Features

- **Secure key storage to complement AT88SA100S and AT88SA102S Devices**
- **Superior SHA-256 Hash Algorithm**
- **Guaranteed Unique 48 bit Serial Number**
- **High speed single wire interface, optionally shared with client**
- **Supply Voltage: 2.5 – 5.5V**
- **1.8V – 5.5V communications voltage**
- **<100nA Sleep Current**
- **4KV ESD protection**
- **Multi-level hardware security**
- **Secure personalization**
- **Green compliant (exceeds RoHS) 3 pin SOT-23 package**

Applications

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

1. Introduction

The CryptoAuthentication family of chips are 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, refer to “AT88SA100S” and “AT88SA102S” Datasheets, along with the application notes dedicated to this product family.

The host CryptoAuthentication performs 3 separate operations (named HOST0, HOST1 & 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 the AT88SA10HS does not include a random number generator.

CryptoAuthentication™ Host Security Chip

AT88SA10HS

Preliminary

8595B–SMEM–09/09

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. Refer to 3.5 for more details.

1.1. Memory Resources

Fuse Block of 128 fuse bits that can be written through the 1 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 value is assigned by Atmel and is always the same for all chips of a particular model number. For the AT88SA10HS, this value is 0xFF FF. ROM MfrID can be read by accessing ROM bytes 0 & 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 & 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 1 when shipped from the Atmel factory and transition to a 0 when they are burned. These fuses are burned at system personalization and cannot be changed after that time.

Table 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 Enable	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, can't be modified in the field
96 → 127	Fuse SN	See Section 1.3. Set by Atmel, can't be modified in the field

BurnSecure Enable This fuse is used to prevent repetitive operation of the two personalization commands: GenPersonalizationKey and BurnSecure. This fuse is always burned by the BurnSecure command.

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 is 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 Enable 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 that 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 Atmel AT88SA10HS provides a secure hardware mechanism to validate responses to determine if they are authentic.

1.5. SHA-256 Computation

The 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 the AT88SA10HS. The secret fuses are conditionally appended, depending on the parameters to the HOST command.

For complete sample calculations, refer to”AT88SA100S” and/or “AT88SA102S” Datasheets.

1.6. Security Features

The 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 the 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 the AT88SA10HS take place over a single asynchronously timed wire using a pulse count scheme. The overall communications structure is a hierarchy:

Table 2. 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 the AT88SA10HS command or status information from the AT88SA10HS.

Refer to Applications Notes on Atmel's 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 0
- One — Send a single bit from system to the AT88SA10HS with a value of 1

Output: (From AT88SA10HS)

- ZeroOut — Send a single bit from the AT88SA10HS to the system with a value of 0
- OneOut — Send a single bit from the AT88SA10HS to the system with a value of 1

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 the 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 230.4K 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. Refer to Applications Notes on Atmel's website for more details.

2.2. AC Parameters

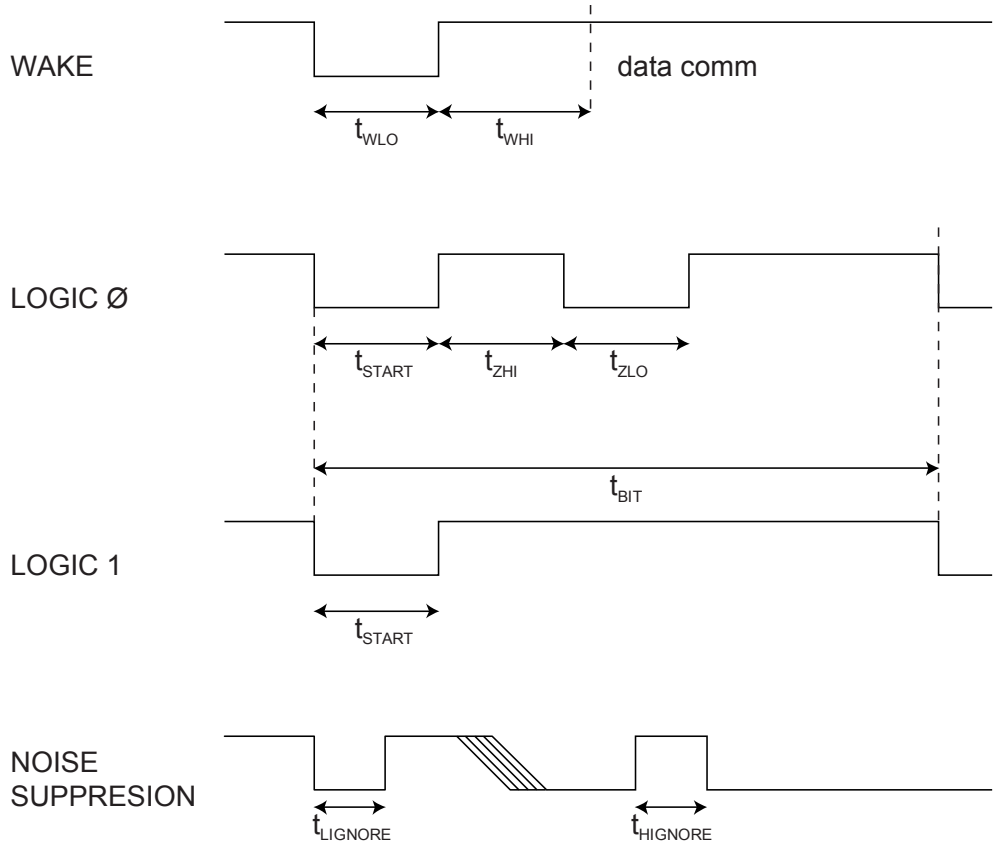


Table 3. AC Parameters

Parameter	Symbol	Direction	Min	Typ	Max	Unit	Notes
Wake Low Duration	t_{WLO}	To CryptoAuthentication	60		-	µs	Signal can be stable in either high or low levels during extended sleep intervals.
Wake Delay to Data Comm.	t_{WHI}	To CryptoAuthentication	1		-	ms	Signal should be stable high for this entire duration.
Start pulse duration	t_{START}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Zero transmission high pulse	t_{ZHI}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Zero transmission low pulse	t_{ZLO}	To CryptoAuthentication	4.1	4.34	4.56	µs	
		From CryptoAuthentication	4.62	6.0	8.6	µs	
Bit time‡	t_{BIT}	To CryptoAuthentication	37.1	39	-	µs	If the bit time exceeds $t_{TIMEOUT}$ then CryptoAuthentication will enter sleep mode and the wake token must be resent.
		From CryptoAuthentication	46.2	60	86	µs	
Turn around delay	$t_{TURNAROUND}$	From CryptoAuthentication	46.2	60	86	µs	CryptoAuthentication will initiate the first low going transition after this time interval following the end of the Transmit flag
		To CryptoAuthentication	46.2	60	86	µs	After CryptoAuthentication 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 CryptoAuthentication	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 CryptoAuthentication	45			ns	Pulses shorter than this in width will be ignored by the chip, regardless of its state when active
High side glitch filter @ sleep	$t_{HIGNORE\_S}$	To CryptoAuthentication	2			µs	Pulses shorter than this in width will be ignored by the chip when in sleep mode
Low side glitch filter @ sleep	$t_{LIGNORE\_S}$	To CryptoAuthentication	2			µs	Pulses shorter than this in width will be ignored by the chip when in sleep mode
IO Timeout	$t_{TIMEOUT}$	To CryptoAuthentication	7	10	13	ms	Starting as soon as 7ms up to 13ms after the initial signal transition of a token the chip will enter sleep if no complete & valid token is received.
Watchdog reset	$t_{WATCHDOG}$	To CryptoAuthentication	3	4	5.2	s	Max. time from wake until chip is forced into sleep mode. Refer to Section 3.5
Pause Length	t_{PAUSE}	-	18	25	32	ms	Duration during which the chip will ignore IO on the bus. See PauseShort command.

‡ START, ZLO, ZHI & BIT are designed to be compatible with a standard UART running at 230.4K baud for both transmit and receive.

3. DC Parameters

Table 4. DC Parameters

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Operating temperature	T_A	-40		85	°C	
Power Supply Voltage	V_{cc}	2.5		5.5	V	
Fuse Burning Voltage	V_{BURN}	3.0		5.5	V	Voltage is applied to V_{cc} pin
Active Power Supply Current	I_{CC}		-	10	mA	
Sleep Power Supply Current	I_{SLEEP}			100	nA	When chip is in sleep mode, V_{cc} = 3.7V, Vsig = 0.0 to 0.5V or Vsig = V_{cc}-0.5V to V_{cc}.
Input Low Voltage @ V_{cc} = 5.5V	V_{IL}	-0.5		.15 * V_{cc}	V	Voltage levels for wake token when chip is in sleep mode
Input Low Voltage @ V_{cc} = 2.5V	V_{IL}	-0.5		0.5	V	Voltage levels for wake token when chip is in sleep mode
Input High Voltage @ V_{cc} = 5.5V	V_{IH}	.25 * V_{cc}		6.0	V	Voltage levels for wake token when chip is in sleep mode
Input High Voltage @ V_{cc} = 2.5V	V_{IH}	1.0		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} = 2.5 – 5.5V
Input High Voltage when Active	V_{IH}	1.2		6.0	V	When chip is in active mode, V_{cc} = 2.5 – 5.5V
Output Low voltage	V_{OL}			0.4	V	When chip is in active mode, V_{cc} = 2.5 – 5.5V
Output Low current	I_{OL}			4	mA	When chip is in active mode, V_{cc} = 2.5 – 5.5V, V_{OL} = 0.4V
Maximum Input Voltage	V_{MAX}			V_{cc} + 0.5	V	
ESD	V_{ESD}		4		KV	Human Body Model, Sig & V_{cc} pins.

3.1. 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 that the values of flag for the AT88SA10HSS host are different from that of the two clients, the AT88SA100S and AT88SA102S. In this manner, both the AT88SA102S (or AT88SA100S) and AT88SA10HSS 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 theAT88SS100S 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.

3.1.1. 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. Command Timing

Parameter	Symbol	Min	Max	Unit	Notes
Parsing Delay	t_{PARSE}	0	50	µs	Delay to check CRC and parse opcode and parameters before an error indication will be available
HostDelay	t_{EXEC_HOST}	15	30	ms	Delay to execute any of the 4 HOST commands
MemoryDelay	t_{EXEC_READ}	50	100	µs	Delay to execute Read command
SecureDelay	t_{EXEC_SECURE}	13	26	ms	Max delay to execute BurnSecure command at Vcc > 4.5V. See Section 4.6 for more details.
PersonalizeDelay	t_{PERSON}	7	15	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.

3.1.2. 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 6. 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 the 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 resent to the AT88SA10HS repeatedly if a re-read of the output is necessary.
Execution error	0x0F	Command was properly received but could not be executed by the 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 the 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 4 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} (Refer to 3.1.1). 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.

3.1.3. Sleep Flag

The sleep flag is used to transition the AT88SA10HS to the low power state, which causes a complete reset of the AT88SA10HS’s internal command engine and input/output buffer. It can be sent to the AT88SA10HS at any time when the 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 0.5V 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.

3.2. 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 4. Values outside this range will cause unpredictable operation.
1 to (N-2)	Packet	Command, parameters and data, or response. Refer to Section 3.1.2 & Section 4 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 0 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.

3.3. 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 the 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}, Refer to Section 3.1.1.
9. System sends Transmit flag.
10. Receive output block from the AT88SA10HS, system checks CRC.
11. If CRC from the AT88SA10HS is incorrect, indication 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 6, 7 & 8 and replace this with a wait of duration $t_{PARSE} + t_{EXEC}$.

3.4. 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, the AT88SA10HS implements a timeout that forces the AT88SA10HS to sleep.

3.4.1. IO Timeout

After a leading transition for any data token has been received, the AT88SA10HS will expect another token to be transmitted within a $t_{TIMEOUT}$ interval. If the leading edge of the next token is not received within this period of time, the AT88SA10HS assumes that the synchronization with the host is lost and transitions to a sleep state.

After the AT88SA10HS receives the last bit of a command block, this timeout circuitry is disabled. If the command is properly formatted, then it is re-enabled with the first transmit token that occurs after t_{PARSE} + t_{EXEC}. If there is an error in the command, then it is re-enabled with the first transmit token that occurs after t_{PARSE}.

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

3.4.2. 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 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.

3.5. 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.

3.6. Byte & 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.

4. 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 1's. 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.

4.1. HOST0

Concatenates the key stored in the 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 7. 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 8. Output Parameters

Name	Size	Notes
Success	1	Upon successful completion of HOST0, a value of 0 will be returned by the 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.

4.2. 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 9. 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 0x00 00
Data	OtherInfo	13	Input portion of message to be digested.

Table 10. Output Parameters

Name	Size	Notes
Success	1	Upon successful completion of HOST1, a value of 0 will be returned by AT88SA10HS.

The contents of the second block to be digested are listed below. Note that 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 & param2 values sent to the AT88SA100S/AT88SA102S
64 bits	Fuse[0-63]	If enabled by bit 5 of the input mode parameter and if Fuse[87] is burned, else forced to 0
24 bits	OtherInfo[32-55]	Status fuse values from AT88SA102S, or 0's
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 0's
16 bits	ROM MfrID	Should match between AT88SA10HS and AT88SA100S/AT88SA102S.
16 bits	OtherInfo[88-103]	ROM SN from AT88SA100S/AT88SA102S, or 0's

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

Mode Encoding

Bit 5 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 the AT88SA10HS and should be 0.

Table 11. 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 0's in the above message generation step as a security measure.

4.3. 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 & HOST2 commands must be re-executed – HOST2 cannot be repeatedly executed.

Table 12. Input Parameters

	Name	Size	Notes
Opcode	HOST2	1	0x80
Param1	Zero1	1	Must be 0x00
Param2	Zero2	2	Must be 0x00 00
Data	ClientResponse	32	Response from the client.

Table 13. Output Parameters

Name	Size	Notes
Success	1	If the input ClientResponse matches the internally generated response, a value of 0 will be returned by the 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.

4.4. 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 14. 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 0 & 1 are used, all others must be 0's
Data	Ignored	0	

Table 15. Output Parameters

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

Table 16. Mode Encoding

Name	Value	Notes
ROM	0x00	Reads four bytes from the ROM. Bit 1 of the address parameter must be 0.
Fuse	0x01	Reads the value of 32 fuses. Bit 1 of the address parameter must be 1.

4.5. 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 17. 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 the AT88SA10HS.

Table 18. Output Parameters

Name	Size	Notes
Success	1	Upon successful execution, a value of 0 will be returned by the 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 1’s
127 bits	Seed from input stream
1 bits	‘1’ pad
64 bits	length of message in bits, fixed at 512

4.6. 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 0, then the corresponding fuse is left in its current state. The first bit sent to the AT88SA102S corresponds to Fuse[0] and so on up to Fuse[87]. Note that 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 AT88SA102S, this map may be encrypted before being sent to the chip. If this mode is desired, then the Decrypt parameter should be set to 1 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 Vcc is greater than 4.5V, then the BurnTime parameter should be set to 0x00 and the internal burn time will be 250µs. If Vcc is less than 4.5V but greater than V_{BURN} then the BurnTime parameter should be set to 0x8000 and the internal burn time will be 190ms 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 Vcc is less than 4.5V, 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, the AT88SA102S 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, optionally in combination with other fuses.

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

Table 19. 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 0x00 00 if Vcc > 4.5V, must be 0x80 00 otherwise.
Data	Map	11	Which fuses to burn, may be encrypted.

Table 20. Output Parameters

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

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

4.7. 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 21. Input Parameters

	Name	Size	Notes
Opcode	PAUSESHORT	1	0x00
Param1	Ignored	1	Must be 0x00
Param2	Ignored	2	Must be 0x00 00
Data	Ignored	0	

Table 22. Output Parameters

Name	Size	Notes
Success	1	After a delay of t_{PAUSE}, the AT88SA10HSS will return a value of 0 in response to a transmit flag.

5. Pinout

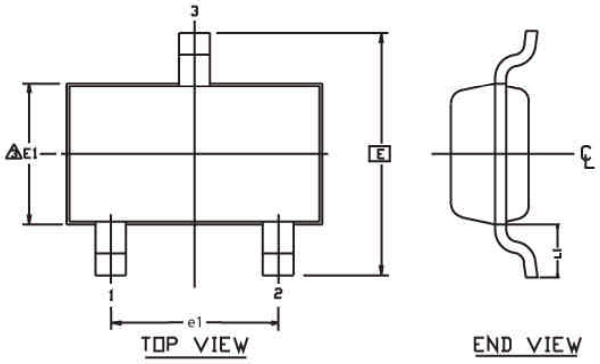
There are three pins on the chip.

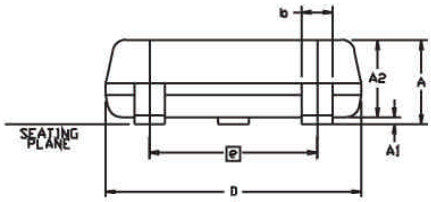
Table 23. Chip Pins

Pin #	Name	Description
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 V_{SS}.
2	V_{CC}	Power supply, 2.5 – 5.5V. This pin should be bypassed with a high quality 0.1µF capacitor close to this pin with a short trace to V_{SS}. Refer to Applications Notes on Atmel's website for more details.
3	V_{SS}	Connect to system ground.

6. Package Drawing

3TS1 - Shrink SOT





SIDE VIEW

1. Dimension D does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.25 mm per end. Dimension E1 does not include interlead flash or protrusion. Interlead flash or protrusion shall not exceed 0.25 mm 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 0.15 mm from the lead tip.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.89	-	1.12	
A_1	0.01	-	0.10	
A_2	0.88	-	1.02	
D	2.80	2.90	3.04	1,2
E	2.10	-	2.64	
E_1	1.20	1.30	1.40	1,2
L_1	0.54 REF			
e1	1.90 BSC			
b	0.30	-	0.50	3

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

11/5/08

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

7. Revision History

Table 24. Revision History

Doc. Rev.	Date	Comments
8595A	04/2009	Initial document release.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
securemem@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

8595A–SMEM–04/09