iPod Authentication Coprocessor 2.0C Specification

Release R1

2011-06-22

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

Figures and Tables

Chapter 7 CP Device Characteristics 33

Introduction

Note: This document uses the term "Apple device" to refer generically to iPods, iPhones, and iPads, all of which support the iPod Accessory Protocol (iAP) interface. Among these products, those that also run iOS (Apple's mobile operating system) are referred to as "iOS devices." Specifications in this document that are designated for iOS devices apply only to those products. Specifications designated for iPods apply only to Apple devices that are not iOS devices.

Overview

An Apple device verifies whether a third-party accessory attached to it is authorized for use with the Apple device by issuing an authentication challenge to the accessory. The accessory must respond to the Apple device's challenge, and it can do so only with the assistance of an **iPod Authentication Coprocessor** (CP) chip located in the accessory. Conversely, the accessory can use its CP chip to authenticate the iPod. Certain control and reporting functions of the Apple device are made available externally only after it has authenticated an attached accessory as being authorized.

Earlier versions of the iPod Authentication Coprocessor (1.0, 2.0A, and 2.0B) were implemented in QFN-40, QFN-20, and SOP-8 packages. The current version, 2.0C, is supplied in a smaller and more efficient PG-USON-8-1 package. This document describes the configuration, usage, and specifications of Apple's iPod Authentication Coprocessor 2.0C.

Authentication Protocol

The authentication protocol supported by the iPod Authentication Coprocessor 2.0C is based on standard X.509 version 3 certification. Each certificate is generated and signed by a recognized certificate authority and has a unique serial number. Information about the X.509 standard can be found at the IETF website http://tools.ietf.org/html/3280.

For information about the iAP General lingo commands required to perform authentication using the iPod Authentication Coprocessor 2.0C, see Apple's *MFi Accessory Firmware Specification*.

The iPod Authentication Coprocessor 2.0C supports iAP General lingo commands 0x14 through 0x1F, providing five authentication-related services:

For Apple device authentication of the accessory:

- **Certificate delivery:** To initiate authenticatication of the accessory that contains it, the CP supplies an X.509 digital certificate for public key verification by the attached Apple device.
- **Signature generation:** To complete authentication of the accessory that contains it, the CP generates a valid digital signature in response to a challenge from an attached Apple device. This signature authorizes the Apple device to respond to messages and commands from the accessory.

For accessory authentication of the Apple device:

- **Apple device certificate validation:** To initiate the authentication of an Apple device attached to an accessory, the CP verifies that the X.509 certificate supplied by Apple device has been signed by the proper certificate authority.
- **Challenge generation:** To continue the authentication of an Apple device attached to an accessory, the accessory's CP can generate a challenge to be sent to the Apple device.
- **Signature verification:** To complete the authentication of an Apple device attached to the accessory, the CP can verify the signature returned by the Apple device in response to the previous challenge.

Terminology Used in This Document

Certain technical terms specific to this document are defined in Table 1-1.

Table 1-1 Document-specific terminology

Term	Definition
Accessory controller	The microcontroller in an accessory responsible for implementing application-specific logic.
Authentication coprocessor	A device in an accessory controller that provides Apple device-related digital signature creation and verification services.
Challenge	A random number sent via iAP from an Apple device to an accessory controller, or vice versa. The device being challenged must perform a digital signature computation on the offered challenge and return the resulting digital signature to the challenging device for verification.
Digital signature	The result obtained by performing a digital signing process on an offered challenge.
iAP	iPod Accessory Protocol. See Apple's *MFi Accessory Firmware Specification*.
I^2C bus	A 2-wire serial bus designed by Philips to allow easy communication between components that reside on the same circuit board. The I^2C specification is located at http://www.semiconductors.philips.com/ acrobat_download/literature/9398/39340011.pdf.

Term	Definition
X.509 certification	A standard defined by the International Telecommunication Union (ITU) that governs the format of certificates used for authentication and sender identity verification in public-key cryptography. X.509 certificates contain the public keys used in the Apple device's accessory authentication process.

General Specification Terms

Parts of this document contain specification requirements that are incorporated by reference into legal agreements between Apple Inc. and its licensees. The use of the words "must," "should," and "may" in these specifications have the following meanings:

- "Must" means that the specification is an absolute requirement.
- "Must not" means that the specification is an absolute prohibition.
- "Should" means that there may be valid reasons in particular circumstances to ignore the specification, but their full implications must be understood and carefully weighed before choosing to do so.
- "Should not" means that there may be valid reasons in particular circumstances that make the specified action or feature acceptable, but their full implications must be understood and carefully weighed before choosing to include it.
- "May" means that the indicated action or feature does not contravene this specification.

Related Documents

For further information about authenticating Apple devices and their attached accessories, see Apple's *MFi Accessory Firmware Specification* and *MFi Accessory Hardware Specification*.

Signal Descriptions and Reference Circuit

This chapter defines the pinouts, signals, and reference circuitry of the iPod Authentication Coprocessor 2.0C.

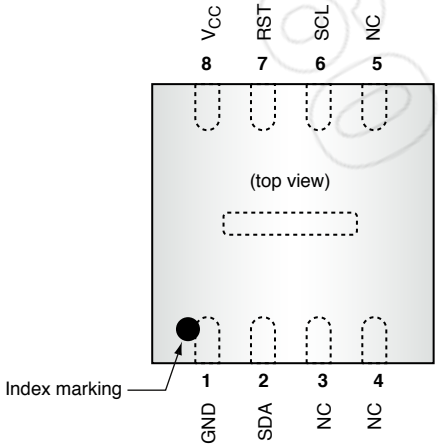
CP Signals and Pinouts

The 2.0C CP chip signal descriptions are given in Table 2-1 and its pinouts are shown in Figure 2-1.

Table 2-1 CP signals

Signal name	Pin	I/O	Description
GND	1		Supply voltage, negative terminal
SDA	2	I/O	I^2C data
NC	3-5		Must not be connected
SCL	6	I	I^2C clock
RST	7	I	At reset: selects I^2C slave address. During operation: CP warm reset.
V_{CC}	8		Supply voltage, positive terminal

Figure 2-1 CP chip pinouts, top view



PG-USON-8-1 package

The thermal pad on the bottom of the CP may be left unconnected or optionally connected to GND.

Address Selection

After power-up or in response to a warm reset, the state of RST is used to select the CP's I²C slave addresses, as shown in Table 2-2.

Table 2-2 Address selection signals

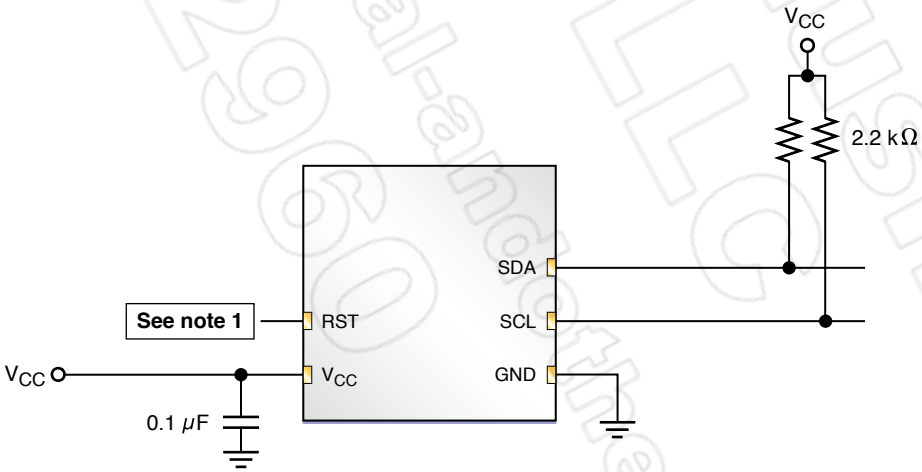
RST state	I²C addresses	
	Write	Read
0	0x20	0x21
1	0x22	0x23

See "Communication Process" (page 15) for the interface requirements of the CP's I²C slave communication transport.

Reference Circuit

The reference circuit for I²C operation of the CP is shown in Figure 2-2.

Figure 2-2 Reference circuit for CP



Note 1: If the CP's warm reset function is not needed, RST can be tied to either V_{CC} or GND, depending on which of the addressing modes shown in Table 2-2 is used. If the warm reset function is needed, RST should be connected to a general-purpose I/O line on the accessory's controller. For further details, see Starting Up the CP by Warm Reset (page 14).

Hardware Configuration and Interface

This chapter describes the operating modes of the iPod Authentication Coprocessor 2.0C and the ways that it interacts with other circuitry.

System Voltage

The 2.0C CP may be used either in an accessory powered by an attached Apple device or in an accessory that has its own power source.

Startup of the I²C Interface

The CP's I²C communication interface can be started up either by supplying power to the V_{CC} line or by performing a warm reset on the RST line. The accessory actions required for these two procedures are specified in the next sections.

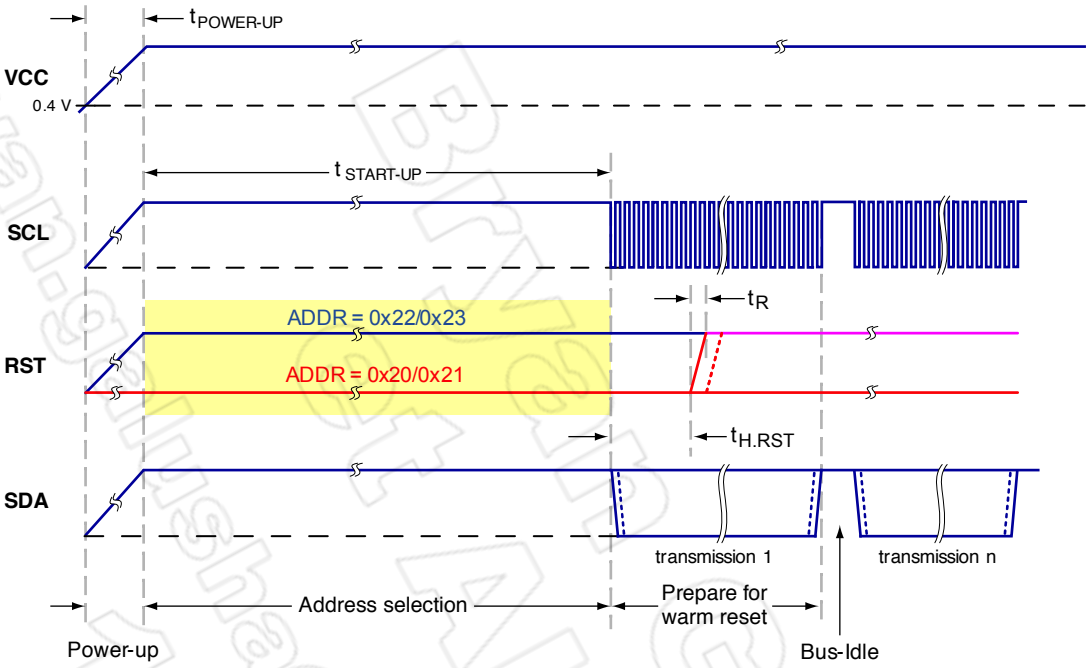
Starting Up the CP by Turning Power On

To activate the I²C interface by supplying power to the V_{CC} line, the accessory must perform the following startup procedure. This procedure is required both to support address selection by means of the RST line and to support the option of a warm reset later.

- The V_{CC} line must be supplied with power and the SDA and SCL lines must be set high during the entire procedure.
- The RST line, used to select addressing as described in Address Selection (page 12), must be kept either low or high during the entire procedure. If the RST line is kept low during the procedure, the accessory must set it high not earlier than 10 ms after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than 10 ms after the V_{CC} line goes high. If the RST line has been kept low and the accessory might need to perform a warm reset of the CP chip later, the accessory must set the RST line high not earlier that 1 ms after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-1 (page 14)). If the option of a warm reset later is not needed, the accessory may tie RST directly to either V_{CC} or GND.

Figure 3-1 (page 14) diagrams the timing of the I²C interface when it is started up by turning power on.

Figure 3-1 I²C interface startup timing



In this diagram, $t_{STARTUP} \geq 10$ ms and $t_{H.RST} \geq 1$ ms. The rise time of t_R and the fall time of t_F are both < 1 µs from 10% to 90% of the signal amplitude, and $t_{POWER-UP} < 200$ µs from VCC = 0.4 V until VCC = 90% of the target supply voltage.

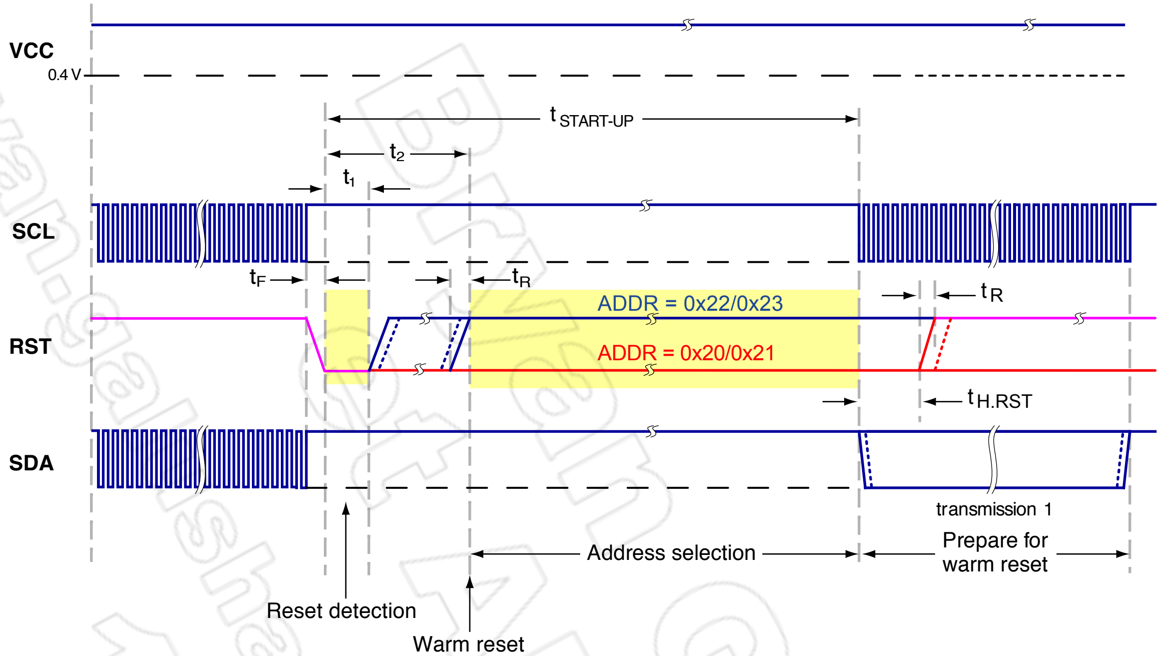
Starting Up the CP by Warm Reset

To reset the I²C interface through the RST line, after activating it through power-up as specified in Starting Up the CP by Turning Power On (page 13), the accessory must perform the following procedure:

- The V_{CC} line must be supplied with power during the entire procedure.
- The terminal must halt I²C communication, and the SDA and SCL lines must be set high before the RST line is set low.
- The RST line must be set low and kept low for at least 10 µs. Not later than 1 ms after the falling edge of the RST signal, the accessory must finish its I²C address selection by either keeping RST low or driving RST high. If the RST line is kept low during the warm reset, the accessory must set it high not earlier than 10 ms after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than 10 ms after the falling edge of the RST signal. If the RST line has been kept low during the warm reset, the accessory must set it high not earlier that 1 ms after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-2 (page 15)).

Figure 3-2 (page 15) diagrams the timing of the I²C interface when it is reset by software toggling the RST line without a power off/on cycle.

Figure 3-2 I²C interface warm reset timing



In this diagram, $t_{STARTUP} \geq 10$ ms, $t_{H.RST} \geq 1$ ms, $t_1 > 10$ µs, and $t_2 = 3$ ms. The rise time of t_R and the fall time of t_F are both < 1 µs from 10% to 90% of the signal amplitude.

Communication Process

When the CP is addressed using I²C, it acts as a standard 7-bit I²C slave. The I²C slave address is configured upon reset and is based on the RST input. The I²C effective slave address for writing is shown in Figure 3-3 and the corresponding read address in Figure 3-4.

Figure 3-3 I²C slave write address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	0

Figure 3-4 I²C slave read address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	1

In I²C mode, the CP has both a write address and a read address, as is typical for an I²C device. The I²C write address of the CP consists of the seven bits [A6:A0] followed by 0 for the R/nW bit. The I²C read address of the CP consists of the seven bits [A6:A0] followed by 1 for the R/nW bit. If the RST input is connected to ground, the write and read addresses of the CP are 0x20 and 0x21 respectively; if it is pulled high, the write and read addresses of the CP are 0x22 and 0x23.

Low-Power Sleep Mode

The 2.0C CP conserves power by entering a Sleep mode. It enters this mode automatically and cannot be forced to it externally. When in Sleep mode, the CP automatically wakes in response to any I²C communications sent to its address.

Coprocessor Registers

Registers within the iPod Authentication Coprocessor 2.0C (CP) are accessed via I²C transport, as described in Communication Process (page 15). This chapter specifies the CP's register addressing details and telegram formats.

Register Addresses

Registers and their addresses in the CP are listed in Table 4-1. Each register is discussed in the sections that follow.

Note: Registers in the same block with consecutive addresses may be read from sequentially in increasing numerical order, except as noted in the last column of Table 4-1. Registers must not be written to sequentially except as noted. Multibyte numeric values are stored in big-endian order; for example, the first byte in a two-byte register is the MSB of the stored value and the second byte is its LSB.

Table 4-1 iPod Authentication Coprocessor 2.0C register map

Register address	Block	Register name	Length in bytes	Contents after reset	Access	Notes
0x00	0	Device Version	1	0x05	Read-only	
0x01	0	Firmware Version	1	0x01	Read-only	
0x02	0	Authentication Protocol Major Version	1	0x02	Read-only	
0x03	0	Authentication Protocol Minor Version	1	0x00	Read-only	
0x04	0	Device ID	4	0x00000200	Read-only	
0x05	0	Error Code	1	0x00	Read-only (cleared on read)	1
0x10	1	Authentication Control and Status	1	0x00	Read/write	
0x11	1	Signature Data Length	2	128	Read/write	2
0x12	1	Signature Data	128	Undefined	Read/write	

Register address	Block	Register name	Length in bytes	Contents after reset	Access	Notes
0x20	2	Challenge Data Length	2	20	Read/write	2
0x21	2	Challenge Data	128	Undefined	Read/write	
0x30	3	Accessory Certificate Data Length	2	≤1280	Read-only	
0x31	3	Accessory Certificate Data (Page 1)	128	Certificate	Read-only	
0x32	3	Accessory Certificate Data (Page 2)	128	Certificate	Read-only	
0x33	3	Accessory Certificate Data (Page 3)	128	Certificate	Read-only	
0x34	3	Accessory Certificate Data (Page 4)	128	Certificate	Read-only	
0x35	3	Accessory Certificate Data (Page 5)	128	Certificate	Read-only	
0x36	3	Accessory Certificate Data (Page 6)	128	Certificate	Read-only	
0x37	3	Accessory Certificate Data (Page 7)	128	Certificate	Read-only	
0x38	3	Accessory Certificate Data (Page 8)	128	Certificate	Read-only	
0x39	3	Accessory Certificate Data (Page 9)	128	Certificate	Read-only	
0x3A	3	Accessory Certificate Data (Page 10)	128	Certificate	Read-only	
0x40	4	Self-test Control and Status	1	0x00	Read/write	
0x41–0x4C	4	Reserved				
0x4D	1	System Event Counter (SEC)	1	Undefined	Read-only	

Register address	Block	Register name	Length in bytes	Contents after reset	Access	Notes
0x50	5	Apple Device Certificate Data Length	2	0x0000	Read/write	
0x51	5	Apple Device Certificate Data (Page 1)	128	Undefined	Read/write	
0x52	5	Apple Device Certificate Data (Page 2)	128	Undefined	Read/write	
0x53	5	Apple Device Certificate Data (Page 3)	128	Undefined	Read/write	
0x54	5	Apple Device Certificate Data (Page 4)	128	Undefined	Read/write	
0x55	5	Apple Device Certificate Data (Page 5)	128	Undefined	Read/write	
0x56	5	Apple Device Certificate Data (Page 6)	128	Undefined	Read/write	
0x57	5	Apple Device Certificate Data (Page 7)	128	Undefined	Read/write	
0x58	5	Apple Device Certificate Data (Page 8)	128	Undefined	Read/write	

Note 1: Register 0x05 can be read sequentially only as part of a sequence that begins with a register in the range 0x00–0x04, in which case the read operation does not clear it.

Note 2: Registers 0x11 and 0x20 may each be written sequentially with registers 0x12 and 0x21, respectively.

Register Descriptions

This section describes the ways that the CP registers listed in Table 4-1 are used.

Device Version

The Device Version read-only register contains the version number of the coprocessor device. The current Authentication 2.0C coprocessor is designated as device version 0x05.

Firmware Version

The Firmware Version read-only register contains the version number of the coprocessor firmware. Firmware version numbers advance by whole integers.

Authentication Protocol Major and Minor Versions

The Authentication Protocol Major Version and Authentication Protocol Minor Version read-only registers provide the version number of the authentication protocol that the CP supports. This information is accessed by the iAP command `RetDevAuthenticationInfo` during accessory authentication.

Device ID

The Device ID read-only register identifies the accessory and is accessed by the iAP command `SetFIDTokenValues` during accessory identification.

Error Code

The Error Code read-only register stores the most recent communication or authentication process error code generated since the register was last cleared. The error code register is cleared after it is read. The possible error codes are listed in Table 4-2.

If a single communication operation happens to produce multiple errors (for example, by writing an invalid signature length during a multiregister write that also attempts to continue past the end of the corresponding block) then only the highest-numbered error code is stored.

Table 4-2 Error codes

Code	Description
0x00	No error
0x01	Invalid register for read
0x02	Invalid register for write
0x03	Invalid signature length
0x04	Invalid challenge length
0x05	Invalid certificate length
0x06	Internal process error during signature generation
0x07	Internal process error during challenge generation
0x08	Internal process error during signature verification
0x09	Internal process error during certificate validation
0x0A	Invalid process control
0x0B	Process control out of sequence
0x0C–0xFF	Reserved

Authentication Control and Status

The Authentication Control and Status read/write register provides control and status information for the CP's authentication processes.

When read from, the Authentication Control and Status register provides the status of the most recently requested CP process, as shown in Figure 4-1 and Tables 4-3 and 4-4.

Figure 4-1 Authentication Control and Status register, read-only bits

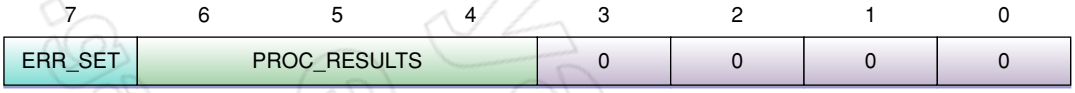


Table 4-3 Authentication ERR_SET values

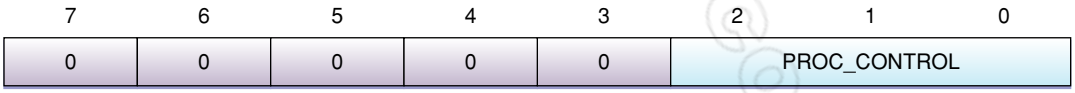
Value	Description
0	The Error Code register does not contain a code generated by the most recent command execution. However, it may still contain the most recent error code (greater than 0x00) generated by an earlier command.
1	The Error Code register contains the most recent process or communication error. Both this bit and the Error Code register contents are cleared after the Error Code register is next read. This bit is also cleared after every successful command execution.

Table 4-4 Authentication PROC_RESULTS values

Value	Description
0	Most recent process did not produce valid results.
1	Accessory signature successfully generated.
2	Challenge successfully generated.
3	Apple device signature successfully verified.
4	Apple device certificate successfully validated.
5-7	Reserved.

When written to, the Authentication Control and Status register controls the start of CP processes, as shown in Figure 4-2 and Table 4-5 (page 22).

Figure 4-2 Authentication Control and Status register, write-only bits



Note: Attempts to write to other bits are ignored.

Table 4-5 Authentication PROC_CONTROL values

Value	Description	Notes
0	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
1	Start new signature-generation process	
2	Start new challenge-generation process	The length of the challenge to be generated is defined by the Challenge Data Length register and ranges from 1 to 128 bytes.
3	Start new signature-verification process	
4	Start new certificate-validation process	Do not attempt to read the accessory certificate after writing the Apple device certificate but before validating it by this control.
5	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
6-7	Reserved.	

Signature Data Length

The Signature Data Length read/write register holds the length in bytes of the results of the most recent signature-generation process (if the Apple device is authenticating an accessory) or signature-verification process (if the accessory is authenticating the Apple device).

Before a signature-generation process begins, this register should contain 0x80, the maximum allowable signature length. After completion of the signature-generation process, the CP updates this register to contain the actual length of the generated signature. This updated value should be read in order to determine how much of the Signature Data register contains valid signature bytes.

Before a signature-verification process begins, this register should hold the actual length of the signature being verified.

Signature Data

In the case of a signature generation process, the Signature Data register holds the newly-generated data. In the case of a signature verification process, it holds the signature to be verified.

Challenge Data Length

The Challenge Data Length read/write register holds the length, in bytes, of the current challenge. This challenge may either be written into the CP, during Apple device authentication of an accessory, or generated by the CP during accessory authentication of an Apple device.

Before starting a signature-generation process on the current challenge during Apple device authentication of an accessory, this register must contain the length of the challenge.

Before starting a new challenge-generation process during accessory authentication of an Apple device, this register should contain the requested challenge length. The length must be in the range is 1 to 128 bytes; writing any other value will cause an error.

The required length of an authentication challenge is 20 bytes.

Challenge Data

The Challenge Data register holds the current challenge data. This data is either written into the CP or generated by the CP depending on the specific operation. The number of bytes used or generated is determined by the value of the Challenge Length Data register.

Accessory Certificate Data Length

The Accessory Certificate Data Length read-only register holds the length of the X.509 certificate that the Apple device uses to authenticate an accessory. The length of a certificate varies but is always less than or equal to 1280 bytes. This length limit may not hold for future versions of the authentication protocol.

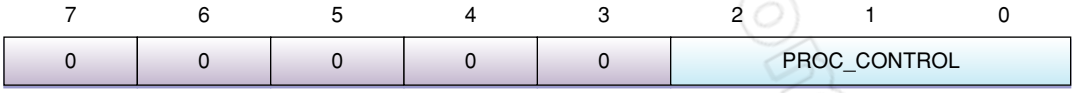
Accessory Certificate Data

The Accessory Certificate Data read-only register holds the PKCS#7-wrapped X.509 certificate that the Apple device uses to authenticate an accessory. The Accessory Certificate may be read from the coprocessor in 128-byte pages starting at any Accessory Certificate Data Page address, or it may be read in a continuous stream starting at Page 1. Since the length of the Accessory Certificate varies, fewer than all of the pages may be used. The Accessory Certificate Data Length value can be read to determine which Accessory Certificate Data Pages contain the certificate data.

Self-Test Control and Status

The Self-test Control and Status read/write register provides access to the built-in self-test functions of the coprocessor. When it is set to a value of 1, the Self-test Control and Status register initiates a self-test process, as shown in Figure 4-3 and Table 4-6 (page 24).

Figure 4-3 Self-test Control and Status register, write-only bits



Note: Attempts to write to other bits are ignored.

Table 4-6 Self-test PROC_CONTROL values

Value	Test process to be run
0	None
1	Run X.509 certificate and private key tests
2-7	Reserved

When read from, bits 7–4 of the Self-test Control and Status register report the results of the X.509 certificate and private key tests, as shown in Figure 4-4 and Table 4-7 (page 24). The CP detects a read cycle and resets the Control and Status register to 0x00 after it; hence bits 7–4 must all be retrieved in one operation.

Figure 4-4 Self-test Control and Status register, read-only bits

Table 4-7 Self-test result bits

Bit	Test	Bit value	
		0	1
7	X.509 certificate	Certificate not found	Certificate found in memory (see note below)
6	Private key	Private key not found	Private key found in memory (see note below)
5-4	Reserved		

Note: The X.509 and private key tests only verify that these elements are present in Flash memory; no authentication is performed.

System Event Counter

The System Event Counter (SEC) is a non-volatile register that holds the current value of the CP's event counter. The event counter automatically decrements one count per second while the CP is powered, stopping at 0. If the accessory controls power to the CP, it must wait until the SEC has decremented to 0 before removing power.

Apple Device Certificate Data Length

The Apple Device Certificate Data Length register holds the length of the X.509 certificate supplied by the attached Apple device. An accessory uses this certificate to authenticate an Apple device in both the certificate validation and signature verification processes. The length of an Apple device certificate varies but is always less than or equal to 1024 bytes. This length limit may not hold for future versions of the authentication protocol.

Writing a value in a range greater than 0 and less than or equal to 1024 will cause the CP to validate the data contained in the iPod Certificate Data registers. If the CP invalidates the iPod Certificate Data, it sets this register to 0.

Apple Device Certificate Data

The Apple Device Certificate Data register holds the X.509 Certificate that an accessory uses to authenticate an Apple device in both the certificate validation and signature verification processes. The Apple Device Certificate may be written to the coprocessor in 128-byte pages starting at any Apple Device Certificate Data Page address, but it may not be written in a multipage stream. Since the length of the Apple Device Certificate varies, not all of the pages need to be used. The Apple Device Certificate Data Length value determines which Apple Device Certificate Data Pages contain valid certificate data.

Authentication Data Flows

Authentication involves communication between the accessory controller (AC), the Authentication Coprocessor (CP) in the accessory, and the Apple device attached to the accessory.

Communication between the accessory controller and the CP takes place via the I^2C transport link described in Communication Process (page 15). Communication between the accessory controller and the Apple device takes place via the iPod Accessory Protocol. See Apple's *MFi Accessory Firmware Specification* for full details.

This chapter summarizes the kinds of information that pass between the AC, the CP, and the attached Apple device (Dev).

Apple Device Authentication of Accessory

The sequence of interactions by which an Apple device authenticates an accessory is shown in Table 5-1 (page 27). At the beginning of this process the accessory controller is granted access by the Apple device to the iAP lingo or lingoes it requests; however, if the process does not finish successfully that access is terminated.

Table 5-1 Sequence of interactions by which an Apple device authenticates an accessory

Command or action	Direction	Comments
Read Authentication Protocol Version and Device ID	CP –> AC	Accessory controller reads authentication protocol version and device ID from CP
`StartIDPS` (iAP)	AC –> Dev	The accessory controller initiates and completes the Identify Device Preferences and Settings (IDPS) process. It sends the Apple device a set of ID tokens, one of which includes the device ID returned by the CP. See *MFi Accessory Firmware Specification*, Chapter 2 and Appendix A.
`SetFIDTokenValues` (iAP)	AC –> Dev	
`EndIDPS` (iAP)	AC –> Dev	
`GetDevAuthenticationInfo` (iAP)	Dev –> AC	The Apple device requests device authentication info
Read Accessory Certificate Length and Data	CP –> AC	Accessory controller reads Accessory certificate from CP
`RetDevAuthenticationInfo` (iAP)	AC –> Dev	Accessory controller returns information needed for authentication process, using the authentication protocol version number and X.509 certificate returned by the CP
`AckDevAuthenticationInfo` (iAP)	Dev –> AC	The status of the authentication version comparison is returned to the accessory controller. The returned status includes information about the validity of the X.509 certificate.

Command or action	Direction	Comments
`GetDevAuthenticationSignature` (iAP)	Dev –> AC	The Apple device sends accessory controller a challenge and requests that it provide corresponding digital signature
Write Challenge Length and Challenge Data	AC –> CP	Accessory controller writes challenge into CP
Write Authentication Control: PROC_CONTROL = 1	AC –> CP	Accessory controller starts signature-generation process in CP
Wait for process completion	CP –> AC	Accessory controller waits for CP to finish processing
Read Authentication Status	CP –> AC	Accessory controller reads Authentication Status and checks PROC_RESULTS field
Read Signature Data Length and Signature Data	CP –> AC	Accessory controller reads signature from CP
`RetDevAuthenticationSignature` (iAP)	AC –> Dev	Accessory controller returns signature to the Apple device
`AckDevAuthenticationStatus` (iAP)	Dev –> AC	Resulting success or failure of signature verification sent to accessory controller by the Apple device

Accessory Authentication of the Apple Device

The sequence of interactions by which an accessory authenticates an Apple device is shown in Table 5-2.

Table 5-2 Sequence of interactions by which an accessory authenticates an Apple device

Command or action	Direction	Comments
Read Authentication Protocol Version and Device ID	CP –> AC	Accessory controller reads authentication protocol version and device ID from CP
The accessory controller performs the accessory identification and authentication processes listed in Table 5-1 (page 27). These processes, by which the Apple device authenticates the accessory, must finish successfully before the sequence by which the accessory authenticates the Apple device can continue.		
`GetiPodAuthenticationInfo` (iAP)	AC –> Dev	Accessory controller requests Apple device authentication information
`RetiPodAuthenticationInfo` (iAP)	Dev –> AC	The Apple device returns its authentication version and certificate
Write Apple device Certificate length and data	AC –> CP	Accessory controller writes Apple device Certificate into CP

Command or action	Direction	Comments
Write Authentication Control: PROC_CONTROL = 4	AC –> CP	Accessory controller starts certificate validation process in CP
Wait for process completion	CP –> AC	Accessory controller waits for CP to finish processing
Read Authentication Status	CP –> AC	Accessory controller reads Authentication Status and checks PROC_RESULTS field
`AckiPodAuthenticationInfo` (iAP)	AC –> Dev	Results of the authentication information comparison are returned to the Apple device
Write Authentication Control: PROC_CONTROL = 2	AC –> CP	Accessory controller starts challenge-generation process in CP to calculate new challenge
Wait for process completion	CP –> AC	Accessory controller waits for CP to finish processing
Read Authentication Status	CP –> AC	Accessory controller reads the authentication status and checks PROC_RESULTS field
Read Challenge Length and Challenge Data	CP –> AC	Accessory controller reads challenge from CP
`GetiPodAuthentication-Signature` (iAP)	AC –> Dev	Accessory controller sends challenge to the Apple device and requests that it calculate digital signature
`RetiPodAuthentication-Signature` (iAP)	Dev –> AC	The Apple device returns a digital signature to the Accessory controller
Write Signature Data Length and Signature Data	AC –> CP	Accessory controller writes digital signature into CP
Write Challenge Length and Challenge	AC –> CP	Accessory controller writes challenge into CP (it needs to write this into the CP only if the challenge has been changed in the meantime)
Write Authentication Control: PROC_CONTROL = 3	AC –> CP	Accessory controller starts signature-verification process in CP
Wait for process completion	CP –> AC	Accessory controller waits for CP to finish processing
Read Authentication Status	CP –> AC	Accessory controller reads authentication status and checks PROC_RESULTS field
`AckiPodAuthenticationStatus` (iAP)	AC –> Dev	Signature verification status is returned to the Apple device.

I2C Communication Protocol

The iPod Authentication Coprocessor (CP) supports the I²C communication protocol, acting as an I²C slave. Its SCL signal is the I²C clock line and is driven by the accessory controller. Its SDA signal is the I²C data line and is driven by whichever device is currently sending data.

Unlike the iPod Authentication Coprocessor version 2.0B, CP 2.0C does not perform clock synchronization by stretching SCL. It may, however, not-acknowledge (NACK) a requested register operation if busy, so the I²C master should expect retry operations as a normal part of CP 2.0C use.

The maximum supported I²C clock rate is 400 kHz.

Slave Selection and Reset

During reset, the RST signal must specify the CP's I²C slave address and must be held stable for at least 10 ms after power-up or reset, as described in "Communication Process" (page 15). As an I²C slave, the CP is then selected in-band via its I²C address. The least significant bit of the I²C slave address controls whether a write or a read operation is to be performed, as described in "Address Selection" (page 12).

Coprocessor Busy

When the CP is busy processing it is unable to handle incoming communication attempts. If the coprocessor does not ACK its slave address during an attempted I²C communication, then the coprocessor is busy. The accessory controller must repeatedly attempt communication until the coprocessor sends an ACK after receiving its slave address.

Writing to the Coprocessor

To write data to the coprocessor, follow these steps:

1. Send the I²C start sequence.
2. Send the I²C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 1.
4. Send the register address at which to begin writing.
5. Send the data bytes.

6. Send the I^2C stop sequence.

Reading from the Coprocessor

To read data from the coprocessor, follow these steps:

1. Send the I^2C start sequence.
2. Send the I^2C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 1.
4. Send the register address at which to begin reading.
5. Optional: send the I^2C stop sequence.
6. Send the I^2C start sequence.
7. Send the I^2C read address of the CP.
8. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 6.
9. Read the data bytes.
10. Send the I^2C stop sequence.

Any additional reads after an I^2C read stop sequence continue with the byte following the previous byte read until an invalid register address or an end of block is reached, at which point the slave returns 0xFF in response to all further reads.

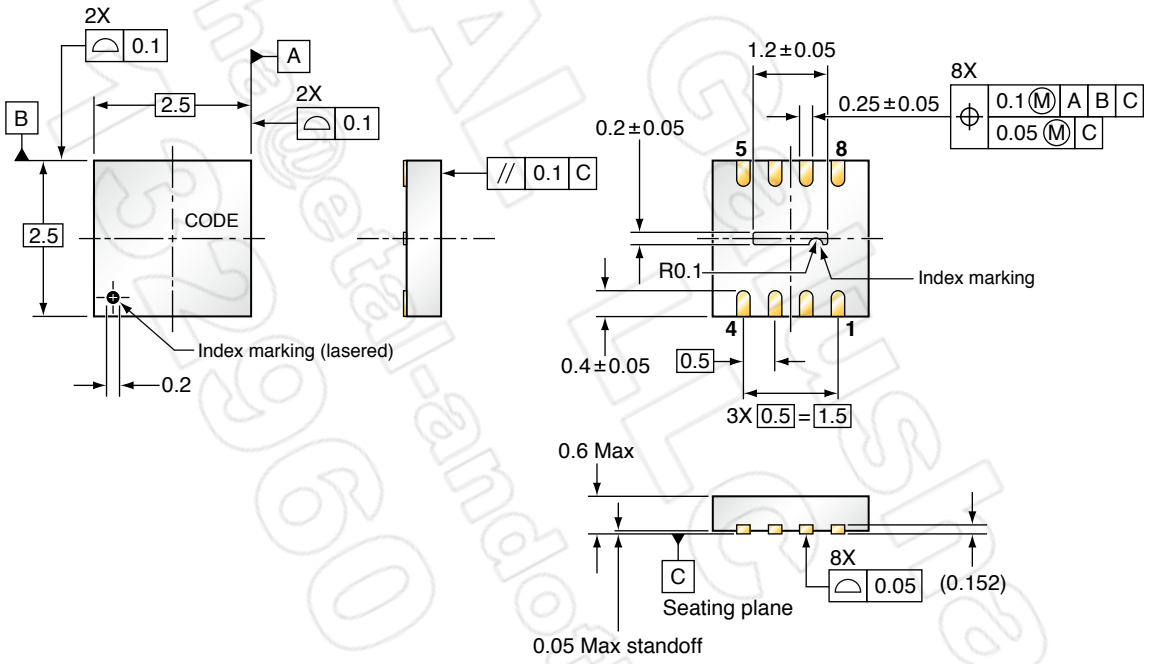
CP Device Characteristics

This chapter provides technical details and tolerances for the Apple iPod Authentication Coprocessor 2.0C (CP) chip.

Physical Configuration

Figure 7-1 shows the CP package's layout, pin locations, and dimensional tolerances.

Figure 7-1 Authentication coprocessor 2.0C package



Maximum Environmental Conditions

Table 7-1 (page 34) lists the CP's absolute maximum electrical and free-air temperature ranges. Stresses to the CP chip beyond the ranges listed in Table 7-1 may cause permanent damage. Exposure to either end of any range for extended periods may affect device reliability.

Table 7-1 Maximum electrical and temperature ranges

Condition	Maximum range
Voltage applied at V_{CC} relative to V_{SS}	–0.3 V to +7.0 V
Voltage applied to any pin	–0.3 V to V_{CC} + 0.3 V
Storage temperature	–40 °C to +125 °C

Recommended Operating Conditions

The CP is available only in a standard temperature range configuration. Internal sensors may force it to its reset state if any of the conditions listed in Table 7-2 are exceeded. Attempting to operate the CP in this state is not recommended and may lead to device failure or unreliability.

Table 7-2 Recommended operating conditions

Condition	Minimum	Maximum	Unit
Operating free-air temperature	–25	+85	°C
Supply voltage during program execution	1.62	5.5	V

I2C Interface Characteristics

Table 7-3 specifies the limits of the I²C interface between the CP and other components.

Table 7-3 I²C interface ranges

Parameter	Minimum	Maximum	Unit
SCL clock frequency (f_{SCL})	10	400	kHz
External bus capacitance to ground (C_b)		100	pF

DC Electrical Characteristics

Tables 7-4 through 7-6 (page 35) show the DC electrical characteristics of the CP chip over its recommended voltage and temperature ranges. Unless otherwise specified in these tables, V_{CC} = 1.62 to 5.5 V and T_A = –25 °C to +85 °C.

Table 7-4 Supply current into V_{CC}, excluding external current

Parameter	Test conditions	Minimum	Typical	Maximum	Unit
$I_{(AM)}$ Active mode (authentication process running)			6	7.5	mA
$I_{(sleep)}$ Sleep mode	T_A = 25 °C		35	80	µA

Table 7-5 Inputs

Symbol	Parameter	Test conditions	Minimum	Typical	Maximum	Unit
V_{IH}	High-level input voltage		$V_{CC} \times 0.7$		$V_{CC} + 0.3$	V
V_{IL}	Low-level input voltage		–0.3		$V_{CC} \times 0.2$	
I_i	Input current		–10	1	10	µA

Table 7-6 Outputs

Symbol	Parameter	Test conditions	Minimum	Typical	Maximum	Unit
V_{OL}	Low-level output voltage	$I_{OL(max)}$ = –1 mA;	V_{SS}		$V_{SS} + 0.4$	V

Timing Characteristics

When power is turned on to the CP, V_{CC} must reach 90% of the CP's target supply voltage within 200 µs after it exceeds 400 mV.

Figure 7-2 illustrates the CP's typical I/O port input signal timing and voltage limits. Table 7-7 (page 36) lists the parameter values in Figure 7-2.

Figure 7-2 Typical I/O port input waveform

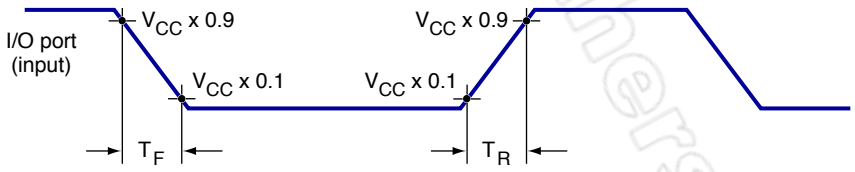


Table 7-7 Values for Figure 7-2

Symbol	Description	Maximum value
T_F	Fall time	1.0 µs
T_R	Rise time	1.0 µs

Coprocessor 2.0B to 2.0C Migration Guide

This appendix summarizes the ways that the 2.0C version of the iPod Authentication Coprocessor differs from the older 2.0B version, as a guide to migrating accessory compatibility from one to the other.

Only I2C Communication Protocol

The 2.0C CP supports only the I²C communication protocol. Accessory designs that use the SPI communication protocol must switch to I²C.

Increased SCL Speed

When running in the I²C communication protocol, the 2.0B CP supported a maximum SCL speed of only 50 kHz. The 2.0C CP supports a maximum SCL speed of 400 kHz (see I2C Interface Characteristics (page 34)).

Compatibility with both CPs: CP register 0x00 (Device Version) contains a value of 0x03 in the 2.0B CP and 0x05 in the 2.0C. If an accessory needs to be compatible with both CPs, its microcontroller should read the contents of this register to determine which version of the CP is present. It should do this at an SCL speed no greater than 50 kHz and a reset cycle delay of 30 ms (see "Shorter Reset Cycle," below), to assure its compatibility with either version. If the 2.0C CP is present, the accessory may then increase its SCL speed and decrease its reset delay.

Shorter Reset Cycle

The reset cycle in the 2.0C CP is shorter than it was in the 2.0B CP—10 ms instead of 30 ms. An accessory designed for the iAC 2.0B reset timing will still work with the 2.0C CP, but it may be advantageous to redesign it for a shorter delay.

Warm Reset Supported

The 2.0C CP supports optional warm reset, using the RST pin. This pin is also used for I²C slave address selection.

Note: The 2.0B CP had a dedicated reset pin, but the 2.0C CP does not.

Automatic Sleep State Entry and Exit

The 2.0C CP automatically enters and exits deep sleep without explicit program control. The 2.0B CP required an explicit sleep-entry command to drive it to its deepest sleep state, and it required either V_{cc} power cycling or a reset pin signal to exit this state.

Note: An accessory may send explicit sleep entry commands to the 2.0C CP, but doing so will have no effect.

NACK Responses Replace Clock Stretching

The 2.0C CP does not use the I²C clock stretching mechanism to control communication timing. Instead, it will frequently send I²C NACK (not-acknowledge) responses if a requested register operation has not yet finished. The I²C master in the accessory's microcontroller must expect such NACK responses as part of its normal operation.

Shorter Accessory Certificate Data

The 2.0C CP contains fewer registers dedicated to holding Accessory Certificate Data and a shorter maximum value for its Accessory Certificate Data Length register. If the accessory microcontroller first reads the Accessory Certificate Data Length register and then reads only that much data from the Accessory Certificate Data registers, migrating from the 2.0B CP to the 2.0C CP will not require design changes.

System Event Counter Must Be Zero Before Power-Down

If the accessory removes power from the 2.0C CP, it must first verify that register 0x4D (the System Event Counter) is at zero (see System Event Counter (page 24)). The 2.0B CP had no equivalent requirement.

Document Revision History

This table describes the changes to *iPod Authentication Coprocessor 2.0C Specification*.

Date	Notes
2011-06-22	*Revision R1:* First release.

Document Revision History