Z-Accel 2.4 GHz ZigBee® Processor

Accelerate your ZigBee Development

Applications

- *ZigBee™ systems*
- *Home/Building automation*
- *Industrial control and monitoring*
- *Low power wireless sensor networks*
- *Set-top boxes and remote controls*
- *Automated Meter Reading*

Description

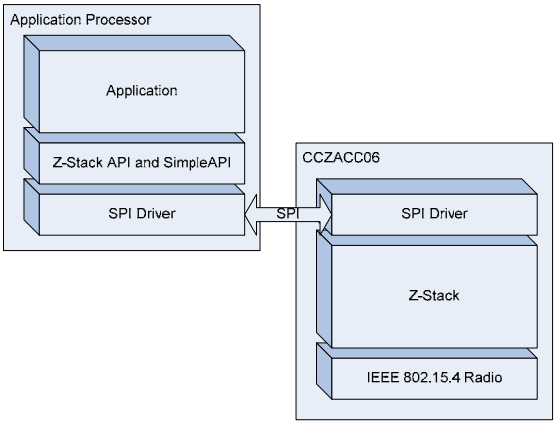
The ***CCZACC06*** is a cost-effective, low power, Z-Accel ZigBee Processor that provides full ZigBee functionality with a minimal development effort.

Z-Accel is a solution where TI's ZigBee stack, Z-Stack, runs on a ZigBee Processor and the application runs on an external microcontroller. The ***CCZACC06*** handles all the timing critical and processing intensive ZigBee protocol tasks, and leaves the resources of the application microcontroller free to handle the application.

Z-Accel makes it easy to add ZigBee to new or existing products at the same time as it provides great flexibility in choice of microcontroller.

CCZACC06 interfaces any microcontroller through an SPI or UART interface. There is no need to learn a new microcontroller or new tools. ***CCZACC06*** can for example be combined with an MSP430.

CCZACC06 supports TI's SimpleAPI. SimpleAPI has only 10 API calls to learn, which drastically simplifies the development of ZigBee applications.



Key Features

- *All the powerful features of the ZigBee system-on-chip with a simplified application interface.*
- *SPI or UART interface to host processor with SPI speeds up to 4 MHz.*
- *Designed for low power operation when using SPI interface with maximum time spent in low power mode when using SPI interface.*
- *Access to 12-bit analog-to-digital converter, gpio pins, non-volatile memory, hardware random number generator and upto 4 software timers.*

Table Of Contents

1 Abbreviations

API	Application Programming Interface.
AREQ	Asynchronous Request
CTS	Clear To Send
FCS	Frame Check Sequence
POLL	Poll request
RPC	Remote Procedure Call.

RTS	Ready To Send
SOF	Start Of Frame
SPI	Serial Peripheral Interface bus.
SREQ	Synchronous request
AF	ZigBee Application Framework
ZDO	ZigBee Device Objects

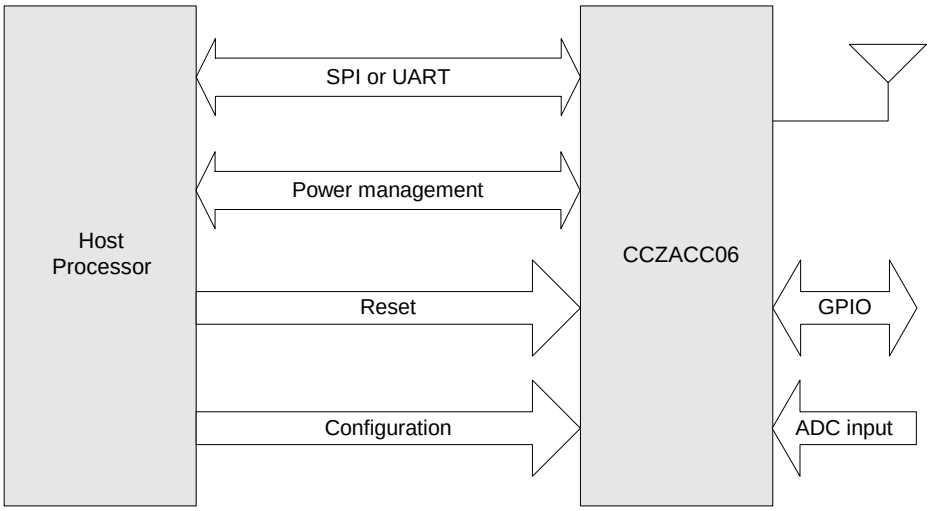
2 References

[1] CCZACC06 Data Sheet SWRS074
http://www.ti.com/lit/pdf/swrs074

3 Introduction

3.1 Architecture

The diagram below shows how a host processor interfaces with CCZACC06.

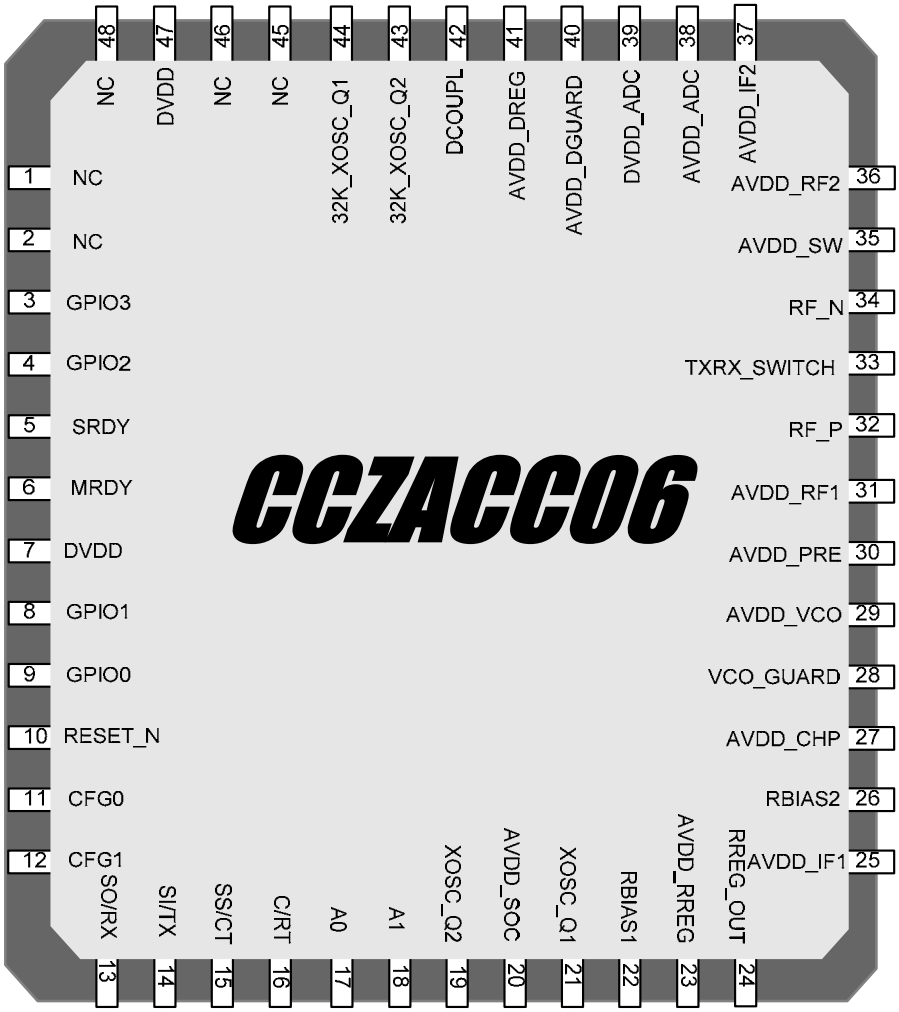


The SPI or UART physical interface is used to communicate between the two processors. The other interfaces are described below.

- Power Management: This interface consists of two signals (SRDY and MRDY) and is used to communicate the power management status and to wake up sleeping devices. This interface is only required if SPI transport is used.
- Reset: The host processor can reset CCZACC06 through the RESET_N pin. In addition, a software reset interface is provided.
- Configuration: This interface consists of the CFG0 and CFG1 pins on the CCZACC06 and is used to select SPI or UART transport and to select whether a 32 kHz crystal is installed. In addition, several other configuration parameters may be configured on the CCZACC06 through the software interface.
- ADC inputs: CCZACC06 has an onboard 12-bit ADC and 2 ADC input pins (A0 and A1). A software interface is provided for the host processor to perform an ADC conversion and read the value. A built-in temperature sensor and battery monitor can be also read through the ADC interface.
- GPIO: Four configurable GPIO pins (GPIO0-3) are available on CCZACC06. A software interface is provided for the host processor to read, write and toggle the GPIO pins.
- Non volatile parameters: This software interface allows the host processor to store and access 4 2-byte parameters and 2 16-byte parameters in the non volatile memory of CCZACC06.
- Software timers: Upto four software timers may be configured by the host processor on CCZACC06.

4 Pin configuration

The pin-out of the CCZACC06 is shown below.



Pin	Name	Direction	Connection	Description
3	GPIO3	Configurable	Optional	Configurable GPIO pin.
4	GPIO2	Configurable	Optional	Configurable GPIO pin.
5	SRDY	Out	Mandatory for SPI	Slave ready. Used for SPI transport.
6	MRDY	In	Mandatory for SPI	Master ready. Used for SPI transport.
8	GPIO1	Configurable	Optional	Configurable GPIO pin.
9	GPIO0	Configurable	Optional	Configurable GPIO pin.
10	RESET_N	In	Recommended	Reset, active low.
11	CFG0	In	Optional	Configuration input 0.
12	CFG1	In	Optional	Configuration input 1.
13	SO/RX	Out/In	Mandatory	SPI Slave output or UART RX data.
14	SI/TX	In/Out	Mandatory	SPI Slave input or UART TX data.
15	SS/CT	In/Out	Mandatory	SPI slave select or UART CTS.

16	C/RT	In	Mandatory	SPI clock or UART RTS.
17	A0	In	Optional	Analog input to ADC.
18	A1	In	Optional	Analog input to ADC.

GPIO 0-3: These are GPIO pins that can be configured and controlled by the host processor. See 6.1.11 for details.

SRDY: This signal is asserted by the CCZACC06 for power management and transaction control when using SPI transport. It is typically connected to a host processor GPIO with falling-edge interrupt capabilities. See 5.1.3 for details

MRDY: This signal is asserted by the host processor for power management and transaction control when using SPI transport. This is typically hardwired to SS/CT. See 5.1.3 for details.

RESET_N: This signal is used by the host processor to reset CCZACC06. Use of this signal is recommended over the software reset interface. See 7.1 for details on recommended reset procedures.

CFG0, CFG1: These two signals allow the host processor to configure the following features on CCZACC06 as shown below. The CCZACC06 reads these signals at powerup and operates accordingly.

CFG0	1	32 kHz crystal is installed. Default.
	0	32 kHz crystal is not installed.
CFG1	1	SPI transport selected. Default.
	0	UART transport selected.

SO/RX, SI/TX, SS/CT, C/RT: These are the standard signals used for SPI or UART communication. See 5.1 and 5.2 for details.

A0, A1: These are analog inputs to the CCZACC06 ADC. See 6.1.10 for details.

5 CCZACC06 Physical Interface

CCZACC06 supports either a SPI or UART transport interface to the host processor.

5.1 SPI Transport

5.1.1 Configuration

The following SPI configuration is supported by CCZACC06:

- SPI slave.
- Clock speed up to 4 MHz on CCZACC06.
- Clock polarity 0 and clock phase 0 on CCZACC06.
- Bit order MSB first on CCZACC06.

5.1.2 Frame Format

SPI transport uses the general frame format described in 5.3.

5.1.3 Signal Description

The following standard SPI signals are used:

- SCK: Serial clock.
- SS: Slave select.
- SI/TX: Master-output slave-input data.
- SO/RX: Master-input slave-output data.

The following additional signals are required for SPI transaction handling and power management:

- MRDY: Master ready. This signal is set by the host processor when it has data ready to send to the CCZACC06. It is active low. This signal can either be controlled independently or it can be hardwired to the slave select signal. The scenarios in this document assume MRDY is hardwired to SS.
- SRDY: Slave ready. This signal is set by the CCZACC06 when it is ready to receive or send data. When set low, it indicates the CCZACC06 is ready to receive data. When set high during an SPI POLL or SREQ transaction it indicates the CCZACC06 is ready to send data. When set high during an SPI AREQ transaction it indicates the CCZACC06 is done receiving data.

5.1.4 Signal Operation

The signals operate according to the following rules:

1. The host processor initiates a transaction by setting MRDY low and then waits for SRDY to go low.
2. The host processor shall never set MRDY high to end a transaction before all bytes of the frame have been transferred.
3. When receiving a POLL or SREQ, the CCZACC06 shall set SRDY high when it has data ready for the host processor.
4. When receiving an AREQ, the CCZACC06 shall set SRDY high when all bytes of the frame have been received.

5.1.5 Protocol Scenarios

5.1.5.1 AREQ Command

The following figure shows an AREQ command sent from the host processor to the CCZACC06.

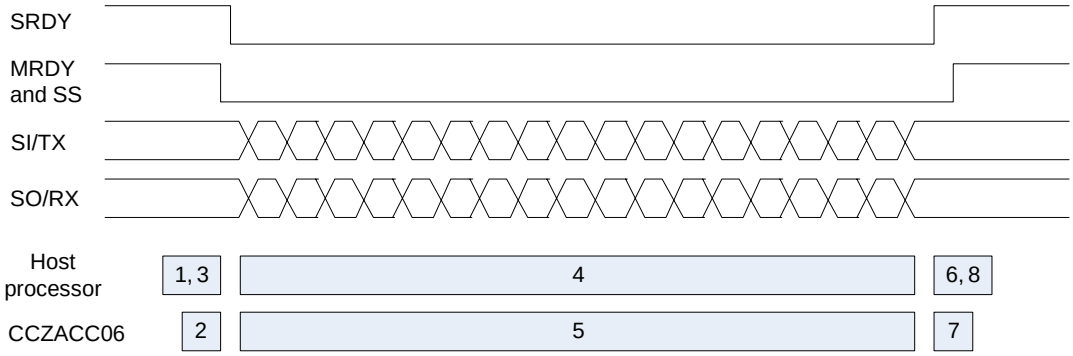


Figure 1

The following sequence of events occurs on the host processor and CCZACC06:

1. Host processor has an AREQ frame to send. Set MRDY low and wait for SRDY to go low.
2. CCZACC06 receives falling edge of MRDY. When ready to receive data set SRDY low.
3. Host processor reads SRDY low. Start data transmission.
4. Host processor transmits data until frame is complete.
5. CCZACC06 receives data until frame is complete.
6. Host processor waits for SRDY to go high.
7. CCZACC06 receives complete frame and sets SRDY high.
8. Host processor reads SRDY high. Set MRDY high.

5.1.5.2 POLL Command

The following figure shows a POLL command sent from the host processor to the CCZACC06.

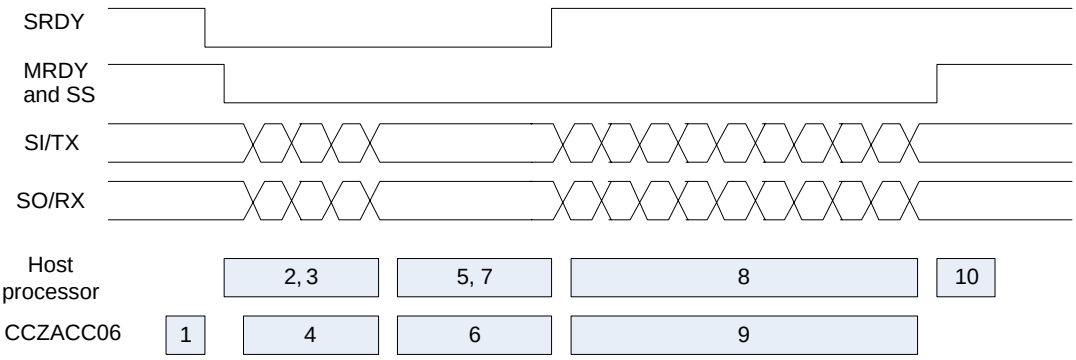


Figure 2

The following sequence of events occurs on the host processor and CCZACC06:

1. CCZACC06 has an AREQ frame to send. When ready to receive data set SRDY low.
2. Host processor detects SRDY low and sets MRDY low. Prepare POLL command and start data transmission.
3. Host processor transmits data until frame is complete.
4. CCZACC06 receives data until frame is complete.
5. Host processor waits for SRDY to go high.
6. CCZACC06 prepares AREQ frame for transmission. When ready to transmit set SRDY high.
7. Host processor reads SRDY high. Start data reception.
8. Host processor receives data until frame is complete.
9. CCZACC06 transmits data until frame is complete.
10. Host processor receives complete frame. Set MRDY high.

5.1.5.3 SREQ Command

The following figure shows a SREQ command sent from the host processor to the CCZACC06.

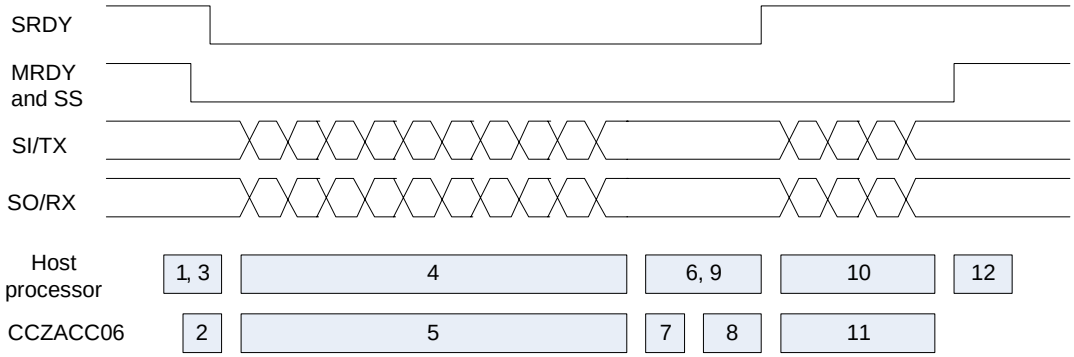


Figure 3

The following sequence of events occurs on the host processor and CCZACC06:

1. Host processor has an SREQ frame to send. Set MRDY low and wait for SRDY to go low.
2. CCZACC06 receives falling edge of MRDY. When ready to receive data set SRDY low.
3. Host processor reads SRDY low. Start data transmission.
4. Host processor transmits data until frame is complete.
5. CCZACC06 receives data until frame is complete.
6. Host processor waits for SRDY to go high.
7. CCZACC06 processes SREQ command and executes function
8. CCZACC06 prepares SRSP frame. When ready to transmit data set SRDY high.
9. Host processor reads SRDY high. Start data reception.
10. Host processor receives data until frame is complete.
11. CCZACC06 transmits data until frame is complete.
12. Host processor receives complete frame. Set MRDY high.

5.2 UART Transport

5.2.1 Configuration

The following UART configuration is supported:

- Baud rate: 115200.
- Hardware (RTS/CTS) flow control.
- 8N1 byte format.

5.2.2 Frame Format

UART transport frame format is shown in the following figure. The left-most field is transmitted first over the wire.

Bytes: 1	3-256	1
SOF	General format frame	FCS

Figure 4

SOF: Start of frame indicator. This is always set to 0xFE.

General frame format: This is the general frame format as described in 5.3.

FCS: Frame-check sequence. This field is computed as an XOR of all the bytes in the general format frame fields.

Shown below is a C example for the FCS calculation:

```
unsigned char calcFCS(unsigned char *pMsg, unsigned char len)
{
  unsigned char result = 0;
  while (len--)
  {
    result ^= *pMsg++;
  }
  return result;
}
```

5.2.3 Signal Description

The following standard UART signals are used:

- TXD: Transmit data.
- RXD: Receive data.
- CTS: Clear to send.
- RTS: Ready to send.
- The additional MRDY and SRDY signals are not used with UART transport as low power operation is not supported in this mode. The CCZACC06 CPU will always remain in active mode when UART transport is used.

5.2.4 Signal Operation

UART transport sends and receives data asynchronously. Data can be sent and received simultaneously and the transfer of a frame can be initiated at any time by either the host processor or the CCZACC06.

5.3 General Frame Format

The general frame format is shown in the following figure. The left-most field is transmitted first over the wire. For multi-byte fields, the lowest order byte is transmitted first.

Bytes: 1	2	0-253
Length	Command	Data

Figure 5

Length: The length of the data field of the frame. The length can range from 0-253.

Command: The command of the frame.

Data: The frame data. This depends on the command field and is described for each command in Section 6.

5.3.1 Command Field

The command field is constructed of two bytes. The bytes are formatted as shown in the following figure. The Cmd0 byte is transmitted first in a frame.

Cmd0		Cmd1
Bits: 7-5	4-0	7-0
Type	Subsystem	ID

Figure 6

Type: The command type has one of the following values:

- 0: POLL. A POLL command is used to retrieve queued data. This command is only applicable to SPI transport. For a POLL command the subsystem and ID are set to zero and data length is zero.
- 1: SREQ: A synchronous request that requires an immediate response. For example, a function call with a return value would use an SREQ command.
- 2: AREQ: An asynchronous request. For example, a callback event or a function call with no return value would use an AREQ command.
- 3: SRSP: A synchronous response. This type of command is only sent in response to a SREQ command. For an SRSP command the subsystem and ID are set to the same values as the corresponding SREQ. The length of an SRSP is generally nonzero, so an SRSP with length=0 can be used to indicate an error.
- 4-7: Reserved.

Subsystem: The subsystem of the command. Values are shown below:

Subsystem Value	Subsystem Name
0	Reserved
1	SYS interface
2	Reserved
3	Reserved
4	AF interface
5	ZDO interface

6	Simple API interface
7-32	Reserved

ID: The command ID. The ID maps to a particular interface message. Value range: 0-255.

6 CCZACC06 Application Interface

The following subsections describe the CCZACC06 application command interface. They are sub-divided into the following categories

- The SYS interface provides the application processor with a low level interface to the CCZACC06 hardware and software. The CCZACC06 functions that are accessible over this interface include the ADC (analog-to-digital converter), NV memory, GPIO pins, software timers and the hardware random number generator.
- The Configuration interface allows the application processor to configure various parameters of the CCZACC06 device.
- The Simple API interface is a simplified ZigBee interface that can be used to quickly create simple ZigBee compliant networked applications. It allows for easy device configuration, network formation, binding and data transfer.
- The AF and ZDO interfaces feature the complete ZigBee interface and can be used to create a full range of ZigBee compliant applications. The AF (Application Framework) interface allows the application processor to register its application with the CCZACC06 and send and receive data. The ZDO (ZigBee Device Objects) interface provides various ZigBee management functions like device and service discovery.

In all the message formats shown below, the left-most field is transmitted first over the wire. For multi-byte fields, the lowest order byte is transmitted first.

6.1 SYS interface

6.1.1 SYS_RESET_REQ

6.1.1.1 Description

This command is issued by the application processor to reset the CCZACC06 device. The reset is achieved through an internal watchdog reset on the CCZACC06. Note that the hardware reset interface is recommended over using this interface.

6.1.1.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x41	Cmd1 = 0x00	Type

Type – 1 byte – This should be set to value of 0.

6.1.2 SYS_RESET_IND

6.1.2.1 Description

This command is generated by the CCZACC06 device automatically immediately after a reset.

6.1.2.2 Usage

AREQ:

1	1	1	1	1	1
Length = 0x06	Cmd0 = 0x41	Cmd1 = 0x80	Reason	TransportRev	ProductId

1	1	1
MajorRel	MinorRel	HwRev

Reason – 1 byte – One of the following values indicating the reason for the reset.

Resolution	Value
Power-up	0x00
External	0x01
Watch-dog	0x02

TransportRev – 1 byte – Transport protocol revision. This is set to value of 2.

Product – 1 byte – Product ID. This is set to value of 1.

MajorRel – 1 byte – Major release number.

MinorRel – 1 byte – Minor release number.

HwRev – 1 byte – Hardware revision number.

6.1.3 SYS_VERSION

6.1.3.1 Description

This command is issued by the application processor to request for the CCZACC06 software version information.

6.1.3.2 Usage

SREQ:

1	1	1
Length = 0x00	Cmd0 = 0x21	Cmd1 = 0x02

SRSP:

1	1	1	1	1	1	1	1
Length = 0x05	Cmd0 = 0x61	Cmd1 = 0x02	TransportRev	Product	MajorRel	MinorRel	HwRev

TransportRev – 1 byte – The transport protocol revision number. This is set to value of 2.

Product – 1 byte – Product ID. This is set to value of 1.

MajorRel – 1 byte – Software major release number.

MinorRel – 1 byte – Software minor release number.

HwRev – 1 byte – Chip hardware revision.

6.1.4 SYS_OSAL_NV_READ

6.1.4.1 Description

This command is used by the application processor to read an item stored in the CCZACC06 NV memory. The command accepts an attribute ID value and returns the value for that attribute ID.

6.1.4.2 Usage

SREQ:

t	1	1	2	1
Length = 0x03	Cmd0 = 0x21	Cmd1 = 0x08	Id	Offset

Id – 2 bytes – The attribute id of the NV item. It can take one of the following values

NV Item	Size	Value
ZP_NV_APP_ITEM_1	2 bytes	0x0300
ZP_NV_APP_ITEM_2	2 bytes	0x0301
ZP_NV_APP_ITEM_3	2 bytes	0x0302

ZP_NV_APP_ITEM_4	2 bytes	0x0303
ZP_NV_APP_ITEM_5	16 bytes	0x0304
ZP_NV_APP_ITEM_6	16 bytes	0x0305

Offset – 1 byte – Number of bytes offset from the beginning or the NV value.

SRSP:

1	1	1	1	1	0-16
Length = 0x02-0x12	Cmd0 = 0x61	Cmd1 = 0x08	Status	Len	Value

Status – 1 byte – See 6.6 for a listing of the status values.

Len – 1 byte – The length of the NV value.

Value – 0-16 bytes – The value of the NV item.

6.1.5 SYS_OSAL_NV_WRITE

6.1.5.1 Description

This command is used by the application processor to write to an NV item in the CCZACC06 NV memory.

6.1.5.2 Usage

SREQ:

1	1	1	2	1	1	0-16
Length = 0x04-0x14	Cmd0 = 0x21	Cmd1 = 0x09	Id	Offset	Len	Value

Id – 2 bytes – The attribute id of the NV item. It can take one of the following values

NV Item	Size	Value
ZP_NV_APP_ITEM_1	2 bytes	0x0300
ZP_NV_APP_ITEM_2	2 bytes	0x0301
ZP_NV_APP_ITEM_3	2 bytes	0x0302
ZP_NV_APP_ITEM_4	2 bytes	0x0303
ZP_NV_APP_ITEM_5	16 bytes	0x0304
ZP_NV_APP_ITEM_6	16 bytes	0x0305

Offset – 1 byte - Number of bytes offset from the beginning or the NV value.

Len – 1 byte – Length of the NV value.

Value – 0-128 bytes – Value of the NV item.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x09	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.1.6 SYS_OSAL_START_TIMER

6.1.6.1 Description

This command is used by the application processor to start a software timer event in the CCZACC06. The event will expire after the indicated amount of time and a notification will be sent back to the application processor. Upto 4 software timers may be scheduled on the device.

6.1.6.2 Usage

SREQ:

1	1	1	1	2
Length = 0x03	Cmd0 = 0x21	Cmd1 = 0x0A	Id	Timeout

Id – 1 byte – The id of the timer event (0-3)

Timeout – 2 bytes – Amount of time (in milliseconds) before the event expires.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x0A	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.1.7 SYS_OSAL_STOP_TIMER

6.1.7.1 Description

This command is used by the application processor to stop a previously scheduled software timer event on the CCZACC06.

6.1.7.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x21	Cmd1 = 0x0B	Id

Id – 1 byte – The id of the timer event. It takes values form 0 through 3.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x0B	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.1.8 SYS_OSAL_TIMER_EXPIRED

6.1.8.1 Description

This command is sent by the CCZACC06 device to the application processor to indicate that a scheduled software timer event has expired.

6.1.8.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x41	Cmd1 = 0x81	Id

Id – 1 byte – The id of the timer event. It takes values form 0 through 3.

6.1.9 SYS_RANDOM

6.1.9.1 Description

This command is used by the application processor to read a 16-bit random number from the CCZACC06 hardware random number generator.

6.1.9.2 Usage

SREQ:

1	1	1
Length = 0x00	Cmd0 = 0x21	Cmd1 = 0x0C

SRSP:

1	1	1	2
Length = 0x02	Cmd0 = 0x61	Cmd1 = 0x0C	Value

Value – 2 bytes – The random value.

6.1.10 SYS_ADC_READ

6.1.10.1 Description

This command is used by the application processor to read from the CCZACC06 ADC (analog-to-digital converter).

6.1.10.2 Usage

SREQ:

1	1	1	1	1
Length = 0x02	Cmd0 = 0x21	Cmd1 = 0x0D	Channel	Resolution

Channel – 1 byte – The following channels are available.

Channel	Value
AIN0	0x06
AIN1	0x07
AIN0-1 (differential input)	0x0B
Temperature Sensor	0x0E
Voltage Reading	0x0F

Resolution – 1byte – The resolution of the ADC conversion. It can be 7-bit, 9-bit, 10-bit or 12-bit.

Resolution	Value
7-bit	0x00
9-bit	0x01
10-bit	0x02
12-bit	0x03

SRSP:

1	1	1	2
Length = 0x02	Cmd0 = 0x61	Cmd1 = 0x0D	Value

Value – 2 bytes – Value of the ADC conversion based on the specified information. This is a signed value in two's complement representation. Depending on the resolution of the conversion, the appropriate number of lowest order bits should be ignored. For example, for a 7-bit resolution, the lowest 9 bits should be ignored.

6.1.11 SYS_GPIO

6.1.11.1 Description

This command is used by the application processor to configure the accessible GPIO pins on the CCZACC06 device. There are four accessible GPIO pins (GPIO0-3) on the CCZACC06 device.

6.1.11.2 Usage

SREQ:

1	1	1	1	1
Length = 0x02	Cmd0 = 0x21	Cmd1 = 0x0E	Operation	Value

Operation - 1 byte - The type of operation to perform on the GPIO pins. It can take following values

Operation	Value	Description
Set direction	0x00	Configures the direction of the GPIO pins. A value of 0 in a bit position configures the corresponding GPIO pin as an Input while a value of 1 configures it as Output.
Set Input mode	0x01	Configures the Input mode of the GPIO pins. A value of 0 in a bit position configures it as pull-up mode while a 1 configures it in tri-state Input mode. (Note: GPIO0 and GPIO1 can only be set in tri-state input mode).
Set	0x02	A value of 1 in a bit position will set the corresponding GPIO pin (writes a 1).
Clear	0x03	A value of 0 in a bit position will clear the corresponding GPIO pin (writes a 0).
Toggle	0x04	A value of 1 in a bit position will toggle the corresponding GPIO pin.
Read	0x05	Reads the GPIO pins.

Value – 1 byte – Each bit position in this field contains the requested value for one of the GPIO pins. The GPIO pins are mapped to the bit positions as follows. The rest of the bit positions are ignored.

GPIO	Bit position
0	0
1	1
2	4
3	5

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x0E	Value

Value – 1 byte – The value after the requested operation is performed.

Note: The state of the GPIO pins after a reset is an Input with pull-up. When the device is in sleep mode, the GPIO pins maintain their output value and I/O mode.

6.1.12 SYS_TEST_RF

6.1.12.1 Description

This command is used by the application processor to configure the CCZACC06 device RF test modes. The test modes configure the frequency and transmit power of the CCZACC06 device and transmit either an unmodulated carrier or a carrier modulated with pseudo-random data or configure the radio for receive mode.

6.1.12.2 Usage

AREQ:

1	1	1	1	2	1
Length = 0x04	Cmd0 = 0x41	Cmd1 = 0x40	Test mode	Frequency	TxPower

Test mode – 1 byte - The type of test mode to configure the CCZACC06 device. It can take one of the following values

Test mode	Description
0x00	The CCZACC06 device will transmit unmodulated carrier with the specified frequency and transmit power
0x01	The CCZACC06 device will transmit pseudo-random data with the specified frequency and transmit power
0x02	The CCZACC06 device will have the radio placed in receive mode on the specified frequency.

Frequency – 2 bytes – The frequency of operation. The valid range is from 0 to 1023. The CCZACC06 device will set to (2048 + frequency) MHz.

TxPower – 1byte – The RF transmit power is determined by this parameter as described in the table below. This parameter is ignored in test mode 0x02.

TxPower	Output power [dBm]
0xFF	0.6
0xDF	0.5
0xBF	0.3
0x9F	0.2
0x7F	-0.1
0x5F	-0.4
0x3F	-0.9
0x1F	-1.5
0x1B	-2.7
0x17	-4.0
0x13	-5.7
0x0F	-7.9
0x0B	-10.8
0x07	-15.4
0x06	-18.6
0x03	-25.2

Note that executing this command will leave the radio in a different configuration than is needed for regular operation. It is expected that the device will have to be reset before it can be used again for regular RF operations.

6.1.13 SYS_TEST_LOOPBACK

6.1.13.1 Description

This command is used by the application processor to test the physical interface (SPI or UART) to the CCZACC06.

6.1.13.2 Usage

SREQ**:**

1	1	1	variable
Length = variable	Cmd0 = 0x21	Cmd1 = 0x41	Test data

Test data – variable bytes – This data will be returned by CCZACC06 in the response.

SRSP:

1	1	1	variable
Length = variable	Cmd0 = 0x61	Cmd1 = 0x41	Test data

Test data – variable bytes – The Test data from the request is returned.

6.2 Configuration interface

The CCZACC06 device has several parameters that can be configured by the host processor. These configuration parameters are stored in non volatile memory on the CCZACC06 device and their values persist across a device reset.

The configuration parameters are divided into “network-specific” and “device-specific” parameters. The “network-specific” configuration parameters should be set to the same value for all CCZACC06 devices in a ZigBee network to ensure proper network operation. The “device-specific” parameters can be set to different values on each device. These parameters are listed in detail in 6.2.3 and 6.2.4.

When the CCZACC06 device powers up, it reads two of the configuration parameters immediately. These are the `STARTOPT_CLEAR_CONFIG` bit (part of the `ZCD_NV_STARTUP_OPTION` parameter) and the `ZCD_NV_LOGICAL_TYPE` parameters. Any modification of these parameters will require a CCZACC06 device reset before they can take effect.

The rest of the configuration parameters are read when the CCZACC06 device starts operation of the ZigBee stack (when the `ZB_START_REQUEST` is issued).

Each of the configuration parameters has a default value that is used if it is not explicitly configured. It is possible to erase all the configuration settings and restore the device to this initial configuration by setting the `STARTOPT_CLEAR_CONFIG` bit option. This is useful if it is necessary to bring the CCZACC06 device configuration to a known state.

6.2.1 ZB_READ_CONFIGURATION

6.2.1.1 Description

This command is used to read the value of a configuration parameter from the CCZACC06 device.

6.2.1.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x26	Cmd1 = 0x04	ConfigId

ConfigId – 1 byte – Specifies the identifier for the configuration property.

SRSP:

1	1	1	1	1	1	0-128
Length = 0x03-0x83	Cmd0 = 0x66	Cmd1 = 0x04	Status	ConfigId	Len	Value

Status – 1 byte – See 6.6 for a listing of the status values.

ConfigId – 1 byte – Specifies the identifier for the configuration property.

Len – 1 byte – Specifies the size of the Value buffer in bytes.

Value – 0-128 bytes – A buffer to hold the configuration property.

6.2.2 ZB_WRITE_CONFIGURATION

6.2.2.1 Description

This command is used to write a configuration parameter to the CCZACC06 device.

6.2.2.2 Usage

SREQ:

1	1	1	1	1	1-128
Length = 0x03-0x83	Cmd0 = 0x26	Cmd1 = 0x05	ConfigId	Len	Value

ConfigId – 1 byte – The identifier for the configuration property

Len – 1 byte – Specifies the size of the Value buffer in bytes.

Value – 1-128 bytes – The buffer containing the new value of the configuration property

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x66	Cmd1 = 0x05	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.2.3 Device specific configuration parameters

6.2.3.1 ZCD_NV_STARTUP_OPTION

Configuration ID: 0x0003; Size: 1byte; Default value: 0

This parameter controls the device startup options. This is a bit mask of the following values

Bit position	7-2	1	0
Description	Reserved	STARTOPT_CLEAR_STATE	STARTOPT_CLEAR_CONFIG

- STARTOPT_CLEAR_CONFIG – If this option is set, the device will overwrite all the configuration parameters (except this one) with the “default” values that it is programmed with. This is used to erase the existing configuration and bring the device into a known state.

  Note:
  The `STARTOPT_CLEAR_CONFIG` bit is read by the CCZACC06 device immediately when it powers up after a reset.
  When the configuration parameters are restored to defaults, the ZCD_NV_STARTUP_OPTION itself is not restored except for clearing the STARTOPT_CLEAR_CONFIG bit.

- STARTOPT_CLEAR_STATE – If this option is set, the device will clear its previous network state (which would exist if the device had been operating on a network prior to the reset). This is typically used during application development. During regular device operation, this flag is typically not set, so that an accidental device reset will not cause loss of network state.

  Notes:
  The CCZACC06 device has two kinds of information stored in non-volatile memory. The configuration parameters (listed in this section) and network state information.
  The configuration parameters are configured by the user before start of ZigBee operation.
  The network state information is collected by the device after it joins a network and creates bindings etc. (at runtime). This is not set by the application processor. This information is stored so that if the

device were to reset accidentally, it can restore itself without going through all the network joining and binding process again.
If the application processor does not wish to continue operating in the previous ZigBee network, it needs to instruct the CCZACC06 device to clear the network state information and start again based on the configuration parameters. This is done by setting the STARTOPT_CLEAR_STATE bit in the startup option.

6.2.3.2 ZCD_NV_LOGICAL_TYPE

Configuration ID: 0x0087; Size: 1byte; Default value: 0x00

This is the logical type of the device in the ZigBee network. This can be set to either a COORDINATOR (0x00), ROUTER (0x01) or ENDDEVICE (0x02).

Note:
This parameter is read by the CCZACC06 device immediately when it powers up after a reset.

6.2.3.3 ZCD_NV_POLL_RATE

Configuration ID: 0x0024; Size: 2byte; Default value: 2000

If this parameter is set to a non-zero value, a CCZACC06 device that is configured as an end-device will wake up periodically with this duration to check for data with its parent device. This value is specified in milliseconds and can range from 1 to 65000.

If this parameter is set to zero, the device will not automatically wake up to poll for data. Instead, an external trigger or an internal event (for example, via a software timer event) can be used to wake up the device.

6.2.3.4 ZCD_NV_QUEUED_POLL_RATE

Configuration ID: 0x0025; Size: 2bytes; Default value: 100

When an end-device polls for data with its parent and finds that it does have data, it can poll again with a shorter duration in case there is more data queued for it at its parent device. This value is specified in milliseconds. This feature can be turned off by setting this value to zero.

6.2.3.5 ZCD_NV_RESPONSE_POLL_RATE

Configuration ID: 0x0026; Size: 2byte; Default value: 100

When an end-device sends a data packet, it can poll again with a shorter duration, specified by this parameter, if the application is expecting to receive an application level packet in response. This value is specified in milliseconds. This feature can be turned off by setting the value to zero.

Note: The setting of the queued and response poll rates has to be done with caution if the device is sending and receiving at the same time or if the device is sending data too fast.

If the device is sending data too fast, setting a queued poll rate with a higher duration than the sending rate will cause the poll event to be continuously rescheduled to the future. Then the device will never poll for data with its parent and consequently it may miss any packets destined for it.

6.2.3.6 ZCD_NV_POLL_FAILURE_RETRIES

Configuration ID: 0x0029; Size: 1byte; Default value: 2.

The number of times an end-device will fail when communicating with its parent before invoking the rejoin mechanism to find and join a new parent.

6.2.3.7 ZCD_NV_INDIRECT_MSG_TIMEOUT

Configuration ID: 0x002B; Size: 1byte; Default value: 7

The amount of time (in seconds) that a router or coordinator device will buffer messages destined to their end-device child nodes. It is recommended that this is at least greater than the poll rate (ZCD_NV_POLL_RATE) to ensure that end-device will have a chance to wakeup and poll for the data.

6.2.3.8 ZCD_NV_APS_FRAME_RETRIES

Configuration ID: 0x0043; Size: 1bytes; Default value: 3

The number of retransmissions performed on a data packet at the application layer if the packet was transmitted with the end-to-end acknowledgement option enabled.

6.2.3.9 ZCD_NV_APS_ACK_WAIT_DURATION

Configuration ID: 0x0044; Size: 2bytes; Default value: 3000

The amount of time (in milliseconds) a device will wait transmitting a packet with end-to-end acknowledgement option set for the acknowledgement packet to arrive from the destination device. If the acknowledgement packet is not received by this time, the sending device will assume a failure and attemp a retransmission.

Note: This is recommended to be set to approximately the expected round trip time for the packet. Note that if the destination (or source) device is an end-device, the round trip time for the packet will include an additional delay upto the poll duration. This is in addition to the delay normally caused by the network.

6.2.3.10 ZCD_NV_BINDING_TIME

Configuration ID: 0x0046; Size: 2bytes; Default value: 8000

The amount of time (in milliseconds) a device will wait for a response to a binding request.

6.2.3.11 ZCD_NV_USERDESC

Configuration ID: 0x0081; Size: 17bytes; Default value: "CCZACC06 x……" (dots represent the device IEEE address)

An optional user-defined data (upto 16bytes) that can be configured in a CCZACC06 device so that it can easily identified or described later. The first byte is the length of the user descriptor data and must not be greater than 16.

6.2.4 Network specific configuration parameters

6.2.4.1 ZCD_NV_PANID

Configuration ID: 0x0083; Size: 2bytes; Default value: 0xFFFF

This parameter identifies the ZigBee network. This should be set to a value between 0 and 0x3FFF. Networks that exist in the same vicinity must have different values for this parameter. It can be set to a special value of 0xFFFF to indicate *"don't care"*.

6.2.4.2 ZCD_NV_CHANLIST

Configuration ID: 0x0084; Size: 4bytes; Default value: 0x00000800

This parameter is a bit mask of the channels on which this network can operate (note that multiple channels can be selected). Multiple networks that exist in the same vicinity are encouraged to have different values.

If multiple channels are selected, the coordinator will pick one of the channels for network operation. First, an energy scan is performed on each channel and those channels with a high

energy level are discarded. Then, the coordinator deteremines the number of existing ZigBee networks on each of the remaining channels and picks the one with the fewest networks. For routers and end-devices, the device will simply scan all the selected channels until it finds the ZigBee network.

6.2.4.3 ZCD_NV_PRECFGKEY

Configuration ID: 0x0062; Size: 16bytes; Default value: [0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F]

This is used for securing and un-securing packets in the network, if security is enabled for the network.

6.2.4.4 ZCD_NV_PRECFGKEYS_ENABLE

Configuration ID: 0x0063; Size: 1byte; Default value: TRUE

If security functionality is enabled, there are two options to distribute the security key to all devices in the network.

If this parameter is true, the same security key is assumed to be pre-configured in all devices in the network.

If it is set to false, then the key only needs to be configured on the coordinator device. In this case, the key is distributed to each device upon joining by the coordinator. This key distribution will happen in the “clear” on the last hop of the packet transmission and this constitutes a brief “period of vulnerability” during which a malicious device can capture the key. Hence it is not recommended unless it can be ensured that there are no malicious devices in the vicinity at the time of network formation.

6.2.4.5 ZCD_NV_SECURITY_MODE

Configuration ID: 0x0064; Size: 1byte; Default value: 0

This parameter determines if security is used or not in this network. It can be set to 0 (to turn off security) or 1 (to turn on security).

6.2.4.6 ZCD_NV_BCAST_RETRIES

Configuration ID: 0x002E; Size: 1byte; Default value: 2.

The maximum number of retransmissions that a device will attempt when trying to transmit a broadcast packet. The typical range is from 1 through 3.

6.2.4.7 ZCD_NV_PASSIVE_ACK_TIMEOUT

Configuration ID: 0x002F; Size: 1byte; Default value: 5

The amount of time (in units of 100milliseconds) a device will wait before retransmitting a broadcast packet. The retransmission will not happen if the node hears that each of its neighbor nodes have all transmitted that packet.

6.2.4.8 ZCD_NV_BCAST_DELIVERY_TIME

Configuration ID: 0x0030; Size: 1byte; Default value: 30.

The maximum amount of time (in units of 100ms) that it can take for a broadcast packet to propagate through the entire network. This includes time for all retransmissions.

Note: This parameter must be set with caution. It must be set to a value of at least

*(**ZCD_NV_BCAST_RETRIES** + 1) * **ZCD_NV_PASSIVE_ACK_TIMEOUT***

To be safe, the actual value should be higher than the above minimum by about 500ms or more.

6.2.4.9 ZCD_NV_ROUTE_EXPIRY_TIME

Configuration ID: 0x002C; Size: 1byte; Default value: 60.

The amount of time (in seconds) for which a route must be idle (*i.e.* no packets are transmitted on that route) before that routing entry is marked as expired. An expired entry may be deleted if the table is full and the space is needed for another new routing entry.

This can be set to a special value of 0 to turn off route expiry. In this case, route entries are not expired.

6.3 Simple API interface

The Simple API interface is intended to present a simplified ZigBee API to the application developer. The complete ZigBee interface is provided via the AF and ZDO interfaces. But since the majority of the applications do not use the full feature set available in ZigBee, this simplified interface is an easy way for the developer to begin ZigBee application development. It contains the necessary interface to commission a ZigBee network, perform bindings between devices and send and receive data.

6.3.1 ZB_APP_REGISTER_REQUEST

6.3.1.1 Description

This command enables the application processor to register its application with the CCZACC06 device.

6.3.1.2 Usage

SREQ:

1	1	1	1	2	2	1
Length = variable	Cmd0 = 0x26	Cmd1 = 0x0A	AppEndPoint	AppProfileID	DeviceId	DeviceVersion

1	1	2 x Input commands	1	2 x Output commands
Unused	InputCommandsNum	InputCommandsList	OutputCommandsNum	OutputCommandsList

AppEndPoint – 1 byte – Specifies the endpoint of the device. This should be in the range of 1 through 240 and should be set to same value for all devices in the network.

AppProfileID – 2 bytes – Specifies the profile id of the application. This should be set to same value to all devices in the network. This number is assigned by the ZigBee Alliance.

DeviceId – 2 bytes – This is an application-specific identifier. It identifies the device type within the particular profile id. This is not used by the ZigBee stack in any way other than to identify itself when requested.

DeviceVersion – 1 byte – This is an application-specific identifier. It identifies the version of the device. This is not used by the ZigBee stack in any way other than to identify itself when requested.

Unused – 1 byte – Unused parameter.

InputCommandsNum – 1 byte – Specifies the number of Input commands that this application will process.

InputCommandsList – variable – List of input command identifiers that are processed by this application.

OutputCommandsNum – 1 byte – Specifies the number of Output commands that this application will generate.

OutputCommandsList – variable – List of output command identifiers that are generated by this application.

Note:

It is mandatory that the application register itself with the CCZACC06 device after every reset if it wishes to use the simple api interface. The AppEndPoint and AppProfileId fields should be populated with the same values for all devices in the network.

This list of input and output commands are only used by the ZigBee stack when performing binding. If the binding feature is not used, these may be ignored.

The DeviceId and DeviceVersion are not used by the ZigBee stack and may be ignored.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x66	Cmd1 = 0x0A	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.3.2 ZB_START_REQUEST

6.3.2.1 Description

This command starts the ZigBee stack in the CCZACC06 device. When the ZigBee stack starts, the device reads the programmed configuration parameters and operates accordingly. After the start request process completes, the device is ready to send, receive, and route network traffic.

6.3.2.2 Usage

SREQ:

1	1	1
Length = 0x00	Cmd0 = 0x26	Cmd1 = 0x00

SRSP:

1	1	1
Length = 0x00	Cmd0 = 0x66	Cmd1 = 0x00

6.3.3 ZB_START_CONFIRM

6.3.3.1 Description

This command is issued by the CCZACC06 device to return the results from a ZB_START_REQUEST command.

6.3.3.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x46	Cmd1 = 0x80	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.3.4 ZB_PERMIT_JOINING_REQUEST

6.3.4.1 Description

This command is used to control the joining permissions and thus allow or disallow new devices from joining the network.

6.3.4.2 Usage

SREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x26	Cmd1 = 0x08	Destination	Timeout

Destination – 2 bytes – The destination parameter indicates the address of the device for which the joining permissions should be set. This is usually the local device address or the special broadcast address that denotes all routers and coordinator (0xFFFC). This way the joining permissions of a single device or the whole network can be controlled.

Timeout – 1 byte – Indicates the amount of time in seconds for which the joining permissions should be turned on. If timeout is set to 0x00, the device will turn off the joining permissions indefinitely. If it is set to 0xFF, the joining permissions will be turned on indefinitely.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x66	Cmd1 = 0x08	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.3.5 ZB_BIND_DEVICE

6.3.5.1 Description

This command is used to create or delete a 'binding' to another device on the network. Once bound, an application can send messages to a device by referencing the commandId for the binding. This command can also be issued with a NULL destination address (set to all zeros). In that case, a binding will be established with another device that is in the Allow Bind mode.

6.3.5.2 Usage

SREQ:

1	1	1	1	2	8
Length = 0x0B	Cmd0 = 0x26	Cmd1 = 0x01	Create	CommandId	Destination

Create – 1 byte – TRUE to create a binding, FALSE to remove a binding.

CommandId – 2 bytes – The identifier of the binding.

Destination – 8 bytes – Specifies the 64-bit IEEE address of the device to bind to. Set to NULL if the destination address is unknown and instead the destination device is set to Allow Bind mode.

SRSP:

1	1	1
Length = 0x00	Cmd0 = 0x66	Cmd1 = 0x01

6.3.6 ZB_BIND_CONFIRM

6.3.6.1 Description

This command is issued by the CCZACC06 device to return the results from a `ZB_BIND_DEVICE` command.

6.3.6.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x46	Cmd1 = 0x81	CommandId	Status

CommandId – 2 bytes – The command ID of the binding being confirmed.

Status – 1 byte – See 6.6 for a listing of the status values.

6.3.7 ZB_ALLOW_BIND

6.3.7.1 Description

This command puts the device into the Allow Binding Mode for a given period of time. This allows a peer device to establish a binding with this device (in the Allow Binding Mode) by issuing the zb_BindDevice with a destination address of NULL.

6.3.7.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x26	Cmd1 = 0x02	Timeout

Timeout – 1 byte – The number of seconds to remain in the allow binding mode. Valid values range from 1 through 65. If 0, the Allow Bind mode will be set false without timeout. If greater than 64, the Allow Bind mode will be true indefinitely.

SRSP:

1	1	1
Length = 0x00	Cmd0 = 0x66	Cmd1 = 0x02

6.3.8 ZB_ALLOW_BIND_CONFIRM

6.3.8.1 Description

This command is issued by the CCZACC06 device when it responds to a bind request from a remote device.

6.3.8.2 Usage

AREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x46	Cmd1 = 0x82	Source

Source – 2 bytes – Contains the address of the device attempted to bind to this device.

6.3.9 ZB_SEND_DATA_REQUEST

6.3.9.1 Description

This command initiates transmission of data to another device in the network. This command can only be issued after the host processor has registered its application using the ZB_APP_REGISTER_REQUEST and the device has successfully created or joined a network.

6.3.9.2 Usage

SREQ:

1	1	1	2	2	1
Length = 0x08-0x5C	Cmd0 = 0x26	Cmd1 = 0x03	Destination	CommandId	Handle

1	1	1	0-84
Ack	Radius	Len	Data

Destination – 2 bytes – The destination address of the data packet. It can be one of the following values.

Address	Description
0 – 0xFFF7	16-bit short address of the destination device
0xFFFC	Group of all routers and coordinator

0xFFFD	Group of all devices with receiver turned on
0xFFFE	This is the binding address and should be used when a binding entry has been previously created for this particular CommandId. The destination address will be determined from the binding table by the CCZACC06
0xFFFF	Broadcast group of all devices in the network

CommandId – 2 bytes – The command ID to send with the message. If the binding address is used for destination, this parameter also indicates the binding to use.

Handle – 1 byte – A handle used to identify the send data request.

Ack – 1 byte – TRUE if requesting acknowledgement from the destination.

Radius – 1 byte – The max number of hops the packet can travel through before it is dropped.

Len – 1 byte – Specifies the size of the Data buffer in bytes.

Data – 0-84 bytes – Data.

SRSP:

1	1	1
Length = 0x00	Cmd0 = 0x66	Cmd1 = 0x03

6.3.10 ZB_SEND_DATA_CONFIRM

6.3.10.1 Description

This command is issued by the CCZACC06 device to return the results from a ZB_SEND_DATA_REQUEST command.

6.3.10.2 Usage

AREQ:

1	1	1	1	1
Length = 0x02	Cmd0 = 0x46	Cmd1 = 0x83	Handle	Status

Handle – 1 byte – Specifies the handle.

Status – 1 byte – See 6.6 for a listing of the status values.

6.3.11 ZB_RECEIVE_DATA_INDICATION

6.3.11.1 Description

This callback is called asynchronously by the CCZACC06 device when it has received a packet from a remote device.

6.3.11.2 Usage

AREQ:

1	1	1	2	2	2	0-84
Length = 0x06-5A	Cmd0 = 0x46	Cmd1 = 0x87	Source	Command	Len	Data

Source – 2 bytes – Specifies the short address of the peer device that sent the data.

Command – 2 bytes – The command ID associated with the data.

Len – 2 bytes – Specifies the number of bytes in the Data parameter.

Data – Array of bytes – The data sent by the peer device

6.3.12 ZB_GET_DEVICE_INFO

6.3.12.1 Description

This command retrieves a Device Information Property.

6.3.12.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x26	Cmd1 = 0x06	Param

Param – 1 byte – The identifier of the device information. It can take one of the following values.

Parameter	Size	Description
0	1 byte	Device state
1	8 bytes	Device IEEE address
2	2 bytes	Device short address
3	2 bytes	Short address of the parent device
4	8 bytes	IEEE address of the parent device
5	1 byte	Channel on which the ZigBee network is operating
6	2 bytes	PAN ID of the ZigBee network
7	8 bytes	Extended PAN Id of the ZigBee network

SRSP:

1	1	1	1	8
Length = 0x09	Cmd0 = 0x66	Cmd1 = 0x06	Param	Value

Param – 1 byte – The identifier of the requested device information.

Value – 8 byte – The value of the requested device information. This is always 8bytes in length even though the actual value may be smaller in size.

6.3.13 ZB_FIND_DEVICE_REQUEST

6.3.13.1 Description

This command is used to determine the short address for a device in the network. The device initiating a call to zb_FindDeviceRequest and the device being discovered must both be a member of the same network. When the search is complete, the zv_FindDeviceConfirm callback function is called.

SREQ:

1	1	1	8
Length = 0x08	Cmd0 = 0x26	Cmd1 = 0x07	SearchKey

SearchKey – 8 bytes – Specifies the value to search on.

SRSP:

1	1	1
Length = 0x00	Cmd0 = 0x66	Cmd1 = 0x07

6.3.14 ZB_FIND_DEVICE_CONFIRM

6.3.14.1 Description

This command is issued by the CCZACC06 device to return the results from a `ZB_FIND_DEVICE_REQUEST` command.

6.3.14.2 Usage

AREQ:

1	1	1	1	2	8
Length = 0x0B	Cmd0 = 0x46	Cmd1 = 0x85	SearchType = 0x01	SearchKey	Result

SearchType – 1 byte – The type of search that was performed.

SearchKey – 2 bytes – Value that the search was executed on.

Result – 8 bytes – The result of the search.

6.4 AF interface

6.4.1 AF_REGISTER

6.4.1.1 Description

This command enables the host processor to register an application endpoint description with the CCZACC06.

6.4.1.2 Usage

SREQ:

1	1	1	1	2	2	1
Length = 0x09-0x49	Cmd0 = 0x24	Cmd1 = 0x00	EndPoint	AppProfId	AppDeviceId	AppDevVer

1	1	0-32	1	0-32
Unused	AppNumInClusters	AppInClusterList	AppNumOutClusters	AppOutClusterList

EndPoint – 1 byte – Specifies the endpoint of the device.

AppProfId – 2 bytes – Specifies the profile id of the application.

AppDeviceId – 2 bytes – Specifies the device description id for this endpoint.

AppDevVer – 1 byte – Specifies the device version number.

Unused – 1 byte – Set this field to zero.

AppNumInClusters – 1 byte – Specifies the number of Input cluster Ids following in the AppInClusterList

AppInClusterList – 0-32 bytes – Specifies the list of Input Cluster Ids.

AppNumOutClusters – 1 byte – Specifies the number of Output cluster Ids following in the AppOutClusterList

AppOutClusterList – 0-32 bytes – Specifies the list of Output Cluster Ids.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x00	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.4.2 AF_DATA_REQUEST

6.4.2.1 Description

This command will cause the CCZACC06 device to send an application data packet to a remote device.

6.4.2.2 Usage

SREQ:

1	1	1	2	1	1
Length = 0x0A-0x5E	Cmd0 = 0x24	Cmd1 = 0x01	DstAddr	DestEndpoint	SrcEndpoint

2	1	1	1	1	0-128
ClusterID	TransID	Options	Radius	Len	Data

DstAddr – 2 bytes – Specifies the destination address of the message

DestEndpoint – 1 byte – Specifies the destination endpoint of the message.

SrcEndpoint – 1 byte – Specifies the source endpoint of the message.

ClusterID – 2 bytes – Specifies the cluster ID.

TransID – 1 byte – Specifies the transaction sequence number of the message.

Options – 1 byte – Transmit options.

Radius – 1 byte – Specifies the number of hops allowed delivering the message; usually using 7.

Len – 1 byte – Length of the data.

Data – 0-128 bytes – The data.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x01	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.4.3 AF_DATA_CONFIRM

6.4.3.1 Description

This command is issued by the CCZACC06 device to return the results from a AF_SEND_DATA_REQUEST command.

6.4.3.2 Usage

AREQ:

1	1	1	1	1	1
Length = 0x03	Cmd0 = 0x44	Cmd1 = 0x80	Status	Endpoint	TransID

Status – 1 byte – See 6.6 for a listing of the status values.

Endpoint – 1 byte – Endpoint of the device

TransId – 1 byte – Specified the transaction sequence number of the message.

6.4.4 AF_INCOMING_MSG

6.4.4.1 Description

This command is issued by the CCZACC06 device when it has received an incoming packet (addressed to one of the registered endpoints) from a remote device.

6.4.4.2 Usage

AREQ:

1	1	1	2	2	2
Length = 0x11-0x65	Cmd0 = 0x44	Cmd1 = 0x81	GroupID	ClusterID	SrcAddr

1	1	1	1	1	4
SrcEndpoint	DestEndpoint	WasBroadcast	LinkQuality	SecurityUse	Timestamp

1	1	0-128
TransSeqNumber	Len	Data

GroupID – 2 bytes – Specifies the group ID of the device.

ClusterID – 2 bytes – Specifies the cluster ID (only the LSB is used in V1.0 networks.)

SrcAddr – 2 bytes – Specifies the ZigBee network address of the source device sending the message.

SrcEndpoint – 1 byte – Specifies the source endpoint of the message.

DestEndpoint – 1 byte – Specifies the destination endpoint of the message.

WasBroadcast – 1 byte – Specifies if the message was a broadcast or not.

LinkQuality – 1 byte – Indicates the link quality measured during reception.

SecurityUse – 1 byte – Specifies if the security is used or not.

TimeStamp – 4 bytes – Specifies the timestamp of the message.

TransSeqNumber – 1 byte – Specifies transaction sequence number of the message.

Len – 1 byte – Specifies the length of the data.

Data – 0-128 bytes – Contains 0 to 128 bytes of data.

6.5 ZDO interface

6.5.1 ZDO_NWK_ADDR_REQ

6.5.1.1 Description

This command will cause the CCZACC06 device to send a “Network Address Request” packet to request a remote device’s short address. This message sends a broadcast message looking for a 16 bit address with a known 64 bit IEEE address. The results of this request are returned by CCZACC06 via a ZDO_NWK_ADDR_RSP command.

6.5.1.2 Usage

SREQ:

1	1	1	8	1	1
Length = 0x0A	Cmd0 = 0x25	Cmd1 = 0x00	IEEEAddress	ReqType	StartIndex

IEEEAddress – 8 bytes – 64 bit IEEE address of the device.

ReqType – 1 byte – Request type:

Value	Description
0	Single device response
1	Extended – include associated devices

StartIndex – 1 byte – Starting index into the list of children. This is used to get more of the list if the list is too large for one message.

SRSP:

1	1	1	1

Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x00	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.2 ZDO_NWK_ADDR_RSP

6.5.2.1 Description

This command is issued by the CCZACC06 to return the results from a `ZDO_NWK_ADDR_REQ`.

6.5.2.2 Usage

AREQ:

1	1	1	1	8	2
Length = 0x0D-0x53	Cmd0 = 0x45	Cmd1 = 0x80	Status	IEEEAddr	NwkAddr

1	1	0-70
StartIndex	NumAssocDev	AssocDevList

Status – 1 byte – See 6.6 for a listing of the status values.

IEEEAddr – 8 bytes – 64 bit IEEE address of source device.

NwkAddr – 2 bytes – Specifies the short network address of responding device.

StartIndex – 1 byte – Specifies the starting index into the list of associated devices for this report.

NumAssocDev – 1 byte – Specifies the number of associated devices

AssocDevList – 0-70 bytes –Contains the list of network address for associated devices. This list can be a partial list if the entire list doesn't fit into a packet. If it is a partial list, the starting index is StartIndex.

6.5.3 ZDO_IEEE_ADDR_REQ

6.5.3.1 Description

This command will cause the CCZACC06 device to issue an "IEEE address request" packet to request a remote device's IEEE 64-bit address. The results of this request are returned by CCZACC06 via a `ZDO_IEEE_ADDR_RSP` command.

6.5.3.2 Usage

SREQ:

1	1	1	2	1	1
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x01	ShortAddr	ReqType	StartIndex

ShortAddr – 2 bytes – Specifies the short address of the device.

ReqType – 1 byte – Request type:

Value	Description
0	Single device response
1	Extended – include associated devices

StartIndex – 1 byte – Starting index into the list of children. This is used to get more of the list if the list is too large for one message.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x01	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.4 ZDO_IEEE_ADDR_RSP

6.5.4.1 Description

This command is issued by the CCZACC06 to return the results from a ZDO_IEEE_ADDR_REQ.

6.5.4.2 Usage

AREQ:

1	1	1	1	8	2
Length = 0x0D-0x53	Cmd0 = 0x45	Cmd1 = 0x81	Status	IEEEAddr	NwkAddr

1	1	0-70
StartIndex	NumAssocDev	AssocDevList

Status – 1 byte – See 6.6 for a listing of the status values.

IEEEAddr – 8 bytes – 64 bit IEEE address of source device.

NwkAddr – 2 bytes – Specifies the short network address of responding device.

StartIndex – 1 byte – Specifies the starting index into the list of associated devices for this report.

NumAssocDev – 1 byte – Specifies the number of associated devices

AssocDevList – 0-70 bytes –Contains the list of network address for associated devices. This list can be a partial list if the entire list doesn't fit into a packet. If it is a partial list, the starting index is StartIndex.

6.5.5 ZDO_NODE_DESC_REQ

6.5.5.1 Description

This command will cause the CCZACC06 device to issue an "Node descriptor request" packet to request a remote device's node descriptor. The results of this request are returned by CCZACC06 via a ZDO_NODE_DESC_RSP command.

6.5.5.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x02	DstAddr	NWKAddrOfInterest

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

NWKAddrOfInterest – 2 – Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x02	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.6 ZDO_NODE_DESC_RSP

6.5.6.1 Description

This command is issued by the CCZACC06 to return the results from a ZDO_NODE_DESC_REQ.

6.5.6.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x0F	Cmd0 = 0x45	Cmd1 = 0x82	SrcAddr	Status	NwkAddr

1	1	1	2
LogicalType/ ComplexDescAvailable/ UserDescAvailable	APSFlags/ FrequencyBand	MACCapabilityFlags	ManufacturerCode

1	2	2
MaxBufferSize	MaxTransferSize	ServerMask

SrcAddr – 2 bytes – the message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddrOfInterest – 2 bytes – Device's short address of this Node descriptor

LogicalType/ ComplexDescriptorAvailable/ UserDescriptorAvailable – 1 byte – defined as follows:

- LogicalType: Bit 0-2:

Description	Value
ZigBee Coordinator	0
ZigBee Router	1
ZigBee End Device	2

- ComplexDescriptorAvailable – Bit 4 – Indicates if complex descriptor is available for the node
- UserDescriptorAvailable – Bit 5 – Indicates if user descriptor is available for the node APSFlags/ NodeFrequencyBand – 1 byte
- APSFlags – Bit 0-4 – Node Flags assigned for APS. For V1.0 all bits are reserved.
- NodeFrequencyBand – Bit 5-7 – Identifies node frequency band capabilities

Capabilities – byte – Capability flags stored for the MAC

ManufacturerCode – 2 bytes – specifies a manufacturer code that is allocated by the ZigBee Alliance, relating to the manufacturer to the device.

MaxBufferSize – 1 byte - Indicates size of maximum NPDU. This field is used as a high level indication for management.

MaxTransferSize – 2 bytes – Indicates maximum size of Transfer up to 0x7fff (This field is reserved in version 1.0 and shall be set to zero).

ServerMask – 2 bytes – Specifies the system server capability. It is defined as follows:

Bit	Assignment
0	Primary Trust Center
1	Backup Trust Center
2	Primary Binding Table Cache
3	Backup Binding Table Cache
4	Primary Discovery Cache
5	Backup Discovery Cache
6– 15	Reserved

6.5.7 ZDO_SIMPLE_DESC_REQ

6.5.7.1 Description

This command will cause the CCZACC06 device to issue an “Simple descriptor request” packet to request a remote device’s endpoint simple descriptor. The results of this request are returned by CCZACC06 via a `ZDO_SIMPLE_DESC_RSP` command.

6.5.7.2 Usage

SREQ:

1	1	1	2	2	1
Length = 0x05	Cmd0 = 0x25	Cmd1 = 0x04	DstAddr	NWKAddrOfInterest	Endpoint

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

NWKAddrOfInterest – 2 – Specifies NWK address of the destination device being queried.

Endpoint – 1 byte – Specifies the application endpoint the data is from.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x04	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.8 ZDO_SIMPLE_DESC_RSP

6.5.8.1 Description

This command is issued by the CCZACC06 device to return the results from a `ZDO_SIMPLE_DESC_REQ.`

6.5.8.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-4E	Cmd0 = 0x45	Cmd1 = 0x84	SrcAddr	Status	NwkAddr	Len

1	2	2	1
Endpoint	ProfileID	DeviceID	DeviceVersion

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

SrcAddr – 2 bytes – Specifies the message’s source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddr – 2 bytes – Specifies Device’s short address that this response describes.

Len – 1 byte – Specifies the length of the simple descriptor

Endpoint – 1 byte –

ProfileId – 2 bytes – The profile ID for this endpoint.

DeviceID – 2 bytes – The Device Description ID for this endpoint.

DeviceVersion – 1 byte – Defined as the following format

Description	Value

Version 1.00	0
Reserved	0x1 – 0xf

NumInClusters – 1 byte – The number of input clusters in the InClusterList.

InClusterList – 0-32 bytes – List of input cluster IDs supported.

NumOutClusters – 1 byte – The number of output clusters in the OutClusterList.

OutClusterList – 0-32 bytes – List of output cluster IDs supported.

6.5.9 ZDO_ACTIVE_EP_REQ

6.5.9.1 Description

This command will cause the CCZACC06 device to issue an "Active endpoint request" packet to request the list of active endpoints on a remote device. The results of this request are returned by CCZACC06 via a `ZDO_ACTIVE_EP_RSP` command.

6.5.9.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x05	DstAddr	NWKAddrOfInterest

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

NWKAddrOfInterest – 2 – Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x05	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.10 ZDO_ACTIVE_EP_RSP

6.5.10.1 Description

This command is issued by the CCZACC06 device to return the results from a `ZDO_ACTIVE_EP_REQ.`

6.5.10.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-0x53	Cmd0 = 0x45	Cmd1 = 0x85	SrcAddr	Status	NwkAddr	ActiveEPCount

0-77
ActiveEPList

SrcAddr – 2 bytes – the message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddr – 2 bytes – Device's short address that this response describes.

ActiveEPCount – 1 byte – Number of active endpoint in the list

ActiveEPList – 0-77 bytes – Array of active endpoints on this device.

6.5.11 ZDO_MATCH_DESC_REQ

6.5.11.1 Description

This command will cause the CCZACC06 device to issue an “Match descriptor request” packet to find remote device’s that match a given list of clusters. The results of this request are returned by CCZACC06 via a ZDO_MATCH_DESC_RSP command.

6.5.11.2 Usage

SREQ:

1	1	1	2	2	2
Length = 0x08-0x48	Cmd0 = 0x25	Cmd1 = 0x06	DstAddr	NwkAddrOfInterest	ProfileID

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

NWKAddrOfInterest – 2 bytes – Specifies NWK address of the destination device being queried.

ProfileID – 2 bytes – Specifies

NumInClusters – 1 byte – Specifies the number of IDs in the InClusterList.

InClusterList – 0-32 bytes – Contains the input cluster IDs.

NumOutClusters – 1 byte – Specifies the number of IDs in the OutClusterList.

OutClusterList – 0-32 bytes – Contains the output cluster IDs.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x06	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.12 ZDO_MATCH_DESC_RSP

6.5.12.1 Description

This command is issued by the CCZACC06 device to return the results from a ZDO_MATCH_DESC_REQ.

6.5.12.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-0x53	Cmd0 = 0x45	Cmd1 = 0x86	SrcAddr	Status	NwkAddr	MatchLength

0-77
MatchList

SrcAddr – 16 bit – the message’s source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddr – 2 bytes – Device’s short address that this response describes.

MatchCount – 1 byte – Number of descriptors in the list.

MatchList – 0-77 bytes – Array of match descriptor on this device.

6.5.13 ZDO_MATCH_DESC_RSP_SENT

6.5.13.1 Description

This command is issued by the CCZACC06 device if it has responded to a "Match descriptor request" packet from a remote device.

6.5.13.2 Usage

AREQ:

1	1	1	2
Length = 0x04-0x24	Cmd0 = 0x45	Cmd1 = 0xC2	NwkAddr

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

NwkAddr – 2 bytes – Specifies the device's short address

NumInClusters – 1 byte – The number of input clusters in the InClusterList.

InClusterList – 0-32 bytes – List of input cluster IDs supported.

NumOutClusters – 1 byte – The number of output clusters in the OutClusterList.

OutClusterList – 0-32 bytes – List of output cluster IDs supported.

6.5.14 ZDO_USER_DESC_REQ

6.5.14.1 Description

This command will cause the CCZACC06 device to issue an "User descriptor request" packet to request a remote device's user descriptor. The results of this request are returned by CCZACC06 via a ZDO_USER_DESC_RSP command.

6.5.14.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x08	DstAddr	NWKAddrOfInterest

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

NWKAddrOfInterest – 2 bytes – Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x08	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.15 ZDO_USER_DESC_RSP

6.5.15.1 Description

This command is issued by the CCZACC06 device to return the results from a ZDO_USER_DESC_REQ.

6.5.15.2 Usage

AREQ:

1	1	1	2	1	2	1	0-77
Length = 0x06-0x16	Cmd0 = 0x45	Cmd1 = 0x88	SrcAddr	Status	NwkAddr	Len	UserDescriptor

SrcAddr – 2 bytes – the message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddr – 2 bytes – Device's short address that this response describes.

Len – 1 byte – Length, in bytes, of the user descriptor

UserDescriptor – 0-77 bytes – User descriptor array (can be up to 77 bytes).

6.5.16 ZDO_USER_DESC_SET

6.5.16.1 Description

This command will cause the CCZACC06 device to issue an "User descriptor set" packet to set a remote device's user descriptor. The results of this request are returned by CCZACC06 via a ZDO_USER_DESC_CONF command.

6.5.16.2 Usage

SREQ:

1	1	1	2	2	1	0-16
Length = 0x05-0x15	Cmd0 = 0x25	Cmd1 = 0x0B	DstAddr	NWKAddrOfInterest	Len	UserDescriptor

DstAddr – 2 bytes – Specifies network address of the device generating the set request.

NWKAddrOfInterest – 2 bytes – Specifies NWK address of the device targeted for the set request.

Len – 1 byte – Specifies the length of the user descriptor.

UserDescriptor – 0-16 bytes – User descriptor array (can be up to 16 bytes).

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x0B	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.17 ZDO_USER_DESC_CONF

6.5.17.1 Description

This command is issued by the CCZACC06 device to return the results from a ZDO_USER_DESC_SET.

6.5.17.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x05	Cmd0 = 0x45	Cmd1 = 0x89	SrcAddr	Status	NwkAddr

SrcAddr – 2 bytes – The message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

NWKAddr – 2 bytes – Device's short address that this response describes.

6.5.18 ZDO_END_DEVICE_ANNCE

6.5.18.1 Description

This command will cause the CCZACC06 device to issue an "End device announce" broadcast packet to the network. This is typically used by an end-device to announce itself to the network.

6.5.18.2 Usage

SREQ:

1	1	1	2	8	1
Length = 0x0B	Cmd0 = 0x25	Cmd1 = 0x0A	NwkAddr	IEEEAddr	Capabilites

NwkAddr – 2 bytes – Specifies network address of the device generating the request.

IEEEAddr – 8 bytes – Specifies the 64 bit IEEE Address of the device being announced

Capabilites – 1 byte – Specifies MAC capabilities

Bit	Description
0	Alternate PAN Coordinator
1	Device type: 1 – ZigBee Router 0 – End Device
2	Power Source: 1 – Mains powered
3	Receiver on when idle
4-5	Reserved
6	Security capability
7	Reserved

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x0A	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.19 ZDO_END_DEVICE_ANNCE_IND

6.5.19.1 Description

This command is issued by the CCZACC06 device when it has received an “End device announce” packet from a remote device.

6.5.19.2 Usage

AREQ:

1	1	1	2	2	8	1
Length = 0x0D	Cmd0 = 0x45	Cmd1 = 0xC1	SrcAddr	NwkAddr	IEEEAddr	Capabilites

SrcAddr – 2 bytes – Source address of the message.

NwkAddr – 2 bytes – Specifies the device’s short address.

IEEEAddr – 8 bytes – Specifies the 64 bit IEEE address of source device.

Capabilites – 1 byte – Specifies the MAC capabilities of the device.

Bit	Description
0	Alternate PAN Coordinator
1	Device type: 1 – ZigBee Router 0 – End Device
2	Power Source: 1 – Mains powered
3	Receiver on when idle

4-5	Reserved
6	Security capability
7	Reserved

6.5.20 ZDO_END_DEVICE_BIND_REQ

6.5.20.1 Description

This command is generated to request an End Device Bind with the destination device.

6.5.20.2 Usage

SREQ:

1	1	1	2	2	1
Length = 0x09-0x49	Cmd0 = 0x25	Cmd1 = 0x20	DstAddr	LocalCoordinator	Endpoint

2	1	0-32	1	0-32
ProfileID	NumInClusters	InClusterList	NumOutClusters	OutClusterList

DstAddr – 2 bytes – Specifies NWK address of the device generating the inquiry.

LocalCoordinator – 2 bytes – Specifies local coordinator's short address. In the case of source binding, it's the short address of the source address

Endpoint – 1 byte – Device's endpoint.

ProfileID – 2 bytes – Specifies the profile ID of the device.

NumInClusters – 1 byte – Specifies the number of IDs in the InClusterList.

InClusterList – 0-32 bytes – Contains the input cluster IDs.

NumOutClusters – 1 byte – Specifies the number of IDs in the OutClusterList.

OutClusterList – 0-32 bytes – Contains the output cluster IDs.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x20	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.21 ZDO_END_DEVICE_BIND_RSP

6.5.21.1 Description

This callback message is in response to the ZDO End Device Bind Request

6.5.21.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA0	SrcAddr	Status

SrcAddr – 2 bytes – The message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.22 ZDO_BIND_REQ

6.5.22.1 Description

This command is generated to request a Bind.

6.5.22.2 Usage

SREQ:

1	1	1	2	8	1	2
Length = 0x10-0x17	Cmd0 = 0x25	Cmd1 = 0x21	DstAddr	SrcAddress	SrcEndpoint	ClusterID

1	2/8	0/1
DstAddrMode	DstAddress	DstEndpoint

DstAddr – 2 bytes – Specifies the destination address of the device generating the bind request

SrcAddress – 8 bytes – 64 bit Binding source IEEE address

SrcEndpoint – 1 byte – Specifies the binding source endpoint.

ClusterID – 2 byte2 – Specifies the cluster ID to match in messages.

DstAddrMode – 1 byte – Specifies destination address mode: 01 for Group address; 03 for Extended address.

DstAddress – 8 bytes / 2bytes – Binding destination IEEE address. Not to be confused with DstAddr.

DstEndpoint– 1 byte / 0 byte – Specifies the binding destination endpoint. It is used only when DstAddrMode is 64 bits extended address

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x21	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.23 ZDO_BIND_RSP

6.5.23.1 Description

This callback message is in response to the ZDO Bind Request.

6.5.23.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA1	SrcAddr	Status

SrcAddr – 2 bytes – The message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.24 ZDO_UNBIND_REQ

6.5.24.1 Description

This command is generated to request an UnBind

6.5.24.2 Usage

SREQ:

1	1	1	2	8	1	2
Length = 0x10-0x17	Cmd0 = 0x25	Cmd1 = 0x22	DstAddr	SrcAddress	SrcEndpoint	ClusterID

1	2/8	0/1

DstAddrMode	DstAddress	DstEndpoint

DstAddr – 16 bits – Specifies destination address of the device generating the bind request.

SrcAddress – 8 bytes – Specifies 64 bit Binding source IEEE address

SrcEndpoint – 1 byte – Specifies the binding source endpoint.

ClusterID – 2 byte – Specifies cluster ID to match in messages.

DstAddrMode – 1 byte – Specifies destination address mode: 01 for Group address; 03 for Extended address.

DstAddress – 8 bytes – Specifies 64 bit Binding destination IEEE address. Not to be confused with DstAddr.

DstEndpoint – 1 byte – Specifies the binding destination endpoint.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x22	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.25 ZDO_UNBIND_RSP

6.5.25.1 Description

This callback message is in response to the ZDO Unbind Request.

6.5.25.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA2	SrcAddr	Status

SrcAddr – 2 bytes – The message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.26 ZDO_MGMT_LQI_REQ

6.5.26.1 Description

This command is generated to request the destination device to perform a LQI query of other devices in the network.

6.5.26.2 Usage

SREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x25	Cmd1 = 0x31	DstAddr	StartIndex

DstAddr – 2 bytes – Specifies the network address the device generating the query.

StartIndex – 1 byte – Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x31	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.27 ZDO_MGMT_LQI_RSP

6.5.27.1 Description

This callback message is in response to the ZDO Management LQI Request

6.5.27.2 Usage

AREQ:

1	1	1	2	1	1	1
Length = 0x06-0x48	Cmd0 = 0x45	Cmd1 = 0xB1	SrcAddr	Status	NeighborTableEntries	StartIndex

1	0-66
NeighborTableListCount	NeighborTableList Records

SrcAddr – 2 bytes – Source address of the message.

Status – 1 byte – See 6.6 for a listing of the status values.

NeighborTableEntries – 1 byte – Total number of entries available in the device.

StartIndex – 1 byte – Where in the total number of entries this response starts.

NeighborTableListCount – 1 byte – Number of entries in this response.

NeighborTableListRecords – 0-66 bytes – an array of NeighborLqiList items. NeighborLQICount contains the number of items in this table.

Name	Size	Description
ExtendedPanID	8 bytes	PAN ID of the neighbor device
ExtendedAddress	8 bytes	Network extended address
NetworkAddress	2 bytes	Device short address
DeviceType/ RxOnWhenIdle/ Relationship	1 byte	DeviceType: bits 1-0 RxOnWhenIdle: bits 3-2 Relationship: bits 6-4
PermitJoining	1 byte	PermitJoining: bits 1-0
Depth	1 byte	
LQI	1 byte	

6.5.28 ZDO_MGMT_LEAVE_REQ

6.5.28.1 Description

This command is generated to request the destination device to perform a LQI query of other devices in the network.

6.5.28.2 Usage

SREQ:

1	1	1	2	8	1
Length = 0x03	Cmd0 = 0x25	Cmd1 = 0x34	DstAddr	DeviceAddress	Options

DstAddr – 2 bytes – Specifies the network address the device generating the query.

StartIndex – 1 byte – Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x34	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.29 ZDO_MGMT_LEAVE_RSP

6.5.29.1 Description

This callback message is in response to the ZDO Management Leave Request.

6.5.29.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xB4	SrcAddr	Status

SrcAddr – 2 bytes – The message's source network address.

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.30 ZDO_MGMT_PERMIT_JOIN_REQ

6.5.30.1 Description

This command is generated to set the Permit Join for the destination device

6.5.30.2 Usage

SREQ:

1	1	1	2	1	1
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x36	DstAddr	Duration	TCSignificance

DstAddr – 2 bytes – Specifies the network address of the destination device whose Permit Join information is to be modified.

Duration – 1 byte – Specifies the duration to permit joining. 0 = join disabled. 0xff = join enabled. 0x01-0xfe = number of seconds to permit joining.

TC Significance – 1 byte - Trust Center Significance.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x36	Status

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.31 ZDO_MGMT_PERMIT_JOIN_RSP

6.5.31.1 Description

This callback message is in response to the ZDO Management Permit Join Request

6.5.31.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xB6	SrcAddr	Status

SrcAddr – 2 bytes – Source address of the message.

Status – 1 byte – See 6.6 for a listing of the status values.

6.5.32 ZDO_STATE_CHANGE_IND

6.5.32.1 Description

This callback message indicates the ZDO state change.

6.5.32.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x45	Cmd1 = 0xC0	State

State – 1 byte – Specifies the changed ZDO state.

6.6 Return Status Values

The status parameter that is returned from the CCZACC06 device may take one of the following values.

Name	Value
ZSuccess	0x00
ZFailure	0x01
ZInvalidParameter	0x02
ZMemError	0x10
ZBufferFull	0x11
ZUnsupportedMode	0x12
ZUnsupportedMode	0x12
ZMacMemError	0x13
ZSecNoKey	0xa1
ZSecMaxFrmCount	0xa3
zdoInvalidRequestType	0x80
zdoInvalidEndpoint	0x82
zdoUnsupported	0x84
zdoTimeout	0x85
zdoNoMatch	0x86
zdoTableFull	0x87
zdoNoBindEntry	0x88
ZApsFail	0xb1
ZApsTableFull	0xb2
ZApsIllegalRequest	0xb3
ZApsInvalidBinding	0xb4
ZApsUnsupportedAttrib	0xb5
ZApsNotSupported	0xb6
ZApsNoAck	0xb7
ZApsDuplicateEntry	0xb8

ZApsNoBoundDevice	0xb9
zdoInvalidEndpoint	0x82
zdoUnsupported	0x84
zdoTimeout	0x85
zdoNoMatch	0x86
zdoTableFull	0x87
zdoNoBindEntry	0x88
ZApsFail	0xb1
ZApsTableFull	0xb2
ZApsIllegalRequest	0xb3
ZApsInvalidBinding	0xb4
ZApsUnsupportedAttrib	0xb5
ZApsNotSupported	0xb6
ZApsNoAck	0xb7
ZApsDuplicateEntry	0xb8
ZApsNoBoundDevice	0xb9
ZNwkInvalidParam	0xc1
ZNwkInvalidRequest	0xc2
ZNwkNotPermitted	0xc3
ZNwkStartupFailure	0xc4
ZNwkTableFull	0xc7
ZNwkUnknownDevice	0xc8
ZNwkUnsupportedAttribute	0xc9
ZNwkNoNetworks	0xca
ZNwkLeaveUnconfirmed	0xcb
ZNwkNoAck	0xcc
ZNwkNoRoute	0xcd

7 Miscellaneous

7.1 CCZACC06 power-up procedure

The recommended power-up procedure is as follows:

1. Host processor and CCZACC06 power up.
2. Host processor sets CCZACC06 RESET_N pin low, holding CCZACC06 in reset.
3. The host processor sets the optional CCZACC06 CFG0 and CFG1 pins (if these pins are controlled by the host processor).
4. The host processor initializes its UART or SPI interface.
5. The host processor sets CCZACC06 RESET_N pin high and CCZACC06 starts operation.
6. CCZACC06 sends a `SYS_RESET_IND` message to the host processor. When SPI transport is used CCZACC06 will set SRDY low to indicate a message is available and the host processor should retrieve the message.
7. The host processor receives the `SYS_RESET_IND` message.

If the CCZACC06 device was configured as an end-device (and using SPI transport), it will automatically enter low power state after the host processor retrieves the `SYS_RESET_IND` command from the CCZACC06.

CCZACC06 can also be reset when the host processor sends a `SYS_RESET_REQ` message. However, resetting CCZACC06 with the RESET_N pin is recommended because it is faster and more reliable.

7.2 CCZACC06 configuration procedure

After the CCZACC06 power-up, the host processor should configure it according to its application needs. The exact configuration sequence can vary depending on the cause of the reset and which parameters need to be configured on the CCZACC06. Some of the common scenarios are described below.

7.2.1 Scenario 1

The typical configuration sequence when the host processor powers-up for the first time

- Reset CCZACC06 device during host processor power-up sequence
- Receive SYS_RESET_IND
- Configure device type and reset CCZACC06
- Receive SYS_RESET_IND again
- Configure other parameters if necessary
- Set `ZCD_NV_STARTUP_OPTION` to 0x02

7.2.2 Scenario 2

If CCZACC06 device has accidentally reset and the host wishes for it to continue operating as before in the same network (this could also be because the host has reset accidentally and, in its startup sequence, caused CCZACC06 to reset)

- Receive SYS_RESET_IND
- Set `ZCD_NV_STARTUP_OPTION` to 0x00

7.2.3 Scenario 3

If a CCZACC06 device is already operating in a network and the host wishes to clear its state and, possibly reconfigure some parameters, and go into a new network

- Configure device type (if necessary) and reset CCZACC06
- Receive SYS_RESET_IND
- Configure other parameters if necessary
- Set ZCD_NV_STARTUP_OPTION to 0x02

7.2.4 Scenario 4

The host wishes to clear CCZACC06 of all configuration and restore them to default values

- Set ZCD_NV_STARTUP_OPTION to 0x01 and reset CCZACC06
- Receive SYS_RESET_IND

Note that the host processor must be aware of the state of the CCZACC06 device and its configuration so that it can operate properly. It is recommended that this state parameter is itself stored in non volatile memory so that a host processor reset will not cause it to lose information on the CCZACC06 state.

7.3 CCZACC06 ZigBee stack startup procedure

After configuration of the CCZACC06 device, the host processor should register its application and then start the operation of the ZigBee stack.

The host processor can register its application through either the ZB_APP_REGISTER_REQUEST or the AF_REGISTER commands. The former is used to register the application to use the simple api interface. The latter is used if the application would use the full AF and ZDO interface.

After registering the application, the host processor issues the ZB_START_REQUEST command to start the ZigBee stack on the CCZACC06 device.

Once the ZigBee stack has successfully started (indicated by a successful ZB_START_CONFIRM), the CCZACC06 is part of the ZigBee network. The host processor may now issue commands to perform discovery, binding and sending and receiving of packets.

8 General Information

8.1 Document History

Table 1: Document History

Revision	Date	Description/Changes
1.0	2008-03-27	Version of documents for RTM

9 Address Information

Texas Instruments Norway AS
Gaustadalléen 21
N-0349 Oslo
NORWAY
Tel: +47 22 95 85 44
Fax: +47 22 95 85 46
Web site: http://www.ti.com/lpw

10 TI Worldwide Technical Support

Internet

TI Semiconductor Product Information Center Home Page: support.ti.com
TI Semiconductor KnowledgeBase Home Page: support.ti.com/sc/knowledgebase

Product Information Centers

Americas

Phone: +1(972) 644-5580
Fax: +1(972) 927-6377
Internet/Email: support.ti.com/sc/pic/americas.htm

Europe, Middle East and Africa

Phone:
Belgium (English) +32 (0) 27 45 54 32
Finland (English) +358 (0) 9 25173948
France +33 (0) 1 30 70 11 64
Germany +49 (0) 8161 80 33 11
Israel (English) 180 949 0107
Italy 800 79 11 37
Netherlands (English) +31 (0) 546 87 95 45
Russia +7 (0) 95 363 4824
Spain +34 902 35 40 28
Sweden (English) +46 (0) 8587 555 22
United Kingdom +44 (0) 1604 66 33 99
Fax: +49 (0) 8161 80 2045
Internet: support.ti.com/sc/pic/euro.htm

Japan

Fax	International	+81-3-3344-5317
	Domestic	0120-81-0036
Internet/Email	International	support.ti.com/sc/pic/japan.htm
	Domestic	www.tij.co.jp/pic

Asia

Phone	International	+886-2-23786800
	Domestic	Toll-Free Number
	Australia	1-800-999-084
	China	800-820-8682
	Hong Kon	800-96-5941
	India	+91-80-51381665 (Toll)
	Indonesia	001-803-8861-1006
	Korea	080-551-2804
	Malaysia	1-800-80-3973
	New Zealand	0800-446-934
	Philippines	1-800-765-7404
	Singapore	800-886-1028
	Taiwan	0800-006800
	Thailand	001-800-886-0010
Fax		+886-2-2378-6808
Email		tiasia@ti.com or ti-china@ti.com
Internet		support.ti.com/sc/pic/asia.htm

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265