

Telegesis™	 <b>SILICON LABS</b>	TG-PM-0513-CICIE R31X
HA Evaluation Kits (CICIE)		Product Manual (CICIE R31X)

Telegesis™ is a trademark of Silicon Laboratories Inc.

---

# Product Manual

---

## AT Command Set for Combined Interface & Control Indicating Equipment



**Current Firmware CICIE R311**  
**EmberZNet 4.7.1 stack**

Rev: 13  
 October 2015

## Table of Contents

<b>1</b>	<b>GENERAL.....</b>	<b>3</b>
1.1	Revision History .....	3
1.2	Review History .....	5
1.3	Approval History .....	5
1.4	References.....	5
<b>2</b>	<b>OVERVIEW.....</b>	<b>7</b>
2.1	Objective .....	7
2.2	ZigBee Home Automation Profile .....	7
2.2.1	Home Automation Devices .....	7
2.2.2	Combined Interface.....	8
2.2.3	IAS Control and Indicating Equipment (CIE).....	9
2.2.4	On/Off Output.....	9
2.3	Compatibility with other devices .....	10
<b>3</b>	<b>AT STYLE COMMAND CONVENTIONS .....</b>	<b>11</b>
3.1	Parameters .....	12
3.2	Prompt Overview.....	13
3.3	Device Overview .....	21
3.4	Addressing modes .....	21
3.5	AT Command Overview .....	22
3.5.1	Common Command Set.....	26
3.5.2	Combined Interface Specified Command Set.....	79
3.5.3	CIE Specified Command Set.....	120
3.5.4	Tunnel Cluster Commands.....	129
3.5.5	Time Cluster Command .....	132
3.5.6	OTA Communication Commands.....	133
3.5.7	OTA Server Configuration Commands.....	140
3.5.8	Poll Control Cluster Commands .....	143
<b>4</b>	<b>LIST OF STATUS CODES.....</b>	<b>146</b>
<b>5</b>	<b>S-REGISTERS .....</b>	<b>148</b>
5.1	S-Registers for Network Setup .....	149
5.2	S-Registers for Module Setup .....	155
5.3	I/O related S-Registers.....	167
<b>6</b>	<b>INTERPRETING RSSI ENERGY LEVELS.....</b>	<b>168</b>
<b>7</b>	<b>FIRMWARE UPGRADES .....</b>	<b>168</b>
7.1	Firmware Upgrades via Serial Port.....	168
<b>8</b>	<b>APPENDIX A .....</b>	<b>170</b>
<b>9</b>	<b>APPENDIX B .....</b>	<b>173</b>
<b>10</b>	<b>APPENDIX C .....</b>	<b>176</b>
<b>11</b>	<b>APPENDIX D .....</b>	<b>178</b>
D.1	CICIE OTA Server Implementation.....	181
D.2	Use Cases.....	181
D.2.1	Host Application Handle Incoming OTA Client Cluster Commands.....	182
D.2.2	CICIE OTA Plugin Handle Incoming OTA Client Cluster Commands.....	182
D.2.3	Both Host and CICIE OTA Plugin Partially Handle Incoming OTA Cluster Commands....	183

---

D.3 Explanation of Upgrade Policy.....	183
D.3.1 QueryPolicy (It is used when S62 Bit 0 is set to true).....	183
D.3.2 UpgradeRequestPolicy (It is used when S62 Bit 1 is set to true).....	184

# 1 General

## 1.1 Revision History

Version	Rev	Date	Change Essentials	Reason	Author
1	0	26/03/2014	Created up the CICIE R300 manual. Added OTA server Added Time server Added Tunnelling client and server Added poll control client		Yuanlong Liu
	1	28/05/2014	Added Appendix A to list command and prompts for user specified commands and response Modified description of AT+RAWZCL command Added description of use of BIT4 for S0A	Added description for the features required by user	Yuanlong Liu
	2	02/07/2014	Added AT+MATTRDISC to support discovery of manufacturer specified attributes Added AT+INTERPAN command to support sending interpan message to a remote node. Introduced INTERPAN prompt to show received interpan message Zone table is expended to contain 32 entries Added usage of S61 bit 5 to allow enrolment even when local zone table is full. Changed DELZENTRY command		Yuanlong Liu
	3	28/08/2014	Added new command AT+SENDUCASTB	Request by user to input long command	Yuanlong Liu
	4	08/09/2014	Added description for AT+WRITEATR		Yuanlong Liu
		15/10/2014	Corrected typo mistake in AT+PJOIN description		Yuanlong Liu

			Added Appendix C		
	5	05/11/2014 26/01/2015  19/02/2015 24/02/2015 03/03/2015	Corrected section 2.2.2 Added description for EBIND command Fixed firmware issue in AT_EUIREQ Removed prompt MED Added feature to enable printing RSSI LQI while doing panscan Corrected AT+GPGET description Corrected AT+FPSTOP description		Yuanlong Liu
	6	13/03/2015	Added new command AT+CSWITCH to change channel silently Added S48-S4C to enable user configuration for endpoint 0x01		Yuanlong Liu
	7	08/04/2015	Changed default settings for S4B and S4C Child table supports 24 entries		Yuanlong Liu
	8	10/04/2015 13/04/2015	Introduced S4D and S4E to configure MAC polling time out and end device polling time out Introduced command AT+TIMERD (please see command description)		Yuanlong Liu
	9	29/05/2015	Introduced new prompt to handle OTA page request command Introduced AT+DFTRSP command specifically for OTA cluster Added support for handling HA1.2 specified Arm command	Update to comply to latest HA profile specification	Yuanlong Liu
	10	11/06/2015	Introduced two new commands to support IAS ZONE test	Enhance IAS support	Yuanlong Liu
	11	30/06/2015	Fixed Write Attribute command bug		Yuanlong Liu

			<p>Corrected some typo mistake in S48 and S4C description</p> <p>Added Usage for S61 Bit 8 to for handling ACE client commands.</p> <p>New prompt to show incoming ACE client commands have been added in S61 Descriptions</p>		
	12	17/08/2015	Modified description of AT+EN	Review	Yuanlong Liu
	13	05/10/2015	Added usage of bit 2 of S63 for disable automatic Check-in response sent by CICIE	Update	Yuanlong Liu

## 1.2 Review History

Rev	Date	Reviewer	Review Results

## 1.3 Approval History

Version	Date	Approved by	Evidence of Approval

## 1.4 References

This document is developed on the basis of the following:

- [1] ETRX3 Series Product Manuals – The Product Manuals concentrate on the hardware specification of the modules.
- [2] Evaluation Kits User Guides – The Evaluation Kits User Guide contains all of the information required to set up your Evaluation Kits and run firmware upgrades where necessary.
- [3] Application notes from [www.telegesis.com](http://www.telegesis.com)
- [4] ZigBee PRO specification, document 053474r17
- [5] ZigBee ZCL specification, document 075123r03ZB
- [6] ZigBee Home Automation Public Application Profile 1.1 specification Revision 26, document 053520r26
- [7] ZigBee Home Automation Public Application Profile Version 1.2 Revision 29 053520-29

It is highly recommended to refer to the ZigBee Cluster Library and ZigBee Home Automation Profile specification to further understand the various AT Commands and their usage.

**Please note:**

This document inherits contains of the previous revisions of CICIE R300 manuals. New information has been added to introduce Over-The-Air upgrade server cluster support, time server cluster support and tunnelling client/server cluster support.

The R311 version CICIE is developed upon the HA1.2.1 Certified R310 CICIE. It shall have all supported features in R310, which make it interoperable with the software developed for R310.

The revision history table provides brief information of the changes in this document. Further information about the changes of the firmware is provided in the Release notes document, which should be sent to the user with firmware ebl file.

## 2 Overview

### 2.1 Objective

This document describes the Home Automation (HA) AT Command Set firmware to be used with Telegesis ZigBee PRO wireless meshing modules on a HA device that can act as a Combined Interface(CI), Control and Indicating Equipment(CIE) device or On/Off output.

The HA AT Command Set provides a simple AT command interface to implement the CI, CIE or on/off output of the ZigBee HA Application Profile. The AT command set described in this document can be used to evaluate Telegesis implementation of HA applications. By using the CICIE with the Five-In-One device in the evaluation kits, the user can demonstrate a range of HA applications, for example turn on/off a light, dim a light, get current temperature reading and so on. It can also work with other IAS devices to demonstrate and evaluate the application of an IAS system.

The AT-Command interface described in this document also enables quick development of a combined interface, IAS CIE or an on/off output device (for example a switch), which can shorten the R&D period of a Home Automation product significantly. It enables developers to simply design and build ZigBee Home Automation compliant combined interface or on/off output products without ever having to do embedded firmware or RF work.

Telegesis ZigBee modules are based on the EM357 chipset porting the industry leading EmberZNet ZigBee stack. They can be built into any device and provide a low cost, low power ZigBee solution. RF experience or expertise is not required to apply this powerful wireless networking capability to your products. Telegesis ZigBee Modules offer fast integration opportunities and the shortest possible time to market for your product. This document is meant as an AT-Command and S-Register reference for HA R3xx firmware based on EmberZNet4.x.

### 2.2 ZigBee Home Automation Profile

The ZigBee HA Profile is a public application profile which defines device descriptions and standard practices for applications commonly found in a residential or light commercial environment. Installation scenarios range from a single room to an entire home. The key applications included in this profile are lighting, HVAC, window shades and security.

#### 2.2.1 Home Automation Devices

Various types of devices can participate in a Home Area Network (HAN) to enable HA applications and market programs. These devices are categorized in several groups as following:

- **Generic Devices:** On/Off Switch, Level Control Switch, On/Off Output, Level Controllable Output, Scene Selector, Configuration Tool, Remote Control, Combined Interface, Range Extender, Mains Power Outlet, Door Lock, Door Lock Controller, Simple Sensor.
- **Lighting Devices:** On/Off Light, Dimmable Light, Colour Dimmable Light, On/Off Light Switch, Dimmer Switch, Colour Dimmer Switch, Light Sensor, Occupancy Sensor.
- **Closure Devices:** Shade, Shade Controller, Window Covering, Window Covering controller
- **HVAC Devices:** Heating/Cooling Unit, Thermostat, Temperature Sensor, Pump, Pump Controller, Pressure Sensor, Flow Sensor.
- **Intruder Alarm System Devices:** IAS CIE (Control and Indicating Equipment), IAS ACE (Ancillary Control Equipment), IAS Zone, IAS WD (Warning Device).

The Telegesis HA device which is introduced in this document supports three endpoints, 0x01, 0x02 and 0x0A. The end point 0x01 has been configured to a combined interface; the end point 0x02 has



been configured to a IAS CIE device and the end point 0x0A has been configured to an On/Off output device. The following table lists the supported clusters on each end point.

### 2.2.2 Combined Interface

The Combined Interface device is capable of controlling and monitoring other devices. It is typically a mains-powered device such as a personal computer.

Cluster	Cluster ID	Implementation	Description
Basic	0x0000	Server & Client	Provides basic information about a device
Power configuration	0x0001	Client	For determining detailed information about a device's power source(s).
Device temperature configuration	0x0002	Client	For determining information about a device's internal temperature
Identify	0x0003	Server & Client	To put a device into an Identification mode
Group	0x0004	Client	Provides the capability for group addressing
Scenes	0x0005	Client	For setting up and recalling scenes
On/off	0x0006	Client	Attributes and commands for switching devices between 'On' and 'Off' states
On/Off switch configuration	0x0007	Client	Attributes and commands for configuring On/Off switching devices
Level control	0x0008	Client	Provides an interface for controlling a characteristic of a device that can be set to a level
Time	0x000A	Server	Provides a basic interface to a real-time clock.
OTA	0x0019	Server	Provides an interoperable mean for devices from different manufacturers to upgrade each other's image
Poll Control	0x0020	Client	Provides a mechanism for management of an end device's MAC data Request rate.
Shade configuration	0x0100	Client	Provides an interface for reading information about a shade, and configuring its open and closed limits
Door lock	0x0101	Client	Provides an interface to a generic way to secure a door
Window covering	0x0102	Client	Provides an interface for controlling and adjusting automatic window coverings
Pump configuration and control	0x0200	Client	Provides an interface for the setup and control of pump devices
Thermostat	0x0201	Client	Provides an interface to the functionality of a thermostat
Thermostat user interface configuration	0x0204	Client	Provides an interface to allow configuration of the user interface for a thermostat
Colour control	0x0300	Client	Provides an interface for changing the colour of a light
Illuminance measurement	0x0400	Client	Provides an interface to illuminance measurement functionality

Illuminance level sensing	0x0401	Client	Provides an interface to illuminance level sensing functionality
Temperature measurement	0x0402	Client	Provides an interface to temperature measurement functionality
Simple Metering	0x0702	Client	Provides a mechanism to retrieve usage information from Electric, Gas, Water metering devices.

**Table 1: Clusters Supported by Combined Interface**

### 2.2.3 IAS Control and Indicating Equipment (CIE)

The IAS CIE device is the central Control and Indicating Equipment for an Intruder Alarm System. It receives inputs from sensors (Zone Devices) and control equipment (ACE), and sends output to a warning device (WD).

Cluster	Cluster ID	Implementation	Description
Basic	0x0000	Server	Provides basic information about a device
Identify	0x0003	Server/Client	To put a device into an Identification mode
IAS Zone	0x0500	Client	Defines an interface to the functionality of an IAS security zone device
IAS ACE	0x0501	Server	Defines an interface to the functionality of any Ancillary Control Equipment of the IAS system
IAS WD	0x0502	Client	Provides an interface to the functionality of any Warning Device equipment of the IAS system

**Table 2: Clusters Supported by Combined Interface**

### 2.2.4 On/Off Output

The On/Off Output device is capable of being switched on and off. This device should only be used when a more specific device specification (for example, a Basic Light) is not available.

Cluster	Cluster ID	Implementation	Description
Basic	0x0000	Server	Provides basic information about a device
Power configuration	0x0001	Server	For determining detailed information about a device's power source(s).
Device temperature configuration	0x0002	Server	For determining information about a device's internal temperature
Identify	0x0003	Server	To put a device into an Identification mode
Group	0x0004	Server	Provides the capability for group addressing
Scenes	0x0005	Server	For setting up and recalling scenes
On/off	0x0006	Server	Attributes and commands for switching devices between 'On' and 'Off' states
Generic Tunnel	0x0600	Server/Client	This is the minimum requirement for use tunnelling.
Tunnelling	0x0704	Server/Client	Provides an interface for tunnelling protocols.

**Table 3: Clusters Supported by On/Off Output**

In addition, the firmware supports 16 entries of flash maintained Zone table, 16 entries of RAM maintained Zone table 22 entries of binding table, 3 entries of reporting table, 8 entries of address table and 24 entries of child table. It also supports 3 scene entries to be used with OnOff server on endpoint 0x0A.

Please note: The CICIE device is not designated as a primary cache device which is capable of buffering messages for sleepy end device with slow polling rate. Though it supports 26 seconds buffering period before discard the message, it is recommended to check sleepy end device's polling settings to work with the CICIE.

It also worth to mention that the CI endpoint and CIE endpoint are HA 1.2.1 certified. The endpoint 0x0A (OnOff output) is not tested as this endpoint is introduced for demonstration purpose. So it contains some customized features, which may affect compliance test. If the user intends to develop a certified OnOff output (e.g. light) product, it is highly recommended to contact Telegesis for solution.

## **2.3 Compatibility with other devices**

The CICIE R3xx AT-Command line Interpreter is based on the ZigBee Home Automation specification [6], using the ZigBee PRO feature set [4]. Interoperability with other devices which implement the Home Automation Application Profile and use the ZigBee PRO feature set is supported (Note: those devices should support corresponding clusters which have been listed in Table 1, 2, 3 and 4).

However, CICIE R3xx is not compatible with the devices that do not implement the Home Automation Profile. Nor is it compatible with earlier versions of ZigBee.

### 3 AT Style Command Conventions

To simplify the communication with the Telegesis modules, an AT-style command set, similar to the industry standard Hayes modem control language, is used.

Each command must be preceded by an “AT” or “at” prefix. To terminate a command enter <CR>. Any data not following this pattern is either not accepted by the module or will cause an error message in response.

Commands are followed by an optional response that includes <CR><LF><Response><CR><LF> and/or a prompt <CR><LF><Prompt><CR><LF> where the prompt could also be an error message.

Example:

```
ATS00? <CR>
<CR><LF>FFFF<CR><LF>
<CR><LF>OK<CR><LF>
```

It is recommended to wait for an “OK” or “ERROR: XX” prompt before issuing the next command.

Any data which is prompted to the user is delivered in the format <CR><LF><prompt><CR><LF>.

Example:

```
<CR><LF><JPAN:11,1789,37BF1CD42CC5E673><CR><LF>
```

A prompt intersecting a command being entered will not affect the command itself.

Throughout this document, only the responses and prompts are presented, <CR><LF> are omitted intentionally. Sequences of AT commands in a single line are not supported.

Telegesis ETRX357 features a 256 byte FIFO buffer for incoming radio messages, which allow rapid reception of multiple messages without loss of characters.

Read Command <b>ATXXX?</b>	Commands ending with a ‘?’ return the currently set value of the parameter or parameters
Write Command <b>ATXXX=&lt;...&gt;</b>	This command sets user-definable parameters as indicated by the ‘=’ sign.
Execute Command <b>ATXXX</b>	This command executes routines of the module and returns parameters

**Table 4: Types of AT commands**

### 3.1 Parameters

Each parameter must be entered in the correct format for any of the AT commands to execute correctly. Optional parameters are marked with square brackets [...]. Some general abbreviations are as followings:

<b>XX</b>	8-bit hexadecimal number. Valid characters are 0-9, a-f and A-F
<b>XXXX</b>	16-bit hexadecimal number. Valid characters are 0-9, a-f and A-F
<b>n</b>	Number from 0-9
<b>s</b>	Sign
<b>b</b>	Bit (0 or 1)
<b>c</b>	character
<b>&lt;PANID&gt;</b>	16-bit hexadecimal PAN ID (0000 to FFFF)
<b>&lt;EPANID&gt;</b>	64-bit hexadecimal extended PAN ID
<b>&lt;channel&gt;</b>	decimal channel (802.15.4 channel 11-26)
<b>&lt;password&gt;</b>	8 character password
<b>&lt;EUI64&gt;</b>	64-bit IEEE 802.15.4 address in hexadecimal

Table 5: Different formats of parameters

## 3.2 Prompt Overview

The following prompts can show up during the operation of the AT Command set for a CICIE device.

Prompt Overview	
<b>WRONG HARDWARE</b>	Please check if the module is Telegesis ETRX3xx
<b>OK</b>	OK terminator
<b>ERROR:XX</b>	Error number XX occurred
<b>ACK:XX</b>	Acknowledgement for message XX was received
<b>NACK:XX</b>	Acknowledgement for msg XX was not received
<b>LeftPAN</b>	Local Node has left the PAN
<b>LostPAN</b>	Local Node (not COO) has lost its parent node
<b>FFD:&lt;IEEE Address&gt;,&lt;NodeID&gt;</b>	A router joins CICIE's network
<b>SED:&lt;IEEE Address&gt;,&lt;NodeID&gt;</b>	A sleep end device joins CICIE's network
<b>ZED:&lt;IEEE Address&gt;,&lt;NodeID&gt;</b>	A ZigBee end device joins CICIE's network
<b>JPAN:&lt;channel&gt;,&lt;PANID&gt;,&lt;EPANID&gt;</b>	Local Node has joined PAN with given parameters
<b>Bind:&lt;NodeID&gt;,&lt;Status&gt;</b>	Bind to the target device
<b>Unbind: &lt;NodeID&gt;,&lt;Status&gt;</b>	Unbind the target device
<b>ATTR:&lt;Attribute Value&gt;</b>	See +GETATR command description
<b>RESPATTR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,&lt;Status&gt;,&lt;AttrInfo&gt;</b>	Please see +READATR command description
<b>RESPMATR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ManufCode&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,&lt;Status&gt;,&lt;AttrInfo&gt;</b>	Please see +READMATR command description
<b>WRITEATTR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,&lt;Status&gt;</b>	Please see +WRITEATR command description
<b>WRITEMATR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ManufCode&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,&lt;Status&gt;</b>	Please see +WRITEMATR command description
<b>REPORTATTR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ClusterID&gt;,&lt;AttributeID&gt;,&lt;DataType&gt;,&lt;AttributeValue&gt;</b>	This prompt is shown when the CICIE receives <i>ReportAttribute</i> command
<b>REPORTMATR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ManufacturerCode&gt;,&lt;ClusterID&gt;,&lt;AttributeID&gt;,&lt;DataType&gt;,&lt;AttributeValue&gt;</b>	This prompt is shown when the CICIE receives <i>ReportAttribute</i> command for manufacturer specified attribute
<b>DISCATTR:&lt;NodeID&gt;,&lt;EndPoint&gt;</b>	See +ATTRDISC description for information
<b>ENDDISCATTR</b>	This prompt shows at the end of attribute discovery response
<b>DISCMATR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ManufacturerCode&gt;</b>	See +MATTRDISC description for information
<b>ENDDISCMATR</b>	This prompt shows at the end of manufacturer specified attribute discovery response
<b>DFTREP:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;Cluster&gt;,&lt;CMDID&gt;,&lt;Status&gt;</b>	Received default response
<b>DEV:&lt;Node ID&gt;,&lt;EndPoint&gt;</b>	ZigBee Service Discovery information. See +DISCOVER for more information
<b>IDENTIFY:&lt;DestEP&gt;,&lt;SourceNodeID&gt;,&lt;SourceEP&gt;,&lt;TimeOut&gt;</b>	Receive an Identify command
<b>On:&lt;Source NodeID&gt;,&lt;Source EP&gt;</b>	Receive an On command
<b>Off:&lt;Source NodeID&gt;,&lt;Source EP&gt;</b>	Receive an Off command

<b>Toggle:</b> <Source NodeID>,<Source EP>	Receive a Toggle command
<b>RX:</b> <NodeID>,<profileID>,<destination Endpoint>,<SourceEndpoint>,<ClusterID>,<length>:<payload>	Need to set S0F bit 9 to enable this prompt(please check section 5.2 for information)
<b>CMD:</b> <NodeID>,<CMDID> <b>PAYLOAD:</b> <Raw Hex data>	This prompt is used to show received ZDO commands which are not handled with a specific prompt for example: MatchDesc for match descriptor response (other ZDO response can be found in section 3.5.1)
<b>EBINDREQ:</b> <NodeID>,<EUI>,<Endpoint>,<ProfileID> <b>INPUTCLUSTER:</b> <ClusterID>... <b>OUTPUTCLUSTER:</b> <ClusterID> ...	This prompt is used to show that an End device binding request is received by the CICIE device. The CICIE can handle end device binding request and set up binding for two device endpoint. If the user want to do end device binding for the OnOff output on CICIE firmware with a switch, end device bind request should be sent from the switch to CICIE, then AT+EBIND:0A can be used after <b>EBINDREQ</b> prompt is received.
<b>EBINDRSP:</b> <Status>	This prompt is used to show that an End device binding response is received by the CICIE device.
<b>ZENROLLREQ:</b> <NodeID>,<EndPoint>,<ZoneType>,<ManufactureCode>	This prompt shows when CICIE received a enrol request command. CICIE will automatically handle the request and send a response back if local zone table is not full.
<b>ENROLLED:</b> <ZID>,<ZoneType>,<EUI>	This prompt shows when a zone device enrolls successfully.
<b>ZONEENTRY:</b> <ZID>,<ZoneType>,<EUI>	Show zone table entry information. Please see AT+ZENTRY or AT+ZTABLE
<b>ZMAPSECTION</b> <Index>:<Map>	Show local zone map sections. Please see command AT+ PRINTZMAP
<b>GETZONEINFO:</b> <NodeID>,<EndPoint>,<ZoneID>,<SequenceNum>	Receive a getZoneInformation command (Note: this prompt will only show when the user set S61 bit 6 zone ID is larger than 31).
<b>NOTFOUNDZONE</b>	This prompt will show if the user delete a zone entry which is not existing.
<b>MZONETYPE</b> <index>:<ZoneType>	Show user specified zone type.
<b>ARM:</b> <NodeID>,<EP>,<ARM Mode>,<Arm/Disarm code>,<ZoneID>]	Receive an <i>ARM</i> command. Please note: if the ARM command is sent from a HA1.2 device, <Arm/Disarm code>,<ZoneID> will show.
<b>BYPASS:</b> <NodeID>,<EP>,<Number of Zone>,<ZoneID>...,<Arm/Disarm code>	Receive a <i>Bypass</i> command.
<b>EMERGENCY:</b> <NodeID>,<EP>	Receive an <i>Emergency</i> command.
<b>FIRE:</b> <NodeID>,<EP>	Receive a <i>Fire</i> command.
<b>PANIC:</b> <NodeID>,<EP>	Receive a <i>Panic</i> command.
<b>ZONESTATUS:</b> <NodeID>,<EP>,<ZoneStatus>,<ExtendStatus>[,<ZoneID>,<Delay>]	Receive a <i>Zone Status Change Notification</i> command. <ZoneStatus> - 16 bit hex (Please check ZigBee Cluster Library ZoneStatus attribute section for more information)



	<p>&lt;ExtendStatus&gt; - 8 bit hex, this parameter is reserved for additional status</p> <p>If this notification is sent by a HA1.2 device the following information will show:</p> <p>&lt;ZoneID&gt; - 8 bit hex, zone ID</p> <p>&lt;Delay&gt; - 16 bit hex, the amount of time, in quarter seconds, from moment when a change takes place in one or more bits of the <i>ZoneStatus</i> and the successful transmission of the Zone Status Change Notification. This is designed to help congested networks or offline servers quantify the amount of time from when an event was detected and when it could be reported to the client</p>
ALARM:<NodeID>,<EP>,<AlarmCode>,<ClusterID>	The CICIE does not support Alarm cluster. This prompt is used to show that the CICIE receives an Alarm command from a remote device.
TNLOPEN:<TunnelID>,<Status>	Indicate a tunnel has been opened.
TNLDATA:<TunnelID>,<Length>,<Data>	Receive tunnel data with length.
TNLError:<TunnelID>,<TunnelError>	Receive a <i>TransferDataError</i> command after sending tunnel data to the target.
TNLCLOCE:<TunnelID>	Indicate a tunnel has been closed.
TNLC00:<EUI>,<ClientEP>,<SourceEP>,<TunnelID>	Local client tunnel information.
TNLS00:<EUI>,<ClientEP>,<SourceEP>,<TunnelID>	Local server tunnel information.
TNLOPEN:<TunnelID>,<Status>	Indicate a tunnel has been opened.
INTERPAN:<ProfileID>,<ClusterID>,<Msgtype>,<Option>,[GroupID],<PanID>,<SrcAddr>,<MsgLength>,<Msg>	<p>This prompt will show when the CICIE receives an interpan message.</p> <p>&lt;ProfileID&gt; - 16 bit hex</p> <p>&lt;ClusterID&gt; - 16 bit hex</p> <p>&lt;Msgtype&gt; - 8 bit hex, 0x00 – Unicast; 0x08 – Broadcast; 0x0C- Multicast</p> <p>&lt;Option&gt; - 16 bit hex, if it is 0x0002, the &lt;SrcAddr&gt; will be source long address, and otherwise it is source node ID</p> <p>[GroupID] – 16 bit hex, if the message is send to a group, this parameter will show.</p> <p>&lt;PanID&gt; - 16 bit hex, source PANID</p> <p>&lt;SrcAddr&gt; - 16 bit hex source node ID or EUI</p> <p>&lt;MsgLength&gt; - 8 bit hex, message length</p> <p>&lt;Msg&gt; - received message in hex format</p>
GETPANELSTATUS:<NodeID>,<EP>,<Seq>	This prompt will show when the CICIE receives a GetPanelStatus command. The user shall construct a response and send back with the same sequence number.
GETBYPASSZLIST:<NodeID>,<EP>,<Seq>	This prompt will show when the CICIE receives a GetBypassZoneList command. The user shall



	construct a response and send back with the same sequence number.
<b>GETZSTATUS:&lt;NodeID&gt;,&lt;EP&gt;,&lt;startingZoneId&gt;,&lt;maxNumberOfZoneIds&gt;,&lt;zoneStatusMaskFlag&gt;,&lt;zoneStatusMask&gt;,&lt;Sequence&gt;</b>	This prompt will show when the CICIE receives a GetZoneStatus command. The user shall construct a response and send back with the same sequence number.
<b>CHECKIN:&lt;NodeID&gt;,&lt;EP&gt;[,Seq]</b>	<p>This prompt indicate that a Check in command is received by CICIE. If S63 bit 2 is set to TRUE, the sequence number parameter will show. The host shall construct and send a Check-in response using this sequence number.</p> <p>By default, S63 will not be set, CICIE will automatically respond. In that case, the host shall use AT+FPSET to set the parameter for the check-in response.</p>
<b>The following prompts can show up during the OTA upgrade operation of the AT Command set for a CICIE device.</b>	
<b>IMGQUERY:&lt;NodeID&gt;,&lt;EP&gt;,&lt;FiledControl&gt;,&lt;ManufCode&gt;,&lt;ImgType&gt;,&lt;CurrentFileVer&gt;[,&lt;HardwareVer&gt;],&lt;SequenceNumber&gt;</b>	<p>This prompt is used to notify the Host application of received <i>QueryNextImageRequest</i> command.</p> <p>&lt;NodeID&gt; - 16 bit Hex. The source network address.</p> <p>&lt;EP&gt; - 8 bit Hex. The source endpoint.</p> <p>&lt;FieldControl&gt; - 8 bit Hex indicating whether hardware version parameter present in this command or not.</p> <p>&lt;ManufCode&gt; - 16 bit Hex. The value shall be the device's assigned manufacturer code.</p> <p>&lt;ImgType&gt; - 16 bit Hex, image type</p> <p>&lt;CurrentFileVer&gt; - 32 bit Hex, represents the requesting device's current running image version</p> <p>&lt;HardwareVer&gt; - 16 bit Hex, if the &lt;FieldControl&gt; is 1, this parameter is expected. It represents the requesting device's current running hardware.</p> <p>&lt;SequenceNumber&gt; - 8 bit Hex which is the sequence number of the received request command. The user need to use this sequence number in response command (AT+QIMGRSP).</p> <p>Note: Upon receiving this prompt, the Host application shall use a AT+QIMGRSP command as response in 250 ms.</p>
<b>IMGBREQ:&lt;NodeID&gt;,&lt;EP&gt;,&lt;FiledControl&gt;,&lt;ManufCode&gt;,&lt;ImgType&gt;,&lt;FileVer&gt;,&lt;Offset&gt;,&lt;MaxDataSize&gt;[,&lt;RequestNodeAddress&gt;],</b>	<p>This prompt is used to notify the Host application of received <i>ImageBlockRequest</i> command.</p> <p>&lt;NodeID&gt; - 16 bit Hex. The source nodeID.</p>

<p><b>&lt;BlockRequestDelay&gt;],&lt;SequenceNumber&gt;</b></p>	<p><b>&lt;EP&gt;</b> - 8 bit Hex. The source endpoint.</p> <p><b>&lt;FieldControl&gt;</b> - 8 bit Hex. It is used to indicate additional optional fields that may be included in the payload of Image Block Request command.</p> <p>Field control value 0x00 (bit 0 not set) indicates that the client is requesting a generic OTA upgrade file.</p> <p>Field control value of 0x01 (bit 0 set) means that the client's IEEE address is included in the payload.</p> <p>If bit 1 is set, the BlockRequestDelay field presents in this prompt.</p> <p><b>&lt;ManufCode&gt;</b> - 16 bit Hex. The value shall be the device's assigned manufacturer code.</p> <p><b>&lt;ImgType&gt;</b> - 16 bit Hex, image type.</p> <p><b>&lt;FileVer&gt;</b> - 32 bit hex, representing the OTA image file version that is being requested.</p> <p><b>&lt;Offset&gt;</b> - 32 bit Hex, File offset, its value indicates number of bytes of data offset from the beginning of the file. It essentially points to the location in the OTA image file that the client is requesting the data from. The value reflects the amount of (OTA upgrade image file) data (in bytes) that the client has received so far.</p> <p><b>&lt;MaxDataSize&gt;</b> - 8 bit Hex, the value indicates the largest possible length of data (in bytes) that the client can receive at once. The server shall respect the value and not send the data that is larger than the maximum data size.</p> <p>If the corresponding bit is set, then the corresponding fields are expected:</p> <p><b>&lt;RequestNodeAddress&gt;</b> - This is the IEEE address of the client device sending the Image Block Request command.</p> <p><b>&lt;BlockRequestDelay&gt;</b> - 16 bit Hex. This is the current value of the BlockRequestDelay attribute of the device that is making the request as set by the server. If the device supports the attribute then it SHALL include this field in the request. The value is in milliseconds.</p> <p><b>&lt;SequenceNumber&gt;</b> - 8 bit Hex which is the sequence number of the received request command. The user need to use this sequence number in response command (AT+IMGBRSP).</p> <p>Note: Upon receiving this prompt, the Host application shall use a AT+IMGBRSP command as response in 250 ms.</p>
---	---

	<p>If the host cannot serve the requested OTA file to ZigBee module any more during OTA file downloading progress, it can send a ZCL <i>DefaultResponse</i> with <i>status</i> NO_IMAGE_AVAILABLE (0x98) to the client</p>
<b>IMGPREQ:&lt;NodeID&gt;,&lt;EP&gt;,&lt;FieldControl&gt;,&lt;ManufCode&gt;,&lt;ImgType&gt;,&lt;FileVer&gt;,&lt;Offset&gt;,&lt;MaxDataSize&gt;,&lt;PageSize&gt;,&lt;ResponseSpacing&gt;[,&lt;RequestNodeAddress&gt;],&lt;SequenceNumber&gt;</b>	<p>This prompt is used to notify the Host application of received <i>ImagePageRequest</i> command.</p> <p>&lt;NodeID&gt; - 16 bit Hex. The source nodeID.</p> <p>&lt;EP&gt; - 8 bit Hex. The source endpoint.</p> <p>&lt;FieldControl&gt; - 8 bit Hex. It is used to indicate additional optional fields that may be included in the payload of Image Block Request command.</p> <p>Field control value 0x00 (bit 0 not set) indicates that the client is requesting a generic OTA upgrade file. Hence RequestNodeAddress, as an optional field, is not included.</p> <p>Field control value of 0x01 (bit 0 set) means that the client's IEEE address (RequestNodeAddress) is included in the payload.</p> <p>&lt;ManufCode&gt; - 16 bit Hex. The value shall be the device's assigned manufacturer code.</p> <p>&lt;ImgType&gt; - 16 bit Hex, image type.</p> <p>&lt;FileVer&gt; - 32 bit hex, representing the OTA image file version that is being requested.</p> <p>&lt;Offset&gt; - 32 bit Hex, File offset, its value indicates number of bytes of data offset from the beginning of the file. It essentially points to the location in the OTA image file that the client is requesting the data from. The value reflects the amount of (OTA upgrade image file) data (in bytes) that the client has received so far.</p> <p>&lt;MaxDataSize&gt; - 8 bit Hex, the value indicates the largest possible length of data (in bytes) that the client can receive at once. The server shall respect the value and not send the data that is larger than the maximum data size.</p> <p>&lt;PageSize&gt; - 8 bit Hex, the value indicates the number of bytes to be sent by the server before the client sends another Image Page Request command. In general, page size value shall be larger than the maximum data size value.</p> <p>&lt;ResponseSpacing&gt; - 8 bit Hex, the value indicates how fast the server shall send the data (via Image Block Response command) to the client. The value is determined by the client. The server shall wait at the minimum the (response)</p>

	<p>spacing value before sending more data to the client. The value is in milliseconds</p> <p>If the corresponding bit is set, then the RequestNodeAddress fields is expected:</p> <p>&lt;RequestNodeAddress&gt; - This is the IEEE address of the client device sending the Image Block Request command.</p> <p>&lt;SequenceNumber&gt; - 8 bit Hex which is the sequence number of the received request command. The user need to use this sequence number in response command (AT+IMGBRSP).</p> <p>Note: Upon receiving this prompt, the Host application shall prepare the OTA file page (usually including multiple blocks), then send blocks in sequence by using AT+IMGBRSP commands.</p> <p>The first ImageBlockResponse shall use the same sequence number contained in this prompt. Then the host shall use incrementing sequence number. Please be advised that that the sequence number may not be sequential (for example, if the server is also upgrading another client simultaneously); additionally ZCL sequence numbers are only 8-bit and may wrap.</p> <p>If the host cannot serve the requested OTA file to ZigBee module any more during OTA file downloading progress, it can send a ZCL <i>DefaultResponse</i> with <i>status</i> NO_IMAGE_AVAILABLE (0x98) to the client</p>
<p><b>UPGRADEREQ: &lt;NodeID&gt;,&lt;EP&gt;,&lt;Status&gt;,&lt;ManufCode&gt;,&lt;ImgType&gt;,&lt;FileVer&gt;,&lt;SequenceNumber&gt;</b></p>	<p>This prompt is used to notify the Host application of received <i>UpgradeEndRequest</i> command.</p> <p>&lt;NodeID&gt; - 16 bit Hex. The source network address.</p> <p>&lt;EP&gt; - 8 bit Hex. The source endpoint.</p> <p>&lt;Status&gt; - 8 bit Hex, status code.</p> <p>&lt;ManufCode&gt; - 16 bit Hex. The value shall be the device's assigned manufacturer code.</p> <p>&lt;ImgType&gt; - 16 bit Hex, image type</p> <p>&lt;FileVer&gt; - 32 bit Hex, newly downloaded OTA upgrade image file version.</p> <p>&lt;SequenceNumber&gt; - 8 bit Hex which is the sequence number of the received request command. The user need to use this sequence number in response command (AT+UPGRADE).</p> <p>Note: Upon receiving this prompt, the Host application shall use a AT+UPGRADE command as response in 250 ms.</p>

<p><b>OTAPOLICY:&lt;QueryPolicy&gt;,&lt;UpdatePolicy&gt;</b></p>	<p>This prompt is a response of the local command AT+GETPOLICY.          &lt;QueryPolicy&gt; - 8 bit Hex          00: Upgrade if server has newer          01: Downgrade if server has older          02: Reinstall if server has same          03: No next version          &lt;UpgradeRequestPolicy&gt;          00: Upgrade Now          01: Upgrade In a few minutes          02: Ask me later to upgrade          03: Abort upgrade</p>
<p><b>OTAIMGID:&lt;ManufCode&gt;,&lt;ImgType&gt;,&lt;FwVer&gt;,&lt;MinHwVer&gt;,&lt;MaxHwVer&gt;,&lt;HostDefinedNodeId&gt;,&lt;ImageSize&gt;</b></p>	<p>&lt;ManufCode&gt; - 16 bit Hex. The value shall be the device's assigned manufacturer code.          &lt;ImgType&gt; - 16 bit hex, OTA file's image type id          &lt; FwVer &gt; - 32 bit Hex, represents the server OTA image's firmware version.          &lt; MinHwVer &gt; - 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.          The high byte represents the version and the low byte represents the revision.          &lt; MaxHwVer&gt;- 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.          The high byte represents the version and the low byte represents the revision.  <b>Note: on the client side, the hardware version of the device should not be earlier than the minimum (hardware) version and should not be later than the maximum (hardware) version in order to run the OTA upgrade file.</b>          &lt;HostDefinedNodeId&gt; - 16 bit hex. This firmware provides a feature to allow the host to configure the client's node id in order to provide an image to a specific node (OTA client) for downloading. This parameter can be set by using AT+SETIMGID command.          &lt;ImageSize&gt; - 32 bit Hex, represents the server OTA image's size.</p>
<p>Some other prompts related to response will be described with corresponding commands in the followings sections.</p>	

Table 6: Prompt overview

### 3.3 Device Overview

Table 7 gives an overview of the ZigBee device types mentioned in this document.

Name	Device Types	ZigBee Naming Convention
COO	Coordinator	ZigBee Coordinator (ZC)
FFD	Router	ZigBee Router (ZR)
ZED	End Device (non-sleepy)	ZigBee End Device (ZED)
SED	Sleepy End Device	

Table 7: Device Overview

### 3.4 Addressing modes

Many of the AT commands take a device address as a parameter, which can usually be expressed in several different formats.

EUI64. The Extended Unique Identifier (EUI) of each module consists of 16 hexadecimal characters. This is flashed onto the chip at manufacture and cannot be changed by the user. It can be compared to the permanent MAC address of an IP-based device.

Network address (Node ID). 4 hexadecimal characters. This is allocated to the device when it joins the PAN and cannot be changed or preset, except that 0x0000 is always the coordinator. It is analogous to a temporary IP address.

Address table entry. Range 00-07. Used in some ZDO command.

Binding table entry. Range 10-1D (hexadecimal). Used to send ZCL messages through binding.

FF. In many commands address FF represents the local device.

### 3.5 AT Command Overview

The following table gives a quick reference of all commands available.

Command Overview	
<b>Module control and configuration</b>	
<b>ATI</b>	<b>Display Product Identification Information</b>
<b>ATZ</b>	<b>Software Reset</b>
<b>AT+MTOSTATUS</b>	<b>Concentrator Status</b>
<b>AT+MTOSTART</b>	<b>Concentrator Start</b>
<b>AT+MTOSTOP</b>	<b>Concentrator Stop</b>
<b>AT&amp;F</b>	<b>Restore Local Device's Factory Defaults</b>
<b>AT+RF</b>	<b>Send A Restore Factory Defaults Command To Remote Device</b>
<b>AT+BLOAD</b>	<b>Enter The Bootloader Menu</b>
<b>ATS</b>	<b>S-Register Access</b>
<b>AT+TOKDUMP</b>	<b>Display All S-Registers</b>
<b>AT+LONOFF</b>	<b>Switch Local Device On/Off</b>
<b>AT+PASSTHROUGH</b>	<b>Pass new Firmware Image To Remote Node</b>
<b>AT+RECOVER</b>	<b>Recover From A Failed Passthrough Attempt</b>
<b>AT+RAWZCL</b>	<b>Construct A Raw ZCL Message And Send To Target</b>
<b>AT+RAWZDO</b>	<b>Construct A Raw ZDO Message And Send To Target</b>
<b>AT+RAWZCLP</b>	<b>Send A Message With Specified Profile ID And Specified Source EP</b>
<b>AT+SENDUCASTB</b>	<b>Send Raw Binary Data To A Remote Node</b>
<b>AT+INTERPAN</b>	<b>Send An Interpan Message</b>
<b>AT+TIMERD</b>	<b>Set Flag to Serve Local Time To The Network</b>
<b>Network control and configuration</b>	
<b>AT+ESCAN</b>	<b>Scan The Energy Of All Channels</b>
<b>AT+PANSCAN</b>	<b>Scan For Active PAN's</b>
<b>AT+EN</b>	<b>Establish Network</b>
<b>AT+PJOIN</b>	<b>Switch on "Permit Joining" Flag</b>
<b>AT+JN</b>	<b>Join Network</b>
<b>AT+JPAN</b>	<b>Join Specific PAN</b>
<b>AT+SJN</b>	<b>Silent Join</b>
<b>AT+REJOIN</b>	<b>Initiate A Rejoin (used if the device is not a COO)</b>
<b>AT+DASSL</b>	<b>Disassociate Local Node From PAN</b>
<b>AT+DASSR</b>	<b>Disassociate Remote Node From PAN (ZDO)</b>
<b>AT+N</b>	<b>Display Network Information</b>
<b>AT+NTABLE</b>	<b>Display Neighbour Table (ZDO)</b>
<b>AT+RTABLE</b>	<b>Display Routing Table (ZDO)</b>
<b>AT+IDREQ</b>	<b>Request Node's NodeID (ZDO)</b>
<b>AT+EUIREQ</b>	<b>Request Node's EUI (ZDO)</b>
<b>AT+NODEDESC</b>	<b>Request Node's Descriptor (ZDO)</b>
<b>AT+POWERDESC</b>	<b>Request Node's Power Descriptor (ZDO)</b>
<b>AT+ACTEPDESC</b>	<b>Request Node's Active Endpoint List (ZDO)</b>
<b>AT+SIMPLEDESC</b>	<b>Request Endpoint's Simple Descriptor (ZDO)</b>
<b>AT+MATCHREQ</b>	<b>Find Nodes Which Match A Specific Descriptor (ZDO)</b>
<b>AT+ANNCE</b>	<b>Announce Local Device In The Network (ZDO)</b>
<b>AT+KEYUPD</b>	<b>Update the Network Key (ZDO)</b>
<b>AT+CCHANGE</b>	<b>Change the network's Channel</b>



<b>AT+CSWITCH</b>	<b>Switch the network's Channel Without Notify Other Nodes</b>
<b>Messaging</b>	
<b>AT+ATABLE</b>	<b>Display Address Table</b>
<b>AT+ASET</b>	<b>Set Address Table Entry</b>
<b>AT+LBTABLE</b>	<b>Display Local Binding Table</b>
<b>AT+BSET</b>	<b>Set Local Binding Table Entry</b>
<b>AT+BCLR</b>	<b>Clear Local Binding Table Entry</b>
<b>AT+BTABLE</b>	<b>Display Remote Binding Table (ZDO)</b>
<b>AT+BIND</b>	<b>Create Binding on Remote Device (ZDO)</b>
<b>AT+UNBIND</b>	<b>Delete Binding on Remote Device (ZDO)</b>
<b>AT+EBIND</b>	<b>End Device Binding</b>
<b>ZigBee discovery and attributes commands</b>	
<b>AT+DISCOVER</b>	<b>Discover HA Device On The HAN</b>
<b>AT+DISCOVERP</b>	<b>Discover Devices Using Profile ID and Cluster ID</b>
<b>AT+CLUSDISC</b>	<b>Discover All Supported Clusters on A Remote Device</b>
<b>AT+ATTRDISC</b>	<b>Discover Supported Attributes on A Remote Device</b>
<b>AT+READATR</b>	<b>Get Server Attributes From Targets</b>
<b>AT+READCATR</b>	<b>Get Client Attributes From Targets</b>
<b>AT+READMATR</b>	<b>Get Manufacturer Specific Server Attributes From Targets</b>
<b>AT+READMCATR</b>	<b>Get Manufacturer Specific Client Attributes From Targets</b>
<b>AT+WRITEATR</b>	<b>Write Server Attributes To Targets</b>
<b>AT+WRITECATR</b>	<b>Write Client Attributes To Targets</b>
<b>AT+WRITEMATR</b>	<b>Write Manufacturer Specific Server Attributes To Targets</b>
<b>AT+WRITEMCATR:</b>	<b>Write Manufacturer Specific Client Attributes To Targets</b>
<b>AT+READMRPTCFG</b>	<b>Read Manufacture Reporting Configuration From Targets</b>
<b>AT+READRPTCFG</b>	<b>Read Reporting Configuration From Targets</b>
<b>AT+CFGMRPT</b>	<b>Configure Manufacture Defined Attribute Reporting</b>
<b>AT+CFGMRPT</b>	<b>Configure Attributes Reporting</b>
<b>AT+SETATR</b>	<b>Set Value Of A Local ZCL Attribute</b>
<b>AT+GETATR</b>	<b>Get Value Of A Local ZCL Attribute</b>
<b>Combined Interface specified commands</b>	
<b>AT+IDENTIFY</b>	<b>Starts or Stops The Receiving Device Identifying Itself.</b>
<b>AT+IDQUERY</b>	<b>Query If Target Device(s) In Identifying Mode</b>
<b>AT+GPADD</b>	<b>Add Group On Target Device</b>
<b>AT+GPADIFID</b>	<b>Add Group Membership On Target That Is In Identifying Mode</b>
<b>AT+GPVIEW</b>	<b>View Group Information Of Target Devices</b>
<b>AT+GPGET</b>	<b>Get Group Membership Information From Target Devices</b>
<b>AT+GPRMV</b>	<b>Remove A Group Membership On Target Device</b>
<b>AT+GPRMALL</b>	<b>Remove All Group Associations On Target Device</b>
<b>AT+SCADD</b>	<b>Add Scenes</b>
<b>AT+SCVIEW</b>	<b>View Scenes</b>
<b>AT+SCRMV</b>	<b>Remove Scenes</b>
<b>AT+SCRMALL</b>	<b>Remove All Scenes</b>
<b>AT+SCSTORE</b>	<b>Store Scenes</b>
<b>AT+SCRECAL</b>	<b>Recall Scenes</b>
<b>AT+SCGETMSH</b>	<b>Get Scene Membership</b>
<b>AT+RONOFF</b>	<b>Switching Target Devices Between 'On' and 'Off' States</b>
<b>AT+LCMVTOLEV</b>	<b>Level Control Move to Level Command</b>
<b>AT+LCMV</b>	<b>Level Control Move Command</b>



<b>AT+LCSTEP</b>	<b>Level Control Step Command</b>
<b>AT+LCSTOP</b>	<b>Level Control Stop Command</b>
<b>AT+CCMVTOCOL</b>	<b>Colour Control Move to Colour Command</b>
<b>AT+CCMVCOL</b>	<b>Colour Control Move Colour Command</b>
<b>AT+CCSTEPCOL</b>	<b>Colour Control Step Colour Command</b>
<b>AT+CCMVTOHUE</b>	<b>Colour Control Move to Hue Command</b>
<b>AT+CCMVHUE</b>	<b>Colour Control Move Hue Command</b>
<b>AT+CCSTEPHUE</b>	<b>Colour Control Step Hue Command</b>
<b>AT+CCMVTOSAT</b>	<b>Colour Control Move to Saturation Command</b>
<b>AT+CCMVSAT</b>	<b>Colour Control Move Saturation Command</b>
<b>AT+CCSTEP SAT</b>	<b>Colour Control Step Hue Command</b>
<b>AT+CCMVTOHUS</b>	<b>Colour Control Move To Hue and Saturation Command</b>
<b>AT+CCMVTOCT</b>	<b>Colour Control Move to Colour Temperature Command</b>
<b>AT+TSTATSET</b>	<b>Thermostat Set</b>
<b>AT+DRLOCK</b>	<b>Lock/Unlock Door</b>
<b>AT+WNPGSETPT</b>	<b>Window Program Setpoint</b>
<b>AT+WNSETPT</b>	<b>Window Go To Lift/Tile Setpoint</b>
<b>AT+WNVALUE</b>	<b>Window Go To Lift/Tile Value</b>
<b>AT+WNPCTG</b>	<b>Window Go To Lift/Tile Percentage</b>
<b>AT+WNCTRL</b>	<b>Open, Close or Stop Window</b>
<b>CIE specified commands</b>	
<b>AT+WARNING</b>	<b>Send A Start Warning Command To A Warning Device</b>
<b>AT+SQUAWK</b>	<b>Send A Squawk Command To A Warning Device</b>
<b>AT+ADDZENTRY</b>	<b>Add Zone Table Entry</b>
<b>AT+DELZENTRY</b>	<b>Delete Zone Table Entry</b>
<b>AT+ZENTRY</b>	<b>Print A Local Zone Table Entry</b>
<b>AT+ZTABLE</b>	<b>Print Local Zone Table</b>
<b>AT+ZCLEAR</b>	<b>Clear Local Zone Table</b>
<b>AT+PRINTMZTYPE</b>	<b>Print Local Table of Manufacture Defined Zone Type</b>
<b>AT+MZONETYPE</b>	<b>Add Manufacture Defined Zone Type in Local Zone Type Table</b>
<b>AT+ARMPRINT</b>	<b>Show CIE Arm Mode</b>
<b>AT+PRINTZMAP</b>	<b>Print Local Zone Map</b>
<b>AT+EDITZMAP</b>	<b>Edit Local Zone Map</b>
<b>AT+ZINFORSP</b>	<b>Construct And Send A Get Zone Information Response</b>
<b>AT+ZTEST</b>	<b>Place A Remote Zone Server Device into Test Mode</b>
<b>AT+ZNORMAL</b>	<b>Set A Remote Zone Server Device To Normal Working Mode</b>
<b>Tunnel Commands</b>	
<b>AT+TNLOPEN</b>	<b>Send An Open Tunnel Request To The Target</b>
<b>AT+TNLCLOSE</b>	<b>Send A Close Tunnel Command To The Target</b>
<b>AT+TNLSEND</b>	<b>Send Data To The Target Through An Established Tunnel</b>
<b>AT+TNLSTATUS</b>	<b>Check Local Tunnel Information</b>
<b>Time Cluster Commands</b>	
<b>AT+SETTIME</b>	<b>Set The Local Time</b>
<b>AT+GETTIME</b>	<b>Get The Local Time</b>
<b>OTA Communication Commands</b>	
<b>AT+IMGNOTIFY</b>	<b>Notify OTA client of the newly available Image</b>
<b>AT+UPGRADE</b>	<b>Instruct a device to upgrade now</b>
<b>AT+QIMGRSP</b>	<b>Query next image response</b>
<b>AT+IMGBRSP</b>	<b>Image block response</b>

<b>AT+DFTRSP</b>	<b>Default response for OTA Request</b>
<b>OTA Server Configuration Commands</b>	
<b>AT+GETPOLICY</b>	<b>Print the OTA server policy</b>
<b>AT+SETPOLICY</b>	<b>Set the OTA server policy</b>
<b>AT+SETIMGID</b>	<b>Set the OTA server image id and some other relevant information</b>
<b>AT+GETIMGID</b>	<b>Print the OTA server image id and hardware version</b>
<b>AT+SETHWVER</b>	<b>Set the allowable OTA client's hardware version range</b>
<b>Poll Control Client Commands</b>	
<b>AT+FPSET</b>	<b>Set Local Flag and Timeout for Starting Fast Poll</b>
<b>AT+FPSTOP</b>	<b>Stop Fast Poll</b>
<b>AT+LPINTVL</b>	<b>Set Long Poll Interval</b>
<b>AT+SPINTVL</b>	<b>Set Short Poll Interval</b>

Table 8: Command Overview

### 3.5.1 Common Command Set

#### 3.5.1.1 Module Control & Configuration Commands

##### I – Display Product Identification Information

Execute Command	Response
<b>ATI</b>	<b>Telegesis &lt;DeviceName&gt;</b>  <b>&lt;Firmware Revision&gt;</b>  <b>&lt;EUI64&gt;</b>  <b>OK</b>  <DeviceName> is the device type <Firmware Revision> is the firmware revision <EUI64> is the device's IEEE 802.15.4 Identifier

##### Z – Software Reset

Execute Command	Response
<b>ATZ</b>	<b>OK</b>  Module performs software reset. All non-volatile S Registers keep the user defined values, if the module was part of a PAN it will remain part of it.

##### +MTOSTATUS – Check Local Many-To-One Route Discovery

Execute Command	Response
<b>AT+MTOSTATUS</b>	<b>ACTIVE:Y /N</b>  <b>OK</b>  Y means Enabled N means Disabled

##### +MTOSTART – Start Many-To-One Route Discovery

Execute Command	Response
<b>AT+MTOSTART</b>	<b>MTOSTART</b>  <b>OK</b>

### **+MTOSTOP – Stop Many-To-One Route Discovery**

Execute Command

**AT+MTOSTOP**

*Note:*

If a message is not delivered successful (usually show ERROR: 66), MTO will be re-activated automatically.

Response

**MTOSTOP**

**OK**

### **&F – Restore Local Device's Factory Defaults**

Execute Command

**AT&F**

Response

**OK**

Module performs a factory reset. All non-volatile S Registers are updated with their factory defaults and the node leaves the currently joined network.

### **+RF – Restore Target Device's Factory Defaults**

Execute Command

**AT+RF:<Address>,<EP>,<SendMode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

*Notes:*

If sending the command to a group, the second parameter <EP> will be ignored. User can leave it blank. For example:

AT+RF:0001,0A,0 (send the command to a node)

AT+RF:1234,,1 (send the command to a group)

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** represents the error code (please see section 0).

*Note:*

It may also receive a default response to show the device does not support this command, as this is an optional command in Basic cluster.

### **+BLOAD – Enter The Boot Loader Menu**

Execute Command

#### **AT+BLOAD**

Please note: the bootloader will run at a baud rate of 115k, no parity, and 8 data bits regardless of the current serial port settings.

Response

**<entering bootloader>**

The device leaves the AT command line and enters the Ember bootloader menu for downloading new firmware. A description of the bootloading process can be found in the section 7 of this document. A description of the bootloading process can be found in the Evaluation Kits Product Manual.

### **S – S-Register Access**

Read Command

#### **ATSXX[x]?**

Response

**<data>**

**OK**

or **ERROR:<errorcode>**

The module displays the contents of S-register xx or an error message, where <errorcode> represents the error code explained in section 0.

All 16-bit registers can also be accessed bit by bit. In order to do this [x] may specify the bit which is to be read. The result when reading a single bit will always be 0 or 1.

Write Command

#### **ATSXX[x]=<data>**

Response

**OK or ERROR:<errorcode>**

*Notes*

Some S-Registers require a password for write access. See S-Register description for details.

**The default password is “password”.**

Some S-Registers are read-only and will return an error if you are trying to write to them

The data is written to S-register number XX and if applicable stored in non-volatile memory. The data format for each individual S Register is given in the S-Register description.

**<errorcode>** please see section 0.

All 16-bit registers individual bits can also be set or cleared by specifying the bit using [x] and setting it to either 0 or 1.

### **TOKDUMP – Display All S-Registers**

Execute Command	Response
<b>AT+TOKDUMP</b>	<b>&lt;data&gt;</b>
<i>Notes</i>	<b>OK</b>
Only used on the local node.	The module displays the contents of all local S-Registers. The data format for each individual S Register is given in the S-Register description in section 5.

### **+LONOFF – Set On or Off Status for the Local Node (For Local LED )**

Execute Command	Response
<b>AT+LONOFF:&lt;On/Off&gt;</b>	<b>OK</b>
<b>&lt;On/Off&gt;</b> - Use 1 means on; 0 means off. This parameter can be omit and just use AT+LONOFF to toggle the LED on USB.	or
<i>Use on:</i>	<b>ERROR:&lt;errorcode&gt;</b> (please see section 0).
Local node	

### **+PASSTHROUGH – Pass new Firmware Image To Remote Node (ETRX3 Series Only)**

Execute Command	Response
<b>AT+PASSTHROUGH:&lt;EUI64&gt;,&lt;password&gt;</b>	<b>PASSTHROUGH BLOAD...</b>
<i>Use on:</i> Source: FFD, COO Destination: FFD, COO, ZED	Please start .ebl upload image...
<i>Notes</i>	Remote Response
The default password for R3xx nodes is “password”. A description of the passthrough process can be found in the Evaluation Kits Product Manual; it is the same procedure as cloning.	<b>ENTERING BLOAD</b>
The ETRX357(HR)-LRS module cannot be reliably upgraded by the passthrough process	or <b>ERROR&lt;errorcode&gt;</b>
	Where <errorcode> represents the error code explained in section 4.
	<password> represents the remote node’s 8-character password. After completion a soft reset is caused on the remote end.

**+RECOVER – Recover From A Failed Clone or Passthrough Attempt**

Execute Command

**AT+RECOVER**
*Use on:*

Source: FFD, COO

Destination: All device types

*Note*

Use this command in cases where Passthrough Bootloading operation was interrupted and the target device therefore remains in the bootloader. Whenever the target device has been reset channel 13 must be used for recovering.

Response

**Recovering...**

or

**ERROR<errorcode>**

Where <errorcode> represents the error code explained in section 4.

Enters Passthrough (ETRX3 Series) mode to a remote node which is already in the bootloader.

**+RAWZCL – Construct A Raw ZCL Message And Send To Target**

Execute Command

**AT+RAWZCL:<NodeID>,<EP>,<ClusterID>  
,<data>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<data>** - a constructed ZCL command, in hexadecimal format including frame control and sequence number .

For instance:

A raw *ReadAttributeRequest* command to read Basic cluster attribute (0x0001) can be:

AT+RAWZCL:1234,01,0000,0002000100

Payload part: 00 – frame control

02 – sequence number

00 – commandID

0100 – attributeID (little endian)

Please check ZigBee Cluster Library for reference.

Response

**OK**
**<Response data>**

or

**ERROR:<errorcode>** (please see section 0).

The response may vary depending on the sent command.

Note: When constructing a Raw ZCL message, please check ZigBee ZCL Specification [5] and ZigBee HA Specification [6] for references. Incorrect construction can be transmitted but will not expect any response.

### **+RAWZDO – Construct A Raw ZDO Message And Send To Target**

Execute Command	Response
<b>AT+RAWZDO:&lt;NodeID&gt;,&lt;ClusterID&gt;,&lt;data&gt;</b> <b>&gt;</b>	<b>OK</b>
<b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number, network address of a target device or FFFD to broadcast to all routers.	<b>CMD:&lt;ClusterID&gt;,&lt;PAYLOAD&gt;&lt;Response data&gt;</b> or
<b>&lt;ClusterID&gt;</b> - ZDO Command ID, see ZigBee Specification [4] for more information.	<b>ERROR:&lt;errorcode&gt;</b> (please see section 0).
<b>&lt;data&gt;</b> - constructed payload of a ZDO command (please check ZigBee Specification for reference)	The response may vary depending on sent command.  Note: When constructing a Raw ZDO message, please check ZigBee Specification [4] for references. Incorrect construction can be transmitted but will not expect any response.

### **+RAWZCLP – Send A Raw ZCL Message With Specified ProfileID And Source Endpoint**

Execute Command	Response
<b>AT+RAWZCL:&lt;NodeID&gt;,&lt;DstEP&gt;,[&lt;SrcEP&gt;],</b> <b>[&lt;ProfileID&gt;,&lt;ClusterID&gt;,&lt;data&gt;</b>	<b>OK</b>
<b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number, network address of a remote device.	<b>&lt;Response data&gt;</b> or
<b>&lt;DstEP&gt;</b> - 8 bit hexadecimal number, destination endpoint of a remote device.	<b>ERROR:&lt;errorcode&gt;</b> (please see section 0).
<b>[&lt;SrcEP&gt;</b> - 8 bit hexadecimal number, source endpoint of a local device. If it is omitted, source endpoint is set to 0x02	The response may vary depending on the sent command.  Note: When constructing a Raw ZCL message, please check ZigBee ZCL Specification [5] and ZigBee HA Specification [6] for references. Incorrect construction can be transmitted but will not expect any response.
<b>[&lt;ProfileID&gt;</b> - 16 bit hexadecimal number which represents profile ID. E.g. 0x0104 for ZigBee home automation profile. If it is omitted, profile ID is set to 0xC216	
<b>&lt;ClusterID&gt;</b> - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2	
<b>&lt;data&gt;</b> - a constructed ZCL command in hexadecimal format (please check ZigBee Cluster Library for references)	



### **+SENDUCASTB – Send Raw Binary Data To A Remote Node**

Execute Command

**AT+SENDUCASTB:<NodeID>,<DstEP>,<SendMode>,<SrcEP>,<ProfileID>,<ClusterID>,<dataLength>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device.

**<DstEP>** - 8 bit hexadecimal number, destination endpoint of a remote device.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<SrcEP>** - 8 bit hexadecimal number, source endpoint of a local device.

**<ProfileID>** - 16 bit hexadecimal number which represents profile ID.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<dataLength>** - 8 bit hex, input data length

Response

><data being entered>

**OK**

or

**ERROR:<errorcode>** (please see section 0).

The response may vary depending on the sent command.

Note: When constructing a Raw message, please check ZigBee ZCL Specification [5] and ZigBee HA Specification [6] for references. Incorrect construction can be transmitted but will not expect any response.

### **+INTERPAN – Send an Interpan Command**

Execute Command	Response
<b>AT+INTERPAN:&lt;AddressMode&gt;,&lt;DstAddress&gt;,&lt;DstPAN&gt;,&lt;ProfileID&gt;,&lt;ClusterID&gt;,&lt;Payload&gt;</b>	<b>OK</b>
<p><b>&lt;AddressMode&gt;</b> - 8 bit hexadecimal number. The user shall use this parameter to specify which type of the destination address is used. 00 - Node ID; 01- Group ID; 02- EUI address</p> <p><b>&lt;DstAddress&gt;</b> - 16 bit hexadecimal number if the user uses Node ID or Group ID.</p> <p>Or EUI address, if the user uses long destination address.</p> <p><b>&lt;DstPAN&gt;</b> - 16 bit hexadecimal number representing destination PAN ID.</p> <p><b>&lt;ProfileID&gt;</b> - 16 bit hexadecimal number representing profile ID. e.g. 0x0104 for Home automation. 0xC05E for ZigBee Light Link.</p> <p><b>&lt;ClusterID&gt;</b> - 16 bit hex number representing Cluster ID.</p> <p><b>&lt;Payload&gt;</b> - Command payload, please input ASCII hex data.</p>	<p>Or</p> <p><b>ERROR:&lt;errorcode&gt;</b> (please see section 0).</p>

### **TIMERD – Set Flag to Serve Local Time To The Network**

Execute Command	Response
<b>AT+TIMERD</b>	<b>OK</b>
<p><i>Notes</i></p> <p>This command need to be used after the user set Time cluster attribute (time, timeStatus attributes) with valid value.</p> <p>When CICIE initiates, it will have invalid time. So it will respond to a read time attribute command with a default response and failure status.</p> <p>The user (host) need to set time cluster attributes (time and time status) by using AT+SETATR command. After that the user need to apply this command in order to serve the time to a HA network. From then on, the CICIE will send read attribute response with its maintained clock time upon receiving a request.</p>	<p>Please check Appendix C for more information of using CICIE time server cluster.</p>

### 3.5.1.2 Network Control & Configuration Commands

#### **+ESCAN – Scan The Energy Of All Channels**

Execute Command

**AT+ESCAN**

*Use on:*  
All nodes

*Note:*

Scanning all channels can take up to 4 seconds. Channels masked out in S00 are not scanned.

Response

**+ESCAN:**

**11:XX**

...

**26:XX**

**OK**

or **ERROR:<errorcode>** (please see section 0).

**XX** represents the average energy on the respective channel (see description in section 6).

**+PANSCAN – Scan For Active PANs**

Execute Command	Response
<b>AT+PANSCAN</b>	<b>+PANSCAN:&lt;channel&gt;,&lt;PANID&gt;,&lt;EPANID&gt;,XX,b</b>
<i>Use on:</i>	<b>OK</b>
All nodes	or
	<b>+PANSCAN:&lt;channel&gt;,&lt;PID&gt;,&lt;EPID&gt;,XX,b,&lt;rssi&gt;,&lt;LQI&gt;</b>
	or <b>ERROR:&lt;errorcode&gt;</b> (please see section 0).
<i>Note:</i>	<b>&lt;channel&gt;</b> represents the channel,
Scanning for active PANs can take up to 4 seconds. The node does not join any of the PANs found. Channels masked out in S00 are not scanned.	<b>&lt;PANID&gt;</b> the PAN ID,
	<b>&lt;EPANID&gt;</b> the extended PAN ID,
	The node gives a list of all PANs found.
	<b>XX</b> the ZigBee stack profile (00 = Custom, 01 = ZigBee, 02 = ZigBee PRO)
	<b>b</b> indicates whether the network is allowing additional nodes to join (1 = joining permitted).
	If bit 1 of S63 is set the response includes RSSI and LQI

### **+EN – Establish Network**

Execute Command

**AT+EN:[<channel>],[<POWER>],[<PANID>]**

*Use on:*

Coordinator which are not part of a PAN

*Note:*

Establishing a PAN can take up to 4 seconds.

This command can only be executed if the local node is not part of a PAN already.

Response

**OK**

**JPAN:<channel>,<PANID>,<EPANID>**

or

**ERROR:<errorcode>** (please see section 0).

The local node becomes a coordinator and performs an energy scan on all channels selected in S00. It then starts a PAN with a random unused PAN ID and extended PAN ID on the quietest channel. If a PAN ID and/or extended PAN ID is specified in S02 or S03 the provided IDs are used instead of random ones, given the selected IDs are not already in use by other networks within range

### **+JN – Join Network**

Execute Command

**AT+JN**

*Use on:*

All nodes which are not part of a PAN

*Note:*

Joining a PAN can take up to 4 seconds, depending on the number of channels which need scanning.

This command can only be executed if the local node is not part of a PAN already.

Response

**OK**

**JPAN:<channel>,<PANID>,<EPANID>**

or **ERROR:<errorcode>** (please see section 0).

The local node scans all channels selected in register S00 for the existence of a PAN. When finding a PAN which allows joining it will automatically join via the router with the best signal quality.

Where registers S02 and S03 differ from the default value of all zeroes the node will only join a PAN with the specified PAN ID and/or extended PAN ID.

### **+PJOIN – Permit joining**

Read Command

**AT+PJOIN[:<sec>,<NodeID>]**

**<sec>** - 8 bit hexadecimal number which represents the length of time in seconds during which the ZigBee coordinator or router will allow associations

**<NodeID>** - 16 bit hexadecimal number, network address of a target device. If FFFC is used, the constructed command will be sent as a broadcast.

*Use on:*

Coordinator or router.

*Note:*

The HA network should not leave permit join on permanently, S0A has been set to switch off permit join.

If <sec> and <NodeID> are omitted, the CICIE will permit join for 60 seconds.

If there is <sec> input, the CICIE will apply the period that is specified by the user.

If <NodeID> is filled with FFFC, the CICIE will broadcast a permit join command to the other nodes on the HA network to set permit join on. If the user wants to send a unicast permit join command to a device, <NodeID> should be the network address of that node.

Response

**OK**

or

**ERROR<errorcode>**

**<errorcode>** please check section 0.

If the command is undeliverable (for unicast case), prompt “NACK” will show. Please note: there is no ACK prompt for this command for both unicast and broadcast.

### **+JPAN – Join Specific PAN**

Execute Command

**AT+JPAN:[<channel>],[<PANID>],[<EPANID>]**

**<channel>** - 2 digit decimal number which represents channel number. Range from 11 to 26.

**<PANID>** - 16 bit hexadecimal number which represents short PAN ID of ZigBee network.

**<EPANID>** - 64 bit hexadecimal number which represents extended PAN ID of ZigBee network.

Each parameter is optional for input, but “,” is required when the parameter is left blank.

*Examples:*

**AT+JPAN:20 (only specifies channel)**

**AT+JPAN: ,1234, ( only specifies PANID)**

*Note:*

This command can only be executed if the local node is not part of a PAN already.

Response

**OK**

**JPAN:<channel>,<PANID>,<EPANID>**

or

**ERROR: <errorcode>** (please see section 0).

The local node joins a particular PAN on <channel> with the specified <PID> or <EPID> via the router with an adequate signal quality and the fewest hops to the COO

### **+SJM – Silent Join**

Execute Command

**AT+SJM:<channel>,<TCEUI64>,<NMNodeID>,<nwk update ID>**

“Silent” joining is joining via the commissioning method. All data required to enter the network is provided to the node, so that no joining procedure itself is required. The node will appear in the target network without any joining procedure given that the supplied data is correct.

**<channel>** is a decimal number

Other parameters are hexadecimal

*Use on*

All joining Devices

*Example*

**AT+SJM:11,000D6F00000AAAD0,AFFE,00**

Response

**JPAN:<channel>,<PANID>,<EPANID>**

**OK**

or **ERROR:<errorcode>**

**<errorcode>** represents the error code explained in section 4.

The local node will become part of the network with the channel specified in **<channel>**, the trust centre EUI64 specified in **<TC EUI64>**, the NodeID of the network manager specified in **<NM NodeID>**, the 8 bit network update ID specified in **<nwk update ID>**, the network key provided in S08, the trust centre link key provided in S09, the PAN ID provided in S02 and the extended PAN ID provided in S03. It is assumed that the key-sequence-number of the network key is 0 when issuing this command.

### **+REJOIN – Rejoin the network**

Execute Command

**AT+REJOIN:b**

If b is set to 0 join without the known network key (unencrypted) and if b is set to 1 join encrypted.

*Notes*

This command cannot be used on a COO.

Response

**OK**

or **ERROR<errorcode>**

**<errorcode>** please check section 6

If the contact with the network has been lost because the network has changed channel, or updated its encryption key the command AT+REJOIN can be used to re-join the network.



### **+DASSL – Disassociate Local Device From PAN**

Execute Command

**AT+DASSL**

*Use on*

All Devices

*Note*

Use with care on a Coordinator. It will not be able to re-join the PAN

Response

**OK**

or **ERROR<errorcode>**

Prompt

**LeftPAN**

**<errorcode>** please see section 4.

Instruct local device to leave the PAN.

### **+DASSR – Disassociate Remote Node from PAN**

**(ZDO)**

Execute Command

**AT+DASSR:<address>**

Where <address> can be a node's EUI64, Node ID or address table index

*Use on*

All Devices

*Note*

Use with care when targeting a Coordinator. It will not be able to re-join the PAN

*Remote Action*

Node leaves PAN

Response

**SEQ:XX**

**OK**

or **ERROR:<errorcode>**

**<errorcode>** please see section 4.

Instruct device to leave the PAN.

Prompt

**LeftPAN**

**+N – Display Network Information**

Read Command	Response
<b>AT+N</b>	<b>+N=&lt;devicetype&gt;,&lt;channel&gt;,&lt;power&gt;,&lt;PANID&gt;,&lt;EPANID&gt;</b> or <b>+N=NoPAN</b> followed by <b>OK</b>
<i>Use on</i>	<b>&lt;devicetype&gt;</b> represents the node's functionality in the PAN (FFD,COO,ZED,SED)
All Devices	<b>&lt;power&gt;</b> the node's output power in dBm <b>&lt;channel&gt;</b> the IEEE 802.15.4 radio channel (11-26) <b>&lt;PANID&gt;</b> the node's PAN ID <b>&lt;EPANID&gt;</b> the node's extended PAN ID.

**+NTABLE – Display Neighbour Table**
**(ZDO)**

Read Command	Response																				
<b>AT+NTABLE:XX,&lt;address&gt;</b>	<b>SEQ:XX</b> <b>OK</b> or <b>ERROR&lt;errorcode&gt;</b>																				
Where XX is the start index of the remote LQI table and <address> can be the remote node's EUI64, NodeID or address table entry.	This command requests the target node to respond by listing its neighbour table starting from the requested index. Can be used to find the identity of all ZigBee devices in the network including non-Telegesis devices.																				
Use on	Prompt (example) <b>NTable:&lt;NodeID&gt;,&lt;errorcode&gt;</b> <b>Length:03</b>																				
FFD, COO as the target device	<table><tr><th>No.</th><th>Type</th><th>EUI</th><th>ID</th><th>LQI</th></tr><tr><td>0.</td><td>FFD</td><td>000D6F000015896B</td><td>BC04</td><td>FF</td></tr><tr><td>1.</td><td>FFD</td><td>000D6F00000B3E77</td><td>739D</td><td>FF</td></tr><tr><td>2.</td><td>FFD</td><td>000D6F00000AAD11</td><td>75E3</td><td>FF</td></tr></table>	No.	Type	EUI	ID	LQI	0.	FFD	000D6F000015896B	BC04	FF	1.	FFD	000D6F00000B3E77	739D	FF	2.	FFD	000D6F00000AAD11	75E3	FF
No.	Type	EUI	ID	LQI																	
0.	FFD	000D6F000015896B	BC04	FF																	
1.	FFD	000D6F00000B3E77	739D	FF																	
2.	FFD	000D6F00000AAD11	75E3	FF																	
Note:	In this example the neighbour table of the remote node with the short ID shown in <NodeID> contains three entries (hexadecimal), which are displayed. In case the table contains more than three entries it may be required to repeat this command and increase the index count until the full table is derived.																				
Also the local node can be the target of this command (e.g. use address table entry FF as the address)	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.  After successful transmission, the sequence number of the unicast is stated using the “SEQ:XX” prompt. When acknowledged (or not) the accompanying “ACK:XX” (or “NACK:XX”) prompt is displayed.																				

+RTABLE – Display Routing Table (ZDO)																	
Read Command	Response																
<b>AT+RTABLE:XX,&lt;address&gt;</b>	<b>SEQ:XX</b>																
Where XX is the start index of the remote routing table and <address> can be the remote node’s EUI64, NodeID or address table entry.	<b>OK</b>																
	or <b>ERROR&lt;errorcode&gt;</b>																
<i>Use on</i>	This command requests the target node to respond by listing its routing table starting from the requested index.																
FFD, COO as the target device	Prompt (example) <b>RTable:&lt;NodeID&gt;,&lt;errorcode&gt;</b> <b>Length:03</b>																
	<table><tr><th>No.</th><th>Dest</th><th>Next</th><th>Status</th></tr><tr><td>0.</td><td>1234</td><td>ABCD</td><td>00</td></tr><tr><td>1.</td><td>4321</td><td>739D</td><td>00</td></tr><tr><td>2.</td><td>0000</td><td>0000</td><td>03</td></tr></table>	No.	Dest	Next	Status	0.	1234	ABCD	00	1.	4321	739D	00	2.	0000	0000	03
No.	Dest	Next	Status														
0.	1234	ABCD	00														
1.	4321	739D	00														
2.	0000	0000	03														
<i>Note:</i>																	
Also the local node can be the target of this command (e.g. use address table entry FF as local address)	In this example the routing table of the remote node with the short ID shown in <NodeID> contains 64 entries (hexadecimal 0x40), of which the first three are displayed. When the table contains more than the displayed entries it may be required to repeat this command and increase the index count until the full table is derived.																
	The status shown is as described in table 2.128 of the ZigBee Pro Specification [4].																
	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.																
	After successful transmission, the sequence number of the unicast is stated using the “SEQ:XX” prompt. When acknowledged (or not) the accompanying “ACK:XX” (or “NACK:XX”) prompt is displayed.																

<b>+IDREQ – Request Node's NodeID (ZDO)</b>	
Execute Command	Response
<b>AT+IDREQ:&lt;Address&gt;[,XX]</b>	<b>OK</b>
Where <Address> can be a node's EUI64, or address table entry and XX is an optional index number. When an index number is provided, an extended response is requested asking the remote device to list its associated devices (i.e. children).	or <b>ERROR:&lt;errorcode&gt;</b>
Sends a broadcast to obtain the specified Device's NodeID and optionally also elements of its associated devices list.	<b>&lt;errorcode&gt;</b> represents the error code in
<i>Use on</i>	Prompt
All Devices	<b>AddrResp:&lt;errorcode&gt;[,&lt;NodeID&gt;,&lt;EUI64&gt;]</b>
<i>Note</i>	<b>[nn. &lt;NodeID&gt;]</b>
Providing FF as an address table entry addresses the local node	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.
	<EUI64> is the Remote node's EUI64 and <NodeID> is its NodeID. In case an extended response has been requested the requested NodeIDs from the associated devices list are listed as well.

<b>+EUIREQ – Request Node's EUI64 (ZDO)</b>	
Execute Command	Response
<b>AT+EUIREQ:&lt; Address&gt;,&lt;NodeID&gt;[,XX]</b>	<b>SEQ:XX</b>
Where <Address> is the EUI64, NodeID or address table entry of the node which is to be interrogated about the node with the Node ID specified in <NodeID>. XX is an optional index number. In case an index number is provided, an extended response is requested asking the remote device to list its associated devices (i.e. children).	<b>OK</b>
Sends a unicast to obtain the specified device's EUI64 and optionally also elements of its associated devices list (extended response).	or <b>ERROR:&lt;errorcode&gt; (please see section 4)</b>
<i>Use on</i>	Prompt
All Devices	<b>AddrResp:&lt;errorcode&gt;[,&lt;NodeID&gt;,&lt;EUI64&gt;]</b>
<i>Note</i>	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.
Providing FF as an address table entry addresses the local node	<EUI64> is the Remote node's EUI64 and <NodeID> is its NodeID. Where an extended response has been requested the requested NodeIDs from the associated devices list are listed.
	After successful transmission, the sequence number of the unicast is stated using the "SEQ:XX" prompt. When acknowledged (or not) the accompanying "ACK:XX" (or "NACK:XX") prompt is displayed.

<b>+NODEDESC – Request Node’s Descriptor (ZDO)</b>	
Execute Command	Response
<b>AT+NODEDESC:&lt;Address&gt;,&lt;NodeID&gt;</b>	<b>SEQ:XX</b>
Where <Address> is the EUI64, NodeID or Address table entry of the node which is to be interrogated about the node with the NodeID specified in <NodeID>.	<b>OK</b> or
Sends a unicast to obtain the specified device’s node descriptor.	<b>ERROR:&lt;errorcode&gt;</b> (see section 4)
<i>Use on</i>	Prompt (example) <b>NodeDesc:&lt;NodeID&gt;,&lt;errorcode&gt;</b> <b>Type:FFD</b> <b>ComplexDesc:No</b> <b>UserDesc:No</b> <b>APSFlags:00</b> <b>FreqBand:40</b> <b>MacCap:8E</b> <b>ManufCode:1010</b> <b>MaxBufSize:52</b> <b>MaxInSize:0080</b> <b>SrvMask:0000</b> <b>MaxOutSize:0080</b> <b>DescCap:00</b>
All Devices	
<i>Note</i>	
Providing FF as an address table entry addresses the local node	In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.  <NodeID> is the Remote node’s NodeID. In addition the node descriptor is displayed. The individual fields of the Node Descriptor are described in section 2.3.2.3 of the ZigBee Pro specification [4].  After successful transmission, the sequence number of the unicast is stated using the “SEQ:XX” prompt. When acknowledged (or not) the accompanying “ACK:XX” (or “NACK:XX”) prompt is displayed.

<b>+POWERDESC – Request Node's Power Descriptor (ZDO)</b>	
Execute Command	Response
<b>AT+POWERDESC:&lt;Address&gt;,&lt;NodeID&gt;</b>	<b>SEQ:XX</b>
Where <Address> is the EUI64, NodeID or Address table entry of the node which is to be interrogated about the node with the NodeID specified in <NodeID>.	<b>OK</b>
Sends a unicast to obtain the specified device's power descriptor.	or <b>ERROR:&lt;errorcode&gt;</b> (please see section 4).
<i>Use on</i>	Prompt
All Devices	<b>PowerDesc:&lt;NodeID&gt;,&lt;errorcode&gt;[,&lt;PowerDescriptor&gt;]</b>
	In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode
	<NodeID> is the Remote node's NodeID. In addition the <PowerDescriptor> is displayed as a 16 bit hexadecimal number as described in section 2.3.2.4. of the ZigBee Pro Specification.
	After successful transmission, the sequence number of the unicast is stated using the "SEQ:XX" prompt. When acknowledged (or not) the accompanying "ACK:XX" (or "NACK:XX") prompt is displayed.

<b>+ACTEPDESC – Request Node's Active Endpoint List (ZDO)</b>	
Execute Command	Response
<b>AT+ACTEPDESC:&lt;Address&gt;,&lt;NodeID&gt;</b>	<b>SEQ:XX</b>
Where <Address> is the EUI64, NodeID or Address table entry of the node which is to be interrogated about the node with the NodeID specified in <NodeID>.	<b>OK</b>
Sends a unicast to obtain the specified device's active endpoint list.	or <b>ERROR:&lt;errorcode&gt;</b> (please see section 4).
<i>Use on</i>	Prompt
All Devices	<b>ActEpDesc:&lt;NodeID&gt;,&lt;errorcode&gt;[,XX,...]</b>
	<NodeID> is the Remote node's NodeID. In addition all active endpoints are listed as 8-bit hexadecimal numbers separated by commas. In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode
	After successful transmission, the sequence number of the unicast is stated using the "SEQ:XX" prompt. When acknowledged (or not) the accompanying "ACK:XX" (or "NACK:XX") prompt is displayed.

<b>+SIMPLEDESC – Request Endpoint’s Simple Descripto (ZDO)</b>	
Execute Command	Response
<b>AT+SIMPLEDESC:&lt;Address&gt;,&lt;NodeID&gt;,&lt;XX&gt;</b>	<b>SEQ:XX</b>
Where <Address> is the EUI64, NodeID or Address table entry of the node which is to be interrogated about the node with the NodeID specified in <NodeID> and XX is the number of the endpoint, which simple descriptor is to be read. XX should be hexadecimal number.	<b>OK</b>
Sends a unicast to obtain the specified device’s active endpoint list.	or <b>ERROR:&lt;errorcode&gt;</b>
<i>Use on</i>	<b>&lt;errorcode&gt;</b> please see section 4.
All Devices	Prompt <b>SimpleDesc:&lt;NodeID&gt;,&lt;errorcode&gt;</b> <b>EP:XX</b> <b>ProfileID:XXXX</b> <b>DeviceID:XXXXvXX</b> <b>InCluster:&lt;Cluster List&gt;</b> <b>OutCluster:&lt;Cluster List&gt;</b>
	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.
	<NodeID> is the Remote node’s NodeID. In addition all active endpoints are listed as 8 bit hexadecimal numbers separated by commas.
	After successful transmission the sequence number of the unicast is stated using the “SEQ:XX” prompt. When acknowledged (or not) the accompanying “ACK:XX” (or “NACK:XX”) prompt is displayed.



<b>+MATCHREQ – Find Nodes which Match a Specific Descriptor (ZDO)</b>	
Execute Command	Response
<b>AT+MATCHREQ:&lt;ProfileID&gt;,&lt;NumInClusters&gt;[,&lt;InClusterList&gt;],&lt;NumOutClusters&gt;[,&lt;OutClusterList&gt;]</b>	<b>OK</b>  or <b>ERROR:&lt;errorcode&gt;</b>
<p>Where &lt;ProfileID&gt; Required profile ID of the device being searched for followed by a specification of required input and output clusters. If a remote node has a matching ProfileID and matches at least one of the specified clusters it will respond to this broadcast listing the matching endpoint(s).</p> <p><b>&lt;NumInClusters&gt;</b> and <b>&lt;NumOutClusters&gt;</b> must be 2 hexadecimal digits</p> <p><i>Example</i></p> <p><b>AT+MATCHREQ:0109,01,0000,02,0700,0701</b></p> <p><i>Use on</i></p> <p>All Devices</p>	<p><b>&lt;errorcode&gt;</b> please see section 4.</p> <p>Prompt</p> <p><b>MatchDesc:&lt;NodeID&gt;,&lt;errorcode&gt;,XX,...</b></p> <p>In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.</p> <p>Where &lt;NodeID&gt; is the Remote node's NodeID. In addition all endpoints of this node matching the search criterion are listed as 8 bit hexadecimal numbers separated by commas.</p>

<b>+ANNCE – Announce Local Device In The Network (ZDO)</b>	
Execute Command	Response
<b>AT+ANNCE</b>	<b>OK</b>  or <b>ERROR&lt;errorcode&gt;</b>  <b>&lt;errorcode&gt;</b> please see in section 4.
<p>Send a ZigBee device announcement. Broadcast announcing the local node on the network.</p> <p><i>Use on</i></p> <p>All Devices</p>	
Remote Action	<p>Prompt</p> <p><b>FFD:&lt;EUI64&gt;,&lt;NodeID&gt;</b></p> <p>The prompt above will be displayed on all nodes which can hear the announcement.</p>

### **+KEYUPD – Update the Network Key**

Execute Command

#### **AT+KEYUPD**

Updates the Network Key with a new random key.

*Use on*

Trust Centre

*Note*

Can only be used on the Trust Centre

Response

**OK**

or **ERROR<errorcode>**

**<errorcode>** please check section 4.

Prompt:

NWKKEYUPDATED (if the key is updated successfully)

### **+CCHANGE – Change the network's channel**

Execute Command

#### **AT+CCHANGE[:channel]**

Ask all nodes in the network to change their channel. Please use decimal number 11-26 to specify the channel. If no channel is specified a random channel out of the channels masked in S00 is picked which wasn't previously blacklisted because of excessive packet loss (**NM:ES REPORT WARNING** prompt)

*Use on*

Network Manager

*Note*

The New channel needs to be masked in S00 for all nodes on the network. Ideally S00 should be identical for all nodes on a network.

Response

**OK**

or

**ERROR<errorcode>**

**<errorcode>** please see in section 4.

Parameters

Optional channel ranging from 11 to 26.

---

**+CSWITCH – Switch the Network’s Channel without Notifying Other Nodes**

Execute Command

**AT+CSWITCH:<channel>**
**<channel>**:decimal number range from 11 to 26

This command can be used to switch channel without notifying all nodes on the network. Usage scenario can be that the CICIE wants to move to a channel to make a ZLL light join it then bring the ZLL light back to its original channel.

The user need to use AT+CSWITCH:<The channel where the ZLL light is using>

After 5 seconds, do AT+N to check if it has successfully moved to the target channel

Then apply AT+INTERPAN command.

After the ZLL light joins it, the CICIE should use AT+CCHANGE:<CICIE’s original channel>

*Use on*

Network Manager

Response

**OK**

or

**ERROR<errorcode>**
**<errorcode>** please see in section 4.

### 3.5.1.3 Messaging Commands

#### +ATABLE – Display Address Table

Read Command

**AT+ATABLE**

*Use on*

All Devices

*Notes*

The address table is volatile and its contents are lost if the device is powered down.

Response

No. | Active | ID | EUI

00 | N | 0000 | 000D6F0000012345  
(...)

OK

The Address Table contains nodes which can be addressed by referring to the corresponding address table entry. The “Active” column shows nodes to which a message is currently in flight.

#### +ASET – Set Address Table Entry

Read Command

**AT+ASET:XX,<NodeID>,<EUI64>**

Where XX is the entry number of the address table entry which is to be written. If the NodeID is unknown, the NodeID must be substituted with “FFFF”.

*Use on*

All Devices

Response

OK

or **ERROR:<errorcode>**

<errorcode> represents the error code in section 4.

**+LBTABLE – Display Local Binding Table**

Read Command

Response

**AT+LBTABLE**
*Use on*

All Devices

The binding table is cleared  
by a reset

No.	Type	Active	LocalEP	ClusterID	Addr	
RemEP						
10.	Ucast	No	01	DEAD	1234567887654321	01
11.	MTO	No	01	DEAD	E012345678876543	88
12.	Mcast	No	01	DEAD	CDAB	
13.	Unused					
14.	Unused					
15.	Unused					
16.	Unused					
17.	Unused					
18.	Unused					
19.	Unused					

*Note:*

When using this command,  
local display binding index  
starts from 10.

### **+BSET – Set local Binding Table Entry**

Read Command	Response
<p><b>AT+BSET:&lt;type&gt;,&lt;LocalEP&gt;,&lt;ClusterID&gt;,&lt;DstAddress&gt;[,&lt;DstEP&gt;]</b></p> <p>Where:</p> <p><b>&lt;Type&gt;</b> is the type of binding:</p> <p><i>1= Unicast Binding with EUI64 and remote EP specified</i>  <i>2= Many to one Binding with EUI64 and remote EP Specified</i>  <i>3= Multicast Binding with Multicast ID Specified</i></p> <p><b>&lt;LocalEP&gt;</b> is the local endpoint</p> <p><b>&lt;ClusterID&gt;</b> is the t cluster ID, Address is either the EUI64 of the target device, or a multicast ID</p> <p><b>&lt;DstEP&gt;</b> the remote endpoint which is not specified in case of a multicast binding.</p> <p>The new binding is created in the next available free binding table entry.</p> <p><i>Use on</i></p> <p>All Devices</p>	<p><b>OK</b></p> <p>Or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> represents the error code explained in section 4.</p>

### **+BCLR – Clear local Binding Table Entry**

Read Command	Response
<p><b>AT+BCLR:XX</b></p> <p>Where <b>XX</b> is the entry number of the binding table which is to be cleared.</p> <p><i>Use on</i></p> <p>All Devices</p>	<p><b>OK</b></p> <p>or <b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> is explained in section 4.</p>

+BTABLE – Display Binding Table (ZDO)																									
Read Command	Response																								
<b>AT+BTABLE:XX,&lt;address&gt;</b>	<b>SEQ:XX</b>																								
Where XX is the start index of the remote binding table and <address> can be the remote node's EUI64, Node ID or address/binding table entry.	<b>OK or</b> <b>ERROR&lt;errorcode&gt;</b>																								
<i>Use on</i>	This command requests the target node to respond by listing its binding table starting from the requested index.																								
All devices	The response indicates success or failure in sending this message. The acknowledgement as well as the actual response to this request will follow as asynchronous prompts.																								
<i>Note:</i>																									
Also the local node can be the target of this command (e.g. use address table entry FF as the address)																									
	Example																								
	<b>AT+BTABLE:00,0000</b>																								
	<b>SEQ:01</b>																								
	<b>OK</b>																								
	<b>BTable:0000,00</b> <b>Length:03</b>																								
	<table><tr><td>No.</td><td>SrcAddr</td><td>SrcEP</td><td>ClusterID</td><td>DstAddr</td><td>DstEP</td></tr><tr><td>00.</td><td>000D6F000059474E</td><td>01</td><td>DEAD</td><td>1234567887654321</td><td>12</td></tr><tr><td>01.</td><td>000D6F000059474E</td><td>01</td><td>DEAD</td><td>E012345678876543</td><td>E0</td></tr><tr><td>02.</td><td>000D6F000059474E</td><td>01</td><td>DEAD</td><td>ABCD</td><td></td></tr></table>	No.	SrcAddr	SrcEP	ClusterID	DstAddr	DstEP	00.	000D6F000059474E	01	DEAD	1234567887654321	12	01.	000D6F000059474E	01	DEAD	E012345678876543	E0	02.	000D6F000059474E	01	DEAD	ABCD	
No.	SrcAddr	SrcEP	ClusterID	DstAddr	DstEP																				
00.	000D6F000059474E	01	DEAD	1234567887654321	12																				
01.	000D6F000059474E	01	DEAD	E012345678876543	E0																				
02.	000D6F000059474E	01	DEAD	ABCD																					
	<b>ACK:01</b>																								
	In this example the neighbour table of the remote node with the short ID shown in <NodeID> contains three entries (hexadecimal), which are displayed. In case the table contains more than three entries it may be required to repeat this command and increase the index count until the full table is derived.																								
	In case of an error, an errorcode other than 00 will be displayed and the prompt will end after the errorcode.																								

<b>+BIND – Create Binding on Remote Device (ZDO)</b>	
Read Command	Response
<b>AT+BIND:&lt;address&gt;,&lt;type&gt;,&lt;SrcAddress&gt;,&lt;SrcEP&gt;,&lt;ClusterID&gt;,&lt;DstAddress&gt; [,&lt;DstEP&gt;]</b>	<b>SEQ:XX</b>
Create Binding on a remote device with	<b>OK</b>
<b>&lt;address&gt;</b> the target Node's EUI64, NodeID, or Address Table entry	or <b>ERROR:&lt;errorcode&gt;</b>
<b>&lt;type&gt;</b> the Addressing mode shown as blow	The response indicates success or failure in sending this message. The acknowledgement as well as the actual response to this request will follow as asynchronous prompts.
<b>&lt;SrcAddress&gt;</b> The EUI64 of the Source	Prompt
<b>&lt;SrcEP&gt;</b> The source Endpoint	<b>Bind:&lt;NodeID&gt;,&lt;status&gt;</b>
<b>&lt;ClusterID&gt;</b> The Cluster ID on the source Device	In case of an error an status other than 00 will be displayed
<b>&lt;DstAddress&gt;</b> The EUI64 or 16-bit multicast ID, depending on <type>	<b>&lt;NodeID&gt;</b> is the Remote node's Node ID.
<b>&lt;DstEP&gt;</b> Only in Mode 2: The destination endpoint	As with all unicasts after successful transmission the sequence number of the unicast is stated using the "SEQ:XX" prompt. When acknowledged (or not) the accompanying "ACK:XX" (or "NACK:XX") prompt is displayed.
<i>Type:</i>	Example
<i>1= Multicast Binding with Multicast ID Specified in &lt;DstAddress&gt;</i>	<b>AT+BIND:0000,3,000d6f000059474e,01,abcd,000D6F0000123456,01</b>
<i>3= Unicast Binding with destination EUI64 in &lt;DstAddress&gt; and and destination EP in &lt;DstEP&gt;</i>	<b>SEQ:01</b>
<i>Note:</i>	<b>OK</b>
Also the local node can be the target of this command (e.g. use address table entry FF as the address)	<b>Bind:0000,00</b>
<i>Use on</i>	<b>ACK:01</b>
All devices	



<b>+UNBIND – Delete Binding on Remote Device (ZDO)</b>	
Read Command	Response
<b>AT+UNBIND:&lt;address&gt;,&lt;type&gt;,&lt;SrcAddress&gt;,&lt;SrcEP&gt;,&lt;ClusterID&gt;,&lt;DstAddress&gt;[,&lt;DstEP&gt;]</b>	<b>SEQ:XX</b>
Delete Binding on a remote device with	<b>OK</b>
<address> the target Node's EUI64, Node ID, or Address Table entry	or <b>ERROR:&lt;errorcode&gt;</b>
<type> the addressing mode as shown below	The response indicates success or failure in sending this message. The acknowledgement as well as the actual response to this request will follow as asynchronous prompts.
<SrcAddress> The EUI64 of the Source	Prompt
<SrcEP> The source Endpoint	<b>Unbind:&lt;NodeID&gt;,&lt;status&gt;</b>
<ClusterID> The Cluster ID on the source Device	In case of an error an status other than 00 will be displayed
<DstAddress> The EUI64 or 16-bit multicast ID, depending on <type>	<b>&lt;NodeID&gt;</b> is the Remote node's NodeID.
<DstEP> Only in Mode 2: The destination endpoint	As with all unicasts after successful transmission the sequence number of the unicast is stated using the "SEQ:XX" prompt. When acknowledged (or not) the accompanying "ACK:XX" (or "NACK:XX") prompt is displayed.
Types:	Example
1= Multicast Binding with Multicast ID Specified in <DstAddress>	<b>AT+UNBIND:0000,3,000d6f000059474e,01,abcd,000D6F0000123456,01</b>
3= Unicast Binding with destination EUI64 in <DstAddress> and destination EP in <DstEP>	<b>SEQ:01</b>
Use on	<b>OK</b>
All devices	<b>Unbind:0000,00</b>
Note:	<b>ACK:01</b>
Also the local node can be the target of this command (e.g. use address table entry FF as the address)	

### **+EBIND – End Device Bind**

Execute Command	Response
<b>AT+EBIND:&lt;EP&gt;</b>  <b>&lt;EP&gt;</b> - Local Endpoint which will initiate end device binding.  <i>Notes</i>  Only used on the local node.	<b>EBINDACK</b> (meaning the request is acknowledged by coo)  <b>EBINDRSP: &lt;Status&gt;</b>  <b>OK</b>  If status is not 00, it is an error code (please check section 4)

### **3.5.1.4 ZigBee Discovery Commands**

#### **+DISCOVER – Discover HA Devices On The HAN**

Execute Command	Response
<b>AT+DISCOVER:&lt;Cluster ID&gt;</b>  <b>&lt;Cluster ID&gt;</b> - 16 bit hexadecimal cluster ID (please see section 2.2.2). The Combined Interface can search for HA devices based on a specified cluster ID.  <i>Notes:</i>  Response shall show up all HA devices that match. Discovery will only search for server cluster.	<b>OK</b>  or <b>ERROR:&lt;errorcode&gt;</b> (Please check section 4)  Prompt:  <b>DEV:&lt;NodeID&gt;,&lt;EndPoint&gt;</b>  Carry out the ZigBee Service Discovery to find HA devices that support the given match criteria.  <b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number which represents network address of found device.  <b>&lt;EndPoint&gt;</b> - 8 bit hexadecimal number representing endpoint on found remote device supporting the given match criteria.

### **+DISCOVERP – Discover Devices Using Profile ID and Cluster ID**

Execute Command	Response
<b>AT+DISCOVERP:[&lt;ProfileID&gt;],&lt;ClusterID&gt;[,&lt;Option&gt;]</b>  <b>&lt;ProfileID&gt;</b> - 16 bit hexadecimal number which represents the profile ID, by default the profile ID is 0x0104 (HA profile).  <b>&lt;ClusterID&gt;</b> - 16 bit hexadecimal number which represents the cluster ID listed in section 2.2.2, The Combined Interface can search for HA devices based on the specified cluster ID.  <b>&lt;Option&gt;</b> - a Boolean type, 0 means discover server cluster, 1 means discover client cluster.	<b>OK</b>  or <b>ERROR:&lt;errorcode&gt;</b> (Please check section 4)  Prompt:  <b>DEV:&lt;NodeID&gt;,&lt;EndPoint&gt;</b>  Carry out the ZigBee Service Discovery to find ZigBee HA devices that support the given match criteria.  <b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number which represents network address of found device.  <b>&lt;EndPoint&gt;</b> - 8 bit hexadecimal number representing endpoint on found remote device supporting the given match criteria.

### **+CLUSDISC – Find All Supported Clusters On A Remote Device End Point**

Execute Command	Response
<b>AT+CLUSDISC:&lt;Node ID&gt;,&lt;EndPoint&gt;</b>  <b>&lt;Node ID&gt;</b> - 16 bit hexadecimal number. The network address of the target device.  <b>&lt;EndPoint&gt;</b> - 8 bit hexadecimal number. The end point of the target device.  <i>Notes:</i>  Response show up all supported clusters on specified device's endpoint.	<b>OK</b>  <b>DISCCLUS:&lt;Node ID&gt;,&lt;Status&gt;,&lt;EndPoint&gt;</b>  or <b>ERROR:&lt;errorcode&gt;</b> (see section 4)  Prompt (will show, if <Status> is 00)  <b>SERVER:&lt;ClusterID&gt;,&lt;ClusterID&gt;,&lt;ClusterID&gt;</b> <b>CLIENT:&lt;ClusterID&gt;,&lt;ClusterID&gt;,&lt;ClusterID&gt;</b>

**+ATTRDISC - Find Supported Attributes On A Remote Device End Point**

Execute Command	Response
<b>AT+ATTRDISC:&lt;NodeID&gt;,&lt;EP&gt;,&lt;ClusterID&gt;,&lt;AttributeID&gt;,&lt;MaxNumofAttr&gt;</b>  <b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number represents target device's network address  <b>&lt;EP&gt;</b> - 8 bit hexadecimal number represents target device's end point  <b>&lt;ClusterID&gt;</b> - 16 bit hexadecimal number  <b>&lt;AttributeID&gt;</b> -16 bit hexadecimal number the (discover will start from this attribute)  <b>&lt;MaxNumofAttr&gt;</b> - decimal number represent the number of attributes that required to be discovered, e.g: 01, 10	<b>OK</b>  <b>DISCATTR:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Complete code&gt;</b>  <b>CLUS:&lt;ClusterID&gt;,ATTR:&lt;AttributeID&gt;,TYPE:&lt;DataType&gt;</b>  or <b>ERROR:&lt;errorcode&gt;</b> (see section 4)  Note:  <b>&lt;Complete code&gt;</b> represents if all attributes support by this cluster have been discovered.  00 – Completed discovery  01 – Uncompleted discovery  If the CICIE receives attribute discovery response, a prompt “ <b>ENDDISCATTR</b> ” will show at the end of the response.

**+MATTRDISC - Find Supported Manufacturer Defined Attributes On A Remote Device**

Execute Command

**AT+MATTRDISC:<NodeID>,<EP>,<Mcode>,<ClusterID>,<AttributeID>,<MaxNumofAttr>[,<ProfileID>]**

**<NodeID>** - 16 bit hexadecimal number represents target device's network address

**<EP>** - 8 bit hexadecimal number represents target device's end point

**<Mcode>** - 16 bit hexadecimal number, which is manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number

**<AttributeID>** -16 bit hexadecimal number the (discover will start from this attribute)

**<MaxNumofAttr>** - decimal number represent the number of attributes that required to be discovered, e.g: 01, 10

**<ProfileID>** 16 bit hexadecimal number, representing profile ID. If the manufacturer specified attributes are defined under a new profile ID, the user need to specify this parameter. If this parameter is omitted, Home Automation Profile ID (0x0104) will be used.

Response

**OK**

**DISCMATTR:<NodeID>,<EP>,<Mcode>,<Complete code>**

**CLUS:<ClusterID>,ATTR:<AttributeID>,TYPE:<DataType>**

or

**ERROR:<errorcode>** (see section 4)

Note:

**<Complete code>** represents if all attributes support by this cluster have been discovered.

00 – Completed discovery

01 – Uncompleted discovery

If the CICIE receives attribute discovery response, a prompt "**ENDDISCMATTR**" will show at the end of the response.

### 3.5.1.5 ZCL Attributes commands

<b>+READATR – Read Server Attribute</b>	
Execute Command	Response
<p><b>AT+READATR:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,...&lt;AttrID&gt;</b></p> <p><b>AT+READATR:,,,&lt;Cluster&gt;,&lt;AttrID&gt;,...&lt;AttrID&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group</p> <p><b>&lt;ClusterID&gt;</b> 16 bit hexadecimal number which represents cluster ID, see section 2.2.2</p> <p><b>&lt;AttrID&gt;</b> - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.</p> <p><i>Note:</i></p> <p>This command can be used to read multiple attributes (up to 16 in a cluster)</p> <p>If the address is a node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and sendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p><b>RESPATTR:&lt;NodeID&gt;,&lt;EP&gt;,&lt;ClusterID&gt;,&lt;AttrID&gt;,&lt;Status&gt;,&lt;AttrInfo&gt;</b></p> <p>or <b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number. It is the source Node ID of response.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, the source endpoint of the response.</p> <p><b>&lt;ClusterID&gt;</b> - cluster ID, 16 bit hexadecimal number, see section 2.2.2</p> <p><b>&lt;AttrID&gt;</b>: attribute ID 16 bit hexadecimal number</p> <p><b>&lt;Status&gt;</b> - 8 bit hexadecimal number which indicates the result of the requested operation.</p> <p><b>&lt;AttrInfo&gt;</b> - hexadecimal number of char string (size depends on the attribute requested). &lt;AttrInfo&gt; shall only be valid if &lt;Status&gt; = 0x00. If &lt;Status&gt; indicates error, &lt;AttrInfo&gt; is not returned.</p> <p><b>&lt;errorcode&gt;</b> represents the error code (section 0).</p> <p>Use this command to get the requested attribute from a cluster server on a remote HA device which supports this server cluster.</p> <p>Please Note:</p> <p>If the attribute is in Temperature measurement cluster or Illuminance measurement cluster:</p> <p><b>ILLUMINANCE</b> and <b>TEMPERATURE</b> prompts will show respectively and take place of</p>

**RESPATTR.** In the response, cluster ID is omitted. For example:

ILLUMINANCE:38C8,03,0000,00,07AD

TEMPERATURE:38C8,04,0000,00,00CD

### **+READCATR – Read Client Attribute**

Execute Command

**AT+READCATR:<Address>,<EP>,<SendMode>,<ClusterID>,<AttrID>,...<AttrID>**

**AT+READCATR:,,,<Cluster>,<AttrID>,...<AttrID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<ClusterID>** 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.

*Note:*

This command can be used to read multiple attributes (up to 16 in a cluster)

If the address is a node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and sendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**RESPATTR:<NodeID>,<EP>,<ClusterID>,<AttrID>,<Status>,<AttrInfo>**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

**<AttrInfo>** - hexadecimal number of char string (size depends on the attribute requested). <AttrInfo> shall only be valid if <Status> = 0x00. If <Status> indicates error, <AttrInfo> is not returned.

**<errorcode>** represents the error code (section 0).

Use this command to get the requested attribute from a cluster server on a remote HA device which supports this server cluster.



### **+READMATR – Read Manufacturer Specific Server Attribute**

Execute Command

**AT+READMATR:<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<AttrID>,...<AttrID>**

**AT+READMATR:,,,<Mcode>,<ClusterID>,<AttrID>,...<AttrID>**

**<Address>** - 16 bit hexadecimal number. It shall be The Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to manufacture definition.

*Note:*

This command can be used to read multiple attributes (up to 16 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**RESPMATR:<NodeID>,<EP>,<Mcode>,<ClusterID>,<AttrID>,<Status>,<AttrInfo>**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacture code.

**<ClusterID>** - 16 bit hexadecimal cluster ID

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

**<AttrInfo>** - hexadecimal number of char string (size depends on the attribute requested). <AttrInfo> shall only be valid if <Status> = 0x00. If <Status> indicates error, <AttrInfo> is not returned.

**<errorcode>** represents the error code (section 0).

Use this command to get the requested attribute from a cluster server on a remote HA device supporting this server cluster.

### **+READMCATR – Read Manufacturer Specific Client Attribute**

Execute Command

**AT+READMCATR:**<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<AttrID>,...<AttrID>

**AT+READMCATR:**,,,,<Mcode>,<ClusterID>,<AttrID>,... <AttrID>

**<Address>** - 16 bit hexadecimal number. It shall be The Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to manufacture definition.

*Note:*

This command can be used to read multiple attributes (up to 16 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and sendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**RESPMATTR:**<NodeID>,<EP>,<Mcode>,<ClusterID>,<AttrID>,<Status>,<AttrInfo>

or **ERROR:**<errorcode>

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacture code.

**<ClusterID>** - 16 bit hexadecimal cluster ID

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

**<AttrInfo>** - hexadecimal number of char string (size depends on the attribute requested). <AttrInfo> shall only be valid if <Status> = 0x00. If <Status> indicates error, <AttrInfo> is not returned.

**<errorcode>** represents the error code (section 0).

Use this command to get the requested attribute from a cluster server on a remote HA device supporting this server cluster.

### AT+WRITEATR – Write Server Attribute

Execute Command

**AT+WRITEATR:<Address>,<EP>,<SendMode>,<ClusterID>,<AttrID>,<DataType>,<Data>**

**AT+WRITEATR:,,,<ClusterID>,<AttrID>,<DataType>,<AttrValue>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.

**<DataType>** - 8 bit hexadecimal number that represents the type of the data accepted by this Attribute (please check HA specification)

**<AttrValue>** - If attribute value has an integer type this field shall contain hexadecimal representation in big-endian format. If attribute value has a string type this field contains sequence of characters.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**WRITEATTR:<NodeID>,<EP>,<ClusterID>,<AttrID>,<Status>**

or

**ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If < Status > is not 00, it will be an errorcode which is listed in section 0.

Please note: if the user uses this command to write multiple attributes and the attributes are written successfully, then the response will only contain status code. For example:

**WRITEATTR:1234,01,1010,00**

This is expected behaviour according to ZigBee cluster library specification.

Use this command to set value to a writeable attribute on a remote HA device supporting this cluster.

**AT+WRITECATR – Write Client Attribute**

Execute Command

**AT+WRITECATR:<Address>,<EP>,<SendMode>,<ClusterID>,<AttrID>,<DataType>,<Data>**
**AT+WRITECATR:,,,,<ClusterID>,<AttrID>,<DataType>,<AttrValue>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.

**<DataType>** - 8 bit hexadecimal number that represents the type of the data accepted by this Attribute (please check HA specification)

**<AttrValue>** - If attribute value has an integer type this field shall contain hexadecimal representation in big-endian format. If attribute value has a string type this field contains sequence of characters.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**
**WRITEATTR:<NodeID>,<EP>,<ClusterID>,<AttrID>,<Status>**

or

**ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If < Status > is not 00, it will be an errorcode which is listed in section 0.

Please note: if the user uses this command to write multiple attributes and the attributes are written successfully, then the response will only contain status code. For example:

**WRITEATTR:1234,01,1010,00**

This is expected behaviour according to ZigBee cluster library specification.

Use this command to set value to a writeable attribute on a remote HA device supporting this cluster.

### AT+WRITEMATR – Write Manufacturer Specific Server Attribute

Execute Command

**AT+WRITEMATR:**<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<AttrID>,<DataType>,<Data>

**AT+WRITEMATR:**,,,<Mcode>,<ClusterID>, <AttrID>,<DataType>,<AttrValue>

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according manufacture definition.

**<DataType>** - 8 bit hexadecimal number that represents the type of the data accepted by this Attribute (Depend on the manufacture definition)

**<AttrValue>** - If attribute value has an integer type this field shall contain hexadecimal representation in big-endian format. If attribute value has a string type this field contains a sequence of characters.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**WRITEMATTR:**<NodeID>,<EP>,<Mcode>,<ClusterID>,<AttrID>,<Status>

or **ERROR:**<errorcode>

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacture code.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If < Status > is not 00, it will be an errorcode which is listed in section 0.

Please note: if the user uses this command to write multiple attributes and the attributes are written successfully, then the response will only contain status code. For example:

**WRITEMATTR:1234,01,1010,,00**

This is expected behaviour according to ZigBee cluster library specification.

Use this command to write attribute values to a writeable attribute on a remote HA device supporting this cluster.

**AT+WRITEMCATR – Write Manufacturer Specific Client Attribute**

Execute Command

**AT+WRITEMCATR:<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<AttrID>,<DataType>,<Data>**
**AT+WRITEMCATR:,,,,<Mcode>,<ClusterID>,<AttrID>,<DataType>,<AttrValue>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according manufacture definition.

**<DataType>** - 8 bit hexadecimal number that represents the type of the data accepted by this Attribute (Depend on the manufacture definition)

**<AttrValue>** - If attribute value has an integer type this field shall contain hexadecimal representation in big-endian format. If attribute value has a string type this field contains a sequence of characters.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**
**WRITEMATTR:<NodeID>,<EP>,<Mcode>,<ClusterID>,<AttrID>,<Status>**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacture code.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If < Status > is not 00, it will be an errorcode which is listed in section 0.

Please note: if the user uses this command to write multiple attributes and the attributes are written successfully, then the response will only contain status code. For example:

**WRITEMATTR:1234,01,1010,,00**

This is expected behaviour according to ZigBee cluster library specification.

Use this command to write attribute values to a writeable attribute on a remote HA device supporting this cluster.



**+READRPTCFG – Read Reporting Configuration From Remote Node**

Execute Command

**AT+READRPTCFG:<Address>,<EP>,<SendMode>,<ClusterID>,<Direction1>,<AttrID1>,...,<Direction16>,<AttrID16>**

**AT+READRPTCFG:,,,,<ClusterID>,<Direction1>,<AttrID1>,...,<Direction16>,<AttrID16>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group IDd if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<Direction>** - a Boolean type specifies whether values of the attribute are reported (0), or whether reports of the attribute are received (1).

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.

*Note:*

This command can be used to read multiple attributes (up to 16 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and sendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance.

Response

**OK**

**READRPTCFGRSP:<NodeID>,<EP>,<ClusterID>,<Status>,<Direction>,<AttrID>,[<DataType>,<MinimumReportingInterval>,<MaximumReportingInterval>,<ReportableChange>][<Timeout>]**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

**<Direction>** - 8 bit hexadecimal number specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01)

**<AttrID>**: attribute ID 16 bit hexadecimal number

If received <Direction> field is 0x00, the following information is expected:

**<DataType>** - 8 bit hexadecimal number which represents the attribute data type

**<MinimumReportingInterval>** - The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports for the attribute specified in the attribute Identifier field. If the minimum reporting interval has not been configured, this field shall contain the value 0xffff.

**<MaximumReportingInterval>** - The maximum reporting interval field is 16-bits in length and shall contain the maximum interval, in seconds,

Thereafter it will not be necessary to fill in the <address> parameter.

between issuing reports for the attribute specified in the attribute Identifier field. If the maximum reporting interval has not been configured, this field shall contain the value 0xffff.

**<ReportableChange>** - The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. For attributes with 'analog' data type (see ZigBee Cluster Library 075123r03) the field has the same data type as the attribute. If the reportable change has not been configured, this field shall contain the invalid value for the relevant data type.

For attributes of 'discrete' data type (see ZigBee Cluster Library) this field is omitted.

If received <Direction> field is 0x01, the following information is expected:

**<Timeout>** - The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute identifier field. If the timeout period has not been configured, this field shall contain the value 0xffff.

**<errorcode>** please see section 0.



### **+READMRPTCFG – Read Manufacture Reporting Configuration From Remote Node**

Execute Command

**AT+READMRPTCFG:<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<Direction1>,<AttrID1>,...,<Direction16>,<AttrID16>**

**AT+READMRPTCFG:,,,,<Mcode>,<ClusterID>,<Direction1>,<AttrID1>,...,<DirectionX>,<AttrIDX>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<Direction>** - A Boolean type specifies whether values of the attribute are reported (0), or whether reports of the attribute are received (1).

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to manufacture definition.

*Note:*

This command can be used to read multiple attributes (up to 16 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a

Response

**OK**

**READMRPTCFGRSP:<NodeID>,<EP>,<Mcode>,<ClusterID>,<Status>,<Direction>,<AttrID>,[<DataType>,<MinimumReportingInterval>,<MaximumReportingInterval>,<ReportableChange>][<Timeout>]**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

**<Direction>** - 8 bit hexadecimal number specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01)

**<AttrID>**: attribute ID 16 bit hexadecimal number

If received <Direction> field is 0x00, the following information is expected:

**<DataType>** - 8 bit hexadecimal number which represents the attribute data type.

**<MinimumReportingInterval>** - The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports for the attribute specified in the attribute Identifier field. If the minimum reporting interval has not been configured, this field shall contain the value 0xffff.

destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

**<MaximumReportingInterval>** - The maximum reporting interval field is 16-bits in length and shall contain the maximum interval, in seconds, between issuing reports for the attribute specified in the attribute Identifier field. If the maximum reporting interval has not been configured, this field shall contain the value 0xffff.

**<ReportableChange>** - The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. For attributes with 'analog' data type (see ZigBee Cluster Library 075123r03) the field has the same data type as the attribute. If the reportable change has not been configured, this field shall contain the invalid value for the relevant data type.

For attributes of 'discrete' data type (see ZigBee Cluster Library) this field is omitted.

If received <Direction> field is 0x01, the following information is expected:

**<Timeout>** - The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute identifier field. If the timeout period has not been configured, this field shall contain the value 0xffff.

**<errorcode>** please see section 0.

### **+CFGRPT – Configure Attribute Reporting**

Execute Command

**AT+CFGRPT:**<Address>,<EP>,<SendMode>,<ClusterID>,<Direction>,<AttrID>,[<DataType>,<MinimumReportingInterval>,<MaximumReportingInterval>,<ReportableChange>][<Timeout>]

**AT+CFGRPT:**,,,<ClusterID>,<Direction>,<AttrID>,[<DataType>,<MinimumReportingInterval>,<MaximumReportingInterval>,<ReportableChange>][<Timeout>]

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<Direction>** - a Boolean type specifies whether values of the attribute are reported (0), or whether reports of the attribute are received (1).

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to HA and ZigBee Cluster library.

If <Direction> field is set to 0, the following information need to be field:

**<DataType>** - 8 bit hexadecimal number which represents the attribute data type

**<MinimumReportingInterval>** - The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports for the attribute specified in the attribute Identifier field.

**<MaximumReportingInterval>** - The maximum reporting interval field is 16-bits in length and

Response

**OK**

**CFGRPTRSP:**<NodeID>,<EP>,<ClusterID>,<Status>[,<Direction>,<AttrID>]

or **ERROR:**<errorcode>

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<ClusterID>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If **<Status>** is not 00, the prompt information will include: <Direction> and <AttrID>

**<Direction>** - a Boolean type specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01).

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<errorcode>** represents the error code (please see section 0).

Use this command to configure attribute reporting on a remote HA device which supports the corresponding cluster and attribute.

shall contain the maximum interval, in seconds, between issuing reports for the attribute specified in the attribute Identifier field.

**<ReportableChange>** - The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. For attributes with 'analog' data type (see ZigBee Cluster Library 075123r03) the field has the same data type as the attribute.

If received <Direction> field is 1, the following information is expected:

**<Timeout>** - The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute Identifier field.

*Note:*

This command can be used to configure multiple attributes reporting (up to 4 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (", " cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

## +CFGMRPT – Configure Manufacture Defined Attribute Reporting

Execute Command

**AT+CFGMRPT:**<Address>,<EP>,<SendMode>,<Mcode>,<ClusterID>,<Direction>,<AttrID>,<[DataType]>,<MinimumReportingInterval>,<MaximumReportingInterval>,<ReportableChange>][<Timeout>]

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<Direction>** - a Boolean type specifies whether values of the attribute are reported (0), or whether reports of the attribute are received (1).

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to manufacture definition.

If <Direction> field is set to 0, the following information need to be filled:

**<DataType>** - 8 bit hexadecimal number which represents the attribute data type

**<MinimumReportingInterval>** - The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports for the attribute specified in the attribute identifier field.

**<MaximumReportingInterval>** - The maximum reporting interval field is 16-bits in length and shall contain the maximum interval, in seconds,

Response

OK

**CFGMRPTRSP:**<NodeID>,<EP>,<Cluster>,<Mcode>,<Status>[,<Direction>,<AttrID>]

or **ERROR:**<errorcode>

**<NodeID>** - 16 bit hexadecimal number. It is the source Node ID of response.

**<EP>** - 8 bit hexadecimal number, the source endpoint of the response.

**<Mcode>** - 16 bit hexadecimal manufacturer code.

**<Cluster>** - cluster ID, 16 bit hexadecimal number, see section 2.2.2

**<Status>** - 8 bit hexadecimal number which indicates the result of the requested operation.

If **<Status>** is not 00, the prompt information will include: <Direction> and <AttrID>

**<Direction>** - a Boolean type specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01).

**<AttrID>**: attribute ID 16 bit hexadecimal number

**<errorcode>** please see section 0.

Use this command to configure reporting for manufacture defined attribute on a remote HA device which supports the corresponding cluster and attribute.

between issuing reports for the attribute specified in the attribute identifier field.

**<ReportableChange>** - The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. For attributes with 'analog' data type (see ZigBee Cluster Library 075123r03) the field has the same data type as the attribute.

If received <Direction> field is 1, the following information is expected:

**<Timeout>** - The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute identifier field.

*Note:*

This command can be used to configure multiple attributes reporting (up to 4 in a cluster)

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (", " cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

### **+SETATR – Set Value Of A Local ZCL Attribute**

Execute Command

**AT+SETATR:<ClusterID>,<AttrID>,<AttrValue>**

**<ClusterID>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID (Please refer to ZigBee Home Automation specification).

**<AttrValue>** - If attribute value has an integer type this field shall be in big-endian format. If attribute value has a string type, this field contains sequence of characters. Maximum length is 110 characters.

Notes: The values of local time and standard time attribute in time cluster cannot be set using this command.

Response

**OK**

or **ERROR:<errorcode>**

**<errorcode>** represents the error code (section 0).

Use this command to set a new value to attribute from a cluster on a local device supporting this cluster.

### **+GETATR – Get value of a local ZCL attribute**

Execute Command

**AT+GETATR:<Cluster>,<AttrID>**

**<Cluster>** - 16 bit hexadecimal number which represents cluster ID, see section 2.2.2

**<AttrID>** - 16 bit hexadecimal number which represents attribute ID according to ZigBee Home Automation specification.

Response

**ATTR:<AttrValue>**

**OK**

or **ERROR:<errorcode>**

**<AttrValue>** - If attribute value has an integer type this field shall contain hexadecimal representation. If attribute value has a string type this field contains sequence of characters. Maximum length is 110 characters.

**<errorcode>** represents the error code (section 0).

Use this command to get a value of an attribute from a cluster server on a local device supporting this cluster.

### 3.5.2 Combined Interface Specified Command Set

#### 3.5.2.1 Default Response to ZCL Command Supported By CI

This section introduces the ZCL commands supported by the combined interface. When the user sends a ZCL command to a target device or group, a default response should be expected if that command does not have a ZCL specified response. In such a case, the prompt will be show as following:

**DFTREP:<NodeID>,<EP>,<ClusterID>,<CMD>,<Status>**

where **<NodeID>** represents the address of the device which sends back the response.

**<EP>** is the endpoint where is the response from.

**<ClusterID>** shows the cluster which the command belongs to.

**<CMD>** is command ID which the default response responds to.

**<Status>** indicate if the command is implemented successfully or not. If this field is not 00, it is an error code, please refer to section 4.

In the following sections, the field of default response will not be repeated.



### 3.5.2.2 Identify Cluster – Client

#### +IDENTIFY – Start or Stop the Target Identifying itself

Execute Command

**AT+IDENTIFY:<Address>,<EP>,<SendMode>,<Time>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the target

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<Time>** - 16 bit hexadecimal number represents the Identification time

*Use on*

All Devices

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

Response

**OK**

**DFTREP:<NodeID>,<EP>,<ClusterID>,<CMD>,<Status>**

or

**ERROR<errorcode>**

**<errorcode>** represents the error code explained in section 4.

**+IDQUERY – Query If Target Device(s) In Identifying Mode**

Execute Command

**AT+IDQUERY:<Address>,<EP>,<SendMode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the target

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

*Use on*

All Devices

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

Response

**OK**
**IDQUERYRSP:<NodeID>,<EP>,<TimeOut>**

or **ERROR<errorcode>**

**<NodeID>** - 16 bit hexadecimal number, source of the response. It should be the same with the target's Node ID.

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the target

**<TimeOut>** - 16 bit hexadecimal number, represents the length of time, in seconds, that the device will continue to identify itself.

**<errorcode>** - represents the error code explained in section 4.

### 3.5.2.3 Group Cluster – Client

<b>+GPADD - Add Group On Target Device</b>	
Execute Command	Response
<b>AT+GPADD:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;GPID&gt;,&lt;GPName&gt;</b>  <b>AT+GPADD:,,,,&lt;GPID&gt;,&lt;GPName&gt;</b>	<b>OK</b>  <b>ADDGPRSP:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Status&gt;,&lt;GPID&gt;</b>
<p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number represent the Endpoint of the target</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group</p> <p><b>&lt;GPID&gt;</b> - 16 bit hexadecimal number, represent the specified group ID</p> <p><b>&lt;GPName&gt;</b> - String up to 16 characters; it can be omitted, if the target does not support group name.</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;Node ID&gt;</b> - 16-bit hexadecimal number represents the NodeID of the node which responds</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number represent the Endpoint of the node which responds</p> <p><b>&lt;Status&gt;</b> - 8-bit hexadecimal number, if it is not 00, it will be a error code. 8A means</p> <p><b>&lt;GPID&gt;</b> - 16 bit hexadecimal number, represent the group ID that will be added on the target device</p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p>

**+GPADIFID - Add Group Membership On Target Devices That Is In Identifying Mode**

Execute Command	Response
<p><b>AT+GPADIFID:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;GPID&gt;,&lt;GPName&gt;</b></p> <p><b>AT+GPADIFID:,,,,&lt;GPID&gt;,&lt;GPName&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number represent the Endpoint of the target</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group</p> <p><b>&lt;GPID&gt;</b> - 16 bit hexadecimal number, represent the specified group ID</p> <p><b>&lt;GPName&gt;</b> - String up to 16 characters; it can be omitted, if the target does not support group name.</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p><b>DFTREP:&lt;NodeID&gt;,&lt;EP&gt;,&lt;ClusterID&gt;,&lt;CMD&gt;,&lt;Status&gt;</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> please see section 4.</p> <p>No ZCL response is defined as this command is expected to be multicast or broadcast.</p>

### + GPVIEW - View Group Information Of Target Devices

Execute Command

**AT+GPVIEW:<Address>,<EP>,<SendMode>,<GPID>**

**AT+GPVIEW:,,, <GPID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**VIEWGPRSP:<NodeID>,<EP>,<Status>,<GPID>,<GPName>]**

or **ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> and <GPName> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<errorcode>** : the code explained in section 4.

### **+GPGET - Get Group Membership Information From Target Devices**

Execute Command

**AT+GPGET:<Address>,<EP>,<SendMode>,<Count>[,<GPList>]**

**AT+GPGET:,,,<Count>[,<GPList>]**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<Count>** - a 8 bit hexadecimal number reflecting the number of groups which is inquired.

**<GPList>** - a series of 16 bit hexadecimal number, represent a list of GPIDs (the number of GPIDs should correspond to the count number)

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**GETGPRSP:<NodeID>,<EP>,<Capacity>,<Count>,<GPID>...**

or

**ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Capacity>** - 8 bit hexadecimal number. The Capacity field shall contain the remaining capacity of the group table of the device. The following values apply:

0 No further groups may be added.

0 < Capacity < 0xfe Capacity holds the number of groups that may be added

0xfe At least 1 further group may be added (exact number is unknown)

0xff It is unknown if any further groups may be added

**<Count>** - 8 bit hexadecimal number. It represents the number of groups contained in the group list field.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<errorcode>** : the code explained in section 4.

**+GPRMV - Remove A Group Membership On Target Device**

Execute Command

**AT+GPRMV:<Address>,<EP>,<SendMode>,<GPID>**
**AT+GPRMV:,,,< GPID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**
**RMVGPRSP:<NodeID>,<EP>,<Status>,<GPID>**

or **ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<errorcode>** : the code explained in section 4.

### **+GPRMALL - Remove All Group Associations On Target Device**

Execute Command

**AT+GPRMALL:<Address>,<EP>,<SendMode>**

**AT+GPRMALL:,,,**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Removing all groups necessitates the removal of all associated scenes as well. (Note: scenes not associated with a group need not be removed).

Response

**OK**

**DFTREP:<NodeID>,<EP>,<ClusterID>,<CMD>,<Status>**

or

**ERROR:<errorcode>** (the error code explained in section 4.



### 3.5.2.4 Scenes Cluster – Client

<b>+SCADD - Add Scenes on Target Devices</b>	
Execute Command	Response
<b>AT+SCADD:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;GPID&gt;,&lt;SCID&gt;,&lt;TransTime&gt;,&lt;SCName&gt;,[Extension field sets]</b>	<b>OK</b>
<b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device. According to ZigBee Cluster Library, The Add Scene command shall be addressed to a single device (not a group).	<b>ADDSC:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Status&gt;[,&lt;GPID&gt;,&lt;SCID&gt;]</b>
<b>&lt;EP&gt;</b> - 16 bit hexadecimal number represent the Endpoint of the target	or <b>ERROR:&lt;errorcode&gt;</b>
<b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group. According to ZigBee Cluster Specification, this command shall be unicast. So SendMode value has been set to 0 for this command.	<b>&lt;NodeID&gt;</b> - 16-bit hexadecimal number represents the Node ID of the node which responds
<b>&lt;GPID&gt;</b> - 16 bit hexadecimal number, represent the specified group ID	<b>&lt;EP&gt;</b> - 8 bit hexadecimal number represent the Endpoint of the node which responds
<b>&lt;SCID&gt;</b> - 8 bit hexadecimal number scene ID.	<b>&lt;Status&gt;</b> - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> and <SCID> will not show.
<b>&lt;TransTime&gt;</b> - 16 bit hexadecimal number, the amount of time, in seconds it will take for the device to change from its current state to the requested scene.	<b>&lt;GPID&gt;</b> - 16 bit hexadecimal number, represent the group ID that will be added on the target device
<b>&lt;SCName&gt;</b> - String up to 16 characters representing scenes name	<b>&lt;SCID&gt;</b> - 8 bit hexadecimal number scene ID.
<b>[Extension field sets]</b> – It holds a set of values of attributes for a cluster implemented on the device. It can be constructed as: <ClusterID>,<Length>,<AttributeValue>. The sum of all such sets defines a scene. This is an optional parameter which can be omitted. Please check ZigBee Cluster Library for more details	<b>&lt;errorcode&gt;</b> : the code explained in section 4.
An example of a constructed command can be:	

**AT+SCADD:0234,01,0,0001,01,001C,Light,0006,01,01**

*Note:*

If there is already a scene in the table with the same Scene ID and Group ID, the new Scene entry shall overwrite the previous one.

### **+SCVIEW - View Scenes Information of Target Devices**

Execute Command

**AT+SCVIEW:<Address>,<EP>,<SendMode>,<GPID>,<SCID>**

**AT+SCVIEW:,,,<GPID>,<SCID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device. According to ZigBee Cluster Library, The View Scene command shall be addressed to a single device (not a group).

**<EP>** - 16 bit hexadecimal number represent the Endpoint of the target

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group. According to ZigBee Cluster Specification, this command shall be unicast. So SendMode value has been set to 0 for this command.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

**<SCID>** - 8 bit hexadecimal number scene ID.

Response

**OK**

**VIEWSC:<NodeID>,<EP>,<Status>,[<GPID>,<SCID>,<SCName>]**

or **ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID>, <SCID> and <SCName> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<SCID>** - 8 bit hexadecimal number scene ID

**<SCName>** - String up to 16 characters representing scenes name

**<errorcode>** : the code explained in section 4.

### +SCRMV - Remove A Scene On Target Device

Execute Command

**AT+SCRMV:<Address>,<EP>,<SendMode>,<GPID>,<SCID>**

**AT+SCRMV:,,,< GPID>,<SCID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

**<SCID>** - 8 bit hexadecimal number, scene ID.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**RMVSCRSP:<NodeID>,<EP>,<Status>,<GPID>,<SCID>**

or **ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> and <SCID> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<SCID>** - 8 bit hexadecimal number, scene ID.

**<errorcode>** : the code explained in section 4.

### +SCRMALL - Remove All Scenes On Target Device

Execute Command

**AT+SCRMALL:<Address>,<EP>,<SendMode>,<GPID>**

**AT+SCRMALL:,,,<GPID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

*Note:*

If the user does not determine a destination address the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**RMASCSRSP:<NodeID>,<EP>,<Status>,<GPID>**

or **ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> and <SCID> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<errorcode>** : the code explained in section 4.

### +SCSTORE – Store Scenes

Execute Command

**AT+SCSTORE:<Address>,<EP>,<SendMode>,<GPID>,<SCID>**

**AT+SCSTORE:,,,,<GPID>,<SCID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

**<SCID>** - 8 bit hexadecimal number, scene ID

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

**STORSCRSP:<NodeID>,<EP>,<Status>,<GPID>,<SCID>**

or **ERROR:<errorcode>**

**<NodeID>** - 16 bit hexadecimal number that represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8 bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <GPID> and <SCID> will not show.

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<SCID>** - 8 bit hexadecimal number, scene ID.

**<errorcode>** : the code explained in section 4.

### +SCRECAL - Recall Scenes

Execute Command

Response

**AT+SCRECAL:<Address>,<EP>,<SendMode>,<GPID>,<SCID>  
>**

**OK**

**AT+SCRECAL:,,,,<GPID>,<SCID>**

or

**ERROR:<errorcode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<errorcode>** please see section 4.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

**<SCID>** - 8 bit hexadecimal number, scene ID

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

**+SCGETMSH – Get Scene Membership**

Execute Command

**AT+SCGETMSH:<Address>,<EP>,<SendMode>,<GPID>**
**AT+SCGETMSH:,,,<GPID>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<GPID>** - 16 bit hexadecimal number, represent the specified group ID

*Note:*

This Command is used to find an unused scene number within the group when no commissioning tool is in the network, or for a commissioning tool to get used scenes for a group on a single device or on all group members.

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**
**SCMSHRSP:<NodeID>,<EP>,<Status>,<Capacity>,<GPID>,<SCCount>,<SCList>**

or

**ERROR:<errorcode>**

**<NodeID>** - 16-bit hexadecimal number represents the Node ID of the node which responds

**<EP>** - 8 bit hexadecimal number represent the Endpoint of the node which responds

**<Status>** - 8-bit hexadecimal number, if it is not 00, it will be an error code (please check section 4), then <Capacity>, <GPID>, <SCCount> and <SCList> will not show.

**<Capacity>** - 8 bit hexadecimal number. The Capacity field shall contain the remaining capacity of the group table of the device. The following values apply:

0 No further groups may be added.

0 < Capacity < 0xfe Capacity holds the number of groups that may be added

0xfe At least 1 further group may be added (exact number is unknown)

0xff It is unknown if any further groups may be added

**<GPID>** - 16 bit hexadecimal number, represent the group ID that will be added on the target device

**<SCCount>** - 8 bit hexadecimal number, represent contain the number of scenes contained in the Scene list field.

**<SCList>** - shall contain the identifiers of all the scenes in the scene table with the corresponding Group ID.

**<errorcode>** - explained in section 4.

### 3.5.2.5 On/Off Cluster – Client

#### +RONOFF – Switching Target Devices Between ‘On’ and ‘Off’ States

Execute Command	Response
<b>AT+RONOFF:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;[,&lt;ON/OFF&gt;]</b>	<b>OK</b>
<b>AT+RONOFF:,,,[&lt;ON/OFF&gt;]</b>	or
<p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;ON/OFF&gt;</b> - A Boolean type to choose transmission mode, 0 – means Off; 1 – means On. If this field is left blank, the command will be a toggle command.</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>



### 3.5.2.6 Level Control Cluster – Client

<b>+LCMVTOLEV – Move To Level</b>	
Execute Command	Response
<b>AT+LCMVTOLEV:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;ON/OFF&gt;,&lt;LevelValue&gt;,&lt;TransTime&gt;</b>	OK
<b>AT+LCMVTOLEV:,,,,&lt;ON/OFF&gt;,&lt;LevelValue&gt;,&lt;TransTime&gt;</b>	or ERROR:<errorcode>
<p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p>	<p><b>&lt;errorcode&gt;</b> - the code explained in section 4.</p>
<p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p>	<p>Default response is expected to show the status of the required operation on the target.</p>
<p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p>	
<p><b>&lt;ON/OFF&gt;</b> - 1 Boolean type number represents if the command is used with On/Off. If it is set to 0, it means the command is implemented as <i>Move to Level</i> command. If it is set to 1, it means the command will be implemented <i>Move to Level (with On/Off)</i> command.</p>	
<p><b>&lt;LevelValue&gt;</b> - 8 bit hexadecimal number, The meaning of 'level' is device dependent – e.g. for a light it may mean brightness level.</p>	
<p><b>&lt;TransTime&gt;</b> - 16 bit hexadecimal number, The movement shall be as continuous as technically practical, i.e. not a step function, and the time taken to move to the new level shall be equal to the value of the Transition time field, in tenths of a second, or as close to this as the device is able.</p>	
<p><i>Note:</i></p> <p>Before commencing any command that has the effect of increasing, CurrentLevel, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to On. If any command that decreases CurrentLevel reduces it to the minimum level allowed by the device, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to Off.</p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p>	

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

### +LCMV – Level Control Move Command

Execute Command

**AT+LCMV:<Address>,<EP>,<SendMode>,<ON/OFF>,<Mode>,<Rate>**

**AT+LCMV:,,,<ON/OFF>,<Mode>,<Rate>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<ON/OFF>** - 1 Boolean type number represents if the command is used with On/Off. If it is set to 0, it means the command is implemented as *Move* command. If it is set to 1, it means the command will be implemented *Move (with On/Off)* command.

**<Mode>** - 8 bit hexadecimal number representing move mode. 0x00 means up; 0x01 means down

**<Rate>** - 8 bit hexadecimal number, specifies the rate of movement in units per second. The actual rate of movement should be as close to this rate as the device is able. If the Rate field is 0xff the device should move as fast as it is able.

*Note:*

Before commencing any command that has the effect of increasing, CurrentLevel, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to On. If any command that decreases CurrentLevel reduces it to the minimum level allowed by the device, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to Off.

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (", " cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

### **+LCSTEP – Level Control Step Command**

Execute Command

**AT+LCSTEP:<Address>,<EP>,<SendMode>,<ON/OFF>,<Mode>,<Size>,<TransTime>**

**AT+LCSTEP:,,,<ON/OFF>,<Mode>,<Size>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<ON/OFF>** - 1 Boolean type number represents if the command is used with On/Off. If it is set to 0, it means the command is implemented as *Move* command. If it is set to 1, it means the command will be implemented *Move (with On/Off)* command.

**<Mode>** - 8 bit hexadecimal number representing move mode. 0x00 means up; 0x01 means down

**<Size>** - 8 bit hexadecimal number

**<TransTime>** - 16 bit hexadecimal number, The movement shall be as continuous as technically practical, i.e. not a step function, and the time taken to move to the new level shall be equal to the value of the Transition time field, in tenths of a second, or as close to this as the device is able.

*Note:*

Before commencing any command that has the effect of increasing, CurrentLevel, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to On. If any command that decreases CurrentLevel reduces it to the minimum level allowed by the device, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to Off.

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

If destination address, endpoint and SendMode are not given (", " cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

### **+LCSTOP – Level Control Stop Command**

Execute Command	Response
<p><b>AT+LCSTOP:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;ON/OFF&gt;</b></p> <p><b>&gt;</b></p> <p><b>AT+LCSTOP:&lt;ON/OFF&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;ON/OFF&gt;</b> - 1 Boolean type number represents if the command is used with On/Off. If it is set to 0, it means the command is implemented as <i>Move</i> command. If it is set to 1, it means the command will be implemented <i>Move (with On/Off)</i> command.</p> <p><i>Note:</i></p> <p>Before commencing any command that has the effect of increasing, CurrentLevel, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to On. If any command that decreases CurrentLevel reduces it to the minimum level allowed by the device, the On/Off attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to Off.</p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p>OK</p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>

### 3.5.2.7 Colour Control Cluster – Client

#### +CCMVTOCOL – Colour Control Move to Colour Command

Execute Command	Response
<b>AT+CCMVTOCOL:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;ColourX&gt;,&lt;ColourY&gt;,&lt;TransTime&gt;</b>  <b>AT+CCMVTOCOL:,,,,&lt;ColourX&gt;,&lt;ColourY&gt;,&lt;TransTime&gt;</b>	<b>OK</b>  or <b>ERROR:&lt;errorcode&gt;</b>
<p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;ColourX&gt;</b> - 16 bit hexadecimal number.</p> <p><b>&lt;ColourY&gt;</b> - 16 bit hexadecimal number.</p> <p><b>&lt;TransTime&gt;</b> - 16 bit hexadecimal number.</p> <p>Please check ZigBee Cluster Library for more field explanation</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>&lt;errorcode&gt;</b>: the code explained in section 4</p> <p>Default response is expected to show the status of the required operation on the target.</p>

**+CCMVCOL – Colour Control Move Colour Command**

Execute Command

**AT+CCMVTOCOL:<Address>,<EP>,<SendMode>,<RateX>,<RateY>**
**AT+CCMVTOCOL:,,,<RateX>,<RateY>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<RateX>** - 16 bit hexadecimal number

**<RateY>** - 16 bit hexadecimal number

Please check ZigBee Cluster Library for more field explanation

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>**: the code explained in section 4.

Default response is expected to show the status of the required operation on the target.



**+CCSTEPCOL – Colour Control Step Colour Command**

Execute Command

Response

**AT+CCSTEPCOL:<Address>,<EP>,<SendMode>,<StepX>,<StepY>,<TransTime>**
**OK**
**AT+CCSTEPCOL:,,,<StepX>,<StepY>,<TransTime>**

or

**ERROR:<errorcode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<errorcode>** : the code explained in section 4.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

Default response is expected to show the status of the required operation on the target.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

The <StepX> and <StepY> fields specify the change to be added to the device's CurrentX attribute and CurrentY attribute respectively.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

### **+CCMVTOHUE – Colour Control Move to Hue Command**

Execute Command

**AT+CCMVTOHUE:<Address>,<EP>,<SendMode>,<Hue>,<Direction>,<TransTime>**

**AT+CCMVTOHUE:,,,,<Hue>,<Direction>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Hue>** - 8 bit hexadecimal number.

**<Direction>** - 8 bit hexadecimal number.

**<TransTime>** - 16 bit hexadecimal number. The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform the step. A step is a change in the device's saturation of 'Step size' units.

Please check ZigBee Cluster Library for more field explanation

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given ("," cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

**+CCMVHUE – Colour Control Move Hue Command**

Execute Command

Response

**AT+CCMVHUE:<Address>,<EP>,<SendMode>,<Mode>  
<Rate>**
**OK**
**AT+CCMVHUE:,,,<Mode>,<Rate>**

or

**ERROR:<errorcode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<errorcode>** : the code explained in section 4.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

Default response is expected to show the status of the required operation on the target.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Mode>** - 8 bit hexadecimal number , move mode

**<Rate>** - 8 bit hexadecimal number

Please check ZigBee Cluster Library for more field explanation

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

**+CCSTEPHUE – Colour Control Step Hue Command**

Execute Command

**AT+CCSTEPHUE:<Address>,<EP>,<SendMode>,<StepMode>,<StepSize>,<TransTime>**
**AT+CCSTEPHUE:,,,<StepMode>,<StepSize>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid endpoint is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<StepMode>** - 8 bit hexadecimal number, 0x01 means Up; 0x03 means Down

**<StepSize>** - 8 bit hexadecimal number

**<TransTime>** - 16 bit hexadecimal number

Please check ZigBee Cluster Library for more field explanation

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

### **+CCMVTOSAT – Colour Control Move to Saturation Command**

Execute Command

**AT+CCMVTOSAT:<Address>,<EP>,<SendMode>,<Sat>,<TransTime>**

**AT+CCMVTOSAT:,,,<Sat>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Sat>** - 8 bit hexadecimal number representing saturation

**<TransTime>** - 16 bit hexadecimal number. The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform the step. A step is a change in the device's saturation of 'Step size' units.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given ("," cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

### **+CCMVSAT – Colour Control Move Saturation Command**

Execute Command	Response
<p><b>AT+CCMVSAT:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Mode&gt;,&lt;Rate&gt;</b></p> <p><b>AT+CCMVSAT:,,,,&lt;Mode&gt;,&lt;Rate&gt;</b></p> <p><b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number represents the target's Node ID</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Mode&gt;</b> - 8 bit hexadecimal number, 0x01 means Up; 0x03 means Down</p> <p><b>&lt;Rate&gt;</b> - 8 bit hexadecimal number</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p>OK</p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>

### **+CCSTEPSAT – Colour Control Step Saturation Command**

Execute Command

**AT+CCSTEPSAT:<Address>,<EP>,<SendMode>,<StepMode>,<Step Size>,<TransTime>**

**AT+CCSTEPSAT:<StepMode>,<StepSize>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid endpoint is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<StepMode>** - 8 bit hexadecimal number, 0x01 means Up; 0x03 means Down

**<StepSize>** - 8 bit hexadecimal number, The change to be added to (or subtracted from) the current value of the device's saturation.

**<TransTime>** - 16 bit hexadecimal number. The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform the step. A step is a change in the device's saturation of 'Step size' units.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** - the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

### **+CCMVTOCT – Colour Control Move to Colour Temperature Command**

Execute Command

**AT+CCMVTOCT:<Address>,<EP>,<SendMode>,<CT>,<TransTime>**

**AT+CCMVTOCT:,,,<CT>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<CT>** - 16 bit hexadecimal number, colour temperature

**<TransTime>** - 16 bit hexadecimal number. The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform the step. A step is a change in the device's saturation of 'Step size' units.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : please see section 4.

Default response is expected to show the status of the required operation on the target.



### **+CCMVTOHUS – Colour Control Move to Hue Saturation Command**

Execute Command

**AT+CCMVTOHUS:<Address>,<EP>,<SendMode>,<Hue><Sat>,<TransTime>**

**AT+CCMVTOHUS:,,,<Hue>,<Sat>,<TransTime>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid endpoint is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Hue>** - 8 bit hexadecimal number.

**<Sat>** - 8 bit hexadecimal number representing saturation

**<TransTime>** - 16 bit hexadecimal number. The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform the step. A step is a change in the device's saturation of 'Step size' units.

Please check ZigBee Cluster Library for more field explanation

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : please see section 4.

Default response is expected to show the status of the required operation on the target.

### 3.5.2.8 Thermostat – Client

<b>+TSTATSET – Send A Thermostat Set Command</b>	
Execute Command	Response
<b>AT+TSTATSET:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;SetMode&gt;,&lt;Amount&gt;</b>  <b>AT+TSTATSET: ,,,&lt;SetMode&gt;,&lt;Amount&gt;</b>  <b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.  <b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.  <b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.  <b>&lt;SetMode&gt;</b> - 8 bit hexadecimal number, specifies which setpoint is to be configured as following:  0x00 - Heat (adjust Heat Setpoint)  0x01 - Cool (adjust Cool Setpoint)  0x02 Both (adjust Heat Setpoint and Cool Setpoint)  <b>&lt;Amount&gt;</b> - 8 bit hexadecimal number representing the amount the setpoint(s) are to be a increased (or decreased) e.g.: +01 means increased by 0.1 °C  <i>Note:</i>  If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.  If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.	<b>OK</b>  <b>or ERROR:&lt;errorcode&gt;</b>  <b>&lt;errorcode&gt;</b> : please see section 4.  Default response is expected to show the status of the required operation on the target.

### 3.5.2.9 Door Lock – Client

#### +DRLOCK – Issue A Lock/Unlock Door Command

Execute Command	Response
<p><b>AT+DRLOCK:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Lock/Unlock&gt;</b></p> <p><b>AT+DRLOCK:,,, &lt;Lock/Unlock&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Lock/Unlock&gt;</b> - A Boolean type to choose transmission mode, 0 – means Unlock; 1 – means Lock.</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>If the destination node supports Door Lock cluster, a door lock/unlock response is expected to report the status and the following prompt may show respectively:</p> <p>DRLOCRSP:&lt;nodeID&gt;,&lt;ep&gt;,&lt;status&gt;</p> <p>Or</p> <p>DRUNLOCKRSP:&lt;nodeID&gt;,&lt;ep&gt;,&lt;status&gt;</p> <p>If the destination node does not support the door lock/unlock command, a Default Response will be received and <b>DFTREP</b> prompt shall show.</p>

### 3.5.2.10 Window Covering – Client

#### +WNPGETPT – Issue A Window Program Setpoint Command to Target

Execute Command	Response
<b>AT+WNPGETPT:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Lift/Tile&gt;,&lt;SetpointIndex&gt;,&lt;SetpointValue&gt;</b>  <b>AT+WNPGETPT:,,,,&lt;Lift/Tile&gt;,&lt;SetpointIndex&gt;,&lt;SetpointValue&gt;</b>  <b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.  <b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid endpoint is 0x01 to 0xF0.  <b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.  <b>&lt;Lift/Tile&gt;</b> - a Boolean type specifies the command as following:  0 - Go to lift percentage and 1 - Go to tile percentage  <b>&lt;SetpointIndex&gt;</b> - 8 bit hexadecimal number represents the index of Lift/Tile Setpoint  <b>&lt;SetpointValue&gt;</b> - 16 bit hexadecimal number representing the setpoint value  <i>Note:</i>  If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.  If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the <address> parameter.	OK  or  <b>ERROR:&lt;errorcode&gt;</b>  <b>&lt;errorcode&gt;</b> : the code explained in section 4.  Default response is expected to show the status of the required operation on the target.

### **+WNSSETPT – Issue A Window Go To Lift/Tile Setpoint Command to Target**

Execute Command	Response
<p><b>AT+WNSSETPT:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Lift/Tile&gt;,&lt;Setpoint&gt;</b></p> <p><b>AT+WNSSETPT:,,, &lt;Lift/Tile&gt;,&lt;Setpoint&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Lift/Tile&gt;</b> - a Boolean type specifies the command as following:</p> <p>0 - Go to lift percentage and 1 - Go to tile percentage</p> <p><b>&lt;Setpoint&gt;</b> - 8 bit hexadecimal number represents the index of Lift/Tile Setpoint</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>

**+WNVALUE – Issue A Window Go To Lift/Tile Value Command to Target**

Execute Command	Response
<p><b>AT+WNVALUE:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Lift/Tile&gt;,&lt;Value&gt;</b></p> <p><b>AT+WNVALUE:,,, &lt;Lift/Tile&gt;,&lt;Value&gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Lift/Tile&gt;</b> - a Boolean type specifies the command as following: 0 - Go to lift value and 1 - Go to tile value</p> <p><b>&lt;Value&gt;</b> - 16 bit hexadecimal number representing the value of movement</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : please see section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>

### **+WNPCTG – Issue A Window Go To Lift/Tile Percentage Command to Target**

Execute Command	Response
<p><b>AT+WNPCTG:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Lift/Tile&gt;,&lt;Percentage&gt;</b></p> <p><b>AT+WNPCTG:,,,&lt;SetMode&gt;,&lt; Percentage &gt;</b></p> <p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Lift/Tile&gt;</b> - a Boolean type specifies the command as following:</p> <p>0 - Go to lift percentage and 1 - Go to tile percentage</p> <p><b>&lt;Percentage&gt;</b> - 8 bit hexadecimal number, which specifies the movement percentage</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>OK</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>

**+WNCTRL – Issue A Window Control Command to Target (Open, Close or Stop Movement)**

Execute Command	Response
<b>AT+WNCTRL:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;Control&gt;</b>	<b>OK</b>
<b>AT+WNCTRL:,,, &lt;Control&gt;</b>	or
<p><b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.</p> <p><b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.</p> <p><b>&lt;Control&gt;</b> - 1 decimal number, specifies the adjusting movement of the target window:</p> <p>0 – Window open or move up</p> <p>1 - Window closing or move down</p> <p>2 - Window stop any adjusting movement</p> <p><i>Note:</i></p> <p>If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.</p> <p>If destination address, endpoint and SendMode are not given (“,” cannot be omitted) the command will search the binding table for a destination. If this is the user's preferred method then it is necessary to add the required destination to the local binding table in advance. Thereafter it will not be necessary to fill in the &lt;address&gt; parameter.</p>	<p><b>ERROR:&lt;errorcode&gt;</b></p> <p><b>&lt;errorcode&gt;</b> : the code explained in section 4.</p> <p>Default response is expected to show the status of the required operation on the target.</p>



### 3.5.3 CIE Specified Command Set

#### +WARNING – Issue A Start Warning Command to Target (Warning Device)

Execute Command

**AT+WARNING:<Address>,<EP>,<SendMode>,<Warning>,<Warning Duration>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Warning>** - 8 bit hexadecimal number,

4 bit	2 bit	2 bit
Warning mode	Strobe	reserved

Warning mode :

0	Stop(no warning)
1	Burglar
2	Fire
3	Emergency
4-15	Reserved

Strobe:

0	No strobe
1	Use strobe in parallel to warning
2-3	Reserved

**<Warning Duration>** - 16 bit hexadecimal number. Requested duration of warning, in seconds. If both Strobe and Warning Mode are "0" this field shall be ignored.

*Note:*

If the address is a Node ID, please set SendMode to be 0, or the address will be recognized as a group ID.

Response

OK

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

**+SQUAWK– Issue A Squawk Command to Target (Warning Device)**

Execute Command

**AT+SQUAWK:<Address>,<EP>,<SendMode>,<Squawk>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device if the command is sent directly to a node or it shall be a group ID if the command is sent to a group.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device. Valid end point is 0x01 to 0xF0.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

**<Squawk>** - 8 bit hexadecimal number,

4 bit	1 bit	1 bit	2 bit
Squawk mode	Strobe	Reserved	Squawk Level

Squawk Mode:

Value	Meaning
0	Notification sound for "System is armed"
1	Notification sound for "System is disarmed"
2-15	Reserved

Strobe

Value	Meaning
0	No strobe
1	Use strobe blink in parallel to squawk

Squawk Level

Value	Meaning
0	Low level sound
1	Medium level sound
2	High level sound
3	Very High level sound

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

Default response is expected to show the status of the required operation on the target.

### **+ADDZENTRY – Add A Zone Table Entry To Local Zone Table**

Execute Command

**AT+ADDZENTRY:<Index>,<Zone Type>,<Node IEEE >**

**<Index>** - 8 bit hexadecimal number. Rang from 00 to 1F. It is the index of a zone table entry. Note: entries from 00 to 0F are flash maintained; entries 10 to 1F are RAM maintained.

**<Zone Type>** - 16 bit hexadecimal number, represent zone type of zone device. Please check ZigBee Cluster Library for more information.

**<Node IEEE>** - A Node EUI address.

*Note:*

This command can be used to add a zone table entry into local zone table. It should be advised to use AT+ZTABLE to find and use a free entry. (An entry with all "F" is a free entry).

Response

**OK**

or

**ERROR:<errorcode>**

**<errorcode>** : the code explained in section 4.

### **+DELZENTRY – Delete A Zone Table Entry From Local Zone Table**

Execute Command

**AT+DELZENTRY:<Index or EUI>**

**<Index or EUI>** - the user can use 8 bit hexadecimal Zone ID (00 - 1F) or a device's EUI to remove a zone entry form local zone table.

Response

**OK**

or

**ERROR:<errorcode>** the code explained in section 4.

If the trying to remove a non-existing entry using EUI, NOTFOUNDZONE prompt will show.

**+ZENTRY – Print A Zone Table Entry In Local Zone Table**

Execute Command

**AT+ZENTRY:<Index>**

**<Index>** - 8 bit hexadecimal number. Rang from 00 to 1F. It is the index of a zone table entry

Response

**ZONEENTRY:<Index>,<Zone Type>,<Node EUI>**
**OK**

or

**ERROR:<errorcode>** the code explained in section 4.

**<Index>** - 8 bit hexadecimal number. Rang from 00 to 0F. It is the index of a zone table entry

**<Zone Type>** - 16 bit hexadecimal number, represent zone type of zone device. Please check ZigBee Cluster Library for more information.

**<Node IEEE>** - EUI address of a node which has been enrolled in CIE's Zone.

**+ZTABLE – Print Local Zone Table**

Execute Command

**AT+ZTABLE**

Response

**ZONEENTRY:<Index>,<Zone Type>,<Node EUI>**

....

**OK**

or

**ERROR:<errorcode>** the code explained in section 4

**<Index>** - 8 bit hexadecimal number. Rang from 00 to 1F. It is the index of a zone table entry

**<Zone Type>** - 16 bit hexadecimal number, represent zone type of zone device. Please check ZigBee Cluster Library for more information.

**<Node IEEE>** - EUI address of a node which has been enrolled in CIE's Zone.

*Note:*

Using this command to show local zone table information.

**+ZCLEAR – Delete A Zone Table Entry From Local Zone Table**

Execute Command	Response
<b>AT+ZCLEAR</b>	<b>OK</b>
<i>Note:</i>	or
Using this command to clear local zone table. All entry will be erased.	<b>ERROR:&lt;errorcode&gt;</b> the code explained in section 4.
When the CIE leaves network, local zone table will be cleared.	

**+PRINTMZTYPE – Print Local Table of Manufacture Defined Zone Type**

Execute Command	Response
<b>AT+PRINTMZTYPE</b>	<b>MZONETYPE 00:&lt;ZoneType&gt;</b>
	<b>MZONETYPE 01:&lt;ZoneType&gt;</b>
<i>Note:</i>	<b>MZONETYPE 02:&lt;ZoneType&gt;</b>
Using this command to print CIE supported manufacture defined zone type. There are six entries which can be used to add manufacture defined zone type. These entries are not self-maintained during power circle.	<b>MZONETYPE 03:&lt;ZoneType&gt;</b>
	<b>MZONETYPE 04:&lt;ZoneType&gt;</b>
	<b>MZONETYPE 05:&lt;ZoneType&gt;</b>
Please check Appendix B for more information.	<b>OK</b>
	or
	<b>ERROR:&lt;errorcode&gt;</b> the code explained in section 4.

### **+MZONETYPE – Add User Defined Zone Type in Local Zone Type Table**

Execute Command	Response
<b>AT+MZONETYPE:&lt;Index&gt;,&lt;ZoneType&gt;</b>  <b>&lt;Index&gt;</b> - 8 bit hexadecimal number, index of local table of supported user specified zone type. Range from 00 to 05.  <b>&lt;ZoneType&gt;</b> - 16 bit hexadecimal number, which represents the zone type.  <i>Note:</i>  Using this command to add user defined zone type which is used to filter out unwanted zone type during enrolment.  S61 can be set to check Zone Type before accept enrol request. Please check section 5 and Appendix B for more information.	<b>OK</b>  or  <b>ERROR:&lt;errorcode&gt;</b>  The error code explained in section 4.

### **+PRINTZMAP – Print Local Zone Map**

Execute Command	Response
<b>AT+PRINTZMAP</b>  <i>Note:</i>  Using this command to print local Zone map. This zone map information is used to respond to getZoneldMap command sent by ACE client	<b>ZMAPSECTION 00:&lt;Section Value&gt;</b>  ...  <b>ZMAPSECTION 0F:&lt;Section Value&gt;</b>

### **+EDITZMAP – Edit Local Zone Map**

Execute Command

**AT+EDITZMAP**

*Note:*

Using this command to print local Zone map. This zone map information is used to respond to getZoneIDMap command sent by ACE client

Response

ZMASECTION 00:<Section Value>

ZMASECTION 0F:<Section Value>

### **+ZINFORSP – Construct And Send A Get Zone Information Response**

Execute Command

**AT+ZINFORSP:<Address>,<EP>,  
<SendMode>,<ZoneID>,<ZoneType>,  
<EUI>,<SequenceNumber>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - The user should use 0 to send a unicast command.

**<ZoneID>** 8 bit hexadecimal number which represents zone ID, the user should use this zone ID to retrieve corresponding zone entry information. If there is no such a zone in zone table, the user should use 0xFFFF for Zone Type and 0xFFFFFFFFFFFFFFFF for EUI.

**<ZoneType>** - 16 bit hexadecimal number which represents zone type.

**<EUI>** - the zone device's IEEE address

**<SequenceNumber>** - 8 bit hexadecimal sequence number. This sequence number should be the same with the sequence number contained in **<GETZONEINFO>**

*Note:*

Using this command to respond to a *get zone information* command when a **"GETZONEINFO"** prompt shows.

Response

**DFTREP** prompt is expected after sending this command.

#### **+ARMPRINT – Show CIE Arm Mode**

Execute Command	Response
<b>AT+ARMPRINT</b>	<b>ARMMODE:&lt;Arm Mode&gt;</b>
	<b>OK</b>
<i>Note:</i>	or
Using this command to check local CIE ARM mode.	<b>ERROR:&lt;errorcode&gt;</b> the code explained in section 4.
	According to ZigBee Cluster Library, Arm Mode Field Value has following meanings:
	0x00 : Disarm
	0x01 : Arm Day/Home Zones Only
	0x02 : Arm Night/Sleep Zones Only
	0x03 : Arm All Zones

#### **+ZTEST – Place A Remote Zone Server Device into Test Mode**

Execute Command	Response
<b>AT+ZTEST:&lt;Address&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;TestModeDuration&gt;,&lt;CurrentZoneSensitivityLevel&gt;</b>	<b>OK</b>
	or
<b>&lt;Address&gt;</b> - 16 bit hexadecimal number. It shall be the Node ID of a remote device.	<b>ERROR:&lt;errorcode&gt;</b> the code explained in section 4.
<b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device.	
<b>&lt;SendMode&gt;</b> - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.	
<b>&lt;TestModeDuration&gt;</b> - 8 bit unsigned hex. It specifies the duration, in seconds, for which the IAS Zone server SHALL operate in its test mode.	
<b>&lt;CurrentZoneSensitivityLevel&gt;</b> - 8 bit unsigned hex. It specifies the sensitivity level the IAS Zone server SHALL use for the duration.	



---

**+ZNORMAL – Set A Remote Zone Server Device To Normal Working Mode**

Execute Command

**AT+ZNORMAL:<Address>,<EP>,<SendMode>**

**<Address>** - 16 bit hexadecimal number. It shall be the Node ID of a remote device.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - A Boolean type to choose transmission mode, 0 – means sending command directly; 1 – means sending command to a group.

Response

**OK**

or

**ERROR:<errorcode>** the code explained in section 4.

### 3.5.4 Tunnel Cluster Commands

#### **+TNLOPEN – Send An Open Tunnel Request To The Target**

Execute Command

**AT+TNLOPEN:<NodeID>,<EP>**

**<NodeID>** - 16 bit hexadecimal number which represents target's node ID.

**<EP>** - 8 bit hexadecimal number which represents Endpoint number.

Response

**OK**

**TNLOPEN:<TunnelID>,<Status>**

or **ERROR:<errorcode>**

**<TunnelID>** - 16 bit hexadecimal number, which represents the tunnel ID.

**<Status>** - 8 bit hexadecimal number, please refer to Table 1 for reference

**<errorcode>** represents the error code. Note: if the errorcode is 01, 02, 03, 04, please find Table 1 for meaning, other code can be found in section 4.

**Please note: when the TunnelID is FFFF, check if the Status is 00.**

Table 1 Tunnel Status Code

Value	Description
0x00	Success
0x01	Busy
0x02	No more tunnel ID
0x03	Protocol not supported
0x04	Flow control not supported
0x05	Reserved

#### **+TNLCLOSE – Send A Close Tunnel Request To The Target**

Execute Command

**AT+TNLCLOSE:<TunnelID>**

**<TunnelID>** - 16 bit hexadecimal number which represents the tunnel ID to be closed.

Use this command to send a *CloseTunnel* command to Tunnelling server.

*Note*

This command can only be used to send a Close Tunnel Request. It cannot close local server tunnel.

Response

**OK**

or

**ERROR:<errorcode>** (see section 4)

Please note: If Tunnelling client does not send a *CloseTunnel* command. The tunnel will be closed automatically by the server when Tunnel life time runs out.

If errorcode is 0x05, please check if the input tunnelID is correct and the tunnel exists.

<b>+TNLSEND– Send Tunnel Data To The Target</b>	
Execute Command	Response
<b>AT+TNLSEND:&lt;TunnelID&gt;,&lt;Length&gt;</b>  <b>&lt;TunnelID&gt;</b> - 16 bit hexadecimal number which represents Tunnel ID.  <b>&lt;Length&gt;</b> - 8 bit hexadecimal number which indicates the length of the transmitted tunnel data.	<b>&gt;&lt;data being entered&gt;</b>  <b>or ERROR:&lt;errorcode&gt;</b>  <b>&lt;errorcode&gt;</b> represents the error code (section 4).  If errorcode is 0x05, please check if the input tunnelID is correct and the tunnel exists.  In case that data is sent successfully but to a wrong address or tunnel has been closed (time out), the following prompt will show:
<i>Note</i>  After issuing the command (using \r), the prompt ">" will occur then please input binary data. E.g.  AT+TNLSEND:5747,06  >123456  The binary data will be counted using the <Length>. Once the length is reached, data will be sent immediately through tunnel	<b>TNLERROR:&lt;TunnelID&gt;,&lt;TunnelError&gt;</b>  <b>&lt;TunnelID&gt;</b> - 16 bit hexadecimal number indicating the tunnel which is used for sending data and incur error.  <b>&lt;TunnelError&gt;</b> - 8 bit hexadecimal number indicating the reason of error.  0x00 – No such tunnel (The TransferData command contains a TunnelID of a non-existent tunnel)  0x01 – Wrong device (The TransferData command contains a TunnelID that does not match the device sending the data)  Use this command to send tunnel data to the target.  <b>On another site of the tunnel, it should show:</b>  <b>TNLDATA:&lt;TunnelID&gt;,&lt;Length&gt;,&lt;Data&gt;</b>

**+TNLSTATUS – Check Local Tunnel Status**

Execute Command

**AT+TNLSTATUS**

To check if there is any opened tunnel on local server and client tunnel tables

*Note*

This firmware only supports one client tunnel entry and one server tunnel entry

Response

TNLC 00: <EUI>,<ClientEP>,<SourceEP>,<TunnelID>  
TNLS 00:<EUI>,<ClientEP>,<SourceEP>,<TunnelID>

**OK**

**<EUI> is the EUI number of the node, which the tunnel has been built with.**

### 3.5.5 Time Cluster Command

#### **+SETTIME - Set Local Time**

Execute Command

**AT+SETTIME: <time>**

**<time>** - 32 bit hexadecimal number representing time in UTC format (number of seconds since 01.01.2000 00:00)

Response

**OK**

or

**ERROR:<errorcode>**

#### **+GETTIME - Get The Local Time**

Execute Command

**AT+GETTIME**

Response

**TIME:<time>**

**OK**

or **ERROR:<errorcode>**

Returns current local time in UTC format.

**<time>** - 32 bit hexadecimal value representing the local time, number of seconds since 0 hours, 0 minutes, 0 seconds, on the 1st of January, 2000 UTC.

Notes:

The CICIE firmware supports Time Server cluster on it endpoint 0x01. The +SETTIME and +GETTIME command can be used to set and get local clock time. The user shall use AT+SETATR:000A,<AttributeID>,<AttributeValue> to set other time server cluster attributes for example *timeStatus*, *dstStart*, *dstEnd* and so on. For more information, please refer to Appendix C.

### 3.5.6 OTA Communication Commands

#### **+IMGNOTIFY** - Notify OTA client of the newly available Image

Execute Command

**AT+IMGNOTIFY:<NodeID>,<EP>,  
<SendMode>,<PayloadType>, <QueryJitter>  
[,<ManufCode>, <ImageType>,<FileVersion>]**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address, a group ID or broadcast address (e.g FFFC)

**<EP>** - 8 bit hex number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group, 6 – means sending the command as broadcast

**<PayloadType>** - 8 bit hexadecimal number *representing* the Image Notify Command Payload Type. This parameter will determine and indicate which parameter will be included in the command

Payload type	Description
0x00	Query jitter
0x01	Query jitter and manufacturer code
0x02	Query jitter, manufacturer code, and image type
0x03	Query jitter, manufacturer code, image type, and new file version
0x04 – 0xff	Reserved

**<QueryJitter>** - 8 bit hexadecimal number range from 0x00 to 0x64 It indicates whether the client receiving Image Notify Command should send in Query Next Image Request command or not.

**<ManufCode>** - 16 bit hexadecimal number representing manufacture code. Manufacturer code when included in the command should contain the specific value that indicates certain manufacturer. If the server intends for the command to be applied to all manufacturers then the value should be omitted.

**<ImageType>** - 16 bit hexadecimal number. When imageType is included in the command, it should contain the specific value that indicates certain file type. If the server intends for the command to be applied to all image type values then the value (0xffff) should be used

**<FileVersion>** - 32 bit hexadecimal number. The value shall be the OTA upgrade file version that the server tries to upgrade client devices in the network to. If the server intends for the command to be applied to all file version values then the wild card value (0xffffffff) should be used.

Response

**OK**

Or

**ERROR:<errorcode>**

This command does not have response prompt.

**+UPGRADE** - Instruct a device to upgrade now

Execute Command

**AT+UPGRADE:<NodeID>,<EP>,<SendMode>,<ManufacturerCode>,<ImageType>,<FileVersion>,<CurrentTime>,<UpgradeTime>,<Seq>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address, a group ID or broadcast address (e.g FFFC)

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group

**<ManufCode>** - 16 bit hexadecimal number representing manufacture code. Manufacturer code when included in the command should contain the specific value that indicates certain manufacturer. If the server intends for the command to be applied to all manufacturers then the value should be omitted.

**<ImageType>** - 16 bit hexadecimal number. When imageType is included in the command, it should contain the specific value that indicates certain file type. If the server intends for the command to be applied to all image type values then the wild card value (0xffff) should be used.

**<FileVersion>** - 32 bit hexadecimal number. The value shall be the OTA upgrade file version that the server tries to upgrade client devices in the network to. If the server intends for the command to be applied to all file version values then the wild card value (0xffffffff) should be used.

**<CurrentTime>** - 32 bit UTC time which is the server current time.

**<UpgradeTime>** - 32 bit UTC time which instruct the upgrade time for the OTA client.

**<Seq>** - 8 bit hexadecimal number, this is the sequence number of the response. The user should use the sequence number contained in the UPGRADEREQ prompt.

Response

**OK**

Or

**ERROR:<errorcode>**

This command does not have response prompt.

**+QIMGRSP - Query next image response (It is used to respond to Query next image request)**

Response Command

**AT+QIMGRSP:<NodeID>,<EP>,<SendMode>,<Status>[,<ManufCode>,<ImgType>,<FileVersion>,<ImgSize>], <Seq>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID.

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly.

**<Status>** - 8 bit hexadecimal number representing the response status. Only if the status is SUCCESS that other fields are included. If the code is not success, please apply the following code according to status.

7E - Server is not authorized to upgrade the client

80 - The command received is badly formatted. It usually means the command is missing certain fields or values included in the fields are invalid.

98 - No OTA upgrade image available for a particular client

More status code can be found in OTA cluster specification section 6.10.2.

If **<Status>** is **Success**, the following parts will show

**<ManufCode>** - 16 bit hexadecimal number representing manufacture code. The value shall be the one received by the server in the Query Next Image Request command.

**<ImgType>** - 16 bit hexadecimal number. The value shall be the one received by the server in the Query Next Image Request command.

**<FileVersion>** - 32 bit hexadecimal number. It indicates the image version that the client is required to install.

**<ImgSize>** - 32 bit hexadecimal number. The value represents the total size of the image (in bytes) including header and all sub-elements

**<Seq>** - 8 bit hexadecimal number, this is the sequence number of the response. The user should use the sequence number contained from the IMGQUERY prompt.

Response

**OK**

Or

**ERROR:<errorcode>**

This command does not have response prompt.

Note:

This command is only used by the Host application to respond to a received QueryNextImageRequest command. The parameters like ManufCode, ImgType and so on can be obtained from the QueryNextImageRequest command.



**+IMGBRSP - Image block response** (It is used to respond to Image block request)

Response Command

**AT+IMGBRSP:<NodeID>,<EP>,<SendMode>,<Status> [,Appended parameters], <Seq>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group

If this command is used to respond to a ZCL *ImagePageRequest* command, the user SHALL use set this field to 5. This sendMode value is specifically used for this command to disable APS retry option (this is an requirement in OTA Cluster specification)

**<Status>** - 8 bit hexadecimal number representing the response status.

This command will have three forms depending on **<Status>**, the status can be **SUCCESS**, **ABORT** or **WAIT\_FOR\_DATA**

If **<Status>** is **SUCCESS** (0x00), the [Appended parameters] will include the following part.

**<ManufCode>** - 16 bit hexadecimal number representing manufacture code. The value shall be the same as the one included in Image Block/Page Request command.

**<ImgType>** - 16 bit hexadecimal number. The value shall be the same as the one included in Image Block/Page Request command.

**<FileVersion>** - 32 bit hexadecimal number. It indicates the image version that the client is required to install. The version value may be lower than the current image version on the client if the server decides to perform a downgrade. The version value may be the same as the client's current version if the server decides to perform a reinstall. However, in general, the version value

Response

&gt;&lt;data being entered&gt;

**OK**

Or

**ERROR:<errorcode>**

This command does not have response prompt.

*Note:*

This command is only used by the Host application for responding ImageBlockRequest command.

The parameters like ManufCode, ImgType and so on can be obtained when receiving the request command.

should be higher than the current image version on the client to indicate an upgrade.

**<Fileoffset>** - 32 bit hexadecimal number. The value represents the location of the data requested by the client. For most cases, the file offset value included in the (Image Block) response should be the same as the value requested by the client. For (unsolicited) Image Block responses generated as a result of Image Page Request, the file offset value shall be incremented to indicate the next data location.

**<DataSize>** - 8 bit hexadecimal number. The value indicates the length of the image data (in bytes) that is being included in the command. The value may be equal or smaller than the maximum data size value requested by the client.

*Note:*

After the parameter **<DataSize>** has been given, please use a (\r). When the prompt ">" occurs, please input **<ImageData>**

For example, if we want to send hex data "0xAB 0xBD 0x02 0x13 0x04 0x05" the raw hex values should be sent over the serial port.

The data will be counted using the **<DataSize>**. Once the **DataSize** is reached, data will be sent immediately to the destination.

**<ImageData>** is the actual OTA upgrade image data with the length equals to data size value

If **<Status>** is **WAIT\_FOR\_DATA** (0x97), the [Appended parameters] will include the following part.

**<CurrentTime>** - UTC time which is the OTA server's current time.

**<RequestTime>** - UTC time. It instructs that the client shall retry the request command and wait at least the request time value before trying again.

The client shall wait at least the request time value before trying again. In case of sleepy device, it may choose to wait longer than the specified time in order to not disrupt its sleeping cycle. If the current time value is zero that means the server does not support UTC time and the client shall treat the request time value as offset time. If neither time value is zero, and the client supports UTC time, it shall treat the request time value as UTC time. If the client does not support UTC time, it shall calculate the offset time from the difference between the two time values. The offset indicates the minimum

amount of time to wait in seconds. The UTC time indicates the actual time moment that needs to pass before the client should try again.

**<BlockRequestDelay>** - 16 bit hexadecimal number This value is only included if the server supports rate limiting. If the server does not support rate limiting or does not wish to slow the client's download, the field shall be set to 0.

If **<Status>** is **ABORT** (0x95), there will be no appended field.

No matter which status code is used, the user need to feed the **<Seq>**.

**<Seq>** - 8 bit hexadecimal number, this is the sequence number of the response. The user should use the sequence number contained in the IMGBREQ prompt.

<b>+DFTRSP - Image block response</b> (It is used to respond to Image block request)	
<p>Response Command</p> <p><b>AT+DFTRSP:&lt;NodeID&gt;,&lt;EP&gt;,&lt;SendMode&gt;,&lt;ClusterID&gt;,&lt;CmdID&gt;,&lt;Status&gt;,&lt;Seq&gt;</b></p> <p><b>&lt;NodeID&gt;</b> - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID</p> <p><b>&lt;EP&gt;</b> - 8 bit hexadecimal number, endpoint of a remote device.</p> <p><b>&lt;SendMode&gt;</b> - One digit to be set to choose transmission mode, input 0 – means sending command directly.</p> <p><b>&lt;ClusterID&gt;</b> - 16 bit hexadecimal number representing the cluster ID. As this AT command is created for responding OTA cluster request, the user should use 0x0019 as cluster ID.</p> <p><b>&lt;CmdID&gt;</b> - 8 bit hexadecimal number. It is the command ID that the default response is responding to. It may be 0x03 for imageBlockRequest or 0x04 for imagePageRequest.</p> <p><b>&lt;Status&gt;</b> - 8 bit hexadecimal number representing the response status.</p> <p>No matter which status code is used, the user need to feed the &lt;Seq&gt;.</p> <p><b>&lt;Seq&gt;</b> - 8 bit hexadecimal number, this is the sequence number of the response. The user should use the sequence number contained in the IMGBREQ or IMGPREQ prompt.</p>	<p>Response</p> <p><b>OK</b></p> <p>or</p> <p><b>ERROR:&lt;errorcode&gt;</b></p> <p>This command does not have response prompt.</p> <p><i>Note:</i></p> <p>This command is specifically created for OTA cluster implementation.</p> <p>The only usage scenario will be that the host loses the OTA file after an OTA upgrading progress has started. In that case, the user may apply this command to respond to an imageBlockRequest or imagePageRequest to stop an OTA upgrade process.</p> <p>An example command can be:</p> <p>AT+DFTRSP:53F3,01,0,0019,04,98,06</p>

### 3.5.7 OTA Server Configuration Commands

#### **+GETPOLICY** - Print the policy of local OTA server

Execute Command  
**AT+GETPOLICY**

Response

**OK**

**OTAPOLICY:<QueryPolicy>,<UpgradeRequestPolicy>**

**<QueryPolicy>** - 8 bit hex

00: Upgrade if server has newer

01: Downgrade if server has older

02: Reinstall if server has same

03: No next version

**<UpgradeRequestPolicy>**

00: Upgrade Now

01: Upgrade In a few minutes

02: Ask me later to upgrade

03: Abort upgrade

#### **+SETPOLICY** - Print the policy of local OTA server

Execute Command  
**AT+SETPOLICY:<QueryPolicy>,<UpgradeRequestPolicy>**

Response

**OK**

**<QueryPolicy>**

00: Upgrade if server has newer

01: Downgrade if server has older

02: Reinstall if server has same

03: No next version

**<UpgradeRequestPolicy>**

00: Upgrade Now

01: Upgrade in 2 minutes

02: Ask me later to upgrade

03: Abort upgrade

This command does not have response prompt.

**+SETIMGID** - *Set the OTA server image id and some other relevant image information*

Execute Command

**AT+SETIMGID:<ManufCode>,<ImgType>,<FwVersion>,<SpecificNodeId>,<ImageSize>**

**<ManufCode>** - 16 bit hexadecimal number representing manufacture code.

**<ImgType>** - 16 bit hexadecimal number representing image typeid.

**<FileVersion>** - 32 bit hexadecimal number. It indicates the image version of the server equipped firmware.

**<SpecificNodeId>** - 16 bit hexadecimal, the user can specify this parameter to indicate the equipped OTA file is for a specific device.

**<ImageSize>** - 32 bit hexadecimal number. It indicates the image size of the server equipped firmware. This parameter is used in the QueryNextImageResponse command.

**Note:** this command should be used to set the imageid which includes manufacture ID, image type ID and firmware version. These parameter will be used to filter out the queryNextImageRequest from the devices which are not supposed to be upgraded.

These settings will take effect when the user set S62.

Response

**OK**

This command does not have response prompt.

**+GETIMGID - Print the OTA server image id and hardware version**

Execute Command

**AT+GETIMGID**

Response

**OTAIMGID:<ManufCode>,<ImgType>,<FwVer>,<MinHwVer>,<MaxHwVer>,<HostDefinedNodeId>,<ImageSize>**

**OK**

**<ManufCode>** - 16 bit Hex. The value shall be the device's assigned manufacturer code.

**<ImgType>** - 16 bit Hex, OTA file's image type id

**<FwVer>** - 32 bit Hex, represents the server OTA image's firmware version.

**<MinHwVer>** - 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.

The high byte represents the version and the low byte represents the revision.

**<MaxHwVer>**- 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.

The high byte represents the version and the low byte represents the revision.

The **<MinHwVer>** and **<MaxHwVer>** can be set by using the **AT+SETHWVER** command.

**Note: on the client side, the hardware version of the device should not be earlier than the minimum (hardware) version and should not be later than the maximum (hardware) version in order to run the OTA upgrade file.**

**<HostDefinedNodeId>** - 16 bit hex. This firmware provides a feature to allow the host to configure the client's node id in order to provide an image to a specific node (OTA client) for downloading. This parameter can be set by using AT+SETIMGID command.

**<ImageSize>** - 32 bit Hex, represents the server OTA image's size

**+SETHWVER - Set the allowable OTA client's hardware version range**

Execute Command

**AT+SETHWVER:<MinHwVer>,<MaxHwVer>**

Response

**OK**

**<MinHwVer>** - 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.

The high byte represents the version and the low byte represents the revision.

**<MaxHwVer>**- 16 bit Hex, the value represents the earliest hardware platform version this image should be used on.

This command does not have response prompt.

### 3.5.8 Poll Control Cluster Commands

#### **+FPSET - Set Local Flag and Timeout for Starting Fast Poll**

Execute Command

**AT+FPSET:<FastPollFlag>,<Timeout>**

**<FastPollFlag>** - Boolean type, set to 01 for instructing the poll control server to do fast poll, whereas 00 is for not doing fast poll.

**<Timeout>**- 16 bit Hex, the value represents the fast polling period. If the FastPollFlag is set to 00, the poll control server device will omit the timeout which is set and sent in checkInResponse command.

This command is used to set local flag and timeout which will be used in checkInResponse command.

Note:

If S63 is used to disable the automatic Check-in response, the CICIE will not send response when it receives a check-in command.

Response

**OK**

This command does not have response prompt.



#### **+LPINTVL - Set Long Poll Interval**

Execute Command

**AT+LPINTVL:<NodeID>,<EP>,<SendMode>,<LongPollInterval>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID

**<EP>** - 8 bit hexadecimal number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group

**<LongPollInterval>** - 32 bit hexadecimal number representing the longPollInterval to be set on the poll control server cluster.

This command can be sent to a poll control server device to configure the long poll interval.

Response

**OK**

or

**ERROR:<error\_code>**

If the set value is not acceptable, the Poll Control Server will send back a default response of INVALID\_VALUE (0x87)

#### **+FPSTOP - Stop Fast Poll**

Execute Command

**AT+FPSTOP:<NodeID>,<EP>,<SendMode>**

This command can be sent to a poll control server device to quit fast poll mode.

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID

**<EP>** - 8 bit hex number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group

Response

**OK**

or

**ERROR:<error\_code>**

#### **+SPINTVL** - *Set Short Poll Interval*

Execute Command

**AT+SPINTVL:<NodeID>,<EP>,<SendMode>,<ShortPollInterval>**

**<NodeID>** - 16 bit hexadecimal number, network address of a remote device. It can be a node's network address or a group ID

**<EP>** - 8 bit hex number, endpoint of a remote device.

**<SendMode>** - One digit to be set to choose transmission mode, input 0 – means sending command directly; 1 – means sending command to a group

**<ShortPollInterval>** - 16 bit hexadecimal number representing the longPollInterval to be set on the poll control server cluster.

This command can be sent to a poll control server device to set the short poll interval.

Response

**OK**

or

**ERROR:<error\_code>**

If the set value is not acceptable, the Poll Control Server will send back a default response of INVALID\_VALUE (0x87)

#### 4 List of Status codes

00	Everything OK - Success
01	Fatal Error
02	Unknown command
04	Invalid S-Register
05	Invalid parameter
06	Recipient could not be reached
07	Message was not acknowledged
0A	Message could not be sent
0C	Too many characters
0D	License Problem
0E	PAN could not be established because duplicate PAN ID was detected
0F	Fatal error initialising the network
10	Error bootloading
12	Fatal error initialising the stack
18	Node has run out of Buffers
19	Trying to write read-only register
20	Invalid password
25	Cannot form network
27	No network found
28	Operation cannot be completed if node is part of a PAN
2C	Error leaving the PAN
2D	Error scanning for PANs
33	No response from the remote boot loader
35	Binary data input timeout
39	MAC transmit queue is full
6C	Invalid binding index
66	Message is not sent to the target successfully
70	Invalid operation
72	The maximum number of in-flight messages has been reached
74	Message too long
80	ZDP invalid request type (if ZDO command has been sent)
81	ZDP device not found (if ZDO command has been sent)
82	ZDP invalid endpoint (if ZDO command has been sent)
83	ZDP not active (if ZDO command has been sent)
84	ZDP not supported (if ZDO command has been sent)
91	Operation only possible if joined to a PAN
93	Node is not part of a Network
94	Cannot join network
96	Mobile End Device Move to new Parent Failed
98	Cannot join ZigBee 2006 Network as Router
A1	More than 8 broadcasts were sent within 8 seconds
A6	Error in trying to encrypt at APS level. No link key entry in the table for the destination
AB	Trying to join, but no beacons could be heard
AC	Network key was sent in the clear when trying to join secured
AD	Did not receive Network Key
AE	No Link Key received
AF	Preconfigured Key required
B1	Not attached to a meter
B2	ESI end point not known
C5	NWK already present
C7	NWK table full
C8	NWK unknown device

**Extra code for ZigBee Cluster (ZCL) Command**

Please note: ZCL commands are referred to the commands in section 3.5.1.5 and section 3.5.2. The following code will be shown in ZCL command responses (e.g read attribute response, add group response, etc) or Default responses.

- 80 Malformed Command**
- 81 Unsupported Cluster Command**
- 82 Unsupported General Command**
- 83 Unsupported Manufacturer Cluster Command**
- 84 Unsupported Manufacturer General Command**
- 85 Invalid Field**
- 86 Unsupported Attribute**
- 87 Invalid Value**
- 88 Read Only**
- 89 Insufficient Space**
- 8A Duplicate Exists**
- 8B Not Found**
- 8C Unreportable Attribute**
- 8D Invalid Data Type**

## 5 S-Registers

Most S-Registers of the modules can be read and written. All s-registers are stored in non-volatile memory and will keep their user defined settings unless reset to the factory defaults using the “AT&F” command. The S-Registers used in this firmware are summarized in the table below.

S-Register Overview		Local R/W
<b>S00</b>	Channel Mask	(•/•)
<b>S01</b>	Transmit Power Level	(•/•)
<b>S02</b>	Preferred PAN ID	(•/•)
<b>S03</b>	Preferred Extended PAN ID	(•/•)
<b>S04</b>	Local EUI	(•/-)
<b>S05</b>	Local NodeID	(•/-)
<b>S06</b>	Parent's EUI	(•/-)
<b>S07</b>	Parent's NodeID	(•/-)
<b>S08</b>	Network Key1	(-/•)
<b>S09</b>	Link Key1	(-/•)
<b>S0A</b>	Main Function1	(•/•)
<b>S0B</b>	User Readable Name 1	(•/•)
<b>S0C</b>	Password1	(-/•)
<b>S0D</b>	Device Information	(•/-)
<b>S0F</b>	Prompt Enable	(•/•)
<b>S12</b>	UART Setup	(•/•)
<b>S48</b>	Endpoint 1 Profile ID	(•/•)
<b>S49</b>	Endpoint 1 Device ID	(•/•)
<b>S4A</b>	Endpoint 1 Device Version	(•/•)
<b>S4B</b>	Endpoint 1 Input Cluster List	(•/•)
<b>S4C</b>	Endpoint 1 Output Cluster List	(•/•)
<b>S4E</b>	End Device Poll Timeout	(•/•)
<b>S4F</b>	MAC Timeout	(•/•)
<b>S60</b>	Manufacturer Code	(•/•)
<b>S61</b>	IAS Enrol Control	(•/•)
<b>S62</b>	OTA Upgrade Control	(•/•)
<b>S63</b>	Enable RSSI,LQI printing and Check-in response control	(•/•)
<b>S64</b>	Licence Code	(•/•)
<b>S65</b>	Protocol ID	(•/•)
<b>S66</b>	Server Tunnel ID	(•/•)

**Table 9: S-Register Overview (Note: “1” means password Protected Registers)**

## 5.1 S-Registers for Network Setup

### S00 – Channel Mask

Description	Parameters
The 802.15.4 channel mask.	<b>XXXX</b>
Operations	Where XXXX represents a 16-bit decimal number enabling IEEE 802.15.4 channel numbers 11 to 26. Writing a bit to 1 enables a channel and subsequently writing a bit to 0 disables a channel for scanning, joining and establishing networks. e.g. when setting S00 to 0001, only channel 11 will be used for all following operations.
<b>R/W LOCAL</b>	
Becomes effective	
when Joining, Scanning or establishing a PAN	
Note	Range
The channel mask does not affect the AT+JPAN command	<b>0001 - FFFF</b>
Storage	Factory Default
	<b>FFFF</b>
<b>Non-Volatile</b>	

### S01 – Transmit Power Level

Description	Parameters
The device's transmit power level in dBm.	<b>snn</b>
Operations	Where snn represents a signed 8-bit decimal number.
<b>R/W LOCAL</b>	
Notes	Range
The output power of the "-PA" and "-LR" variants is higher than the value in S01. Please refer to the respective hardware manuals.	<b>ETRX3: 8 to -43</b>
	<b>ETRX3 LRS Variants: -7 to -43</b>
Becomes effective	Actual values are {8, 7, 6, 5, 4, 3, 2, 1, -1, -2, -3, -4, -5, -6, -7, -8, -9, -11, -12, -14, -17, -20, -26, -43} Entering a value not on this list (such as -19) will result in the next lowest output power.
<b>When Joining or establishing a PAN</b>	Entering a value higher than 3 will automatically enable boost mode.
Storage	Factory Default
	<b>8 or (-7 for LRS)</b>
<b>Non-Volatile</b>	

<b>S02 – Preferred PAN ID</b>	
Description	Parameters
The 802.15.4 PAN ID.	<b>&lt;PANID&gt;</b>
Operations	Where <PANID> represents a 16-bit hexadecimal number
<b>R/W LOCAL</b>	Range
Becomes effective	<b>0000 – FFFF</b>
<b>When Joining or establishing a PAN</b>	When establishing a PAN the coordinator will pick a random PAN ID if S02 is set to 0000. If set to any value between 0001 and FFFF this number will be used as PAN ID instead, unless trying to use a PAN ID which already exists on the same channel. In this case a random PAN ID will be used instead.
Notes	When joining only a PAN with the ID stored in S02 will be joined unless S02 is set to 0000. In this case the next best PAN which allows joining is joined.
Two networks operating on the same channel with the same PAN ID, but a different EPANID are detected to be in conflict with each other. PAN ID conflicts are detected by the stack and resolved by one of the networks dynamically changing its PAN ID.	Factory Default
The preferred PAN ID in S02 does not affect the AT+JPAN command	<b>0000</b>
Storage	
<b>Non-Volatile</b>	

**S03 – Preferred Extended PAN ID**

Description	Parameters
The extended PAN ID.	<b>&lt;EPANID&gt;</b>
Operations	Where <EPANID> represents a 64-bit hexadecimal number
<b>R/W LOCAL</b>	Range
Becomes effective	<b>0000000000000000 – FFFFFFFF</b>
<b>When Joining or establishing a PAN</b>	When establishing a PAN the coordinator will pick a random EPANID if S03 is set to all 0's. If set to any other value this number will be used as the EPANID instead.
<i>Note</i>	When the command AT+JN is used to join a PAN the module will check if the PAN it finds has the EPID which matches the value in S03. If the value in S03 is set to 0 then there will be no EPID check. By default, S03 is set at all zeros
The EPANID is used for PAN ID conflict detection. It is therefore recommended to use a random EPANID at all times.	
The preferred EPANID does not affect the AT+JPAN command	
Storage	Factory Default
<b>Non-Volatile</b>	<b>0000000000000000</b>

**S04 – Local EUI64**

Description	Parameters
The local node's unique EUI64 identifier.	<b>&lt;EUI64&gt;</b>
Operations	Range
<b>R LOCAL</b>	<b>0000000000000000 – FFFFFFFF</b>
Storage	Factory Default
<b>Non-Volatile</b>	<b>&lt;unique number&gt;</b>



**S05 – Local 16-Bit NodeID**

Description	Parameters
The local node's 16-bit NodeID.	<b>&lt;NodeID&gt;</b>
<i>Note</i>	Range
Reading this register while not associated with a network will result in an undefined return value.	<b>0000-FFFF</b>
Operations	Factory Default
<b>R LOCAL</b>	<b>n/a</b>
Storage	
<b>Non-Volatile</b>	

**S06 – Parent's EUI64**

Description	Parameters
The parent node's unique EUI64 identifier.	<b>&lt;EUI64&gt;</b>
<i>Note</i>	Range
The return value is undefined for nodes without parents (coordinator and nodes that are not joined to a network)	<b>0000000000000000 – FFFFFFFFFFFFFFFF</b>
Operations	Factory Default
<b>R LOCAL</b>	<b>n/a</b>
Storage	
<b>Non-Volatile</b>	

### **S07 – Parent’s 16-Bit NodeID**

Description	Parameters
The parent node’s 16-bit NodeID.	<b>&lt;NodeID&gt;</b>
Operations	Range
<b>R LOCAL</b>	<b>0000-FFFF</b>
Note	Factory Default
The return value is undefined for nodes without parents (coordinator and nodes that are not joined to a network)	<b>n/a</b>
Storage	
<b>Non-Volatile</b>	

### **S08 – Network Key**

Description	Range
The network key which can be written using the password. The default password is “password”.	<b>From 0 to 2<sup>128</sup>-1</b>
Operations	The 128-bit AES network key in hexadecimal representation (32 characters).
<b>W LOCAL</b>	This register has been set to all 0’s (default) and a random network key is generated when establishing a PAN. Please set it when use silent join (AT+SJN).
Write operation	Factory Default
<b>ATS08=&lt;key&gt;:&lt;password&gt;</b>	
Storage	<b>00000000000000000000000000000000</b>
<b>Non-Volatile</b>	

### **S09 – Trust Centre Link Key**

Description	Range
<p>The link key which can be written using the password. The default password is “password”.</p> <p>Operations</p> <p><b>W LOCAL</b></p> <p>Write operation</p> <p><b>ATS09=&lt;key&gt;:&lt;password&gt;</b></p> <p>Becomes effective</p> <p><b>When Joining or establishing a PAN</b></p> <p>Storage</p> <p><b>Non-Volatile</b></p>	<p><b>From 0 to 2<sup>128</sup>-1</b></p> <p>The 128-bit trust centre link key in hexadecimal representation (32 characters).</p> <p>When set to all 0s (default) a random trust centre link key is generated when establishing a PAN.</p> <p>Factor Default</p> <p><b>5A6967426565416C6C69616E63653039</b></p>

## 5.2 S-Registers for Module Setup

<b>S0A – Main Function</b>	
<i>Description:</i>  Defines the behaviour of the Device.	Parameters  <b>XXXX</b>  Where XXXX represents a 16-bit hexadecimal number.
<i>Operations:</i>  <b>R/W LOCAL</b>	Range  <b>0000 to FFFF</b>
<i>Write operation:</i>  <b>ATS0A=XXXX:&lt;Password&gt;</b>	<b>Bit 0:</b> Set to not allow other nodes to join the network. Please use <b>AT+PJOIN</b> to allow other node to join HA network.  <b>Bit 4:</b> Set to Send Network key unencrypted to joining nodes
<i>Note:</i>  For security reasons this register is password protected. The default password is "password".	<b>Bit 7:</b> Device uses S09 stored preconfigured link key when joining. By default, S09 stores HA link key, which can be used in joining or silent join.  The user can specify a link key and use it in forming/joining a network.
<i>Storage:</i>  <b>Non-Volatile</b>	<b>Other Bits:</b> Reserved  Factory Default  <b>0001</b>

**S0B – User Readable Name**

<i>Description</i>	<i>Parameters</i>
<p>Password protected user defined name which can be used to identify the node</p> <p><i>Operations</i></p> <p><b>R/W LOCAL</b></p> <p><i>Write operation</i></p> <p><b>ATS0B=&lt;name&gt;:&lt;password&gt;</b></p> <p>Becomes effective</p> <p><b>Instantly.</b></p> <p><i>Storage</i></p> <p><b>Non-Volatile</b></p>	<p><b>0123456789ABCDEF</b></p> <p>Name with up to 16 characters.</p>

**S0C – Password**

<i>Description</i>	<i>Parameters</i>
<p>The local node's password.</p> <p><i>Operations</i></p> <p><b>W LOCAL</b></p> <p><i>Write operation</i></p> <p><b>ATS0C=&lt;NEW&gt;:&lt;OLD&gt;</b></p> <p>Becomes effective</p> <p><b>Instantly.</b></p> <p><i>Storage</i></p> <p><b>Non-Volatile</b></p>	<p><b>12345678</b></p> <p>8 case sensitive characters (8 bytes).</p> <p><b>Note that the password must have exactly 8 characters.</b></p> <p>Factory Default <b>password</b></p>

**S0D – Firmware Revision**

<i>Description</i>	Parameters
String containing the module's order code and firmware revision.	ccc...ccc
	Text string
<i>Operations</i>	Example
<b>R LOCAL</b>	CI R300
<i>Storage</i>	
<b>Non-Volatile</b>	Factory Default
	<b>N/A</b>

**S0F – Prompt Enable**

<i>Description</i>	Parameters
Enable/Disable default response prompts and prompts for message from other profile.	<b>XXXX</b>
	Where XXXX represents a 16-bit hexadecimal number.
<i>Operations</i>	Range
<b>R/W LOCAL</b>	<b>0000 to FFFF</b>
Becomes effective	<b>Bit A:</b> Set: Show received default response
<b>Instantly</b>	<b>Bit 9:</b> Set: Show received raw messages. It can be used to display unhandled message with Non-HA profile ID
<i>Notes</i>	<b>All the other bits are reserved</b>
Use of bit 8 requires bit 1 to be unset	
<i>Storage</i>	Factory Default
<b>Non-Volatile</b>	<b>0400</b>

**S48 – Endpoint 1 Profile ID**

Description	Parameters
Register defining the Profile ID of the Node's endpoint 1.	<b>XXXX</b>
Operations	The 16 bit Profile ID.
<b>R/W LOCAL</b>	Factory Default
Becomes effective	<b>0000</b>
<b>Instantly</b>	<b>Note:</b> if the user use default value, the CICIE will use 0x0104 as default Profile ID.
Storage	
<b>Non-Volatile</b>	

**S49 – Endpoint 1 Device ID**

Description	Parameters
Register defining the Device ID of the Node's endpoint 1	<b>XXXX</b>
Operations	The 16 bit device ID.
<b>R/W LOCAL</b>	Factory Default
Becomes effective	<b>0000</b>
<b>Instantly</b>	<b>Note:</b> if the user use default value, the CICIE will use 0x0007 as default Device ID for endpoint 1.
Storage	
<b>Non-Volatile</b>	

**S4A – Endpoint 1 Device Version**

Description	Parameters
Register defining the version of the device attached to the Node's endpoint 1	<b>00XX</b>
Operations	The 8 bit version preceded by two leading 0's.
<b>R/W LOCAL</b>	Factory Default
Becomes effective	<b>0000</b>
<b>Instantly</b>	<b>Note:</b> if the user use default value, the CICIE will use 0x0002 as default Device version number for endpoint 1.
Storage	
<b>Non-Volatile</b>	

**S4B – Endpoint 1 Input Cluster List**

Description	Parameters
Register defining the input cluster list of the device attached to the Node's endpoint 1	<b>[XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX]</b>
Operations	A list of a maximum of 4 x 16 bit cluster IDs separated by commas. The list may have any length from 0 to 4 clusters
<b>R/W LOCAL</b>	Example ATS4B=0000,0003
Becomes effective	Factory Default
<b>Instantly</b>	<b>NULL</b>
Storage	<b>Note:</b> if the user use default value, the CICIE ep1 will use its original configurations and supports clusters listed in section 2.2.2.
<b>Non-Volatile</b>	



**S4C – Endpoint 1 Output Cluster List**

Description	Parameters
Register defining the output cluster list of the device attached to the Node's endpoint 2	<b>[XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX]</b>
Operations	A list of a maximum of 21 x 16 bit cluster IDs separated by commas. The list may have any length from 0 to 21 clusters
<b>R/W LOCAL</b>	
Becomes effective	Factory Default
<b>Instantly</b>	<b>NULL</b>
Storage	<b>Note:</b> if the user use default value, the CICIE ep1 will use its original configurations and supports clusters listed in section 2.2.2.
<b>Non-Volatile</b>	

**S4E – End Device Poll Timeout**

Description	Parameters
Register in a parent node defining the amount of time after which an SED or ZED times out of its parent's child table if it has not polled	<b>XXYY</b>  Where YY is the timeout in seconds left-shifted by XX ( $YY * 2^{XX}$ ). The default number results in a timeout of 5 Minutes, whereas the maximum number results in a timeout of approximately 48 days.
Operations	
<b>R/W LOCAL</b>	
Becomes effective	Maximum
<b>After Soft or Hard Reset</b>	<b>0EFF</b>
Storage	Factory Default
<b>Non-Volatile</b>	<b>0605</b>

**S4F – MAC Timeout**

Description Register defining the MAC timeout	Parameters <b>XXXX</b>
Operations <b>R/W LOCAL</b>	The Ember_Indirect_Transmission_Timeout is the amount of time in milliseconds that the MAC in a parent node will hold a message for indirect transmission to a child. In addition to this is also the basis for the timeout after which an acknowledged unicast to an end device is nacked when no ACK is received.
Becomes effective  <b>After Soft or Hard Reset</b>	The default value is 7680ms. The maximum is 30s
Storage  <b>Non-Volatile</b>	Maximum <b>7530</b>  Factory Default <b>1E00</b>

**S60 – Manufacturer Code**

Description  String containing the manufacture code which is used to define manufacturer defined clusters and attributes.	Parameters cccc Text string
Operations  <b>R/W LOCAL</b>	Example <b>1010</b>
Storage  <b>Non-Volatile</b>	Factory Default <b>N/A</b>

## **S61 – IAS Enrol Control**

Description	Parameters
Set to enable IAS Zone Device enrol	<b>XXXX</b>
<i>Operations</i>	XXXX is a 16-bit hexadecimal number.
<b>R/W LOCAL</b>	<b>Bit C:</b> When set this bit to 1, the CIE will clear flash maintained zone table entries during software reset.
<i>Storage</i>	<b>Bit 8:</b> When this bit is set, the user need to use AT+RAWZCLP command to respond to ACE client commands. When relevant prompts shows.
<b>Non-Volatile</b>	GETPANELSTATUS:<NodeID>,<EP>,<Sequence>
<i>Note:</i>	GETBYPASSZLIST:<NodeID>,<EP>,<Sequence>
Please check Appendix B for more information.	GETZSTATUS:<NodeID>,<EP>,<StartingZoneId>,<maxNumberOfZoneIds>,<zoneStatusMaskFlag>,<zoneStatusMask>,<Sequence>
	<b>Bit 6:</b> When set this bit to 1, the user need to handle ACE Get Zone Information command if Zone ID is larger than 31. (see Appendix B)
	<b>Bit 5:</b> When set this bit to 1, the CIE will allow zone device to enrol even when the Zone table is full.
	<b>Bit 4:</b> When set this bit to 1, the CIE will allow zone devices with user specified zone types to be enrolled.
	<b>Bit 3:</b> When set this bit to 1, the CIE will only allow zone devices with ZigBee Cluster Library defined zone type to be enrolled.
	<b>Bit 2:</b> When set this bit to 1, the CIE will check the Zone Type of the received <i>Enrol Request</i> , then decide if the Zone device is allowed to enrol. If the Zone Type is neither ZigBee Spec nor the user specified, Enrol request is rejected.
	AT+MZONETYPE can be used to add user specified zone type in order to allow a device with user defined zone type to be enrolled.
	<b>Bit 1:</b> When set this bit to 1, the CIE will check local Zone Table and only allow the device which has been resisted in the Zone Table to enrol.
	The user can use AT+ADDZENTRY to configure local zone table before commissioning one or a

group of IAS devices, then set this bit for commissioning.

**Bit 0:** Set: to 1, the CIE will permit enrol when a Zone device joins the network and sends a enrol request to the CIE. Please note if this bit is set to 0, other bits of this S-register setting will not take effect.

**All the other bits are reserved**

Factory Default

**0001**

**S62 – OTA Upgrade Control**

Description	Parameters
<p>This S-register will be used as a switch to determine whether the CICIE firmware will handle the OTA client commands or pass the commands to host application.</p>	<p><b>XXXX</b></p> <p>Where XXXX represents a 16-bit hexadecimal number.</p> <p>Bit E - if this bit is set, the disableDefaultResponse bit in frame control is set to TRUE in the constructed imageBlockResponse</p>
<p>Operations</p>	
<p><b>R/W LOCAL</b></p>	<p>Bit 0 – set this bit to 1 for instructing the CICIE to handle queryNextImageRequest command sent by the OTA clients, which means the CICIE firmware will automatically respond to the request and filter out the request from the clients which are not supposed to be upgraded. The CICIE will send a response with IMAGE_NOT_AVAILABLE to those devices.</p>
<p>Becomes effective</p>	
<p>Instantly.</p>	<p>If this bit is not set, the CICIE firmware will print out an IMGQUERY prompt to notify the host application and wait for the host to handle the request. In that case the host is required to contracture an AT+QIMGRSP command and send it back to the OTA client.</p>
	<p>Bit 1 – set this bit to 1 for instructing the CICIE to handle updateEndRequest command sent by the OTA clients, which means the CICIE firmware will automatically respond to the request with the user pre-set response parameters. If this bit is set, the user need to use <b>AT+SETPOLICY</b> to specify the upgrade policy. By default, the upgrade policy is set to UPDATE_NOW indicating the client should update itself immediately.</p>
	<p>If this bit is not set, the CICIE firmware will print out an <b>UPGRADEREQ</b> prompt to notify the host application and wait for the host to handle the request. In that case the host is required to contracture an AT+UPGRADE command and send it back to the OTA client.</p>
	<p>By default, the CICIE will not wait for the Host application to construct response and send back to the requesting device automatically. But the user should check and set the policy and image ID using corresponding AT commands</p>
	<p>Please note: If Bit 0 and 1 is not set to 1, the CICIE firmware will respond to the receiving request command. But the <b>IMGQUERY</b> and <b>UPGRADEREQ</b> prompts will still show when the requests are handled and responses are sent to client.</p>
	<p>Factory Default</p>
	<p><b>0003</b></p>

**S63 – Enable RSSI and LQI printing**

Description	Parameters
<p>This S-register is used to enable/disable the RSSI and LQI value shown in the incoming message prompt.</p> <p><i>Operations</i></p> <p><b>R/W LOCAL</b></p> <p>Becomes effective</p> <p>Instantly.</p>	<p><b>XXXX</b></p> <p>Where XXXX represents a 16-bit hexadecimal number.</p> <p><b>Bit 0</b> – set this bit to 1, then the incoming message prompts will include the RSSI and LQI value at the end. For instance, the read attribute response prompt will be:</p> <p>RESPATTR:&lt;NodeID&gt;,&lt;EndPoint&gt;,&lt;ClusterID&gt;,&lt;AttributeID&gt;,&lt;Status&gt;,&lt;AttributeValue&gt;,&lt;RSSI&gt;,&lt;LQI&gt;</p> <p><b>Bit 1</b> – set this bit to 1, then +PANSCAN prompt will have RSSI and LQI reading of the Beacon at the end.</p> <p><b>Bit 2</b> – set this bit to 1 to disable automatic Check-in response sent by CICIE. If this bit is set, CHECKIN prompt will show the sequence number of the check-in command, in that case the host shall use this sequence number in its constructed Check-in command and send to the originator node.</p> <p>Please note: CICIE will automatically send check-in response by default. The host shall apply AT+FPSET to set the payload for the check-in response.</p> <p>Factory Default</p> <p><b>0000</b></p>

**S64 – License Key**

Description	Parameters
<p>This S-register is used to store the license key. This key is used to activate the firmware.</p> <p>Please check document TG-APP-0022 Copy Protection Release for more information</p> <p><i>Operations</i></p> <p><b>R/W LOCAL</b></p> <p>Becomes effective</p> <p>Instantly.</p>	<p><b>XXXXXXXXXXXXXXXXXXXX</b></p> <p>Where <b>XXXXXXXXXXXXXXXXXXXX</b> represents a 128-bit hexadecimal number.</p> <p>Factory Default</p> <p><b>000000000000000000</b></p>

**S65 – Protocol ID**

<i>Description</i>	Parameters
8 bit hexadecimal number represent protocol ID for using tunnel.	01
<i>Operations</i>	
<b>R/W LOCAL</b>	Factory Default
<i>Storage</i>	<b>C8</b>
<b>Non-Volatile</b>	

**S66 – Server Tunnel ID**

<i>Description</i>	Parameters
16 bit hexadecimal number represent server Tunnel ID, which will be used when a client request tunnel from the CICIE.	0000
<i>Operations</i>	
<b>R/W LOCAL</b>	Factory Default
<i>Storage</i>	<b>5747</b>
<b>Non-Volatile</b>	

### 5.3 I/O related S-Registers

<b>S12 – UART Setup</b>	
<i>Description</i>	Parameters
The device's RS232 Baud rate and mode.	<b>XXXX</b>
The default setting of 0500 results in: 19200bps, no parity, 1 stop bit, 8 data bits.	Where XXXX represents a 16-bit hexadecimal number.
<i>Operations</i>	Range of the most significant byte
<b>R/W LOCAL</b>	<b>00 to 0C</b>
Becomes effective	00: 1200 baud 01: 2400 baud 02: 4800 baud 03: 9600 baud 04: 14400 baud 05: 19200 baud 06: 28800 baud 07: 38400 baud 08: 50000 baud 09: 57600 baud 0A: 76800 baud 0B: 100000 baud 0C: 115200 baud
Instantly.	
<i>Note</i>	
Note that when the 128-byte output buffer of the module is full data will be dropped.	
<i>Storage</i>	
<b>Non-Volatile</b>	<b>00 to FF</b> bit 6-7 Reserved bit 5 set: H/W flow control enabled bit 4 set: no command echo bit 3 set: 7 data bits instead of 8 bit 2 set: 2 stop bits instead of one bit 1 set: odd parity enabled bit 0 set: even parity enabled
	Factory Default
	<b>0500</b>



## 6 Interpreting RSSI Energy Levels

The readings from “AT+ESCAN” represent the hexadecimal numbers with offset by +127 to make them positive numbers.

The RSSI is calculated over an 8-symbol period as well as at the end of a received packet. The algorithm utilizes the RX gain settings and the output level of the ADC. The linear range of RSSI is specified to be 40dB over all temperatures. At room temperature, the linear range is approximately 60dB (-90 dBm to -30dBm).

## 7 Firmware upgrades

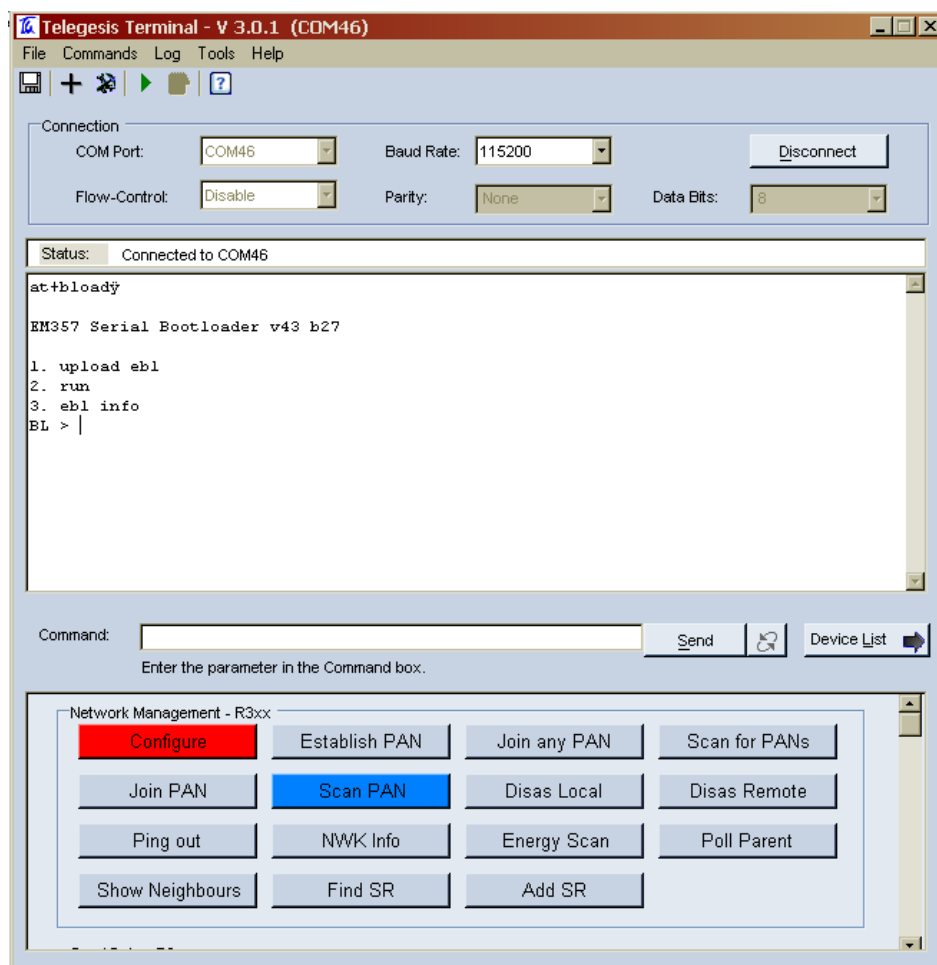
If required, the firmware of the ETRX3 modules can be upgraded serially.

### 7.1 Firmware Upgrades via Serial Port

In order to upgrade the firmware of the ETRX3 module using the serial bootloader, issue the “AT+BLOAD” command either by typing it in, or by pressing the respective button in the “Module Control” group of the Telegesis Terminal Application.

After entering the bootloader, the connection parameters need to be changed to 115200bps, 8 data bits, 1 stop bit, no parity, no flow control (providing that it is not already set to these values). This is achieved by pressing the ‘Disconnect’ button, changing the settings and then pressing the ‘Connect’ button. (If you only need to change the connection speed disconnecting and reconnecting is not required).

After pressing ‘Enter’, the bootloader menu will be shown in the terminal window as shown in Figure 1.



**Figure 1 Options in Bootload Mode**

Pressing '1' initiates the upload of the new firmware and a number of 'C' characters will indicate that the ETRX3 is ready to receive data. Within 60 seconds, select Tools / Transfer File... and browse for the new firmware file.

Firmware files for the ETRX3 will be in the .bl format. After checking that the protocol is set to XMODEM (128 Bytes), press the Send button and the new firmware will be downloaded as shown in Figure 2.

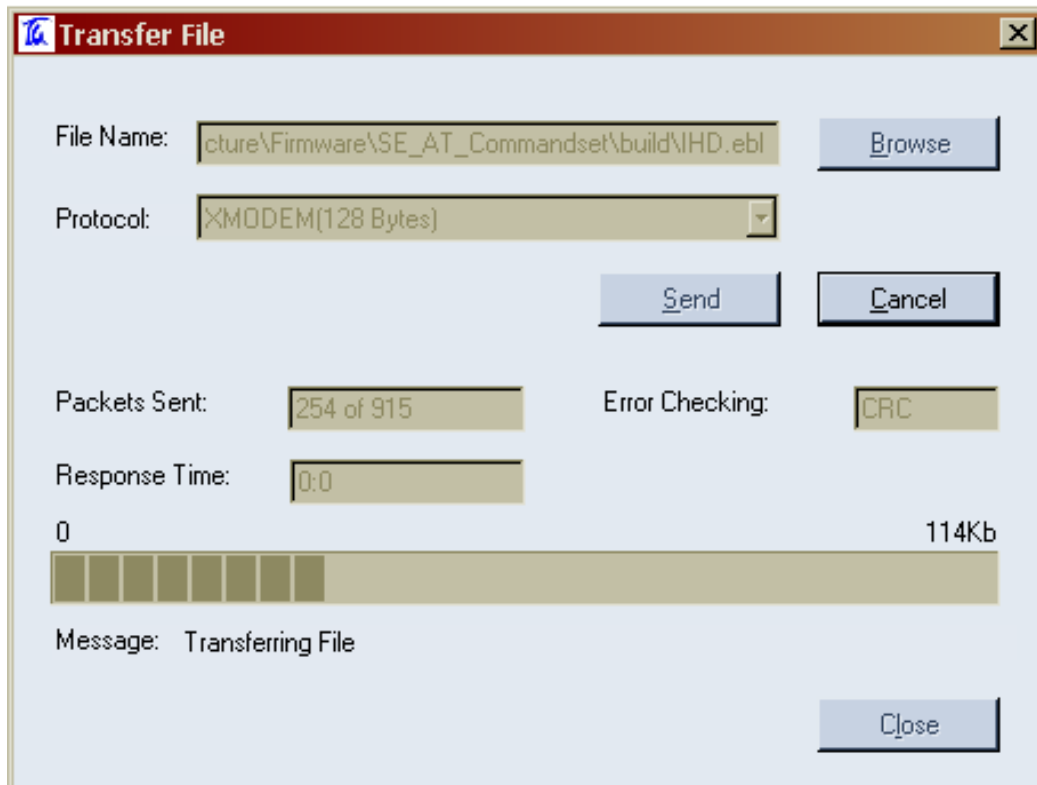


Figure 2 EBL Fill Transferred in Bootload Mode

When the transfer has been completed successfully, press Enter again in order to return to the bootloader menu (shown in figure 10) and option '2' to run the downloaded application software. If the application software has a baud rate other than 115200bps, this will need to be changed to the application baud rate as described above.

## 8 Appendix A

This Appendix includes information of prompts defined for user specified cluster commands/responses. The cluster commands are communicated with user specified profile (ProfileID: 0xC216).

Prompt Overview	
<p><b>LIFESIGN:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;,&lt;StatusFlag&gt;,&lt;MsTimer&gt;,[PsuVoltage],[Temperature],[Rssi],[Lqi],[SwitchMask],[SwitchStates]</b></p> <p>This prompt will show when CICIE R31x receives a life sign command.</p>	<p><b>&lt;NodeID&gt;-</b> 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;-</b> 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;-</b> 8 bit hexadecimal number representing the ZCL sequence number.</p> <p><b>&lt;StatusFlag&gt;-</b> 8 bit hexadecimal bitmap.</p> <p>Bit 0: If set, Battery Voltage is included in the lifesign command.</p>

	<p>Bit 1: If set, Temperature is included.</p> <p>Bit 2: If set, Switch Status is included.</p> <p>Bit 3: If set, LQI is included.</p> <p>Bit 4: If set, RSSI is included.</p> <p>Bit 5-7 reserved</p> <p><b>&lt;MsTimer&gt;</b>- 32 bit hexadecimal. Current reading from the millisecond timer.</p> <p><b>[PsuVoltage]</b>- 16 bit hexadecimal. Current power supply voltage reading in mV.</p> <p><b>[Temperature]</b> - 16 bit hexadecimal. Current temperature reading.</p> <p><b>[Rssi]</b>- Signed decimal number. Received Signal Strength Indication. dBm</p> <p><b>[Lqi]</b>- 8 bit hexadecimal, Link Quality Indicator.</p> <p><b>[SwitchMask]</b>- 8 bit hexadecimal. Availability mask of first 8 logical switches</p> <p><b>[SwitchStates]</b>- 8 bit hexadecimal. Logical activation state of first 8 logical switches</p>
<p><b>GETRTC:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;</b></p> <p>This prompt will show when CICIE R31x receives a get rtc command.</p>	<p><b>&lt;NodeID&gt;</b>- 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;</b>- 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;</b>- 8 bit hexadecimal number representing the ZCL sequence number.</p>
<p><b>FAULTREPORT:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;,&lt;ManufID&gt;,&lt;ModelID&gt;,&lt;FaultID&gt;</b></p> <p>This prompt will show when CICIE R31x receives a fault report command.</p>	<p><b>&lt;NodeID&gt;</b>- 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;</b>- 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;</b>- 8 bit hexadecimal number representing the ZCL sequence number.</p> <p><b>&lt;ManufID&gt;</b>- 16 bit hexadecimal number representing manufacturer ID.</p>

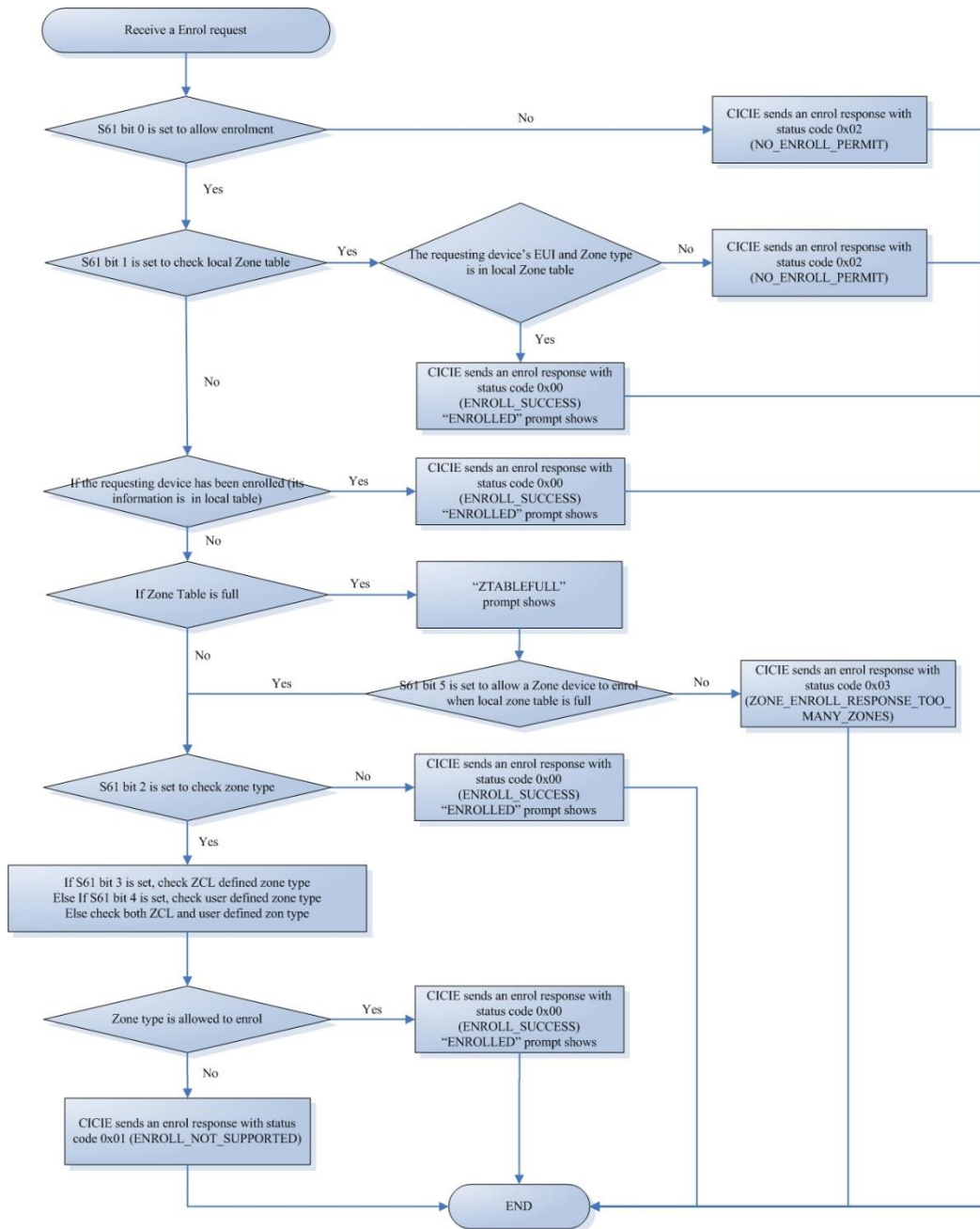
	<p><b>&lt;ModelID&gt;</b>- 16 bit hexadecimal number representing mode ID.</p> <p><b>&lt;FaultID&gt;</b>- 16 bit hexadecimal number representing fault ID.</p>
<p><b>STDOUT:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;,&lt;Message&gt;</b></p> <p>This prompt will show when CICIE R31x receives a std out command.</p>	<p><b>&lt;NodeID&gt;</b>- 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;</b>- 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;</b>- 8 bit hexadecimal number representing the ZCL sequence number.</p> <p><b>&lt;Message&gt;</b>- A null terminated ASCII hex string.</p>
<p><b>HELLORSP:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;,&lt;Address&gt;,&lt;EUI&gt;,&lt;ManufID&gt;,&lt;DeviceType&gt;,&lt;ReleaseCode&gt;,&lt;AppVer&gt;,&lt;HWMinorVer&gt;,&lt;HWMajorVer&gt;,&lt;MfgString&gt;,&lt;ModelString&gt;,&lt;DataCodeString&gt;</b></p> <p>This prompt will show when CICIE R31x receives a hello response.</p>	<p><b>&lt;NodeID&gt;</b>- 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;</b>- 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;</b>- 8 bit hexadecimal number representing the ZCL sequence number.</p> <p><b>&lt;Address&gt;</b>- 16 bit hexadecimal. Network node identification.</p> <p><b>&lt;EUI&gt;</b>- IEEE address of the source node.</p> <p><b>&lt;ManufID&gt;</b>- 16 bit hexadecimal. Manufacturer ID code.</p> <p><b>&lt;DeviceType&gt;</b>- 16 bit hexadecimal. Device type code.</p> <p><b>&lt;ReleaseCode&gt;</b>- 8 bit hexadecimal. Release code information for application code</p> <p><b>&lt;AppVer&gt;</b>- 8 bit hexadecimal. Version information for application code.</p> <p><b>&lt;HWMinorVer&gt;</b>- 8 bit hexadecimal. Version information for hardware platform (minor).</p> <p><b>&lt;HWMajorVer&gt;</b>- 8 bit hexadecimal. Version information for hardware platform (major).</p> <p><b>&lt;MfgString&gt;</b>- Character string. Manufacturer information string.</p>

	<p><b>&lt;ModelString&gt;</b>- Character string. Model information string.</p> <p><b>&lt;DataCodeString&gt;</b>- Character string. Date code string.</p>
<p><b>RANGETEST:&lt;NodeID&gt;,&lt;EP&gt;,&lt;Seq&gt;,&lt;Rssi&gt;,&lt;Lqi&gt;</b></p> <p>This prompt will show when CICIE R31x receives a range test command</p>	<p><b>&lt;NodeID&gt;</b>- 16 bit hexadecimal number representing source network address.</p> <p><b>&lt;EP&gt;</b>- 8 bit hexadecimal number representing the source endpoint.</p> <p><b>&lt;Seq&gt;</b>- 8 bit hexadecimal number representing the ZCL sequence number.</p> <p><b>[Rssi]</b>- Signed decimal number. Received Signal Strength Indication. dBm</p> <p><b>[Lqi]</b>- 8 bit hexadecimal, Link Quality Indicator.</p>

## 9 Appendix B

This Appendix provides some information about Zone Enrolment and example usage of supported AT commands and S register. Please note: this Appendix cannot cover all possible usage scenarios. The user is recommended to check ZigBee Cluster Library and ZigBee Home Automation Profile specification for more information.

From the Build B030714, CICIE R31x supports 16 flash maintained zone entries and 16 RAM maintained zone entries. Therefore it can allow up to 32 zone devices to enrol. In order to allow more devices to enrol, the user can set S61 bit 5. But in this situation, the CICIE will not be able to maintain the zone entry, the user needs to use external memory and maintain the zone entries with Zone ID larger than 31. The following figure illustrates how CICIE processes a zone enrol request and responds to it.



## Please note:

An IAS zone device needs to complete an enrolment process before it can work on a HA network. After joining a HA network, a zone device will send a enrol request to the IAS CIE device. The CIE will then respond to it with a status code. If the status code is success, the zone device will start working on network. Usually, it will periodically send a *Zone Status Change Notification* command to CIE. If the status code is not success, the zone device may choose to leave network.

Before sending enrol request, a zone device needs to have knowledge of CIE's IEEE address (IAS\_CIE\_Address attribute 0x0010). It is up to the zone's specific implementation to implement some auto-detect for CIE or require the intervention of a Configuration Tool (CT) or wait for CIE to notify the zone of its IEEE address.

If a zone device requires the intervention of a CT, the CT usually sets CIE's IEEE address in the zone device's local IAS\_CIE\_Address attribute 0x0010 before commissioning a zone device to join a network.

The followings provides two other instances of enrolling a zone device.

a> Zone Device (ZD) discovers CIE and auto-sets IAS\_CIE\_Address

1. CIE forms a network and permit join
2. ZD joins the network and broadcasts an announcement
3. ZD sends a broadcast match descriptor to search for Zone Client cluster (0x0500).
4. CIE responds to the match descriptor with its node ID and end point.
5. ZD sends IEEE address requests to CIE using CIE's node ID.
6. CIE sends IEEE address response to ZD.
7. ZD sets CIE's IEEE address to local IAS\_CIE\_Address attribute.
8. ZD sends an enrol request to CIE.
9. CIE responds to ZD with an enrol response.

b> CIE writes its IEEE Address to ZD's IAS\_CIE\_Address

- 1 CIE forms a network and enables permit join
- 2 ZD joins the network and broadcasts an announcement
- 3 CIE sends a broadcast match descriptor request to discover Zone Server cluster (0x0500).
- 4 ZD sends a match descriptor response with its node ID and endpoint.
- 5 CIE sends a write attribute request to ZD to set its IEEE address to ZD's IAS\_CIE\_Address.
- 6 If ZD permits remote write attribute, it sends a write attribute response to CIE with success status code.
- 7 ZD sends an enrol request to CIE
- 8 CIE responds to ZD with an enrol response.

## **Enrol More Zone Devices**

From build B030714, the CICIE firmware supports 32 zones entries. Zone entries 00 - 0F are flash maintained and entries 10 – 1F are RAM maintained. The user can also set S61 Bit 5 to allow more zone devices to enrol even when CIE's zone table is full. The following describes an example procedure for using extended zone table (up to 255 entries).

Several assumptions need to be made

1. CIE's zone table has been full.
2. The user has set S61 to allow a zone device with any zone types to enrol
3. The joining zone device has never enrolled with the CIE
4. The total number of enrolled devices has not reached 255.

Commissioning procedure:

1. Set S61 Bit 5 (using command ATS615 = 1).
2. When the joining zone device joins CIE's network, it sends an enrol request. When CIE receives this enrol request, it shows a prompt "ZENROLLREQ".
3. As S61 Bit 5 has been set, the CIE will send an enrol response with success status code and a Zone ID that is larger than 0x1F. Meanwhile the CIE will show a prompt "ENROLLED". The



user should consider copying the information provided with “ENROLLED” prompt and saving it as a zone entry.

4. The user should use AT+EDITZMAP command to edit zone map information. The zone map information will be used to respond to ACE device’s get zone map request.

A zone ID map has 16 sections. Each section is a 16 bit bitmap. If CIE’s local zone table is full, section 0 and section 1 will be both 0xFFFF. (Please note: section 0 and section 1 are generated by CIE automatically according to the usage of local zone entries. The user cannot edit section 0 and 1).

An example:

If CIE shows prompt ENROLLED:20,1234, 0021ED1000000003. It means that the CIE has sent an enrol response to a joining zone device with success status code and zone ID 0x20. Then the user should change corresponding bit for this zone, indicating that the zone device has been enrolled. To do this, the user can use AT+EDITZMAP:02,0001, as bit 0 of section 2 is associated with zone ID 0x20.

5. By default, CIE will handle get zone information command sent by ACE device. If CIE receives a get zone information command with zone ID less than 32, it will check local zone table to retrieve the information and respond.

If the extended zone table is used (zone ID larger than 31), the user should set S61 bit 6 to enable user handling of the zone information request by using AT+ZINFORSP command. In such a case, when CIE receives a get zone information command with zone ID that is larger than 31, the “GETZONEINFO” prompt will be presented as a notification. Then the user should respond to it with the same sequence number contained in get zone information request command.

## 10 Appendix C

This Appendix provides some application notes of using time server cluster of CICIE as a time master on a HA network.

The CICIE supports Time Server cluster (0x000A). The *Time* attribute (0x0000) of time server cluster holds the time value of a real time clock. When the module is reset or during power cycle, this attribute will be reset and start counting from 0x00000000. Therefore, the user should set the time (using AT+SETTIME) after reset or power cycle the module. After setting valid value to Time and Time status attributes, the user need to use AT+TIMERD command to notify the module that the clock time is now ready to be used.

**Note: if the user does not use AT+TIMERD command, CICIE will respond to a read time cluster attribute command with a default response (status code failuer).**

In addition, some HA device need to synchronize its time with network COO’s local time, so the CICE supports *StandardTime* (0x0006) and *LocalTime* (0x0007).

The local Standard Time, i.e. the time adjusted for the time zone, but not adjusted for Daylight Saving Time (DST) is given by

$$\text{Standard Time} = \text{Time} + \text{TimeZone}$$

The Local Time, i.e. the time adjusted for both the time zone and DST, is given by

$$\text{Local Time} = \text{Standard Time} + \text{DstShift (if DstStart} \leq \text{Time} \leq \text{DstEnd)}$$

$$\text{Local Time} = \text{Standard Time (if Time} < \text{DstStart or Time} > \text{DstEnd)}$$

The CICIE firmware has integrated the mechanism to calculate standard time and local time. The calculated values are stored in attributes *StandardTime* (0x0006) and *LocalTime* (0x0007) respectively.

In order to serve Local Time to other devices on network, the CICIE needs to have valid *TimeZone* (0x0002), *DstStart* (0x0003), *DstEnd* (0x0004), and *DstShift* (0x0005). Therefore, it is recommended that the user shall utilize AT+SETATR command to set these four attributes after initiating the CICIE. By default the four attribute will be set to 0, so both standard time and local time will be the same as clock time.

It is strongly recommended that the user shall maintain the *TimeStatus* (0x0002) attribute which holds a number of bit fields to indicate the status of local clock time.

Bit	Meaning	Values	Notes
0	Master	1 – master clock 0 – not master clock	This bit is not writeable. By default this bit is set to 1
1	Synchronized	1 – synchronized 0 – not synchronized	By default this bit is set to 0
2	MasterZoneDst	1 – master for Time Zone and DST 0 – not master for Time Zone and DST	This bit is not writeable. By default this bit is set to 0
3	Superseding	1 – time synchronization should be superseded 0 - time synchronization should not be superseded	By default this bit is set to 0
Others	Reserved		

The Master and Synchronized bits together provide information on how closely the Time attribute conforms to the time standard.

The Master bit specifies whether the real time clock corresponding to the Time attribute is internally set to the time standard.

The Synchronized bit specifies whether Time has been set over the ZigBee network to synchronize it (as close as may be practical) to the time standard.

The MasterZoneDst bit specifies whether the *TimeZone*, *DstStart*, *DstEnd* and *DstShift* attributes are set internally to correct values for the location of the clock.

The user can follow the following the steps in sequence to set up a time server.

1. After the module power up, set local clock time. e.g. AT+SETTIME:1BDBE518 (1BDBE518 is utc time representing 16:11:00 23/10/2014 ).
2. Set Bit 1 of TimeStatus since local clock time has been synchronized with current time. Please use AT+SETATR:000A,0001,03 (Note: please set bit 0 to 1 as the time is used as master time from local clock)

3. Set *TimeZone*, *DstStart*, *DstEnd* and *DstShift* attributes accordingly.
4. Set Bit 2 of *TimeStatus* attribute. Please use `AT+SETATR:000A,0001,07` (Note: please set bit 0 to 1)

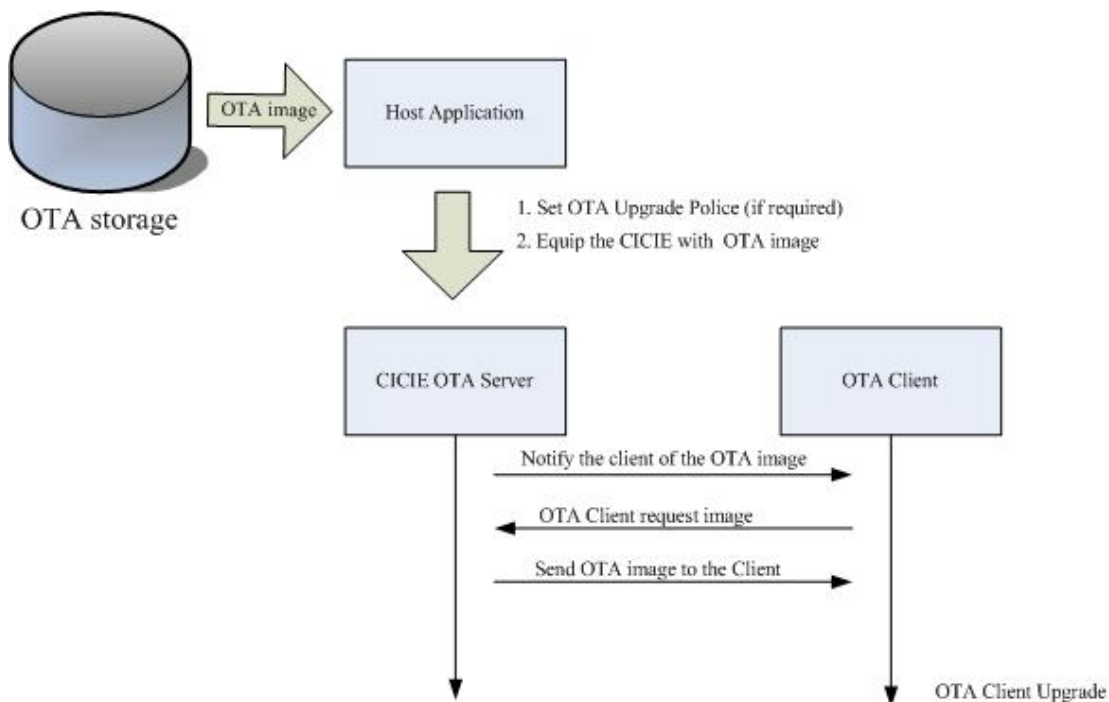
Then the CICIE can produce local time for remote device to use during time synchronization.

## 11 Appendix D

The OTA cluster provides a standard way to upgrade devices in the network via OTA messages. Thus the upgrade process may be performed between two devices from different manufacturers. Devices are required to have application bootloader and additional memory space in order to successfully implement the cluster. This Appendix will brief OTA upgrade progress.

According to OTA cluster specification, it is the responsibility of the server to indicate to the clients when the update images are available. The client may be upgraded, downgraded or reinstalled. The upgrade server must know which client devices to upgrade and to what file version. The upgrade server may be notified of such information via the backend system. In this project, the Host application will be responsible of storing and manage the information of client devices for upgrading.

The OTA cluster is implemented in a way that the client service works on both ZED and ZR devices. Being able to handle polling is mandatory for all server devices. Therefore, the CICIE will provide APIs to the Host application to handle the poll. Meanwhile an notify command is supported for sending broadcast or unicast new image notification, while being able to send a notification is optional. The following picture illustrates the way how OTA upgrade cluster works. Firstly, the host need to prepare an OTA image and set the upgrade police for the OTA server. Then the CICIE OTA server can either send a notification to the OTA client or wait for the client to query the image. After the OTA client is informed of the availability of new image, it will start downloading the image till complete.



The OTA upgrade progress may include multiple command or message exchange between OTA server and client. The following table listed these commands

**Table D.1 OTA Upgrade cluster commands**

<b>Command ID</b>	<b>Description</b>	<b>Direction</b>	<b>Mandatory/Optional</b>
0x00	<i>Image Notify</i>	Server	O
0x01	<i>Query Next Image Request</i>	Client	M
0x02	<i>Query Next Image Response</i>	Server	M
0x03	<i>Image Block Request</i>	Client	M
0x04	<i>Image Page Request</i>	Client	O
0x05	<i>Image Block Response</i>	Server	M
0x06	<i>Upgrade End Request</i>	Client	M
0x07	<i>Upgrade End Response</i>	Server	M
0x08	<i>Query Specific File Request</i>	Client	O (not supported by CICIE)
0x09	<i>Query Specific File Response</i>	Server	O (not supported by CICIE)

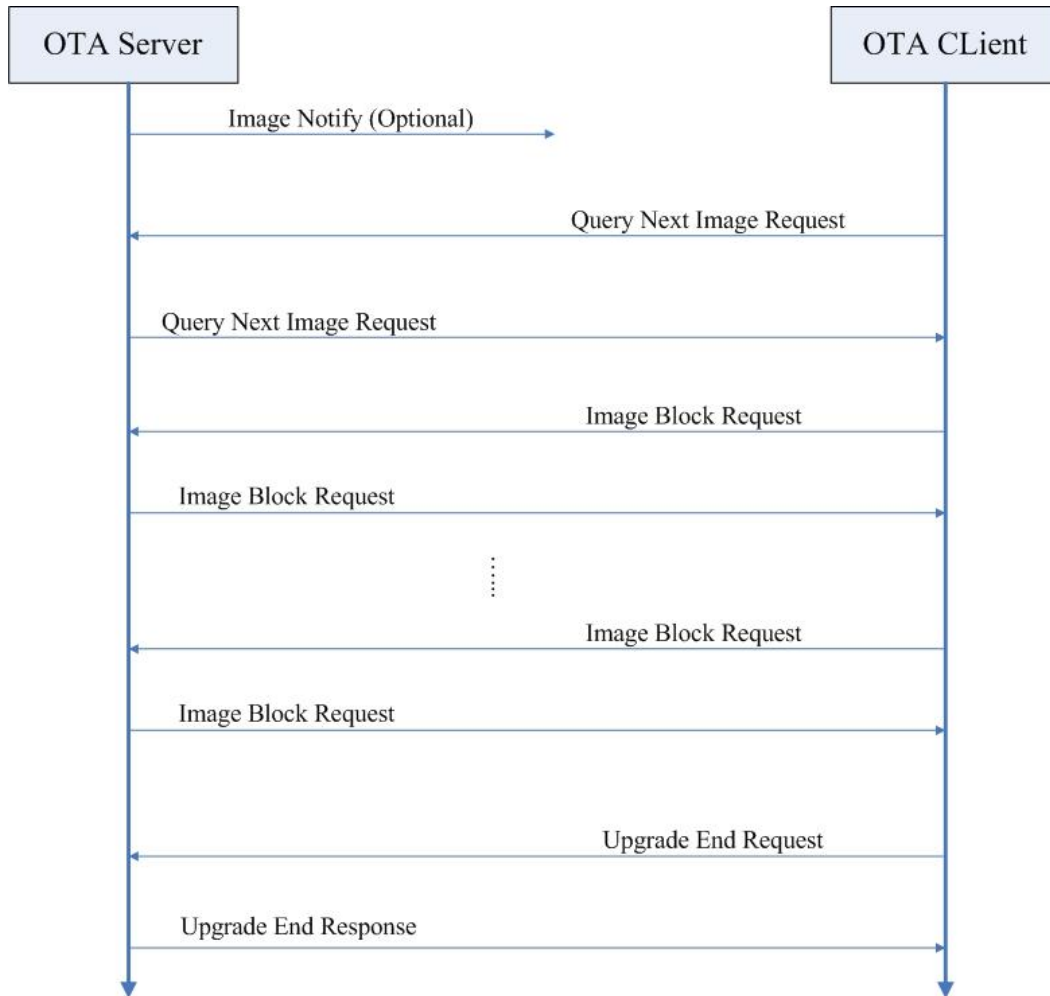


Figure D.1 Typical OTA upgrade process

The above figure illustrate a common OTA upgrade process using the commands listed in Table D.1. An OTA upgrade progress may be divided in three main parts.

1. The OTA client gets notification of availability of a new image.

According to the OTA cluster specification, the server may notify devices in the network when it receives new OTA upgrade image by sending an Image Notify Command.

The client device will send Query Next Image Request Command if the information in the Image Notify Command is of interest and after applying the jitter value.

It is worth mentioning that sending Image Notify Command to OTA client is optional. So all OTA client devices shall send in a Query Next Image Request Command periodically regardless of whether an Image Notify was sent by the OTA server.

Once the OTA server receives a Query Next Image Request, it shall check the information contained in the request and make decision for responding to the requesting device.

2. The OTA client download image blocks from server

When the device has received a response to its query indicating a new OTA upgrade image is available, the client device shall request blocks of the OTA upgrade image. The process continues until the client receives all image data.

3. The OTA client query the server for upgrade time

When the client has finished image downloading, it shall verify the integrity of the whole image received and send Upgrade End Request Command along with the upgrade status. The server shall notify the client of when to upgrade to new image in the Upgrade End Response.

## D.1 CICIE OTA Server Implementation

From the description above, it can be found that an OTA server will normally receive three commands and issue three response commands. In addition, the server should be able to send an image notify command. Therefore three prompts and four AT commands are defined in OTA server implementation.

The host can use AT+IMGNOTIFY command to send unicast/broadcast to OTA client to notify them of the availability of a new image. Then it will receive query commands.

1. IMGQUERY – This prompt is shown, when Query Next Image Request command is received

The host can use AT+QIMGRSP command to send back response.

2. IMGBREQ – This prompt is shown, when Image Block Request command is received.

The host can use AT+IMGBRSP command to send back response.

3. UPGRADEREQ - This prompt is shown, when Image Block Request command is received.

The host can use AT+UPGRADE command to send back response.

Note: In the later build version of CICIE (from B280515), a new prompt IMGPREQ is introduced to support image page request. When this prompt shows, the host should prepare the request OTA file page and then use a series of AT+IMGBRSP command to send the page block by block. **Moreover, when the host issues AT+IMGBRSP commands in response to a page request, the host shall set 5 in <sendMode> parameter to disable APS retries as per OTA cluster specification.**

## D.2 Use Cases

This section will introduce some use cases with purpose of assisting the user to test and understand the way of use OTA server features.

In CICIE firmware design, an S-register S62 is introduced for configuring OTA commands handling. By changing the setting in S62, the user can select to handle all incoming OTA client commands or partially handle them. Please note: block request command will always be passed to host application since the host takes in charge of storing and serving OTA images.

### D.2.1 Host Application Handle Incoming OTA Client Cluster Commands

In this scenario, S62 should be set to 0000. The CICIE will not check and handle the OTA client request commands, all incoming requests will be passed to the host application. In such a case, the host application need to wait for the IMGQUERY prompt and based on the parameters present following the prompt to find the relevant image in its file system. Then the Host shall use AT+QIMGRSP to respond to the requestor and indicate the availability of the requested file.

If the host can serve the requested OTA file, the next step will be that the remote device start requesting OTA block data. Upon receiving a block request, the CICIE will use an IMGBREQ prompt to notify the host. Then the host shall use AT+IMGBRSP to transmit the block to the requestor. The requestor will send another block request till it receives the whole OTA file.

If the host gets an IMGPREQ prompt, it shall prepare the page of the OTA file (contains multiple blocks) and apply a series of AT+IMGBRSP to transmit all blocks of the requested page to the requestor.

If the host loses the OTA file and cannot serve it anymore after an OTA progress has started. The user may use AT+DFTRSP with status 0x98(IMAGE\_NOT\_AVAILABLE) to terminate the OTA upgrade progress. This is a specified behaviour in OTA cluster specification.

Once an OTA file is received by the requestor, it will send an upgrade end request command to query the update time. The CICIE firmware will show UPGRADEREQ prompt to the host then the host shall respond with AT+UPGRADE to finish off the OTA upgrade process.

The above produce requires the host application to parse the OTA image file to retrieve the manufacturer id, hw/fw firmware versions, etc to supply correct image data to the CICIE firmware.

### D.2.2 CICIE OTA Plugin Handle Incoming OTA Client Cluster Commands

As it has been mentioned that all OTA client device shall periodically send Query Next Image Request command to server, the host may require some filtering mechanism to assist it to handle some incoming request. Also the host may want to set the CICIE to respond to its clients about the upgrade time. In such a case, S62 should be set to 0003 (both bit0 and bit1 are set to true).

If the host has got a new image for downloading, AT+SETPOLICY, AT+SETIMGID and AT+SETHWVER command can be used to set OTA upgrade policy and image ID, hardware(hw) version, software(sw) version and so on. If the new image is for a specific node, the host can also use AT+SETIMGID to set that node's network address. These information will be used to filter out all request which contained unmatched information. Only the request with all matched information will be shown to the host application with the IMGQUERY prompt.

In the case that a request contained unmatched information (e.g. manufacture Id, image type or fw version), the CICIE will automatically send a QueryNextImageResponse with status code No\_Image\_Available to the requesting client.

If a request is received with all matched information, the CICIE will automatically send a QueryNextImageResponse with status code Success to the requesting client. In the meantime, the host will receive the IMGQUERY prompt, which is to notify it for preparing the image downloading.

Then the remote device may start requesting the block for data from the OTA server at that point the host application will receive IMGBREQ prompt with block address for the host application to supply



that data block from the OTA image to the ZigBee for sending to the remote device using AT+IMGBRSP.

Once all blocks have been transmitted to the remote device the remote device sends an upgrade end request which is provided to the host application as UPGRADEREQ prompt. In this scenario, the UPGRADEREQ prompt is used to notify the host that the client has finished image downloading. Since the S62 Bit 1 is set to true, the CICIE will automatically sends an Upgrade End Response to the client. If the host did not change the upgrade policy, the CICIE will send the response to inform the client to upgrade the downloaded firmware immediately. The host can change the policy using AT+SETPOLICY command. This will be introduced in Section 5.

### **D.2.3 Both Host and CICIE OTA Plugin Partially Handle Incoming OTA Cluster Commands**

If the S62 is set to 0001 or 0002, CICIE will only handle Query Next Image Request or Upgrade End Request respectively. In such a case, the host need to use corresponding AT command to construct and send response when IMGQUERY or UPGRADEREQ is shown.

For instance, the host only want the CICIE to handle incoming Query Next Image Request command. It should set S62 to 0001. Then the CICIE will only pass the request which contained all matched information (manufacture ID, device type and so on) and print the IMGQUERY prompt. It will also send back response to the requesting clients.

Then the remote device may start requesting the block for data from the OTA server at that point the host application will receive IMGBREQ prompt with block address for the host application to supply that data block from the OTA image to the ZigBee for sending to the remote device using AT+IMGBRSP.

Once all blocks have been transmitted to the remote device the remote device sends a upgrade end request which is provided to the host application as UPGRADEREQ prompt and the host then responds to the message with AT+UPGRADE to finish off the process.

## **D.3 Explanation of Upgrade Policy**

The upgrade policy will only be used when S62 is set to none 0 value. The Upgrade Policy includes Query Policy and UpgradeRequestPolicy. The AT+SETPOLICY command can be used to set these two policy. And AT+GETPOLICY can be used to check the pre-set policy.

### **D.3.1 QueryPolicy (It is used when S62 Bit 0 is set to true)**

- 00: Upgrade if server has newer
- 01: Downgrade if server has older
- 02: Reinstall if server has same
- 03: No next version

Assume:

1. Manufacture code, ImageType, hw version etc parameters contained in the Query Next Image request match the OTA server' corresponding information.
2. The OTA server gets an image with Firmware Version: 0x00000002
3. The Query Next Image request contained Current Firmware Version: 0x00000001



If it has been set to 00, because the device got its current firmware version 0x00000001, the CICIE has a newer one (0x00000002). Then the CICIE will print a prompt to notify the host to prepare block for downloading. The prompt is "IMGQUERY". Meanwhile, it sends a response to the client notify that there is new image for downloading.

If it has been set to 01, because the device got its current firmware version 0x00000001, the CICIE does not has an older one (may be 0x00000000). Then the CICIE will not print a prompt. Meanwhile, it sends a response to the client notify that there is no available image for downloading.

If it has been set to 02, because the device got its current firmware version 0x00000001, the CICIE does not has a same version (should be 0x00000001). Then the CICIE will not print a prompt. Meanwhile, it sends a response to the client notify that there is no available image for downloading.

If it has been set to 03, the CICIE sends a response to the client notify that there is no available image for downloading.

### **D.3.2 UpgradeRequestPolicy (It is used when S62 Bit 1 is set to true)**

00: Upgrade Now

01: Upgrade In a few minutes (2 minutes)

02: Ask me later to upgrade

03: Abort upgrade

Please note this policy will be used only if the ATS63 Bit1 is set to true. In that case, the host does not need to issue an AT+UPGRADE command as the CICIE will automatically respond to the upgradeEndRequest command.

After the OTA client finished downloading OTA image, it will send an upgradeEndRequest command to the CICIE.

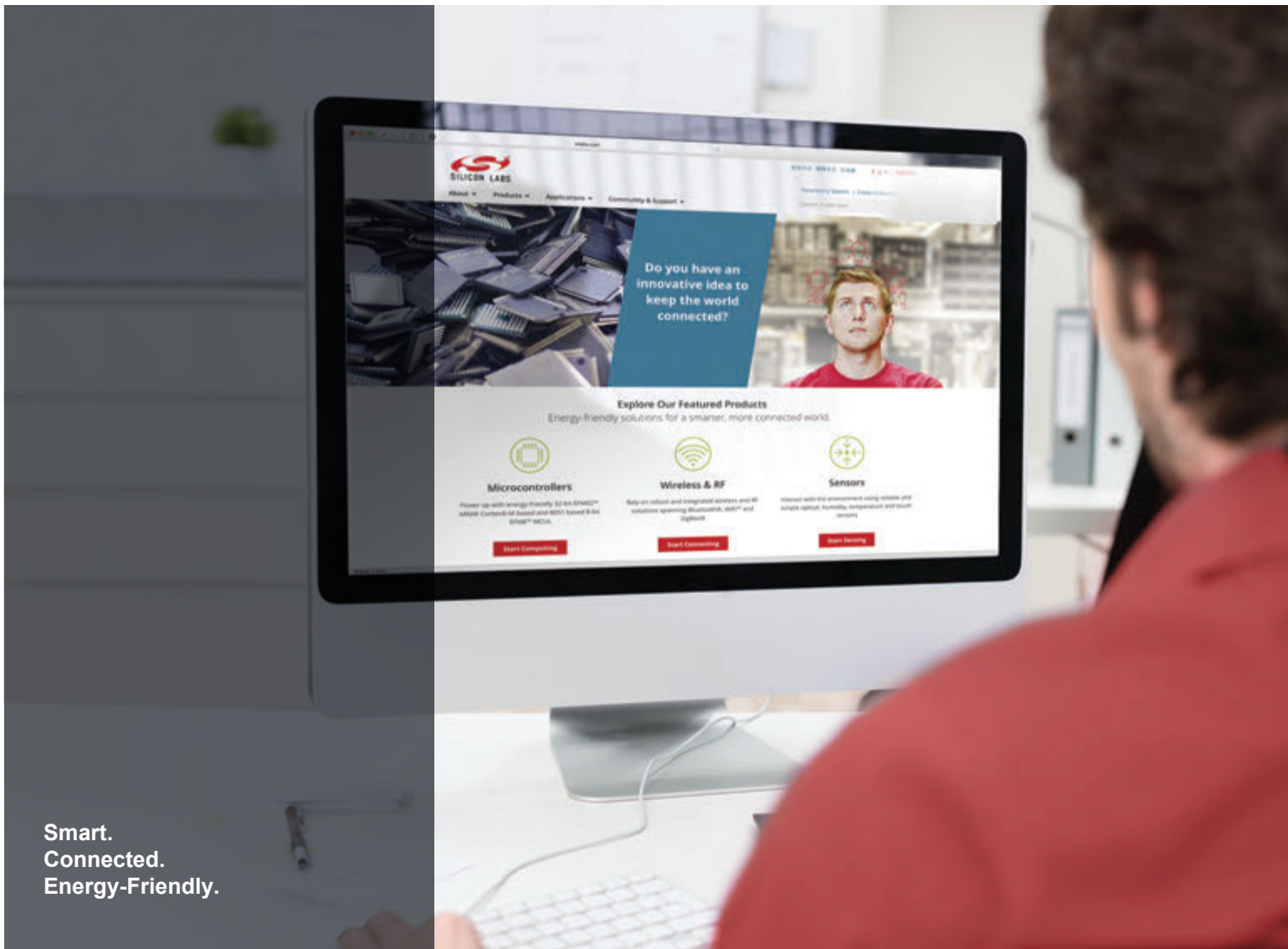
If we have set UpgradeRequestPolicy to 00, the CICIE will send back an upgradeEndResponse with current time (0x00000000) and upgrade time (0x00000000). In this case, the OTA client is expect to upgrade itself immediately.

If we have set UpgradeRequestPolicy to 01, the CICIE will send back an upgradeEndResponse with current time (0x00000000) and upgrade time (0x00000078). In this case, the OTA client is expect to upgrade itself after 2 minutes.

If we have set UpgradeRequestPolicy to 02, the CICIE will send back an upgradeEndResponse with current time (0x00000000) and upgrade time (0xFFFFFFFF). In this case, the OTA client is expect not to upgrade itself until receiving an unsolicited upgradeEndResponse command with an update time which is not all F.

If we have set UpgradeRequestPolicy to 03, the CICIE will send back a default response with error code 0x95 indicating Abort status.

Note: This appendix only provide information to assist the user for evaluating HA OTA produce. More information about OTA upgrade cluster can be found in docs-09-5264-23-00zi-zigbee-ota-upgrade-cluster-specification.pdf.



Smart.  
Connected.  
Energy-Friendly.



**Products**  
[www.silabs.com/products](http://www.silabs.com/products)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

#### Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are not designed or authorized for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

#### Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



**Silicon Laboratories Inc.**  
400 West Cesar Chavez  
Austin, TX 78701  
USA

<http://www.silabs.com>