# Atmel AT88SA10HS

## Atmel CryptoAuthentication Host Security Chip

### DATASHEET

**Not Recommended for New Designs**
**Replaced by ATSHA204**

## Features

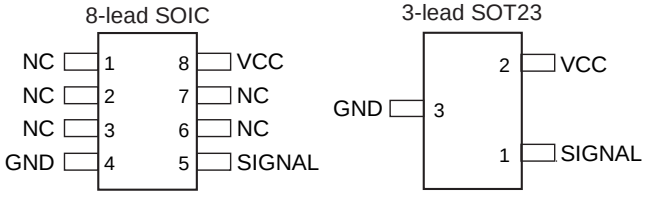
- Secure key storage to complement the Atmel® AT88SA100S and the Atmel AT88SA102S devices
- Superior SHA-256 hash algorithm
- Guaranteed Unique 48-bit serial number
- High speed single wire interface, optionally shared with client
- Supply voltage: 2.7V – 5.25V
- 1.8V – 5.5V communications voltage
- <150nA sleep current
- 4KV ESD protection
- Multi-level hardware security
- Secure personalization
- Green compliant (exceeds RoHS) 3-pin SOT-23 or 8-lead SOIC packages

## Applications

- Consumable device (battery, toner, other supplies) authentication
- Network and computer access control
- Authenticated communications for control networks
- Anti-clone authentication for daughter cards
- Physical access control (electronic lock and key)

**Figure 1. Pin Configurations**

| Pin name | Function |
|---|---|
| SIGNAL | Serial data, single-wire clock and data |
| GND | Ground |
| VCC | Power supply |



# 1. Introduction

The Atmel CryptoAuthentication™ family of chips is the first cost-effective authentication devices to implement the SHA-256 hash algorithm, which is part of the latest set of recommended algorithms by the US Government. The 256-bit key space renders any exhaustive attacks impossible.

The AT88SA10HS host version of CryptoAuthentication chips is capable of validating the response coming from the SHA-256 engine within an authentic CryptoAuthentication client (SA100S or SA102S), even if that response includes within the computation the serial number of the client. For detailed information on the cryptographic protocols, algorithm test values and usage models. See “Atmel AT88SA100S” and “Atmel AT88SA102S” datasheets, along with the application notes dedicated to this product family.

The host CryptoAuthentication performs three separate operations (named HOST0, HOST1, and HOST2) to implement this validation. The AT88SA10HS chip takes both the challenge and response as inputs and returns a single Boolean indicating whether or not the response is valid, in order to prevent the host chip from being used to model a valid client.

The host system is responsible for generating the random challenge that is sent to both the client and host CryptoAuthentication devices as AT88SA10HS does not include a random number generator.

Note: The chip implements a failsafe internal watchdog timer that forces it into a very low power mode after a certain time interval regardless of any current activity. System programming must take this into consideration. See Section 5.5 for more details.

## 1.1 Memory Resources

**Fuse** Block of 128-fuse bits that can be written through the one wire interface. Fuse[87] has special meanings. See Section 1.2 for more details. Fuses[88:95] are part of the manufacturer ID value fixed by Atmel. Fuses[96:127] are part of the serial number programmed by Atmel which is guaranteed to be unique. See Section 1.3 for more details on the Manufacturing ID and Serial Number.

**ROM** Metal mask programmed memory. Unrestricted reads are permitted on the first 64-bits of this array. The physical ROM will be larger and will contain other information that cannot be read. The following three fields are stored in the ROM:

**ROM MfrID** 2-bytes of ROM that specifies part of the manufacturing ID code. This Atmel assigned value is always the same for all chips of a particular model number. For the AT88SA10HS, this value is 0x2301. (Appears on the bus: 0x0123), ROM MfrID can be read by accessing ROM bytes 0 and 1 of Address 0.

**ROM SN** 2-bytes of ROM that can be used to identify chips among others on the wafer. These bits reduce the number of fuses necessary to construct a unique serial number. The MaskSN is read by accessing ROM bytes 2 and 3 of Address 0. The serial number can always be read by the system but is never included in the message digested by the HOST command.

**RevNum** 4-bytes of ROM that are used by Atmel to identify the model mask and/or design revision of the AT88SA10HS chip. These bytes can be freely read as the four bytes returned by ROM Address 1; however, system code should not depend on this value as it may change from time to time.

## 1.2 Fuse Map

The AT88SA10HS incorporates 128 one-time fuses within the chip. Once burned, there is no way to reset the value of a fuse. All fuses, with the exception of the Fuse MfrID and Fuse SN bits initialized by Atmel, have a value of one when shipped from the Atmel factory and transition to zero when they are burned. These fuses are burned at system personalization and cannot be changed after that time.

**Table 1-1. Fuse Map**

| Fuse # | Name | Description |
|---|---|---|
| 0 → 63 | Secret Fuses | These fuses can be securely written by the BurnSecure command but can never be read with the read command. |
| 64 → 86 | Status Fuses | These fuses can be written with the BurnSecure command and can always be read with the Read command. |
| 87 | Fuse Disable | The HOST commands ignore the values of Fuse[0-63] until this bit is burned. Once this bit is burned, the BurnSecure command is disabled. |
| 88 → 95 | Fuse MfrID | See Section 1.3. Set by Atmel, cannot be modified in the field. |
| 96 → 127 | Fuse SN | See Section 1.3. Set by Atmel, cannot be modified in the field. |

**Secret Fuses** These 64-fuses are used to augment the mask programmed keys stored in the chip by Atmel. Knowledge of both the mask keys and the values of the secret fuses are required to calculate the response value expected by HOST2. The BurnSecure command can be used to burn an arbitrary selection of these 64-bits.

**Status Fuses** These 23-fuses should be used to store information which is not secret, as their value can always be determined using the read command. Typical usage would be model or configuration information. They cannot be automatically included in the messages to be hashed by the HOST commands, but the system may read them and pass them back to HOST1 in the input stream if desired.

**Fuse Disable** This fuse is used to prevent access to fuses on chips in which a partial set of fuses has been burned. This fuse must be burned using the BurnSecure command.

## 1.3 Chip Identification

The chip includes a total of 72-bits of information that can be used to distinguish between individual chips in a reliable manner. The information is distributed between the ROM and fuse blocks in the following manner.

**Serial Number** This 48-bit value is composed of ROM SN (16-bits) and Fuse SN (32-bits). Together they form a serial number that is guaranteed to be unique for all devices ever manufactured within the CryptoAuthentication family. This value is optionally included in the MAC calculation.

**Manufacturing ID** This 24-bit value is composed of ROM MfrID (16-bits) and Fuse MfrID (8-bits). Typically this value is the same for all chips of a given type. It is always included in the cryptographic computations.

## 1.4 Key Values

The values stored in the AT88SA10HS internal key array are hardwired into the masking layers of the chip during wafer manufacture. All chips have the same keys stored internally, though the value of a particular key cannot be determined externally from the chip. For this reason, customers should ensure they program a unique (and secret) number into the 64-secret fuses and they should store the Atmel provided key values securely.

Individual key values are made available to qualified customers upon request to Atmel and are always transmitted in a secure manner.

When the serial number is included in the MAC calculation, the response is considered to be diversified and the host needs to know the base secret in order to be able to verify the authenticity of the client. A diversified response can also be obtained by including the serial number in the computation of the value written to the secret fuses. The AT88SA10HS provides a secure hardware mechanism to validate responses to determine if they are authentic.

## 1.5 SHA-256 Computation

AT88SA10HS performs only one cryptographic calculation – a keyed digest of an input challenge. It optionally includes various other information stored on the chip within the digested message.

The AT88SA10HS computes the SHA-256 digest based on the algorithm documented here:

http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

As a security measure, the 24-bit MfrID code (both ROM and Fuse bits) is automatically included in every message digested by AT88SA10HS. The secret fuses are conditionally appended, depending on the parameters to the HOST command.

For complete sample calculations, see “Atmel AT88SA100S” and/or “Atmel AT88SA102S” datasheets.

## 1.6 Security Features

AT88SA10HS incorporates a number of physical security features designed to protect the keys from release. These include an active shield over the entire surface of the part, internal memory encryption, internal clock generation, glitch protection, voltage tamper detection, and other physical design features.

Pre-programmed keys stored on AT88SA10HS, are encrypted in such a way as to make retrieval of their values via outside analysis very difficult.

Both the clock and logic supply voltage are internally generated, preventing any direct attack via the pins on these two signals.

# 2. IO Protocol

Communications to and from AT88SA10HS; take place over a single asynchronously timed wire using a pulse count scheme. The overall communications structure is a hierarchy:

**Table 2-1. IO Hierarchy**

| | |
|---|---|
| **Tokens** | Implement a single data bit transmitted on the bus, or the wake-up event. |
| **Flags** | Comprised of eight tokens (bits) which convey the direction and meaning of the next group of bits (if any), which may be transmitted. |
| **Blocks** | Data following the command and Transmit flags. They incorporate both a byte count and a checksum to ensure proper data transmission. |
| **Packets** | Bytes forming the core of the block without the count and CRC. They are either the input or output parameters of an AT88SA10HS command or status information from AT88SA10HS. |

See applications notes on the Atmel website for more details on how to use any microprocessor to easily generate the signaling necessary to send these values to the chip.

## 2.1 IO Tokens

There are a number of IO **tokens** that may be transmitted along the bus:

Input: (To AT88SA10HS)

| | |
|---|---|
| Wake | Wake the AT88SA10HS up from sleep (low power) state |
| Zero | Send a single bit from system to the AT88SA10HS with a value of zero |
| One | Send a single bit from system to the AT88SA10HS with a value of one |

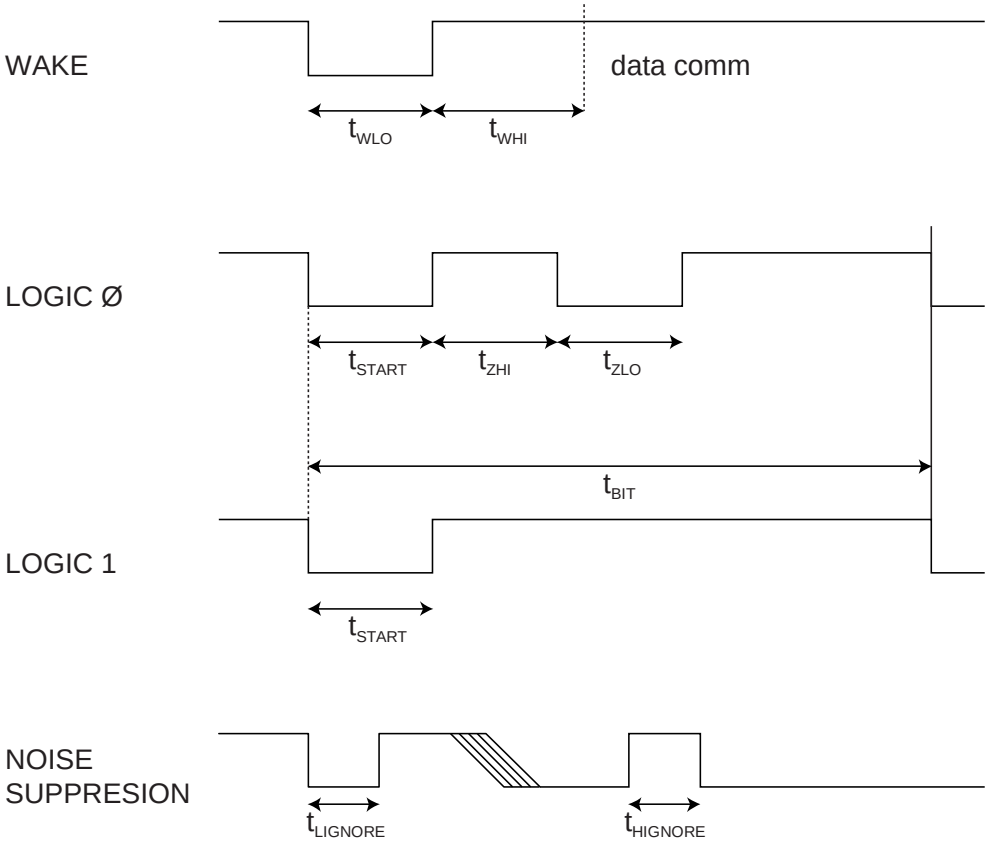
Output: (From AT88SA10HS)

| | |
|---|---|
| ZeroOut | Send a single bit from the AT88SA10HS to the system with a value of zero |
| OneOut | Send a single bit from the AT88SA10HS to the system with a value of one |

The waveforms are the same in either direction, however there are some differences in timing based on the expectation that the host has a very accurate and consistent clock while AT88SA10HS has significant variation in its internal clock generator due to normal manufacturing and environmental fluctuations.

The bit timings are designed to permit a standard UART running at 230.4 K baud to transmit and receive the tokens efficiently. Each byte transmitted or received by the UART corresponds to a single bit received or transmitted by the AT88SA10HS. See application notes on the Atmel website for more details.

## 2.2 AC Parameters



## 3. Absolute Maximum Ratings*

| | |
|---|---|
| Operating temperature | −40° C to +85° C |
| Storage temperature | −65° C to + 150° C |
| Voltage on any pin with respect to ground | − 0.5 to $V_{CC}$+0.5 V |

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

## 4. AC Parameters

**Table 4-1. AC Parameters**

| Parameter | Symbol | Direction | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Wake low duration | $t_{WLO}$ | To AT88SA10HS | 60 | | - | µs | Signal can be stable in either high or low levels during extended sleep intervals. |
| Wake delay to data comm. | $t_{WHI}$ | To AT88SA10HS | 2.5 | | 45 | ms | Signal should be stable high for this entire duration. $t_{WHI}$ must not exceed $t_{TIMEOUT}$ or the chip will transition to sleep. |
| Start pulse duration | $t_{START}$ | To AT88SA10HS | 4.1 | 4.34 | 4.56 | µs | |
| | | From AT88SA10HS | 4.6 | 6.0 | 8.6 | µs | |
| Zero transmission high pulse | $t_{ZHI}$ | To AT88SA10HS | 4.1 | 4.34 | 4.56 | µs | |
| | | From AT88SA10HS | 4.6 | 6.0 | 8.6 | µs | |
| Zero transmission low pulse | $t_{ZLO}$ | To AT88SA10HS | 4.1 | 4.34 | 4.56 | µs | |
| | | From AT88SA10HS | 4.6 | 6.0 | 8.6 | µs | |
| Bit time‡ | $t_{BIT}$ | To AT88SA10HS | 37 | 39 | - | µs | If the bit time exceeds $t_{TIMEOUT}$ then AT88SA10HS will enter sleep mode and the Wake token must be resent. |
| | | From AT88SA10HS | 41 | 54 | 78 | µs | |
| Turn around delay | $t_{TURNAROUND}$ | From AT88SA10HS | 28 | 60 | 95 | µs | AT88SA10HS will initiate the first low going transition after this time interval following the end of the Transmit flag. |
| | | To AT88SA10HS | 15µs | | 45ms | | After AT88SA10HS transmits the last bit of a block, system must wait this interval before sending the first bit of a flag. |
| High side glitch filter @ active | $t_{HIGNORE\_A}$ | To AT88SA10HS | 45 | | | ns | Pulses shorter than this in width will be ignored by the chip, regardless of its state when active. |
| Low side glitch filter @ active | $t_{LIGNORE\_A}$ | To AT88SA10HS | 45 | | | ns | Pulses shorter than this in width will be ignored by the chip, regardless of its state when active |
| Low side glitch filter @ sleep | $t_{LIGNORE\_S}$ | To AT88SA10HS | 500 | | | ns | Pulses shorter than this in width will be ignored by the chip when in sleep mode. |
| IO Timeout | $t_{TIMEOUT}$ | To AT88SA10HS | 45 | 65 | 85 | ms | See Section 5.4.1. |
| Watchdog reset | $t_{WATCHDOG}$ | To AT88SA10HS | 3 | 4 | 5.7 | s | Max. time from Wake until chip is forced into sleep mode. See Section 5.5. |
| Pause Length | $t_{PAUSE}$ | - | 18 | 25 | 32 | ms | Duration during which the chip will ignore IO on the bus. See PauseShort command, Section 6.7. |

# 5. DC Parameters

**Table 5-1. DC Parameters**

| Parameter | Symbol | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|
| Operating temperature | $T_A$ | -40 | | 85 | °C | |
| Power supply voltage | $V_{cc}$ | 2.7 | | 5.25 | V | |
| Fuse burning voltage | $V_{BURN}$ | 3.0 | | 5.25 | V | Voltage applied to $V_{cc}$ pin. See Section 6.6. |
| Active power supply current | $I_{CC}$ | | - | 6 | mA | |
| Sleep power supply current @ -40° C to 55° C | $I_{SLEEP}$ | | | 150 | nA | When chip is in sleep mode, $V_{cc}$ = 5.25V, Vsig = 0.0 to 0.3V or Vsig = $V_{cc}$-0.3V to $V_{cc}$ |
| Sleep power supply current @ 85° C | $I_{SLEEP}$ | | | 1 | µA | When chip is in sleep mode, $V_{cc}$ = 5.25V, Vsig = 0.0 to 0.3V or Vsig = $V_{cc}$-0.3V to $V_{cc}$ |
| Input low voltage @ $V_{cc}$ = 5.25 V | $V_{IL}$ | -0.5 | | 0.75 | V | Voltage levels for Wake token when chip is in sleep mode. |
| Input low voltage @ $V_{cc}$ = 2.7 V | $V_{IL}$ | -0.5 | | 0.5 | V | Voltage levels for Wake token when chip is in sleep mode. |
| Input high voltage @ $V_{cc}$ = 5.25 V | $V_{IH}$ | 1.5 | | 5.25 | V | Voltage levels for Wake token when chip is in sleep mode. |
| Input high voltage @ $V_{cc}$ = 2.7 V | $V_{IH}$ | 1.25 | | 3.0 | V | Voltage levels for Wake token when chip is in sleep mode. |
| Input low voltage when active | $V_{IL}$ | -0.5 | | 0.5 | V | When chip is in active mode, $V_{cc}$ = 2.7 – 5.25V |
| Input high voltage when active | $V_{IH}$ | 1.2 | | 5.25 | V | When chip is in active mode, $V_{cc}$ = 2.7 – 5.25V |
| Output low voltage | $V_{OL}$ | | | 0.4 | V | When chip is in active mode, $V_{cc}$ = 2.7 – 5.25V |
| Maximum input voltage | $V_{MAX}$ | | | 5.25 | V | |
| ESD | $V_{ESD}$ | | 4 | | KV | Human body model, Sig and $V_{cc}$ pins. |

## 5.1 IO Flags

The system is always the bus master, so before any IO transaction, the system must send an 8-bit **flag** to the chip to indicate the IO operation that is to be performed, as follows:

| *Value* | *Name* | *Meaning* |
|---|---|---|
| 0x66 | Command | After this flag, the system starts sending a command block to the chip. The first bit of the block can follow immediately after the last bit of the flag |
| 0x99 | Transmit | After a turn-around delay, the chip will start transmitting the response for a previously transmitted command block |
| 0xCC | Sleep | Upon receipt of a sleep flag, the chip will enter a low power mode until the next Wake token is received |

All other values are reserved and will be ignored.

Note: The values of flag for the AT88SA10HS host are different from that of the two clients, the AT88SA100S and AT88SA102S. In this manner, both AT88SA102S (or AT88SA100S) and AT88SA10HS can share the same communications pin on the system controller. While the AT88SA10HS will wake up when communications are sent to the client, it will ignore all such transactions.

It is possible that data values transmitted to a client authentication chip (either the AT88SS100S or the AT88SA102S) could be interpreted by the AT88SA10HS host chip as a legal transmit flag. In this case there could be a bus conflict as both the host and client chips drive the signal wire at the same time. To prevent this, the PauseShort command should be used to prevent the AT88SA10HS host chip from looking at the signal wire during any IO transaction to the client.

### 5.1.1 Command Timing

After a command flag is transmitted, a command block should be sent to the chip. During parsing of the parameters and subsequent execution of a properly received command, the chip will be busy and not respond to transitions on the signal pin. The delays for these operations are listed in the table below:

**Table 5-2. Command Timing** (Guaranteed by design; not tested)

| Parameter | Symbol | Max | Unit | Notes |
|---|---|---|---|---|
| Parsing Delay | $t_{PARSE}$ | 100 | µs | Delay to check CRC and parse opcode and parameters before an error indication will be available. |
| Host0Delay | $t_{EXEC\_HOST0}$ | 13 | ms | Delay to execute any of the HOST0 command. |
| Host1Delay | $t_{EXEC\_HOST1}$ | 7 | ms | Delay to execute any of the HOST1 command. |
| Host2Delay | $t_{EXEC\_HOST2}$ | 0.5 | ms | Delay to execute any of the HOST2 command. |
| MemoryDelay | $t_{EXEC\_READ}$ | 3 | ms | Delay to execute Read command. |
| SecureDelay | $t_{EXEC\_SECURE}$ | 36 | ms | Max delay to execute BurnSecure command. See Section 6.6 for more details. |
| PersonalizeDelay | $t_{PERSON}$ | 13 | ms | Delay to execute GenPersonalizationKey. |

In this document, $t_{EXEC}$ is used as shorthand for the delay corresponding to whatever command has been sent to the chip.

### 5.1.2 Transmit Flag

The Transmit flag is used to turn around the signal so that the AT88SA10HS can send data back to the system, depending on its current state. The bytes that the AT88SA10HS returns to the system depend on its current state as follows:

**Table 5-3. Return Codes**

| State Description | Error/Status | Description |
|---|---|---|
| After Wake, but prior to first command | 0x11 | Indication that a proper Wake token has been received by AT88SA10HS. |
| After successful command execution | – | Return bytes per “Output Parameters” in Command section of this document. In some cases this is a single byte with a value of 0x00 indicating success. The Transmit flag can be re-sent to AT88SA10HS repeatedly if a re-read of the output is necessary. |
| Execution error | 0x0F | Command was properly received but could not be executed by AT88SA10HS. Changes in the AT88SA10HS state or the value of the command bits must happen before it is re-attempted. |
| After CRC or other communications error | 0xFF | Command was *not* properly received by AT88SA10HS and should be re-issued by the system. No attempt was made to execute the command. |

The AT88SA10HS always transmits complete blocks to the system, so in the above table, the status/error bytes result in four bytes going to the system – count, error, CRC x 2.

After receipt of a command block, the AT88SA10HS will parse the command for errors, a process which takes $t_{PARSE}$ (See Section 5.1.1). After this interval the system can send a transmit token to the AT88SA10HS – if there was an error, the AT88SA10HS will respond with an error code. If there is no error, the AT88SA10HS internally transitions automatically from $t_{PARSE}$ to $t_{EXEC}$ and will not respond to any transmit tokens until both delays are complete.

### 5.1.3 Sleep Flag

The sleep flag is used to transition the AT88SA10HS to the low power state, which causes a complete reset of the internal command engine of the AT88SA10HS and input/output buffer. It can be sent to AT88SA10HS at any time when AT88SA10HS will accept a flag.

To achieve the specified $I_{SLEEP}$, Atmel recommends that the input signal be brought below $V_{IL}$ when the chip is asleep. To achieve $I_{SLEEP}$ if the sleep state of the input pin is high, the voltage on the input signal should be within 0.3V of $V_{CC}$ to avoid additional leakage on the input circuit of the chip.

The system must calculate the total time required for all commands to be sent to the AT88SA10HS during a single session, including any inter-bit/byte delays. If this total time exceeds $t_{WATCHDOG}$ then the system must issue a partial set of commands, then a Sleep flag, then a Wake token, and finally after the Wake delay, issue the remaining commands.

## 5.2 IO Blocks

Commands are sent to the chip, and responses received from the chip, within a **block** that is constructed in the following way:

| *Byte Number* | *Name* | *Meaning* |
|---|---|---|
| 0 | Count | Number of bytes to be transferred to the chip in the block, including count, packet and checksum, so this byte should always have a value of (N+1). The maximum size block is 39 and the minimum size block is four. Values outside this range will cause unpredictable operation. |
| 1 to (N-2) | Packet | Command, parameters and data, or response. See Section 6 for more details. |
| N-1, N | Checksum | CRC-16 verification of the count and packet bytes. The CRC polynomial is 0x8005, the initial register value should be zero and after the last bit of the count and packet have been transmitted the internal CRC register should have a value that matches that in the block. The first byte transmitted (N-1) is the least significant byte of the CRC value so the last byte of the block is the most significant byte of the CRC. |

## 5.3 IO Flow

The general IO flow for the commands is as follows:

1. System sends Wake token
2. System sends transmit flag
3. Receive 0x11 value from AT88SA10HS to verify proper wakeup synchronization.
4. System sends command flag
5. System sends complete command block
6. System waits $t_{PARSE}$ for the AT88SA10HS to check for command formation errors
7. System sends transmit flag. If command format is OK, the AT88SA10HS ignores this flag because the computation engine is busy. If there was an error, the AT88SA10HS responds with an error code
8. System waits $t_{EXEC}$, see Section 5.1.1
9. System sends transmit flag
10. Receive output block from the AT88SA10HS, system checks CRC
11. If CRC from AT88SA10HS is incorrect, indicating transmission error, system resends transmit flag
12. System sends sleep flag to the AT88SA10HS

Where the command in question has a short execution delay the system should omit steps six, seven and eight and replace this with a wait of duration $t_{PARSE}$ + $t_{EXEC}$.

## 5.4 Synchronization

Because the communications protocol is half duplex, there is the possibility that the system and the AT88SA10HS will fall out of synchronization with each other. In order to speed recovery, AT88SA10HS implements a timeout that forces the AT88SA10HS to sleep.

### 5.4.1 IO Timeout

After a leading transition for any data token has been received, AT88SA10HS will expect the remaining bits of the token to be properly received by the chip within the $t_{TIMEOUT}$ interval. Failure to send enough bits or the transmission of an illegal token (a low pulse exceeding $t_{ZLO}$) will cause the chip to enter the sleep state after the $t_{TIMEOUT}$ interval.

The same timeout applies during the transmission of the command block. After the transmission of a legal command flag, the IO timeout circuitry is enabled until the last expected data bit is received. Note that the timeout counter is reset after every legal token, so the total time to transmit the command may exceed the $t_{TIMEOUT}$ interval while the time between bits may not.

In order to limit the active current if the AT88SA10HS is inadvertently awakened, the IO timeout circuitry is also enabled when the AT88SA10HS receives a wake-up. If the first token does not come within the $t_{TIMEOUT}$ interval, the AT88SA10HS will go back to the sleep mode without performing any operations.

The IO Timeout circuitry is disabled when the chip is busy executing a command.

### 5.4.2 Synchronization Procedures

When the system and the AT88SA10HS fall out of synchronization, the system will ultimately end up sending a Transmit flag which will not generate a response from the AT88SA10HS. The system should implement its own timeout which waits for $t_{TIMEOUT}$ during which time the AT88SA10HS should go to sleep automatically. At this point, the system should send a Wake token and after $t_{WLO}$ + $t_{WHI}$, a Transmit token. The 0x11 status indicates that the resynchronization was successful.

It may be possible that the system does not get the 0x11 code from the AT88SA10HS for one of the following reasons:

1. The system did not wait a full $t_{TIMEOUT}$ delay with the IO signal idle in which case the Atmel AT88SA10HS may have interpreted the Wake token and Transmit flag as data bits. Recommended resolution is to wait twice the $t_{TIMEOUT}$ delay and re-issue the Wake token.
2. The AT88SA10HS went into the sleep mode for some reason while the system was transmitting data. In this case, the AT88SA10HS will interpret the next data bit as a Wake token, but ignore some of the subsequently transmitted bits during its wake-up delay. If any bytes are transmitted after the wake-up delay, they may be interpreted as a legal flag, though the following bytes would not be interpreted as a legal command due to an incorrect count or the lack of a correct CRC. Recommended resolution is to wait the $t_{TIMEOUT}$ delay and re-issue the Wake token.
3. There are some internal error conditions within the AT88SA10HS which will be automatically reset after a $t_{WATCHDOG}$ interval, see below. There is no way to externally reset the AT88SA10HS – the system should leave the IO pin idle for this interval and issue the Wake token.

## 5.5 Watchdog Failsafe

After the Wake token has been received by the AT88SA10HS, a watchdog counter is started within the chip. After $t_{WATCHDOG}$, the chip will enter sleep mode, regardless of whether it is in the middle of execution of a command and/or whether some IO transmission is in progress. There is no way to reset the counter other than to put the chip to sleep and wake it up again.

This is implemented as a fail-safe so that no matter what happens on either the system side or inside the various state machines of the AT88SA10HS including any IO synchronization issue, power consumption will fall to the low sleep level automatically.

## 5.6 Byte and Bit Ordering

The AT88SA10HS is a little-endian chip:

- All multi-byte aggregate elements within this spec are treated as arrays of bytes and are processed in the order received
- Data is transferred to/from the AT88SA10HS least significant bit first on the bus
- In this document, the most significant bit and/or byte appears towards the left hand side of the page

# 6. Commands

The command packet is broken down in the following way:

| *Byte* | *Name* | *Meaning* |
|---|---|---|
| 0 | Opcode | The command code |
| 1 | Param1 | The first parameter – always present |
| 2-3 | Param2 | The second parameter – always present |
| 4 + | Data | Optional remaining input data |

If a command fails because the CRC within the block is incorrect or there is some other communications error, then immediately after $t_{PARSE}$ the system will be able to retrieve an error response block containing a single byte packet. The value of that byte will be all ones. In this situation, the system should re-transmit the command block including the proceeding transmit flag – providing there is sufficient time before the expiration of the watchdog timeout.

If the opcode is invalid, one of the parameters is illegal, or the AT88SA10HS is in an illegal state for the execution of this command, then immediately after $t_{PARSE}$ the system will be able to retrieve an error response block containing a single byte packet. The value of that byte will be 0x0F. In this situation, the condition must be corrected before the (modified) command is sent back to the AT88SA10HS.

If a command is received successfully, the system will be able to retrieve the output block as described in the individual command descriptions below after the appropriate execution delay.

In the individual command description tables following, the “Size” column describes the number of bytes in the parameter documented in each particular row. The total size of the block for each of the commands is fixed, though that value is different for each command. If the block size for a particular command is incorrect, the chip will not attempt the command execution and returns an error.

## 6.1 HOST0

Concatenates the key stored in AT88SA10HS with an input 256-bit challenge and generates the digest of this message. The result is left in internal memory and cannot be read. In general, the challenge should be a random number generated by the host system, which will be sent to both the host (AT88SA10HS) and client (AT88SA100S or AT88SA102S).

**Table 6-1. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | HOST0 | 1 | 0x08 |
| *Param1* | Overwrite | 1 | If non-zero, overwrite part of internally generated key with secret fuses. |
| *Param2* | KeyID | 2 | The internal key to be used to generate the digest. |
| *Data* | Challenge | 32 | Challenge to be sent to the client AT88SA100S or AT88SA102S. |

**Table 6-2. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | Upon successful completion of HOST0, a value of zero will be returned by AT88SA10HS. |

The 512-bit message block that will be hashed with the SHA-256 algorithm will consist of:

256-bits key[KeyID]
256-bits challenge

If the overwrite parameter is 0, then the 512-bit message block that will be hashed using the SHA-256 algorithm will consist of:

256-bits key[KeyID]
256-bits challenge

If the overwrite parameter has a value of 0x01, then the 512-bit message block that will be hashed using the SHA-256 algorithm will consist of:

192-bits key[KeyID]
64-bits Fuse[0-63]
256-bits challenge

All other values of the overwrite parameter are not recommended for use.

## 6.2 HOST1

Completes the two block SHA-256 digest started by HOST0 and leaves the resulting digest within the internal memory of the AT88SA10HS. This command returns an error if HOST0 has not been successfully run previously within this Wake cycle.

As a security precaution, this command does not return the digest. A subsequent command is required to compare the response generated by the client with the one generated by the host.

**Table 6-3. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | HOST1 | 1 | 0x40 |
| *Param1* | Mode | 1 | Controls composition of message, see below for details. |
| *Param2* | Zero | 2 | Must be 0x0000. |
| *Data* | OtherInfo | 13 | Input portion of message to be digested. |

**Table 6-4. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | Upon successful completion of HOST1, a value of zero will be returned by AT88SA10HS. |

The contents of the second block to be digested are listed below.

Note: To simplify this documentation; the bit addresses for OtherInfo are listed in the table below

| Size | Source | Notes |
|---|---|---|
| 32-bits | OtherInfo[0-31] | Opcode, param1 and param2 values sent to AT88SA100S/AT88SA102S. |
| 64-bits | Fuse[0-63] | If enabled by bit five of the input mode parameter and if Fuse[87] is burned, else forced to zero. |
| 24-bits | OtherInfo[32-55] | Status fuse values from ATSA100S/AT88SA102S, or zeros. |
| 8-bits | Fuse[88-95] | Fuse MfrID, should match between AT88SA10HS and AT88SA100S/AT88SA102S. |
| 32-bits | OtherInfo[56-87] | Fuse SN from AT88SA100S/AT88SA102S (Fuse[96-127]), or zeros. |
| 16-bits | ROM MfrID | Should match between AT88SA10HS and AT88SA100S/AT88SA102S. |
| 16-bits | OtherInfo[88-103] | ROM SN from AT88SA100S/AT88SA102S, or zeros. |

These bits are followed by the necessary ‘1’ bit, ‘0’ padding and 64-bit length as specified in the SHA-256 specification.

### 6.2.1.1 Mode Encoding

Bit five of the mode is used to indicate whether or not the secret fuse bits are to be included in the calculation. The remaining bits of the mode field are ignored by AT88SA10HS and should be zero.

**Table 6-5. Mode Encoding**

| Bit[5] | Fuse Block |
|---|---|
| 0 | No fuse values inserted. |
| 1 | Insert the values of Fuse[0-63] in the message. |

If Fuse[87] has not been burned, then the values of Fuse[0-63] will be replaced by zeros in the above message generation step as a security measure.

## 6.3 HOST2

Compares the value previously generated by the AT88SA10HS using HOST0 and HOST1 with that on the input stream coming from the client and returns status to indicate whether or not the two matched. This command returns an error if HOST1 has not been previously successfully run within this Wake cycle.

If the two digests do not match, the AT88SA10HS provides no information as to the source of the mismatch, which must be deduced from the inputs to the three HOSTX commands. On a match failure, the entire set of HOST0, HOST1, and HOST2 commands must be re-executed – HOST2 cannot be repeatedly executed.

**Table 6-6. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | HOST2 | 1 | 0x80 |
| *Param1* | Zero1 | 1 | Must be 0x00. |
| *Param2* | Zero2 | 2 | Must be 0x0000. |
| *Data* | ClientResponse | 32 | Response from the client. |

**Table 6-7. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | If the input ClientResponse matches the internally generated response, a value of zero will be returned by AT88SA10HS after a $T_{HOST}$ delay. If the two digests do *not* match, a value of 0x0F will be returned after a $T_{HOST}$ delay |

## 6.4 Read

Reads 4-bytes from Fuse or ROM; returns an error if an attempt is made to read any fuses or ROM locations which are illegal.

**Table 6-8. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | Read | 1 | 0x02 |
| *Param1* | Mode | 1 | Fuse or ROM. |
| *Param2* | Address | 2 | Which 4-bytes within array. Only bits zero and one are used, all others must be zeros. |
| *Data* | Ignored | 0 | |

**Table 6-9. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Contents | 4 | The contents of the specified memory location. |

**Table 6-10. Mode Encoding**

| Name | Value | Notes |
|---|---|---|
| ROM | 0x00 | Reads four bytes from the ROM. Bit one of the address parameter must be zero. |
| Fuse | 0x01 | Reads the value of 32-fuses. Bit one of the address parameter must be one. |

## 6.5 GenPersonalizationKey

Loads a personalization key into internal memory and then uses that key along with an input seed to generate a decryption digest using SHA-256. Neither the key nor the decryption digest can be read from the chip. Upon completion, an internal bit is set indicating that a secure personalization digest has been loaded and is ready to use by the BurnSecure command. This bit is cleared (and the digest lost) when the watchdog timer expires or the power is cycled.

This command will fail if Fuse[87] has been burned.

**Table 6-11. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | GenPers | 1 | 0x20 |
| *Param1* | Zero | 1 | Must be 0x00. |
| *Param2* | KeyID | 2 | Identification number of the personalization key to be loaded. |
| *Data* | Seed | 16 | Seed for digest generation. The least significant bit of the last byte is ignored by AT88SA10HS. |

**Table 6-12. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | Upon successful execution, a value of 0 will be returned by Atmel AT88SA10HS. |

The SHA-256 message body used to create the resulting digest internally stored in the chip consists of the following 512-bits:

| | |
|---|---|
| 256-bits | PersonalizeKey[KeyID] |
| 64-bits | Fixed value of all ones |
| 127-bits | Seed from input stream |
| 1-bits | ‘1’ pad |
| 64-bits | length of message in bits, fixed at 447 |

## 6.6 BurnSecure

Burns any combination of the first 88-fuse bits. Verification that the proper secret fuse bits have been burned must occur using the MAC command – there is no way to read the values in the first 64-fuses to verify their state. The 24-status fuses can be verified with the read command.

The fuses to be burned are specified by the 88-bit input map parameter. If a bit in the map is set to a '1', then the corresponding fuse is burned. If a bit in the map parameter is zero, then the corresponding fuse is left in its current state. The first bit sent to AT88SA10HS corresponds to Fuse[0] and so on up to Fuse[87].

Note: Since a '1' bit in the map parameter results in a '0' data value in the actual fuse array, the value in the Map parameter should be the inverse of the desired secret or status value. See Section 1.2 for more details

To facilitate secure personalization of the AT88SA10HS, this map may be encrypted before being sent to the chip. If this mode is desired, then the Decrypt parameter should be set to one in the input parameter list. The decryption (transport) key is computed by the GenPersonalizationKey command, which must have been run immediately prior to the execution of BurnSecure. In this case, prior to burning any fuses, the input Map parameter is XOR'd with the first 88-bits of that digest from the GenPersonalizationKey command. The GenPersonalizationKey and BurnSecure commands must be run within a single Wake cycle prior to the expiration of the watchdog timer.

The power supply pin must meet the $V_{BURN}$ specification during the entire BurnSecure command in order to burn fuses reliably. If $V_{CC}$ is greater than or equal to 3.7V, then the BurnTime parameter should be set to 0x00 and the internal burn time will be 250µs. If Vcc is less than 3.7V but greater than $V_{BURN}$ then the BurnTime parameter should be set to 0xFFFF and the internal burn time will be 262ms per fuse bit burned. The chip does *not* internally check the supply voltage level.

The total BurnSecure execution delay is directly proportional to the total number of fuses being burned. If $V_{CC}$ is less than 3.7V, then the total BurnSecure execution time may exceed the interval remaining before the expiration of the watchdog timer. In this case, the BurnSecure command should be run repeatedly, with each repetition burning only as many fuses as there is time available. The system software is responsible for counting the number of '1' bits in the clear-text version of the map parameter sent to the chip – no error is returned if the fuse burn count is too high. Other than Fuse[87] (see below), the fuses may be burned in any order.

Prior to execution of BurnSecure, AT88SA10HS verifies that Fuse[87] is un-burned. If it has been burned, then the BurnSecure command will return an error. Fuse[87] must be burned during the last repetition of BurnSecure as it cannot be individually burned with BurnFuse.

There are a series of very small intervals during $t_{EXEC\_SECURE}$ when the fuse element is actually being burned. During this interval, the power supply must not be removed and the watchdog timer must not be allowed to expire, or the fuse may end up in a state where it reads as un-burned but cannot be burned.

**Table 6-13. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | BURNSECURE | 1 | 0x10 |
| *Param1* | Decrypt | 1 | If 1, decrypt Map data before usage. If 0, the map is transmitted in plain text. |
| *Param2* | BurnTime | 2 | Must be 0x0000 if $V_{CC}$ >=3.7 V; must be 0xFFFF otherwise. |
| *Data* | Map | 11 | Which fuses to burn, may be encrypted. |

**Table 6-14. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | Upon successful execution, a value of zero will be returned by AT88SA10HS. |

This command takes a constant time to execute regardless of the number of fuses being burned.

## 6.7 PauseShort

Forces the chip into a busy mode for a period of $t_{PAUSE}$. During execution of this command the chip will ignore all activity on the IO signal. This command is used to prevent bus conflicts in a system that also includes one or more AT88SA100S or AT88SA102S client chips sharing the same signal wire.

**Table 6-15. Input Parameters**

| | Name | Size | Notes |
|---|---|---|---|
| *Opcode* | PAUSESHORT | 1 | 0x00 |
| *Param1* | Ignored | 1 | Must be 0x00. |
| *Param2* | Ignored | 2 | Must be 0x0000. |
| *Data* | Ignored | 0 | |

**Table 6-16. Output Parameters**

| Name | Size | Notes |
|---|---|---|
| Success | 1 | After a delay of $t_{PAUSE}$, the AT88SA10HS will return a value of zero in response to a transmit flag. |

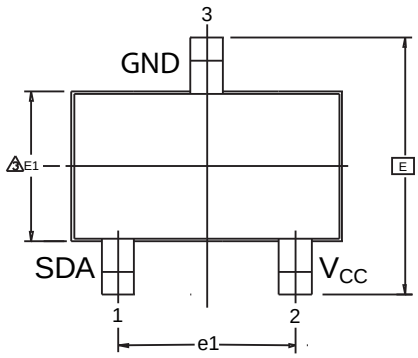
# 7. Pinout

**Table 7-1. Pin Definitions**

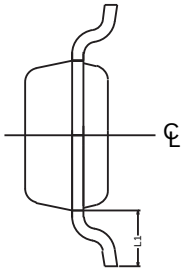
| SOIC | Pin # | Name | Description |
|---|---|---|---|
| 5 | 1 | Signal | IO channel to the system, open drain output. It is expected that an external pull-up resistor will be provided to pull this signal up to $V_{CC}$ for proper communications. When the chip is not in use this pin can be pulled to either $V_{CC}$ or GND. |
| 8 | 2 | $V_{CC}$ | Power supply, 2.7 – 5.25V. This pin should be bypassed with a high quality 0.1µF capacitor close to this pin with a short trace to GND. See applications notes on the Atmel website for more details. |
| 4 | 3 | GND | Connect to system ground. |
| 1,2,3,6,7 | -- | NC | Not connected. |

# 8. Package Drawing

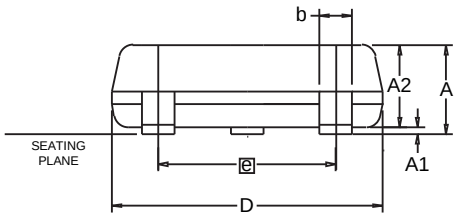
## 3TS1 – 3-lead Shrink SOT




Top View




End View




Side View

Notes:
1. Dimension D does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.25mm per end. Dimension E1 does not include interlead flash or protrusion. Interlead flash or protrusion shall not exceed 0.25mm per side.
2. The package top may be smaller than the package bottom. Dimensions D and E1 are determined at the outermost extremes of the plastic body exclusive of mold flash, tie bar burrs, gate burrs and interlead flash, but including any mismatch between the top and bottom of the plastic body.
3. These dimensions apply to the flat section of the lead between 0.08 mm and 0.15mm from the lead tip.

This drawing is for general information only. Refer to JEDEC Drawing TO-236, Variation AB for additional information.

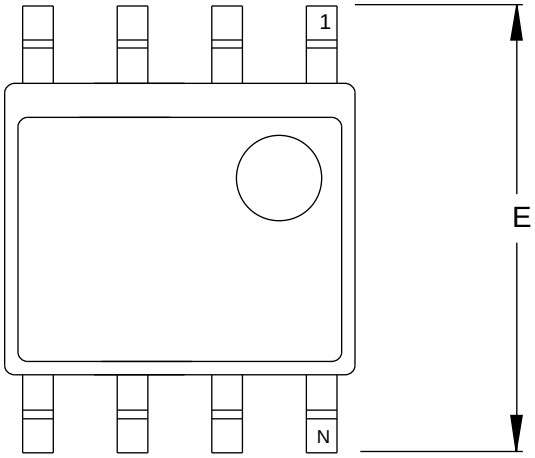
**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.89 | - | 1.12 | |
| A1 | 0.01 | - | 0.10 | |
| A2 | 0.88 | - | 1.02 | |
| D | 2.80 | 2.90 | 3.04 | 1,2 |
| E | 2.10 | - | 2.64 | |
| E1 | 1.20 | 1.30 | 1.40 | 1,2 |
| L1 | | 0.54 REF | | |
| e1 | | 1.90 BSC | | |
| b | 0.30 | - | 0.50 | 3 |

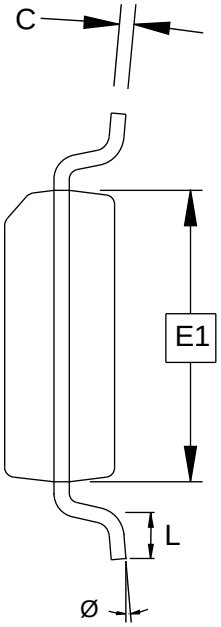
12/11/09

| Atmel | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| Package Drawing Contact: packagedrawings@atmel.com | **3TS1, 3-lead, 1.30mm Body, Plastic Thin** Shrink Small Outline Package (Shrink SOT) | TBG | 3TS1 | B |

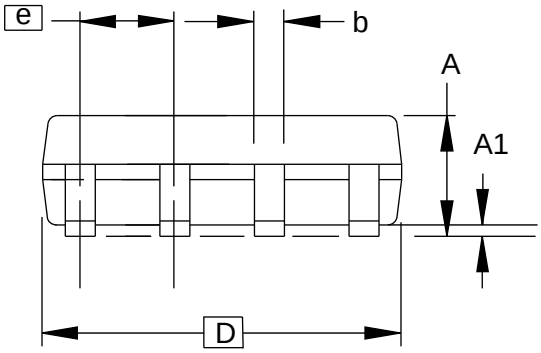
## 8S1 – 8-lead JEDEC SOIC




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 | 1.35 | – | 1.75 | |
| A1 | 0.10 | – | 0.25 | |
| b | 0.31 | – | 0.51 | |
| C | 0.17 | – | 0.25 | |
| D | 4.80 | – | 5.05 | |
| E1 | 3.81 | – | 3.99 | |
| E | 5.79 | – | 6.20 | |
| e | 1.27 BSC | | | |
| L | 0.40 | – | 1.27 | |
| Ø | 0° | – | 8° | |

6/22/11

| Atmel<br>Package Drawing Contact:<br>packagedrawings@atmel.com | **TITLE**<br>8S1, 8-lead (0.150” Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC) | **GPC**<br>SWB | **DRAWING NO.**<br>8S1 | **REV.**<br>G |
|---|---|---|---|---|

# 9. Ordering Codes

### Atmel AT88SA10HS Ordering Information

| Atmel Ordering Code | Package Type | Voltage Range | Temperature Range |
|---|---|---|---|
| AT88SA10HS-TSU-T | SOT, Tape and Reel | 2.7 V–5.25 V | Green compliant (exceeds RoHS)/Industrial (−40°C to 85°C) |
| AT88SA10HS-SH-T | SOIC, Tape and Reel | 2.7 V–5.25 V | Green compliant (exceeds RoHS)/Industrial (−40°C to 85°C) |

# 10. Revision History

| Doc. Rev. | Date | Comments |
|---|---|---|
| 8596H | 09/2012 | Remove TSSOP package option.<br>Update Atmel logo and disclaimer page. |
| 8595G | 05/2012 | Not recommended for new designs; Replaced by ATSHA204. |
| 8595G | 09/2011 | Correct references and section numbers.<br>Section 5.1.3, Sleep Flag, change " within 0.5V of $V_{CC}$" to "within 0.3V of $V_{CC}$". |
| 8595F | 08/2010 | Update IO Timeout description. |
| 8595E | 06/2010 | Update to Table 3:  AC Parameters. |
| 8595D | 05/2010 | Expansion of IO Timeout specification. |
| 8595C | 04/2010 | Added 8ld TSSOP. |
| 8595B | 02/2010 | Updated parameter tables and added 8ld SOIC. |
| 8595A | 04/2009 | Initial document release. |

Atmel | Enabling Unlimited Possibilities®

**Atmel Corporation**
1600 Technology Drive
San Jose, CA 95110
USA
**Tel:** (+1)(408) 441-0311
**Fax:** (+1)(408) 487-2600
www.atmel.com

**Atmel Asia Limited**
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
**Tel:** (+852) 2245-6100
**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
**Tel:** (+49) 89-31970-0
**Fax:** (+49) 89-3194621

**Atmel Japan G.K.**
16F Shin-Osaki Kangyo Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
**Tel:** (+81)(3) 6417-0300
**Fax:** (+81)(3) 6417-0370