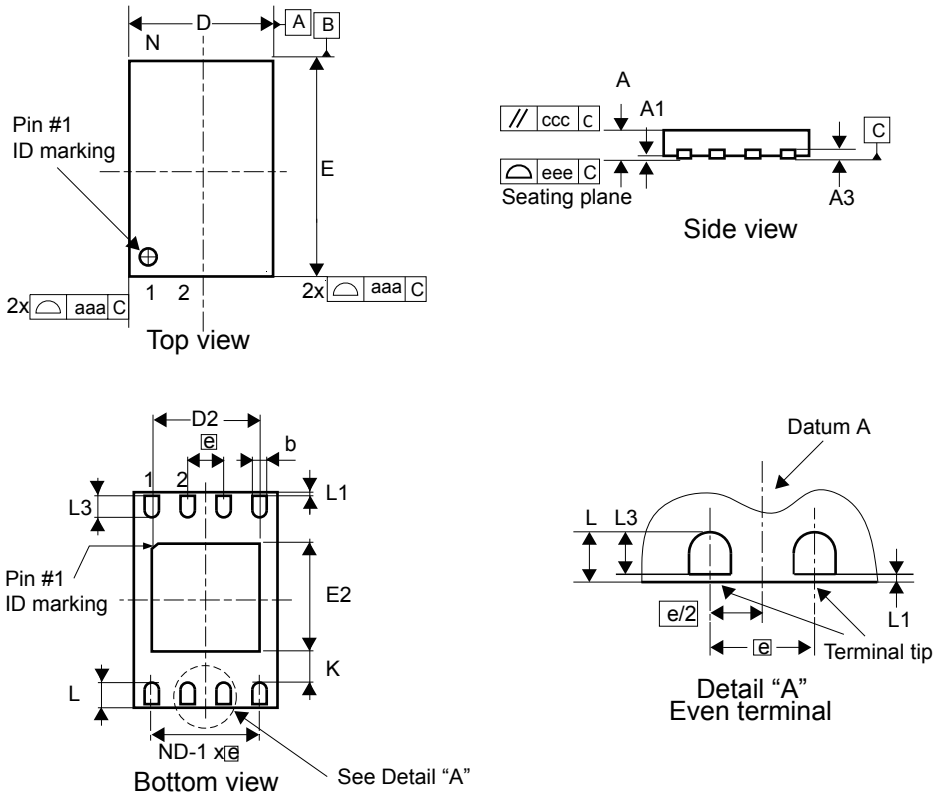


1. *Dimensions are expressed in millimeters.*

8.3 UFDFPN8 (DFN8) package information

UFDFPN8 is an 8-lead, 2 × 3 mm, 0.55 mm thickness ultra thin profile fine pitch dual flat package.

Figure 25. UFDFPN8 - Outline



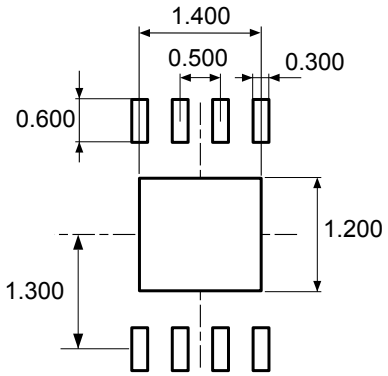
1. *Max. package warpage is 0.05 mm.*
2. *Exposed copper is not systematic and can appear partially or totally according to the cross section.*
3. *Drawing is not to scale.*
4. *The central pad (the area E2 by D2 in the above illustration) must be either connected to V_{SS} or left floating (not connected) in the end application.*

Table 10. UFDFPN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat mechanical data

Symbol	millimeters			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A	0.450	0.550	0.600	0.0177	0.0217	0.0236
A1	0.000	0.020	0.050	0.0000	0.0008	0.0020
b[2]	0.200	0.250	0.300	0.0079	0.0098	0.0118
D	1.900	2.000	2.100	0.0748	0.0787	0.0827
D2	1.200	-	1.600	0.0472	-	0.0630
E	2.900	3.000	3.100	0.1142	0.1181	0.1220
E2	1.200	-	1.600	0.0472	-	0.0630
e	-	0.500	-	-	0.0197	-
K	0.300	-	-	0.0118	-	-
L	0.300	-	0.500	0.0118	-	0.0197
L1	-	-	0.150	-	-	0.0059
L3	0.300	-	-	0.0118	-	-
aaa	-	-	0.150	-	-	0.0059
bbb	-	-	0.100	-	-	0.0039
ccc	-	-	0.100	-	-	0.0039
ddd	-	-	0.050	-	-	0.0020
eee[3]	-	-	0.080	-	-	0.0031

1. *Values in inches are converted from mm and rounded to 4 decimal digits.*
2. *Dimension b applies to plated terminal and is measured between 0.15 and 0.30 mm from the terminal tip.*
3. *Applied for exposed die paddle and terminals. Exclude embedding part of exposed die paddle from measuring.*

Figure 26. UFDFPN8 - 8-lead, 2 × 3 mm, 0.5 mm pitch ultra thin profile fine pitch dual flat footprint example



1. *Dimensions are expressed in millimeters.*

8.4 Tape and reel packagings

This section provides the tape and reel information for SO8N and DFN8 packagings.

Figure 27. S08N tape and reel

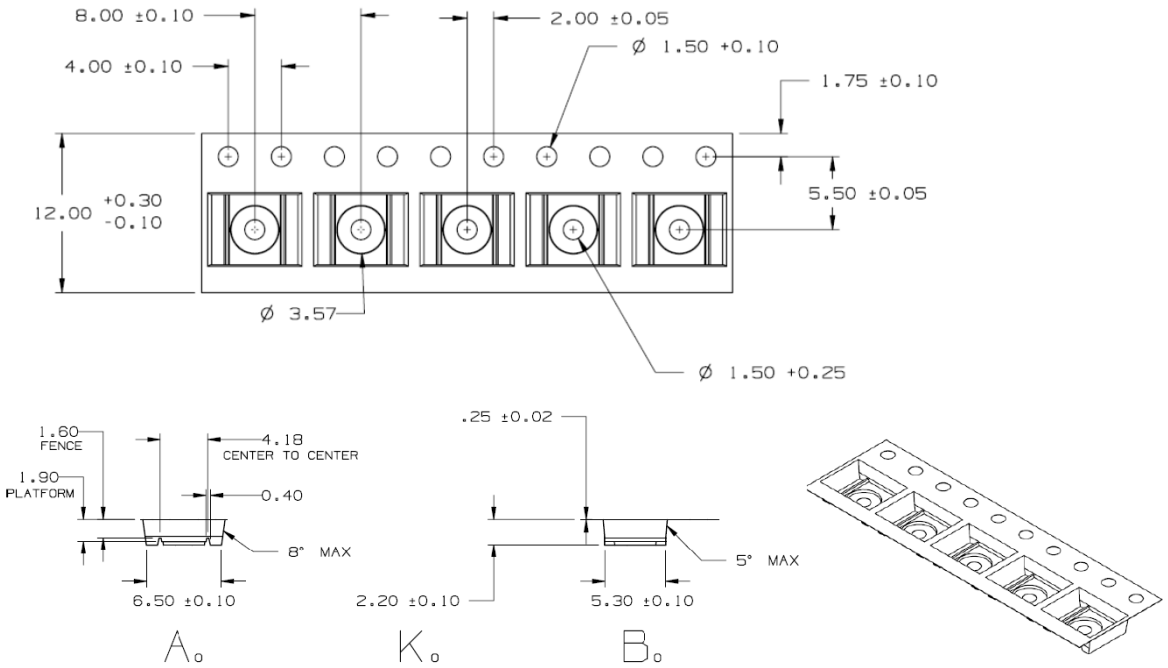
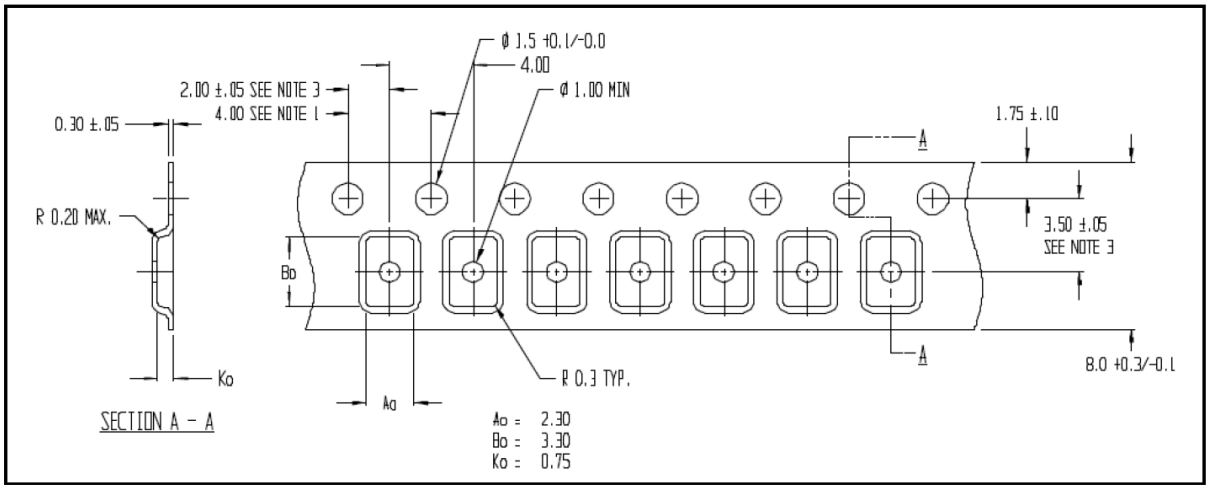


Figure 28. DFN8 tape and reel



9 Ordering information

Example: STSAFA120 S8 xxx yy

Product name

STSAFA120 = STSAFE-A120

Package codification

S8 = SO8N

DF = UFDFPN8 pinout 1

D2 = UFDFPN8 pinout 2

Customer personalization identification

xxx = personalization setup

Personalization revision

yy = personalization revision

Note: For a list of available options (speed, package, etc.) or for further information on any aspect of this device, contact your nearest STMicroelectronics sales office.

Appendix A Glossary

Table 11. **List of terms**

Term	Description
AES	Advanced encryption standard
CA	Certification authority
CC	Common Criteria
CCM	Counter with cipher block chaining message authentication code
CMAC	Cipher-based MAC
CRC	Cyclic redundancy check
EAL	Evaluation assurance level
ECB	Electronic codebook block
ECC	Elliptic curve cryptography
ECDH	Elliptic curve Diffie-Hellman (static key)
ECDHE	Elliptic curve Diffie-Hellman (ephemeral key)
ECDSA	Elliptic curve digital signature algorithm
EdDSA	Edwards-curve digital signature algorithm
GPIO	General purpose input/output
I²C	Inter-integrated circuit bus
IoT	Internet of things
KEK	Key encryption key
MAC	Message authentication code
MCU	Microcontroller unit
NIST	National institute of standards and technology
NVM	Nonvolatile memory
OTA	Over the air
RF	Radio frequency
R-MAC	Response MAC
SHA	Secure hash algorithm
ST	STMicroelectronics
TLS	Transport layer security
TRNG	True random number generator

Revision history

Table 12. **Document revision history**

Date	Revision	Changes
02-Apr-2024	1	Initial release.

Contents

List of tables

List of figures