ATAES132A

32K AES Serial EEPROM Specification

DATASHEET

CryptoAuthentication

Ensures Things and Code are Real, Untampered, and Confidential

Secure Download and Boot
Authentication and Protect Code In-transit

Ecosystem Control
Ensure Only OEM/Licensed Nodes and Accessories Work

Anti-cloning
Prevent Building with Identical BOM or Stolen Code

Message Security
Authentication, Message Integrity, and Confidentiality of Network Nodes (IoT)

Features

- Crypto Element Device with Secure Hardware-based Key Storage
- 32Kb Standard Serial EEPROM Memory
  - Compatible with the Atmel® AT24C32D and the Atmel AT25320B
  - 16 User Zones of 2Kb Each
- High-security Features
  - AES Algorithm with 128-bit Keys
  - AES-CCM for Authentication
  - Message Authentication Code (MAC) Capability
  - Guaranteed Unique Die Serial Number
  - Secure Storage for up to Sixteen 128-bit Keys
  - Encrypted User Memory Read and Write
  - Internal High-quality FIPS Random Number Generator (RNG)
  - 16 High-Endurance Monotonic EEPROM Counters
- Flexible User Configured Security
  - User Zone Access Rights Independently Configured
  - Authentication Prior to Zone Access
- Read/Write, Encrypted, or Read-only User Zone Options
- High-speed Serial Interface Options
  - 10MHz SPI (Mode 0 and 3)
  - 1MHz Standard I^2C Interface
- 2.5V to 5.5V Supply Voltage Range
- <250nA Sleep Current
- 8-pad UDFN and 8-lead SOIC Package Options
- Temperature Range: -40°C to +85°C

Benefits

- Easily Add Security by Replacing Existing Serial EEPROM
- Authenticate Consumables, Components, and Network Access
- Protect Sensitive Firmware
- Securely Store Sensitive Data and Enable Paid-for Features
- Prevent Contract Manufacturers from Overbuilding
- Manage Warranty Claims
- Securely Store Identity Data (i.e. Fingerprints and Pictures)

Description

The Atmel ATAES132A is a high-security, Serial Electrically-Erasable and Programmable Read-Only Memory (EEPROM) providing both authentication and confidential nonvolatile data storage capabilities. Access restrictions for the 16 user zones are independently configured, and any key can be used with any zone. In addition, keys can be used for standalone authentication. This flexibility permits the ATAES132A to be used in a wide range of applications.

The AES-128 cryptographic engine operates in AES-CCM mode to provide authentication, stored data encryption/decryption, and Message Authentication Codes. Data encryption/decryption can be performed for internally stored data or for small external data packets, depending upon the configuration. Data encrypted by one ATAES132A device can be decrypted by another, and vice versa.

The ATAES132A pinout is compatible with standard SPI and I^2C Serial EEPROMs to allow placement on existing PC boards. The SPI and I^2C instruction sets are identical to the Atmel Serial EEPROMs. The extended security functions are accessed by sending command packets to the ATAES132A using standard write instructions, and reading responses using standard read instructions. The ATAES132A secure Serial EEPROM architecture allows it to be inserted into existing applications.

The ATAES132A device incorporates multiple physical security mechanisms to prevent the release of the internally stored secrets. Secure personalization features are provided to facilitate third-party product manufacturing.

Table of Contents

1. Introduction

The ATAES132A is the first device in a family of high-security Serial EEPROMs using the Advanced Encryption Standard (AES) cryptographic algorithm. The ATAES132A provides 32Kb of EEPROM user data memory, sixteen 128-bit Key Registers, sixteen high-endurance monotonic EEPROM Counters, factory unique Die Identification Numbers, and a Configuration Memory. The Configuration Memory registers control access to the User Memory, as well as the restrictions on Key and Counter functionality.

The User Memory can be accessed directly with standard SPI or I^2C commands if a user zone is configured for open or read-only access. If the user zone security is activated, then the extended ATAES132A command set is used to access the contents of a user zone. The extended ATAES132A commands are executed by writing the command packet to the virtual memory using standard SPI or I^2C Write commands. The response packet is retrieved by reading it from the virtual memory using standard SPI or I^2C Read commands.

The ATAES132A packages are compatible with standard SPI and I^2C EEPROM footprints. This allows the ATAES132A to be inserted into many existing Serial EEPROM applications.

1.1 Scope

This *Specification* provides all specifications for configuration and operation of the ATAES132A.

1.2 Conventions

Table 1-1. Nomenclatures

Nomenclature	Definition	Notes
Host	The SPI or I^2C Master Device	The Host initiates all communications with slave devices on the serial interface bus.
Client	The ATAES132A Secure Serial EEPROM Defined by this Specification	Operates as a SPI or I^2C slave.
nnb	Binary Number	Denotes a binary number “nn” (most-significant bit on the left).
0xZZZZ	Hexadecimal Number	Denotes hex number ZZZZ (most-significant bit on the left).
$ZZZZ_h$	Hexadecimal Number	Denotes hex number ZZZZ (most-significant bit on the left).
RegName.FieldName	Field Name	Reference to bit field FieldName in register RegName.
RegArray[xx].FieldName	Field Name	Reference to bit field FieldName in register RegArray[xx], where xx is the array index.
UZ	User Zone	Reference to a User Zone number.
CntID	Counter ID	Reference to a Counter number.
KeyID	Key ID	Reference to a Key Register number.

1.2.1 Byte Order

The ATAES132A device uses a big-endian coding scheme and utilizes the same bit and byte orders as a standard Serial EEPROM. The byte order is identical to the NIST AES specifications (see Appendix A, Standards and Reference Documents):

- The most significant bit of each byte is transmitted first on the bus.
- The most significant byte of multi-byte integers is transmitted prior to the least significant byte. This applies to the CRC, address, and other 16-bit command parameters.
- All arrays are transmitted in index order, with byte index 0 first.
- Configuration fields that are more than eight bits appear on the bus during a Read or Write in the index order in which they appear in this specification. The top byte in the input parameters table is byte[0] and appears first on the bus. These fields are arrays of bytes, not multi-byte integers.

1.3 Abbreviations

Table 1-2. Abbreviations

Abbreviation	Phrase	Definition
AES	Advanced Encryption Standard	Block cipher algorithm standardized by NIST with 128-bit block size.
AES-CCM	AES Cipher Chaining Message	AES mode using the Counter with Cipher Block Chaining-Message Authentication Code Algorithm.
AES-ECB	AES Electronic Code Book	AES mode using the Electronic Code Book Algorithm.
Ciphertext		Data communicated after it has been encrypted.
Cleartext		Data communicated in a nonencrypted state.
MAC	Message Authentication Code	A 128-bit value used to validate the authenticity of ciphertext.
Nonce	Number Used Once	A value used in cryptographic operations.
Plaintext		Data which is either the input to an encryption operation or the output of a decryption operation.
RFU	Reserved For Future Use	Any feature, memory location, or bit that is held as reserved for future use by Atmel.
RNG	Random Number Generator	Produces high-quality pseudo-random numbers.

1.4 Communication

The ATAES132A is designed to interface directly with SPI and I^2C microcontrollers. The read and write commands are similar to the standard Atmel Serial EEPROM commands for ease of use. Since the ATAES132A pinout is also similar to standard Serial EEPROMs, it is possible to use the ATAES132A on existing PC boards in some cases.

When Read and/or Write access to a user zone is unrestricted, the memory is accessed using the standard I^2C or SPI Read and Write commands. Similarly, if *Authentication Only* is required and the authentication requirement has been satisfied, then the memory is accessed directly by the Host using standard I^2C or SPI Read and Write commands.

If the Host begins a Read operation in an open user zone but continues reading until a prohibited section of memory is reached, the ATAES132A will continue to increment the address and will return `0xFF` for each byte in the restricted user zone. If the Host begins a Read operation in an open user zone but continues reading beyond the end of the User Memory, the ATAES132A will return `0xFF` for each byte requested, but will stop incrementing the address.

All other operations, including the execution of the extended commands, are performed by using the standard I^2C or SPI Read and Write commands to exchange data packets via the command and response memory buffers. The Device Status Register reports the state of the device and is used for handshaking between the Host and the ATAES132A.

1.4.1 Sending ATAES132A Commands

The ATAES132A commands described in Section 7, Command Definitions, are executed by writing the command block to virtual memory (Appendix D, Command Memory Map) using standard SPI or I^2C Write commands. The response block is retrieved by reading it from virtual memory using standard SPI or I^2C Read commands.

1.4.1.1 Command Memory Buffer

The Command Memory Buffer is a write-only memory buffer that is used by writing a command block to the buffer at the base address of `0xFE00`. After the Host completes its Write operation to the buffer, the ATAES132A verifies the integrity of the block by checking the 16-bit checksum, and then executes the requested operation. See Section 6.1, Command Block and Packet for a description of the command packet. See Appendix D for additional Command Memory Buffer information.

Table 1-3. Command Memory Buffer Map

Base Address	Base + 1	Base + 2	Base + 3	.....	.....	.....	.....	Base + N-2	Base + N-1
Count	Opcode	Mode	Param1	Param1	Param2	.....	DataX	CRC1	CRC2

1.4.1.2 Response Memory Buffer

The Response Memory Buffer is a read-only memory buffer that is used by reading a response from the buffer at the base address of `0xFE00`. The base address of the Response Memory Buffer contains the first byte of the response packet after an ATAES132A command is processed. See Section 6.1 for a description of the response packet. See Appendix D for additional Response Memory Buffer information.

Table 1-4. Response Memory Buffer Map Following a Crypto Command

Base Address	Base + 1	Base + 2	Base + 3	.....	.....	.....	.....	Base + N-2	Base + N-1
Count	ReturnCode	Data1	Data2	Data3	.....	.....	DataX	CRC1	CRC2

The Response Memory Buffer is also used to report errors which occur during execution of standard I^2C or SPI Write commands. When the I^2C or SPI command execution is complete (as indicated by the STATUS Register), the Response Memory Buffer contains a block containing an error code (ReturnCode) if an error occurred, otherwise it contains a block with ReturnCode = `0x00`. See Section 6.3, ReturnCode, for the error descriptions.

1.4.2 Device Status Register (STATUS)

The Device Status Register is used for handshaking between the Host microcontroller and the ATAES132A. The Host microcontroller is expected to read the STATUS Register before sending a command or reading a response.

The read-only Device Status Register at address `0xFFF0` reports the current status of the ATAES132A device. This register can be read with the standard I^2C or SPI Read Memory commands. The SPI Read Status Register command can also be used to read the STATUS Register, as described in Appendix K.3.5, Read Status Register Command (RDSR).

Reading the STATUS Register does not increment the Memory Read Address, and so a Host microcontroller can easily monitor the ATAES132A device status by repeatedly reading the STATUS Register. See Appendix G, Understanding the STATUS Register for a detailed description of the STATUS Register bits and Status Bit behavior.

Table 1-5. Device Status Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EERR	RRDY	Reserved	CRCE	Reserved	WAKEb	WEN	WIP

The Device Status Register can always be read when the ATAES132A is configured for SPI interface mode, even if the ATAES132A is processing a command or writing the EEPROM. When the ATAES132A is configured for I^2C interface mode, the Host can read the STATUS Register only when the I^2C Device Address is ACKed.

If the ATAES132A is in the Sleep or Standby power state, reading the STATUS Register forces the ATAES132A to wake-up; the STATUS Register is `0xFF` until the wake-up process is complete.

2. Memory

The ATAES132A EEPROM is a nonvolatile memory which is divided into several sections with each section having a different function. The User Memory section contains 32Kb for data storage. The Configuration Memory section contains the configuration information, security control registers, and counters. The Key Memory stores the 16 secret keys used to perform cryptographic functions. The EEPROM page length is 32 bytes. The ATAES132A SRAM buffers and registers are located near the top of the memory address space and are accessed using standard EEPROM Read/Write commands.

The complete memory map is shown in Appendix B, Memory Map. Each portion of the memory is described briefly in the following sections.

2.1 User Memory

The 32Kb User Memory is organized as 16 user zones of 2Kb each. Each user zone has an associated user zone configuration register in the Configuration Memory. A user zone can only be accessed when the security requirements specified in the associated user zone configuration register have been satisfied. All bytes within a user zone have the same access restrictions. Since the user zone access restrictions are independently configured, the security requirements for each user zone can be unique. Any key can be used with any user zone.

Each user zone can be configured to require authentication, Read Encryption, Write Encryption, a combination of these, or no security. The User Memory can be accessed directly with standard SPI or I^2C commands if a user zone is configured for open or read-only access. If the user zone security is activated, then the extended ATAES132A command set is used to access the contents of a user zone.

2.1.1 Automatic Post Write Data Verification

The `Write` and `EncWrite` commands include an automatic data verification function. After the EEPROM `Write` is complete, the Data Verification Logic reads the new EEPROM contents and compares it to the data received from the Host. If the data does not match, the ATAES132A sets the EERR bit in the STATUS Register and returns a DataMatch error code. If the data is correct, then the ReturnCode indicates success.

2.2 Key Memory

The Key Memory securely stores 16 keys which are each 128 bits long. Each key has an associated Key Configuration Register in the Configuration Memory. Keys can only be used for the cryptographic functions enabled in the Key Configuration Register. Individual keys can be configured to require a successful authentication prior to use. Key values can never be read from the ATAES132A under any circumstances. See Appendix F, Key Memory Map.

The Key Memory can be written prior to locking with either encrypted or cleartext data. Encrypted writes are performed using the `EncWrite` command (see Section 7.11, `EncWrite` Command). Cleartext writes are performed using the standard SPI or I^2C Write commands (see Section 5.2, Write). After locking, the Key Registers are managed with the `KeyCreate`, `KeyImport`, `KeyLoad`, and `KeyTransfer` commands. The `KeyTransfer` command allows the User Memory to be used as the Extended Key Memory; eight keys can be stored in each user zone (see Section 7.16, `KeyTransfer` Command).

2.3 Configuration Memory

The Configuration Memory contains all of the registers which control user zone access requirements, the Key usage restrictions, and the Counter usage restrictions. Device-level Configuration Option Registers are also located in Configuration Memory.

The ATAES132A Configuration Memory includes a register programmed with unique, read-only die identification data at the factory. The Configuration Memory also contains several registers for customer information. The Configuration Memory registers can always be read using the `BlockRead` command (see Section 7.4, `BlockRead` Command). The `Lock` command is used to permanently lock the contents of the Configuration Memory after personalization (see Section 7.18, `Lock` Command).

See Table 2-1 for a summary of the Configuration Memory registers sorted by register name. See Appendix E for the Configuration Memory Map.

2.3.1 Counters

The ATAES132A includes 16 high-endurance EEPROM Counters. Each Counter has or can:

- An associated Counter Configuration Register in the Configuration Memory,
- Only be incremented,
- Never be decremented or reset,
- Be used to track system usage or to store small values.

A key can be configured to prevent exhaustive attacks by limiting key usage with a Counter.

Each counter can increment up to a value of 2,097,134 using the `Count` Command, after which they can be no longer changed. Counters attached to keys are incremented each time the key is used; when the Usage Counter reaches its limit, the key is disabled. The Counters include a power interruption protection feature to prevent corruption of the Count value if power is removed during the increment operation.

On shipment from Atmel, the EEPROM locations are initialized to their lowest value. The initial value of each Counter may be written to a different value prior to personalization and prior to locking the configuration. See Appendix H, Understanding Counters.

2.4 SRAM Memory

The ATAES132A SRAM is used to store volatile data and status information. The ATAES132A SRAM buffers and registers are mapped into the top of the memory address space and are accessed using the Standard EEPROM Read/Write commands.

- The Command Memory Buffer is used to send extended commands to the device.
- The Response Memory Buffer is used to read responses to the extended commands from the device.
- An IO Address Reset Register is used to reset the buffer address pointers.
- The STATUS Register reports the state of the device.
- The VolatileKey register and the Authentication Status Register are stored in the SRAM and are managed by the internal logic. These registers cannot be directly written or read by the user.

2.4.1 Nonce

The SRAM is used to store the Nonce and Random Number Generator (RNG) Seed. The RNG Seed is generated automatically by ATAES132A, as described in Section 3.6, Random Numbers. The Nonce is generated using the `Nonce` command or the `NonceCompute` command. The Nonce and RNG Seed Register are erased when the device loses power, enters the Sleep state, or is reset.

2.4.2 VolatileKey

The SRAM contains a session key register named VolatileKey. This key location can be written with the `KeyCreate`, `KeyImport`, `KeyLoad`, or `KeyTransfer` commands. The VolatileKey register is erased when the device loses power, enters the Sleep state, or is reset. Restrictions on VolatileKey are established when the register is created or loaded and persist until the power is lost or the key is reloaded.

VolatileKey can never be used to read or write the User Memory or to increment the Counters. VolatileKey can only be used to perform authentication operations and to encrypt or decrypt external data. See Section 4.3, VolatileKey Configuration for the VolatileKey usage restrictions.

2.4.3 Command Memory Buffer

The Host executes extended ATAES132A commands by writing a command block to the Command Memory Buffer using a standard SPI or I^2C Write command. After the Host completes its write operation to the SRAM buffer, ATAES132A verifies the integrity of the block by checking the 16-bit Checksum and then executes the requested operation.

2.4.4 Response Memory Buffer

The Host receives responses to the extended ATAES132A commands by reading a response block from the Response Memory Buffer using a standard SPI or I^2C Read command. The base address of the Response Memory Buffer contains the first byte of the response packet after an ATAES132A command is processed.

2.4.5 IO Address Reset Register

Writing the IO Address Reset Register causes the address pointers in the Command Memory Buffer and the Response Memory Buffer to be reset to the base address of the buffers. Writing the IO Address Reset Register does *not* alter the contents of the Response Memory Buffer or the value of the STATUS Register.

2.4.6 Device Status Register (STATUS)

The Device Status Register is used for handshaking between the Host microcontroller and ATAES132A. The Host is expected to read the STATUS Register before sending a command or reading a response. Reading the STATUS Register does *not* alter the contents of the Command Memory Buffer, the Response Memory Buffer, or the value of the STATUS Register. See Appendix G, Understanding the STATUS Register for the definition and behavior of the STATUS Register.

2.4.7 Authentication Status Register

The ATAES132A Authentication Status Register stores the result of most recent authentication attempt. The Authentication Status Register contains the Authentication KeyID, the AuthComplete status flag, and the authentication usage restriction bits. Prior to executing the `Auth` command, the AuthComplete status flag is set to NoAuth. After successful Inbound Only or Mutual Authentication, the AuthComplete status flag is set to YesAuth.

Table 2-1. Summary of the Configuration Memory Registers Sorted by Register Name[1]

Name	Description	Write	Read	Bytes
Algorithm	Algorithm ID code (0x0000).	Never	Always	2
ChipConfig	Device-level cryptographic and power-up configuration options.	If LockConfig = Unlocked	Always	1
Counters	16 high-endurance counters, each capable of counting to 2M. See Appendix H, Understanding Counters.	If LockConfig = Unlocked	Always	128
CounterConfig	Configuration information for each counter. See Section 4.4, Counter Configuration.	If LockConfig = Unlocked	Always	32
DeviceNum	Atmel device number code.	Never	Always	1
EEPageSize	Length in bytes of physical EEPROM page (32, 0x20).	Never	Always	1
EncReadSize	Maximum data length in bytes for EncRead (32, 0x20).	Never	Always	1
EncWriteSize	Maximum data length in bytes for EncWrite (32, 0x20).	Never	Always	1
FreeSpace	Free memory for customer data storage.	If LockConfig = Unlocked	Always	96
JEDEC	Atmel JEDEC manufacturer code (0x001F).	Never	Always	2
KeyConfig	Configuration information for each key. See Section 4.2, Key Configuration.	If LockConfig = Unlocked	Always	64
LockConfig	Controls Configuration Memory Write access, except SmallZone. Default is the Unlocked state. (2)	Via Lock Command Only	Always	1
LockKeys	Controls Key Memory Write access. Default is the Unlocked state. (2)	Via Lock Command Only	Always	1
LockSmall	Controls SmallZone Register Write access. Default is the Unlocked state. (2)	Via Lock Command Only	Always	1
LotHistory	Atmel proprietary manufacturing information.	Never	Always	8
ManufacturingID	Two byte manufacturing ID code.	Never	Always	2
PermConfig	Atmel factory device configuration options.	Never	Always	1
SerialNum	Guaranteed unique die serial number. SerialNum is optionally included in cryptographic calculations. See Appendix E.2.1, SerialNum Register.	Never	Always	8
SmallZone	32 byte value. The first four bytes are optionally included in cryptographic calculations. See Appendix E.2.23, SmallZone Register.	If LockSmall = Unlocked	Always	32
I2C ADDR	Selects the serial interface mode and stores the I^2C Device Address.	If LockConfig = Unlocked	Always	1
ZoneConfig	Access and usage permissions for each user zone. See Section 4.1, User Zone Configuration.	If LockConfig = Unlocked	Always	64

gisters take effect immediately which allows the functionality to be tested during the personalization process. Changes to the I2C ADDR register take effect at the next Reset, Power Up, or Wake-Up from the Sleep state.

2. The LockConfig, LockKeys, and LockSmall bytes can only be changed with the Lock command (see Section 7.18.1, User Zone ReadOnly Activation). **Warning:** ATAES132A must always be locked by the customer prior to shipment to the end user to protect the customer secrets.

3. Security Features

All ATAES132A security features are optional. Each feature is enabled or disabled by programming configuration bits in the EEPROM Configuration Memory. Each user zone, Key, and Counter is separately and independently configured.

This section describes the ATAES132A security features and cryptographic capabilities. The functionality associated with each portion of the memory is described in Section 2, Memory.

3.1 Architecture

ATAES132A contains all circuitry for performing authentication, encryption, and decryption using keys stored securely in the internal EEPROM. Since the secrets are stored securely in the ATAES132A, they do not have to be exchanged prior to executing cryptographic operations.

ATAES132A has fixed cryptographic functionality; it is not a microcontroller and cannot accept customer firmware. ATAES132A contains a hardware AES cryptographic engine and has a fixed command set. Although the functionality is fixed, it is also flexible because each feature is enabled or disabled by the customer by programming registers in the EEPROM Configuration Memory. After personalization is complete, fuses lock the configuration so it cannot be changed.

3.1.1 AES

The ATAES132A cryptographic functions are implemented with a hardware cryptographic engine using AES in CCM mode with a 128-bit key. AES-CCM mode provides both confidentiality and integrity checking with a single key. The integrity MAC includes both the encrypted data and additional authenticate-only data bytes, as described in each command definition. Each MAC is unique due to inclusion of a Nonce and an incrementing MacCount Register in the MAC calculation.

See Appendix I, Cryptographic Computations for information about how the AES computations are performed. Hyperlinks to the AES standard are provided in Appendix A, Standards and Reference Documents.

3.1.2 Hardware Security Features

The ATAES132A device contains physical security features to prevent an attacker from determining the internal secrets. ATAES132A includes tamper detectors for voltage, temperature, frequency, and light, as well as an active metal shield over the circuitry, internal memory encryption, and other various features. The ATAES132A physical design and cryptographic protocol are designed to prevent or significantly complicate most algorithmic, timing, and side-channel attacks.

3.2 Authentication

The authentication commands utilize AES-CCM to generate or validate a MAC value computed using an internally stored key. The command set supports both one-way and mutual authentication. One ATAES132A device can generate packets for authentication of a second ATAES132A device containing the same key. The internal authentication status register remembers only the most recent authentication attempt. A user zone can be configured to require prior authentication of a designated key before access to the user zone is permitted.

3.2.1 Key Authentication

Individual keys can be configured to require a successful authentication prior to use. This requirement can be used to prevent some kinds of exhaustive attacks on the keys. The authentication requirement can be chained to require authentication of several keys prior to allowing a particular operation. The internal Authentication Status Registers remember only the most recent authentication attempt.

3.3 Encrypted Memory Read/Write

A user zone can be configured to require AES-CCM encryption for EEPROM Read or Write operations. If encryption is required for Write access, then the MAC is validated before the received (encrypted) data are written to the EEPROM. If encryption is required for read access, then ATAES132A encrypts data when they are read from the internal EEPROM, and generates an associated integrity MAC.

3.4 Data Encryption/Decryption

A key can be configured to allow encryption or decryption of small packets of data using AES-CCM with an internally stored key. The `Encrypt` command encrypts 16 or 32 bytes of plaintext data provided by the Host; the encrypted data, and MAC are returned to the Host. The `Decrypt` command decrypts 16 or 32 bytes of encrypted data after verifying the MAC; the data is returned to the Host only if the MAC is valid. When these commands are used, none of the data is stored in the internal EEPROM.

3.4.1 AES-ECB Encryption/Decryption

A key can be configured to allow AES-ECB mode operations using the `Legacy` command. A single AES-ECB operation is performed using an internally stored key and the 16-byte input packet received with the AES-ECB command. The 16-byte result is returned to the Host. No input or output formatting is performed by this command, and no data is stored in the internal EEPROM.

3.5 Keys

The ATAES132A securely stores sixteen 128-bit keys in the EEPROM. Keys can only be used for the cryptographic functions enabled in the ZoneConfig, CounterConfig, or KeyConfig Register bits in the Configuration Memory. Key values can never be read from ATAES132A under any circumstances. Any key can be used with any user zone.

A seventeenth key register in the internal SRAM can be used for session keys.

See Section 7.11, Encrypted Key Writes, for the `EncWrite` command. See Section 7.18, User Zone ReadOnly Activation, for the `Lock` command.

3.5.1 Key Management

The Key registers can be written with plaintext data or with encrypted data before the Key Memory is locked. After the Key Memory is locked, a key register can only be updated only if the corresponding KeyConfig Register allows updates.

Several key management commands are available for updating or generating the keys:

1. An encrypted key provided by the Host can be written to an internal key register after validating the MAC. The `KeyImport` command and `KeyLoad` command performs this function.
2. The internal RNG can be used to create a key for use as a session key or for storage in an internal Key Register. The new key can be encrypted and returned to the Host for use as the encrypted key input to another ATAES132A device. The `KeyCreate` command performs this function.
3. Keys stored in User Memory can be transferred to an internal key register or used as a session key. A user zone configured as extended Key Memory can be used to store eight keys. The `KeyTransfer` command performs this function.

3.5.2 Limited Use Keys

To prevent exhaustive attacks on the keys, the ATAES132A can be configured to limit key usage with a Counter. If a key is configured with a Usage Counter, then the following steps are performed for any command using that key:

1. Read the Counter from memory to check if the count has reached the maximum count value.
2. If the maximum count has been reached, then the command is not executed and an error code is returned.
3. If the maximum count has not been reached, then the Counter is incremented and the command is executed.

By default, the Counters are configured to allow 2,000,000 counts, allowing 2,000,000 operations using a key with the usage limits enabled. Atmel recommends the customer configure the Key Usage Counters to a smaller number at personalization; the appropriate key usage limit is dependent on the application. See Appendix H, Understanding Counters for additional information.

3.5.3 Secure Personalization

The ATAES132A is designed to allow personalization of keys using encryption so that the secret key values cannot be determined by a third party. AES encryption of the keys prevents them from being determined by observation of data communicated to or from ATAES132A.

A Transport Key is programmed into the KeyID 00 Register by Atmel during the device manufacturing process. This Transport Key is securely exchanged between the customer and Atmel. During personalization, the secret keys are encrypted using the Transport Key before being written to ATAES132A.

Atmel also offers a secure personalization service at an additional cost which uses a Hardware Security Module (HSM) to store the customer secrets.

3.5.3.1 Key Diversification

Atmel recommends that each unit should contain one or more unique keys to minimize the potential impact of cloning. The keys stored in the ATAES132A should be a cryptographic combination of a root secret not stored in the device along with the unique ATAES132A SerialNum Register value. The Host must have a secure place to store the root secret to protect the integrity of the diversified keys.

It may also be beneficial for the ATAES132A devices to contain secrets for validating the authenticity of the Host. These secrets may need to be the same on all ATAES132A devices for a particular application to permit any Client to validate any Host. See Section 7.13, `KeyCreate` Command, Mode bit 2.

3.6 Random Numbers

The ATAES132A includes a high-quality RNG for Nonce generation, child key creation, and general random number generation. The ATAES132A commands can generate random numbers for internal or external use. Sixteen byte random numbers for external use are generated using the internal RNG and the AES engine, as described in NIST SP800-90.

The RNG can be used to generate the Nonce for cryptographic operations. A mechanism is also provided to synchronize the Nonces in two ATAES132A devices using random numbers generated by both devices. A key can be configured to require that the cryptographic operations using the key use a Nonce generated with the internal RNG.

3.6.1 Random Number Generation

The RNG architecture includes both a hardware RNG and a stored random seed. On power-up, the stored seed is read from the EEPROM, cryptographically combined with the hardware RNG output, and then stored in SRAM. Whenever a random number is required, this SRAM Seed is cryptographically combined with the hardware RNG output and the optional input seed to create both a new SRAM Seed and the random number.

For the highest security, the EEPROM Seed should be updated at every power cycle in which the RNG is used. However, the EEPROM Seed Register has a maximum life expectancy of 100,000 writes per unit. The Host system is expected to manage the EEPROM Seed by using the command mode option to suppress automatic EEPROM Seed updates.

4. Security Configuration Registers

4.1 User Zone Configuration

Access permissions to each user zone are controlled by the ZoneConfig Registers in the Configuration Memory. There is one ZoneConfig Register for each User Memory zone.

Table 4-1. Definition of the ZoneConfig Register Bits[1]

ZoneConfig Field	Byte	Bit	Description
AuthRead	0	0	1b = Authentication is required to read data. 0b = Authentication is not required to read data.
AuthWrite	0	1	1b = Authentication is required to write data. 0b = Authentication is not required to write data.
EncRead	0	2	1b = Encryption is required to read data. 0b = Encryption is not required to read data.
EncWrite	0	3	1b = Encryption is required to write data. 0b = Encryption is not required to write data.
WriteMode	0	4 to 5	00b = Zone is permanently read/write. 01b = Zone is permanently read-only. 10b = The ReadOnly byte determines if writes are permitted. 11b = The ReadOnly byte determines if writes are permitted, and the Lock command must include an authenticating MAC calculated using the KeyID stored in ZoneConfig[UZ].WriteID .
UseSerial	0	6	UseSerial = 1b and EncWrite = 1b, then SerialNum must be included in EncWrite operations. EncWrite = 0b, then this bit is ignored.
UseSmall	0	7	UseSmall = 1b and EncWrite = 1b, the first four bytes of SmallZone must be included in EncWrite operations. EncWrite = 0b, this bit is ignored.
ReadID	1	0 to 3	KeyID which is used to encrypt data read from this zone. The same key is used to generate the MAC.
AuthID	1	4 to 7	KeyID which is used for inbound authentication before access is permitted.
VolatileTransferOK	2	0	1b = Key transfer from this User Zone to VolatileKey is permitted. 0b = Key transfer from this User Zone to VolatileKey is prohibited.
Reserved	2	1 to 3	Reserved for future use.
WriteID	2	4 to7	KeyID that is used to decrypt data written to this zone. The same key is used to verify the MAC.
ReadOnly	3	0 to 7	The contents of this byte are ignored unless WriteMode contains 10b or 11b. If 0x55, then the user zone is Read/Write. If any other value, then the user zone is read-only. This byte can be updated after the Configuration Memory is locked using the Lock command (See Section 7.18, Lock Command).

Note: 1. Most changes to the ZoneConfig Registers take effect immediately. Changes to the AuthRead and EncRead bits do not affect the SPI or I^2C Read command until the next reset or power-up.
Warning: The ATAES132A must always be locked by the customer prior to shipment to the end user to protect the customer secrets. See Section 7.18.

4.2 Key Configuration

Restrictions on key usage are controlled by the KeyConfig Registers in the Configuration Memory. There is one KeyConfig Register for each key.

Table 4-2. Definition of the KeyConfig Register Bits[1][2][4]

KeyConfig Field	Byte	Bit	Description
ExternalCrypto	0	0	1b = The key can be used with the Encrypt and Decrypt commands.[3] 0b = The Encrypt and Decrypt commands are prohibited.
InboundAuth	0	1	1b = The key can only be used by the Auth command for Inbound Only or Mutual Authentication. The key cannot be used by any other command, but KeyID can be the target of a key management command. 0b = The key can be used for any purpose not prohibited by another KeyConfig bit, including Outbound Only authentication.
RandomNonce	0	2	1b = Operations using this key requires a random Nonce (see Section 7.19). 0b = The Nonce is not required to be random.
LegacyOK	0	3	1b = The key can be used with the Legacy command. 0b = The key cannot be used with the Legacy command.
AuthKey	0	4	1b = The key requires prior authentication using the KeyID stored in LinkPointer. 0b = Prior authentication is not required.
Child	0	5	1b = The key is permitted to be the target of a KeyCreate or KeyLoad command. 0b = This use is prohibited.
Parent	0	6	1b = This key can be used as the parent when writing VolatileKey via KeyCreate, KeyImport, or KeyLoad (see Section 4.3). 0b = This use is prohibited.
ChangeKeys	0	7	1b = Key updates are permitted after locking. The new key is written using the EncWrite command with a MAC generated with the current value of key. (see Section 7.11). 0b = Key updates with EncWrite command are prohibited.
CounterLimit	1	0	1b = Usage count limits are enabled for this key (see CounterNum). 0b = No usage limits.
ChildMac	1	1	1b = An input MAC is required to modify this key using the KeyCreate command. 0b = The KeyCreate command does not require an input MAC (it will be ignored, if provided).
AuthOut	1	2	1b = I2C Auth signaling is enabled for this key (see Appendix J.5). 0b = I2C Auth signaling is disabled for this key.
AuthOutHold	1	3	1b = The I2C AuthO output state is unchanged when an authentication reset is executed using this key. 0b = Then the I2C AuthO output is reset when an authentication reset is executed using this key (see Appendix J.5).
ImportOK	1	4	1b = The key is permitted to be the target of a KeyImport command. 0b = KeyImport command is prohibited.
ChildAuth	1	5	1b = The KeyCreate command requires prior authentication using the KeyID stored in LinkPointer. 0b = Prior authentication is not required.

KeyConfig Field	Byte	Bit	Description
TransferOK	1	6	1b = The key is permitted to be the target of a KeyTransfer command (see Section 7.16). 0b = KeyTransfer command is prohibited.
AuthCompute	1	7	1b = The key can be used with the AuthCompute command. 0b = The key cannot be used with the AuthCompute command.
LinkPointer	2	0 to 3	For child keys; stores the ParentKeyID. For all other keys; the KeyID of the authorizing key (see AuthKey).
CounterNum	2	4 to 7	Stores the CntID of the Monotonic Counter attached to this key for usage limits or for MAC calculation. MAC calculations will include the Counter if Command Mode bit 5 is 1b even if key usage limits are disabled.
DecRead	3	0	1b = The DecRead and WriteCompute commands can be run using this key. 0b = The DecRead and WriteCompute are prohibited.
Reserved	3	1 to 7	Reserved for future use.

Notes:
1. Changes to the KeyConfig Registers take effect immediately which allows the functionality to be verified during the personalization process.
2. **Warning:** The ATAES132A must always be locked by the customer prior to shipment to the end user to protect the customer secrets. See Section 7.18, Lock Command.
3. **Warning:** Since the Encrypt command does not include an input MAC, the Encrypt command can exhaustively be run with selected input data to attack the key. Requiring authentication prior to allowing encryption makes these attacks more difficult. To require prior authentication, the AuthKey and RandomNonce bits must be set to 1b.
4. A key can be disabled by setting KeyConfig[KeyN].AuthKey to 1b, and KeyConfig[KeyN].LinkPointer to contain “KeyN”, where KeyN = KeyID of the key being configured.

4.3 VolatileKey Configuration

There is a seventeenth key register, named VolatileKey, which has a KeyID of 0xFF and is stored in the internal SRAM. This key location can be written with the KeyCreate (see Section 7.13, KeyCreate Command), KeyImport (see Section 7.14, KeyImport Command), KeyLoad (see Section 7.15, KeyLoad Command), or KeyTransfer (see Section 7.16, KeyTransfer Command) commands. The contents of the VolatileKey Register are erased when the device is powered down, enters the Sleep state, or is reset.

When the VolatileKey Register is loaded, restrictions are placed on its usage which persists until the power is lost or the key is reloaded. The definition of the VolUsage field is shown in Table 4-3.

Table 4-3. VolUsage Field Bit Definitions in the KeyCreate or KeyLoad Command at VolatileKey Creation

VolUsage Field Name	Byte	Bit	Description
AuthOK	0	0	1b = Auth command can be run using this key. 0b = Auth command is prohibited.
EncryptOK	0	1 to 2	00b = Encrypt command is prohibited. 01b = Encrypt command can be run using this key without a prior authentication.(1) 10b or 11b = Encrypt command can be run using this key only with a prior authentication.(1)
DecryptOK	0	3	1b = Decrypt command can be run using this key. 0b = Decrypt command is prohibited.
RandomNonce	0	4	1b = Operations using this key require a random Nonce (see Section 7.19, Nonce Command). 0b = A fixed (input-only) Nonce is permitted.
AuthCompute	0	5	1b = AuthCompute command can be run using this key. 0b = AuthCompute command is prohibited.
LegacyOK	0	6	1b = Legacy command can be run using this key. 0b = Legacy command is prohibited.
Reserved	0	7	Reserved for future use. All bits must be 0b.
WriteCompute	1	0	1b = WriteCompute command can be run using this key. 0b = WriteCompute command is prohibited.
DecRead	1	1	1b = DecRead command can be run using this key. 0b = DecRead command is prohibited.
Reserved	1	2 to 7	Reserved for future use. All bits must be 0b.

Note: 1. **Warning:** Since the Encrypt command does not include an input MAC, the Encrypt command can be exhaustively run with selected input data to attack VolatileKey. Requiring authentication prior to allowing encryption makes these attacks more difficult. To implement this, the Auth and RandomNonce bits must be set to 1b, and the Encrypt bits must be set to 10b or 11b when the VolatileKey is created.

4.4 Counter Configuration

The CounterConfig Registers impose restrictions on the usage of the `Counter` command with a Counter (see Section 7.5, `Counter` Command). There is one CounterConfig Register for each Counter. Each Counter can increment up to a value of 2,097,134 using the `Count` command, after which they can no longer be changed. See Appendix H, Understanding Counters for additional Counter information.

The CounterConfig bits have no impact on the functionality of a Key Usage Counter. If a Counter is identified in a KeyConfig Register (see Section 4.2, Key Configuration) as a Key Usage Counter, then the Counter will increment each time the key is used. The CounterConfig[CntID].IncrementOK bit is typically set to `0b` to prohibit the `Counter` command from incrementing a Key Usage Counter.

Table 4-4. CounterConfig Register Bit Definitions (1)(2)

CounterConfig Field	Byte	Bit	Description
IncrementOK	0	0	`1b` = Increments using the `Counter` command are permitted. `0b` = Increments using the `Counter` command are prohibited.
RequireMAC	0	1	`1b` = The increment operation requires an input MAC. `0b` = An input MAC is prohibited.
Reserved	0	2 to 7	Reserved for future use. All bits must be `0b`.
IncrID	1	0 to 3	KeyID of the key used to generate the `Counter` command input MAC for increment operations.
MacID	1	4 to7	KeyID of the key used to generate the `Counter` command output MAC for Counter Read operations.

Notes:
1. Changes to the CounterConfig Registers take effect immediately, allowing the functionality to be verified during the personalization process.
2. **Warning:** The ATAES132A must always be locked by the customer prior to shipment to the end user to protect the customer secrets. See Section 7.18, `Lock` Command.

5. Standard Serial EEPROM Read and Write Commands

This section provides a summary of the operations that can be performed using the standard Serial EEPROM Read and Write commands. For detailed information, see the specification sections that are referenced below.

Table 5-1. Standard Serial EEPROM Read and Write Commands

Name	Description
Read	The Read command is used to read cleartext from the user zones, to retrieve a response by reading the Response Memory Buffer, or to read the STATUS Register.
Write	The Write command is used to write cleartext to unrestricted memory or to send a command by writing the command packet to the Command Memory Buffer. The Write command is also used to write the IO Address Reset Register.

5.1 Read

The ATAES132A supports the standard Serial EEPROM commands to read from the User Memory. All bytes in the User Memory address space may be read; however, only bytes in the user zones in which neither authentication nor encryption is required will return the actual data from the memory. All other locations will return the value `0xFF`. See Appendix J, I^2C Interface for I^2C Read command information and Appendix K, SPI Interface for SPI Read command information.

When a Read command is received, the device looks at the AuthRead and EncRead bits in the ZoneConfig Register for the user zone to determine whether to return `0xFF` or the EEPROM data. If the EncRead bit is `1b` or the AuthRead bit is `1b`, then `0xFF` will always be returned.

If the ZoneConfig AuthRead bit is `1b` and the EncRead is `0b`, then the `BlockRead` command must be used to read the user zone (see Section 7.4, `BlockRead` Command). If the EncRead bit is `1b`, then the `EncRead` command must be used to read the user zone (see Section 7.9, `EncRead` Command).

The standard SPI and I^2C Read commands can be used to read any number of bytes in a single operation. `Read` operations can cross EEPROM page boundaries.

5.1.1 Read the Response Memory Buffer

The Host sends the ATAES132A commands to the device by writing the command packet to the Command Memory Buffer using a standard SPI or I^2C Write command. ATAES132A processes the command packet and places the response in the Response Memory Buffer. The Host retrieves the response by reading the response packet using a standard SPI or I^2C Read command. See Appendix D, Command Memory Map for additional information. See Appendix G, Understanding the STATUS Register for examples.

When any error occurs, the EERR bit of the STATUS Register is set to 1b to indicate an error. See Appendix G.1, Device Status Register (STATUS) Definition for more information.

5.1.2 Read the Key Memory or Configuration Memory

Reading the Key Memory is never allowed.

The Read command can never be used to read data from the Configuration Memory. The `BlockRead` command is used to access the Configuration Memory (see Section 7.4, `BlockRead` Command).

If a standard SPI or I^2C Read command is used within the Configuration Memory or Key Memory address space, then `0xFF` will be returned for each byte. `0xFF` is also returned for address locations that do not physically exist. The EERR bit of the STATUS Register is set to `1b` if `0xFF` was substituted for any byte returned by a read command. See Appendix G.1, Device Status Register (STATUS) Definition for more information.

5.1.3 Read the STATUS Register

The Host reads the STATUS Register by reading address 0xFFF0. In SPI interface mode, the Host can also read STATUS using the RDSR command. See Appendix G, Understanding the STATUS Register, for detailed information and examples.

5.2 Write

The ATAES132A supports the standard Serial EEPROM commands to write to unrestricted User Memory (AuthWrite and EncWrite are both 0b). See Appendix J, I²C Interface for I²C Write command information and Appendix K, SPI Interface for SPI Write command information. The ATAES132A is capable of writing 1 to 32 bytes on a single physical page with each Write operation.

The Write command can only write data to a single user zone; the data cannot span multiple user zones. The Write command can write data only to a single EEPROM page; the data cannot cross page boundaries. The EERR bit of the STATUS Register is set to 1b to indicate an error if a prohibited Write is attempted. See Appendix G.1, Device Status Register (STATUS) Definition for more information.

5.2.1 Write the Command Memory Buffer

The Host sends the ATAES132A commands to the device by writing the command packet to the Command Memory Buffer using a standard SPI or I²C Write command. The ATAES132A processes the command packet and places the response in the Response Memory Buffer. The Host retrieves the response by reading the response packet using a standard SPI or I²C Read command. See Appendix D, Command Memory Map for additional information. See Appendix G, for examples.

When any error occurs, either the EERR or CRCE bit of the STATUS Register is set to 1b to indicate an error. See Appendix G.1 for more information.

5.2.2 Write the IO Address Reset Register

The Host can reset the pointer in the Command Memory Buffer and the Response Memory Buffer by writing to address 0xFFFE. See Appendix D.3, IO Address Reset Register for additional information.

5.2.3 Write the Key Memory or Configuration Memory

The ATAES132A supports the standard Serial EEPROM commands to write the Configuration Memory or the Key Memory prior to locking. The ATAES132A is capable of writing 1 to 32 bytes on a single physical page with each Write operation.

Partial writes to key registers are prohibited.

If LockKeys has a value of 0x55 (unlocked) and the address points to Key Memory, then the starting address must be the first byte of a key register, and 16 bytes of cleartext data must be sent. If these conditions are not satisfied, then an error response will be generated and the EEPROM will remain unchanged.

If LockConfig has a value of 0x00 (locked) and the address points to the Configuration Memory, then a Write command will generate an error and the EEPROM will be unchanged.

If LockConfig has a value of 0x55 (unlocked), then the User Zone write restrictions imposed by ZoneConfig are enforced, but can be changed.

Atmel does not recommend writing secret data into the User Zones prior to locking the Configuration Memory due to the fact an attacker can change the ZoneConfig bits to allow a read of the User Zone if the Configuration Memory is unlocked.

When any error occurs, either the EERR bit or the CRCE bit of the STATUS Register is set to 1b to indicate an error. See Appenidx G.1 for more information. See the Lock command (Section 7.18.1, User Zone ReadOnly Activation) for additional information.

6. Commands

6.1 Command Block and Packet

The Host sends the ATAES132A extended commands to the device in a block of at least nine bytes. The ATAES132A responses are returned to the Host in a block of at least four bytes. The command and response blocks are constructed in the following manner:

Table 6-1. Command and Response Blocks Descriptions

Byte	Name	Meaning
0	Count	Number of bytes to be transferred to the device in the block, including Count, Packet, and Checksum. This byte will always have a value of N.
1 to (N-3)	Packet	Command, parameters, and data or response. Data is transmitted in the byte order shown in the command definitions.
N-2, N-1	Checksum	Atmel CRC-16 verification of the Count and Packet bytes. See Appendix M, Block Checksum for additional information and examples.

Table 6-2. Input Command Packet Descriptions within the Command Block

Byte	Name	Meaning
1	Opcode	Command Code
2	Mode	Command Modifier
3, 4	Param1	First Command Parameter
5, 6	Param2	Second Command Parameter
7+	Data	Optional Input Data

Table 6-3. Response Packet Descriptions within the Response Block

Byte	Name	Meaning
1	ReturnCode	Command Return Code (See Section 6.3, ReturnCode)
2+	Data	Optional Output Data

Table 6-4. Response Packet Descriptions Contains when an Error Occurs

Byte	Name	Meaning
1	ReturnCode	Error Code (See Section 6.3, ReturnCode)

The Host sends the ATAES132A commands to the device by writing the command block to the Command Memory Buffer using a standard SPI or I^2C Write command. The ATAES132A processes the Command Packet and places the response block in the Response Memory Buffer. The Host retrieves the response by reading the response block using a standard SPI or I^2C Read command. If the Host reads beyond the end of the block, then `0xFF` is returned.

6.2 Command Summary

Table 6-5 shows the command set sorted by the opcode value. Table 6-6 shows the command set in alphabetical order by command name. See Section 7, Command Definitions for the ATAES132A command definitions.

Table 6-5. Extended ATAES132A Command Set Sorted by Opcode Value

Opcode[1]	Name	Description
0x00	Reset	Resets the device, clearing the cryptographic status.
0x01	Nonce	Generates a 128-bit Nonce from the internal RNG for use by the cryptographic commands. This command can also be used to write a Host Nonce directly into the Nonce Register.
0x02	Random	Returns a 128-bit random number from the internal RNG.
0x03	Auth	Performs one-way or mutual authentication using the specified key.
0x04	EncRead	Encrypts 1 to 32 bytes of data from User Memory and returns the encrypted data and integrity MAC.
0x05	EncWrite	Writes 1 to 32 bytes of encrypted data into the User Memory or Key Memory after verifying the integrity MAC.
0x06	Encrypt	Encrypts 16 or 32 bytes of plaintext data provided by the Host.
0x07	Decrypt	Decrypts 16 or 32 bytes of data provided by the Host after verifying the integrity MAC.
0x08	KeyCreate	Generates a random number, stores it in Key Memory, and returns the encrypted key to the Host.
0x09	KeyLoad	Writes an encrypted key to Key Memory after verifying the integrity MAC.
0x0A	Counter	Increments a High Endurance Counter and/or returns the Counter value.
0x0B	Crunch	Processes a seed value through the internal crunch engine. This function is used to detect clones.
0x0C	Info	Returns device information: MacCount, Authentication status, or hardware revision code.
0x0D	Lock	Permanently locks the Configuration Memory or Key Memory. Locked memory can never be unlocked.
0x0F	Legacy	Performs a single AES-ECB mode operation on 16 bytes of data provided by the Host.
0x10	BlockRead	Reads 1 to 32 bytes of data from User Memory or the Configuration Memory. Returns cleartext data.
0x11	Sleep	Places the device in the Sleep state or Standby state to reduce power consumption.
0x13	NonceCompute	Generates a Nonce in a manner that allows two ATAES132A devices to have identical Nonce values.
0x14	AuthCompute	Computes the input MAC required to execute the Auth command or to increment a counter using the Counter command on a second ATAES132A device.
0x15	AuthCheck	Checks the output MAC generated by the Auth command or by reading a counter using the Counter command on a second ATAES132A device.
0x16	WriteCompute	Encrypts data and generates the input MAC required to execute the EncWrite command.
0x17	DecRead	Checks the output MAC and decrypts data that was encrypted by the EncRead command.
0x19	KeyImport	Decrypts and writes a key that was output by the KeyCreate command.
0x1A	KeyTransfer	Transfers a key from User Memory into the Key Memory or into the VolatileKey Register.

Note: 1. The most-significant three bits of the command opcode may contain any value; they are ignored by the ATAES132A command decoder.

Table 6-6. Extended ATAES132A Command Set Sorted by Command Name

Opcode[1]	Name	Description
0x03	Auth	Performs one-way or mutual authentication using the specified key.
0x15	AuthCheck	Checks the output MAC generated by the Auth command or by reading a counter using the Counter command on a second ATAES132A device.
0x14	AuthCompute	Computes the input MAC required to execute the Auth command or to increment a counter using the Counter command on a second ATAES132A device.
0x10	BlockRead	Reads 1 to 32 bytes of data from User Memory or the Configuration Memory. Returns cleartext data.
0x0A	Counter	Increments a high endurance Counter and/or returns the counter value.
0x0B	Crunch	Processes a seed value through the internal crunch engine. This function is used to detect clones.
0x17	DecRead	Checks the output MAC and decrypts data that was encrypted by the EncRead command.
0x07	Decrypt	Decrypts 16 or 32 bytes of data provided by the Host after verifying the integrity MAC.
0x04	EncRead	Encrypts 1 to 32 bytes of data from User Memory and returns the encrypted data and integrity MAC.
0x06	Encrypt	Encrypts 16 or 32 bytes of plaintext data provided by the Host.
0x05	EncWrite	Writes 1 to 32 bytes of encrypted data into the User Memory or Key Memory after verifying the integrity MAC.
0x0C	Info	Returns device information: MacCount, Authentication status, or hardware revision code.
0x08	KeyCreate	Generates a random number, stores it in Key Memory, and returns the encrypted key to the Host.
0x19	KeyImport	Decrypts and writes a key that was output by the KeyCreate command.
0x09	KeyLoad	Writes an encrypted key to Key Memory after verifying the integrity MAC.
0x1A	KeyTransfer	Transfers a key from User Memory into the Key Memory or into the VolatileKey Register.
0x0F	Legacy	Performs a single AES-ECB mode operation on 16 bytes of data provided by the Host.
0x0D	Lock	Permanently locks the Configuration Memory or Key Memory. Locked memory can never be unlocked.
0x01	Nonce	Generates a 128-bit Nonce from the internal RNG for use by the cryptographic commands. This command can also be used to write a Host Nonce directly into the Nonce Register.
0x13	NonceCompute	Generates a Nonce in a manner that allows two ATAES132A devices to have identical Nonce values.
0x02	Random	Returns a 128-bit random number from the internal RNG.
0x00	Reset	Resets the device, clearing the cryptographic status.
0x11	Sleep	Places the device in the Sleep state or Standby state to reduce power consumption.
0x16	WriteCompute	Encrypts data and generates the input MAC required to execute the EncWrite command.

Note: 1. The most-significant three bits of the command opcode may contain any value; they are ignored by the ATAES132A command decoder.

6.3 ReturnCode

The response packet for each ATAES132A command includes a ReturnCode to report success or failure to the Host.

The `Reset` command and the `Sleep` command do not generate a ReturnCode because they do not generate a response packet. All other ATAES132A commands generate a ReturnCode.

Table 6-7. ReturnCode Field Sorted By Value

Value	Name	Notes
`0x00`	Success	No errors.
`0x02`	BoundaryError	Crossed a page boundary for a Write, BlockRead, or EncRead; crossed a Key Register boundary for a Write or EncWrite.
`0x04`	RWConfig	Access to the specified user zone is not permitted due to the configuration or internal state.
`0x08`	BadAddr	Attempted to Write Locked Memory, address is not implemented, or address is illegal for this command.
`0x10`	CountErr	Counter limit reached, count usage error, or restricted key error.
`0x20`	NonceError	Nonce invalid or not available, Nonce not generated with internal RNG. MacCount limit has been reached.
`0x40`	MacError	Missing input MAC, or MAC compare failed.
`0x50`	ParseError	Bad opcode, bad mode, bad param, invalid length, or other encoding failure.
`0x60`	DataMatch	EEPROM post-write automatic data verification failed due to data mismatch.
`0x70`	LockError	`Lock` command contained bad Checksum or bad MAC.
`0x80`	KeyErr	Key not permitted to be used for this operation or wrong key was used for operation. Prior authentication has not been performed. Other authentication error or other key error.

If ReturnCode has any value other than `0x00`, no additional data will be returned by the ATAES132A. If the ReturnCode is greater than zero for any command that performs cryptographic operations, then the Nonce will be invalidated. A non-zero ReturnCode only reports the first error encountered; although, multiple errors might exist.

7. Command Definitions

The ATAES132A extended command definitions are described in this section. The commands are presented in alphabetical order by command name. The standard Serial EEPROM Read and Write commands are in Section 5, Standard Serial EEPROM Read and Write Commands and are not included in this section. The cryptographic operations performed by the ATAES132A extended commands are described in Appendix I, Cryptographic Computations.

7.1 Auth Command

The Auth command performs a one-way or mutual authentication using AES-CCM. The Auth command options are shown in Table 7-1. The Nonce Register value is used as the CCM Nonce for all Auth command MAC calculations.

- **Mutual Authentication**
  The InMAC is verified, and upon success, an OutMAC is calculated and returned to the Host. The AuthComplete status flag is set to YesAuth if the InMAC is verified.
- **Outbound Only Authentication**
  The OutMAC is calculated and output to the Host. The AuthComplete status flag is set to NoAuth. Outbound-only Authentication is also known as Challenge-Response Authentication.
- **Inbound Only Authentication**
  The InMAC value is verified, and the success or failure is reported to the Host. The AuthComplete status flag is set to YesAuth if the InMAC is verified.
- **Authentication Reset**
  The AuthComplete status flag is set to NoAuth.

Table 7-1. Auth Command Options

Mode Bit 1	Mode Bit 0	Description	InMAC	OutMAC
1b	1b	Mutual Authentication	Required	Generated
1b	0b	Outbound Only Authentication	Prohibited	Generated
0b	1b	Inbound Only Authentication	Required	No
0b	0b	Authentication Reset	Prohibited	No

If a MAC is required or will be generated by the Auth command, then a valid Nonce is required. If the KeyConfig[AKeyID].RandomNonce bit is 1b, then the Nonce must be random.

The AuthCompute command can be used to generate the InMac required for Inbound Only Authentication, or Mutual Authentication (see Section 7.3, AuthCompute Command). The AuthCheck command can be used to validate the OutMac (see Section 7.2, AuthCheck Command).

In the I^2C interface mode, the Auth command can also used for Auth signaling. See Appendix J.5, I^2C Auth Signaling.

7.1.1 Authentication Status Register

The Authentication Status Register contains the AKeyID, the AuthComplete status flag, and the usage bits. Prior to executing the Auth command, the AuthComplete status flag is set to NoAuth. If the InMAC is successfully verified in the Inbound Only or Mutual Authentication mode, then the AuthComplete status flag is set to YesAuth.

The ATAES132A Authentication Status Register only stores the result of the most recent authentication attempt. If there is a parsing or execution error, then the prior Authentication status will be lost.

7.1.2 Authentication Usage

The usage field (Param2) controls which operations are permitted with a successful Inbound-only or Mutual Authentication (see Table 7-2). If Param2 is 0x0000, the AuthComplete flag is set to NoAuth, but the authentication outputs are generated. Param2 is ignored if outbound-only authentication is performed.

Table 7-2. Auth Command Usage Field Definition (Param2)

Byte	Bit	Name	Notes
0	0	ReadOK	1b = Read and EncRead commands are enabled for user zone reads after successful authentication. 0b = Read and EncRead commands are prohibited for user zone reads if authentication is required in ZoneConfig[UZ] (see Section 4.1, User Zone Configuration).
0	1	WriteOK	1b = Write and EncWrite commands are enabled for user zone writes after successful authentication. 0b = Write and EncWrite commands are prohibited for user zone writes if authentication is required in ZoneConfig[UZ] (see Section 4.1).
0	2	KeyUse	1b = If a key requires authentication (KeyConfig[AKeyID].AuthKey is 1b), the Encrypt, Decrypt, Legacy, KeyCreate, and KeyLoad commands are enabled after successful authentication. 0b = EncRead, EncWrite, Encrypt, Decrypt, Legacy, KeyCreate, and KeyLoad commands using the authenticated key are prohibited after authentication (see Section 4.2, Key Configuration).
0	3 – 7	Zero	Reserved. Must be 0b.
1	0:7	Zero	Reserved. Must be 0x00.

If the AKeyID is VolatileKey, then VolUsage.AuthOK must be 1b when the key is loaded or authentication will fail.

Table 7-3. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Auth	1	0x03
Mode	Mode	1	Bit 0 and Bit 1 if: 11b = Perform mutual authentication. 10b = Perform Outbound Only authentication. 01b = Perform Inbound Only authentication. 00b = Perform authentication reset. Bits 2, 3, and 4: Reserved. Must be 0b. Bit 5: 1b = Include the associated Usage Counter in the authentication. Bit 6: 1b = Include SerialNum in the authentication. Bit 7: 1b = Include the first four bytes of SmallZone in the authentication.
Param1	AKeyID	2	Upper byte is always 0x00. Lower byte is the pointer to the key. Legal values: 0x00 to 0x0F, 0xFF.
Param2	Usage	2	Authentication usage restrictions. Ignored if Mode bits 0 and 1 are 00b or 10b.
Data	InMac	0 or 16	Input MAC to be verified (see Appendix I.3, MAC Generation).

Table 7-4. Output parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutMac	0 or 16	If an output MAC generation was required (and any optional input MAC verification succeeded), then a 16-byte MAC will be returned.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.2 AuthCheck Command

The AuthCheck command is used to check the OutMAC generated by the Auth command or the Counter command on a second ATAES132A device. This command cannot check MACs created by other commands.

To use this command, the Nonce must be identical on both devices (see Section 7.20.1, Nonce Synchronization), and the MacCount must have the same value. Both devices must also contain identical key values, but it is not necessary for the KeyID on the origin device to match the KeyID on the destination device. In this section, the device that generates the MAC is referred to as the origin device, and the device that checks the MAC is referred to as the destination device.

If Mode bit 5, 6, or 7 is 1b, then the associated Usage Counter, SerialNum Register value, or the first four bytes of the SmallZone Register in the SecondBlock field must match the values on the origin device. The ManufacturingID Register must be identical on both devices, since it is always included in the MAC calculation.

A valid Nonce is required to run the AuthCheck command. If the KeyConfig[MacKeyID].RandomNonce bit is 1b, then the Nonce must be random.

The AuthCheck command always sets the AuthComplete status flag to NoAuth.

Table 7-5. Input Parameters

	Name	Size (bytes)	Notes
Opcode	AuthCheck	1	0x15
Mode	Mode	1	Always 0x0000.
Param1	MacKeyID	2	Upper byte is always 0x00. Lower byte is the pointer to the key. Legal values: 0x00 to 0x0F, 0xFF.
Param2	Zero	2	Always 0x0000.
Data1	FirstBlock	11	The value of this field must match the first authenticate-only block used to calculate the MAC on the origin device.
Data2	SecondBlock	16	The value of this field must match the second authenticate-only block used to calculate the MAC being checked on the origin device. If Mode bits 5, 6, and 7 are 0b, then this field must be present, but is ignored.
Data3	InMac	16	MAC to be checked.

Table 7-6. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.3 AuthCompute Command

The AuthCompute command is used to compute a MAC that will be used to execute the Auth command or the Counter command on a second ATAES132A device.

To use this command, the Nonce must be identical on both devices (see Section 7.20.1, Nonce Synchronization) and the MacCount must have the same value. Both devices must also contain identical key values, but it is not necessary for the KeyID on the origin device to match the KeyID on the destination device. In this section, the device that generates the MAC is referred to as the origin device, and the device that checks the MAC is referred to as the destination device.

If Mode bit 5, 6, or 7 is 1b, then the associated Usage Counter, SerialNum Register value, or the first four bytes of the SmallZone Register in the SecondBlock field must match the values on the destination device. The ManufacturingID Register must be identical on both devices, since it is always included in the MAC calculation.

A valid Nonce is required to run the AuthCompute command. If the KeyConfig[MacKeyID].RandomNonce bit is 1b, then the Nonce must be random.

The AuthCompute command always sets the AuthComplete status flag to NoAuth. This command can only be executed if it is enabled for the device by setting ChipConfig.AuthComputeE to 1b.

Table 7-7. Input Parameters

	Name	Size (bytes)	Notes
Opcode	AuthCompute	1	0x14
Mode	Mode	1	Always 0x0000.
Param1	MacKeyID	2	Upper byte is always 0x00. Lower byte is the pointer to the key. Legal values: 0x00 to 0x0F, 0xFF.
Param2	Zero	2	Always 0x0000.
Data1	FirstBlock	11	The value of this field must match the first authenticate-only block to be used when executing the Auth command or the Counter command on the destination device.
Data2	SecondBlock	16	The value of this field must match the second authenticate-only block to be used when executing the Auth command or Counter command on the destination device. If Mode bits 5, 6, and 7 are 0b, then this field must be present, but is ignored.

Table 7-8. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutMac	16	The 16-byte MAC.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.4 BlockRead Command

The BlockRead command reads 1 to 32 bytes of plaintext data from a User Zone or the Configuration Memory. This command differs from the standard Serial EEPROM Read commands, since it can read the Configuration Memory. In addition, this command returns an error code if the Read is unsuccessful. No encryption is performed by the BlockRead command; the EncRead command must be used for encrypted reads (see Section 7.9, EncRead Command).

The BlockRead command can only read data from a single EEPROM page; the requested data cannot cross page boundaries (see Appendix B.2, EEPROM Page Boundary). All bytes within the Configuration Memory can be read with the BlockRead command. If any part of the requested data lies in unimplemented or illegal memory, the command will generate an error code. The Key Memory can never be read under any circumstances; any attempt to read the Key Memory will generate an error code.

User Zone access is dependent upon the value of the EncRead and AuthRead bits of the ZoneConfig[UZ] register. If ZoneConfig[UZ].AuthRead is 0b, then BlockRead can read the user zone. If ZoneConfig[UZ].AuthRead is 1b, then BlockRead can only be used to access the user zone if the authentication requirement has been satisfied. If ZoneConfig[UZ].EncRead is 1b, then BlockRead can never be used to access the user zone. A single BlockRead command can read data from only a single User Zone; the requested data cannot span multiple user zones or multiple EEPROM pages.

Table 7-9. Input Parameters

	Name	Size (bytes)	Notes
Opcode	BlockRead	1	0x10
Mode	Mode	1	Must be 0x00.
Param1	Address	2	The address of data to read.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of bytes to read.

Table 7-10. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutData	0 – 32	Output data (cleartext).

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.5 Counter Command

The Counter command reads or increments the internal, high endurance counters. Each counter can increment up to a value of 2,097,151 using the Count command, after which they can no longer be changed. See Appendix H, Understanding Counters for additional counter information.

Table 7-11. Counter Command Options

Mode bit 1	Mode bit 0	Description	InMAC	OutMAC
1b	1b	Read Counter with MAC	Prohibited	Generated
0b	1b	Read Counter, No MAC	Prohibited	No
1b	0b	Increment Counter with MAC	Required	No
0b	0b	Increment Counter, No MAC	Prohibited	No

The CounterConfig[CntID].RequireMAC Register bit determines if InMAC is required when incrementing the counter (see Section 4.4, Counter Configuration). If CounterConfig[CntID].RequireMAC = 1b, then InMAC is required, and so Mode bit 1 must be set to 1b when incrementing the counter. If CounterConfig[CntID].RequireMAC is 0b, then InMAC is prohibited, and Mode bit 1 must be set to 0b.

If a MAC is required or generated, then a valid Nonce is required to run the Counter command. If the KeyConfig[KeyID].RandomNonce bit is set for the authorizing key, then the Nonce must be random.

The AuthCompute command can be used to generate InMac (see Section 7.3, AuthCompute Command). The AuthCheck command can be used to validate OutMac (see Section 7.2, AuthCheck Command).

Table 7-12. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Counter	1	0x0A
Mode	Mode	1	Bit 0: 1b = Read the Counter. 0b = Increment the Counter. Bit 1: 1b = InMAC is included in the input packet if bit 0 is 0b, or OutMAC is generated if bit 0 is 1b. 0b = Neither the input nor output packets include a MAC. Bits 2 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with the key(1) used to generate the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	CountID	2	Upper byte is always 0x00. Upper nibble of lower byte is always 0x0. Lower nibble of lower byte is the counter to be queried.
Param2	Zero	2	Always 0x0000.
Data	InMac	0 or 16	Integrity MAC for the counter increment operation.

Note: 1. The MAC is generated using the key identified by the KeyID in CounterConfig[CountID].IncrID for increment operations, or the KeyID in CounterConfig[CountID].MacID for Counter Read operations. The Usage Counter included in the MAC when Mode bit 5 is 1b is identified by the CntID stored in KeyConfig[KeyID].CounterNum for the key used to generate the MAC.

Table 7-13. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
CountValue	4	The current value of the Counter.
OutMac	0 or 16	Integrity MAC for the Counter Read operation.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

The equivalent decimal value of the Counter can be determined using the following equation:

$$CountValue = (BinCount*32) + (CountFlag/2)*8 + Lin2Bin(LinCount)$$

Here, Lin2Bin defines a function that converts a linear counter value to corresponding binary value. 0xFFFF converts to zero, 0xFFFE converts to one, and up to 0x8000 which converts to 15.

7.6 Crunch Command

The Crunch command processes a seed value and returns the result within a specified time. The command provides a 16-byte input seed, which is combined with the ManufacturingID Register and processed with the internal hardware crunch calculator. The calculation is performed within a specified time period.

The Host system should read the response within a few milliseconds after the response is specified to be available and compare the returned value to the expected result to determine if authentic Atmel hardware is present. The crunch algorithm is proprietary, and is available only in authentic Atmel hardware.

The Crunch command does not use the AES engine or the Nonce. Executing the Crunch command does not change the authentication status or cryptographic state of the device.

7.6.1 Crunch Response Time

The response to the Crunch command is available after a period of time that is dependent on the Count field value. A large Count value requires more time to process than a small Count value. The expected response time for the Crunch command is computed using the following equation:

$$((count \times 256) + 600)) \times 1.25 \text{ microseconds}$$

Table 7-14. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Crunch	1	0x0B
Mode	Mode	1	Must be 0x00.
Param1	Count	2	Upper byte is always 0x00. Lower byte is the iteration count for the crunch engine.
Param2	Zero	2	Always 0x0000.
Data	Seed	16	Input seed.

Table 7-15. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
Result	16	Result out.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.7 DecRead Command

The DecRead command is used to check the OutMAC generated by an EncRead command on a second ATAES132A device. If the MAC matches, then the 1 to 16 bytes of data is returned to the Host in the DecRead response.

To use this command, the Nonce must be identical on both devices (see Section 7.20.1, Nonce Synchronization), and the MacCount must have the same value. Both devices must also contain identical key values, but it is not necessary for the KeyID on the origin device to match the KeyID on the destination device. In this section, the device that encrypts the data and generates the MAC is referred to as the origin device, and the device that checks the MAC is referred to as the destination device.

If Mode bit 5, 6, or 7 is 1b, then the associated Usage Counter, SerialNum Register value, or the first four bytes of the SmallZone Register in the SecondBlock field, must match the values on the origin device. The ManufacturingID Register must be identical on both devices, since it is always included in the MAC calculation.

A valid Nonce is required to run the DecRead command. If the KeyConfig[DKeyID].RandomNonce bit is 1b, then the Nonce must be random. This command can be executed only if it is enabled for the device by setting ChipConfig.DecReadE to 1b.

Table 7-16. Input Parameters

	Name	Size (bytes)	Notes
Opcode	DecRead	1	0x17
Mode	Mode	1	Always 0x0000.
Param1	DKeyID	2	Upper byte is always 0x00. Lower byte is the pointer to the decrypt key. Legal values: 0x00 to 0x0F, 0xFF.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of data bytes to be decrypted.
Data1	FirstBlock	6	The value of this field must match the first authenticate-only block used when executing the EncRead command on the origin device.
Data2	SecondBlock	16	The value of this field must match the second authenticate-only block used when executing the EncRead command on the origin device. If Mode bits 5, 6, and 7 are 0b, then this field must be present, but is ignored.
Data3	InMac	16	Integrity MAC for the input data.
Data4	InData	16	Input data (ciphertext) to be decrypted.

Table 7-17. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutData	1 to 16	Decrypted (plaintext) output data.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.8 Decrypt Command

The Decrypt command accepts 16 or 32 bytes of ciphertext, decrypts the data, verifies the MAC, and returns the decrypted data if the MAC matches. If the MAC does not match, then an error code is returned.

The Decrypt command has two operating modes:

- Normal Decryption Mode
- Client Decryption Mode

The Client Decryption mode decrypts packets encrypted by an ATAES132A device. The Normal Decryption mode decrypts packets generated by a cryptographic Host. It cannot decrypt packets encrypted by an ATAES132A device.

- If the DKeyID is VolatileKey (see Section 4.3, VolatileKey Configuration), the VolUsage.DecryptOK must be 1b when VolatileKey was loaded.
- If the DKeyID is not VolatileKey, the KeyConfig[DKeyID].ExternalCrypto bit must be 1b.
- If the KeyConfig[DKeyID].AuthKey bit is 1b, prior authentication must be performed using the KeyID stored in KeyConfig[DKeyID].LinkPointer.

A valid Nonce is required to run the Decrypt command. If the KeyConfig[DKeyID].RandomNonce bit is 1b, then the Nonce must be random.

7.8.1 Client Decryption Mode

In the Client Decryption mode, the Decrypt command can be used to decrypt packets encrypted by the ATAES132A (either another device, or by the same device at a later time) using the Encrypt command (see Section 7.10, Encrypt Command). All of the following requirements must be satisfied:

1. The device performing the Encrypt operation (the Encrypt Device) and the device performing the Decrypt operation (the Decrypt Device) must contain identical keys.
2. The KeyID of the key used by the Encrypt Device (called EKeyID) must be known. EKeyID is passed to the Decrypt Device in the upper byte of Decrypt Param1 for use in the MAC calculation.
3. The Nonce used by the Encrypt Device must be known. The Nonce is passed to the Decrypt Device using the Nonce command with Mode bit 0 = 0b (see Section 7.19, Nonce Command), or is synchronized with the Encrypt Device using the procedure in Section 7.20.1, Nonce Synchronization.
4. The lower byte of the Count (Encrypt Param2) used by the Encrypt Device must identical to the value used in the lower byte of Decrypt Param2 by the Decrypt Device. (This is used in the MAC calculation.)
5. The MacCount of the Encrypt Device (called EMacCount) must be known. EMacCount is passed to the Decrypt Device in the upper byte of Decrypt Param2 for use in the Data Decryption operation.
6. The Encrypt/Decrypt command mode bits on both devices must be identical. Mode bit 5 must be 0b. Mode bit 6 must be 0b, unless a single device is performing both the Encrypt and the Decrypt operations. Mode bit 7 can be 1b if the first four bytes of SmallZone are identical on both the Encrypt and the Decrypt Devices.
7. The Decrypt Device KeyConfig[DKeyID] must have ExternalCrypto = 1b and RandomNonce = 0b for the KeyID used for decryption if the Nonce is passed using the Nonce command with Mode bit 0 = 0b.
8. The Encrypt Device KeyConfig[EKeyID] must have ExternalCrypto = 1b and RandomNonce = 1b for the KeyID used for encryption (the EKeyID).

If these conditions are satisfied, then packets encrypted on the Encrypt Device can be decrypted on the Decrypt Device. If a single ATAES132A will be used to encrypt packets for later decryption, then the same key value must be stored in two appropriately configured key registers to allow all of the requirements above to be satisfied.

Table 7-18. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Decrypt	1	0x07
Mode	Mode	1	Bits 0 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with the encryption key in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	DKeyID	2	Normal Decryption Mode: • Upper byte is always 0x00. • Lower byte is the KeyID of the decrypt key. Client Decryption Mode: • Upper byte is the EKeyID. • Lower byte is the KeyID of the decrypt key.
Param2	Count	2	Normal Decryption Mode: • Upper byte is always 0x00. • Lower byte is the number of bytes to be returned after decryption. Client Decryption Mode: • Upper byte is the EMacCount. • Lower byte is the number of bytes to be returned after decryption (see Section 7.8.1, Client Decryption Mode).
Data1	InMac	16	Integrity MAC for the input data.
Data2	InData	16 or 32	Input data (ciphertext) to be decrypted.

Table 7-19. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutData	1 – 32	Decrypted (plaintext) output data.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.9 EncRead Command

EncRead reads 1 to 32 bytes of encrypted data from User Memory, along with an integrity MAC. The EncRead command only performs encrypted reads; the BlockRead command is used for unencrypted reads (see Section 7.4, BlockRead Command).

The ZoneConfig[UZ].EncRead bit determines if a user zone can be accessed with the EncRead command. If the ZoneConfig[UZ].EncRead bit is 1b, then the EncRead command can read the user zone if the access requirements have been satisfied. A single EncRead command reads data from a single user zone; the requested data cannot span multiple user zones. A single EncRead command reads data from a single EEPROM page; the requested data cannot cross page boundaries (see Appendix B.2, EEPROM Page Boundary).

If ZoneConfig[UZ].Auth is 1b, then prior authentication is required with the following restrictions:

- The Auth command Usage.ReadOK bit must be 1b.
- The Authentication Key AKeyID must match ZoneConfig[UZ].AuthID.
- The Auth command must be run in Inbound Only Authentication or Mutual Authentication mode.
- A valid Nonce is required to run the EncRead command. If KeyConfig[KeyID].RandomNonce for the read key is 1b, then the Nonce must be random.

The DecRead command can be used to validate OutMac and decrypt up to 16 bytes of data (see Section 7.7, DecRead Command).

7.9.1 Configuration Memory Signature

The EncRead command cannot be used to read the Configuration Memory. Only the BlockRead command can be used to read the Configuration Memory. Any attempt to read any address in the Configuration Memory with the EncRead command will activate the Configuration Memory Signature Generation mode.

The Configuration Memory Signature is an AES-CCM MAC generated over the entire Configuration Memory, as described in Appendix I.17, EncRead Command Configuration Memory Signature MAC. A valid Nonce is required to run the EncRead command in Configuration Memory Signature Generation mode. If KeyConfig[00].RandomNonce is 1b, then the Nonce must be random. KeyID 00 is always used to generate the Configuration Memory Signature.

The Configuration Memory Signature Generation mode is intended to be used during secure personalization of the ATAES132A device. The signature can be used to validate the contents of the Configuration Memory prior to programming secret data into other portions of the EEPROM.

7.9.2 Key Memory Signature

The EncRead command cannot be used to read the Key Memory. The Key Memory can never be read. Any attempt to read any address in the Key Memory with the EncRead command will activate the Key Memory Signature Generation mode; however, this signature can be generated only once per unit.

The Key Memory Signature is an AES-CCM MAC generated over all 16 key registers, as described in Appendix I.18, EncRead Command Key Memory Signature MAC. A valid Nonce is required to run the EncRead command in Key Memory Signature Generation mode. If KeyConfig[00].RandomNonce is 1b, then the Nonce must be random. KeyID 00 is always used to generate the Key Memory Signature.

The Key Memory Signature Generation mode is intended to be used during secure personalization of the ATAES132A. The signature can be used to validate the contents of the Key Memory before locking the Key Memory.

Table 7-20. Input Parameters

	Name	Size (bytes)	Notes
Opcode	EncRead	1	0x04
Mode	Mode	1	Bits 0 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with the ZoneConfig[UZ].ReadID key in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	Address	2	The address of data to be read.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of bytes to read.
Data	—	0	

Table 7-21. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutMac	16	Integrity MAC for the output data.
OutData	16 or 32	Encrypted output data (ciphertext).

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.10 Encrypt Command

The Encrypt command accepts 1 to 32 bytes of plaintext, encrypts the data, and generates an integrity MAC. The encrypted data and OutMAC are returned to the system.

The Encrypt command can be used to encrypt packets for decryption by the same or another ATAES132A, if the requirements described in Section 7.8.1, Client Decryption Mode are satisfied.

- If the EKeyID specifies a key in the Key Memory, the KeyConfig[EKeyID].ExternalCrypto bit must be 1b.
- If the KeyConfig[EKeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[EKeyID].LinkPointer.
- If the EKeyID specifies the VolatileKey (see Section 4.3, VolatileKey Configuration), the VolUsage.EncryptOK must be set to 01b, 10b, or 11b.
- If the VolUsage.EncryptOK bits are set to 10b or 11b, then prior authentication is required using VolatileKey prior to execution of the Encrypt command.

A valid Nonce command is required to run the Encrypt command. If the KeyConfig[EKeyID].RandomNonce bit is set for the encryption key, then the Nonce must be random.

Table 7-22. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Encrypt	1	0x06
Mode	Mode	1	Bits 0 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with the encryption key in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	EKeyID	2	Upper byte is always 0x00. Lower byte is the KeyID of the encrypt key.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of bytes to be encrypted.
Data	InData	1 – 32	Input data to be encrypted (plaintext).

Table 7-23. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutMac	16	Integrity MAC for the output data.
OutData	16 or 32	Encrypted data (ciphertext).

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.11 EncWrite Command

The EncWrite command decrypts the ciphertext input data, verifies the input MAC, and then writes 1 to 32 bytes to a User Zone or 16 bytes to Key Memory.

The ZoneConfig[UZ].EncWrite bit determines if a User Zone must be accessed with the EncWrite command. If the ZoneConfig[UZ].EncWrite bit is 1b, then the EncWrite command must be used to write the user zone if the access requirements have been satisfied. If the ZoneConfig[UZ].EncWrite bit is 0b, then a Write command or EncWrite command can be used to write the User Zone. A single EncWrite command writes data to a single User Zone; the data cannot span multiple User Zones. A single EncWrite command writes data to a single EEPROM page; the data cannot cross page boundaries (see Appendix B.2, EEPROM Page Boundary).

If ZoneConfig[UZ].Auth is 1b, then prior authentication is required with the following restrictions:

- The Auth command Usage.WriteOK bit must be 1b.
- The Authentication Key (AKeyID) must match ZoneConfig[UZ].AuthID.
- The Auth command must be run in Inbound-Only Authentication or Mutual Authentication mode.
- A valid Nonce is required to run the EncWrite command. If KeyConfig[KeyID].RandomNonce for the write key is 1b, then the Nonce must be random.

7.11.1 Encrypted Key Writes

When EncWrite is used to write the Key Memory prior to locking, the key data must be encrypted using KeyID 00. The input MAC is also calculated using KeyID 00. Writes to Key Memory must be 16 bytes in length and begin at the starting address of the key.

If LockKeys has a value of 0x55 and the EncWrite address points to Key Memory, then Key Personalization mode is selected. In key Personalization mode, the following requirements are in effect:

- The Count field value must be 16.
- The address must match the starting address of the Key Register.
- The input data must be encrypted with the current value in KeyID 00. If KeyConfig[WriteID].RandomNonce is 1b, then the Nonce must be random (See Section 7.19, Nonce Command).
- The input MAC must be generated with the current value in KeyID 00. The input MAC will be verified.

If the Key Memory is locked, then the new key data is encrypted with the current value of the key being written. The key can be updated only if all of the following requirements are satisfied:

- The corresponding KeyConfig[KeyID].ChangeKeys bit is set to 1b (see Section 4.2, Key Configuration).
- The Count field value must be 16.
- The address must match the starting address of the Key Register.
- The input data must be encrypted with the current value of the Key. If KeyConfig[WriteID].RandomNonce is 1b then Nonce be random (See Section 7.19, Nonce Command).
- The input MAC must be generated with the current value of the Key. The input MAC will be verified (See Section7.18, Lock Command).

Table 7-24. Input Parameters

	Name	Size (bytes)	Notes
Opcode	EncWrite	1	0x05
Mode	Mode	1	Bits 0 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with the encryption key in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	Address	2	The starting address of memory to be written.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of bytes to be written.
Data1	InMac	16	Input MAC to be verified.
Data2	InData	16 or 32	Encrypted Data (ciphertext).

Table 7-25. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.12 INFO Command

The INFO command reads various information about the device from the internal registers. Param1 selects the information to read. Operation of this command does not require knowledge of any secrets.

Table 7-26. Selector Field Coding (Param1)

Selector	Name	Description
0x0000	MacCount	Read the MacCount Register. The first byte is always 0x00; the second byte is the MacCount value.
0x0005	AuthStatus	Read the Authentication Status Register. Returns 0xFFFF to indicate that the AuthComplete status flag = NoAuth. If the AuthComplete status flag = YesAuth, then the info returns the AKeyID as 0x00KK, where KK is the Authentication Key ID.
0x0006	DeviceNum	Read the DeviceNum Register. The first byte is the Atmel device code, which is unique to this Atmel catalog number. The second byte provides the device revision number.
0x000C	ChipState	Read the ChipState Device State Register: • 0x0000 indicates ChipState = Active • 0xFFFF indicates ChipState = Power-Up • 0x5555 indicates ChipState = Wake-up from Sleep See Appendix L.3, Understanding the ChipState Register.
All Other	Reserved	Reserved for future use.

Table 7-27. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Info	1	0x0C
Mode	Mode	1	Must be 0x00.
Param1	Selector	2	Selects the register to read.
Param2	Zero	2	Always 0x0000.
Data	—	0	

Table 7-28. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
Result	2	Current value of the register.

The command and response packet is transmitted as a block, beginning with the count and ending with a packet checksum. This block format is described in Section 6.1, Command Block and Packet.

7.13 KeyCreate Command

The KeyCreate command generates a 16-byte random number, and stores it in either the Key Memory or in the VolatileKey Register. The newly generated key is then encrypted with the parent key and returned to the Host along with a MAC.

If Mode bit 0 is 1b, then the target key is in the Key Memory:

- KeyConfig[ChildKeyID].Child must be 1b.
- The KeyCreate command KeyID field contains the ChildKeyID.
- KeyConfig[ChildKeyID].LinkPointer contains the ParentKeyID.

If Mode bit 0 is 0b, then the target key is VolatileKey:

- An InMac is required.
- KeyConfig[ParentKeyID].Parent must be 1b.
- The KeyCreate command KeyID field contains the ParentKeyID.
- The VolUsage field specifies VolatileKey usage restrictions, as defined in Section 4.3, VolatileKey Configuration.

If KeyConfig[ParentKeyID].AuthKey bit is 1b or the KeyConfig[EKeyID].ChildAuth bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[ParentKeyID].LinkPointer.

InMAC and OutMAC are both calculated using the parent key (ParentKeyID). If KeyConfig[ChildKeyID].ChildMac is 1b, then an InMAC must be provided; otherwise, InMAC will be ignored.

A valid Nonce is required to run the KeyCreate command. If the KeyConfig[ParentKeyID].RandomNonce bit is 1b, then the Nonce must be random.

If the LockConfig Register is unlocked (0x55), then the RNG is latched in Test mode, and the KeyCreate command will generate nonrandom key values. If the LockConfig Register is locked (0x00), then the RNG generates random numbers and the KeyCreate command functions normally.

The KeyImport command can be used to load a key generated by the KeyCreate command (see Section 7.14, KeyImport Command).

There is one RNG Seed Register in the EEPROM memory, which is used by the KeyCreate, Nonce, and Random commands. The RNG Seed Register is subject to the same Write endurance limitations as the other bytes in the EEPROM (see Section 9.2, Reliability for the EEPROM specifications). The application developer must not exceed the write endurance limit.

Table 7-29. Input Parameters

	Name	Size (bytes)	Notes
Opcode	KeyCreate	1	0x08
Mode	Mode	1	Bit 0: 1b = Key load target is Key Memory. 0b = Target is VolatileKey (see Section 4.3, VolatileKey Configuration). An InMac is required. Bit 1: 0b = Update the EEPROM RNG Seed Register prior to key generation.(1) 1b = Generate the key using the existing RNG Seed. Bit 2: 1b = A key equivalent to what the KeyCreate InMac would be is generated. Including an InMac with the KeyCreate command is not required. Bits 3-4: Reserved. Must be zero. Bit 5: 1b = Include the Usage Counter associated with the ParentKeyID in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	KeyID	2	Upper byte is always 0x00. Lower byte is the ChildKeyID for Key Memory loads or the ParentKeyID for VolatileKey loads.
Param2	VolUsage	2	Usage restrictions for VolatileKey if Mode bit 0 is 0b (see Section 4.3).
Data	InMac	0 or 16	Optional input MAC (see above).

Note: 1. The RNG Seed Register in the EEPROM will be updated automatically if Mode bit 1 = 0b, unless the Seed Register was previously updated after the most recent Power-On Reset, Wake from the Sleep state, Reset command, or Tamper Event. Updating the RNG Seed Register increases the randomness of the keys generated by the KeyCreate command; however, the EEPROM Write Endurance specification must be respected.

Table 7-30. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code (see Section 6.3, ReturnCode).
OutMac	16	Output MAC for the encrypted key. If Mode bit 2 = 1b, no OutMac is returned.
OutData	16	Encrypted key value (ciphertext). If Mode bit 2 = 1b, no Data is returned.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.14 KeyImport Command

The KeyImport command accepts 16 bytes of ciphertext, decrypts the key, verifies the MAC, and stores the key in the Key Memory or in the VolatileKey Register. The source of the encrypted key is the KeyCreate command.

- If TargetKeyID specifies that the target key is stored in the Key Memory:
  - The KeyConfig[TargetKeyID].ImportOK bit must be 1b.
  - KeyConfig[TargetKeyID].LinkPointer contains the decrypt KeyID.
  - The KeyImport command DKeyID field value is ignored.
- If the KeyConfig[decrypt KeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[decrypt KeyID].LinkPointer.
- If TargetKeyID specifies that the target key is VolatileKey (see Section 4.3, VolatileKey Configuration):
  - The KeyConfig[DKeyID].Parent bit must be 1b.
  - The KeyImport command DKeyID field contains the decrypt KeyID.
- If the KeyConfig[DKeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[DKeyID].LinkPointer.

To use this command, the Nonce must be identical on both devices (see Section 7.20.1, Nonce Synchronization) and the MacCount must have the same value. Both devices must also contain identical key values, but it is not necessary for the encrypt KeyID on the origin device to match the decrypt KeyID on the destination device. In this section, the device that encrypts the key and generates the MAC is referred to as the origin device, and the device that checks the MAC is referred to as the destination device.

If Mode bit 5, 6, or 7 is 1b, then the associated Usage Counter, SerialNum Register value, or the first four bytes of the SmallZone Register in the SecondBlock field must match the values on the origin device. The ManufacturingID Register must be identical on both devices since it is always included in the MAC calculation.

A valid Nonce is required to run the KeyImport command. If the KeyConfig[KeyID].RandomNonce bit is 1b for the Decrypt Key, then the Nonce must be random.

Table 7-31. Input Parameters

	Name	Size (bytes)	Notes
Opcode	KeyImport	1	0x19
Mode	Mode	1	Bit 0: Reserved. Must be 1b. Bits 1 to 4: Reserved. Must be 0b. Bits 5 to 7: This value must match Mode bits 5, 6, and 7 value used when executing the KeyCreate command on the origin device.
Param1	TargetKeyID	2	Upper byte is always 0x00. Lower byte is the location where the decrypted key will be stored. Legal values: 0x00 to 0x0F (standard keys), 0xFF (volatile key).
Param2	DKeyID	2	Upper byte is always 0x00. If TargetKeyID = 0xFF, then lower byte is the pointer to the decrypt key. Legal values: 0x00 to 0x0F. If TargetKeyID = 0x00 to 0x0F, then this field must be present, but is ignored (see above).
Data1	FirstBlock	6	The value of this field must match the first authenticate-only block used when executing the KeyCreate command on the origin device.
Data2	SecondBlock	16	The value of this field must match the second authenticate-only block used when executing the KeyCreate command on the origin device. If Mode bits 5, 6, and 7 are 0b, then this field must be present, but is ignored.
Data3	InMac	16	MAC for the encrypted key.
Data4	InData	16	Input key (ciphertext) to be decrypted.

Table 7-32. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.15 KeyLoad Command

The KeyLoad command decrypts 16 bytes of ciphertext data, verifies the MAC, and then writes the Key Memory or the VolatileKey Register.

- If Mode bit 0 specifies that the target key is stored in the Key Memory:
  - KeyConfig[ChildKeyID].Child bit must be 1b.
  - KeyLoad command KeyID field contains the ChildKeyID.
  - KeyConfig[ChildKeyID].LinkPointer contains the ParentKeyID.
- If the KeyConfig[ParentKeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[ParentKeyID].LinkPointer.
- If Mode bit 0 specifies that the target key is VolatileKey (see Section 4.3, VolatileKey Configuration):
  - KeyConfig[ParentKeyID].Parent bit must be 1b.
  - KeyLoad command KeyID field contains the ParentKeyID.
  - VolUsage field specifies VolatileKey usage restrictions, as defined in Section 4.3.
- If the KeyConfig[ParentKeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[ParentKeyID].LinkPointer.

A valid Nonce is required to run the KeyLoad command. If the appropriate KeyConfig[KeyID].RandomNonce bit is 1b, then the Nonce must be random.

Table 7-33. Input Parameters

	Name	Size (bytes)	Notes
Opcode	KeyLoad	1	0x09
Mode	Mode	1	Bit 0: 1b = The key load target is Key Memory. 0b = Target is VolatileKey (see Section 4.3). Bits 1 to 4: Reserved. Must be 0b. Bit 5: 1b = Include the Usage Counter associated with ParentKeyID in the MAC. Bit 6: 1b = Include SerialNum in the MAC. Bit 7: 1b = Include the first four bytes of SmallZone in the MAC.
Param1	KeyID	2	Upper byte is always 0x00. Lower byte is the ChildKeyID for the Key Memory loads or the ParentKeyID for VolatileKey loads.
Param2	VolUsage	2	Usage restrictions for VolatileKey if Mode bit 0 is 0b (see Section 4.3).
Data1	InMac	16	Integrity MAC for the input data.
Data2	InData	16	Encrypted key value (ciphertext).

Table 7-34. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.16 KeyTransfer Command

The KeyTransfer command copies key data from the User Memory into the VolatileKey Register or into a Key Register in the Key Memory. The KeyTransfer command allows a user zone to be utilized as an Extended Key Memory.

Keys stored in the User Memory cannot be utilized directly by the cryptographic commands; the keys must be transferred into either the VolatileKey Register or into a Key Register in the Key Memory EEPROM prior to use. The usage restrictions for keys transferred into the VolatileKey Register are transferred from the Key Data Structure when the KeyTransfer command is executed. Usage restrictions for keys transferred into the Key Memory are stored in the KeyConfig[TargetKeyID] Register; the KeyTransfer command does not alter the KeyConfig[TargetKeyID] Register.

- If KeyConfig[TargetKeyID].TransferOK is 0b, then the Key Register cannot be updated with the KeyTransfer command.
- If KeyConfig[TargetKeyID].TransferOK is 1b, then the KeyTransfer command can be used to update the Key register; the KeyConfig[TargetKeyID].LinkPointer contains the user zone number of the extended Key Memory.
- If ZoneConfig[UZ].AuthRead is 1b for the user zone number containing the Key Data Structure, then prior authentication is required using the KeyID stored in ZoneConfig[UZ].AuthID before a key can be transferred to either the VolatileKey Register or into a Key Register in the Key Memory EEPROM.

7.16.1 Extended Key Memory Data Structure

When a user zone is utilized as the Extended Key Memory, the keys are stored in the 32-byte Key Data Structure, as shown in Table 7-35. The first 16 bytes contain the key value, two bytes store the VolUsage restrictions, and the remaining bytes should contain all zeros. The starting address of each Key Data Structure is required to be the first byte of a 32-byte physical page (see Appendix B.2, EEPROM Page Boundary). If the VolUsage in the User Zone is zero, then the key must be loaded into a key slot. If the VolUsage in the User Zone is non-zero, then the key must be loaded into VolatileKey. In the latter case, if the intended VolUsage is zero, then set one of the Reserved bits to a one. This prevents usage restrictions from being subverted by loading a key intended for VolatileKey into an EEPROM slot and visa versa.

Table 7-35. Key Data Structure in User Memory

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h	8_h	9_h	A_h	B_h	C_h	D_h	E_h	F_h
$XX00_h$ – $XX0F_h$	Key															
$XX10_h$ – $XX1F_h$	VolUsage	Reserved (All bytes 0x00)														

Table 7-36. Input Parameters

	Name	Size (bytes)	Notes
Opcode	KeyTransfer	1	0x1A
Mode	Mode	1	Must be 0x00.
Param1	TargetKeyID	2	Upper byte is always 0x00. Lower byte is the location where the key will be stored. Legal values: 0x00 to 0x0F (standard keys), 0xFF (volatile key).
Param2	Address	2	Starting address of the key data structure in User Memory.

Table 7-37. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.17 Legacy Command

The Legacy command executes a single block of the AES engine in the Electronic Code Book mode, with no input or output formatting. This is known as AES-ECB mode, and can be used to perform primitive AES encryption or decryption operations. This command does not use the Nonce Register value in the computation since the entire 16-byte AES input value comes from the input packet.

This command can be executed only if it is enabled for the device by setting ChipConfig.LegacyE to 1b and for the key by setting KeyConfig[LKeyID].LegacyOK is 1b.

Atmel recommends that any key with KeyConfig[LKeyID].LegacyOK = 1b should never be used with any other command; the Legacy command can be used to exhaustively attack the key. If the KeyConfig[LKeyID].AuthKey bit is 1b, then prior authentication is required using the KeyID stored in KeyConfig[LKeyID].LinkPointer.

Key usage limits are enforced if KeyConfig[LKeyID].CounterLimit is 1b (see Section 4.2, Key Configuration). See Appendix E.2.16, ChipConfig Register for the ChipConfig Register definition.

Table 7-38. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Legacy	1	0x0F
Mode	Mode	1	Must be 0x00.
Param1	LKeyID	2	Upper byte is always 0x00. Lower byte is the KeyID for the AES key.
Param2	Zero	2	Always 0x0000.
Data	InData	16	Input to the AES block (plaintext).

Table 7-39. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutData	16	The output of the AES block (ciphertext).

The command and response packet is transmitted as a block beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.18 Lock Command

The Lock command permanently locks various segments of the EEPROM, including the Configuration Memory, the Key Memory, and the SmallZone register. Key, Counter, and User Memory access restrictions are locked when the Configuration Memory is locked. SmallZone is locked independently of the other Configuration Memory registers.

The Atmel recommendation is the Key Memory be locked immediately after loading the keys.

The Configuration Memory must be locked before locking the Key Memory. Trying to lock the Key Memory before the Configuration Memory is locked will result in the Lock command failing.

Three registers in the Configuration Memory control the Lock/Unlock status of the memory segments:

1. The Configuration Memory is controlled by the LockConfig Register (see Appendix E.2.11, LockConfig Register).
2. The Key Memory is controlled by the LockKeys Register (see Appendix E.2.9, LockKeys Register).
3. The SmallZone Register is controlled by the LockSmall Register (see Appendix E.2.10, LockSmall Register).

If a Lock Control Register contains 0x55, then the memory segment is unlocked. The Lock command writes 0x00 to the specified lock register to lock the segment. The Lock Control Registers can be written only with the Lock command, but they can always be read with the BlockRead command. (See Section 7.4, BlockRead Command).

The Lock command Param2 is an optional checksum (CRC-16) generated over the memory segment being locked. The value in the Checksum field must match the CRC-16 calculated within the device for the lock operation to succeed. If the Lock command returns a LockError ReturnCode, then the Host system should rewrite the memory segment and try the lock operation again.

7.18.1 User Zone ReadOnly Activation

After the Configuration Memory is locked, the Lock command can be used to activate the ReadOnly user zone feature on appropriately configured user zones. The Lock command changes the user zone from Read/Write to read-only if the following requirements are satisfied:

- ZoneConfig[Zone].WriteMode must be 10b or 11b.
- Lock command Mode bits 0 and 1 must be set to 11b.
- The Lock command zone field contains the target user zone number (Zone).

If Lock command Mode bit 2 is 1b, then the Checksum field contains the CRC-16 of the user zone contents.

If ZoneConfig[Zone].WriteMode is 11b, then the command must include an InMAC generated using the KeyID stored in ZoneConfig[Zone].WriteID; otherwise, the MAC is ignored.

The Lock command changes the ZoneConfig[Zone].ReadOnly byte from 0x55 (Read/Write) to 0x00 when the ReadOnly feature is activated. It is not possible to change a read-only user zone to read/write after Configuration Memory is locked.

Table 7-40. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Lock	1	0x0D
Mode	Mode	1	Bit 0-1: 00b = Lock the SmallZone Register. 01b = Lock the Key Memory. 10b = Lock the Configuration Memory, excluding SmallZone. 11b = Set the ZoneConfig[Zone].ReadOnly byte to ReadOnly. Bit 2: 1b = Validate the memory checksum in Param2. 0b = Suppress the Checksum validation (not recommended by Atmel). Bits 3-4: Reserved. Must be 0x00. Bit 5: 1b = Include the Usage Counter associated with the ZoneConfig[Zone].WriteID key in the MAC (ignored unless Mode[0:1] is 11b). Bit 6: 1b = Include SerialNum in the MAC (ignored unless Mode[0:1] is 11b). Bit 7: 1 = Include the first four bytes of SmallZone in the MAC (ignored unless Mode[0:1] is 11b).
Param1	Zone	2	Upper byte is always 0x00. If Mode[0:1] is 11b, the lower byte is the user zone to be locked (see Section 7.18.1, User Zone ReadOnly Activation). For any other values of Mode[0:1], this field must be 0x0000.
Param2	Checksum	2	If Mode bit 2 is 1b, contains the CRC-16 checksum generated over the memory segment being locked. If Mode bit 2 is 0b, this parameter must be 0x0000.
Data	InMAC	0 or 16	If Mode[0:1] is 11b, contains the MAC authorizing update of ZoneConfig[Zone].ReadOnly, as described in Section 7.18.1. For all other modes, this field is ignored.

Table 7-41. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.19 Nonce Command

The Nonce command generates and/or stores a 96-bit Nonce in the SRAM Nonce Register for use by subsequent cryptographic commands. It is not necessary to generate a new Nonce before each cryptographic operation because the ATAES132A includes the MacCount in the MAC calculations (see Appendix I.1, MacCount) to guarantee uniqueness.

There are two Nonce command options:

- **Inbound Nonce**

  The InSeed value is written directly to the Nonce Register. No random number generation or cryptographic Nonce calculation is performed.

  Note: This option provides no defense against replay attacks or known plaintext attacks.

- **Random Nonce**

  The InSeed value is cryptographically combined with the new output of the RNG and stored in the Nonce Register. The random number used for the Nonce calculation is returned to the Host in the response. See Appendix I.28, Nonce Command for the Nonce algorithm.

If the LockConfig Register is unlocked (0x55), then the RNG is latched in the Test mode, and executing the Nonce command with Mode bit 0 = 1b will generate nonrandom values. If the LockConfig Register is locked (0x00), then the RNG generates random numbers and the Nonce command functions normally.

The Nonce remains valid until one of the following events occurs:

- A MAC compare operation fails.
- MacCount reaches the maximum count (see Appendix I.1, MacCount).
- The cryptographic state machine is reset due to either receipt of a Reset command, power cycling (POR), or activation of the initialization sequence due to Wake-up from the Sleep power state (see Appendix G.2.2, Wake-Up from Sleep).
- Execution of the Nonce command resets MacCount to zero (see Appendix I.1, MacCount).

If a cryptographic operation involves two ATAES132A devices and a synchronized Nonce is required, then the Nonce synchronization procedure in Section 7.20.1, Nonce Synchronization must be used. The Nonce command cannot be used to generate a synchronized random Nonce.

There is one RNG Seed Register in the EEPROM memory, which is used by the KeyCreate, Nonce, and Random Commands. The RNG Seed Register is subject to the same Write endurance limitations as the other bytes in the EEPROM (see Section 9.2, Reliability for the EEPROM specifications). The application developer must not exceed the Write endurance limit.

Table 7-42. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Nonce	1	0x01
Mode	Mode	1	Bit 0: 1b = Generate a random Nonce using the RNG. 0b = Use the InSeed as the Nonce (Inbound Nonce mode), Mode bit 1 is ignored. Bit 1: 0b = Update the EEPROM RNG seed prior to Nonce generation.(1) 1b = Generate a random Nonce using the existing RNG Seed. Bits 2-7: Reserved. Must be 0b.
Param1	Zero	2	Always 0x0000.
Param2	Zero	2	Always 0x0000.
Data	InSeed	12	Input seed (required).

Note: 1. The RNG Seed Register in the EEPROM will be updated automatically if Mode bit 1 = 0b, unless the Seed Register was previously updated after the most recent Power-On Reset, Wake from the Sleep state, Reset command, or Tamper Event. Updating the RNG Seed Register increases the randomness of the Nonce; however, the EEPROM Write endurance specification must be respected.

Table 7-43. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution failure or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
Random	0 or 16	In Random Nonce mode, the random number used to generate the Nonce is returned. In Inbound Nonce mode, no data is returned.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.20 NonceCompute Command

The NonceCompute command generates the Nonce in a manner that allows two ATAES132A devices to have identical random Nonces based on random numbers generated by both devices. The identical Nonce values and identical MacCount values are required to encrypt data on one device for decryption by the other device.

The Random Command must be executed with Mode bit 2 = 1b prior to execution of the NonceCompute command. The Random Command generates a random number, which the NonceCompute command combines with the RandomSeed provided by the second ATAES132A to generate the random Nonce.

The Nonce remains valid until one of the following events occurs:

- A MAC compare operation fails.
- MacCount reaches the maximum count (see Appendix I.1, MacCount).
- Cryptographic state machine is reset due to:
  - Receipt of a Reset command,
  - Power Cycling (POR), or
  - Activation of the initialization sequence due to Wake-Up from the Sleep power state (see Appendix G.2.2, Wake-Up from Sleep).

This command resets MacCount to zero only if the operation succeeds (see Appendix I.1). If an error occurs, the contents of the Nonce Register and the MacCount Register remained unchanged. The NonceValid flag also remains unchanged.

7.20.1 Nonce Synchronization

The following procedure synchronizes the Nonce and the MacCount Register on two ATAES132A devices. In this procedure, the device where the procedure begins is referred to as “A”, and the device it is synchronized with is referred to as “B”.

1. The Random Command is executed on Device A with Mode bit 2 set to 1b. The first 12 bytes of the random field value in the response are stored for use in Step 2.
2. The Nonce command is executed on Device B with Mode bit 1 set to 1b. The 12-byte random number generated in Step 1 is used as the Nonce command InSeed field value. The 12-byte random field value in the response is stored for use in Step 3.
3. The NonceCompute command is executed on Device A using the 12-byte random number generated in Step 2 as the RandomSeed field value.
4. Successful execution of this procedure sets the Nonce status flags on both devices to:
   - NonceValid = YesNonce
   - NonceRandom = Random
   - NonceCompute = No
   - MacCount is zero on both devices.

Table 7-44. Input Parameters

	Name	Size (bytes)	Notes
Opcode	NonceCompute	1	0x13
Mode	Mode	1	The value of this field must match the Mode field value used when executing the Nonce command on the origin device.
Param1	Zero	2	Always 0x0000.
Param2	Zero	2	Always 0x0000.
Data	RandomSeed	12	First 12 bytes output by the Nonce command on the origin device.

Table 7-45. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution failure or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.21 Random Command

The Random command generates a random number using the internal high-quality RNG and the random number generation procedure recommended by NIST in SP800-90 (see Appendix A, Standards and Reference Documents). The Random Command returns the generated random number to the Host.

There are two Random command options:

- **Random Number Generation:**

  If Mode bit 2 = 0b, the 16-byte random number is returned only to the Host; it is not stored internally. This option does not affect the cryptographic state of the device.

- **Nonce Synchronization:**

  If Mode bit 2 = 1b, then the first 12 bytes of the random number are stored in the Nonce Register for later use by the NonceCompute command. The 16-byte random number is returned to the Host. The Nonce status flags are changed to:

  – NonceValid = YesNonce
  – NonceRandom = Fixed
  – NonceCompute = Yes (See Section 7.20, NonceCompute Command for the NonceCompute command and the Nonce synchronization procedure.)

If the LockConfig Register is unlocked (0x55), then the RNG is latched in the test mode, and the Random Command will always return 16 bytes of 0xA5. If the LockConfig register is locked (0x00), then the RNG generates random numbers.

There is one RNG Seed Register in the EEPROM memory, which is used by the KeyCreate, Nonce, and Random Commands. The RNG Seed Register is subject to the same Write endurance limitations as the other bytes in the EEPROM (see Section 9.2, Reliability for the EEPROM specifications). The application developer must not exceed the write endurance limit.

Table 7-46. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Random	1	0x02
Mode	Mode	1	Bit 0: Reserved. Must be 0b. Bit 1: 0b = Update the EEPROM RNG Seed Register prior to random number generation(1) 1b = Generate random number using the existing RNG Seed. Bit 2: 0b = Then return the random number. Do not change the Nonce. 1b = Then store the first 12 bytes of the random number in the Nonce Register, and return the 16-byte random number. Bits 3 to 7: Reserved. Must be 0b.
Param1	Zero	2	Always 0x0000.
Param2	Zero	2	Always 0x0000.
Data	-	0	

Note: 1. The RNG Seed Register in the EEPROM will be updated automatically if Mode bit 1 = 0b, unless the Seed Register was previously updated after the most recent Power On Reset, Wake from the Sleep state, Reset command, or Tamper Event. Updating the RNG Seed Register increases the randomness of the Random Command output; however, the EEPROM Write endurance specification must be respected.

Table 7-47. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution failure or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
Random	16	The random number.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.22 Reset Command

The Reset command forces ATAES132A to reset the logic, including the AES engine, Nonce, and Authentication status flag. This command does not return a response.

When a Reset command is received, ATAES132A performs the same power-up reset sequence that occurs during Wake from the Sleep state. The reset is complete after the WakeUp Ready time, $t_{WupSL.RDY}$ (see Section 9.4.1, Power-Up, Sleep, Standby, and Wake-Up Timing).

7.22.1 SPI Reset

During the reset of an ATAES132A configured for SPI interface mode, the device will answer the SPI Read Status Register command with 0xFF to indicate it is busy. When reset is complete, the WIP Status bit changes to 0b to indicate the device is in the Active state. The ATAES132A will only accept the SPI Read Status Register command while it is resetting; all other commands will be ignored. The SPI Read Status Register command is described in Appendix K.3.5, Read Status Register Command (RDSR).

7.22.2 I2C Reset

During the reset of an ATAES132A configured for I^2C interface mode, the Host is required to perform ACK polling using the matching I^2C Device Address. The ATAES132A will answer the ACK poll with an I2C NAK to indicate the device is busy during reset. The ACK poll reply will change to ACK when the device is in the Active state. ATAES132A will not accept any I^2C commands while it is busy. ACK polling is described in Appendix J.3.7, Acknowledge Polling.

Table 7-48. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Reset	1	0x00
Mode	Mode	1	This byte can be any value.
Param1	Zero	2	Always 0x0000.
Param2	Zero	2	Always 0x0000.
Data	-	0	

Table 7-49. Output Parameters

Name	Size (bytes)	Notes
		No response packet is returned by the Reset command.

The command packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.23 Sleep Command

The Sleep command forces the ATAES132A into one of two Low-Power states; Sleep or Standby. This command does not return a response.

The Sleep state can be used to extend battery life in portable systems by powering down the ATAES132A internal circuitry when the device is sleeping. The Standby state puts the internal circuitry in a low-power state to reduce power consumption while preserving the volatile memory contents and the security state.

A device in the Sleep state will not retain any volatile memory contents or security states. A device in the Sleep state goes through a full power-up sequence upon Wake-Up.

A device in the Standby state will retain all volatile memory contents. A device in the Standby state does not go through a power-up sequence upon Wake-Up.

The ATAES132A exits the Sleep or Standby state if a Wake-Up event occurs on the I/O pins. Wakeup is discussed in Appendix L.2, Power State Transitions.

See Appendix L, Power Management for a detailed description of the ATAES132A sleep, standby, wake-up, and power management functions.

Table 7-50. Input Parameters

	Name	Size (bytes)	Notes
Opcode	Sleep	1	0x11
Mode	Mode	1	Bit 0 to 5: Reserved. Must be 0b. Bit 6: 0b = Activate the Sleep state. 1b = Activate the Standby state. Bits7: Reserved. Must be 0b.
Param1	Zero	2	Always 0x0000.
Param2	Zero	2	Always 0x0000.
Data	-	0	

Table 7-51. Output Parameters

Name	Size (bytes)	Notes
		No response packet is returned by the Reset command.

The command packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

7.24 WriteCompute Command

The WriteCompute command encrypts data and computes the MAC required to execute the EncWrite command on a second ATAES132A device.

To use this command, the Nonce must be identical on both devices (see Section 7.20.1, Nonce Synchronization) and MacCount must have the same value on each device. Both devices must also contain identical key values, but it is not necessary for the KeyID on the origin device to match the KeyID on the Destination device. In this section, the device that encrypts data and generates the MAC is referred to as the Origin device. The device that checks the MAC is referred to as the Destination device.

If Mode bit 5, 6, or 7 is 1b, then the associated Usage Counter, SerialNum Register value, or the first four bytes of the SmallZone Register must be identical on both devices. The ManufacturingID Register must be identical on both devices, since it is always included in the MAC calculation.

A valid Nonce is required to run the WriteCompute command. If the KeyConfig[EKeyID].RandomNonce bit is 1b, then the Nonce must be random.

The value of Param2 in the FirstBlock field must match the Count field value. This command can be executed only if it is enabled for the device by setting ChipConfig.DecReadE to 1b.

Table 7-52. Input Parameters

	Name	Size (bytes)	Notes
Opcode	WriteCompute	1	0x16
Mode	Mode	1	Always 0x0000.
Param1	EKeyID	2	Upper byte is always 0x00. Lower byte is the pointer to the encrypt key. Legal values: 0x00 to 0x0F, 0xFF.
Param2	Count	2	Upper byte is always 0x00. Lower byte is the number of Data bytes to be encrypted.
Data1	FirstBlock	6	The value of this field must match the first authenticate-only block to be used when executing the EncWrite command on the Destination device.
Data2	SecondBlock	16	The value of this field must match the second authenticate-only block to be used when executing the EncWrite command on the Destination device. If Mode bits 5, 6, and 7 are 0b, then this field must be present, but is ignored.
Data3	InData	1 to 32	Input data to be encrypted (plaintext).

Table 7-53. Output Parameters

Name	Size (bytes)	Notes
ReturnCode	1	Upon success, 0x00 will be returned. Any command execution or validation failure generates a nonzero error code, per Section 6.3, ReturnCode.
OutMac	16	The input MAC for the EncWrite command on the destination device.
OutData	16 or 32	The encrypted data (ciphertext) to be written to the destination device using the EncWrite command.

The command and response packet is transmitted as a block, beginning with the Count and ending with a packet Checksum. This block format is described in Section 6.1, Command Block and Packet.

8. Pin Lists

8.1 Package Pin List (SOIC and UDFN)

Table 8-1. Package Pin List

Pin	Name	Description	Type
1	$\overline{CS}$	SPI Mode = $\overline{CS}$ I^2C Mode = *Not* used	Input
2	SO	SPI Mode = Serial Data Out I^2C Mode = *Not* used or AuthO Out	Output
3	NC	No Connect	NC
4	V_{SS}	Ground	Ground
5	SI/SDA	SPI Mode = Serial Data In I^2C Mode = Serial Data I/O	Input/Output
6	SCK	Serial Data Clock	Input
7	NC	No Connect	NC
8	V_{CC}	Power Supply	Power

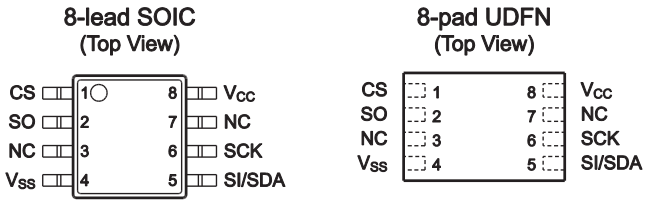
8.2 Pin Descriptions

Table 8-2. Pin Descriptions

Pin	Name	Description
1	$\overline{CS}$	**SPI Chip Select Bar Input.** In the SPI communication mode, this pin functions as the slave select input. In the I^2C communication mode, this pin is *not* used, and should be tied to V_{CC} or V_{SS}.
2	SO	**Serial Data Out.** In the SPI communication mode, this pin functions as the serial data output. In the I^2C communication mode, this pin is *not* used in the default configuration. It is always in the high-impedance state. If Auth signaling is enabled, then this pin functions as the AuthO output (see Appendix J.5, I2C Auth Signaling).
3	NC	**No Connect.** This package pin is not used, and can be left open by the user.
4	V_{SS}	**Ground.**
5	SI/SDA	**Serial Data In.** In SPI communication mode, this pin functions as the serial data input. In I^2C communication mode, this pin functions as the serial data I/O.
6	SCK	**Serial Clock Input.** In both SPI and I^2C serial communication modes, this pin is used as the serial interface clock.
7	NC	**No Connect.** This package pin is not used, and can be left open by the user.
8	V_{CC}	**Supply Voltage.** To insure a stable V_{CC} level, it is recommended that V_{CC} be decoupled with a high quality capacitor, in the order of 0.01µF, positioned close to the V_{CC} and V_{SS} pins of the ATAES132A.

8.3 Pin Configurations

Figure 8-1. Pin Configurations



Note: Drawings are not to scale.

9. Electrical Characteristics

9.1 Absolute Maximum Ratings*

Operating Temperature	-40°C to +85°C
Storage Temperature	-65°C to +150°C
Maximum Operating Voltage	6.0V
DC Output Current	5.0mA
Voltage on any pin	-0.7V to (V_{CC} + 0.7V)
HBM ESD	3kV minimum

Notice*: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only, and the functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.2 Reliability

The ATAES132A is fabricated with the Atmel high reliability CMOS EEPROM manufacturing technology. The reliability ratings in Table 9-1 apply to each byte of the EEPROM memory.

Table 9-1. EEPROM Reliability(1)

Parameter	Min	Typical	Max	Units
Write Endurance (each byte)	100,000			Write Cycles
Data Retention (at 55°C)	10			Years
Data Retention (at 35°C)	30	50		Years
Read Endurance	Unlimited			Read Cycles

Note: 1. These specifications apply to every byte of the User Memory, Configuration Memory, and Key Memory. The Write Endurance specification also applies to the RNG EEPROM Seed Register.

9.3 DC Characteristics

9.3.1 Supply Characteristics

Table 9-2. Supply Voltage and Current Characteristics

Applicable over recommended operating range from T_A = -40°C to +85°C, V_{CC} = +2.5V to +5.5V (unless otherwise noted). (1)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{CC} (1)	Supply Voltage		2.50		5.50	V
I_{CC1}	Supply Current	V_{CC} = 3.3V at f_{max} (4) SO = Open(3), Read, Write, or AES operation.			6	mA
I_{CC2}	Supply Current	V_{CC} = 5.5V at f_{max} (4) SO = Open(3), Read, Write, or AES operation.			10	mA
I_{CC3}	Idle Current	V_{CC} = 3.3V or 5.5V at f_{max} (4) SO = Open(3), Waiting for a command.		600	800	µA
I_{SL1}	Sleep Current	V_{CC} = 3.3V $\overline{CS}$ = V_{CC}(3), Sleep State(5)		0.10	0.25	µA
I_{SL2}	Sleep Current	V_{CC} = 5.5V $\overline{CS}$ = V_{CC}(3), Sleep State(5)		0.25	0.50	µA
I_{SB1}	Standby Current	V_{CC} = 3.3V $\overline{CS}$ = V_{CC}(3), Standby State(5)		15	30	µA
I_{SB2}	Standby Current	V_{CC} = 5.5V $\overline{CS}$ = V_{CC}(3), Standby State(5)		20	40	µA

Notes:
1. Typical values are at 25˚C, and are for reference only. Typical values are not tested or guaranteed.
2. On power-up, V_{CC} must rise continuously from V_{SS} to the operating voltage, with a rise time no faster than 1V/µs.
3. All input pins must be held at either Vss or Vcc during this measurement. In SPI interface mode, the $\overline{CS}$ pin must be at V_{CC}. In I^2C interface mode, the $\overline{CS}$ pin may be in either state.
4. Measurement is performed at the maximum serial clock frequency. In the I^2C interface mode, f_{max} is 1MHz. In the SPI interface mode, f_{max} is 10MHz.
5. See Appendix L, Power Management for Sleep and Standby state information. The `Sleep` command is described in Section 7.23,Sleep Command.
6. The ATAES132A does not support hot swapping or hot plugging. Connecting or disconnecting this device to a system while power is energized can cause permanent damage to the ATAES132A.

9.3.2 I/O Characteristics

Table 9-3. DC Characteristics

Applicable over recommended operating range from T_A = -40°C to +85°C, V_{CC} = +2.5V to +5.5V (unless otherwise noted).

Symbol	Parameter	Test conditions	Min	Max	Units
I_{LI}	Input Current	V_{IN} = 0V or V_{CC}	-3.0	3.0	µA
I_{LO}	Output Leakage	V_{OUT} = 0V or V_{CC}	-3.0	3.0	µA
V_{IL} (1)	Input Low-Voltage		-0.5	V_{CC} x 0.3	V
V_{IH} (1)	Input High-Voltage		V_{CC} x 0.7	V_{CC} + 0.5	V
V_{OL1} (2)	Output Low-Voltage, Except SI/SDA in I^2C Mode	I_{OL} = 3.0mA	0	0.4	V
V_{OH1} (2)	Output High-voltage, Except SI/SDA in I^2C Mode	I_{OH} = -3.0mA	V_{CC} − 0.8	V_{CC}	V
V_{OL2}	Output Low-voltage, SI/SDA Pin in the I^2C Mode *Only*	I_{OL} = 3.0mA	0	0.4	V

Notes: 1. V_{IL} min and V_{IH} max are for reference only, and are not tested.

2. In the I^2C interface mode, if Auth signaling is enabled, the SO pin functions as the AuthO output (see Appendix J.5, I^2C Interface). When AuthO is high, the V_{OH1} specification applies. When AuthO is not high, the pin is in the high-impedance state; the V_{OL1} specification is not applicable.

9.4 AC Characteristics

Table 9-4. AC Characteristics

Applicable over recommended operating range from T_A = -40°C to + 85°C, V_{CC} = +2.5V to +5.5V.

Symbol	Parameter	Min	Max	Units
t_{WC1}	User Zone Write Cycle Time(1)	6.0	9.0	ms
t_{WC2}	Key Zone Write Cycle Time(1)	12.0	16.0	ms
	Command Response Time	See Appendix N.		

Note: 1. The write cycle time includes the EEPROM Erase, Write, and Automatic Data Write verification operations.

9.4.1 Power-Up, Sleep, Standby, and Wake-Up Timing

Table 9-5. Power-Up, Sleep, and Wake-Up Timing Characteristics[1][2]

Applicable over recommended operating range from T_A = -40°C to + 85°C, V_{CC} = +2.5V to +5.5V.

Symbol	Parameter	Min	Typ	Max	Units
$t_{PU.STATUS}$	Power-Up Time, Status		500	600	µs
$t_{PU.RDY}$	Power-Up Ready Time		1200	1500	µs
t_{SB}	Sleep Time, Entering the Standby State		65	100	µs
t_{SL}	Sleep Time, Entering the Sleep State		55	90	µs
$t_{WupSB.STATUS}$	Wake-Up Status Time, Standby State		50	100	µs
$t_{WupSB.RDY}$	Wake-Up Ready Time, Standby State		200	240	µs
$t_{WupSL.STATUS}$	Wake-Up Status, Sleep State		500	600	µs
$t_{WupSL.RDY}$	Wake-Up Ready Time, Sleep State		1200	1500	µs

Notes:
1. All values are based on characterization and are not tested. Typical values are at 25˚C and are for reference only.
2. See Appendix L, Power Management for Power-Up, Sleep, Standby, and Wake-Up specifications. The Sleep command is described in Section 7.23, `Sleep` Command.

9.4.2 I²C Interface Timing

Table 9-6. AC Characteristics of I²C Interface

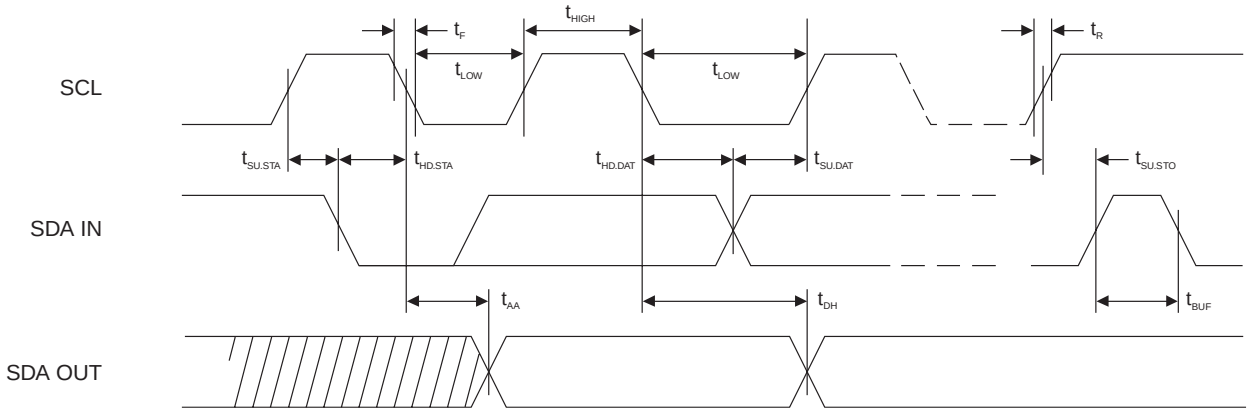
Applicable over recommended operating range from T_A = -40°C to + 85°C, V_{CC} = +2.5V to +5.5V, CL = 1 TTL Gate and 100pF (unless otherwise noted).

Symbol	Parameter	Min	Max	Units
f_{SCK}	SCK Clock Frequency		1	MHz
	SCK Clock Duty Cycle	30	70	percent
t_{HIGH}	SCK High Time	400		ns
t_{LOW}	SCK Low Time	400		ns
$t_{SU.STA}$	Start Setup Time	250		ns
$t_{HD.STA}$	Start Hold Time	250		ns
$t_{SU.STO}$	Stop Setup Time	250		ns
$t_{SU.DAT}$	Data in Setup Time	100		ns
$t_{HD.DAT}$	Data in Hold Time	0		ns
t_R	Input Rise Time(1)		300	ns
t_F	Input Fall Time(1)		100	ns
t_{AA}	Clock Low to Data Out Valid	50	550	ns
t_{DH}	Data Out Hold Time	50		ns
t_{BUF}	Time bus must be free before a new transmission can start.(1)	500		ns

Notes: 1. Values are based on characterization, and are not tested.

2. AC measurement conditions:
   - RL (connects between SDA and V_{CC}): 2.0kΩ (for V_{CC} +2.5V to +5.0V)
   - Input pulse voltages: 0.3V_{CC} to 0.7V_{CC}
   - Input rise and fall times: ≤ 50ns
   - Input and output timing reference voltage: 0.5V_{CC}

Figure 1-1. I²C Synchronous Data Timing



9.4.3 SPI Interface Timing

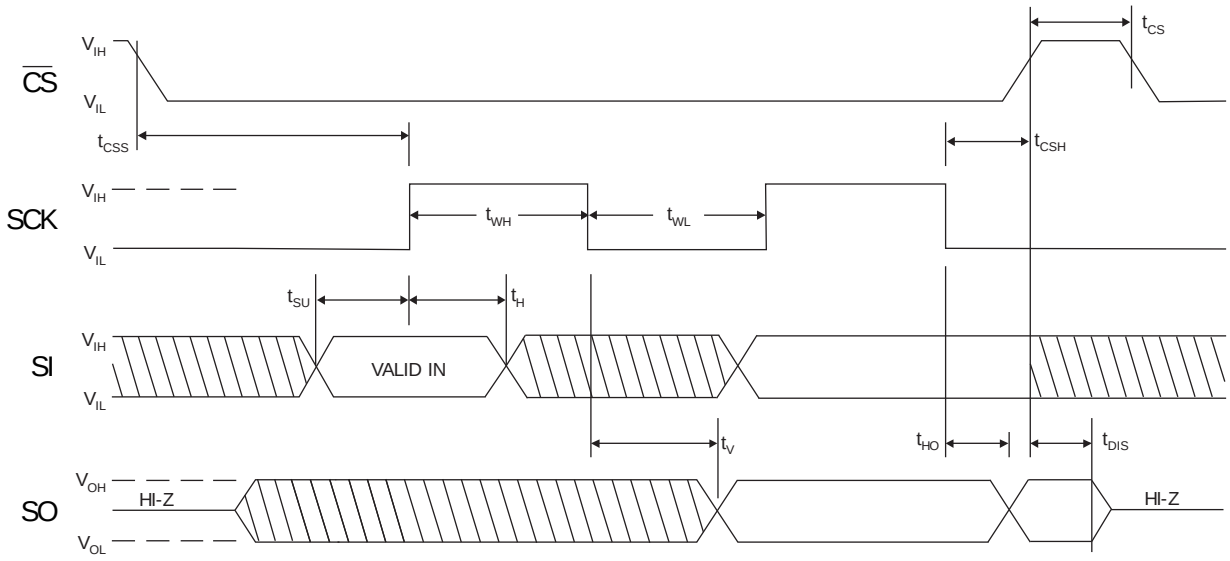
Table 9-7. AC Characteristics of SPI Interface

Applicable over recommended operating range from T_A = -40°C to + 85°C, V_{CC} = +2.5V to +5.5V, CL = 1 TTL Gate and 30pF (unless otherwise noted).

Symbol	Parameter	Min	Max	Units
f_{SCK}	SCK Clock Frequency	0	10	MHz
	SCK Clock Duty Cycle	30	70	percent
t_{WH}	SCK High Time	40		ns
t_{WL}	SCK Low Time	40		ns
t_{CS}	$\overline{CS}$ High Time	50		ns
t_{CSS}	$\overline{CS}$ Setup Time	50		ns
t_{CSH}	$\overline{CS}$ Hold Time	50		ns
t_{SU}	Data In Setup Time	10		ns
t_H	Data In Hold Time	10		ns
t_{RI}	Input Rise Time(1)		2	µs
t_{FI}	Input Fall Time(1)		2	µs
t_V	Output Valid	0	40	ns
t_{HO}	Output Hold Time	0		ns
t_{DIS}	Output Disable Time		50	ns

Note: 1. Values are based on characterization, and are not tested.

Figure 1-2. SPI Synchronous Data Timing



Appendix A. Standards and Reference Documents

A.1 National and International Standards

The ATAES132A is designed to comply with the requirements of the AES Standard.

FIPS-197 *Specification for the Advanced Encryption Standard (AES)*. 26 November 2001.
Available at: http://csrc.nist.gov/groups/ST/toolkit/block_ciphers.html.

A.2 References

SP800-38A *NIST Special Publication 800-38A. Recommendation for Block Cipher Modes of Operation: Methods and Techniques.* December 2001.
Available at: http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html.

SP800-38C *NIST Special Publication 800-38C. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality.* May 2004.
Available at: http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html.

SP800-90 *NIST Special Publication 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators.* (Revised) March 2007.
Available at: http://csrc.nist.gov/groups/ST/toolkit/random_number.html.

JEP106xx *JEDEC Standard. Standard Manufacturer's Identification Code.* JEDEC Solid State Technology Association. Updated periodically. JEP106AA April 2009.
Available at http://www.jedec.org.

ISO/IEC7816-1:1998 *Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 1: Physical Characteristics.* October 1998.
Available at: http://www.iso.org or http://www.ansi.org or from National Standards Body.

ISO/IEC7816-2:2007 *Identification Cards – Integrated Circuit(s) Cards with Contacts – Part 2: Dimension and Location of the Contacts.* October 2007.
Available at: http://www.iso.org or http://www.ansi.org or from National Standards Body.

Appendix B. Memory Map

B.1 Memory Map

Reserved memory cannot be written or read.

Table B-1. ATAES132A Memory Map

Byte Address	Description
0000_h-$0FFF_h$	User Memory (See Appendix C, User Memory Map)
1000_h-$EFFF_h$	Reserved
$F000_h$-$F05F_h$	Configuration Memory – Device Config (See Appendix E, Configuration Memory Map)
$F060_h$-$F07F_h$	Configuration Memory – CounterConfig (See Appendix E)
$F080_h$-$F0BF_h$	Configuration Memory – KeyConfig (See Appendix E)
$F0C0_h$-$F0FF_h$	Configuration Memory – ZoneConfig (See Appendix E)
$F100_h$-$F17F_h$	Configuration Memory - Counters (See Appendix E)
$F180_h$-$F1DF_h$	Configuration Memory – FreeSpace (See Appendix E)
$F1E0_h$-$F1FF_h$	Configuration Memory – SmallZone (See Appendix E)
$F200_h$-$F2FF_h$	Key Memory (See Appendix F, Key Memory Map)
$F300_h$-$FDFF_h$	Reserved
$FE00_h$	Command / Response Memory Buffer (See Appendix D, Command Memory Map)
$FE01_h$-$FFDF_h$	Reserved
$FFE0_h$	I/O Address Reset
$FFE1_h$-$FFEF_h$	Reserved
$FFF0_h$	STATUS Register
$FFF1_h$-$FFFF_h$	Reserved

B.2 EEPROM Page Boundary

The ATAES132A EEPROM has 32-byte physical pages. An EEPROM Write can never cross the boundary between two physical pages. BlockRead and EncRead operations cannot cross the boundary between two physical pages. Table B-2 illustrates the page boundary locations for the ATAES132A.

Table B-2. ATAES132A EEPROM Page Boundary Locations

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h	8_h	9_h	A_h	B_h	C_h	D_h	E_h	F_h
$XX00_h$-$XX0F_h$	32-byte EEPROM Page															
$XX10_h$-$XX1F_h$																
$XX20_h$-$XX2F_h$	32-byte EEPROM Page															
$XX30_h$-$XX3F_h$																
$XX40_h$-$XX4F_h$	32-byte EEPROM Page															
$XX50_h$-$XX5F_h$																
$XX60_h$-$XX6F_h$	32-byte EEPROM Page															
$XX70_h$-$XX7F_h$																
$XX80_h$-$XX8F_h$	32-byte EEPROM Page															
$XX90_h$-$XX9F_h$																
$XXA0_h$-$XXAF_h$	32-byte EEPROM Page															
$XXB0_h$-$XXBF_h$																
$XXC0_h$-$XXCF_h$	32-byte EEPROM Page															
$XXD0_h$-$XXDF_h$																
$XXE0_h$-$XXEF_h$	32-byte EEPROM Page															
$XXF0_h$-$XXFF_h$																

Appendix C. User Memory Map

The 32Kb User Memory consists of 16 user zones, each containing 2Kb (256 bytes) of memory. The physical page size is 32 bytes; Write operations cannot cross page boundaries.

Every Memory Zone has an independent set of access restrictions, and all bytes within a zone have the same access restrictions. The Configuration Memory (Appendix E, Configuration Memory Map) contains an access register for each Memory Zone that defines the access requirements for the User Zone.

Table C-1. User Memory Map

Byte Address	Description
0000_h-$00FF_h$	User Zone 0
0100_h-$01FF_h$	User Zone 1
0200_h-$02FF_h$	User Zone 2
0300_h-$03FF_h$	User Zone 3
0400_h-$04FF_h$	User Zone 4
0500_h-$05FF_h$	User Zone 5
0600_h-$06FF_h$	User Zone 6
0700_h-$07FF_h$	User Zone 7
0800_h-$08FF_h$	User Zone 8
0900_h-$09FF_h$	User Zone 9
$0A00_h$-$0AFF_h$	User Zone A
$0B00_h$-$0BFF_h$	User Zone B
$0C00_h$-$0CFF_h$	User Zone C
$0D00_h$-$0DFF_h$	User Zone D
$0E00_h$-$0EFF_h$	User Zone E
$0F00_h$-$0FFF_h$	User Zone F

Appendix D. Command Memory Map

The ATAES132A commands are executed by writing the command packet to the virtual memory using standard SPI or I^2C Write commands. The response packet is retrieved by reading it from the virtual memory using standard SPI or I^2C Read commands. The Command/Response Memory Buffer is 64 bytes.

The ATAES132A commands are executed by writing the command packet to virtual memory at starting address `0xFE00` using standard Write commands (see Appendix J, I^2C Interface and Appendix K, SPI Interface). The response packet is retrieved by reading from the virtual memory at starting address `0xFE00` using standard Read commands. The Device Status Register (STATUS) is located at `0xFFF0` (see Appendix G, Understanding the STATUS Register).

To reset the address pointer in the Command/Response Memory Buffer to the base address of the buffer, the Host writes one or more bytes to the IO Address Reset Register at address `0xFFE0` using the standard Write command. Any value can be written to the IO Address Reset Register to reset the buffer address pointer.

Table D-1. Command/Response Virtual Memory Map

Byte Address	Description
$FE00_h$	Command/Response Memory Buffer
$FE01_h$-$FFDF_h$	Reserved
$FFE0_h$	I/O Address Reset
$FFE1_h$-$FFEF_h$	Reserved
$FFF0_h$	STATUS Register
$FFF1_h$-$FFFF_h$	Reserved

D.1 Command Memory Buffer

The Command Memory Buffer is a write-only buffer memory that is used by writing a command block to the buffer at the base address of `0xFE00`. After the Host completes its Write operation to the buffer, the ATAES132A verifies the integrity of the block by checking the 16-bit Checksum, and then executes the requested operation. See Section 6.1, Command Block and Packet for a description of the crypto command block.

Write operations that begin at any other location within the buffer are invalid and will not be processed by the ATAES132A.

Table D-2. Command Memory Buffer Map

Base Address	Base + 1	Base + 2	Base + 3	.....	.....	.....	.....	Base + N-2	Base + N-1
Count	Opcode	Mode	Param1	Param1	Param2	.....	DataX	CRC1	CRC2

D.1.1 Using the Command Memory Buffer

The Host should write a single byte to the IO Address Reset Register before writing a new command block to the Command Memory Buffer. This resets the buffer address pointer to the base address. The Host then writes the ATAES132A command block to the buffer using one or more standard SPI or I^2C Write commands. After the entire command block is written by the Host microcontroller, the ATAES132A checks the 16-bit Checksum and executes the command. The Host should read the STATUS Register to determine if an error occurred or if the response is ready to be read.

If a Checksum error occurs, then the buffer address pointer must be reset by the Host before the command block is retransmitted. If no errors occur, then the response can be read from the Response Memory Buffer, as described in Appendix D.2.10, Using the Response Memory Buffer (see Appendix G, Understanding the STATUS Register for examples).

The Command Memory Buffer size is 64 bytes. If the Host writes more than 64 bytes to the buffer, it will cause a buffer overflow error. If the Host hardware must send more bytes to the ATAES132A than are required to transmit a command block (due to Host hardware limitations), then all bytes transmitted after the block Checksum must contain `0xFF`.

D.2 Response Memory Buffer

The Response Memory Buffer is a read-only memory buffer that is used by reading a response from the buffer at the base address of `0xFE00`. The base address of the Response Memory Buffer contains the first byte of the response packet after a Crypto command is processed. See Section 6.1, Command Block and Packet for a description of the crypto response packet.

Read operations that begin at any location above the base address are invalid and will either be NAKed (in I^2C mode) or ignored (output will tri-state in SPI mode).

Table D-3. Response Memory Buffer Map Following a Crypto Command

Base Address	Base + 1	Base + 2	Base + 3	.....	.....	.....	.....	Base + N-2	Base + N-1
Count	ReturnCode	Data1	Data2	Data3	.....	.....	DataX	CRC1	CRC2

The Response Memory Buffer is also used to report errors that occur during execution of standard I^2C or SPI Write commands. When the I^2C or SPI command execution is complete (as indicated by the STATUS Register), the Response Memory Buffer contains a block containing an error code (ReturnCode) if an error occurred; otherwise, it contains a block containing ReturnCode = `0x00`. Reading the Response Memory Buffer does not alter the contents of the Response Memory Buffer or the STATUS Register (see Appendix G). See Section 6.3, ReturnCode for the error descriptions.

Table D-4. Response Memory Buffer Map Following a Standard I^2C or SPI Write Operation

Base Address	Base + 1	Base + N-2	Base + N-1	.....	.....	.....	.....	.....	.....
Count	ReturnCode	CRC1	CRC2	FF_h	FF_h	FF_h	FF_h	FF_h	FF_h

D.2.1 Using the Response Memory Buffer

After an ATAES132A command is executed, the RRDY bit of the STATUS Register is set to `1b` to indicate that a new response is available in the Response Memory Buffer. The Host reads the response block from the buffer using one or more standard SPI or I^2C Read commands. After the entire response block is read, the Host microcontroller checks the 16-bit Checksum.

If a Checksum error occurs, then the buffer address pointer must be reset by the Host before the response block is reread. If the Host reads more bytes from the response buffer than necessary to retrieve the block, then all bytes after the block Checksum will contain `0xFF` (see Appendix G for examples). The Response Memory Buffer size is 64 bytes.

D.3 IO Address Reset Register

Writing the IO Address Reset Register (address `0xFFE0`) with any value causes the address pointers in the Command Memory Buffer and the Response Memory Buffer to be reset to the base address of the buffer. The IO Address Reset Register can be written with 1 to 32 bytes of data without generating an error; the data bytes will be ignored.

Writing the IO Address Reset Register does not alter the contents of the Response Memory Buffer or the value of the STATUS Register. Writing the IO Address Reset Register clears the Command Memory Buffer (see Appendix G, Understanding the STATUS Register for examples).

D.4 Device Status Register (STATUS)

The Device Status Register is used for handshaking between the Host microcontroller and the ATAES132A. The Host is expected to read the STATUS Register before sending a command or reading a response. See Appendix G for the definition and behavior of the STATUS Register. If the ATAES132A is configured in SPI interface mode, the STATUS Register can also be read using the SPI `RDSR` command, as described in Appendix K.3.5, Read Status Register Command (RDSR).

Reading the STATUS Register does not alter the contents of the Command Memory Buffer, the contents of the Response Memory Buffer, or the value of the STATUS Register.

Appendix E. Configuration Memory Map

The ATAES132A Configuration Memory is located from address 0xF000 to address 0xF1FF. The Configuration Memory can always be read using the `BlockRead` command (see Section 7.4, `BlockRead` Command). See Appendix E.2, Configuration Register Descriptions for descriptions of each configuration register. A memory map showing the default register values appears in Appendix O, Default Configuration.

E.1 Configuration Memory Map

Table E-1. ATAES132A Configuration Memory Map

Address	0_h / 8_h	1_h / 9_h	2_h / A_h	3_h / B_h	4_h / C_h	5_h / D_h	6_h / E_h	7_h / F_h
$F000_h$-$F007_h$	SerialNum							
$F008_h$-$F00F_h$	LotHistory							
$F010_h$-$F017_h$	JEDEC		Reserved			Algorithm		EEPageSize
$F018_h$-$F01F_h$	EncReadSize	EncWrtSize	DeviceNum	Reserved				
$F020_h$-$F027_h$	LockKeys	LockSmall	LockConfig	Reserved				
$F028_h$-$F02F_h$	Reserved			ManufacturingID		PermConfig	Reserved	
$F030_h$-$F037_h$	Reserved							
$F038_h$-$F03F_h$								
$F040_h$-$F047_h$	I^2CAddr	ChipConfig	RFU	RFU				
$F048_h$-$F04F_h$	Reserved			RFU				
$F050_h$-$F057_h$	RFU							
$F058_h$-$F05F_h$								
$F060_h$-$F067_h$	CounterConfig 00		CounterConfig 01		CounterConfig 02		CounterConfig 03	
$F068_h$-$F06F_h$	CounterConfig 04		CounterConfig 05		CounterConfig 06		CounterConfig 07	
$F070_h$-$F077_h$	CounterConfig 08		CounterConfig 09		CounterConfig 0A		CounterConfig 0B	
$F078_h$-$F07F_h$	CounterConfig 0C		CounterConfig 0D		CounterConfig 0E		CounterConfig 0F	
$F080_h$-$F087_h$	KeyConfig 00				KeyConfig 01			
$F088_h$-$F08F_h$	KeyConfig 02				KeyConfig 03			
$F090_h$-$F097_h$	KeyConfig 04				KeyConfig 05			
$F098_h$-$F09F_h$	KeyConfig 06				KeyConfig 07			
$F0A0_h$-$F0A7_h$	KeyConfig 08				KeyConfig 09			
$F0A8_h$-$F0AF_h$	KeyConfig 0A				KeyConfig 0B			
$F0B0_h$-$F0B7_h$	KeyConfig 0C				KeyConfig 0D			
$F0B8_h$-$F0BF_h$	KeyConfig 0E				KeyConfig 0F			
$F0C0_h$-$F0C7_h$	ZoneConfig 00				ZoneConfig 01			
$F0C8_h$-$F0CF_h$	ZoneConfig 02				ZoneConfig 03			
$F0D0_h$-$F0D7_h$	ZoneConfig 04				ZoneConfig 05			
$F0D8_h$-$F0DF_h$	ZoneConfig 06				ZoneConfig 07			
$F0E0_h$-$F0E7_h$	ZoneConfig 08				ZoneConfig 09			
$F0E8_h$-$F0EF_h$	ZoneConfig 0A				ZoneConfig 0B			
$F0F0_h$-$F0F7_h$	ZoneConfig 0C				ZoneConfig 0D			

Address	0_h / 8_h	1_h / 9_h	2_h / A_h	3_h / B_h	4_h / C_h	5_h / D_h	6_h / E_h	7_h / F_h
$F0F8_h$-$F0FF_h$	ZoneConfig 0E				ZoneConfig 0F			
$F100_h$-$F107_h$	Counter 00							
$F108_h$-$F10F_h$	Counter 01							
$F110_h$-$F117_h$	Counter 02							
$F118_h$-$F11F_h$	Counter 03							
$F120_h$-$F127_h$	Counter 04							
$F128_h$-$F12F_h$	Counter 05							
$F130_h$-$F137_h$	Counter 06							
$F138_h$-$F13F_h$	Counter 07							
$F140_h$-$F147_h$	Counter 08							
$F148_h$-$F14F_h$	Counter 09							
$F150_h$-$F157_h$	Counter 0A							
$F158_h$-$F15F_h$	Counter 0B							
$F160_h$-$F167_h$	Counter 0C							
$F168_h$-$F16F_h$	Counter 0D							
$F170_h$-$F177_h$	Counter 0E							
$F178_h$-$F17F_h$	Counter 0F							
$F180_h$-$F187_h$	FreeSpace							
$F188_h$-$F18F_h$								
$F190_h$-$F197_h$								
$F198_h$-$F19F_h$								
$F1A0_h$-$F1A7_h$								
$F1A8_h$-$F1AF_h$								
$F1B0_h$-$F1B7_h$								
$F1B8_h$-$F1BF_h$								
$F1C0_h$-$F1C7_h$								
$F1C8_h$-$F1CF_h$								
$F1D0_h$-$F1D7_h$								
$F1D8_h$-$F1DF_h$								
$F1E0_h$-$F1E7_h$	SmallZone							
$F1E8_h$-$F1EF_h$								
$F1F0_h$-$F1F7_h$								
$F1F8_h$-$F1FF_h$								

Notes:
1. Orange registers = Locked at the factory and cannot be changed by the customer.
2. Blue registers = Lock registers can be changed only by using the `Lock` command (see Section 7.18, `EncRead` Command).
3. Green registers = Configuration registers can be written by the customer prior to locking (by setting LockConfig to `0x00` using the `Lock` command).
4. Yellow registers = The SmallZone Register can be written by the customer prior to locking (by setting LockSmall to `0x00` using the `Lock` command). SmallZone is locked separately from the remainder of the Configuration Memory.

E.2 Configuration Register Descriptions

Each register in the Configuration Memory is briefly described in this section. References are provided to detailed information in other sections of this specification. The registers are described in the same order in which they occur in the memory map in Appendix E.1, Configuration Memory Map.

E.2.1 SerialNum Register

SerialNum is an 8-byte, read-only register that is programmed by Atmel at the factory. The contents of this register are guaranteed to be unique on each unit over the production life of the ATAES132A product family. The contents of this register can optionally be included in cryptographic calculations by setting Mode bit 6 to `1b`, as described in the command definitions in Section 7, Command Definitions. This register cannot be changed by the customer.

It is recommended that the SerialNum Register value be used to perform key diversification.

E.2.2 LotHistory Register

LotHistory is an 8-byte, read-only register that is programmed by Atmel at the factory. This register contains proprietary data that is not intended for customer use. This register cannot be changed by the customer.

E.2.3 JEDEC Register

JEDEC is a 2-byte, read-only register that is programmed by Atmel at the factory. The JEDEC register always contains `0x001F`, which is the JEDEC Manufacturing Identification Code assigned to Atmel. This register cannot be changed by the customer.

E.2.4 Algorithm Register

Algorithm is a 2-byte, read-only register that is programmed by Atmel at the factory. The default value of `0x0000` indicates 128-bit AES-CCM mode. This register cannot be changed by the customer.

E.2.5 EEPageSize Register

EEPageSize is a 1-byte, read-only register that is programmed by Atmel at the factory. The default value of `0x20` indicates a 32-byte physical EEPROM page size. This register cannot be changed by the customer.

E.2.6 EncReadSize Register

EncReadSize is a one-byte, read-only register that is programmed by Atmel at the factory. The default value of `0x20` indicates that 32 bytes is the maximum data length that can be returned by the `EncRead` command. This register cannot be changed by the customer.

E.2.7 EncWrtSize Register

EncWrtSize is a 1-byte, read-only register that is programmed by Atmel at the factory. The default value of `0x20` indicates that 32 bytes is the maximum data length that can be written using the `EncWrite` command. This register cannot be changed by the customer.

E.2.8 DeviceNum Register

DeviceNum is a 1-byte, read-only register that is programmed by Atmel at the factory. This byte indicates the device type (32Kb, ATAES1xx family). The `INFO` command returns this byte, along with a hardware revision byte. The DeviceNum will change with device revisions and should not be considered a constant. This register cannot be changed by the customer.

See Section 7.12, `INFO` Command for the `INFO` command description.

E.2.9 LockKeys Register

LockKeys is a 1-byte register that controls write access to Key Memory. The default value of LockKeys is the unlocked state (`0x55`). The LockKeys Register can be changed only by using the `Lock` command (see Section 7.18, Lock Command). After the `Lock` command is run, this register will contain `0x00`, and the Key Memory will be locked. It is impossible to unlock memory that has been locked.

E.2.10 LockSmall Register

LockSmall is a 1-byte register that controls write access to the SmallZone Register. The default value of LockSmall is the unlocked state (`0x55`). The LockSmall Register can be changed only by using the `Lock` command (see Section 7.18). After the `Lock` command is run, this register will contain `0x00`, and the SmallZone Register will be locked. It is impossible to unlock memory that has been locked.

E.2.11 LockConfig Register

LockConfig is a 1-byte register that controls write access to Configuration Memory except the SmallZone Register. The default value of LockConfig is the unlocked state (`0x55`). The LockConfig Register can be changed only by using the `Lock` command (see Section 7.18). After the `Lock` command is run, this register will contain `0x00`, and the Configuration Memory will be locked except for the SmallZone Register, which is controlled by the LockSmall Register. It is impossible to unlock memory that has been locked.

If the LockConfig register is unlocked (`0x55`), then the RNG is latched in Test mode, and the `Random` Command will always return 16 bytes of `0xA5`. The `KeyCreate` and `Nonce` commands will create nonrandom results when the RNG is in Test mode. If the LockConfig Register is locked (`0x00`), then the RNG generates random numbers, and the random `KeyCreate` and `Nonce` commands function normally.

E.2.12 Reserved Registers

Any Configuration Memory locations that are identified as reserved in Table E-1, the Configuration Memory map, are reserved by Atmel for future use. All reserved registers are read-only registers that are programmed by Atmel at the factory. These memory locations are programmed with Atmel proprietary data. The contents of the reserved registers will vary and are not intended for any customer use. These registers cannot be changed by the customer.

E.2.13 ManufacturingID Register

ManufacturingID is a 2-byte, read-only register that is programmed by Atmel at the factory. This register contains a customer-specific value. The default ManufacturingID Register contains `0x00EE`. This register cannot be changed by the customer.

E.2.14 PermConfig Register

PermConfig is a 1-byte read-only register that is programmed by Atmel at the factory. This register cannot be changed by the customer. The default value of `0x01` enables all cryptographic commands.

Table E-2. PermConfig Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved for Future Use							EncryptE

If the EncryptE bit is 1b, then the `Encrypt`, `Decrypt`, and `Legacy` command availability is determined by the ChipConfig.EncDecrE and ChipConfig.LegacyE bits. If the EncryptE bit is 0b, then the `Encrypt`, Decrypt, and `Legacy` commands are disabled. See the ChipConfig Register definition in Appendix E.2.16, ChipConfig Register for additional information.

E.2.15 I2CAddr Register

I2CAddr is a 1-byte register that controls the ATAES132A serial interface. The customer can write the I2CAddr Register using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see Appendix E.2.11, LockConfig Register).

Table E-3. I^2CAddr Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I^2C Device Address							SPI/I^2C

Bit 0 selects the serial interface mode, `0b` selects SPI interface mode, and 1b selects I^2C interface mode. If bit 0 is `0b`, then the contents of bits one to seven are ignored.

The default value of the I2CAddr Register depends on the ordering code (see Appendix Q, Ordering Information): I2CAddr is `0xA1` (the I2C Device Address is `0xA0`) for catalog numbers with an I^2C interface configuration, and I2CAddr is `0x00` for catalog numbers with a SPI interface configuration. See Appendix J, I^2C Interface for the I^2C interface specifications. See Appendix K, SPI Interface for the SPI interface specifications.

E.2.16 ChipConfig Register

ChipConfig is a 1-byte register that controls device-level functionality of the ATAES132A. The customer can write the ChipConfig register using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig Register definition in Section E.2.11).

Table E-4. ChipConfig Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PowerUpState		Reserved for Future Use		AuthComputeE	DecReadE	EncDecrE	LegacyE

If the ChipConfig.LegacyE bit is `1b`, then the `Legacy` command (Section 7.17, `Legacy` Command) is enabled. If ChipConfig.LegacyE is `0b`, then a parse error ReturnCode will be returned in response to a `Legacy` command. If the ChipConfig.EncDecrE bit is `1b`, then the `Encrypt` command (Section 7.10, `Encrypt` Command) and `Decrypt` command (Section 7.8, Decrypt Command) are enabled. If ChipConfig.EncDecrE is `0b`, then a parse error ReturnCode will be returned in response to an `Encrypt` or `Decrypt` command.

The default configuration of the PermConfig Register allows the customer to control the availability of the Encrypt, Decrypt, and Legacy commands using the ChipConfig Register. However, the ChipConfig.EncDecrE bit and ChipConfig.LegacyE bit will be ignored if the ATAES132A is configured at the factory to disable external encryption (see the PermConfig Register definition in Appendix E.2.14, PermConfig Register). If the ChipConfig.DecReadE bit is 1b, then the DecRead and WriteCompute commands (Section 7.7, DecRead Command) (Section 7.24, WriteCompute Command) are enabled. If the ChipConfig.DecReadE bit is 0b, then a parse error ReturnCode will be returned in response to a DecRead or WriteCompute command. If the ChipConfig.AuthComputeE bit is 1b, then the AuthCompute command (Section 7.3, AuthCompute Command) is enabled. If the ChipConfig.AuthComputeE bit is 0b, then a parse error ReturnCode will be returned in response to an AuthCompute command.

Table E-5. Coding of the Power-UpState Bits in the ChipConfig Register

Bit 7	Bit 6	Description
1	1	Device goes to the Active state at Power-Up.
1	0	
0	1	Device goes to the Standby state at Power-Up.
0	0	Device goes to the Sleep state at Power-Up.

The ChipConfig.PowerUpState bits are used to configure the behavior of the ATAES132A at initial power-up. Table E-5 shows the definition of the ChipConfig.PowerUpState bits. See Appendix L, Power Management for detailed information regarding the ATAES132A power management functions.

The default value of the ChipConfig Register is 0xC3. In this configuration, the ATAES132A goes to the Active state at Power-Up, the DecRead, WriteCompute, and AuthCompute commands are disabled and the Encrypt, Decrypt, and Legacy commands are enabled.

E.2.17 RFU Registers

Any Configuration Memory locations that are identified as RFU in Table E-1, the Configuration Memory map, are registers in customer-writable memory that are reserved by Atmel for future use (in a future ATAES family product or in a major product revision). The default value of the RFU registers is 0xFF.

The customer can write the RFU registers using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig Register definition in Appendix E.2.11, LockConfig Register). The RFU registers should be programmed to 0xFF only; all other values are prohibited.

E.2.18 CounterConfig Registers

The 16 CounterConfig Registers are used to individually configure the 16 Counters. Each CounterConfig Register controls one Counter. CounterConfig 00 controls Counter 00, CounterConfig 01 controls Counter 01, etc.

Each CounterConfig register is a 2-byte array that is stored as shown in Table E-6. The customer can write the CounterConfig Registers using standard I^2C or SPI Write commands unless the Configuration Memory has been locked (see the LockConfig Register definition in Section E.2.11). See Appendix H, Understanding Counters for additional Counter information.

Table E-6. Partial Configuration Memory Map Showing CounterConfig Register Byte Locations for Four Registers

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h
$F060_h$-$F067_h$	CounterConfig 0		CounterConfig 1		CounterConfig 2		CounterConfig 3	
	Byte 0	Byte 1	Byte 0	Byte 1	Byte 0	Byte 1	Byte 0	Byte 1

The CounterConfig Register imposes restrictions on the usage of the `Counter` command (see Section 7.5) with a Counter. The CounterConfig bits have no impact on the functionality of a Key Usage Counter. If a Counter is identified in a KeyConfig Register (see Appendix E.2.19) as a Key Usage Counter, then the Counter will increment each time the Key is used. The CounterConfig[CntID].IncrementOK is typically set to `0b` to prohibit the `Counter` Command from incrementing a Key Usage Counter.

Table E-7. Definition of the CounterConfig Register Bits(1)

CounterConfig Field	Byte	Bit	Description
IncrementOK	0	0	`1b` = Increments using the `Counter` command are permitted. `0b` = Increments using the `Counter` command are prohibited.
RequireMAC	0	1	`1b` = Increment operation requires an input MAC. `0b` = An input MAC is prohibited.
Reserved	0	2 to 7	Reserved for future use. All bits must be `0b`.
IncrID	1	0 to 3	KeyID of the key used to generate the `Counter` command input MAC for increment operations.
MacID	1	4 to7	KeyID of the key used to generate the `Counter` command output MAC for counter Read operations.

Note: 1. Changes to the CounterConfig Registers take effect immediately, which allows the functionality to be verified during the personalization process.

E.2.19 KeyConfig Registers

The 16 KeyConfig Registers are used to individually configure the 16 keys. Each KeyConfig Register controls one key. KeyConfig 00 controls Key 00, KeyConfig 01 controls Key 01, etc.

Each KeyConfig Register is a 4-byte array that is stored as shown in Table E-8. The customer can write the KeyConfig Registers using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig Register definition in Appendix E.2.11, LockConfig Register).

Table E-8. Partial Configuration Memory Map Showing KeyConfig Register Byte Locations for Two Registers

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h
$F080_h$-$F087_h$	KeyConfig 0				KeyConfig 1			
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 2	Byte 3

A key can be disabled by setting KeyConfig[KeyN].AuthKey to 1b and KeyConfig[KeyN].LinkPointer to contain “KeyN,” where the KeyN = KeyID of the key being configured.

Table E-9. Definition of the KeyConfig Register Bits [1][3]

KeyConfig Field	Byte	Bit	Description
ExternalCrypto	0	0	1b = Key can be used with the Encrypt and Decrypt commands.[2] 0b = Encrypt and Decrypt commands are prohibited.
InboundAuth	0	1	1b = Key can be used only by the Auth command for inbound-only or mutual authentication. Key cannot be used by any other command, but KeyID can be the target of a Key Management command. 0b = Key can be used for any purpose not prohibited by another KeyConfig bit, including outbound-only authentication.
RandomNonce	0	2	1b = Operations using this key require a random Nonce (see Section 7.19). 0b = The Nonce is not required to be random.
LegacyOK	0	3	1b = Key can be used with the Legacy command. 0b = Key cannot be used with the Legacy command.
AuthKey	0	4	1b = Key requires prior authentication using the KeyID stored in LinkPointer. 0b = Prior authentication is not required.
Child	0	5	1b = Key is permitted to be the target of a KeyCreate for Child and Parent and/or KeyLoad command. 0b = This use is prohibited.
Parent	0	6	1b = Key may be used as the VolatileKey parent by the KeyCreate or KeyLoad commands. The key may also be used as the Decrypt Key by the KeyImport command when the target key is VolatileKey (see Section 4.3). 0b = This use is prohibited.
ChangeKeys	0	7	1b = Key updates are permitted after locking. The new key is written using the EncWrite command with a MAC generated with the current value of key (see Section 7.11). 0b = Key updates with the EncWrite command are prohibited.
CounterLimit	1	0	1b = Usage count limits are enabled for this key (see CounterNum). 0b = There are no usage limits.
ChildMac	1	1	1b = An input MAC is required to modify this key using the KeyCreate command. 0b = KeyCreate command does not require an input MAC (it will be ignored if provided).
AuthOut	1	2	1b = I2C Auth signaling is enabled for this key (see Appendix J.5). 0b = I2C Auth signaling is disabled for this key.
AuthOutHold	1	3	1b = I2C AuthO output state is unchanged when an Authentication Reset is executed using this key. 0b = I2C AuthO output is reset when an Authentication Reset is executed using this key (see Appendix J.5).
ImportOK	1	4	1b = Key is permitted to be the target of a KeyImport command. 0b = KeyImport command is prohibited.
ChildAuth	1	5	1b = The KeyCreate command requires prior authentication using the KeyID stored in LinkPointer. 0b = Prior authentication is not required.

KeyConfig Field	Byte	Bit	Description
TransferOK	1	6	1b = Key is permitted to be the target of a KeyTransfer command (see Section 7.16). 0b = The KeyTransfer command is prohibited.
AuthCompute	1	7	1b = Key can be used with the AuthCompute command. 0b = Key cannot be used with the AuthCompute command.
LinkPointer	2	0 to 3	For child keys, stores the ParentKeyID. For all other keys, the KeyID of the authorizing key (see AuthKey).
CounterNum	2	4 to 7	Stores the CntID of the counter attached to this key for usage limits and/or for MAC calculation. MAC calculations will include the counter if Command Mode bit 5 is 1b, even if key usage limits are disabled.
DecRead	3	0	1b = The DecRead and WriteCompute commands can be run using this key. 0b = The DecRead and WriteCompute are prohibited.
Reserved	3	1 to 7	Reserved for future use.

Notes:
1. Changes to the KeyConfig Registers take effect immediately, which allows the functionality to be verified during the personalization process.
2. **Warning:** Since the Encrypt command does not include an input MAC, the Encrypt command can be exhaustively run with selected input data to attack the Key. Requiring authentication prior to allowing encryption makes these attacks more difficult. To require prior authentication, the AuthKey and RandomNonce bits must be set to 1b.
3. A Key can be disabled by setting KeyConfig[KeyN].AuthKey to 1b and KeyConfig[KeyN].LinkPointer to contain "KeyN", where KeyN = KeyID of the key being configured.

E.2.20 ZoneConfig Registers

The 16 ZoneConfig Registers are used to individually configure the 16 user zones. Each ZoneConfig Register controls one user zone. ZoneConfig 00 controls User Zone 00, ZoneConfig 01 controls User Zone 01, etc.

Each ZoneConfig Register is a 4-byte array that is stored as shown in Table E-10. The customer can write the ZoneConfig Registers using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig Register definition in Appendix E.2.11, LockConfig Register).

Table E-10. Partial Configuration Memory Map Showing ZoneConfig Register Byte Locations for the Two Registers

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h
F0C0$_h$-	ZoneConfig 0				ZoneConfig 1			
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 2	Byte 3

Table E-11. Definition of the ZoneConfig Register Bits (1)

ZoneConfig Field	Byte	Bit	Description
AuthRead	0	0	1b = Authentication is required to read data. 0b = Authentication is not required to read data.
AuthWrite	0	1	1b = Authentication is required to write data. 0b = Authentication is not required to write data.
EncRead	0	2	1b = Encryption is required to read data. 0b = Encryption is not required to read data.
EncWrite	0	3	1b = Encryption is required to write data. 0b = Encryption is not required to write data.
WriteMode	0	4 to 5	00b =Zone is permanently Read/Write. 01b =Zone is permanently Read-only. 10b =The ReadOnly byte determines if writes are permitted. 11b =The ReadOnly byte determines if writes are permitted, and the Lock command must include an authenticating MAC calculated using the KeyID stored in ZoneConfig[UZ].WriteID.
UseSerial	0	6	UseSerial = 1b and EncWrite = 1b, then SerialNum must be included in EncWrite operations. EncWrite = 0b, then this bit is ignored.
UseSmall	0	7	UseSmall = 1b and EncWrite = 1b, the first four bytes of SmallZone must be included in EncWrite operations. EncWrite = 0b, then this bit is ignored.
ReadID	1	0 to 3	KeyID that is used to encrypt data read from this zone. The same key is used to generate the MAC.
AuthID	1	4 to 7	KeyID that is used for inbound authentication before access is permitted.
VolatileTransferOK	2	0	1b = Key transfer from this User Zone to VolatileKey is permitted. 0b = Key transfer from this User Zone to VolatileKey is prohibited.
Reserved	2	1 to 3	Reserved for future use.
WriteID	2	4 to 7	KeyID that is used to decrypt data written to this zone. The same key is used to verify the MAC.
ReadOnly	3	0 to 7	The contents of this byte are ignored unless WriteMode contains 10b or 11b. 0x55 = User zone is Read/Write. If any other value = User zone is Read-only. This byte can be updated after the Configuration Memory is locked by using the Lock command (see Section 7.18, Lock Command).

Note: 1. Most changes to the ZoneConfig Registers take effect immediately. Changes to the AuthRead and EncRead bits do not affect the SPI or I^2C Read command until the next reset or power-up.

E.2.21 Counter Registers

The 16 Counter Registers are used to store the Counter values. The default value of the Counters is equivalent to a count value of zero. See Appendix H, Understanding Counters for Counter information.

The customer can write the Counter Registers using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig register definition in Appendix E.2.11, LockConfig Register).

E.2.22 FreeSpace Register

The FreeSpace Register is 96 bytes of memory for storage of customer data. The customer can write the FreeSpace Register using standard I^2C or SPI Write commands, unless the Configuration Memory has been locked (see the LockConfig register definition in Appendix E.2.11).

The default value of the FreeSpace Register is `0xFF` in all bytes. The FreeSpace Register can be programmed with any value; the contents will not change the behavior of the ATAES132A.

E.2.23 SmallZone Register

The SmallZone Register is 32 bytes of memory for storage of customer data. Optionally, the first four bytes of the SmallZone Register may be included in cryptographic calculations by setting Mode bit 7 to `1b`, as described in the command definitions in Section 7, Command Definitions. The customer can write the SmallZone Register using standard I^2C or SPI Write commands, unless the SmallZone Register has been locked (see the LockSmall Register definition in Appendix E.2.10, LockSmall Register).

The default value of the SmallZone Register is `0xFF` in all bytes. The SmallZone Register can be programmed with any value; the contents will not change the behavior of the ATAES132A.

Appendix F. Key Memory Map

Table F-1. ATAES132A Key Memory Map; Starts at Address 0xF200

Address	0_h / 8_h	1_h / 9_h	2_h / A_h	3_h / B_h	4_h / C_h	5_h / D_h	6_h / E_h	7_h / F_h
$F200_h$-$F207_h$	Key 00							
$F208_h$-$F20F_h$								
$F210_h$-$F217_h$	Key 01							
$F218_h$-$F21F_h$								
$F220_h$-$F227_h$	Key 02							
$F228_h$-$F22F_h$								
$F230_h$-$F237_h$	Key 03							
$F238_h$-$F23F_h$								
$F240_h$-$F247_h$	Key 04							
$F248_h$-$F24F_h$								
$F250_h$-$F257_h$	Key 05							
$F258_h$-$F25F_h$								
$F260_h$-$F267_h$	Key 06							
$F268_h$-$F26F_h$								
$F270_h$-$F277_h$	Key 07							
$F278_h$-$F27F_h$								
$F280_h$-$F287_h$	Key 08							
$F288_h$-$F28F_h$								
$F290_h$-$F297_h$	Key 09							
$F298_h$-$F29F_h$								
$F2A0_h$-$F2A7_h$	Key 0A							
$F2A8_h$-$F2AF_h$								
$F2B0_h$-$F2B7_h$	Key 0B							
$F2B8_h$-$F2BF_h$								
$F2C0_h$-$F2C7_h$	Key 0C							
$F2C8_h$-$F2CF_h$								
$F2D0_h$-$F2D7_h$	Key 0D							
$F2D8_h$-$F2DF_h$								
$F2E0_h$-$F2E7_h$	Key 0E							
$F2E8_h$-$F2EF_h$								
$F2F0_h$-$F2F7_h$	Key 0F							
$F2F8_h$-$F2FF_h$								

VolatileKey (KeyID = 0xFF) does not exist in EEPROM. It is a temporary key that resides in the internal SRAM memory. The internal SRAM cannot be accessed directly. See Section 4.3, VolatileKey Configuration for VolatileKey information.

Prior to locking the Key Memory, it can be written with either encrypted or cleartext data. Encrypted writes are performed using the EncWrite command (see Section 7.11, Encrypted Key Writes). Cleartext writes are performed using standard SPI or I^2C Write commands (see Section 5.2, Write). The Key Memory can never be read with the BlockRead command or the EncRead command, or with standard I^2C or SPI Read commands.

Appendix G. Understanding the STATUS Register

The 8-bit Device Status Register is used for handshaking between the Host microcontroller and the ATAES132A. The Host microcontroller is expected to read the STATUS Register before sending a command or reading a response.

G.1 Device Status Register (STATUS) Definition

Address 0xFFF0 contains the read-only Device Status Register, which indicates the current status of the ATAES132A device. The SPI Read Status Register command can be used to read the STATUS Register, as described in Appendix K.3.5, Read Status Register Command (RDSR).

This register can also be read with the standard I^2C or SPI Read Memory commands. Reading the STATUS Register does not increment the memory read address, and so a Host microcontroller can easily monitor the ATAES132A device status by repeatedly reading the STATUS Register.

Table G-1. Device Status Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EERR	RRDY	Reserved	CRCE	Reserved	WAKEb	WEN	WIP

Table G-2. Definition of the STATUS Register Bits(1)(2)

Bit	Definition
Bit 0 (WIP)	0b =The device is ready, waiting for a command. 1b =A Write cycle or a cryptographic operation is in progress.
Bit 1 (WEN)	0b =The device is not SPI Write enabled or is in I^2C interface mode. 1b =The device is SPI Write enabled.
Bit 2 (WAKEb)	0b = The device is not in the Sleep or Standby power state. 1b =Tthe device is in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b. This bit is reserved for future use. (1)
Bit 4 (CRCE)	0b =The most recent command block contained a correct Checksum (CRC). 1b =The most recent command block contained an error.
Bit 5 (Reserved)	Always 0b. This bit is reserved for future use. (1)
Bit 6 (RRDY)	0b =The Response Memory Buffer is empty. 1b =The Response Memory Buffer is ready to read.
Bit 7 (EERR)	0b =The most recent command did not generate an error during execution. 1b =The most recent command generated an execution error.

Note: 1. When the SPI RDSR command is used to read the STATUS Register during an EEPROM Write or during execution of any ATAES132A command, then Status bits 0 to 7 are 1b (see Appendix K.3.5, Read Status Register Command (RDSR). When the STATUS Register is read from address 0xFFF0 under the same circumstances, the reserved bits will read as 0b.

2. STATUS Register bits 0 to 7 are 1b during wake-up. During the first phase of wake-up and power-up. See Appendix L, Power Management for additional information.

The Device Status Register can always be read when the ATAES132A is configured for SPI interface mode, even if ATAES132A is processing a command or writing the EEPROM. When the ATAES132A is configured for I^2C interface mode, the `Random Read` command can only be used to read the STATUS Register only when the device address is ACKed.

If the ATAES132A is in the Sleep or Standby power state, reading the STATUS Register forces the ATAES132A to wake-up; the STATUS Register is `0xFF` until the wake-up process is complete.

G.1.1 WIP Status Bit [0]

The WIP status bit is used to indicate the device is busy or there is a “Write in progress.” If WIP = `0b`, then the ATAES132A is in the Active state and is waiting to receive a command. If WIP = `1b`, then ATAES132A is in the Active state and is performing an EEPROM Write or processing an ATAES132A command.

G.1.2 WEN Status Bit [1]

If ATAES132A is configured in I^2C interface mode, then the WEN Status bit is always `0b` (see Appendix J, I^2C Interface for I^2C information).

If the ATAES132A is configured in SPI interface mode, then the WEN status bit is 0b after the device initially powers up or exits the Sleep state (see Appendix K, SPI Interface for SPI interface information). When WEN = `0b`, the User Memory is Write protected and any attempt to write the User Memory using the SPI Write command will fail. The Host must send a SPI `WREN` command to the device to set WEN = `1b` prior to each SPI Write command.

If the ATAES132A is configured in SPI interface mode, then the WEN Status bit will return to `0b` when any Write instruction is received. The WEN Status bit can be forced to `0b` by sending a SPI `WRDI` command (See Appendix K.3.2, Write Disable Command (WRDI)), by sending a `RESET` command (See Section 7.22, Reset Command), or by putting the device in the Sleep state. Powering the device off will reset the WEN bit to `0b`. The SPI Read command and SPI `RDSR` command do not affect the state of the WEN bit.

It is not necessary to set WEN = `1b` prior to writing to the Command Memory Buffer or the IO Address Reset Register (see Appendix D, Command Memory Map). Writing the Command Memory Buffer or the IO Address Reset Register forces WEN to `0b`.

G.1.3 WAKEb Status Bit [2]

The WAKEb status bit is `0b` when the ATAES132A has completed a power-up sequence and is in the active state. WAKEb is `1b` when the ATAES132A is in the Sleep or Standby state, or is in the process of waking up.

Note: Reading the STATUS Register will cause a device in the Sleep state or Standby state to wake-up. (See Appendix L, Power Management for power state and power management information.)

G.1.4 CRCE Status Bit [4]

The CRCE status bit is set to `1b` if a block is received with a short Count or bad Checksum or if the block causes a buffer overrun. If only the Checksum (CRC) was incorrect, then the block may be resent without change. If the Command Memory Buffer contains a partial command block, then the CRCE status bit is `1b` and all other status bits are `0b`. This indicates that the correct Checksum has not yet been received. If the CRCE Status bit is `1b` and all the other Status bits are `0b` after the entire block has been sent, the IO Address Reset Register should be written before resending the block (see Appendix D.2, Response Memory Buffer for more information on the IO Address Reset Register).

The EERR bit will remain `0b` when a Checksum error occurs, and the Response Memory Buffer will remain empty because these errors do not result in a ReturnCode being generated. If a buffer overrun occurs, then the CRCE and EERR bits will be set to `1b`.

G.1.5 RRDY Status Bit [6]

The RRDY Status bit is 0b when the Response Memory Buffer is empty. If RRDY = 1b, then the Response Memory Buffer contains a response block or a ReturnCode resulting from the most recent command or command block received (see Appendix D.2, Response Memory Buffer for Response Memory Buffer information).

G.1.6 EERR Status Bit [7]

If the command is processed without error, the EERR bit is set to 0b. When any error other than a Checksum error occurs, the EERR Status bit is set to 1b to indicate an error. The Host can read the error code (ReturnCode) from the Response Memory Buffer (address 0xFE00) using a read command if the RRDY Status bit is 1b.

Reading the STATUS Register does not reset the Status Register bits or alter the Response Memory Buffer contents. Reading the Response Memory Buffer does not alter the contents of the Response Memory Buffer or the STATUS Register. Reading beyond the end of the Response Memory Buffer will not cause the STATUS Register bits to change.

The EERR status bit will be set to 1b if a SPI or I^2C Read is attempted using an invalid address or an address pointing to a protected portion of the memory. EERR will also be set to 1b if a SPI or I^2C read begins at an authorized address but continues into protected memory. In both of these cases, the RRDY status bit is 0b and the Response Memory Buffer will remain empty because these errors do not generate a ReturnCode. Reading beyond the end of user Zone F will not cause the EERR bit to be set to 1b.

Note: If a SPI or I^2C Read begins at an authorized address and continues into protected memory, the EERR bit will be set to 1b.

G.1.7 Reserved Status Bits [3, 5]

The Reserved Status bits are always 0b when the ATAES132A is capable of accepting a command. The Reserved Status bits are 1b during Power-Up and during Wake-Up from the Sleep state or the Standby state.

G.2 STATUS Register Behavior in the I^2C Interface Mode

The following sections describe the device behavior and expected STATUS Register values during commonly performed operations. In the I^2C interface mode, the ATAES132A will always NAK instructions containing a nonmatching I^2C Device Address. The ATAES132A will ACK instructions with a matching I^2C Device Address if the device is capable of accepting an instruction. See Appendix J, I^2C Interface for the I^2C interface specifications.

When the ATAES132A is busy or unable to respond for any reason, it will NAK a matching I^2C Device Address. The ACK/NAK response to the I^2C Device Address operates similar to the way the WIP status bit changes value in the SPI interface mode.

G.2.1 Power-Up

The ATAES132A will NAK all instructions received during Power-Up to indicate that it is not ready to accept a command from the Host. When the Power-Up process is complete (after time $t_{PU.RDY}$), then the ATAES132A will enter the state specified by ChipConfig Register bits 6 and 7; the Active state, the Standby state, or the Sleep state (see Appendix L.2.1, Power-Up). In I^2C interface mode, it is impossible to read the STATUS Register until the completion of Power-Up.

Upon completion of Power-Up, the Command Memory Buffer is empty, the Response Memory Buffer is all 0xFFs, and ChipState = 0xFFFF. The default EEPROM address is set to 0x0000, and the command and Response Memory Buffer pointers are set to the base address of the buffers. If the device is configured to enter the Active state at Power-Up, then STATUS will be 0x00, as shown in Table G-3.

Table G-3. Contents of the STATUS Register After Power-up to the Active State

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is in I^2C interface mode.
Bit 2 (WAKEb)	0b =Device is *not* in the Sleep or Standby Power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b = Response Memory Buffer is empty.
Bit 7 (EERR)	0b = No errors during execution.

If the device is configured to enter the Sleep state, then the ATAES132A will NAK any attempt to read the STATUS Register at the completion of Power-Up, as described in Appendix G.2.2, Wake-Up from Sleep. If the device is configured to enter the Standby state, then the ATAES132A will NAK any attempt to read the STATUS Register at the completion of Power-Up, as described in Appendix G.2.3, Wake-Up from Standby; ChipState will remain 0xFFFF in the Standby state.

Note: ACK polling or attempting to read the STATUS Register after Power-Up is completed will cause the device to Wake-Up.

G.2.2 Wake-Up from Sleep

The ATAES132A will NAK all instructions received during Wake-Up from the Sleep Power state to indicate that it is not ready to accept a command from the Host. When the Wake-Up process is complete (after time $t_{WupSL.RDY}$), then the ATAES132A will enter the Active state. In I^2C interface mode, it is impossible to read the STATUS Register until the Wake-Up is complete.

Upon completion of Wake-Up from Sleep, the Command Memory Buffer is empty, the Response Memory Buffer is all 0xFFs, and ChipState = 0x5555. The default EEPROM address is set to 0x0000, and the command and Response Memory Buffer pointers are set to the base address of the buffers. Upon completion of Wake-Up, the STATUS Register will be 0x00, as shown in Table G-3.

G.2.3 Wake-Up from Standby

The ATAES132A will NAK all instructions received during Wake-Up from the Standby Power state to indicate that it is not ready to accept a command from the Host. When the Wake-Up process is complete (after time $t_{WupSB.RDY}$), the ATAES132A will enter the Active state. In I^2C interface mode, it is impossible to read the STATUS Register until the Wake-Up is complete.

Upon completion of Wake-Up from Standby, the Command Memory Buffer is empty, and the Response Memory Buffer is all 0xFFs. ChipState will have the value it had prior to entering the Standby state. Upon completion of Wake-Up, the STATUS Register will be 0x00, as shown in Table G-3.

G.2.4 Read STATUS Register

To read the STATUS Register, the Host sends a Random Read Instruction (RREAD) with a starting address of 0xFFF0 when ATAES132A ACKs the I²C Device Address. Reading the STATUS Register does not increment the Read address, so the Host can poll the STATUS by reading any number of bytes, beginning with address 0xFFF0.

Reading the STATUS Register does not change the Command Memory Buffer contents or the Response Memory Buffer contents. Reading the STATUS Register does not change the Command Memory Buffer pointer or the Response Memory Buffer pointer. Reading the STATUS Register does not change the STATUS Register.

G.2.5 Read User Memory

The ATAES132A instructions for directly reading the User Memory are identical to the standard Atmel Serial EEPROM instructions. The Host can send a read memory instruction (READ, RREAD, SREAD) whenever the ATAES132A ACKs the I²C Device Address. If the address being read is valid and access is not prohibited, then the contents of that byte will be returned to the Host. If the address is invalid, or access is prohibited for any reason, then 0xFF will be returned to the Host in place of the prohibited byte.

A Read operation begins with an I²C Start condition and ends with an I²C NAK by the Host. If one or more bytes are accessed during the Read operation at an invalid or protected address, then the EERR bit will be set to 1b (see Table G-4). If all bytes accessed by the Read operation are valid and the Host satisfied the required access conditions, then the EERR bit will be set to 0b. The contents of the Command Memory Buffer and the Response Memory Buffer will remain unchanged.

Note: If an I²C Read begins at an authorized address and continues into protected memory, the EERR bit will be set to 1b.

Table G-4. Contents of the STATUS Register After an I²C Read Memory Operation

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is in I²C interface mode.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b = Response Memory Buffer is unchanged.(1)
Bit 7 (EERR)	0b =No errors during execution of the Read operation. 1b =0xFF was returned in place of one or more invalid or prohibited bytes read.

Note: 1. A Read Memory operation does not change the contents of the Response Memory Buffer. The EERR status bit is used to indicate success or an error. No ReturnCode is generated by a memory read error.

G.2.6 Write User Memory

The ATAES132A instructions for directly writing the User Memory are identical to the standard Atmel Serial EEPROM. The Host can send a Write Memory instruction (BWRITE, PWRITE) whenever the ATAES132A ACKs the I^2C Device Address. If the address being written is valid, access requirements have been satisfied and no page boundaries are crossed, then the data provided by the Host will be written after the Host generates an I^2C Stop condition. If the address is invalid or access is prohibited for any reason, then the ATAES132A will discard the data and no EEPROM Write will occur.

A Memory Write operation begins with an I^2C Start condition and ends with an I^2C Stop condition by the Host. If the Host does not provide an I^2C Stop condition, then no Write will occur, no ReturnCode will be generated, and the STATUS Register is `0x00` to indicate the ATAES132A is waiting for a command.

If the Host provides the required I^2C Stop condition, then the ATAES132A will NAK the I^2C Device Address during the EEPROM Write operation. When the Write operation is complete, then ATAES132A will ACK the I^2C Device Address.

Upon completion of a Memory Write operation, the Command Memory Buffer is empty, and the Response Memory Buffer contains a ReturnCode. The command and the Response Memory Buffer pointers are set to the base address of the buffers. The STATUS will be as shown in Table G-5.

Table G-5. STATUS Register Contents After an I^2C Write Memory Operation

Bit	Definition
Bit 0 (WIP)	`0b` =Device is ready, waiting for a command.
Bit 1 (WEN)	`0b` =Device is in I^2C interface mode.
Bit 2 (WAKEb)	`0b` =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always `0b`.
Bit 4 (CRCE)	`0b` =No Checksum error.
Bit 5 (Reserved)	Always `0b`.
Bit 6 (RRDY)	`1b` =Response Memory Buffer contains a response block.
Bit 7 (EERR)	`0b` =No errors during execution of the Write operation. `1b` =Write operation generated an error; see the ReturnCode for the cause.

G.2.7 Write Command Memory Buffer

To write the Command Memory Buffer, the Host sends a Write Memory instruction (BWRITE, PWRITE) with a starting address of `0xFE00` when the ATAES132A ACKs the I^2C Device Address. As each byte is written, the Command Memory Buffer pointer increments by one.

A command block begins with the COUNT byte and ends with the 2-byte Checksum (see Section 6.1, Command Block and Packet). If the entire command block is not received, then the device will not attempt to process the command and will not generate a response block. The STATUS Register will have CRCE = `1b` until the entire command block is received (as shown in Table G-6).

Table G-6. Contents of the STATUS Register if the Command Memory Buffer Contains a Partial Command Block

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is in I²C interface mode.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	1b =Checksum error (the Checksum has not yet been received).
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is unchanged.
Bit 7 (EERR)	0b =No errors during execution of the command block (it was not executed yet).

If the Host provides a complete command block, then the ATAES132A will NAK the I²C Device Address during command processing. When command processing is complete, then the ATAES132A will ACK the I²C Device Address.

If the command block contains a bad Checksum or a short Count or if the block causes a buffer overrun, then the CRCE bit of the STATUS Register will be set to 1b, as shown in Table G-7. The Response Memory Buffer will be unchanged because no ReturnCode is generated by these error conditions. The EERR bit is 1b if a buffer overrun error occurs. The EERR bit is 0b if a bad Checksum or short Count error occurs.

If the command block contains a good Checksum, then the ATAES132A will process the command and load the response in the Response Memory Buffer. Upon completion of command processing, the RRDY bit of the STATUS Register is set to 1b, as shown in Table G-7.

Table G-7. Contents of the STATUS Register After an I²C Write Command Memory Buffer Resulting in CRCE = 1b

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is in I²C interface mode.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	1b =Checksum, Short Count, or command buffer overrun error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is unchanged.
Bit 7 (EERR)	0b =No errors during execution of the command block (it was not executed). 1b =Command buffer overrun error.

Table G-8. Contents of the STATUS Register After an I²C Write Command Memory Buffer Resulting in CRCE = 0b

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is in I²C interface mode.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	1b =Response Memory Buffer contains a response block.
Bit 7 (EERR)	0b =No errors during execution of the command block. 1b =Command block generated an error; see the ReturnCode for the cause.

Writing the Command Memory Buffer resets the Response Memory Buffer pointer to the base address. Writing the Command Memory Buffer does not change the Response Memory Buffer contents until the entire command block is received and processed.

The Host can rewrite the contents of the Command Memory Buffer by resetting the buffer pointer (by writing the IO Address Reset Register) and sending a Write Memory instruction (BWRITE, PWRITE) with a starting address of 0xFE00.

Note: If the Host must write the Command Memory Buffer with more bytes than is required to send the command block due to hardware limitations, then the Host should transmit 0xFF bytes after the checksum. The extra bytes will be discarded by the ATAES132A and will not result in a buffer overrun or any other error.

G.2.8 Read Response Memory Buffer

To read the Response Memory Buffer, the Host sends a Random Read Memory instruction (RREAD) with a starting address of 0xFE00 when the ATAES132A ACKs the I²C Device Address. The Host can read any number of bytes from the Response Memory Buffer without causing an error. As each byte is read, the Response Memory Buffer pointer increments by one. If the Host reads beyond the end of the response block, then 0xFF will be returned for any byte after the Checksum.

Reading the Response Memory Buffer does not change the Command Memory Buffer contents or the Response Memory Buffer contents. Reading the Response Memory Buffer resets the Command Memory Buffer pointer to the base address. Reading the Response Memory Buffer does not change the STATUS Register.

The Host can reread the contents of the Response Memory Buffer by resetting the buffer pointer (by writing the IO Address Reset Register) and sending a Random Read Memory instruction (RREAD) with a starting address of 0xFE00.

G.2.9 Write IO Address Reset Register

To reset the pointer for the Command Memory Buffer and the pointer for the Response Memory Buffer, the Host sends a Write Memory instruction (BWRITE, or PWRITE) with a starting address of 0xFFE0. The IO Address Reset Register can be written with 1 to 32 bytes of data without generating an error; the data bytes will be ignored. The command and the Response Memory Buffer pointers are set to the base address of the buffers. The Command Memory Buffer is empty, and the Response Memory Buffer contents are unchanged. Writing the IO Address Reset Register changes the CRCE Status bit to 0b; all of the other status bits are unchanged.

G.3 STATUS Register Behavior in the SPI Interface Mode

The following sections describe the device behavior and expected STATUS Register values during commonly performed operations. See Appendix K, SPI Interface for the SPI interface specifications. In SPI interface mode, there are two ways to read the STATUS Register:

- Using the SPI RDSR command, or
- Reading STATUS from address 0xFFF0.

When the ATAES132A is busy or unable to respond for any reason, the WIP Status bit is 1b.

G.3.1 Power-Up

The ATAES132A will return 0xFF in response to a SPI RDSR command during Power-Up to indicate that it is not ready to accept a command from the Host. When the power-up process is complete (after time $t_{PU.RDY}$), the ATAES132A will enter the state specified by ChipConfig Register bits 6 and 7 (see Appendix L.2.1, Power-Up): the Active state, the Standby state, or the Sleep state.

Upon completion of Power-Up, the Command Memory Buffer is empty, the Response Memory Buffer is all 0xFF, and ChipState = 0xFFFF. The default EEPROM address is set to 0x0000, and the command and Response Memory Buffer pointers are set to the base address of the buffers. If the device is configured to enter the Active state, then the STATUS will be 0x00, as shown in Table G-9.

Table G-9. Contents of the STATUS Register After Power-up to the Active State

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is empty.
Bit 7 (EERR)	0b =No errors during execution.

If the device is configured to enter the Standby or Sleep mode after power-up, then the STATUS will be 0xFF at the completion of the power-up process as described in this section. STATUS will remain 0xFF while the device is in Standby or Sleep mode.

Note: Reading the STATUS Register after Power-Up is completed will cause the device to Wake-Up.

G.3.2 Wake-Up State from Sleep State

The ATAES132A will return 0xFF in response to a SPI RDSR command during Wake-Up from the Sleep Power state to indicate it is not ready to accept a command from the Host. When the wake-up process is complete (after time $t_{WupSL.RDY}$), ATAES132A will enter the Active state. After time $t_{WupSL.STATUS}$, it is possible to read the STATUS Register.

Upon completion of Wake-Up state from Sleep state, the following occurs:

- Command Memory Buffer is empty,
- Response Memory Buffer is all 0xFFs,
- ChipState = 0x5555,
- Default EEPROM address is set to 0x0000,
- Command and Response Memory buffer pointers are set to the base address of the buffers.

Upon completion of Wake-Up the STATUS will be 0x00 as shown in Table G-3.

G.3.3 Wake-Up State from Standby State

ATAES132A will return 0xFF in response to a SPI RDSR command during Wake-Up state from the Standby Power state to indicate that it is not ready to accept a command from the Host. When the wake-up process is complete (after time $t_{WupSB.RDY}$), ATAES132A will enter the Active state. After time $t_{WupSB.STATUS}$, it is possible to read the STATUS Register.

Upon completion of the Wake-Up state from the Standby state, the

- Command Memory Buffer is empty,
- Response Memory Buffer is all 0xFFs,
- ChipState will be the value it had prior to entering the Standby state.

Upon completion of the wake-up process, the STATUS will be 0x00 as shown in Table G-3.

G.3.4 Read STATUS Register

To read the STATUS Register, the Host sends a Read Memory Instruction (READ) with a starting address of 0xFFF0.

Reading the STATUS Register does not change the Command Memory Buffer contents or the Response Memory Buffer contents. Reading the STATUS Register does not change the Command Memory Buffer pointer or the Response Memory Buffer pointer. Reading the STATUS Register does not change the STATUS Register.

G.3.5 Read User Memory

The ATAES132A instructions for directly reading the User Memory are identical to standard Atmel Serial EEPROM instructions. The Host can send a Read whenever WIP is 0b.

- If the address being read is valid and access is not prohibited, the contents of that byte will be returned to the Host.
- If the address is invalid or access is prohibited for any reason, 0xFF will be returned to the Host in place of the prohibited byte.
- If one or more bytes are accessed during the Read operation at an invalid or protected address, then the EERR bit will be set to 1b (see Table G-10).
- If all bytes accessed by the Read operation are valid and the Host satisfied the required access conditions, the EERR bit will be set to 0b.

The contents of the Command Memory Buffer and the Response Memory Buffer will remain unchanged.

Table G-10. STATUS Register Contents After a SPI Read Memory Operation

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby Power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is unchanged.(1)
Bit 7 (EERR)	0b = No errors during the execution of the Read operation. 1b = 0xFF was returned in place of one or more invalid or prohibited bytes read.

Note: 1. A Read memory operation does not change the contents of the Response Memory Buffer. The EERR Status bit is used to indicate success or to indicate an error. No ReturnCode is generated by a memory Read error.

G.3.6 Write User Memory

The ATAES132A instructions for directly writing the User Memory are identical to standard Atmel Serial EEPROMs. The Host can send a Write Memory Instruction (WRITE) whenever WIP is 0b.

- Data provided by the Host will be written if:
  - The address being written is valid,
  - Access requirements have been satisfied, and
  - No page boundaries are crossed.
- ATAES132A will discard the data and no EEPROM Write will occur if:
  - The address is invalid or
  - Access is prohibited for any reason.

Upon completion of a Memory Write operation:

- Command Memory Buffer is empty,
- Response Memory Buffer contains a ReturnCode,
- Command and Response Memory buffer pointers are set to the base address of the buffers,
- STATUS will be as shown in Table G-11.

Table G-11. STATUS Register Contents After a SPI Write Memory Operation

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer contains a response block.(1)
Bit 7 (EERR)	0b = No errors during the execution of the Write operation. 1b = Write operation generated an error. See the ReturnCode for the cause.

G.3.7 Write Command Memory Buffer

To write the Command Memory Buffer, the Host sends a Write Memory Instruction (WRITE) with a starting address of 0xFE00 whenever WIP is 0b. The Command Memory Buffer pointer increments by one as each byte is written.

A Command Block begins with the COUNT byte and ends with the two byte Checksum (see Section 6.1, Command Block and Packet). If the entire Command Block is not received, then the device will not attempt to process the command; it will not generate a Response Block. The STATUS Register will have the CRCE bit = 1b until the entire Command block is received (as shown in Table G-12).

Table G-12. STATUS Register Contents If the Command Memory Buffer Contains a Partial Command Block

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error (The checksum has not yet been received).
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is unchanged.
Bit 7 (EERR)	0b = No errors during the execution of the Command Block (It was not executed yet).

If the Host provides a complete Command Block, then WIP will be 1b during command processing. When command processing is complete, then WIP will be 0b.

If the Command Block contains a bad Checksum and a short COUNT or the block causes a buffer overrun, then the CRCE bit of the STATUS Register will be set to 1b as shown in Table G-13. The Response Memory Buffer will be unchanged because no ReturnCode is generated by these error conditions. The EERR Status bit is 1b if a buffer overrun error occurs; the EERR bit is 0b if a bad Checksum or short COUNT error occurs.

If the Command Block contains a good Checksum, then ATAES132A will process the command and load the response in the Response Memory Buffer. Upon completion, command processing the RRDY bit of the STATUS Register is set to 1b as shown in Table G-14.

Table G-13. STATUS Register Contents After a SPI Write Command Memory Buffer Resulting in CRCE = 1b

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	1b =Checksum error, short COUNT, or command buffer overrun error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	0b =Response Memory Buffer is unchanged.
Bit 7 (EERR)	0b = No errors during the execution of the Command Block. (It was not executed yet.) 1b = Command buffer overrun error.

Table G-14. STATUS Register Contents After a SPI Write Command Memory Buffer Resulting in CRCE = 0b

Bit	Definition
Bit 0 (WIP)	0b =Device is ready, waiting for a command.
Bit 1 (WEN)	0b =Device is not Write enabled.
Bit 2 (WAKEb)	0b =Device is not in the Sleep or Standby power state.
Bit 3 (Reserved)	Always 0b.
Bit 4 (CRCE)	0b =No Checksum error.
Bit 5 (Reserved)	Always 0b.
Bit 6 (RRDY)	1b =Response Memory Buffer contains a Response block.
Bit 7 (EERR)	0b = No errors during the execution of the Command Block. (It was not executed yet.) 1b = Command buffer generated an error. See the ReturnCode for the cause.

Writing the Command Memory Buffer resets the Response Memory Buffer pointer to the base address. Writing the Command Memory Buffer does not change the Response Memory Buffer contents until the entire Command block is received and processed.

The Host can rewrite the contents of the Command Memory Buffer by resetting the buffer pointer (by writing the IO Address Reset Register) and sending a Write Memory Instruction (WRITE) with a starting address of 0xFE00.

Note: If the Host must write the Command Memory Buffer with more bytes than is required to send the Command Block due to hardware limitations, then the Host should transmit 0xFF bytes after the Checksum. The extra bytes will be discarded by ATAES132A and will not result in a buffer overrun or any other error.

G.3.8 Read Response Memory Buffer

To read the Response Memory Buffer, the Host sends a Read Memory Instruction (READ) with a starting address of 0xFE00. The Host can read any number of bytes from the Response Memory Buffer without causing an error. As each byte is read, the Response Memory Buffer pointer increments by one. If the Host reads beyond the end of the Response Block, then 0xFF will be returned for any byte after the Checksum.

Reading the Response Memory Buffer does not change the Command Memory Buffer contents or the Response Memory Buffer contents. Reading the Response Memory Buffer resets the Command Memory Buffer pointer to the base address. Reading the Response Memory Buffer does not change the STATUS Register.

The Host can reread the contents of the Response Memory Buffer by resetting the buffer pointer (by writing the IO Address Reset Register) and sending a Random Read Memory Instruction (RREAD) with a starting address of 0xFE00.

G.3.9 Write IO Address Reset Register

To reset the pointer for the Command Memory Buffer and the pointer for the Response Memory Buffer, the Host sends a Write Memory Instruction (WRITE) with a starting address of 0xFFE0. The IO Address Reset Register can be written with 1 to 32 bytes of data without generating an error; the data bytes will be ignored. The Command and Response Memory buffer pointers are set to the base address of the buffers. The Command Memory Buffer is empty, and the Response Memory Buffer contents are unchanged. Writing the IO Address Reset Register changes the CRCE Status bit to 0b; all of the other STATUS bits are unchanged.

Appendix H. Understanding Counters

Each Counter can increment up to a value of 2,097,151 using the `Count` command, after which, the Counter can no longer be changed. Counters attached to keys are incremented each time the key is used when the Usage Counter reaches its limit; the key is disabled. Counters can also be incremented using the `Count` Command. The value in the Counter can never be reset or lowered. The Counters include a power interruption protection feature to prevent corruption of the Count value if power is removed during the increment operation.

On shipment from Atmel, the Counter Registers are initialized to their lowest value. The initial value of each Counter may be written to a different value at personalization prior to locking the configuration.

H.1 Counter Registers

Each Counter Register contains two Count values to prevent the Count value from being corrupted if power is interrupted during a Counter increment operation. Each Count value is stored as a combination of two Count fields:

- Counter A is stored in LinCountA and BinCountA.
- Counter B is stored in LinCountB and BinCountB.

Table H-1 shows the location of the Count fields within the Counter register in Configuration Memory.

Table H-1. Partial Configuration Memory Map Showing Counter Register Field Locations

Address	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h
$F100_h$-$F107_h$	Counter 00							
	LinCountA		LinCountB		BinCountB		BinCountA	

Counter Registers can always be read from the Configuration Memory using the `BlockRead` command; however, the `Count` command is the preferred method of reading the Counters. When the Counter is read using the `Count` command, ATAES132A automatically selects the appropriate Counter register fields and returns them to the Host in the Response Packet. See Section 7.5, `Counter` Command.

H.2 Reading the Counter

The `Counter` command is the recommended method for reading a Counter. The `Counter` command returns a four byte CountValue field which is formatted as shown in Figure H-1. Optionally, the `Counter` command can also return a MAC for cryptographic authentication of the CountValue. The definition of the CountValue field is shown in Table H-2. See Section 7.5, `Counter` Command.

Figure H-1. CountValue Field

Byte 0	Byte 1	Byte 2	Byte 3
LinCount	CountFlag	BinCount	

The CountValue contains a Linear Counter Field (LinCount), a Binary Counter field (BinCount), and the CountFlag field. The CountFlag field indicates if the Counter value was read from the Counter A or Counter B EEPROM location. CountFlag also indicates if the 8 bit LinCount field is the Most Significant Byte (MSB) or Least Significant Byte (LSB) of the 16 bit LinCount field in EEPROM. If the LSB of LinCount has been returned, then the LinCount field value is 1 to 8; if the MSB of LinCount has been returned, then the LinCount field value is 9 to 16.

Table H-2. Definition of the CountValue field in the Response to the `Counter` Command

Byte	Name	Description
0	LinCount	Contains the eight bit linear Counter value identified in the CountFlag field.
1	CountFlag	`0x00` = LinCount contains the LSB of LinCountA. BinCount contains the BinCountA value. `0x02` = LinCount contains the MSB of LinCountA. BinCount contains the BinCountA value. `0x04` = LinCount contains the LSB of LinCountB. BinCount contains the BinCountB value. `0x06` = LinCount contains the MSB of LinCountB. BinCount contains the BinCountB value. All other values are reserved for future use.
2	BinCount (MSB)	Contains the Most Significant Byte of the binary counter identified in the CountFlag field.
3	BinCount (LSB)	Contains the Least Significant Byte of the binary counter identified in the CountFlag field.

The equivalent decimal value of the Counter can be determined using the following equation:

$$CountValue = (BinCount*32) + (CountFlag/2)*8 + Lin2Bin(LinCount)$$

Here, Lin2Bin defines a function that converts a linear Counter value to corresponding binary value. `0xFFFF` converts to zero; `0xFFFE` converts to one; and up to `0x8000` which converts to 15.

H.3 Personalizing the Counters

The Counter registers are personalized with initial values prior to locking the Configuration Memory. The standard Serial EEPROM Write commands are used to write Configuration Memory (see Section 5.1.3, Read the STATUS Register). The `Lock` command is used to lock the Configuration Memory (see Section 7.18, Lock Command).

The initial value of the Counter registers can be determined using the following procedure:

Divide the Counter preset value by 32. The quotient is the value of BinCountA.

- If the remainder is less than 0.5, then:
  - BinCountB is one less than BinCountA
  - The remainder x 32 = the number of zeros in LinCountA
  - LinCountB = `0x0000`
- If the remainder is equal or greater than 0.5, then:
  - BinCountB is equal to BinCountA
  - (The remainder – 0.5) x 32 = the number of zeros in LinCountB
  - LinCountA = `0x0000`

Example 1: Preset to 8,159

- 8,159/32 = 254.96875
  - Binary Counter A = `254` or `0x00fe`
  - Binary Counter B = `0x00fe` (remainder is greater than 0.5)
  - Linear Counter B = `0x8000` (0.46875 x 32 = 15, Linear Counter B has 15 zeros)
  - Linear Counter A = `0x0000`

Example 2: Preset to 1,000,000

- 1,000,000/32 = 31250.0
  - Binary Counter A = `31250` or `0x7a12`
  - Binary Counter B = `0x7a11` (remainder is less than 0.5)
  - Linear Counter A = `0xFFFF` (remainder is zero, Linear Counter A has no zeros)
  - Linear Counter B = `0x0000`

Appendix I. Cryptographic Computations

ATAES132A implements all of its cryptographic commands using AES in CCM mode with a 128-bit key length per NIST SP800-38C. CCM mode provides both confidentiality and integrity checking with a single key. The integrity MAC includes both the encrypted data and additional authenticate-only data bytes. The particular information authenticated with each command is described within the command descriptions in Section 7, Command Definitions.

The device construction ensures that the Nonce will be unique for each MAC calculated.

I.1 MacCount

The one byte MacCount is stored in an internal register, and is used in the AES-CCM computations. Since MacCount changes, it speeds up computation by eliminating the need to generate a new random Nonce for every crypto computation. This register is incremented prior to performing each MAC calculation.

The MacCount Register is set to zero when the `Nonce` command is executed, and is subsequently incremented prior to any MAC computation being performed. Because of this, the value that will be used for calculating the first MAC of the first command after the `Nonce` command is MAC = 1.

There are two commands (`Auth` and `KeyCreate`) which can be configured to both verify an input MAC and calculate an output MAC. When either of these two commands is run in mutual-authentication mode, MacCount will be incremented twice.

The value of MacCount for a particular MAC calculation is always one greater than that used for the previous MAC calculation. After 255 MAC calculations, the device will invalidate the internal Nonce, and commands that require a valid Nonce will fail. At this point, a new `Nonce` command must be run to generate a new Nonce.

The MacCount is set to zero if any of the following events occurs:

- The `Nonce` command is executed.
- A MAC compare operation fails.
- MacCount reaches the maximum count.
- A Reset event occurs: Power-Up (see Appendix L.3.1, ChipState = Power-Up), Wake-Up from Sleep (see Appendix L.3.2, ChipState = Wake-Up from Sleep), the `Reset` command (see Section 7.22, `Reset` Command), or a Security Tamper is activated, causing the hardware to reset.

If a CRC error occurs on the incoming command packet, then MacCount will not be incremented. If the device receives any command that does not involve MAC computation, the MacCount will not be incremented.

If a cryptographic command is received that involves MAC computation, then the MacCount will be incremented regardless of whether or not there is a subsequent success or failure of the command. The MacCount will also be incremented regardless of whether or not the particular instance of the command uses the cryptographic engine. If a command fails due to a MAC comparison failure, then the Nonce is invalidated and the MacCount Register is set to zero.

The current value of this register should be known by the system; however, it may also be read out of the device at any time using the `INFO` command (See Section 7.12, `INFO` Command).

I.2 MacFlag

To prevent spoofing of the MAC value, a flag byte is included in each MAC calculation. MacFlag provides information about the state of the device during the MAC calculation.

Table I-1. Definition of the MacFlag bits

Bit	Name	Notes
0	Random	1b = The Nonce command was run with the RNG enabled, and the Nonce is guaranteed to be unique. 0b = The Nonce value has been sent to the device by the system and may not be unique.
1	Input	1b = For MAC values that are sent to the device as inputs. 0b = For MAC values output by the ATAES132A.
3 – 7	Zero	All bits must be 0b.

I.3 MAC Generation

The following example shows how the integrity MAC is calculated for an authentication operation requiring up to 14 bytes of authenticate-only data. This operation involves three passes through the AES crypto engine; all three using the same key. If there are more than 14 bytes of authenticate-only data, then another pass through the AES crypto engine is required.

There are two passes through the AES crypto engine in CBC mode to create the cleartext MAC. The inputs to the crypto engine for those blocks are labeled B0 and B1, and the outputs are B'0 and B'1, respectively.

- **B0** is composed of the following 128 bits:
  - 1 byte flag, a fixed value of b0111 1001.
  - 12 byte Nonce, as generated by the Nonce command.
  - 1 byte MacCount, one for first MAC generation.
  - 2 byte length field, always 0x00 00 for authentication only.
- **B1** is the XOR of B'0 with the following 128 bits:
  - 2 byte length field, size of authenticate-only data.
  - 14 byte data to be authenticated only.
- **B'1** is the cleartext MAC, which must be encrypted before being sent to the system.

There is one additional pass through the AES crypto engine in CTR mode to create the key block that is used to encrypt the MAC. The input to the crypto engine for this block is labeled A0 and the output is A'0. A'0 is the MAC sent to the system as the output parameter of the Auth command.

- **A0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of b0000 0001.
  - 12 byte Nonce – as generated by ATAES132A during Nonce command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always 0x00 00 for A0.
- **A'0** is XOR'd with the cleartext MAC (B'1) and sent to the system.

Input integrity MACs for Auth, Counter, KeyCreate, and Lock are also calculated using this procedure. If the input MAC does not match A'0, then the command returns an AUTH error.

I.4 Data Encryption

The following example shows how the encrypted data and integrity MAC are calculated for a 128 bit data read from the device with up to 14 bytes of authenticate-only data. This operation involves five passes through the AES crypto engine; all five using the same key. If there are more than 14 bytes of authenticate-only data and/or more than 128 bits of data being read, then one, two, or three more passes through the AES crypto engine are required.

There are three passes through the AES crypto engine in CBC mode to create the cleartext MAC. The inputs to the crypto engine for those blocks are labeled B0, B1, and B2, and the outputs are B'0, B'1 and B'2, respectively.

- **B0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0111 1001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte length field – max `0x00 20` if 256 bits of encrypted data, min `0x00 01` for one byte.
- **B1** is the XOR of B'0 with the following 128 bits:
  - 2 byte length field – size of authenticate-only data.
  - 14 byte data to be authenticated only.
- **B2** is the XOR of B'1 with the following 128 bits:
  - 16 bytes cleartext data.
- **B'2** is the cleartext MAC – which must be encrypted before being sent to the system.

There are two passes through the AES crypto engine in CTR mode to create the key block that is used to encrypt the data and the MAC. The inputs to the crypto engine for those blocks are labeled A0 and A1, and the outputs are A'0 and A'1, respectively. A'0 and A'1 are the blocks sent to the system as the output parameters of the `EncRead` and `Decrypt` commands.

- **A0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 00` for A0.
- **A'0** is XOR'd with the cleartext MAC and sent to the system.
- **A1** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by ATAES132A during `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 01` for A1.
- **A'1** is XOR'd with the cleartext data and sent to the system.

This sequence is also used for the `Encrypt` command, in addition to EncRead.

I.5 Data Decryption

The following example shows how the encrypted data and integrity MAC are calculated for a 128 bit data block write to the device with up to 14 bytes of authenticate-only data. This operation involves five passes through the AES crypto engine; all five using the same key. If there are more than 14 bytes of authenticate-only data and/or more than 128 bits of data being written, then one, two, or three more passes through the AES crypto engine are required.

There are two passes through the AES crypto engine in CTR mode to create the key block that is used to decrypt the data and the MAC. The inputs to the crypto engine for those blocks are labeled A0 and A1, and the outputs are A'0 and A'1, respectively. A'0 and A'1 are the blocks sent to the system as the output parameters of the `EncRead` and `Decrypt` commands.

- **A0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 00` for A0.
- **A'0** is XOR'd with the encrypted input MAC and stored in the internal SRAM as the MAC T.
- **A1** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by ATAES132A during `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 01` for A1.
- **A'1** is XOR'd with the encrypted input data and stored in the internal SRAM as the message M.

There are three passes through the AES crypto engine in CBC mode to create the expected MAC value. The inputs to the crypto engine for those blocks are labeled B0, B1, and B2, and the outputs are B'0, B'1, and B'2, respectively.

- **B0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0111 1001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte length field – max `0x00 20` if 256 bits of encrypted data, min `0x00 01` for one byte.
- **B1** is the XOR of B'0 with the following 128 bits:
  - 2 byte length field – size of authenticate-only data.
  - 14 byte data to be authenticated only.
- **B2** is the XOR of B'1 with the following 128 bits:
  - 16 bytes of cleartext message M.
- **B'2** is the cleartext MAC. If this matches the stored T value, then the write to memory proceeds. If there is no match, the device returns an error flag and does not modify memory.

This sequence is also used for the `Decrypt` and `KeyLoad` commands, in addition to EncWrite.

I.6 Auth Command MAC

The MACs are calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
11 bytes	FirstBlock field containing:	
	1 byte	Auth Opcode (0x03)
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
	4 bytes	0x00
1 byte	Padding of value 0x00	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

4 bytes	Usage Counter value, or 0x00 if not selected.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.7 AuthCheck Command – Auth MAC

The Auth command MAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
11 bytes	FirstBlock field containing:	
	1 byte	Auth Opcode (0x03)
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
	4 bytes	0x00
1 byte	Padding of value 0x00	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

16 bytes	SecondBlock field containing:	
	4 bytes	Usage counter value, or 0x00 if not selected.
	8 bytes	SerialNum[0:7], or 0x00 if not selected.
	4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.8 AuthCheck Command – Counter MAC

The Counter command MAC is calculated using the following 14 bytes in the default authenticate-only block:

- 2 bytes — ManufacturingID
- 11 bytes — FirstBlock field containing:
  - 1 byte — Counter Opcode (0x0A)
  - 1 byte — Mode
  - 2 bytes — Param1
  - 2 bytes — Param2
  - 1 byte — MacFlag
  - 4 bytes — CountValue, the output parameter
- 1 byte — Padding of value 0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

- 16 bytes — SecondBlock field containing:
  - 4 bytes — Usage counter value, or 0x00 if not selected.
  - 8 bytes — SerialNum[0:7], or 0x00 if not selected.
  - 4 bytes — SmallZone[0:3], or 0x00 if not selected.

I.9 AuthCompute Command – Auth MAC

The Auth command MAC is calculated using the following 14 bytes in the default authenticate-only block:

- 2 bytes — ManufacturingID
- 11 bytes — FirstBlock field containing:
  - 1 byte — Auth Opcode (0x03)
  - 1 byte — Mode
  - 2 bytes — Param1
  - 2 bytes — Param2
  - 1 byte — MacFlag
  - 4 bytes — 0x00
- 1 byte — Padding of value 0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

- 16 bytes — SecondBlock field containing:
  - 4 bytes — Usage counter value, or 0x00 if not selected.
  - 8 bytes — SerialNum[0:7], or 0x00 if not selected.
  - 4 bytes — SmallZone[0:3], or 0x00 if not selected.

I.10 AuthCompute Command – Counter MAC

The Counter command MAC is calculated using the following 14 bytes in the default authenticate-only block:

- 2 bytes — ManufacturingID
- 11 bytes — FirstBlock field containing:
  - 1 byte — Counter Opcode (0x0A)
  - 1 byte — Mode
  - 2 bytes — Param1
  - 2 bytes — Param2
  - 1 byte — MacFlag
  - 4 bytes — 0x00
- 1 byte — Padding of value 0x00.

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

- 16 bytes — SecondBlock field containing:
  - 4 bytes — Usage counter value, or 0x00 if not selected.
  - 8 bytes — SerialNum[0:7], or 0x00 if not selected.
  - 4 bytes — SmallZone[0:3], or 0x00 if not selected.

I.11 BlockRead Command

The BlockRead command does not perform a cryptographic operation, and does not use or generate a MAC.

I.12 Counter Command MAC

The InMAC is calculated using the following 14 bytes in the default authenticate-only block:

- 2 bytes — ManufacturingID
- 1 byte — Counter Opcode (0x0A)
- 11 bytes — FirstBlock field containing:
  - 1 byte — Mode
  - 2 bytes — Param1
  - 2 bytes — Param2
  - 1 byte — MacFlag
  - 4 bytes — CountValue
  - 1 byte — 0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the InMAC calculation:

- 4 bytes — Usage counter value for MAC generation key, or 0x00 if not selected.
- 8 bytes — SerialNum[0:7], or 0x00 if not selected.
- 4 bytes — SmallZone[0:3], or 0x00 if not selected.

The OutMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID
1 byte	Counter Opcode (0x0A)
1 byte	Mode
2 bytes	Param1
2 bytes	Param2
1 byte	MacFlag
4 bytes	CountValue, the output parameter
1 byte	0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the OutMAC calculation:

4 bytes	Usage counter value for MAC generation key, or 0x00 if not selected.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.13 Crunch Command

The Crunch command does not perform a cryptographic operation, and does not use or generate a MAC.

I.14 DecRead Command

The MAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
1 byte	EncRead Opcode (0x04)	
6 bytes	FirstBlock field containing:	
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
5 bytes	0x00	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

16 bytes	SecondBlock field containing:	
	4 bytes	Usage counter value, or 0x00 if not selected.
	8 bytes	SerialNum[0:7], or 0x00 if not selected.
	4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.15 Decrypt Command MAC

In Normal Decryption mode, the InMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID
1 byte	Decrypt Opcode (0x07)
1 byte	Mode
2 bytes	Param1
2 bytes	Param2
1 byte	MacFlag
5 bytes	0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value, or 0x00 if not selected or if KeyID is VolatileKey.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.15.1 Client Decrypt MAC

In Client Decryption mode, the InMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID
1 byte	Encrypt Opcode (0x06)
1 byte	Mode
2 bytes	Upper byte = 0x00, lower byte = EKeyID
2 bytes	Upper byte = 0x00, lower byte = lower byte of Param2
1 byte	MacFlag = 0x01
5 bytes	0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	0x00 if Usage Counter value is not selected, or 0x00 if KeyID is VolatileKey.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

The Device MacCount will be changed to the EMacCount value when a Decrypt command is received with the Client Decryption mode is selected. The EMacCount will be used when decrypting the data and the MacCount will be incremented by the Decrypt operation. (After processing the command, the device MacCount will equal EMacCount plus one.)

I.16 EncRead Command MAC

The OutMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
1 byte	EncRead opcode (0x05)	
6 bytes	FirstBlock field containing	
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
5 bytes	0x00	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value, or 0x00 if not selected.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.17 EncRead Command Configuration Memory Signature MAC

The following example shows how the integrity MAC is calculated for a 512 byte (32 block) certification of the data from the Configuration Memory. This operation involves multiple passes through the AES crypto engine; all using the same key, KeyID 00. If the mode parameter indicates that there is an additional block of authenticate-only data, then another pass through the AES crypto engine is required.

There are 35 passes through the AES crypto engine in CBC mode to create the cleartext MAC. The inputs to the crypto engine for those blocks are labeled B0, B1, B2 …, and the outputs are B'0, B'1, B'2 …, respectively.

- **B0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of b0111 1001.
  - 12 byte Nonce – as generated by the Nonce command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte length field – always 0x00 00.
- **B1** is the XOR of B'0 with the following 128 bits:
  - 2 byte length field – value of 528 or 544.
  - 14 byte ManufacturingID – Opcode, etc.
- **B2** is the XOR of B'1 with the following 128 bits:
  - 16 bytes counter+serial+small, if mode indicates; otherwise, this block does not exist.
- **B3** is the XOR of B'2 with the following 128 bits:
  - First 16 bytes of Config – in the clear.
- **B4** is the XOR of B'3 with the following 128 bits:
  - Second 16 bytes of Config – in the clear.
  - … and so on …
- **B'34** is the clear text MAC which must be encrypted before being sent to the system.

There is one pass through the AES crypto engine in CTR mode to encrypt the MAC.

- **A0** is composed of the following 128 bits:
  - 1 byte flag – a fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 00`.
- **A'0** is XOR'd with the clear text MAC and sent to the system.

I.18 EncRead Command Key Memory Signature MAC

The following example shows how the integrity MAC is calculated for a 256 byte (16 block) certification of the data from the Key Memory. This operation involves multiple passes through the AES crypto engine; all using the same key, KeyID 00. If the mode parameter indicates that there is an additional block of authenticate-only data, then another pass through the AES crypto engine is required.

There are 19 passes through the AES crypto engine in CBC mode to create the cleartext MAC. The inputs to the crypto engine for those blocks are labeled B0, B1, B2 …, and the outputs are B'0, B'1, B'2 …, respectively.

- **B0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0111 1001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte length field – always `0x00 00`.
- **B1** is the XOR of B'0 with the following 128 bits:
  - 2 byte length field – value of 272 or 288.
  - 14 byte ManufacturingID – Opcode, etc.
- **B2** is the XOR of B'1 with the following 128 bits:
  - 16 bytes counter+serial+small, if mode indicates; otherwise, this block does not exist.
- **B3** is the XOR of B'2 with the following 128 bits:
  - First 16 bytes of config – in the clear.
- **B4** is the XOR of B'3 with the following 128 bits:
  - Second 16 bytes of config – in the clear.
  - … and so on …
- **B'18** is the clear text MAC which must be encrypted before being sent to the system.

There is one pass through the AES crypto engine in CTR mode to encrypt the MAC.

- **A0** is composed of the following 128 bits:
  - 1 byte flag – fixed value of `b0000 0001`.
  - 12 byte Nonce – as generated by the `Nonce` command.
  - 1 byte MacCount – one for first MAC generation.
  - 2 byte counter field – always `0x00 00`.
- **A'0** is XOR'd with the clear text MAC and sent to the system.

I.19 `Encrypt` Command MAC

The OutMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID
1 byte	Encrypt Opcode (`0x06`)
1 byte	Mode
2 bytes	Param1
2 bytes	Param2
1 byte	MacFlag
5 bytes	`0x00`

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value or `0x00` if not selected or if KeyID is VolatileKey.
8 bytes	SerialNum[0:7] or `0x00` if not selected.
4 bytes	SmallZone[0:3] or `0x00` if not selected.

I.20 `EncWrite` Command MAC

The InMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
1 byte	EncWrite Opcode (`0x05`)	
6 bytes	FirstBlock field containing:	
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
5 bytes	`0x00`	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value or `0x00` if not selected.
8 bytes	SerialNum[0:7] or `0x00` if not selected.
4 bytes	SmallZone[0:3] or `0x00` if not selected.

I.21 `INFO` Command

The `INFO` command does not perform a cryptographic operation, and does not use or generate a MAC.

I.22 `KeyCreate` Command MAC

The input and output MACs are both calculated using the parent key.

Both MACs are calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID		
1 byte	KeyCreate Opcode (`0x08`)		
6 bytes	FirstBlock field containing:		
		1 byte	Mode
		2 bytes	Param1
		2 bytes	Param2
		1 byte	MacFlag
5 bytes	`0x00`		

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

4 bytes	Usage counter value or `0x00` if not selected.
8 bytes	SerialNum[0:7] or `0x00` if not selected.
4 bytes	SmallZone[0:3] or `0x00` if not selected.

I.23 `KeyImport` Command — KeyCreate MAC

The MAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID		
1 byte	KeyCreate Opcode (`0x08`)		
6 bytes	FirstBlock field containing:		
		1 byte	Mode (If the target of the `KeyImport` command is a VolatileKey, replace bit 0 of the Mode with a one. Otherwise, replace bit 0 with a zero.)
		2 bytes	Param1
		2 bytes	Param2
		1 byte	MacFlag
5 bytes	`0x00`		

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculations:

16 bytes	SecondBlock field containing:		
		4 bytes	Usage counter value or `0x00` if not selected.
		8 bytes	SerialNum[0:7] or `0x00` if not selected.
		4 bytes	SmallZone[0:3] or `0x00` if not selected.

I.24 KeyLoad Command MAC

The InMAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
1 byte	KeyLoad opcode (0x09)	
6 bytes	FirstBlock field containing:	
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
5 bytes	0x00	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value, or 0x00 if not selected.
8 bytes	SerialNum[0:7], or 0x00 if not selected.
4 bytes	SmallZone[0:3], or 0x00 if not selected.

I.25 KeyTransfer Command

The KeyTransfer command does not perform a cryptographic operation and does not use or generate a MAC.

I.26 Legacy Command

The Legacy command executes a single block of the AES engine with no input or output formatting. This is known as ECB mode and can be used to perform various AES encryption and/or authentication operations. This command does not use the Nonce Register value in the computation since the entire 16 byte AES engine input value comes from the input packet.

I.27 Lock Command MAC

If required, due to the value of the mode parameter and ZoneConfig[UZ].WriteMode, the MAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID
1 byte	Lock Opcode (0x0D)
1 byte	Mode
2 bytes	Param1
2 bytes	Param2
1 byte	MacFlag
5 bytes	0x00

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

4 bytes	Usage counter value or 0x00 if not selected.
8 bytes	SerialNum[0:7] or 0x00 if not selected.
4 bytes	SmallZone[0:3] or 0x00 if not selected.

The AES key used for the MAC calculation is that specified in ZoneConfig[Zone].WriteID.

I.28 Nonce Command

If the Random Nonce option is selected, then the internal Random Nonce is generated using the following function:

Block A is:

1 byte	Nonce Opcode (0x01)
1 byte	Mode
2 bytes	0x00
12 bytes	Input Seed

Block B is:

2 bytes	ManufacturingID
2 bytes	0x00
12 bytes	Internally generated random number

AES is executed in ECB mode with an input value of Block A and a key of Block B. The output of the AES crypto engine is XOR'd with Block A, and the first 12 bytes of the result are stored in the internal Nonce Register.

If the LockConfig Register is unlocked (0x55), then the RNG is latched in test mode, and the Nonce command will generate nonrandom values. If the LockConfig Register is locked (0x00), then the RNG generates random numbers and the Nonce command functions normally.

I.29 NonceCompute Command

The random Nonce is generated using the following function:

Block A is:

1 byte	Nonce opcode (0x01)
1 byte	Mode
2 bytes	0x00
12 bytes	Nonce Register

Block B is:

2 bytes	ManufacturingID
2 bytes	0x00
12 bytes	Random Seed

AES is executed in ECB mode with an input value of Block A and a key of Block B. The output of the AES crypto engine is XOR'd with Block A, and the first 12 bytes of the result are stored in the internal Nonce Register.

I.30 Random Command

Generates a random number using the internal high-quality RNG and the random number generation procedure recommended by NIST in SP800-90 (see Appendix A, Standards and Reference Documents).

I.31 Reset Command

The Reset command does not perform a cryptographic operation and does not use or generate a MAC.

I.32 Sleep Command

The Sleep command does not perform a cryptographic operation and does not use or generate a MAC.

I.33 `WriteCompute` Command

The MAC is calculated using the following 14 bytes in the default authenticate-only block:

2 bytes	ManufacturingID	
1 byte	EncWrite Opcode (`0x05`)	
6 bytes	FirstBlock field containing:	
	1 byte	Mode
	2 bytes	Param1
	2 bytes	Param2
	1 byte	MacFlag
5 bytes	`0x00`	

If *any* of the optional authenticate fields are selected in the mode parameter, then a second authenticate-only block is included in the MAC calculation:

16 bytes	SecondBlock field containing:	
	4 bytes	Usage counter value or `0x00` if not selected.
	8 bytes	SerialNum[0:7] or `0x00` if not selected.
	4 bytes	SmallZone[0:3] or `0x00` if not selected.

Appendix J. I²C Interface

The ATAES132A 2-Wire Serial Interface is designed to interface directly to microcontrollers with I²C interface ports. The serial interface and cleartext Read/Write operations operate similar to those of the Atmel I²C Serial EEPROM.

The Host sends ATAES132A extended commands to the device by writing the command packet to the Command Memory Buffer at address `0xFE00`. The ATAES132A processes the command packet and places the response in the Response Memory Buffer. The Host retrieves the response by reading the response packet from address `0xFE00`.

See Appendix G.2, STATUS Register Behavior in the I²C Interface Mode for additional information regarding the ATAES132A behavior in I²C interface mode. See Appendix J.6, I²C Compatibility for I²C compatibility information.

J.1 I²C Serial Interface Description

When ATAES132A is configured in I²C serial communication mode, the serial interface operates as an I²C compatible standard-mode I²C slave device as described in this appendix. I²C is a synchronous serial interface protocol that is a *de facto* industry standard and is not formally documented or controlled. Multiple I²C devices can share the data bus; however, each I²C slave must have a unique I²C Device Address to prevent bus contention. SCK clock frequencies up to 1MHz are supported by the ATAES132A.

The serial interface communication mode is selected by programming the I2CAddr Register in the Configuration Memory as described in Appendix E.2.15, I2CAddr Register. The I²C Device Address is also located in the I2CAddr Register. The ATAES132A will only respond to I²C instructions that have a matching I²C Device Address.

J.1.1 I²C Master

The I²C master device generates the serial clock and sends instructions to the I²C slave devices. In this specification, the I²C master is usually referred to as the Host or the Host microcontroller.

J.1.2 I²C Slave

I²C slave devices receive the serial clock as an input and receive instructions from the I²C master. I²C slaves can never generate traffic on the I²C interface. Slaves can only respond to instructions provided by the I²C master. The ATAES132A always operates as a slave. In this specification, the slave is usually referred to as the Client or the device.

J.1.3 I²C Device Address

Each ATAES132A has a seven bit I²C Device Address (stored in the I2CAddr Register, as described in Appendix E.2.15) which is used by the Host to direct commands to a specific device on the I²C interface. I²C devices will only respond to instructions with a matching I²C Device Address. When the ATAES132A is in the Standby state or Sleep state, a matching I²C Device Address will cause the device to wake-up (see Appendix L, Power Management for power management specifications).

The LSB of the I²C Device Address byte is the Read/Write operation select bit. A Read operation is initiated if the R/W bit is high, and a Write operation is initiated if the R/W bit is low.

J.1.4 Relationship of Clock to Data

Data on the SDA pin may change only during SCK low time periods. Data changes during SCK high periods indicate an I²C Start or Stop condition. The SDA pin is pulled high by an external resistor when no devices are driving the I²C data bus. The timing requirements for the clock and data signals are illustrated in Appendix J.7, Timing Diagrams.

J.1.5 I²C Start Condition

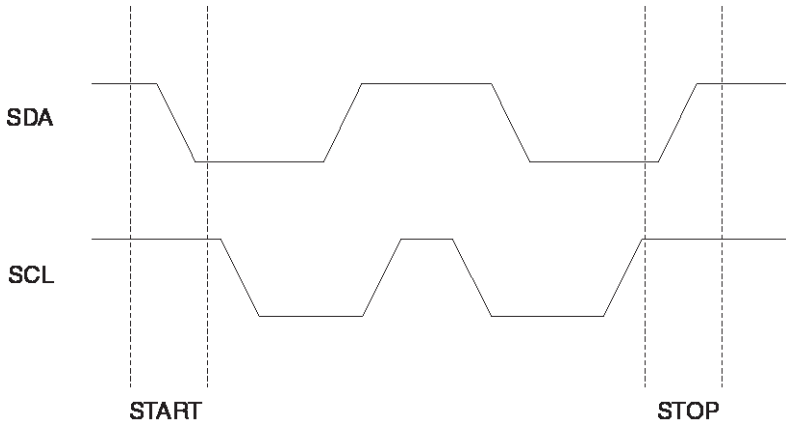
A high-to-low transition of SDA with SCK high is an I²C Start condition. An I²C Start condition must precede the I²C Device Address for any instruction. I²C Start conditions are generated only when the Host is driving the bus; slaves are not allowed to generate an I²C Start condition.

The slave will reset its serial interface immediately when an I²C Start condition is received. An I²C Start condition cannot be followed immediately with an I²C Stop condition. Figure J-1 illustrates an I²C Start condition.

J.1.6 I²C Stop Condition

A low-to-high transition of SDA with SCK high is an I²C Stop condition. I²C Stop conditions are only generated when the Host is driving the bus; slaves are not allowed to generate an I²C Stop condition. Figure J-1 illustrates an I²C Stop condition.

Figure J-1. I²C Start Condition and I²C Stop Condition Definitions



J.1.7 I²C ACK

All addresses and data words are serially transmitted to and from ATAES132A in 8-bit words. The receiving I²C device sends a zero (ACK) during the ninth clock cycle to acknowledge receipt of each byte.

An I²C Host can use acknowledge polling to monitor the progress of an EEPROM Write and to determine if the slave is ready to accept a new instruction. See Appendix J.3.7, Acknowledge Polling for a discussion of ACK polling.

J.1.8 I²C NAK

When the receiving I²C device fails to send a zero during the ninth clock cycle to acknowledge that it has received a byte, then SDA remains high due to the external pull-up resistor. This generates a NO ACK (NAK) signal to the device sending the byte.

J.1.9 Data Format

All instructions and data on the I²C bus must be formatted as 8-bit bytes, followed by a ninth bit (ACK or NAK) generated by the receiving device. The MSB is the first bit of each byte transmitted and received.

J.2 Pin Descriptions

When the ATAES132A is configured in the I^2C interface communication mode, the package pins are assigned the functionality described in this section.

Note: The pin numbers listed here are the SOIC and UDFN package pin numbers.

Table 9-8. I^2C Communication Mode Pin Descriptions

Pin	Name	Description
1	$\overline{CS}$	**SPI Chip Select Bar Input pin.** In the I^2C Communication mode, this pin is not used, and should be tied to V_{CC} or V_{SS}. The state of this pin does not affect the functionality or Active state power consumption of the ATAES132A when I^2C Communication mode is selected.
2	SO	**Serial Data Out pin.** In the I^2C Communication mode, this pin is not used in the default configuration. It is always in the high-impedance state. In this configuration, the pin can be tied to V_{CC} or V_{SS}. The state of this pin does not affect the functionality or Active state power consumption of the ATAES132A when I^2C Communication mode is selected. If Auth signaling is enabled, then the SO pin functions as the AuthO signal output. In this configuration, the AuthO signal is high after a specified key is authenticated. The AuthO output is in the high-impedance state when the device has not authenticated. (See Appendix J.5, I2C Auth Signaling).
3	NC	**No Connect pin.** This package pin is not used, and can be left open by the user. The state of this pin does not affect the functionality or power consumption of the ATAES132A.
4	V_{SS}	**Ground.**
5	SI/SDA	**Bidirectional Serial Data I/O pin.** In the I^2C communication mode, this pin functions as the Serial Data I/O (SDA). This pin is an open-drain buffer and may be wire-ORed with any number of other open-drain or open-collector devices. The SDA pin must be pulled high with an external resistor for the I^2C bus to operate correctly. Data on the SDA pin may change only during the SCK low time periods. Data changes during SCK high periods indicate an I^2C Start or Stop condition. Data transfer on the SDA line is half-duplex, as described by the I^2C command definitions in Appendix J.3, I2C Instruction Set; the Host and Client cannot simultaneously drive the SDA line.
6	SCK	**Serial Clock Input pin.** In the I^2C Communication mode, this pin is used as the Serial Interface Clock (SCK). The SCK input is used to transfer data into the ATAES132A on the rising edge of clock and to transfer data out on the falling edge of clock. The ATAES132A never drives SCK because it is a standard-mode I^2C slave device. Slave device clock stretching is not supported. The SCK line is high when the bus is idle. If the I^2C master uses a normal totem pole output to drive SCK, then no pull-up resistor is required on the SCK line. If the I^2C master uses an open-drain or open-collector output to drive SCK, then an external pull-up resistor is required.
7	NC	**No Connect pin.** This package pin is not used, and can be left open by the user. The state of this pin does not affect the functionality or power consumption of the ATAES132A.
8	V_{CC}	**Supply Voltage.** Power cannot be removed from the ATAES132A when the I^2C interface is active. The device may be permanently damaged if the requirements in Section 9.1, Absolute Maximum Ratings* and Section 9.3, DC Characteristics are exceeded.

J.3 I²C Instruction Set

The ATAES132A utilizes the Atmel AT24C32C Serial EEPROM instruction set. The ATAES132A I²C instruction set is shown in Table J-1.

Table J-1. ATAES132A I²C Instruction Set

Instruction Name	Operation	
BWRITE	Byte Write	Writes one byte to memory.
PWRITE	Page Write	Writes 2 to 32 bytes to memory.
READ	Read	Reads data from memory starting at the current address.
RREAD	Random Read	Reads data from memory starting at the specified address.
SREAD	Sequential Read	Reads additional data from memory.
SRESET	Software Reset	Resets the internal memory address counter to 0000h.

If ATAES132A receives an invalid or undefined instruction code, it will be ignored and the associated data bytes will be discarded. When any error occurs, the EERR bit of the STATUS Register is set to 1b to indicate an error. The Host can read the error code from the Response Memory Buffer at address 0xFE00 using the READ command.

J.3.1 Byte Write (BWRITE)

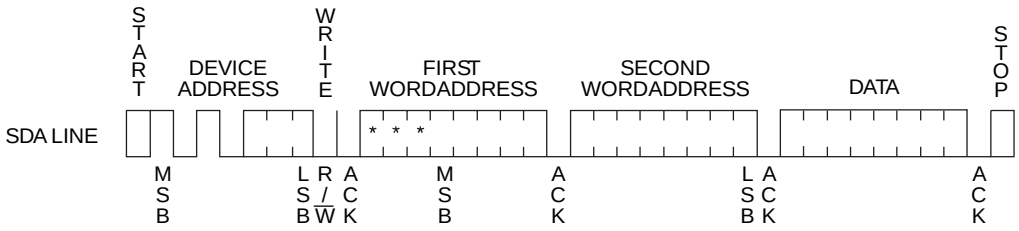
A Byte Write operation requires two 8-bit data word addresses following the I²C Device Address byte. Upon receipt of the Start condition and device address, the ATAES132A will respond with I²C ACK and then clock in the two address bytes (ACKing each byte). The ATAES132A will ACK the receipt of the data byte from the Host. The Host microcontroller must terminate the write sequence with a Stop condition to initiate the Write operation.

At this time, the EEPROM enters an internally-timed write cycle to the nonvolatile memory. All inputs are disabled during this write cycle, and the EEPROM will NAK the device address until the write is complete.

If the Host transmits an invalid address, the EEPROM will NAK the second address byte and any data bytes.

When any error occurs, the RRDY and EERR bits of the STATUS Register are set to 1b to indicate an error. The Host can read the error code from the Response Memory Buffer (address 0xFE00) using the RREAD command. If the command is processed without error, the EERR bit is set to 0b. Reading the Response Memory Buffer does not reset the error code or the STATUS Register.

Figure J-2. Byte Write



J.3.2 Page Write (PWRITE)

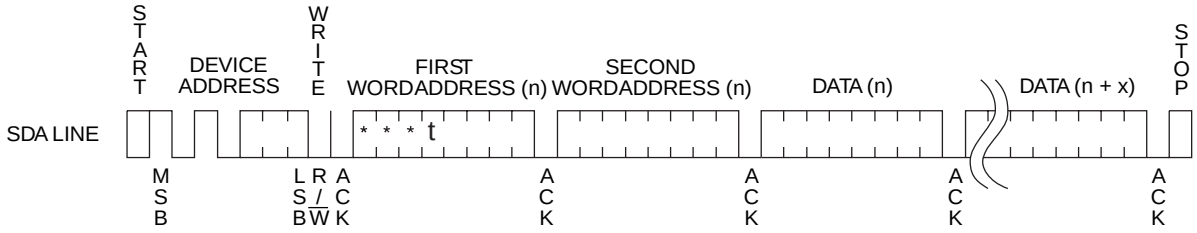
The ATAES132A is capable of 32-byte Page Writes. A Page Write is initiated in the same way as a Byte Write, but the Host microcontroller does not send a Stop condition after the first data byte is clocked in. Instead, after the device ACKs receipt of the first data byte, the Host microcontroller can transmit up to 31 more data bytes (each byte will be ACKed by the ATAES132A). The EEPROM will respond with an I^2C ACK after each data byte is received. The Host must terminate the Page Write sequence with a Stop condition. The data address is internally incremented following the receipt of each data byte.

If more than 32 bytes of data are transmitted or the page boundary is crossed, then no data will be written.

If the Host transmits an invalid word address, the EEPROM will NAK the second address byte and all data bytes.

When any error occurs, the RRDY and EERR bits of the STATUS Register are set to `1b` to indicate an error. The Host can read the error code from the Response Memory Buffer (address `0xFE00`) using the `RREAD` command. If the command is processed without error, the EERR bit is set to `0b`. Reading the Response Memory Buffer does not reset the error code or the STATUS Register.

Figure J-3. Page Write



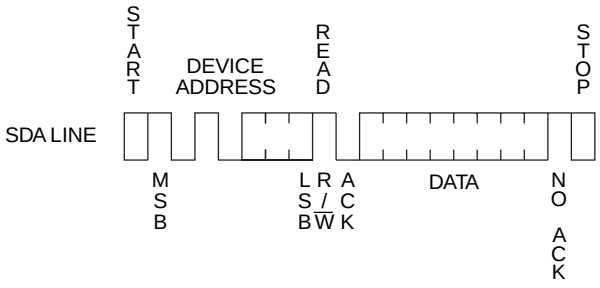
J.3.3 Current Address Read (READ)

The internal data byte address Counter maintains the last address accessed during the last Read or Write operation incremented by one. This address stays valid between operations as long as the device power is maintained.

To perform a Current Address Read, the Host sends the device address with the Read/Write Select bit set to one (READ), and this byte is ACKed by the EEPROM. Then, the Host clocks out the data byte located at the current address. After the byte is received, the Host responds with an I^2C NAK and a following Stop condition to terminate the Read operation.

When any error occurs, the EERR bit of the STATUS Register is set to `1b` to indicate an error. If the command is processed without error, the EERR bit is set to `0b`.

Figure J-4. Current Address Read of One Data Byte



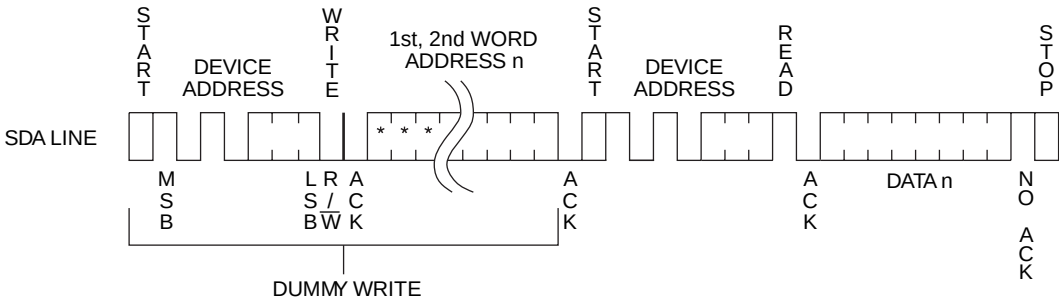
J.3.4 Random Read (RREAD)

A Random Read requires a dummy Byte Write sequence to load in the data byte address. Once the device address and data byte address are clocked in and acknowledged by the ATAES132A, the Host microcontroller must generate another Start condition. The microcontroller then initiates a Current Address Read by sending the device address with the Read/Write Select bit high (READ). The ATAES132A I^{2}C ACKs the device address, and serially clocks out the data byte. After the byte is received, the Host responds with an I^{2}C NAK and a following Stop condition to terminate the Read operation.

If the Host transmits an invalid word address, the EEPROM will NAK the second address byte.

When any error occurs, the EERR bit of the STATUS Register is set to `1b` to indicate an error. If the command is processed without error, the EERR bit is set to `0b`.

Figure J-5. Random Read



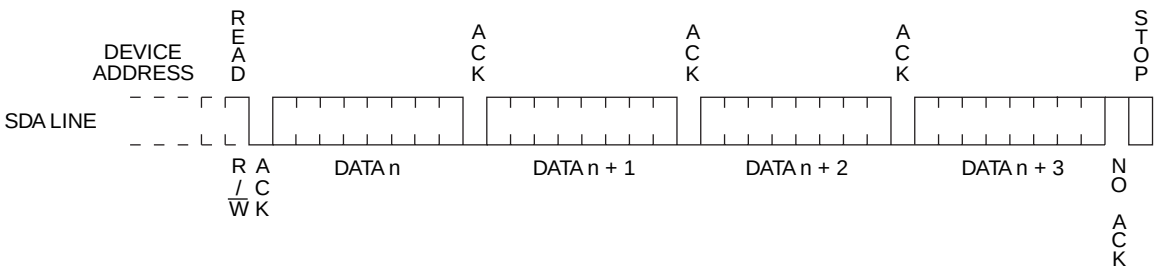
J.3.5 Sequential Read (SREAD)

Sequential Reads are initiated by either a Current Address Read or a Random Read. After the Host microcontroller receives a data byte, it responds with an I^{2}C ACK. As long as the EEPROM receives an acknowledge, it will continue to increment the data byte address and serially clock out sequential data bytes. The Sequential Read operation is terminated when the microcontroller responds with an I^{2}C NAK and a following Stop condition.

When any error occurs, the EERR bit of the STATUS Register is set to `1b` to indicate an error. If the command is processed without error, the EERR bit is set to `0b`.

Note: If an I^{2}C Read begins at an authorized address and continues into protected memory, the EERR bit will be set to `1b`. Attempting to read protected memory will result in `0xFF` data returned to the Host for each protected byte address.

Figure J-6. Sequential Read



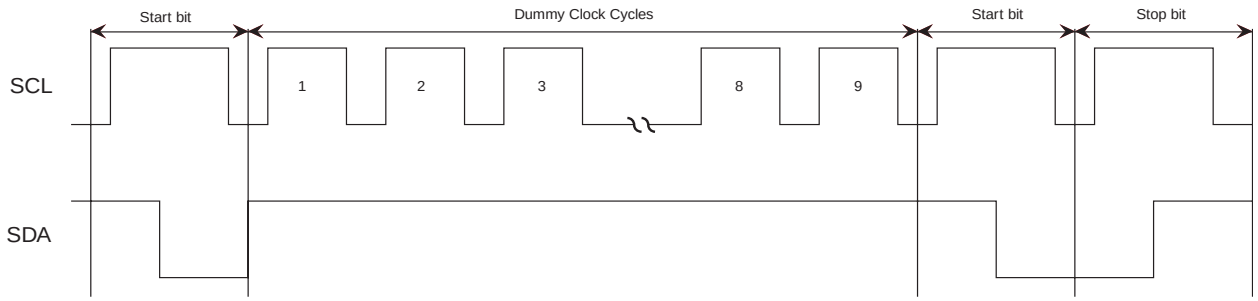
J.3.6 Software Reset (SRESET)

After an interruption in protocol, powerloss, or system reset, the ATAES132A in I^2C interface mode can be protocol reset by following these steps:

- Send a Start condition,
- Clock nine cycles,
- Send another Start condition followed by Stop condition, as shown below.

The device is ready for the next communication after these steps have been completed. The internal data address is also reset to `0000h` by this procedure.

Figure J-7. Software Reset



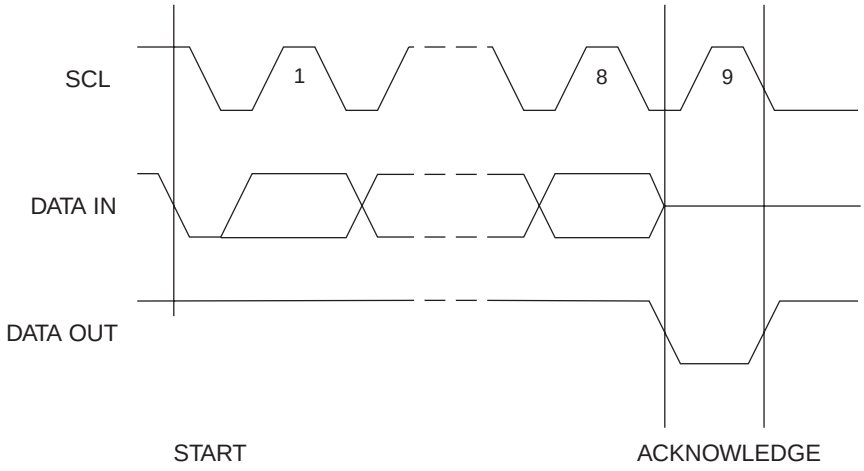
The ATAES132A requires that the clock be pulled low between the Start condition and the Stop condition at the end of the sequence, as illustrated in Figure J-7. It will not reset if this clock transition is omitted. See Appendix J.4, I^2C Interface Synchronization Procedure for detailed I^2C interface resynchronization instructions.

J.3.7 Acknowledge Polling

The Host can initiate Acknowledge (ACK) Polling immediately after a write command or the ATAES132A extended Crypto command is transmitted. Acknowledge polling involves sending a Start condition followed by the I^2C Device Address. The Read/Write bit of the I^2C Device Address is representative of the operation desired by the Host.

During an EEPROM Write operation, the ATAES132A will NAK the I^2C Device Address, indicating the device is busy. When the internal write cycle has completed, the ATAES132A will ACK the I^2C Device Address, allowing the Read or Write sequence to continue. The ATAES132A also NAKs during the processing of Crypto commands, and so Acknowledge Polling can also be used to determine when processing of the ATAES132A extended commands is complete.

Figure J-8. Output Acknowledge (I²C ACK)



J.4 I²C Interface Synchronization Procedure

If the Host and Client I²C interfaces lose synchronization for any reason, the Host should send clocks until SDA goes high followed by the `SRESET` command to reset the ATAES132A interface. See Appendix J.3.6, Software Reset (SRESET).

J.5 I²C Auth Signaling

The Auth signaling option allows an Authentication Signal (AuthO) to be output by ATAES132A. Auth signaling is available only in the I²C interface mode in standard plastic packages.

The Auth signaling option is controlled by two bits in the KeyConfig Registers: the KeyConfig[KeyID].AuthOut bit and the KeyConfig[KeyID].AuthOutHold bit (see Table J-2). By default, the KeyConfig[KeyID].AuthOut bit is `0b` for all keys disabling the Auth signaling option.

Table J-2. Auth Signaling KeyConfig Bit Functions

AuthOut Bit	AuthOutHold Bit	Operation
1b	X	First successful `Auth` command forces AuthO high. Additional `Auth` commands do not change AuthO and the AuthO output remains latched high.
0b	X	Successful or unsuccessful `Auth` commands cause no AuthO change.
X	1b	Authentication Reset does not change the AuthO output state.
X	0b	Authentication Reset forces AuthO to the high-impedance state.

If the KeyConfig[AKeyID].AuthOut bit is `1b` for the Authentication Key (AKeyID), then Auth signaling is enabled for that key and the AuthO signal is output on the SO pin. AuthO is latched high after a successful Inbound-Only Authentication or Mutual Authentication using the `Auth` command (see Section 7.1, `Auth` Command). AuthO will remain high until the device is powered off, unless an Authentication Reset is received.

If the KeyConfig[AKeyID].AuthOutHold bit is `0b` for the key (AKeyID) used to execute an Authentication Reset, then the AuthO signal latch will be latched in the high-impedance state when the command is received (with a correct Checksum). If the KeyConfig[AKeyID].AuthOutHold bit is `1b`, then AuthO will be unchanged by execution of an Authentication Reset sequence.

An Authentication Reset is an `Auth` command with Mode bits 0 and 1 set to `00b`. Knowledge of the key value is not required to execute an Authentication Reset (see Section 7.1). The ATAES132A does not memorize the KeyID used to activate Auth signaling. Each `Auth` command is processed using the KeyConfig[AKeyID] bits of the AKeyID in the command packet.

Auth signaling is not a security feature. The AuthO signal does not reflect the real-time state of the AuthComplete status flag. The `Reset` command, the `Sleep` command, and the Tamper detectors will not change the state of AuthO. The state of the AuthO latch is determined only by success or failure of the `Auth` command and the configuration of the KeyConfig bits. The `INFO` command should be used to determine the authentication status of the device (see Section 7.12, INFO Command).

The KeyConfig[AKeyID].AuthOut bit and the KeyConfig[AKeyID].AuthOutHold bit are ignored when the ATAES132A is configured in SPI Interface mode.

J.5.1 Using the AuthO Output

When Auth signaling is enabled, the AuthO signal output is either a Logic high or in the high-impedance state. AuthO can be used to drive an LED or as a control signal to other circuitry. When AuthO is used as a control signal, a pull-down resistor should be used to transform the high-impedance state into a logic low.

J.6 I²C Compatibility

The ATAES132A is design to operate on a bus with other I²C-compatible devices. ATAES132A is a standard-mode Client device capable of operating at clock speeds up to 1MHz (with bus timing scaled accordingly). The ATAES132A is not a Fast-Mode or High-Speed mode device.

This section lists the I²C options or features that are *not* supported by the ATAES132A. Any feature that differs from the I²C specification is also listed.

- ATAES132A does *not* perform Client clock stretching.
- ATAES132A will *not* respond to an I²C general call command.
- ATAES132A may be damaged if the clock or data signal levels are above V_{CC}. The power supply to the ATAES132A *cannot* be switched off while the bus is active. All of the voltage limits in Section 9.1, Absolute Maximum Ratings* must be respected.
- ATAES132A inputs include Schmitt Triggers and spike suppression; however, the outputs do not include falling edge slope control.
- On I²C devices, a Start condition followed immediately by a Stop condition is never permitted. On the ATAES132A, this sequence is permitted only as part of the `SRESET` command sequence (see Appendix J.3.6, Software Reset (SRESET)).

J.7 Timing Diagrams

Figure J-9. I²C Synchronous Data Timing (see Section 9.4.1 for PC Timing Specifications)

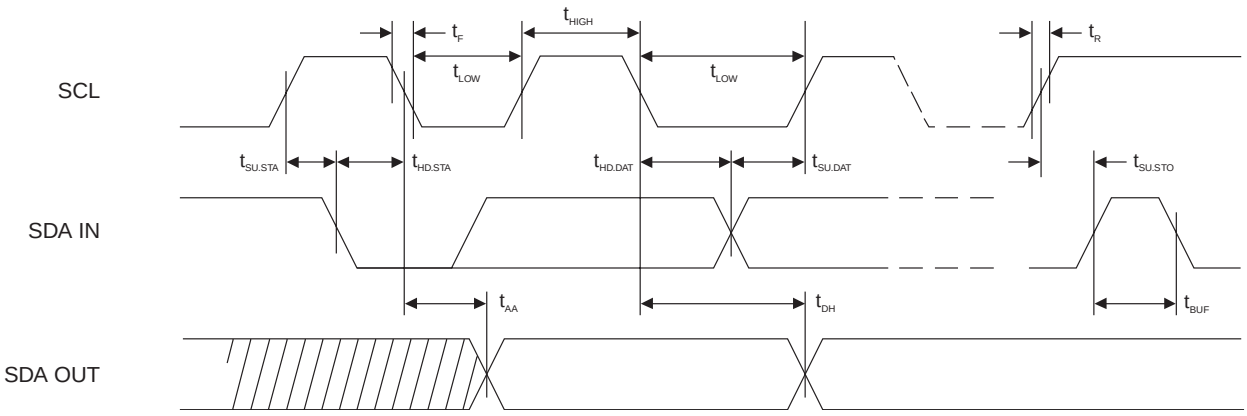
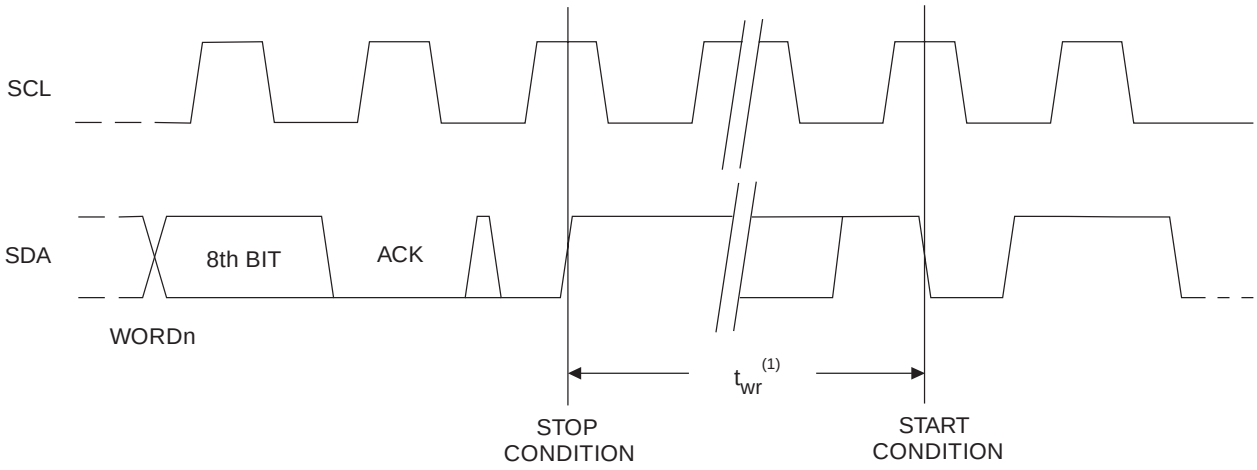


Figure J-10. I²C Write Cycle Timing



Appendix K. SPI Interface

The ATAES132A Serial Peripheral Interface (SPI) is designed to interface directly to microcontrollers using SPI Mode 0 or Mode 3. I/O and Cleartext Read/Write operations operate similarly to those of the Atmel SPI Serial EEPROM.

The Host sends ATAES132A commands to the device by writing the command packet to the Command Memory Buffer at address `0xFE00`. The ATAES132A processes the command packet and places the response in the Response Memory Buffer. The Host retrieves the response by reading the response packet from address `0xFE00`.

See Appendix G.3, STATUS Register Behavior in the SPI Interface Mode for additional information on the ATAES132A behavior in SPI interface mode.

K.1 SPI Serial Interface Description

When ATAES132A is configured in the SPI communication mode, the serial interface operates as a Mode 0 and Mode 3 slave device as described in this appendix. Serial Peripheral Interface (SPI) is a synchronous serial interface protocol that is a *de facto* industry standard and is not formally documented or controlled. Multiple SPI devices can share the data bus; however, each SPI slave must have a separate $\overline{CS}$ control line to prevent bus contention.

The serial interface communication mode is selected by programming the I2CAddr Register in the Configuration Memory as described in Section E.2.15.

K.1.1 SPI Master

The SPI bus master device generates the serial clock and sends instructions to the SPI slave devices. In this specification, the bus master is usually referred to as the Host or the Host microcontroller.

K.1.2 SPI Slave

SPI slave devices receive the serial clock as an input and receive instructions from the bus master. SPI slaves can never generate traffic on the SPI bus, and slaves can only respond to instructions provided by the bus master. The ATAES132A always operates as a slave. In this specification the slave is usually referred to as the Client.

K.1.3 Relationship of Clock to Data

The ATAES132A supports two of the four standard SPI interface modes; Mode 0 and Mode 3.

- In Mode 0:
  - The default state of SCK is low.
  - The data is clocked in (SI) on the rising edge of the clock.
  - Data out (SO) changes on the falling edge of the clock.
- In Mode 3:
  - The default state of SCK is high.
  - The data is clocked in (SI) on the rising edge of the clock.
  - Data out (SO) changes on the falling edge of the clock.

K.1.4 SPI Instruction Code

Each SPI command begins with the SPI master bringing the $\overline{CS}$ input low to select the device followed by transmission of an eight bit SPI instruction code to the SI input of the SPI slave. Following the instruction code, additional bytes may be clocked into SI or out of SO as required by the SPI command (see Appendix K.3, SPI Instruction Set). When the exchange of data bytes related to the SPI instruction code is complete, the $\overline{CS}$ input is brought high to deactivate the SPI slave interface.

If an invalid instruction code is received, then the ATAES132A will ignore any data received on the Data Input pin (SI), and the Data Output pin (SO) will remain in a high-impedance state.

K.1.5 Data Format

All instructions and data on the SPI bus must be formatted as eight bit bytes. The Most-Significant bit (MSB) is the first bit of each byte transmitted and received.

K.2 SPI Communication Mode Pin Descriptions

When ATAES132A is configured in SPI communication mode, the package pins are assigned the functionality described in this section.

Table K-1. Pin Descriptions

Pin	Name	Description
1	$\overline{CS}$	**SPI Chip Select Bar Input pin.** In SPI communication mode, this pin functions as the slave select input. The ATAES132A is selected when the $\overline{CS}$ pin is low, allowing instructions and data to be accepted on the Serial Data Input pin (SI), and allowing data to be transmitted on the Serial Data Output pin (SO). When the device is not selected, data will not be accepted via the SI pin, and the Serial Output pin (SO) will remain in a high-impedance state. When the ATAES132A is in the Standby state or Sleep state, a high-to-low transition on the $\overline{CS}$ pin will cause the device to wake-up (see Appendix L, Power Management for power management specifications). It is recommended that the ($\overline{CS}$) pin be connected to V_{CC} with a pull-up resistor so that the $\overline{CS}$ pin follows V_{CC} during power-up and power-down.
2	SO	**Serial Data Out pin.** In the SPI communication mode, this pin functions as the Serial Data output. When the $\overline{CS}$ pin is high, the SO pin will always be in a high-impedance state because the SPI interface is disabled.
3	NC	**No Connect pin.** This package pin is not used, and can be left open by the user. The state of this pin does not affect the functionality or power consumption of the ATAES132A.
4	V_{SS}	**Ground.**
5	SI/SDA	**Serial Data Input pin.** In the SPI communication mode, this pin functions as the serial data input. When the $\overline{CS}$ pin is high, the SI pin will not accepted data because the SPI interface is disabled.
6	SCK	**Serial Clock Input pin.** In the SPI communication mode, this pin is used as the serial interface clock. All data on the SI and SO pins is synchronized by SCK, as described in Appendix K.1.3, Relationship of Clock to Data.
7	NC	**No Connect pin.** This package pin is not used, and can be left open by the user. The state of this pin does not affect the functionality or power consumption of the ATAES132A.
8	V_{CC}	**Supply Voltage.** Power cannot be removed from the ATAES132A when the SPI bus is active. The device may be permanently damaged if the requirements in Section 9.1, Absolute Maximum Ratings* and Section 9.3, DC Characteristics are exceeded.

K.3 SPI Instruction Set

ATAES132A utilizes an 8-bit SPI instruction register. The SPI instruction set is listed in Table K-2.

Table K-2. ATAES132A SPI Instruction Set

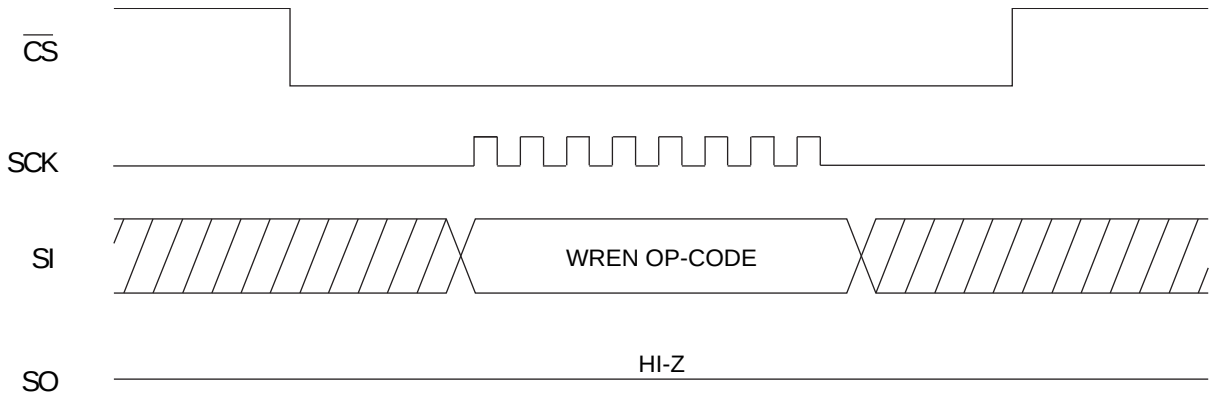
Instruction Name	Instruction Code	Operation
WRITE	0000 0010 b	Write data to memory.
READ	0000 0011 b	Read data from memory.
WRDI	0000 0100 b	Reset Write Enable Register
RDSR	0000 0101 b	Read Status Register
WREN	0000 0110 b	Set Write Enable Latch

If the ATAES132A receives an invalid instruction code or an invalid memory address, then no response will be sent; the SO output will remain in the high-impedance state. When any error occurs, the EERR bit of the STATUS Register is set to 1b to indicate an error. The Host can read the error code from the Response Memory Buffer at address 0xFE00 using the READ command. Reading the Response Memory Buffer does not reset the error code or change the STATUS.

K.3.1 Write Enable Command (WREN)

The device will power-up in the Write Disable state when V_{CC} is applied. All EEPROM Write instructions must therefore be preceded by a Write Enable instruction. It is not necessary to send the Write Enable instruction prior to sending command packets to the Command Memory Buffer.

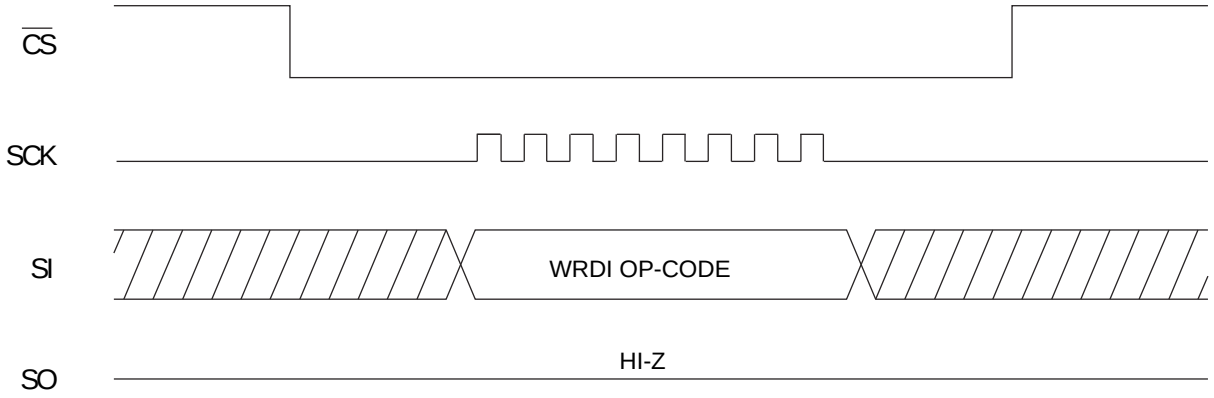
Figure K-1. SPI Write Enable (WREN) Timing



K.3.2 Write Disable Command (WRDI)

The Write Enable flag can be disabled by sending the Write Disable instruction.

Figure K-2. SPI Write Disable (WRDI) Timing



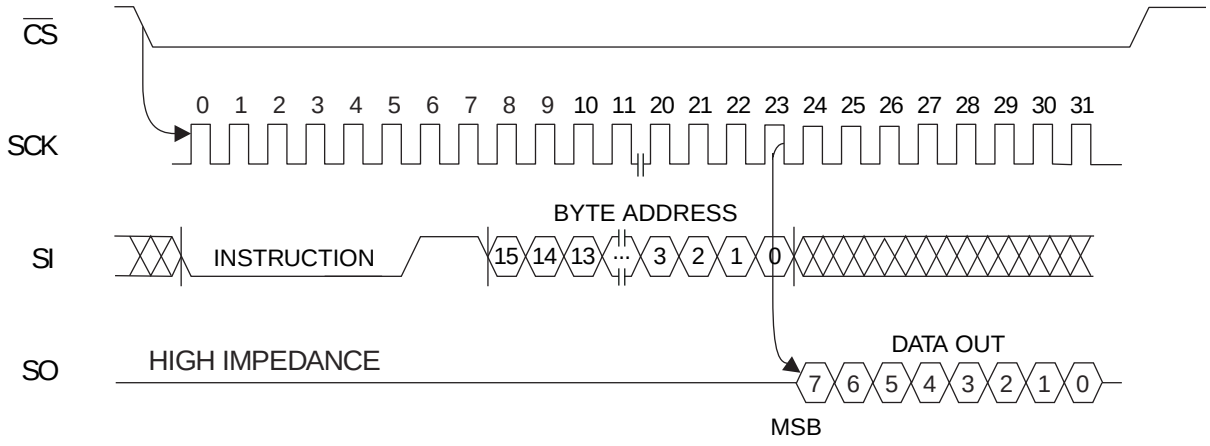
K.3.3 Read Memory Command (READ)

Reading data from the ATAES132A requires the following sequence:

1. The Host drives the $\overline{CS}$ line low to select a device,
2. Then transmits the Read instruction code on the SI line,
3. Then followed by the address of the byte to be read.
4. The Client ignores any data on the SI line that follows a Read Memory instruction.

The Client shifts out the data at the specified address on the SO line. If only one byte is to be read, the $\overline{CS}$ line must be driven high after the data byte comes out. If multiple bytes are to be read, the Host can sequentially clock the data out of the ATAES132A since the byte address is automatically incremented. The $\overline{CS}$ line must be driven high by the Host after the last data byte is read. If the highest address is reached, the Address Counter will not roll over.

Figure K-3. SPI READ Memory Timing



When any error occurs, the EERR bit of the STATUS Register is set to `1b` to indicate an error. If the command is processed without error, the EERR bit is set to `0b`.

Note: If an SPI Read begins at an authorized address but continues into protected memory; the EERR bit will be set to `1b`.

K.3.4 Write Memory Command (WRITE)

In order to write to the ATAES132A, two separate instructions must be executed. First, the device must be write enabled via the Write Enable (WREN) instruction. Then a Write Memory instruction may be executed. All commands received while a write cycle is in progress will be ignored, except the Read Status Register (RDSR) instruction.

A Write Memory command requires the following sequence:

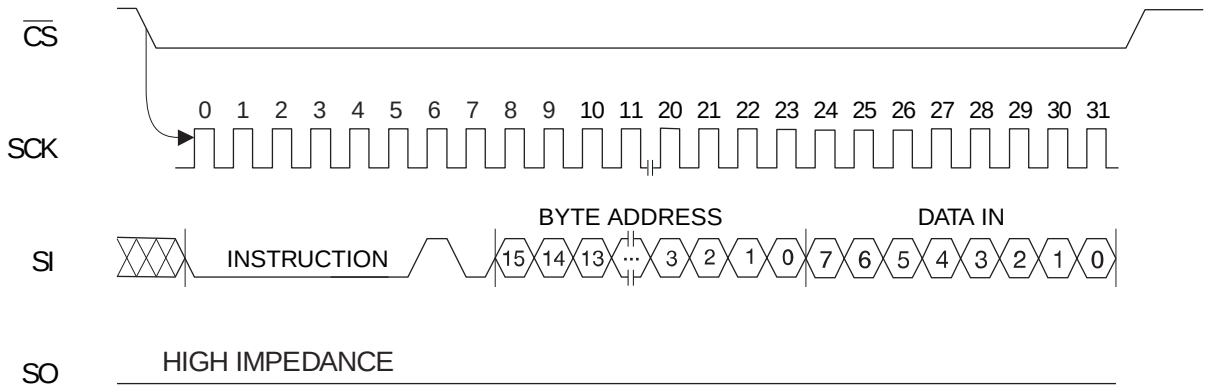
1. The Host drives the $\overline{CS}$ line low to select a device,
2. Then transmits the Write instruction code on the SI line,
3. Then followed by the address of the byte to write and the 1 to 32 data bytes to be written.

The byte address is automatically incremented as each byte is clocked in. The $\overline{CS}$ line must be driven high by the Host during the SCK low time immediately after clocking in the last data bit. The low-to-high transition of the $\overline{CS}$ pin initiates the EEPROM Write process. The SO pin remains in the high-impedance state during the entire Write sequence.

The Ready/Busy Status of the device can be determined by initiating a Read Status Register (RDSR) instruction. If the WIP status bit is `1b`, the write cycle is still in progress. If the WIP Status bit is `0b`, the write cycle has ended, and the ATAES132A is ready to accept a new command. Only the Read Status Register (RDSR) instruction is enabled during the EEPROM Write cycle.

The ATAES132A is capable of a 32-byte Page Write operation. After each byte of data is received, the data address is internally incremented by one. If more than 32 bytes of data are transmitted or if the page boundary is crossed, then no data will be written. The ATAES132A is automatically returned to the write disable state at the completion of a write cycle.

Figure K-4. SPI Write Memory Timing



When any error occurs, the RRDY and EERR bits of the STATUS Register are set to `1b` to indicate an error. The Host can read the error code from the Response Memory Buffer (address `0xFE00`) using the `READ` command. If the command is processed without error, the EERR bit is set to `0b`. Reading the Response Memory Buffer does not reset the error code or the STATUS Register.

If the device is not Write Enabled (WREN), the device will ignore the Write instruction and will return to the waiting for a command. A new $\overline{CS}$ falling edge is required prior to the new instruction code.

K.3.5 Read Status Register Command (RDSR)

The Read Status Register instruction provides access to the STATUS Register. The Ready/Busy status of the device can be determined using the RDSR instruction. Alternately, the STATUS Register can be read directly from memory, as described in Appendix G.2.4, Read STATUS Register.

If the ATAES132A is performing an EEPROM Memory Write or is processing a command when the STATUS read is performed, then all eight bits are ones if the `RDSR` command is used to read the STATUS Register, emulating the behavior of Atmel Serial EEPROM. See Appendix G, Understanding the STATUS Register for a detailed description of the STATUS Register bits and Status bit behavior.

Table K-3. Device Status Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EERR	RRDY	Reserved	CRCE	Reserved	WAKEb	WEN	WIP

The Device Status Register can always be read even if the the ATAES132A is processing a command or writing the EEPROM. The SPI `RDSR` command is the preferred method for reading the STATUS in SPI interface mode.

If the ATAES132A is in the Sleep or Standby power state, reading the STATUS Register forces the ATAES132A to wake-up; the STATUS Register is `0xFF` until the wake-up process is complete.

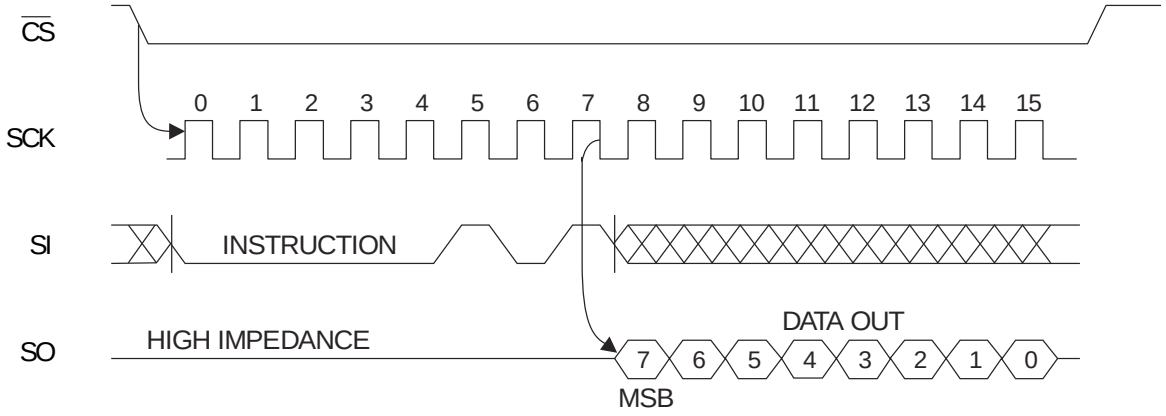
Table K-4. Read Status Register Bit Definition Using the SPI `RDSR` Command(1)(2)

Bit	Definition
Bit 0 (WIP)	`0b` =Device is ready, waiting for a command. `1b` =Write cycle or a cryptographic operation is in progress.
Bit 1 (WEN)	`0b` =Device is not SPI Write enabled. `1b` =Device is SPI Write enabled.
Bit 2 (WAKEb)	`0b` =Device is not in the Sleep or Standby power state. `1b` =Device is in the Sleep or Standby power state.
Bit 3 (Reserved)	Always `0b`. This bit is reserved for future use.(1)
Bit 4 (CRCE)	`0b` =The most recent command block contained a correct Checksum (CRC). `1b` =The most recent command block contained an error.
Bit 5 (Reserved)	Always `0b`. This bit is reserved for future use.(1)
Bit 6 (RRDY)	`0b` =Response Memory Buffer is empty. `1b` =Response Memory Buffer is ready to read.
Bit 7 (EERR)	`0b` =Most recent command did not generate an error during execution. `1b` =Most recent command generated an execution error.

Notes: 1. When the SPI `RDSR` command is used to read the STATUS Register during an EEPROM Write or during execution of any ATAES132A command, then status bits 0 to 7 are `1b`. The reserved bits will read as 0b if the STATUS Register is read directly from memory during an EEPROM Write or during execution of an ATAES132A command.

2. STATUS Register bits 0 to 7 are `1b` during wake-up and power-up. See for Appendix L, Power Management additional information.

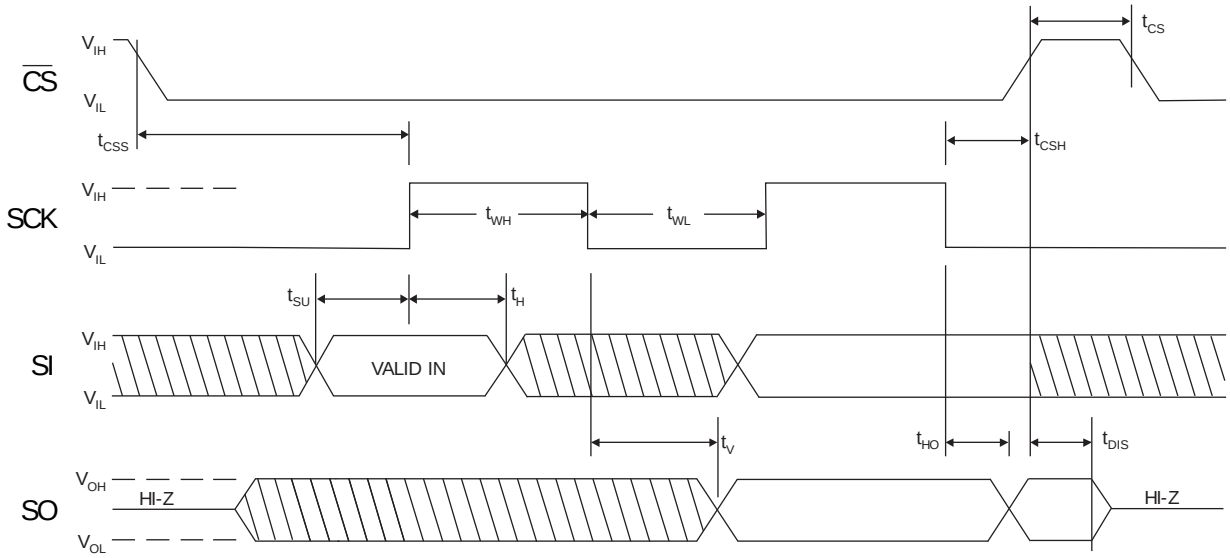
Figure K-5. SPI Read Status Register (RDSR) Timing



Reading the STATUS Register does not change the contents STATUS Register or the contents of the Response Memory Buffer.

K.4 Timing Diagram

Figure K-6. SPI Synchronous Data Timing (see Section 9.4.3, SPI Interface Timing)



Appendix L. Power Management

The ATAES132A contains several features that facilitate power management. This appendix describes the various power states and features.

L.1 Power State Descriptions

The ATAES132A has three powered states and the Off state. Two low-power states are available to reduce power consumption when the system is not using the ATAES132A.

L.1.1 Active State

The ATAES132A is in the Active state after it has completed the power-up process and is fully powered. The WIP Status bit is 0b when the ATAES132A is in the Active state and waiting for a command. The WIP Status bit is `1b` when the ATAES132A is in the Active state and processing a command or performing an EEPROM Write. (See Appendix G.1.1, WIP Status Bit [0] for WIP status bit information)

The supply current of the ATAES132A in the Active state is several milliamps (see Section 9.3.1, Supply Characteristics for I_{CC} specifications).

An ATAES132A in the Active state is capable of accepting a command immediately if the WIP Status bit is 0b. The I^2C timing specifications for the Active state are in Section 9.4.1, Power-Up, Sleep, Standby, and Wake-Up Timing. The SPI timing specifications for the Active state are in Section 9.4.3, SPI Interface Timing.

L.1.2 Standby State

ATAES132A can enter the Standby state in two ways:

- The Host can send a Sleep command to place the ATAES132A into Standby, or
- The ATAES132A will automatically enter the Standby state at power-up if configured to do so (see Appendix L.2.1, Power-Up). The Standby state preserves the ATAES132A volatile memory contents and the security state.

All eight status bits are `1b` when the ATAES132A is in the Standby state and during the wake-up process (see Appendix G, Understanding the STATUS Register for Status bit information).

The supply current of ATAES132A in the Standby state is several microamperes (see Section 9.3.1 for I_{SB} specifications).

An ATAES132A in the Standby state is capable of reporting the device status immediately, but cannot accept a command until the wake-up process is complete. The timing specifications for exiting the Standby state are in Section 9.4.1, Power-Up, Sleep, Standby, and Wake-Up Timing.

L.1.3 Sleep State

The ATAES132A can enter the Sleep state in two ways:

- The Host can send a `Sleep` command to place the ATAES132A into Standby, or
- The ATAES132A will automatically enter the Sleep state at power-up if configured to do so (see Appendix L.2.1).

The Sleep state clears the ATAES132A volatile memory contents and the security state.

All eight Status bits are `1b` when the ATAES132A is in the Sleep state and during the wake-up process (see Appendix G for Status bit information).

The supply current of the ATAES132A in the Standby state is less than one microampere (see Section 9.3.1 for I_{SB} specifications).

An ATAES132A in the Sleep state is capable of reporting the device STATUS immediately but cannot accept a command until the wake-up process is complete. The timing specifications for exiting the Sleep state are in Section 9.4.1, Power-Up, Sleep, Standby, and Wake-Up Timing.

L.1.4 Off State

When the ATAES132A device is unpowered or when V_{CC} is significantly below the minimum V_{CC} voltage, the device is in the Off state. A device in the Off state cannot respond to any commands.

L.2 Power State Transitions

Power-Up is a transition from the Off state to one of the three powered states. Power-down is the transition from a powered state to the Off state. Wake-up is the transition from one of the two low-power states to the Active state.

L.2.1 Power-Up

Power-Up begins when the power supply is turned on, causing the V_{CC} voltage to rise continuously from V_{SS} to the operating voltage. Power-Up occurs in three stages.

1. **First Stage:** The voltage regulator and other analog circuitry are activated.
2. **Second Stage:** The serial interface logic is activated so that the ATAES132A can report the device status to the Host.
3. **Third Stage:** The ATAES132A enters the state specified by the ChipConfig Register.

During the power-up process, the device is unable to accept commands. In the SPI interface mode, the device is ready to receive a Read Status Register command after the Power-Up Time, $t_{PU.STATUS}$. The Power-Up Ready Time ($t_{PU.RDY}$) specifies the time required to complete the power-up process. In the I^2C interface mode, the device will NAK all instructions prior to the completion of Power-Up (time $t_{PU.RDY}$).

The last stage of the power-up procedure is to enter the Active, Standby, or Sleep state specified by bits 6 and 7 of the ChipConfig Register. The ChipState Register is set to `0xFFFF` at power-up (see Appendix L.3, Understanding the ChipState Register).

Table L-1. Coding of the ChipConfig.PowerUpState bits in the ChipConfig Register

Bit 7	Bit 6	Description
1	1	Device goes to the Active state at power-up.
1	0	
0	1	Device goes to the Standby state at power-up.
0	0	Device goes to the Sleep state at power-up.

During power-up, the SPI Chip Select should follow the V_{CC} voltage. It is recommended that the $\overline{CS}$ pin be connected to V_{CC} with a pull-up resistor if the ATAES132A is configured in the SPI interface mode. The ATAES132A does not support hot swapping or hot plugging. Connecting or disconnecting this device to a system while power is energized can cause permanent damage to the ATAES132A.

L.2.2 Power-Down

Before power-down, the device must be deselected (if configured for SPI) and placed in the Active, Standby, or Sleep state. During power-down, the SPI Chip Select should be allowed to follow the V_{CC} voltage if the ATAES132A is configured in SPI interface mode.

The ATAES132A should not be powered down when the WIP status bit indicates that an EEPROM Write or cryptographic operation is in progress. If the WIP status bit is `0b`, then it is safe to power-down the device.

L.2.3 Entering the Standby State

If the ATAES132A is in the Active state, the Host can send a `Sleep` command to place the ATAES132A in the Standby state (see Section 7.23, `Sleep` Command). It is not possible to transition the device directly from the Sleep state to the Standby state. The Host must wake-up the device and then send a `Sleep` command to place the device in standby.

The device can also be configured to enter the Standby state at power-up as described in Appendix L.2.1, Power-Up.

The ATAES132A exits Standby state only if a Wake-Up event occurs on the I/O pins. Wake-Up is discussed in Appendix L.2.5, SPI Wake-Up and L.2.6, I^2C Wake-Up. The ChipState Register does not change when the ATAES132A enters or leaves the Standby state (see see Appendix L.3, Understanding the ChipState Register).

L.2.4 Entering the Sleep State

If the ATAES132A is in the Active state, the Host can send a `Sleep` command to place the ATAES132A in the Sleep state (see Section 7.23). It is not possible to transition the device directly from the Standby state to the Sleep state. The Host must wake-up the device and then send a `Sleep` command to place the device in the Sleep state.

The device can also be configured to enter the Sleep state at power-up, as described in Section L.2.1.

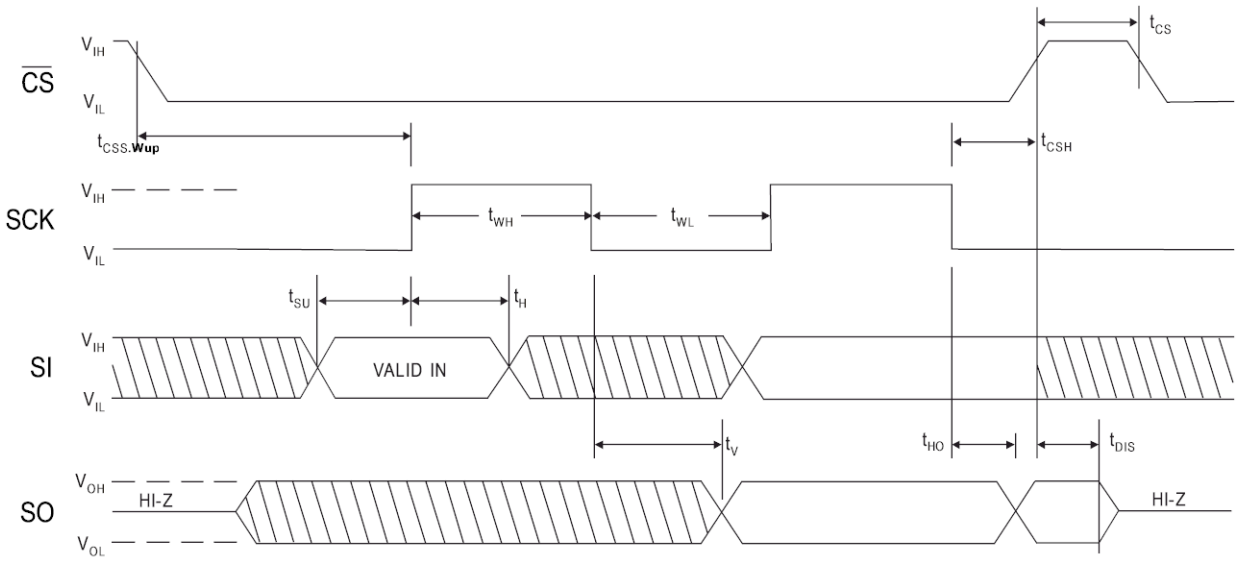
The ATAES132A exits Sleep mode only if a Wake-Up event occurs on the I/O pins. Wake-up is discussed in Sections L.2.5 and L.2.6. The ChipState Register changes to `0x5555` when the ATAES132A leaves the Sleep state (see Appendix Appendix L.3).

L.2.5 SPI Wake-Up

To wake-up the ATAES132A configured for SPI interface mode, the Host is required to read the Status Register using the SPI Read Status Register command. The ATAES132A will answer the SPI Read Status Register command with the device status if the Host has not violated the 100ns minimum $t_{CSS.Wup}$ setup time requirement. The ATAES132A status will indicate the device is busy (status = `0xFF`) during wake-up. When wake-up is complete, the ATAES132A status changes to indicate the device is in the Active state.

The ATAES132A will accept the SPI Read Status Register command only while it is busy. All other commands will be ignored. The SPI Read Status Register command is described in Appendix K.3.5, Read Status Register Command (RDSR).

Figure L-1. SPI Interface Timing, $\overline{CS}$ Setup Time at Wake-Up



The wake-up process begins when a device in the Standby or Sleep state experiences a high-to-low transition of the $\overline{CS}$ pin. The device is ready to receive a `Read Status Register` command from the Host after Wake-Up Time $t_{WupSB.STATUS}$ for the Standby state, or $t_{WupSL.STATUS}$ for the Sleep state. The wake-up is complete after the Wake-Up Ready Time of $t_{WupSB.RDY}$ for the Standby state or $t_{WupSL.RDY}$ for the Sleep state; $t_{WupSB.RDY}$ and $t_{WupSL.RDY}$ begin when the $\overline{CS}$ pin high-to-low transition occurs and end when the device enters the Active state. The Wake-Up timing specifications are in Table 9-5.

L.2.6 I²C Wake-Up

To wake-up an ATAES132A configured for I²C interface mode, the Host is required to perform ACK polling using the matching I²C Device Address. The ATAES132A will answer the ACK poll with an I²C NAK to indicate the device is busy during wake-up. The ACK poll reply will change to ACK when the device is in the Active state.

The ATAES132A will not accept any commands while it is busy. The ATAES132A will NAK the I²C Device Address if it does not match the internal I²C Device Address and will not wake-up if a nonmatching I²C Device Address is received.

The wake-up process begins when a device in the Standby or Sleep state receives an I²C start signal, followed immediately by an I²C Device Address that matches the ATAES132A I2CAddr Register. The device is ready to receive an ACK poll from the Host after Wake-Up Time $t_{WupSB.STATUS}$ for the Standby state or $t_{WupSL.STATUS}$ for the Sleep state. The wake-up is complete after the Wake-Up Ready Time of $t_{WupSB.RDY}$ for the Standby state or $t_{WupSL.RDY}$ for the Sleep state; $t_{WupSB.RDY}$ and $t_{WupSL.RDY}$ begin when a matching I2C Address is received, and end when the device enters the Active state. The wake-up timing specifications are in Table 9-5.

L.3 Understanding the ChipState Register

The INFO command (see Section 7.12, INFO Command) provides access to the ChipState Register. The ChipState Register value indicates if the device has recently experienced a power-up event or wake-up from the Sleep Power state. This information can be useful for determining how to recover from an unexpected transaction error.

Table L-2. Description of the ChipState Register Value Returned by the INFO command

ChipState	Description
0x0000	ChipState = Active. Device has remained active since the previous Crypto command was processed.(1)
0x5555	ChipState = Wake-up from sleep. Device has experienced a wake-up from the Sleep Power state since the previous Crypto command was processed.(1)
0xFFFF	ChipState = Power-up. Device has experienced a power up event since the previous Crypto command was processed.(1)

Note: 1. The following subsections describe the events that cause ChipState to change values and events that do not change ChipState.

L.3.1 ChipState = Power-Up

The following events cause the ChipState Register to be set to the Power-Up state (0xFFFF). The events in this table cause the device to be initialized and placed in the power state specified in the ChipConfig Register (see Appendix L.2.1, Power-Up).

Table L-3. Description of Events Causing the ChipState Register to be Set to 0xFFFF

Event	Event description
Power-up	Power-up of the device (Appendix L.2.1, Power-Up).
Power Interruption	Power interruption or brownout resulting in device reset.

L.3.2 ChipState = Wake-Up from Sleep

The following events cause the ChipState Register to be set to the wake-up from Sleep state (0x5555). The events in this table cause the security registers to be cleared, the logic reinitialized, and the device returned to the Active Power state (ready to receive a command).

Table L-4. Description of Events Causing the ChipState Register to be Set to 0x5555

Event	Event Description
Wake-up from Sleep	Wake-up from the Sleep power state. (Appendix L.1.3, Sleep State)
Reset Command	Device receives a valid Reset command block. (Section 7.22, Reset Command)
Tamper	Device reset initiated by the tamper sensors. (Section 3.1.2, Hardware Security Features)

L.3.3 Events that Do Not Change ChipState

The following events cause *no change* in the ChipState Register value. These events do not modify the security state of the ATAES132; therefore, do not cause the ChipState to change.

Table L-5. Description of Events Causing *No Change* in the ChipState Register

Event	Event Description
Wake-Up from Standby	Wake-Up from the Standby Power state. (Appendix L.1.2, Standby State)
Reading STATUS	Reading the STATUS Register with SPI RDSR or standard read commands. (Appendix G, Understanding the STATUS Register)
Writing IO Address Reset	Writing the IO Address Reset Register. (Appendix D.3, IO Address Reset Register)
Reading a Response	Reading the Response Memory Buffer. (Appendix D.2, Response Memory Buffer)
Command CRC Error	Device receives *any* command block which results in a CRCE error. (1) (Appendix G.1.4, CRCE Status Bit [4])
Command Invalid	Device receives a command block containing an undefined/invalid opcode. (Section 6.2, Command Summary).
ACK Polling	I^2C Acknowledge Polling. (Appendix J.3.7, Acknowledge Polling)
I^2C Read	I^2C Standard Read (READ, RREAD, SREAD instructions) (Appendix J.3, I^2C Instruction Set).
Invalid I^2C Write	I^2C standard Write beginning at any address from 0x1000 to 0xEFFF or above 0xF300, except address 0xFE00 (BWRITE, PWRITE instructions)(2) (Appendix J.3).
I^2C SRESET	I^2C SRESET instruction (Appendix J.3.6, Software Reset (SRESET)).
SPI Read	SPI standard read [READ instruction] (Appendix K.3.3, Read Memory Command (READ)).
Invalid SPI Write	SPI standard write beginning at any address from `0x1000` to `0xEFFF` or above `0xF300`, except address `0xFE00` (WREN, WRITE, WRDI instructions)(2) (Appendix K.3, SPI Instruction Set).
INFO command	Device receives a valid `INFO` command block (Section 7.12, `INFO` Command).

Notes: 1. A CRCE error results from a command block with a short count, bad checksum, or buffer overrun.

2. Writing the Command Memory Buffer (address `0xFE00`) may or may not change ChipState, depending on which command is written to the buffer.

L.3.4 ChipState = Active

The following events cause the ChipState Register to be set to the Active state (0x0000). The events in this table may result in a change in the security state of the device.

Table L-6. Description of Events Causing the ChipState Register to be Set to 0x0000

Event	Event Description	Section
Auth Command	Device receives a valid Auth command block.	7.1
AuthCheck Command	Device receives a valid AuthCheck command block.	7.2
AuthCompute Command	Device receives a valid AuthCompute command block.	7.3
BlockRead Command	Device receives a valid BlockRead command block.	7.4
Counter Command	Device receives a valid Counter command block.	7.5
Crunch Command	Device receives a valid Crunch command block.	7.6
DecRead	Device receives a valid DecRead command block.	7.7
Decrypt Command	Device receives a valid Decrypt command block.	7.8
EncRead Command	Device receives a valid EncRead command block.	7.9
Encrypt Command	Device receives a valid Encrypt command block.	7.10
EncWrite Command	Device receives a valid EncWrite command block.	7.11
KeyCreate Command	Device receives a valid KeyCreate command block.	7.13
KeyImport Command	Device receives a valid KeyImport command block.	7.14
KeyLoad Command	Device receives a valid KeyLoad command block.	7.15
KeyTransfer Command	Device receives a valid KeyTransfer command block.	7.16
Legacy Command	Device receives a valid Legacy command block.	7.17
Lock Command	Device receives a valid Lock command block.	7.18
Nonce Command	Device receives a valid Nonce command block.	7.19
NonceCompute Command	Device receives a valid NonceCompute command block.	7.20
Random Command	Device receives a valid Random Command block.	7.21
Sleep Command	Device receives a valid Sleep command block.	7.23
WriteCompute Command	Device receives a valid WriteCompute command block.	7.24
I^2C Write	I^2C standard Write beginning at any user zone address, any Configuration Memory address, or any Key Memory address (BWRITE, PWRITE instructions).	J.3
SPI Write	SPI standard Write beginning at any user zone address, any Configuration Memory address, or any Key Memory address (WREN, WRITE, WRDI instructions).	K.3

Appendix M. Block Checksum

An Atmel CRC-16 Checksum is used to verify the integrity of blocks communicated to and from the ATAES132A.

The Host sends ATAES132A extended commands to the device in a block of at least four bytes. The ATAES132A responses are returned to the Host in a block of at least four bytes. The command and response blocks are constructed in the following manner:

Byte #	Name	Meaning
0	Count	Number of bytes to be transferred to the device in the block, including count, packet, and checksum. This byte will always have a value of N.
1 to (N-3)	Packet	Command, parameters and data, or response. Data are transmitted in the byte order shown in command definitions in Section 7, Command Definitions.
N-2, N-1	Checksum	Atmel CRC-16 verification of the Count and packet bytes.

The Atmel CRC-16 polynomial is `0x8005`. The initial register value should be `0x0000`. After the last bit of the Count and packet has been transmitted, the internal CRC Register should have a value that matches that in the block. The first Checksum byte transmitted (N-2) is the most-significant byte of the CRC value, and the last byte of the block is the least-significant byte of the CRC.

M.1 Checksum Function

```
/** \This function calculates a 16-bit CRC.
 * \param[in] count number of bytes in data buffer
 * \param[in] data pointer to data
 * \param[out] crc pointer to calculated CRC (high byte at crc[0])
 */
void CalculateCrc(uint8_t length, uint8_t *data, uint8_t *crc)
{
  uint8_t counter;
  uint8_t crcLow = 0, crcHigh = 0, crcCarry;
  uint8_t polyLow = 0x05, polyHigh = 0x80;
  uint8_t shiftRegister;
  uint8_t dataBit, crcBit;

  for (counter = 0; counter < length; counter++) {
  for (shiftRegister = 0x80; shiftRegister > 0x00; shiftRegister >>= 1) {
     dataBit = (data[counter] & shiftRegister) ? 1 : 0;
     crcBit = crcHigh >> 7;

     // Shift CRC to the left by 1.
     crcCarry = crcLow >> 7;
     crcLow <<= 1;
     crcHigh <<= 1;
     crcHigh |= crcCarry;

     if ((dataBit ^ crcBit) != 0) {
      crcLow ^= polyLow;
      crcHigh ^= polyHigh;
     }
   }
  }
  crc[0] = crcHigh;
  crc[1] = crcLow;
}
```

M.2 Checksum Examples

```
DATA = 09 02 02 00 00 00 00 CRC = 0xF960
```

Appendix N. ATAES132A Command Response Time

The typical and maximum time required for the ATAES132A to process an extended command is shown in Table N-1. The response time is the time from sending the last bit of the last byte of the command block to the Command Memory Buffer until the STATUS Register (or I^2C ACK) indicates the response block is available. The *typical* response time is the average time required for an error-free command to be processed on a typical device at room temperature. The *maximum* response time is the worst-case time for the command to be processed over the specified temperature range (with or without an error condition, whichever results in the worst response time).

Table N-1. ATAES132A Extended Commands Typical and Maximum Response Times (1)

Command Description	Typical (2) ms	Maximum (3) ms
Auth, Reset (Mode [0:1] = 00b)	0.5	0.7
Auth, Inbound-Only (Mode [5:7] = 000b)	1.7	2.4
Auth, Inbound-Only (Mode [5:7] not 000b)	2.0	2.8
Auth, Inbound-Only (Mode [5:7] not 000b), with Key Usage. (5)	5.3	21.0
Auth, Outbound-Only (Mode [5:7] = 000b)	1.7	2.4
Auth, Outbound-Only (Mode [5:7] not 000b)	2.0	2.8
Auth, Outbound-Only (Mode [5:7] not 000b), with Key Usage. (5)	5.3	21.0
Auth, Mutual (Mode [5:7] = 000b)	2.6	3.6
Auth, Mutual (Mode [5:7] not 000b)	3.1	4.3
Auth, Mutual (Mode [5:7] not 000b), with Key Usage. (5)	6.4	22.6
AuthCheck	1.9	2.7
AuthChec, with Key Usage. (5)	5.2	20.9
AuthCompute	2.0	2.7
AuthCompute, with Key Usage. (5)	5.3	20.9
BlockRead, 32 bytes	0.9	1.3
Counter, Read, without MAC	0.6	0.8
Counter, Read, with OutMAC (Mode [5:7] = 000b)	1.8	2.5
Counter, Read, with OutMAC (Mode [5:7] not 000b)	2.1	2.9
Counter, Read, with OutMAC (Mode [5:7] not 000b), with Key Usage. (5)	5.4	21.1
Counter, Increment, without MAC	3.9	4.4
Counter, Increment, with InMAC (Mode [5:7] = 000b)	5.1	6.2
Counter, Increment, with InMAC (Mode [5:7] not 000b)	5.4	6.5
Counter, Increment, with InMAC (Mode [5:7] not 000b), with Key Usage. (5)	8.7	24.8
Crunch, with Count 0x0001	0.9	1.2
DecRead	2.5	3.5
DecRead, with Key Usage. (5)	5.9	21.8
Decrypt, 1 to 16 bytes (Mode [5:7] = 000b)	2.4	3.4
Decrypt, 1 to 16 bytes (Mode [5:7] not 000b)	2.7	3.7
Decrypt, 1 to 16 bytes (Mode [5:7] not 000b), with Key Usage. (5)	6.0	21.9
Decrypt, 17 to 32 bytes (Mode [5:7] = 000b)	3.2	4.3

Command Description	Typical [2] ms	Maximum [3] ms
Decrypt, 17 to 32 bytes (Mode [5:7] not 000b)	3.4	4.7
Decrypt, 17 to 32 bytes (Mode [5:7] not 000b), with Key Usage. [5]	6.7	22.9
EncRead, 1 to 16 bytes (Mode [5:7] = 000b)	2.5	3.5
EncRead, 1 to 16 bytes (Mode [5:7] not 000b)	2.8	3.9
EncRead, 1 to 16 bytes (Mode [5:7] not 000b), with Key Usage. [5]	6.1	22.1
EncRead, 17 to 32 bytes (Mode [5:7] = 000b)	3.2	4.5
EncRead, 17 to 32 bytes (Mode [5:7] not 000b)	3.5	4.8
EncRead, 17 to 32 bytes (Mode [5:7] not 000b), with Key Usage. [5]	6.8	23.1
EncRead, Configuration Memory Signature Generation Mode	9.1	12.7
EncRead, Key Memory Signature Generation Mode	13.9	18.4
Encrypt, 1 to 16 bytes (Mode [5:7] = 000b)	2.4	3.4
Encrypt, 1 to 16 bytes (Mode [5:7] not 000b)	2.7	3.7
Encrypt, 1 to 16 bytes (Mode [5:7] not 000b), with Key Usage. [5]	6.0	21.9
Encrypt, 17 to 32 bytes (Mode [5:7] = 000b)	3.0	4.1
Encrypt, 17 to 32 bytes (Mode [5:7] not 000b)	3.2	4.5
Encrypt, 17 to 32 bytes (Mode [5:7] not 000b), with Key Usage. [5]	6.5	22.7
EncWrite, 1 to 16 bytes (Mode [5:7] = 000b)	9.1	10.8
EncWrite, 1 to 16 bytes (Mode [5:7] not 000b)	9.4	11.1
EncWrite, 1 to 16 bytes (Mode [5:7] not 000b), with Key Usage. [5]	12.4	29.0
EncWrite, 17 to 32 bytes (Mode [5:7] = 000b)	9.9	11.9
EncWrite, 17 to 32 bytes (Mode [5:7] not 000b)	10.2	12.2
EncWrite, 17 to 32 bytes (Mode [5:7] not 000b), with Key Usage. [5]	13.2	30.1
EncWrite a Key (Mode [5:7] = 000b)	15.8	18.1
EncWrite a Key (Mode [5:7] not 000b)	16.1	18.5
EncWrite a Key (Mode [5:7] not 000b), with Key Usage. [5]	19.4	36.7
INFO	0.5	0.7
KeyCreate, without RNG Seed Update. (Mode [5:7] = 000b)	17.0	19.9
KeyCreate, without RNG Seed Update. (Mode [5:7] not 000b)	17.3	20.2
KeyCreate, without RNG Seed Update. (Mode [5:7] not 000b), with Key Usage. [5]	20.6	38.5
KeyCreate, with RNG Seed Update. (Mode [5:7] = 000b)	32.4	37.4
KeyCreate, with RNG Seed Update. (Mode [5:7] not 000b)	32.9	38.2
KeyCreate, with RNG Seed Update. (Mode [5:7] not 000b), with Key Usage. [5]	35.2	54.9
KeyCreate, VolatileKey with RNG Seed Update. (Mode [5:7] = 000b)	18.8	22.4
KeyCreate, VolatileKey with RNG Seed Update. (Mode [5:7] not 000b)	19.4	23.1
KeyCreate, VolatileKey with RNG Seed Update. (Mode [5:7] not 000b), with Key Usage. [5]	22.7	41.3
KeyImport (Mode [5:7] = 000b)	15.8	18.2
KeyImport (Mode [5:7] not 000b)	16.1	18.5
KeyImport (Mode [5:7] not 000b), with Key Usage. [5]	19.4	36.7

Command Description	Typical [2] ms	Maximum [3] ms
KeyLoad (Mode [5:7] = 000b)	15.8	18.2
KeyLoad (Mode [5:7] not 000b)	16.1	18.5
KeyLoad (Mode [5:7] not 000b), with Key Usage. [5]	19.4	36.7
KeyTransfer	14.2	15.8
Legacy	1.2	1.7
Legacy, with Key Usage. [5]	4.5	19.9
Lock SmallZone, Key Memory, Configuration Memory, with Checksum.	16.8	20.6
Lock User Zone, without MAC	3.8	4.4
Lock User Zone, with MAC (Mode [5:7] = 000b)	5.1	6.1
Lock User Zone, with MAC (Mode [5:7] not 000b)	5.3	6.5
Lock User Zone, with MAC (Mode [5:7] not 000b), with Key Usage. [5]	8.7	24.7
Nonce, Inbound	0.5	0.7
Nonce, Random, without RNG Seed Update.	2.1	2.9
Nonce, Random, with RNG Seed Update.	16.8	19.5
NonceCompute	0.9	1.3
Random, without RNG Seed Update.	1.7	2.4
Random, with RNG Seed Update.	16.3	18.8
Reset [4]	1.3	1.7
Sleep, enter Standby state. [4]	0.1	0.1
Sleep, enter Sleep state. [4]	0.1	0.1
WriteCompute, 1 to 16 bytes	2.6	3.7
WriteCompute, 1 to 16 bytes, with Key Usage. [5]	5.9	21.8
WriteCompute, 17 to 32 bytes	3.2	4.4
WriteCompute, 17 to 32 bytes		
WriteCompute, 17 to 32 bytes, with Key Usage. [5]	6.5	22.3

Notes:
1. The values in this table are based on characterization and/or simulation. These parameters are not tested.
2. The typical response time is the time required for 60% of devices to place a packet in the Response Memory Buffer and change the WIP status bit to 0b after successful execution of the command at room temperature. If an error occurs, the response will be available in a shorter amount of time.
3. The maximum response time is the time required for 95% of devices to place a packet in the Response Memory Buffer and change the WIP Status bit to 0b after successful execution of the command at the worst case temperature.
   Note: 5% of the devices may be slower than this number. The Host is expected to read the STATUS Register to determine when a response is available (see Appendix G, Understanding the STATUS Register).
4. The Reset command and the Sleep command do not generate a response. The response times are the time required for the operation to be completed.
5. These times are with the Key Usage limits enabled in the KeyConfig Register. All other times are with the Key Usage limits disabled in the KeyConfig Register.

Appendix O. Default Configuration

The ATAES132A memory map is shown in Table O-1 with the default memory values. Reserved memory cannot be written or read.

Table O-1. ATAES132A Memory Map Showing the Default Memory Contents

Byte Address	Description
0000_h-$0FFF_h$	User Memory (Default = All bytes FF_h)
1000_h-$EFFF_h$	Reserved
$F000_h$-$F1FF_h$	Configuration Memory (see Appendix O.1, Configuration Memory Contents for default values)
$F200_h$-$F2FF_h$	Key Memory (see Appendix O.2, Key Memory Contents for default values)
$F300_h$-$FDFF_h$	Reserved
$FE00_h$	Command / Response Memory Buffer
$FE01_h$-$FFFD_h$	Reserved
$FFE0_h$	I/O Address Reset
$FFE1_h$-$FFEF_h$	Reserved
$FFF0_h$	STATUS Register
$FFF1_h$-$FFFF_h$	Reserved

O.1 Configuration Memory Contents

The default contents of the Configuration Memory after completion of production test are shown in Table O-2. This configuration enables most functions, and is expected to be changed by the customer during personalization. See Appendix E, Configuration Memory Map.

Table O-2. Default Configuration Memory contents (All Register Values Shown are Hexadecimal Numbers)

Address	0_h / 8_h	1_h / 9_h	2_h / A_h	3_h / B_h	4_h / C_h	5_h / D_h	6_h / E_h	7_h / F_h
$F000_h$-$F007_h$	Unique Die Serial Number							
$F008_h$-$F00F_h$	Atmel Proprietary Data							
$F010_h$-$F017_h$	00	1F	Atmel Proprietary Data			00	00	20
$F018_h$-$F01F_h$	20	20	0A	Atmel Proprietary Data				
$F020_h$-$F027_h$	55	55	55	Atmel Proprietary Data				
$F028_h$-$F02F_h$	Atmel Proprietary Data			00	EE	03	Atmel Data	
$F030_h$-$F037_h$	Atmel Proprietary Data							
$F038_h$-$F03F_h$								
$F040_h$-$F047_h$	I2CAddr	C3	FF	FF	FF	FF	FF	FF
$F048_h$-$F04F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F050_h$-$F057_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F058_h$-$F05F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F060_h$-$F067_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F068_h$-$F06F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F070_h$-$F077_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F078_h$-$F07F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F080_h$-$F087_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F088_h$-$F08F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F090_h$-$F097_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F098_h$-$F09F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F0A0_h$-$F0A7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F0A8_h$-$F0AF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F0B0_h$-$F0B7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F0B8_h$-$F0BF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F0C0_h$-$F0C7_h$	00	FF	FF	FF	00	FF	FF	FF
$F0C8_h$-$F0CF_h$	00	FF	FF	FF	00	FF	FF	FF
$F0D0_h$-$F0D7_h$	00	FF	FF	FF	00	FF	FF	FF
$F0D8_h$-$F0DF_h$	00	FF	FF	FF	00	FF	FF	FF
$F0E0_h$-$F0E7_h$	00	FF	FF	FF	00	FF	FF	FF
$F0E8_h$-$F0EF_h$	00	FF	FF	FF	00	FF	FF	FF
$F0F0_h$-$F0F7_h$	00	FF	FF	FF	00	FF	FF	FF
$F0F8_h$-$F0FF_h$	00	FF	FF	FF	00	FF	FF	FF
$F100_h$-$F107_h$	FF	FF	00	00	00	00	00	00
$F108_h$-$F10F_h$	FF	FF	00	00	00	00	00	00
$F110_h$-$F117_h$	FF	FF	00	00	00	00	00	00

Address	0_h / 8_h	1_h / 9_h	2_h / A_h	3_h / B_h	4_h / C_h	5_h / D_h	6_h / E_h	7_h / F_h
$F118_h$-$F11F_h$	FF	FF	00	00	00	00	00	00
$F120_h$-$F127_h$	FF	FF	00	00	00	00	00	00
$F128_h$-$F12F_h$	FF	FF	00	00	00	00	00	00
$F130_h$-$F137_h$	FF	FF	00	00	00	00	00	00
$F138_h$-$F13F_h$	FF	FF	00	00	00	00	00	00
$F140_h$-$F147_h$	FF	FF	00	00	00	00	00	00
$F148_h$-$F14F_h$	FF	FF	00	00	00	00	00	00
$F150_h$-$F157_h$	FF	FF	00	00	00	00	00	00
$F158_h$-$F15F_h$	FF	FF	00	00	00	00	00	00
$F160_h$-$F167_h$	FF	FF	00	00	00	00	00	00
$F168_h$-$F16F_h$	FF	FF	00	00	00	00	00	00
$F170_h$-$F177_h$	FF	FF	00	00	00	00	00	00
$F178_h$-$F17F_h$	FF	FF	00	00	00	00	00	00
$F180_h$-$F187_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F188_h$-$F18F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F190_h$-$F197_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F198_h$-$F19F_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1A0_h$-$F1A7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1A8_h$-$F1AF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1B0_h$-$F1B7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1B8_h$-$F1BF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1C0_h$-$F1C7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1C8_h$-$F1CF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1D0_h$-$F1D7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1D8_h$-$F1DF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1E0_h$-$F1E7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1E8_h$-$F1EF_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1F0_h$-$F1F7_h$	FF	FF	FF	FF	FF	FF	FF	FF
$F1F8_h$-$F1FF_h$	FF	FF	FF	FF	FF	FF	FF	FF

Notes:
1. Orange Registers = Locked at the factory and cannot be changed by the customer.
2. Blue Registers = Lock registers can be changed only by using the `Lock` command (Section7.9, `EncRead` Command).
3. Green Registers = Configuration registers can be written by the customer prior to locking (by setting LockConfig to `0x00` using the `Lock` command).
4. Yellow Registers = The SmallZone Register can be written by the customer prior to locking (by setting LockSmall to `0x00` using the `Lock` command). SmallZone is locked separately from the remainder of the Configuration Memory.

The default value of the I2CAddr Register is `0x01` for devices configured for I^2C interface mode. The default value of I^2CAddr is `0x00` for devices configured for SPI interface mode. See Appendix Q, Ordering Information for ordering codes.

O.2 Key Memory Contents

The Key Memory contains pseudorandom values after completion of production test, except for Key 00 which contains the Transport Key. Device personalization can be performed without knowledge of the Transport Key; however, secure personalization can be performed only if the Transport Key value has been obtained from Atmel.

Appendix P. Serial Memory Backward Compatibility

The ATAES132A secure Serial EEPROM architecture was developed to allow security to be retrofitted into systems using standard Atmel Serial EEPROM. The ATAES132A package pinouts, the interface protocol, and the command set are all compatible with standard I^2C and SPI EEPROM, but are not identical.

This section describes the differences that must be considered when the ATAES132A is inserted into systems using I^2C or SPI Serial EEPROM.

P.1 I^2C Serial EEPROM Compatibility

This section describes differences between the Atmel AT24C32C standard 32Kb I^2C Serial EEPROM and the ATAES132A secure Serial EEPROM configured for I^2C communication mode.

P.1.1 Package Pins

On AT24C32C, pins 1, 2, and 3 are used to set I^2C Device Address bits A_0, A_1, and A_2. The AT24C32C pin 7 is the Write Protect (WP) input.

On the ATAES132A, pins 1, 2, 3, and 7 are not used in I^2C communication mode. These pins should be tied to V_{CC} or V_{SS}. The state of these four pins has no impact on the functionality of the ATAES132A in the I^2C communication mode. See Appendix J.2, Pin Descriptions.

P.1.2 I^2C Device Address

The AT24C32C I^2C Device Address is $1010A_2A_1A_0b$, with A_0, A_1, and A_2 determined by the state of pins 1, 2, and 3. A maximum of eight AT24C32C devices are permitted on the I^2C interface.

On the ATAES132A, the I^2C Device Address is determined by the contents of the I2CAddr Register (see Appendix J.1.3, I^2C Device Address). The ATAES132A I^2C Device Address can be any set to any value, allowing up to 127 devices on the I^2C interface.

P.1.3 Write Protect

The AT24C32C Write Protect (WP) input pin inhibits all EEPROM Write operations when the WP pin is high. If WP is low, then EEPROM Write operations are allowed.

On the ATAES132A, the User Memory Write permissions are controlled by the ZoneConfig Registers (see Appendix E.2.20, ZoneConfig Registers). The User Memory is divided into 16 user zones that are independently controlled by 16 ZoneConfig Registers; different Write permissions can be assigned to different sections of the memory. By default, all User Memory has open Write access.

P.1.4 Page Write Operations

If the Host attempts to write data across the physical (32 byte) EEPROM page boundary, the AT24C32C wraps to the beginning of the EEPROM page where the Page Write operation begins and performs the EEPROM Write after receiving a Stop condition. If the Host attempts to write more than 32 bytes in a Page Write operation, then the AT24C32C wraps the data at the page boundary and performs the EEPROM Write after receiving a Stop condition. Partial Page Writes are supported by the AT24C32C.

The ATAES132A does not allow Write operations to cross physical (32 byte) EEPROM page boundaries (see Appendix B.2, EEPROM Page Boundary) and does not allow a Write operation if more than 32 data bytes are received from the Host. In both cases, the EEPROM contents remain unchanged, the data is discarded, and an error bit is set in the STATUS Register (see Appendix J.3.3, Page Write (PWRITE)). Partial Page Writes are supported by the ATAES132A.

P.1.5 Read Operations

Reading beyond the end of physical memory on the AT24C32C causes the internal data address register to roll-over to address zero. The Read operation continues from address zero.

If an ATAES132A Read operation begins at a valid User Memory address but continues past the end of User Memory, the Read operation will not wrap to the beginning of User Memory. Reading beyond the end of User Memory causes `0xFF` to be returned to the Host in reply to the Read, the internal data address register stops incrementing, and an error bit is set in the STATUS Register (see Appendix G.2.5, Read User Memory).

P.1.6 Read Protect

The AT24C32C and other standard I^2C EEPROMs do not have a Read inhibit function.

On the ATAES132A, the User Memory Read permissions are controlled by the ZoneConfig Registers (see Appendix E.2.20, ZoneConfig Registers). The User Memory is divided into 16 user zones that are independently controlled by 16 ZoneConfig Registers; different Read permissions can be assigned to different sections of the memory. If Read access is prohibited, then `0xFF` will be returned to the Host in reply to a read command (see Section 5.1, Read). By default all User Memory has open Read access.

P.1.7 Standby Mode

Standard I^2C EEPROMs automatically enter low-power standby mode upon completion of any internal operation.

The ATAES132A has three powered states:

- Active State and Two Low-Power States
- Standby State
- Sleep State

The ATAES132A will remain in the Active state between operations unless the Host sends a `Sleep` command to activate the Standby state or the Sleep state. The ATAES132A can also be configured to automatically enter a Low-Power state at power-up. See Appendix L, Power Management for details on the power management features.

P.1.8 Operating Voltage

- The AT24C32C operating range is 1.8V minimum to 5.5V maximum.
- The ATAES132A operating range is 2.5V minimum to 5.5V maximum. See Section 9.3, DC Characteristics.

P.2 SPI Serial EEPROM Compatibility

This section describes differences between the AT25320B standard Atmel 32Kb SPI Serial EEPROM and the ATAES132A secure Serial EEPROM configured for SPI communication mode.

P.2.1 Package Pins

On the AT25320B, pin 3 is the $\overline{WP}$ input and pin 7 is the $\overline{HOLD}$ input.

On the ATAES132A, pins 3 and 7 are not used in SPI communication mode; these pins can be tied to V_{CC} or V_{SS}. The state of these two pins have no impact on the functionality of the ATAES132A in the SPI communication mode. See Appendix K.2, SPI Communication Mode Pin Descriptions for the pin descriptions.

P.2.2 Write Protect ($\overline{WP}$)

The AT25320B $\overline{WP}$ input pin inhibits all EEPROM Write operations when the WP pin is low. If WP is high, then EEPROM Write operations are allowed. The Write protect pin can be disabled by writing the WPEN bit in the STATUS Register to `0b`.

On the ATAES132A, the User Memory Write permissions are controlled by the ZoneConfig Registers (see Appendix E.2.20, ZoneConfig Registers). The User Memory is divided into 16 user zones that are independently controlled by 16 ZoneConfig Registers; different Write permissions can be assigned to different sections of the memory. By default, all User Memory has open Write access.

P.2.3 Hold

The AT25320B $\overline{HOLD}$ input pin allows the Host to pause communication with the memory temporarily (by bringing $\overline{HOLD}$ low) and then resume the communication sequence (by bringing $\overline{HOLD}$ high). The sequence continues exactly from the point where it was paused as if there was no interruption.

The ATAES132A does not have a Hold function. If communications are interrupted, the sequence must be restarted beginning with a high-to-low transition on the $\overline{CS}$ input.

P.2.4 Page Write Operations

If the Host attempts to write data across the physical (32-byte) EEPROM page boundary, the AT25320B wraps to the beginning of the EEPROM page where the Page Write operation begins and performs the EEPROM Write after receiving a low-to-high transition on the $\overline{CS}$ input. If the Host attempts to write more than 32 bytes in a Page Write operation, then the AT25320B wraps the data at the page boundary and performs the EEPROM write after receiving a Stop condition. Partial Page Writes are supported by the AT25320B.

The ATAES132A does not allow Write operations to cross physical (32 byte) EEPROM page boundaries (see Appendix B.2, EEPROM Page Boundary and does not allow a Write operation if more than 32 data bytes are received from the Host. In both cases, the EEPROM contents remain unchanged, the data is discarded, and an error bit is set in the STATUS Register (see Appendix J.3.3, Page Write (PWRITE)). Partial Page Writes are supported by the ATAES132A.

P.2.5 Read Operations

Reading beyond the end of physical memory on AT25320B causes the internal data address register to roll-over to address zero. The Read operation continues from address zero.

If an ATAES132A Read operation begins at a valid User Memory address but continues past the end of User Memory, the Read operation will not wrap to the beginning of User Memory. Reading beyond the end of User Memory causes `0xFF` to be returned to the Host in reply to the Read, the internal data address register stops incrementing, and an error bit is set in the STATUS Register.

P.2.6 Read Protect

The Atmel AT25320B and other standard SPI EEPROMs do not have a Read inhibit function.

On the ATAES132A, the User Memory Read permissions are controlled by the ZoneConfig registers (see Appenidx E.2.20). The User Memory is divided into 16 user zones that are independently controlled by 16 ZoneConfig registers; different Read permissions can be assigned to different sections of the memory. If Read access is prohibited, then `0xFF` will be returned to the Host in reply to a read command (see Section 5.1, Read). By default, all User Memory has open Read access.

P.2.7 STATUS Register

The AT25320B STATUS Register definition is shown in Table P-1. The default state of all STATUS bits is 0b. The WPEN bit controls the Write Protect pin. Block Write protection is controlled by the BP0 and BP1 bits. If WEN = 1b, then the device is Write Enabled. If WIP = 0b, the device is ready to accept a command; WIP = 1b indicates a write cycle is in progress. The reserved bits are 0b, except when an internal write cycle is in progress. All bits of the STATUS Register are 1b when an internal write cycle is in progress.

Table P-1. AT25320B STATUS Register Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPEN	Reserved	Reserved	Reserved	BP1	BP0	WEN	WIP

The ATAES132A STATUS Register definition is shown in Table P-2 and described in Appendix G, Understanding the STATUS Register. The default state of all STATUS bits is 0b. The WEN, WIP, and reserved bits are similar to those of standard SPI Serial EEPROM: If WEN = 1b, then the device is Write Enabled. If WIP = 0b, the device is ready to accept a command; WIP = 1b indicates a write cycle or a cryptographic operation is in progress. The reserved bits are 0b except when an internal write cycle or a cryptographic operation is in progress. All bits of the STATUS Register are 1b when an internal write cycle or a cryptographic operation is in progress.

Table P-2. ATAES132A STATUS Register definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EERR	RRDY	Reserved	CRCE	Reserved	WAKEb	WEN	WIP

ATAES132A reports errors to the Host using the EERR and CRCE bits. The RRDY bit indicates if the Response Memory Buffer is empty (0b), or ready to read (1b). The WAKEb bit indicates if the device is in the sleep or standby power state. See Appendix G.1, Device Status Register (STATUS) Definition for detailed descriptions of each STATUS bit.

P.2.8 Write Status Register Command (WRSR)

The AT25320B STATUS Register contains three bits that control the Block Write Protect function and the Write Protect pin. These bits can be changed by sending a Write Status Register (WRSR) command to the memory.

The ATAES132A does not support the Write Status Register (WRSR) command. The WRSR command will be ignored if it is received.

P.2.9 Block Write Protect

The AT25320B STATUS Register contains two block protect bits (BP0 and BP1) that control the Block Write Protect function. By writing the STATUS Register, the user can set the Block Protect bits to inhibit writes in ¼, ½, or the full Memory Array.

On the ATAES132A, the User Memory Write permissions are controlled by the ZoneConfig registers (see Appendix E.2.20, ZoneConfig Registers). The User Memory is divided into 16 user zones that are independently controlled by 16 ZoneConfig Registers; different Write permissions can be assigned to different sections of the memory. By default, all User Memory has open Write access.

P.2.10 Standby Mode

Standard SPI EEPROMs automatically enter low-power Standby mode upon completion of any internal operation.

The ATAES132A has three powered states: the Active state and two Low-Power states, the Standby state and the Sleep state. The ATAES132A will remain in the Active state between operations unless the Host sends a `Sleep` command to activate the Standby state or the Sleep state. The ATAES132A can also be configured to automatically enter a Low-Power state at power-up. See Appendix L, Power Management for details on the power management features.

P.2.11 Operating Voltage

The AT25320B operating voltage range is 1.8V minimum to 5.5V maximum.

The ATAES132A operating voltage range is 2.5V minimum to 5.5V maximum. See Section 9.3, DC Characteristics.

P.2.12 Maximum Operating Frequency

The AT25320B maximum SCK frequency is 10MHz when V_{CC} is 2.7V to 5.5V. The maximum SCK frequency is 20MHz when V_{CC} is 4.5V to 5.5V.

The ATAES132A maximum SCK frequency is 10MHz when V_{CC} is 2.5V to 5.5V. See Section 9.4, AC Characteristics for AC specifications.

Appendix Q. Ordering Information

The ATAES132A production ordering codes are listed in Appendix Q.1, Ordering Codes. To increase security, ATAES132A packages are not marked with the ordering code. The ATAES132A standard packages are marked with a trace code which is unique for each manufacturing lot. Contact Atmel for additional information.

Q.1 Ordering Codes

Atmel Ordering Code	Interface Configuration	Conditioning	Package	Lead Finish	Temperature Range
ATAES132A-SHEQ-B	SPI	Bulk(1)	8S1	NiPdAu Lead-free/Halogen-free (Exceeds RoHS Requirments)	Industrial Temperature (-40°C to 85°C)
ATAES132A-SHER-B	I^2C				
ATAES132A-SHEQ-T	SPI	Tape and Reel(2)			
ATAES132A-SHER-T	I^2C				
ATAES132A-MAHEQ-T	SPI		8MA2		
ATAES132A-MAHER-T	I^2C				

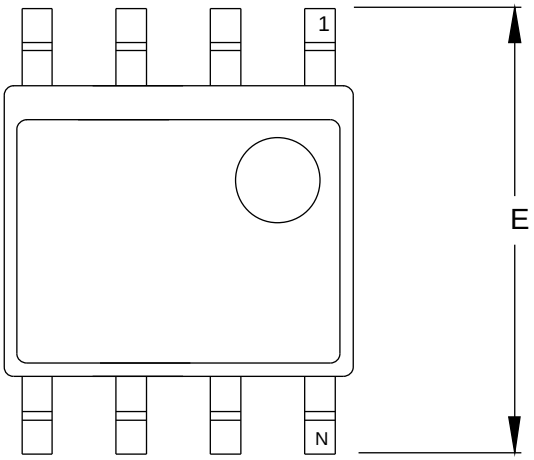
Notes: 1. -B = Bulk
- SOIC = 100 per tube.

2. -T = Tape and Reel
- SOIC = 4,000 per reel.
- UDFN = 15,000 per reel.

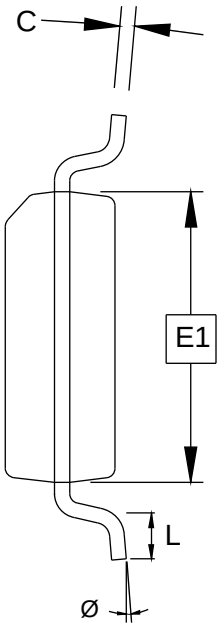
Package Type	
8S1	8-lead, 0.150” wide body, Plastic Gull Wing Small Outline, Green (JEDEC SOIC)
8MA2	8-pad, 2.0mm x 3.0mm x 0.6mm body, Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead, Green (UDFN)

Q.2 Mechanical Information

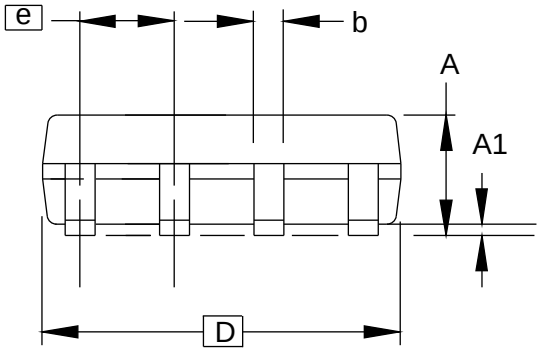
Q.2.1 8S1 — 8-lead JEDEC SOIC



TOP VIEW



END VIEW



SIDE VIEW

Notes: This drawing is for general information only.
Refer to JEDEC Drawing MS-012, Variation AA
for proper dimensions, tolerances, datums, etc.

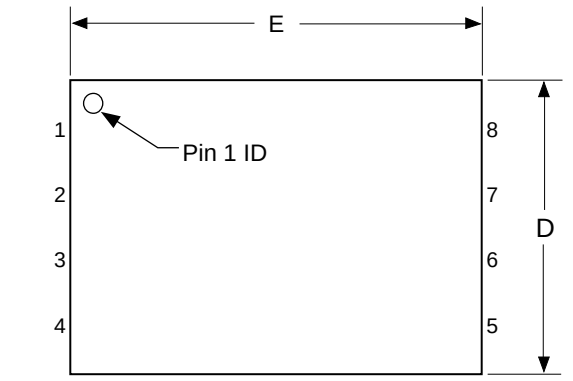
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.75	
A1	0.10	–	0.25	
b	0.31	–	0.51	
C	0.17	–	0.25	
D		4.90 BSC		
E		6.00 BSC		
E1		3.90 BSC		
e		1.27 BSC		
L	0.40	–	1.27	
Ø	0°	–	8°	

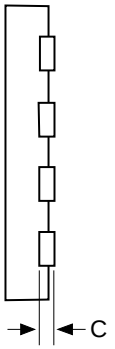
3/6/2015

Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE 8S1, 8-lead (0.150” Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)	GPC SWB	DRAWING NO. 8S1	REV. H

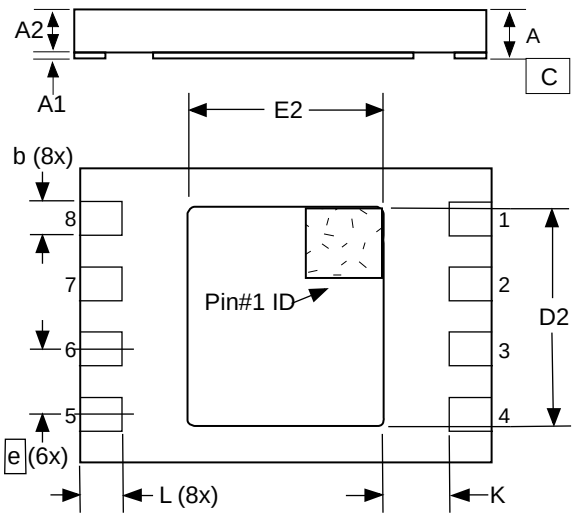
Q.2.2 8MA2 — 8-pad UDFN



TOP VIEW



SIDE VIEW



BOTTOM VIEW

Notes:
1. This drawing is for general information only. Refer to Drawing MO-229, for proper dimensions, tolerances, datums, etc.
2. The Pin #1 ID is a laser-marked feature on Top View.
3. Dimensions b applies to metallized terminal and is measured between 0.15 mm and 0.30 mm from the terminal tip. If the terminal has the optional radius on the other end of the terminal, the dimension should not be measured in that radius area.
4. The Pin #1 ID on the Bottom View is an orientation feature on the thermal pad.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.50	0.55	0.60	
A1	0.0	0.02	0.05	
A2	-	-	0.55	
D	1.90	2.00	2.10	
D2	1.40	1.50	1.60	
E	2.90	3.00	3.10	
E2	1.20	1.30	1.40	
b	0.18	0.25	0.30	3
C	0.152 REF			
L	0.35	0.40	0.45	
e	0.50 BSC			
K	0.20	-	-	

11/2/15

Atmel Package Drawing Contact: packagedrawings@atmel.com	TITLE 8MA2, 8-pad 2 x 3 x 0.6mm Body, Thermally Enhanced Plastic Ultra Thin Dual Flat No-Lead Package (UDFN)	GPC YNZ	DRAWING NO. 8MA2	REV. H

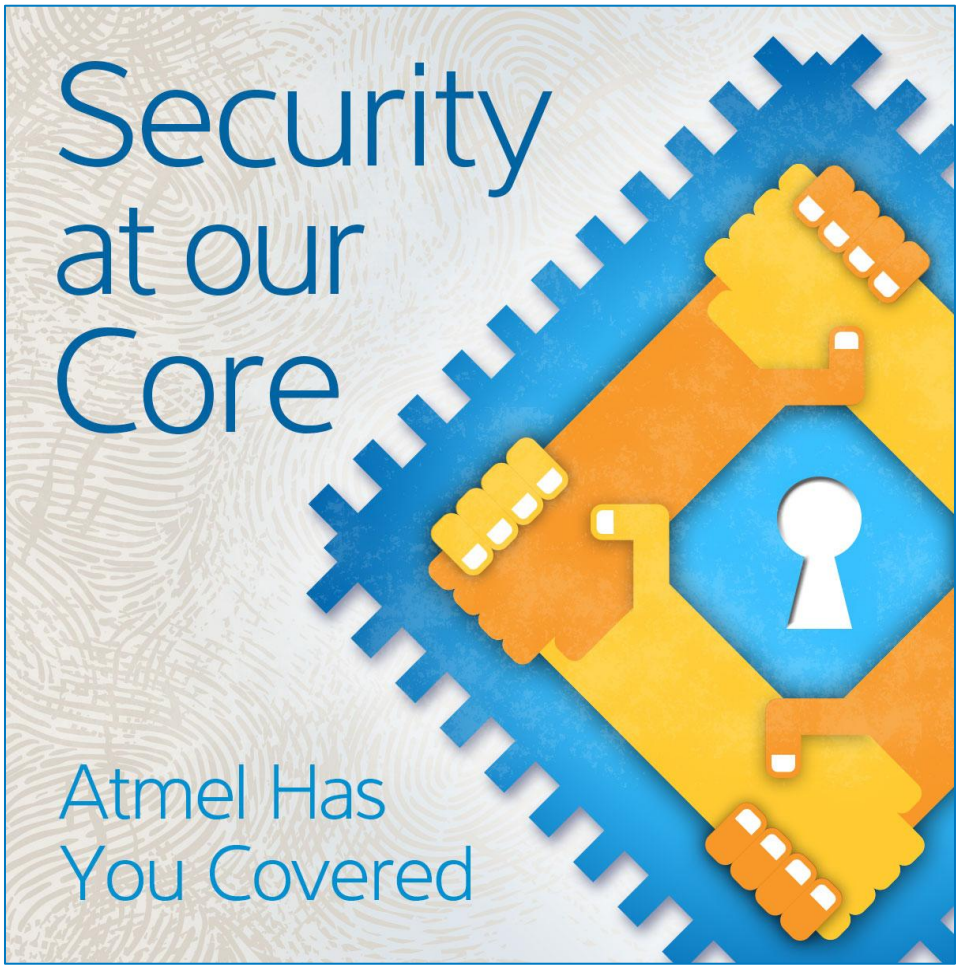
Appendix R. Errata

R.1 KeyCreate Command Executed with Usage Counter

If the KeyCreate command is executed with Mode bit 2 set to 1b and a Key with a Usage Counter attached to it is used. The Usage Counter will not be incremented.

Appendix S. Revision History

Doc. Rev.	Date	Comments
8914B	02/2016	Corrected ordering codes ATAES132A-SHEQ-B and ATAES132A-SHER-B by adding “-B”. Updated 8MA2 package drawing. Added a high-feature bullet, “Guaranteed Unique Die Serial Number.”
8914A	03/2015	Initial document release.



Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**