MAXQ1065

Ultra Low-Power Cryptographic Controller with ChipDNA® for Embedded Devices

General Description

The MAXQ1065 is a security coprocessor that provides turnkey cryptographic functions for root-of-trust, mutual authentication, data confidentiality and integrity, secure boot, secure firmware update, and secure communications with generic key exchange and bulk encryption or complete TLS support. The device integrates 8KB of secure storage for user data, keys, certificates, and counters with user-defined access control and life cycle management. It also has a configurable output pin and a tamper input pin. Commands are accessible through a standard SPI or I^2C interface.

The MAXQ1065's low power consumption makes it suitable for battery-powered applications, and the extremely reduced footprint and pin count allow easy integration into medical and wearable devices. Its lifetime and operating range make it compatible with long-term deployments in harsh environments. The MAXQ1065 life cycle management allows flexible access control rules during the major life cycle stages of the device. Secure key loading protocol and secure factory preprogramming are available.

DeepCover® embedded security solutions cloak sensitive data under multiple layers of advanced security to provide the most secure key storage possible. To protect against device-level security attacks, invasive and noninvasive countermeasures are implemented including active die shield, encrypted storage of keys using the ChipDNA PUF technology, and externally callable algorithmic subroutines.

Applications

The MAXQ1065 enhances the security of connected embedded systems in applications such as industrial IoT, SCADA, medical equipment, building and home automation, smart city, smart metering. It is a key element for the cybersecurity of infrastructures and connected device network nodes, routers, and gateways. It is ideally suited for:

- Secure Communication: Key Exchange, TLS
- Secure Data Storage
- Mutual Authentication and Certificate Management
- Anti-cloning, Anti-counterfeiting, Feature and Usage Control
- System-Level Tamper Protection and Integrity
- Secure Boot, Secure Firmware Update

Benefits and Features

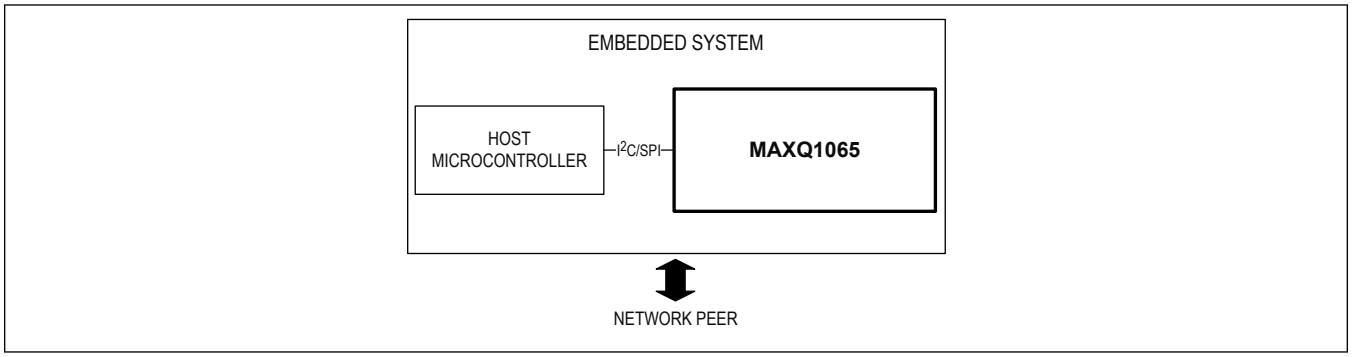
- ECC Compute Engine Using Curve NIST P-256
  - FIPS-186 ECDSA
  - NIST SP800-56Ar3 Key Exchange with Static Unified Model, C(0e, 2s, ECC CDH) with One-Step Key Derivation Using SHA-256
  - On-Board EC Key Generation with SP800-90B/A
- SHA-2 Compute Engine
  - NIST FIPS-180-4 SHA2-256, HMAC-SHA-256
- AES Compute Engine with 128/192/256 Key Sizes
  - ECB, CBC, CCM, GCM Cipher Modes
  - CBC-MAC, CMAC Message Authentication Codes
  - Onboard AES Key Generation with SP800-90A/B
- True Random Number Generator (TRNG)
  - NIST SP800-90A/C Compliant
  - NIST SP800-90B Entropy Source
- Secure Communication
  - TLS/DTLS 1.2 Handshake and Record Layer
    - ECDSA Authentication and ECDHE Key Exchange
    - AES-GCM or CCM Record Layer
  - SP800-56Ar3-Based Key Exchange
- X.509 v3 Certificate Support
  - Storage of Root and Device Certificates
  - Onboard Verification of Chains of Certificates
  - ECDSA Verification on Supported Curves
- High-Speed Interface for Host Microcontroller Communication
  - 10MHz SPI with Mode 0 or Mode 3 Operation
  - 100kbps and 1Mbps I^2C
- 8KB User Flash Array with ChipDNA PUF Encryption
- Unique, Unalterable Factory-Programmed ID Number
- Tamper Input Detects System-Level Intrusion
- Secure Factory Provisioning Service
- 12-Pin, 3mm x 3mm TDFN Package
- -40°C to +105°C, 1.62V to 3.63V
- Low-Power Operation: 100nA (typ) in Standby

ChipDNA is a trademark of Analog Devices, Inc.
DeepCover is a registered trademark of Analog Devices, Inc.

Ordering Information *appears at end of data sheet.*

19-101139; Rev 1; 4/25

Functional Diagrams



Absolute Maximum Ratings

(All voltages with respect to GND, unless otherwise noted.).....
V_{DD} to GND..... -0.3V to 3.63V
Any Pin to GND except V_{DD}.....-0.3V to (V_{DD} + 0.3)V
Operating Temperature Range .....-40°C to +105°C
Storage Temperature Range.....-40°C to +150°C
Junction Temperature ..... +150°C
Soldering Temperature (reflow)..... +260°C

Continuous Package Power Dissipation 12-Pin TDFN (Single-Layer Board) T_A = +70°C, (derate 15.90mW/°C above +70°C).....1269.8mW
Continuous Package Power Dissipation 12-Pin TDFN (Multilayer Board) T_A = +70°C (derate 24.40mW/°C above +70°C).....1951.2mW

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Package Information

12 TDFN

Package Code	TD1233+1C
Outline Number	21-0664
Land Pattern Number	90-0397
Thermal Resistance, Single-Layer Board:	
Junction to Ambient (θ_{JA})	63°C/W
Junction to Case (θ_{JC})	8.5°C/W
Thermal Resistance, Four-Layer Board:	
Junction to Ambient (θ_{JA})	41°C/W
Junction to Case (θ_{JC})	8.5°C/W

For the latest package outline information and land patterns (footprints), go to ***www.analog.com/en/resources/packaging-quality-symbols-footprints/package-index.html***. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to ***www.analog.com/en/resources/technical-articles/thermal-characterization-of-ic-packages.html***.

Electrical Characteristics

(All specifications and characteristics apply across the entire operating conditions range unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY						
Supply Voltage	V_{DD}	(*Note 1*)	1.62	3.3	3.63	V
Active Mode Current	I_A	T_A = +25°C, no cryptographic operation		1.7	3	mA
ECDSA Current	I_{ECDSA}	T_A = +105°C, performing signature operation			3	mA
SHA Current	I_{SHA}	T_A = +105°C, performing SHA-256 operation			3	mA
Programming Current	I_{PROG}	T_A = +105°C, memory programming current			3	mA
Idle Current	I_{IDLE}	T_A = +25°C		0.45		mA
V_{DD} Low-Power Mode Current	I_{PDWN}	T_A = +25°C, V_{PDWN} = 0V, V_{DD} = 1.8V (*Note 4*)		100		nA
Input Low Voltage for All Inputs	V_{IL_IO}				0.3 x V_{DD}	V
Input High Voltage for All Inputs Except Power	V_{IH_IO}		0.7 x V_{DD}			V

Electrical Characteristics (continued)

(All specifications and characteristics apply across the entire operating conditions range unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Low Voltage for All Outputs	V_{OL_IO}	I_{SINK} = 2mA		0.2	0.4	V
Output High Voltage for All Outputs	V_{OH_IO}	I_{SOURCE} = 2mA	V_{DD} - 0.4			V
Input Pull-up Resistor for All Inputs in Pull-up Mode	R_{PU}			20		kΩ
Input Pull-down Resistor for All Inputs in Pull-down Mode	R_{PD}			20		kΩ
NONVOLATILE MEMORY						
Flash Erase Time	t_{P_ERASE}	Page erase		20		ms
Flash Programming Time per Word	t_{PROG}			8		µs
Flash Endurance	N_{END}		10			kcycles
Data Retention	t_{RET}	T_A = +150°C	10			years
FUNCTIONAL TIMING						
Operation Time	t_{OP}				1	ms
Wake-Up Time	t_{WAKEUP}	(*Note 2*)		75		ms
DIGITAL I/O: GENERAL						
Output Voltage High (SPIS_MISO)	V_{OH}	I_{SOURCE} = 2mA	V_{DD} - 0.4			V
Output Voltage Low (SPIS_MISO)	V_{OL}	I_{SINK} = 2mA			0.4	V
Input Voltage High (SPIS_SCK, SPIS_SS, SPIS_MOSI)	V_{IH}		0.7 x V_{DD}			V
Input Voltage Low (SPIS_SCK, SPIS_SS, SPIS_MOSI)	V_{IL}				0.3 x V_{DD}	V
Input Leakage Current Low	I_{IL}	V_{DD} = 3.63V, V_{IN} = 0V	-500		+500	nA
Input Leakage Current High	I_{IH}	V_{DD} = 3.63V, V_{IN} = 3.63V	-500		+500	nA
SPI PERIPHERAL/SUBNODE						
Operating Frequency	f_{SCK}				10	MHz
Clock Period	t_{SCK}			$1/f_{SCK}$		µs
Clock Input High Time	t_{SCH}	(*Note 3*)		$t_{SCK}/2$		µs
Clock Input Low Time	t_{SCL}	(*Note 3*)		$t_{SCK}/2$		µs
SS Active Setup Time	t_{SSE}			10		ns
Data Input Setup Time	t_{SIS}			5		ns
Data Input Hold Time	t_{SIH}			1		ns
Clock Edge to Data Output Valid	t_{SOV}			5		ns

Electrical Characteristics (continued)

(All specifications and characteristics apply across the entire operating conditions range unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SS Inactive Setup Time	t_{SSD}			10		ns
SS Inactive Time	t_{SSH}			$1/f_{SCK}$		µs
Output Disable Time	t_{SLH}			10		ns
Clock Stable to SS Active	t_{SAD}			10		ns
Interface Setup Delay	t_{IF_SETUP}	(*Note 1*)	50			µs
Device Select to Clock Start	t_{SS_SCK}	(*Note 1*)	2			µs
End of Data to RDY Disable	t_{CMD_RDY}				50	µs
I²C PERIPHERAL						
Output Fall Time	t_{OF}	$V_{OH\_IO(MIN)}$ to $V_{OL\_IO(MAX)}$		80		ns
Pulse Width Suppressed by Input Filter	t_{SP}			75		ns
SCL Clock Frequency	f_{SCL}		0		1	MHz
Low Period SCL Clock	t_{LOW}		0.5			µs
High Period SCL Clock	t_{HIGH}		0.26			µs
Setup Time for a Repeated Start Condition	$t_{SU:STA}$		0.26			µs
Hold Time for Repeated Start Condition	$t_{HD:STA}$		0.26			µs
Data Setup Time	$t_{SU:DAT}$			50		ns
Data Hold Time	$t_{HD:DAT}$			10		ns
Rise Time for SDA and SCL	t_R			50		ns
Fall Time for SDA and SCL	t_F			30		ns
Setup Time for Stop Condition	$t_{SU:STO}$		0.26			µs
Bus Free Time Between a Stop and Start Condition	t_{BUF}		0.5			µs
Data Valid Time	$t_{VD:DAT}$		0.45			µs
Data Valid Acknowledge Time	$t_{VD:ACK}$		0.45			µs

Note 1: System requirement.

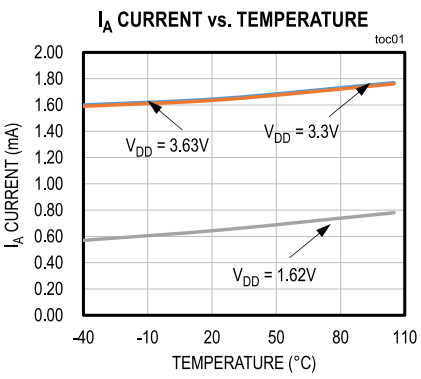
Note 2: The typical wake-up time is 75ms, but if the first command uses the NIST TRNG, then the wake-up time would be at least 160ms.

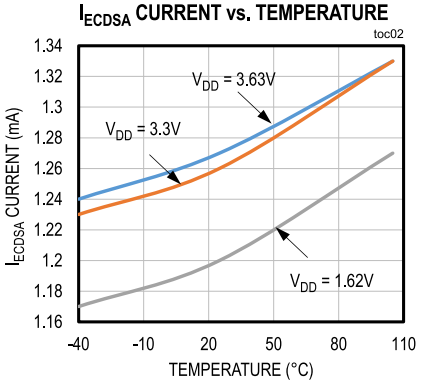
Note 3: $t_{SCH} + t_{SCL} \geq 1/f_{SCK}$ (max)

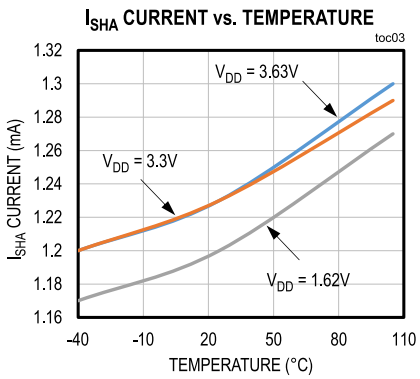
Note 4: For the I²C device, SDA/SCL and the two DNC pins need to be either floating or pulled low. For the SPI device, all four SPI pins need to be either floating or pulled low for the duration of the low-power mode.

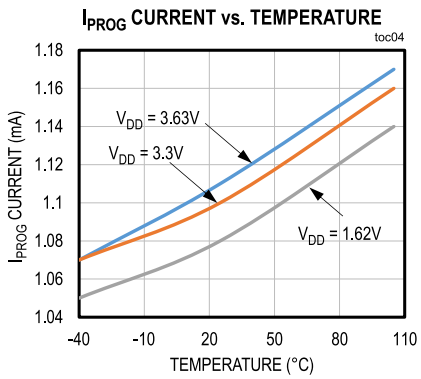
Typical Operating Characteristics

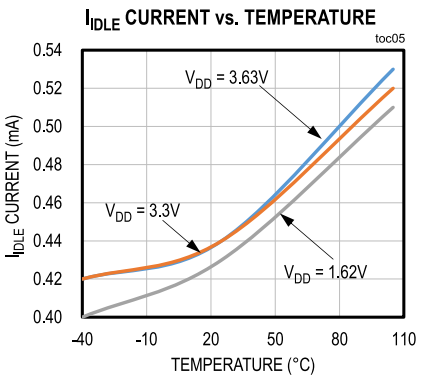
(T_A = T_{MIN} to T_{MAX} unless otherwise noted.)

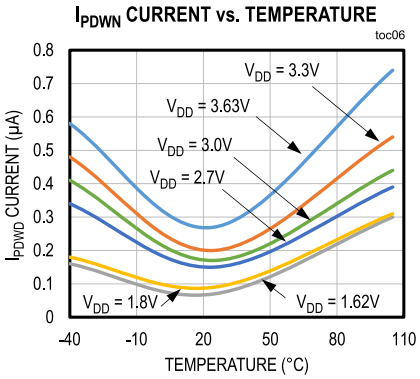






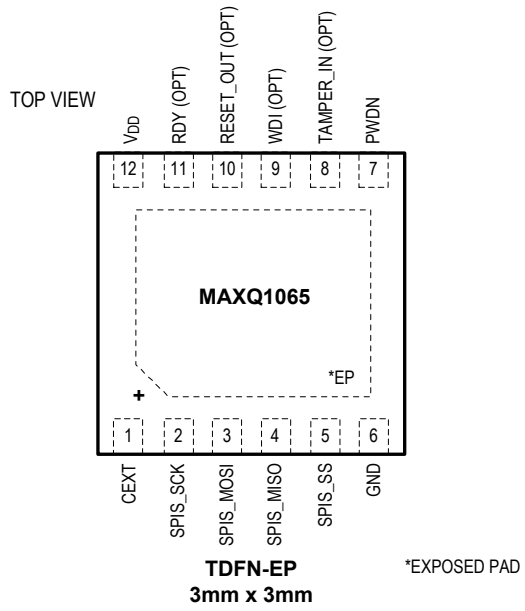






Pin Configurations

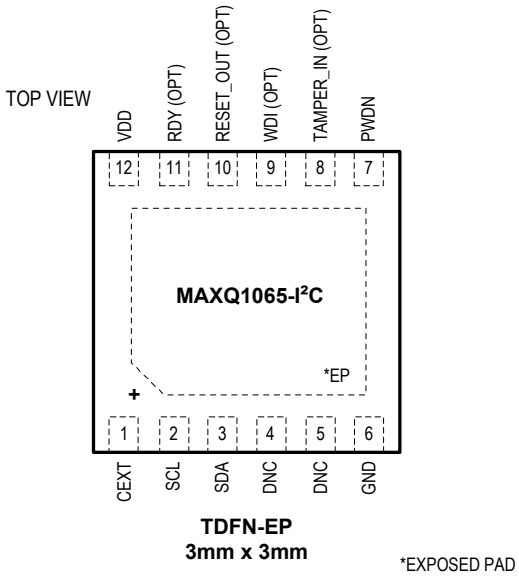
MAXQ1065-SPI



TDFN-EP
3mm x 3mm

*EXPOSED PAD

MAXQ1065-I²C



TDFN-EP
3mm x 3mm

*EXPOSED PAD

Pin Description

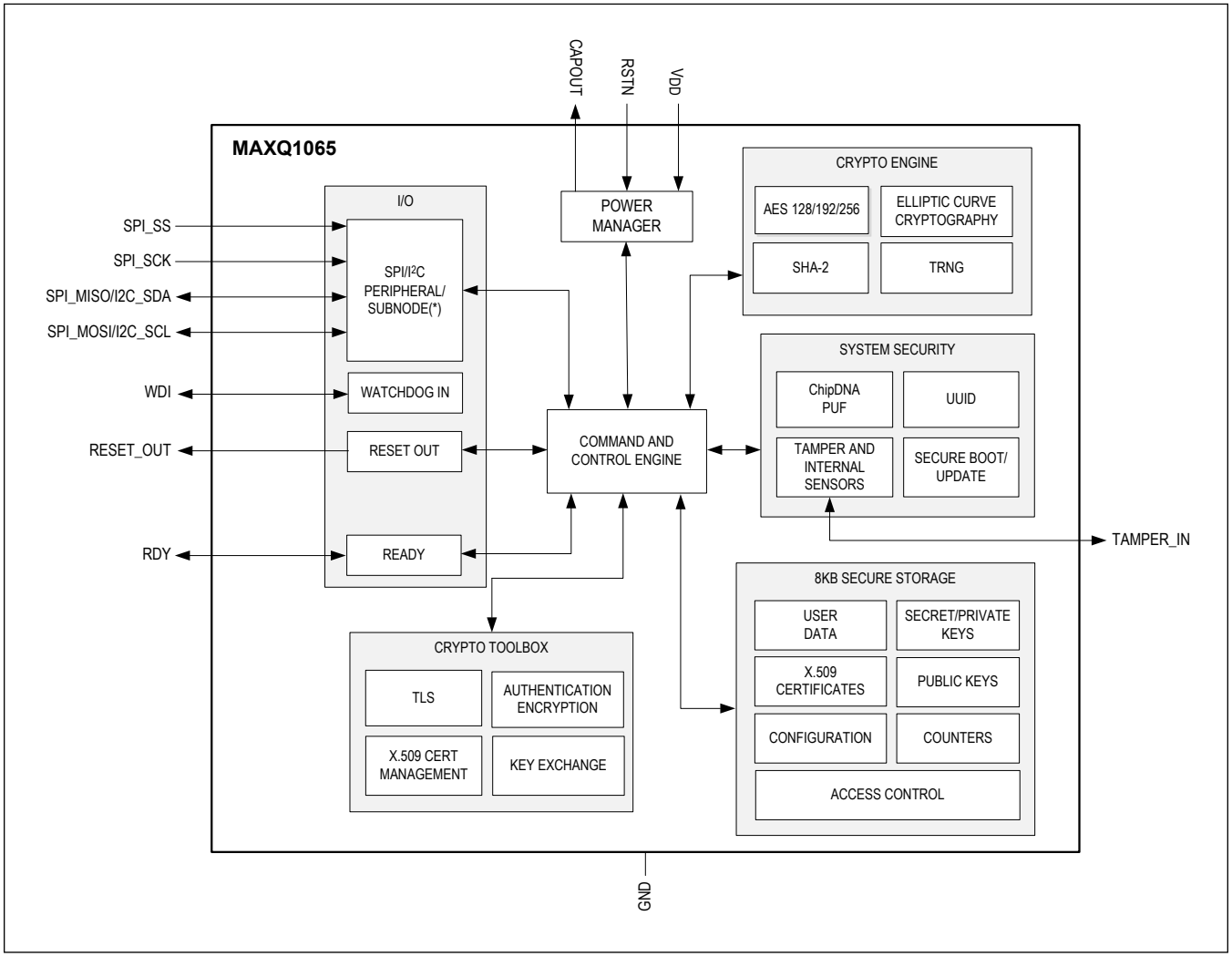
PIN		NAME	FUNCTION	TYPE
MAXQ1065-SPI	MAXQ1065-I2C			
POWER				
1	1	CEXT	External Capacitor. Connect to ground through a 1µF external ceramic chip capacitor. Place the capacitor as close as possible to the CEXT pin. No other components should be connected to the CEXT pin.	
6	6	GND	Digital Ground. Connect directly to the ground plane.	
7	7	PDWN	Power Down. Controls the power state of the MAXQ1065. Setting this pin to GND places the MAXQ1065 into power-down mode. In power-down mode, all volatile/ephemeral registers and data are erased. Set this pin high prior to communicating with the device. This pin should remain in a high state for the duration of any cryptography computations and as long as any ephemeral data/keys are required by the host application.	
12	12	V_{DD}	Supply Voltage. Connect to the external power supply for the MAXQ1065. Bypass to ground with 4.7µF and 0.1µF capacitors in parallel as close as possible to the V_{DD} pin.	
—	—	EP	Exposed Pad. Solder evenly to the board's ground plane for proper operation. Refer to the *Exposed Pads: A Brief Introduction* application note for additional information.	
SPI PERIPHERAL/SUBNODE				
2	—	SPIS_SCK	Peripheral/Subnode Clock (SCK). The SPI clock input from an external SPI main controller.	(MAXQ1065-SPI only)
3	—	SPIS_MOSI	Main Out Subnode In (MOSI). This is the SPI data input line from the SPI main controller.	(MAXQ1065-SPI only)
4	—	SPIS_MISO	Main In Subnode Out (MISO). This is the SPI data output line for data going from the MAXQ1065 to an external SPI main controller.	(MAXQ1065-SPI only)
5	—	SPIS_SS	Subnode Select (SS). An input from a SPI main controller to select the MAXQ1065 for communication.	(MAXQ1065-SPI only)
I²C PERIPHERAL				
—	2	SCL	I²C Clock I/O. An external 2.7K pull-up must be connected when I²C is selected.	(MAXQ1065-I²C only)
—	3	SDA	I²C Data I/O. An external 2.7K pull-up must be connected when I²C is selected.	(MAXQ1065-I²C only)
	4-5	DNC	Do Not Connect. Leave unconnected.	

Pin Description (continued)

PIN		NAME	FUNCTION	TYPE
MAXQ1065-SPI	MAXQ1065-I2C			
CONTROL				
8	8	TAMPER_IN	Tamper Detect Input (Optional Use). Defaults to an active-low input with weak pull-up to V_{DD}. Externally driving this pin low (0) triggers an optional tamper response. The tamper response is user configurable; for example, zeroization of the secret keys. If tamper detection is required, connect this pin to an external tamper sensor such as a switch triggered by unauthorized opening of the system enclosure. The tamper response is only detected if the MAXQ1065 is powered on, the PDWN pin is not asserted, and the part is in valid operating conditions. When enabled, a tamper response is triggered if the tamper pin is driven low. When not enabled, the pin can be left unconnected. TAMPER_IN is disabled by default.	
10	10	RESET_OUT	Reset Output (Optional Use). The output level is either asserted or pulsed when selected events occur (refer to the *MAXQ1065 User Guide*). The pin can output a user-configurable pulse to reset another microcontroller when the function is enabled. When not enabled (default), the RESET_OUT can be left unconnected (since it has a pull-up). When enabled, the RESET_OUT is configured as an open-drain input at all times. However, when an event triggers the RESET_OUT, the RESET_OUT pin is driven low or high by the MAXQ1065, depending on the configuration, for a duration that can also be configured through a command. Then, it is released and returns to open-drain mode.	
9	9	WDI	Watchdog Input (Optional Use). When the watchdog function is enabled, the MAXQ1065 monitors this pin. Watchdog is disabled by default. This pin can be left unconnected when not in use. Internal weak pull-up is enabled.	
11	11	RDY	Ready Output (Optional Use). The pin is set to a low level (0) when the MAXQ1065 is not ready to receive a new command, or is not ready to answer to the last command (the command is being processed). This pin is asserted by the MAXQ1065 (high level: 1): • After boot when the MAXQ1065 is ready to receive a new command, or • After the reception of a command when the processing is finished, and the response can be read by the host. This pin can be left unconnected when not used.	

Functional Diagram

Block Diagram



Detailed Description

The MAXQ1065 is a proven and efficient hardware root of trust for embedded systems. It guarantees the confidentiality, authenticity, and integrity of critical assets and private data, and it helps preserve software intellectual property and revenue models. It can be used in a wide range of industrial, medical, network, and computer peripheral devices such as IoT embedded devices, SCADA devices, PLC, IoT gateways and sensor nodes, network appliances, medical equipment, and wearables.

The MAXQ1065 is controlled over an SPI or I^2C interface. Its low pin count, reduced package size, low power consumption, and adaptable voltage range makes it easy to integrate into an existing board design. Its lifetime also makes it compatible with long-term deployments.

The software development kit (SDK) facilitates the integration of the MAXQ1065 functionality into the host microcontroller's firmware, without the need to deeply understand the communication protocol at the bit level.

The MAXQ1065 cryptographic toolbox supports an array of security needs. Simpler systems may require as little as the provided key generation and storage. For high levels of security, full TLS/DTLS support offers a high level of abstraction. This includes TLS/DTLS 1.2 client-side key negotiation (PSK, ECDH, ECDHE), ECDSA-based TLS/DTLS authentication, digital signature generation and verification, root and device X.509 certificate storage, on-chip peer certificate verification (ECDSA), TLS/DTLS packet encryption, and signature (AES-GCM/CCM). It can also serve as a secure bootloader for the host microcontroller of the system. In addition, it can bring device-level physical security through its tamper detection input, secure inputs/outputs, secure boot, and firmware updates.

The MAXQ1065 user-programmable nonvolatile memory securely stores certificates, public, private and secret keys, monotonic counters, and arbitrary data. Access rights are fully customizable and can be granted to separate stakeholders (device manufacturer, end-user, authenticated host processor, and key importer).

The MAXQ1065 life cycle management allows flexible access control rules during the major life cycle stages of the device. Secure key loading protocol and secure factory provisioning are available.

Software Collateral

Software Ecosystem

The MAXQ1065 comes with a complete software ecosystem in an SDK, enabling seamless software integration into the host microcontroller.

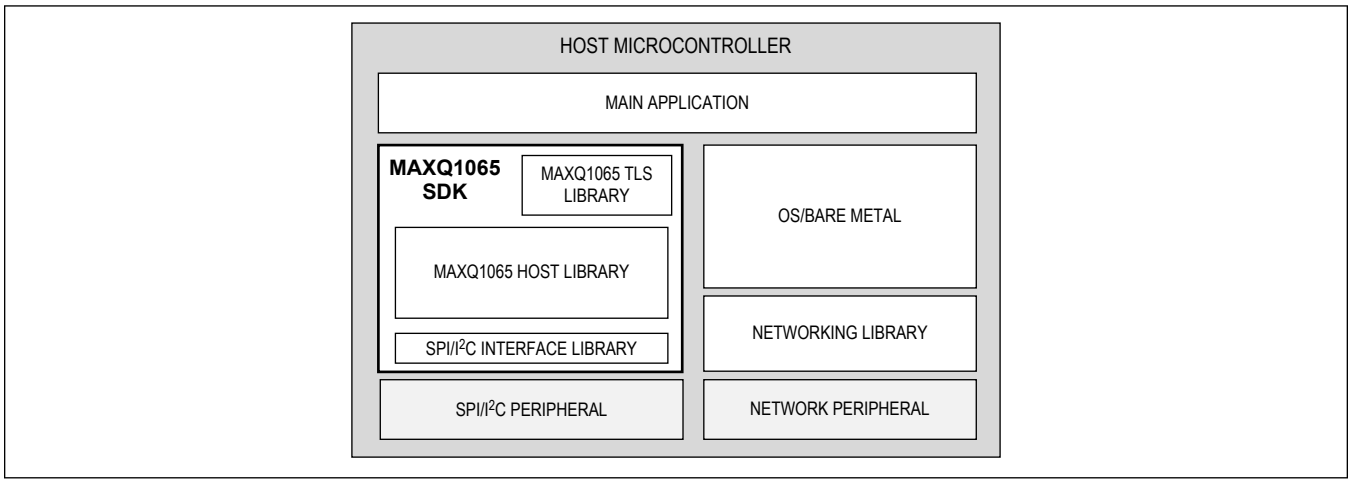


Figure 1. Software Offering

The SDK includes a complete host software solution including the host library, TLS libraries, and the SPI/I^2C interface library.

The host library maps the SPI/I2C commands of the MAXQ1065 into a convenient application programming interface in C language. The SPI/I2C interface library manages the communication protocol between the host microcontroller and the MAXQ1065.

On top of this host library, the MAXQ1065 TLS library is actually a choice between the industry-standard TLS libraries, both providing TLS1.2/DTLS 1.2 in client or server modes. In these TLS libraries, the security-sensitive processing of the TLS protocol is delegated to the MAXQ1065; therefore, the host microcontroller does not need to manipulate or store sensitive/secret data. The Mbed TLS and OpenSSL cryptographic libraries also use the MAXQ1065 as a "cryptographic engine," storing keys and certificates and running cryptographic algorithms in lieu of the host processor. This allows the main microcontroller's firmware to use the standard cryptographic APIs proposed by Mbed TLS or OpenSSL while taking advantage of the MAXQ1065 high-security and convenient provisioning.

The complete host software is implemented in C language, dependent only on the standard C library, with no dependence on the OS. In addition, the software SPI/I2C interface library layer, needed to interface physically with the MAXQ1065 through SPI/I2C and GPIOs, is clearly separated so it can be easily ported. The complete host software is adapted to bare-metal environments, the Arduino® development environment, Arm Mbed OS, Windows®, and Linux® (the provided software only runs in UserLAnd), and can easily be adapted to other environments.

The EV kit board allows easy interfacing with any existing system for rapid evaluation and prototyping. It connects to typical single board computers or microcontroller evaluation boards through standard connectors.

The SDK also includes:

- A simple PKI management and provisioning tool for testing purposes
- Basic examples and use cases
- TLS communication examples

The source code is accessible and reusable with nonconstraining software licenses.

Arduino is a registered trademark of Arduino, LLC.
Arm and Mbed are registered trademarks and service marks of Arm Limited.
Linux is a registered trademark of Linus Torvalds.
Windows is a registered trademark of Microsoft Corporation.

Provisioning Service

Analog Devices can preprogram each MAXQ1065 with unique key pairs and certificates, in addition to a set of common static data, to remove the complexity of deploying a certificate issuance system at the customer's factory. During this factory-programming process, key pairs are generated onboard the MAXQ1065, the private key never leaving the MAXQ1065. The device certificate is then prepared on board the MAXQ1065 by using the certificate authority's private signing key, and this certificate can be read back from the MAXQ1065. Additional static data and customer-specific administration keys are also loaded during provisioning. As a result, the MAXQ1065's ownership is transferred to the customer, and the system using the MAXQ1065 is able to connect through TLS to the targeted network infrastructure.

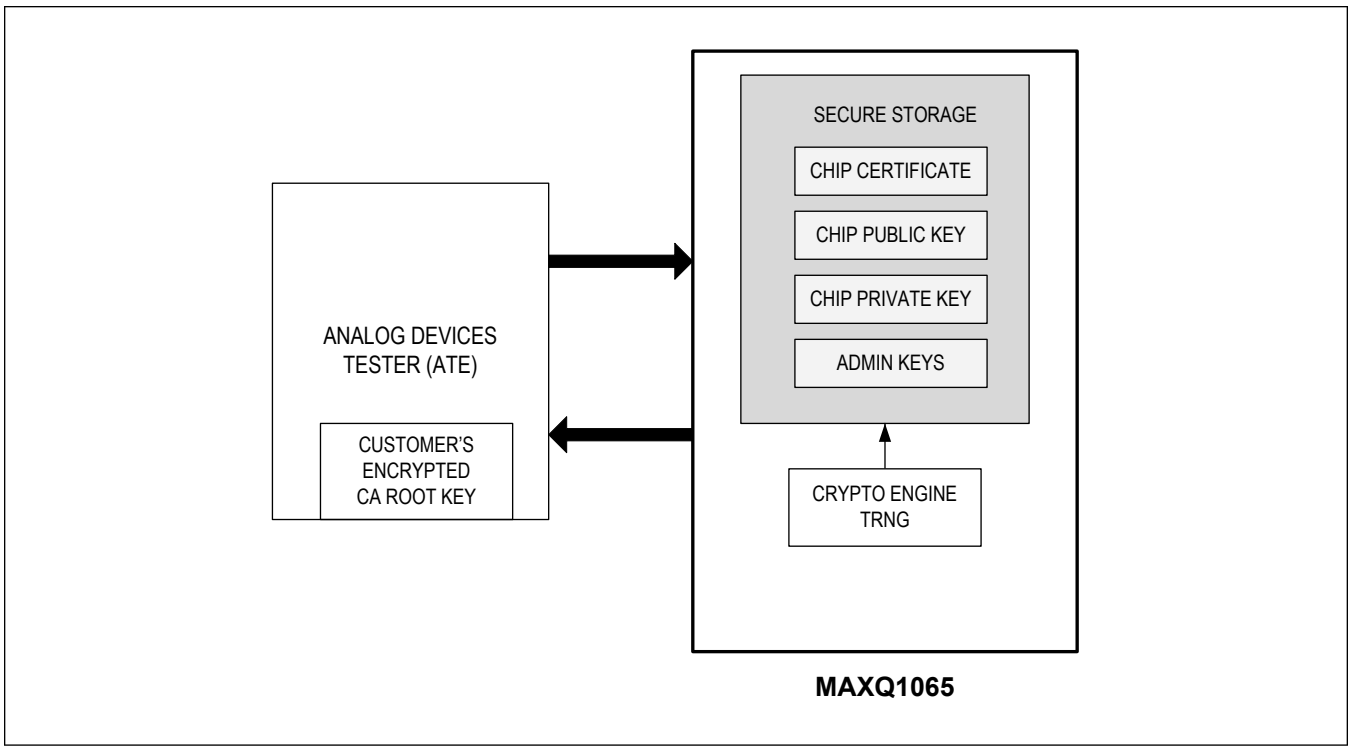


Figure 2. Provisioning

Software Integration

The MAXQ1065 comes with software for the host microcontroller that makes integration into any system very easy. The software enables a higher level of abstraction of the command set and the communication protocol.

Device Features

Cryptographic Toolbox for TLS

The MAXQ1065 cryptographic toolbox increases the security of TLS/DTLS based applications by providing:

- Verification of peer certificates and certificate revocation lists against trustworthy root certificates; root certificates are stored in the internal secure storage and can be securely updated
- ECDSA-based mutual authentication with other peers without exposing device private keys
- TLS handshake (PSK, ECDH, ECDHE) performed without revealing session keys
- Encryption/decryption and signature/verification of messages of the TLS record protocol with session keys
- TLS/DTLS supported algorithms
- TLS/DTLS key handshake
- ECDSA-based mutual authentication
- ECDSA X.509 on-board certificate verification
- TLS/DTLS packet encryption and signature (AES AEAD modes)
- Note: ECC algorithms run on the elliptic curve secp256r1
- Detailed list of supported TLS/DTLS 1.2 cipher suites:
  - RFC 5487:
    - TLS_PSK_WITH_AES_128_GCM_SHA256
  - RFC 6655:
    - TLS_PSK_WITH_AES_128_CCM
    - TLS_PSK_WITH_AES_256_CCM
    - TLS_PSK_WITH_AES_128_CCM_8

    - TLS_PSK_WITH_AES_256_CCM_8
- RFC 5289:
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- RFC 7251:
    - TLS_ECDHE_ECDSA_WITH_AES_128_CCM
    - TLS_ECDHE_ECDSA_WITH_AES_256_CCM
    - TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
    - TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8

Generic Cryptographic Services

The hardware crypto engine of the MAXQ1065 provides:

- Symmetric-key algorithms:
    - AES-128/192/256 (ECB, CBC, CCM, GCM)
- Ellipctic-curve cryptography on curves:
    - ECC NIST secp-256r1
- Secure hash algorithms:
    - SHA-256
- MAC digest algorithms:
    - AES-CBC-MAC
    - AES-CMAC
    - HMAC-SHA-256
- Signature schemes:
    - ECDSA signature and verification
- Key exchange algorithms:
    - TLS 1.2 ECDH_ECDSA, ECDHE_ECDSA, and PSK
    - SP800-56A-r3 static ECC CDH Diffie-Hellman with SP800-56A-r3 one-step KDF
- On-chip key generation:
    - ECC
    - AES
- Random number generation:
    - NIST SP800-90A/B/C
    - Direct conditioned entropy output with health tests

Unique Identifier

Each MAXQ1065 comes with a unique ID, allowing unique identification of the system that contains the MAXQ1065 over a network. A dedicated private key allows for verification of the authenticity of the MAXQ1065.

Secure Channel

The optional secure channel provides confidentiality and integrity of commands and responses exchanged with the host processor on the SPI/I^2C bus through:

- Key exchange between the host processor and the MAXQ1065, using preshared symmetric keys or ECC DH key exchange
- Signature and encryption of the commands and the responses

Tamper Detection

External tamper detection ensures information security at the system level. External tamper detection can selectively erase chosen data and can also assert the RESET_OUT pin if enabled.

Internal consistency checks also guarantee a safe processing of sensitive information within the MAXQ1065. An internal consistency error generates an immediate tamper response.

The TAMPER_IN pin can be connected to a user-defined tamper sensor, such as a switch, indicating a breach of the system enclosure.

Reset Output

A secure output can be controlled when user-configurable events occur, such as failure to perform a secure boot, failure to authenticate to a server, tamper event, internal consistency error, or positive events such as the successful initiation of a secure channel with the host processor or successful TLS authentication. The secure output can control device subsystems such as the system's host microcontroller's reset signal or an LED without requiring the host microcontroller intervention. The possible triggers can be one or more of the following:

- Secure boot failure/success
- Secure channel error/success
- Tamper detection
- Internal consistency error
- Power-on reset of the MAXQ1065
- Watchdog timer expiration

The MAXQ1065 can, therefore, act as an external security watchdog.

Secure Boot, Secure Update

The integrity of the host processor's data and code can be verified through the digital signature verification mechanism. The security policy of the MAXQ1065 can leverage this verification to grant or deny access to some assets, such as the MAXQ1065 specific private key used for TLS authentication, making the system unable to initiate a TLS connection if the firmware is not trusted.

Life Cycle Management

A managed life cycle changes functions and properties over time, as shown in Table 1. At each state of the life cycle, the device and parties are granted initialization, read, or modification rights to specific information. Transitions are always initiated by the administrator using a dedicated command. The life cycle is a useful tool to manage the security policy across the various life cycle stages. Selected keys can be zeroized when moving the life cycle backwards.

Key Loading Protocol

A secure key loading using encryption and authentication protocol allows key importation into the MAXQ1065 secure storage.

Watchdog

An optional watchdog feature can be enabled. When enabled, the MAXQ1065 monitors the WDI input pin for a regular toggling from the main devices' microcontroller. The absence of toggling within a defined time frame means that the main microcontroller is not operating properly.

Field Update

In a fast-moving security landscape, the internal firmware of the MAXQ1065 can be securely updated in order to secure long-term deployments, thanks to its secure update and secure boot mechanisms.

Secure Storage and Access Control

Secure storage is organized in a set of objects of different types as detailed in Table 1. In addition to the type, objects can be defined as volatile or nonvolatile, and also objects can get erased or not on an external tamper event. To be resistant to power loss during write operations, object modification is atomic. That means if a write operation is interrupted, the object will revert to its previous value and does not remain in an intermediate, corrupted state. Objects can be allocated and deallocated at will, and the free space is reclaimed.

Objects are stored with their own user-programmable, role-based, and life cycle state-dependent access conditions. Role authentication is based on challenge-response public key strong authentication. Roles all come with dedicated secure channels providing confidentiality, integrity, and anti-replay over the command interface. The security policy of an object is defined by the administrator using the Create Object command.

Key Storage

Key objects are stored in an integrity-protected manner and can never be read in the clear. They are automatically verified before use. Key pairs can be generated internally and stored in a persistent key pair object. Key pairs can also be generated externally and imported after successful signature verification using an imported public key present in the module. Arbitrary key pairs cannot be used; verification is mandatory.

Certificate Storage

Certificates are stored in an integrity-protected manner. They are automatically verified using one or more parent certificates in the certification chain (certificates already stored in the MAXQ1065). The device verifies the digital signature of the certificates and can extract their public key.

Arbitrary certificates cannot be stored; verification by a parent certificate or by a dedicated public key is mandatory. Since the device has limited certificate-parsing capabilities, the complete parsing of the X.509 certificates is done by the host processor when required.

Storable Objects

Table 1. Storable Objects

TYPE	READABLE (*)	COMMENT
Secret Key	No	Arbitrary symmetric keys are used in cryptographic algorithms. They can be imported or generated in place. They can be exported in an encrypted form using strict access control that uses authentication and encryption.
Public Key	Yes	Arbitrary public keys are used for the verification of key or root certificate importation requests, or for administrator authentication. Importation of public keys is strictly controlled with authentication.
Key Pair	Yes (public key) No (private key)	Arbitrary public and private key pairs are used in asymmetric cryptography algorithms. These key pairs can be imported into the MAXQ1065 or generated in place. The public key can be read out, but the private key is always protected against disclosure. Importation of key pairs is strictly controlled with authentication and encryption.
Transparent	Yes	Arbitrary user data for anything else.
Monotonic Counter	Yes	Increasing counter or decreasing counter. Used for implementing complementary life cycle system or managing number of system errors. Up to 10k write cycles are supported. Counters can be tied to key usage.
X.509 v3 Certificate	Yes	Certificates can be the MAXQ1065's own certificates (that should have been signed by a certification authority, and that can be matched with a key pair type of object also present in the object storage) or they can come from other entities such as PKI Certification authorities or other network peers. Certificates are used to reliably mutually authenticate with other peers. Certificates are protected against modification. Importation of certificates is strictly controlled with authentication.

Communication Interfaces and Power

Command and data transfer occurs over the SPI or I^2C bus. Data transfer is verified when writing and reading by a 16-bit cyclic redundancy check (CRC-16).

Ready Output

The command/response protocol can work using polling or using the MAXQ1065 ready output pin (RDY). When using polling, the host has to periodically send a polling request over the communication interface after the transmission of a command in order to be informed of its completion. The host can alternatively use the RDY pin (optionally using an interrupt-capable GPIO) to get informed of this event.

SPI

The serial peripheral interface (SPI) is a four-wire bus that provides fast, synchronous, full-duplex communication between the MAXQ1065 and the host system. The peripheral provides the following features:

- Peripheral/Subnode mode operation
- Active-low SSEL

- 10MHz (max) SPI peripheral clock speed
- Characters transmitted most significant bit (MSB) first
- Data protocol uses SPI Mode 0 and Mode 3

I²C (MAXQ1065A only)

The I²C bus is a bidirectional, two-wire serial bus that provides a medium-speed communications network. It can operate as a one-to-one, one-to-many, or many-to-many communication medium. It provides the following features:

- Peripheral mode operation
- Multiple transfer rates: 100kbps to 1Mbps
- Default address of 0x60 (configurable)
- Supports standard (7-bit) addressing
- Supports I²C clock stretching

Low-Power Mode

The MAXQ1065 enters low-power mode when the PDWN pin is asserted. Before entering low-power mode, the encrypted internal state of the MAXQ1065 can be read out by a specific command. After the PDWN pin is deasserted, the previously read encrypted internal state can be loaded back to the MAXQ1065 by another specific command.

Idle Mode

The MAXQ1065 automatically enters idle mode when it is waiting for a command. While in idle mode, the external tamper detection is still active. The MAXQ1065 wakes up automatically when the host starts sending a command to the MAXQ1065 through the selected communication interface or when a tamper event occurs.

ChipDNA Physically Unclonable Function (PUF)

ChipDNA PUF security technology provides an exponential increase in protection against the invasive and reverse engineering attacks that hackers use. Attempts to probe or observe ChipDNA operation modifies the underlying circuit characteristics, preventing the discovery of the unique value used by the chip cryptographic functions. Similarly, more exhaustive reverse-engineering attempts are defeated due to the factory conditioning required to make the ChipDNA PUF circuitry operational. The per-device unique key is generated by the ChipDNA PUF circuitry only when needed for cryptographic operations and is then instantaneously deleted.

Most importantly, the ChipDNA secure key never resides statically in registers or memory, nor does it ever leave the electrical boundary of the IC. In addition to the protection benefits, ChipDNA simplifies or eliminates the need for secure IC key management.

Development and Technical Support

Designers must have the following documents to use all the features of this device:

- This data sheet, which contains pin descriptions, feature overviews, and electrical specifications
- The device-appropriate user guide, which contains detailed information about the device features and operation
- Errata sheets for specific revisions noting deviations from published specifications
- The MAXQ1065 host library software and its inline documentation

SPI Modes

The MAXQ1065 supports SPI communications running in either of the following two SPI modes:

- Mode 0 (CPOL = 0, CPHA = 0): Data is sampled at the leading rising edge of the clock.
- Mode 3 (CPOL = 1, CPHA =1): Data is sampled on the trailing rising edge of the clock.

Details of the timing are described in Figure 3 and Figure 4.

If enabled, an autodetect feature is available to detect between Mode 0 and Mode 3. The feature works by checking if the SPIS_SCK signal is low (Mode 0) or high (Mode 3) before the falling edge of the SPIS_SS signal during the t_{SAD} time. Mode 1 and Mode 2 are not supported.

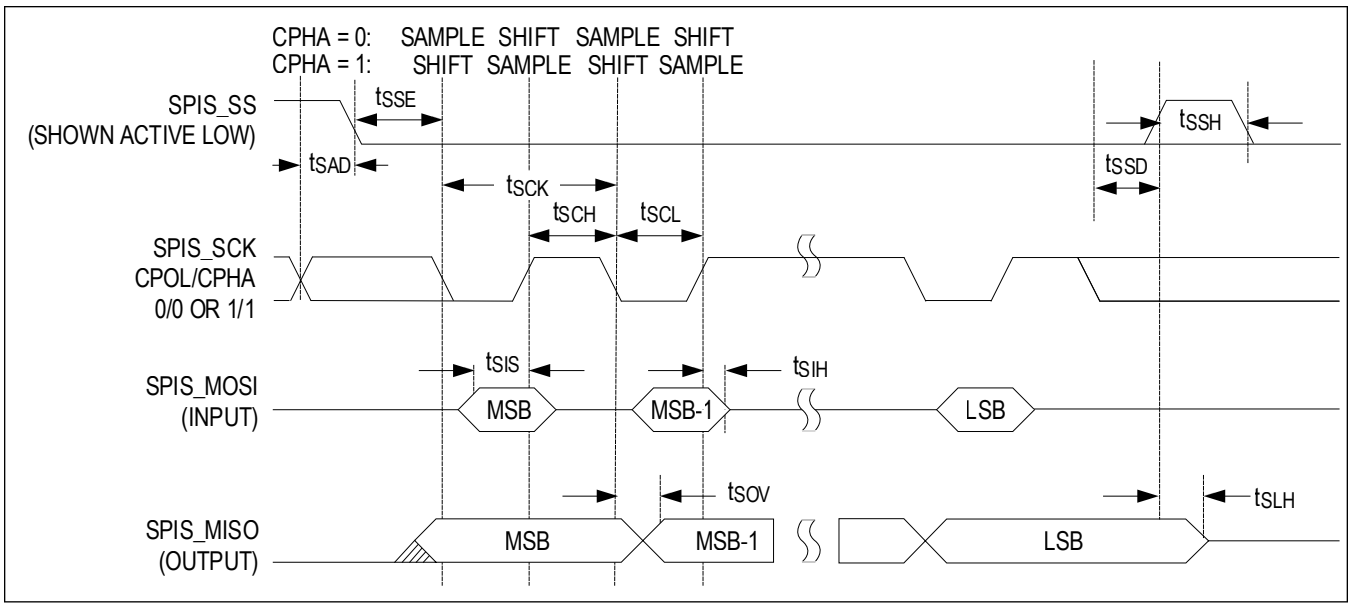


Figure 3. SPI Mode 0 and 3 Data Sampling Edges

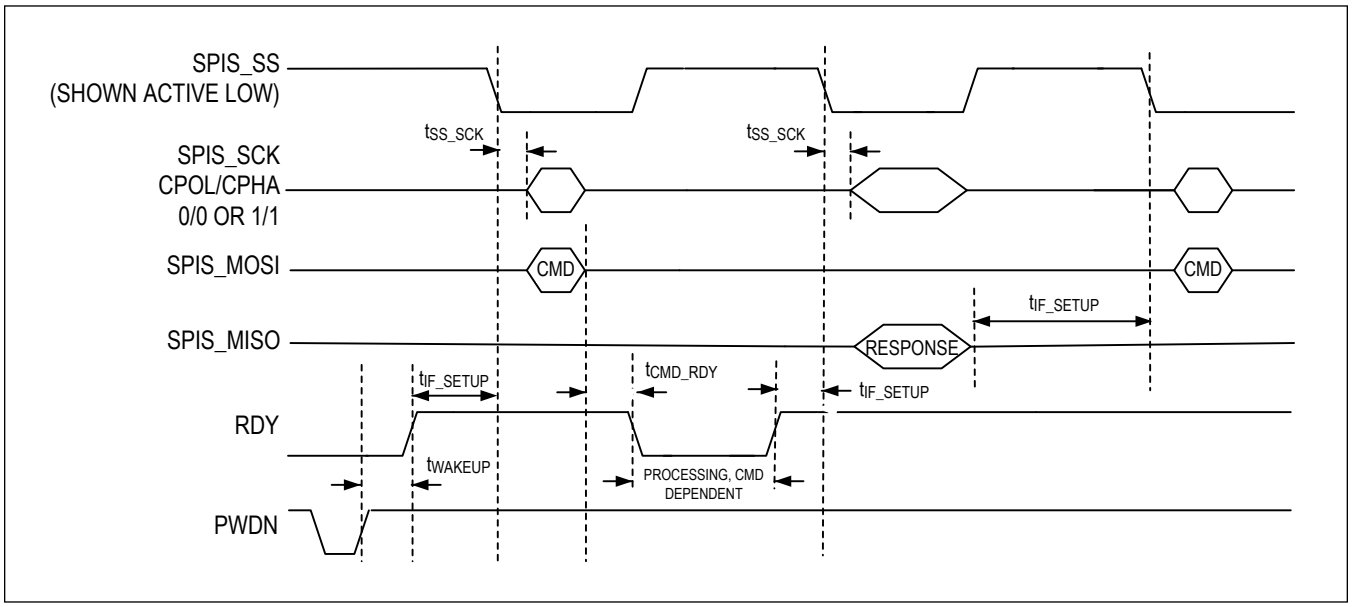


Figure 4. SPI Inter-command Timing

I²C

Overview

The I²C bus (MAXQ1065A only) uses a data line (SDA) plus a clock signal (SCL) for communication. Both SDA and SCL are bidirectional lines connected to a positive supply voltage through a pull-up resistor. When there is no communication, both lines are high. The output stages of devices connected to the bus must have an open drain or open collector to perform the wired-AND function. Data on the I²C bus can be transferred at rates of up to 100kbps in Standard mode, up to 400kbps in Fast mode, and up to 1Mbps in Fast-mode Plus.

A device that sends data on the bus is defined as a transmitter, and a device receiving data is defined as a receiver.

The device that controls the communication is called a controller. The devices that are controlled by the controller are peripherals. To be individually accessed, each device must have a peripheral address that does not conflict with other devices on the bus.

Data transfers can be initiated only when the bus is not busy. The controller generates the serial clock (SCL), controls the bus access, generates the START and STOP conditions, and determines the number of data bytes transferred between START and STOP, as seen in Figure 5. Data is transferred in bytes, with the MSB being transmitted first. After each byte follows an acknowledge bit to allow synchronization between controller and peripheral.

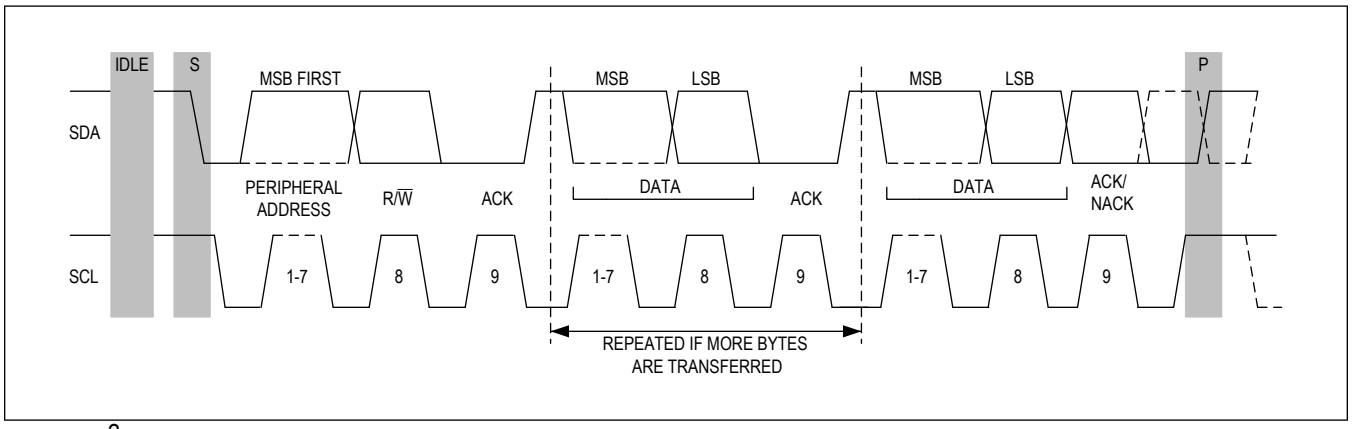


Figure 5. I²C Protocol Overview

I²C Definitions

The following terminology is commonly used to describe I²C data transfers. The timing references are defined in Figure 5.

Bus Idle or Not Busy

Both SDA and SCL are inactive and in their logic-high states.

START Condition

To initiate communication with a peripheral, the controller must generate a START condition. A START condition is defined as a change in state of SDA from high to low while SCL remains high.

STOP Condition

To end communication with a peripheral, the controller must generate a STOP condition. A STOP condition is defined as a change in state of SDA from low to high while SCL remains high.

Repeated START Condition

Repeated STARTs are commonly used for read accesses after having specified a memory address to read from in a preceding write access. The controller can use a repeated START condition at the end of a data transfer to immediately initiate a new data transfer following the current one. A repeated START condition is generated the same way as a normal START condition, but without leaving the bus idle after a STOP condition.

Data Valid

With the exception of the START and STOP condition, transitions of SDA can occur only during the low state of SCL. The data on SDA must remain valid and unchanged during the entire high pulse of SCL plus the required setup and hold time ($t_{HD:DAT}$ after the falling edge of SCL and $t_{SU:DAT}$ before the rising edge of SCL; see Figure 6). There is one clock pulse per bit of data. Data is shifted into the receiving device during the rising edge of the SCL pulse.

When finished with writing, the controller must release the SDA line for a sufficient amount of setup time (minimum $t_{SU:DAT,} + t_R$ in Figure 6) before the next rising edge of SCL to start reading. The peripheral shifts out each data bit on SDA at the falling edge of the previous SCL pulse, and the data bit is valid at the rising edge of the current SCL pulse.

The controller generates all SCL clock pulses, including those needed to read from a peripheral.

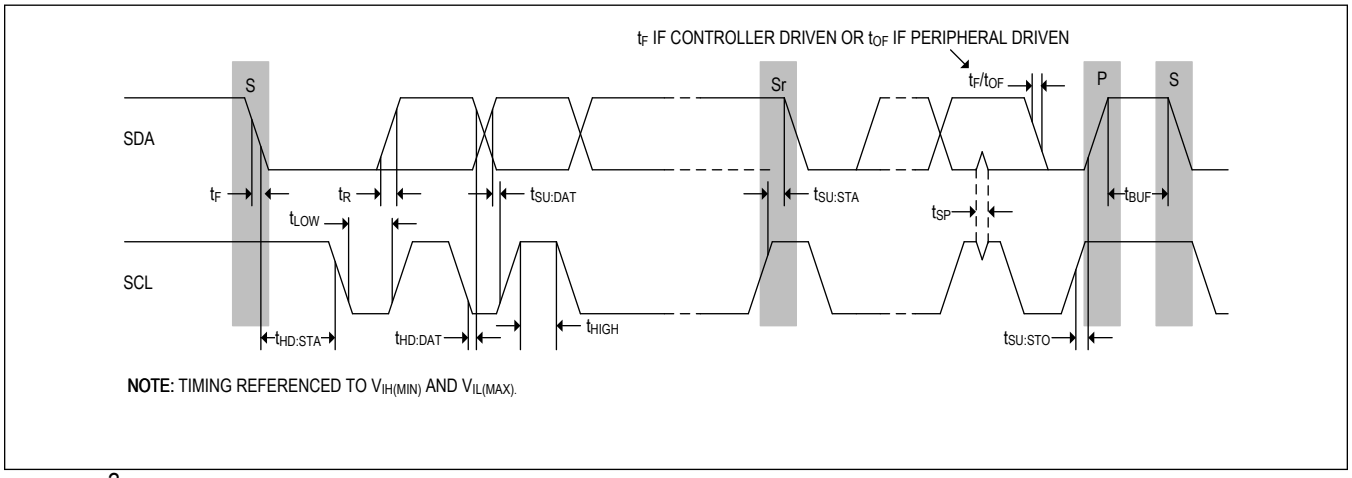
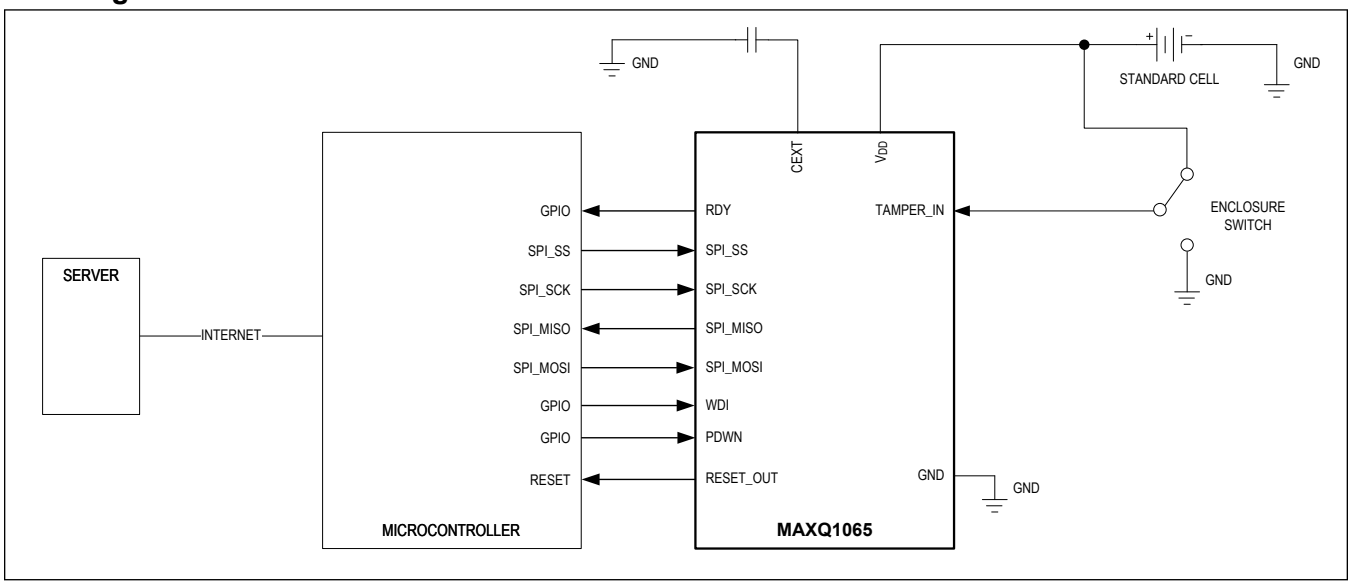


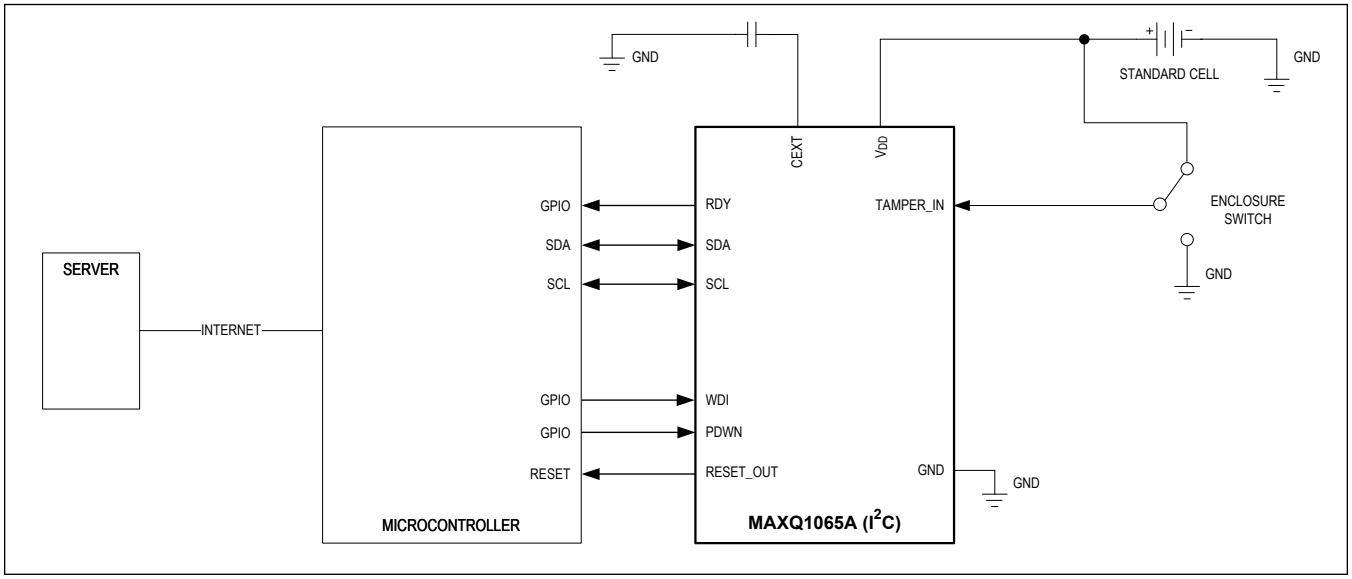
Figure 6. I²C Timing Diagram

Typical Application Circuit

Securing a Connected Device



Securing a Connected Device (I^2C)



Ordering Information

PART	PIN-PACKAGE	INTERFACE TYPE
MAXQ1065GTC+	12 TDFN	SPI
MAXQ1065GTC+T	12 TDFN	SPI
MAXQ1065AGTC+	12 TDFN	I^2C
MAXQ1065AGTC+T	12 TDFN	I^2C

+Denotes a lead(Pb)-free/RoHS-compliant package.

T = Tape and reel. Full reel.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	5/21	Initial release	—
1	4/25	Added I^2C	1–15, 17–23