ETRX3 Series ZigBee Module
AT-Command Dictionary

Firmware R309
EmberZNet 5.4.0 stack
# Table of Contents

## 1 INTRODUCTION

1.1 The first step ................................................................. 4
1.2 Module overview .......................................................... 4
1.3 Document Overview ...................................................... 4
1.4 Network topology .......................................................... 5
1.5 The ADCs ................................................................. 5
1.6 RTC Related Commands .................................................. 5
1.7 A Note on ZigBee® Compliance .......................................... 5
1.8 Important notes ............................................................ 6
1.8.1 Hardware compatibility ................................... 6
1.8.2 Unexpected start-up in bootloader mode .................................... 6
1.8.3 Compatibility with other devices ........................................ 6
1.8.4 Persistence of network parameters .................................... 6

## 2 AT STYLE COMMAND CONVENTIONS

2.1 Parameters ............................................................... 8
2.2 Prompt Overview .......................................................... 9
2.3 Device Overview .......................................................... 11
2.3.1 ZigBee types .......................................................... 11
2.3.2 Non-ZigBee types ..................................................... 11
2.4 Addressing modes ......................................................... 11
2.5 AT Command Overview ................................................ 13
2.6 Module Control & Configuration Commands ......................... 15
2.7 Network Control & Configuration Commands ......................... 21
2.8 Messaging ................................................................. 40
2.9 Binding Management (ETRX3 Series only) .............................. 58
2.10 Time-related commands ................................................ 63

## 3 LIST OF ERROR CODES

................................................................. 65

## 4 S-REGISTERS

4.1 Recovery of the Factory Default Settings ............................ 69
4.2 S-Registers for Network Setup ........................................ 70
4.3 S-Registers for Module Setup .......................................... 75
4.4 I/O related S-Registers ................................................... 81
4.5 S-Registers Defining the Functionality of the Module .............. 90
4.6 Advanced Settings ....................................................... 104

## 5 BUILD IN FUNCTIONALITY

................................................................. 112

## 6 ETRX357 POWER CONSUMPTION (PROVISIONAL DATA)

................................................................. 115

## 7 NOTES ON ENERGY LEVELS AND LQI

7.1 Interpreting LQI ........................................................... 116
7.2 Interpreting RSSI Energy Levels ...................................... 117

## 8 TRADEMARKS

................................................................. 118

## 9 DISCLAIMER

................................................................. 118

## 10 CONTACT INFORMATION

................................................................. 118
11 REFERENCES........................................................................................................................................118
12 APPENDIX A. FORMING A SECURE NETWORK .............................................................................119
13 APPENDIX B. BOOTLOADING NEW Firmware.................................................................................120
  13.1 Bootloading through the serial port .............................................................................................120
  13.2 Bootloading over the air ..............................................................................................................121
1 Introduction

1.1 The first step

Send the command “ATI” to a module to find its firmware version. If it is not R309 then you should refer to the correct version of the AT command manual, or send an e-mail to zigbeesupport@telegesis.com requesting a copy of the R309 firmware file. All four variants of the ETRX357 module share the same file.

Alternatively if your module has R309 and you prefer a different version, this can be provided on request.

All the standard AT command set firmware files are free of charge to users who already have the ETRXn devices, but they must only be used on Telegesis modules.

1.2 Module overview

This document describes the AT-Command interface firmware of the ETRX3 series ZigBee PRO wireless meshing modules. It applies to the R309 firmware, which can be loaded on to all products of the ETRX3 module series, for example:

- ETRX357, ETRX357-LR, ETRX357-LRS
- ETRX357HR, ETRX357HR-LR, ETRX357HR-LRS
- ETRX3585, ETRX3587 and ETRX3588
- ETRX3USB
- ZigBee Communications Gateway

The Telegesis ZigBee modules have been designed to be built into any device and provide a low cost, low power ZigBee solution based on the industry leading EmberZNet ZigBee stack. Integration into a wide range of applications is made easy using a simple AT-style software interface and advanced hardware design.

No RF experience or expertise is required to add 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.

Important note

Using the AT-Command interface described in this document can shorten the time to market significantly, however customers using the range of Telegesis modules also have the option of using Ember’s EZSP interface firmware or of developing custom firmware using the Ember Development tools.

1.3 Document Overview

This document is meant as an AT-Command and S-Register reference for R3xx revisions of the firmware based on EmberZNet3.x and EmberZNet4.x. In order to learn how your products can benefit from wireless mesh networking please also refer to the following documents:

ETRX3 Product Manuals
1.4 Network topology

A network consists of a ZigBee Coordinator (ZC) which started the network, ZigBee Routers (ZR) and ZigBee End Devices (ZED). There do not have to be any routers (other than the coordinator, which functions as a router) or end devices in any given network. Each router can support up to 30 end devices in any combination of non-sleepy, sleepy and mobile End Devices. The network is always formed as a mesh according to the ZigBee PRO feature set of the ZigBee standard; the tree structure is not available.

By default the module joins a PAN as a router, but modifying register S0A allows you to define it as an end device. The coordinator is simply the device that first establishes the PAN, and it should not be allowed to leave the PAN as it is not possible for a node that is already joined to the PAN to take over the role of a coordinator or Trust Centre.

1.5 The ADCs

The ETRX357 can operate up to 4 ADCs, which are individually enabled by setting the appropriate bits in register S15. A reading is taken each time one of the registers S1F-22 is read, or when a built-in function is executed which reads an ADC. If bit 8 of S15 is set the 1.2V Vref level is presented at pin PB0 for the brief interval while the reading is taken.

<table>
<thead>
<tr>
<th>Mode</th>
<th>single-ended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0-1200mV</td>
</tr>
<tr>
<td>Resolution</td>
<td>14 bits</td>
</tr>
<tr>
<td>Units</td>
<td>1 LSB = 0.1mV</td>
</tr>
<tr>
<td>Max load on PB0</td>
<td>1mA</td>
</tr>
</tbody>
</table>

1.6 RTC Related Commands

The module runs a real time clock which can be set, read and synchronized against a time server with the commands shown in this chapter. Please note that the basis of the real time clock is an on-chip RC timer which gets calibrated against the external quartz crystal. Overall the accuracy is not high and will vary with temperature, so if an accurate RTC is to be maintained frequent re-synchronization with a time server is required.

1.7 A Note on ZigBee® Compliance

The Telegesis R300 firmware has been tested and certified for MSP (manufacturer specific profile) compliance by a test house appointed by the ZigBee Alliance.
This certification includes tests guaranteeing that:

- Modules running the Telegesis AT-Command set will not interfere with existing ZigBee Networks in a malicious way
- Modules running the Telegesis AT-Command set can join a 3rd party ZigBee PRO network and use its routing capabilities
- Modules running the Telegesis AT-Command set can allow 3rd party nodes to join into a network consisting of Telegesis nodes and use its routing capabilities

In addition to implementing a manufacturer specific application profile the AT-Command set allows for transparency allowing communication with 3rd party nodes running any public application profile. In addition to this a transparent endpoint has been added allowing a host processor to implement any public application profile in fully transparent mode.

If you want to use the term ZigBee or the ZigBee Logo in your product documentation the current regulations state that you have to

(i) Be at least an adopting member of the ZigBee Alliance in the year you release your product
(ii) Implement a public application profile

If you intend to get your product certified feel free to contact Telegesis for additional information. Also if you intend to build a product compliant to a public application profile (e.g. Home Automation, Smart Energy) feel free to contact us to discuss your options.

1.8 Important notes

1.8.1 Hardware compatibility

R2xx firmware will not run on the ETRX3 series of modules.

1.8.2 Unexpected start-up in bootloader mode

The bootloader in the ETRX357 can be triggered using the command AT+BLOAD as described in section 2, but it can also be triggered in hardware. If the PA5 pin is pulled low during the boot-up of the module, the module will also enter the bootloader, so exercise caution when doing hardware design and ensure that this pin is not grounded during start-up and reset. If unused the pad can be left floating and a pull-up is not required.

1.8.3 Compatibility with other devices

Most features of the R3xx Telegesis AT-Command line Interpreter are part of a Manufacturer Specific Profile using the ZigBee PRO feature set of ZigBee 2007. Interoperability with other devices that use the ZigBee PRO featureset is limited to a number of transparent commands.

R3xx is not compatible with earlier versions of ZigBee which do not implement the ZigBee PRO featureset, including Telegesis R2xx firmware. Also, it is not compatible with the ZigBee Smart Energy profile as it lacks the required security key.

1.8.4 Persistence of network parameters

Once a device has joined a network as a coordinator, router or end device, it will retain its network parameters if it is powered off and on again. It will still be a member of its original PAN, assuming that PAN still exists, though an end device may need to find a new parent and it may have missed an update of the network key. Certain S-registers will have been reset to default values, though, which may change an end device’s power mode for example.
2 AT Style Command Conventions

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

Each command must be preceded by the "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. Every command must be terminated with a <CR>, they cannot be concatenated.

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>. Unless disabled in S0E or S0F prompts may appear whenever the corresponding event occurs.

Example:

<CR><LF><BCAST:000D6F000005A666,04=test><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.

The ETRX357 features a 128-byte FIFO to buffer incoming characters from the host processor, which is sufficient to hold even the longest possible command. The ETRX357 features a 256-byte FIFO buffer for incoming radio messages, which allows rapid reception of multiple messages without loss of characters. To prevent a buffer overflow XON/XOFF handshaking is used. Optional hardware handshaking can be enabled as described in the register description of S12 in section 4.

<table>
<thead>
<tr>
<th>Read Command</th>
<th>Commands ending with a ‘?’ return the currently set value of the parameter or parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATXXX?</td>
<td></td>
</tr>
<tr>
<td>Write Command</td>
<td>This command sets user-definable parameters as indicated by the ‘=’ sign.</td>
</tr>
<tr>
<td>ATXXX=⟨…⟩</td>
<td></td>
</tr>
<tr>
<td>Execute Command</td>
<td>This command executes routines of the module and returns parameters</td>
</tr>
<tr>
<td>ATXXX</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Types of AT commands

When bit 7 of S12 is set each individual reply or prompt is additionally started with the STX and ended with the ETX character to aid the interpretation of the incoming strings on a host processor.
2.1 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 [...].

| XX     | 8-bit hexadecimal number. Valid characters are 0-9, a-f and A-F |
| XXXX   | 16-bit hexadecimal number. Valid characters are 0-9, a-f and A-F |
| n      | Number from 0-9 |
| s      | Sign |
| b      | Bit (0 or 1) |
| c      | character |
| <PID>  | 16-bit hexadecimal PAN ID (0000 to FFFF) |
| <EPID> | 64-bit hexadecimal extended PAN ID |
| <channel> | decimal channel (802.15.4 channel 11-26) |
| <password> | 8 character password |
| <EUI64> | 64-bit IEEE 802.15.4 address in hexadecimal |
| <iorem> | 32-bit hexadecimal number representing the reading of S1A |
| <data> | Custom Data |
| <ClusterList> | A list of 16 bit cluster identifiers in hexadecimal representation |
| <FirmwareRevision> | The Firmware Revision Number |

Table 2: Different formats of parameters
## 2.2 Prompt Overview

The following prompts can show up during the operation of the ETRX357 modules. Most of the prompts can be disabled using register S0E and S0F.

<table>
<thead>
<tr>
<th>Prompt Overview</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>OK terminator</td>
</tr>
<tr>
<td>ERROR:XX</td>
<td>Error number XX occurred</td>
</tr>
<tr>
<td>ACK:XX</td>
<td>Acknowledgement for message XX was received</td>
</tr>
<tr>
<td>NACK:XX</td>
<td>Acknowledgement for message XX was not received</td>
</tr>
<tr>
<td>POLLED:XX</td>
<td>Shown on an end device when it polls its parent and S11 bit D is set. XX codes are:</td>
</tr>
<tr>
<td></td>
<td>00 - Success</td>
</tr>
<tr>
<td></td>
<td>31 - no data pending on host</td>
</tr>
<tr>
<td></td>
<td>66 - poll could not be executed</td>
</tr>
<tr>
<td></td>
<td>40 - no ack from parent</td>
</tr>
<tr>
<td>SR:XX,&lt;EUI64&gt;,&lt;NWK addr&gt;,...</td>
<td>Route Record Message received</td>
</tr>
<tr>
<td>BCAST:[:&lt;EUI64&gt;,]XX=&lt;data&gt; [,&lt;RSSI&gt;,&lt;LQI&gt;]</td>
<td>A Broadcast with XX characters has been received</td>
</tr>
<tr>
<td>MCAST:[:&lt;EUI64&gt;,]XX=&lt;data&gt; [,&lt;RSSI&gt;,&lt;LQI&gt;]</td>
<td>A Multicast with XX characters has been received</td>
</tr>
<tr>
<td>UCAST:[:&lt;EUI64&gt;,]XX=&lt;data&gt; [,&lt;RSSI&gt;,&lt;LQI&gt;]</td>
<td>A Unicast with XX characters has been received</td>
</tr>
<tr>
<td>INTERPAN:&lt;ProfileID&gt;,&lt;ClusterID&gt;,&lt;Msgtype&gt;,&lt;Option&gt;,[GroupID],&lt;PanID&gt;,&lt;SrcAddr&gt;,&lt;MsgLength&gt;,&lt;Msg&gt;</td>
<td>The device has received an interpan message</td>
</tr>
<tr>
<td></td>
<td>&lt;ProfileID&gt; - 16 bit hex</td>
</tr>
<tr>
<td></td>
<td>&lt;ClusterID&gt; - 16 bit hex</td>
</tr>
<tr>
<td></td>
<td>&lt;Msgtype&gt; - 8 bit hex</td>
</tr>
<tr>
<td></td>
<td>0x00 – Unicast</td>
</tr>
<tr>
<td></td>
<td>0x08 – Broadcast</td>
</tr>
<tr>
<td></td>
<td>0x0C - Multicast</td>
</tr>
<tr>
<td></td>
<td>&lt;Option&gt; - 16 bit hex. If it is 0x0002 the &lt;SrcAddr&gt; will be source long address, and otherwise it is source network address</td>
</tr>
<tr>
<td></td>
<td>[GroupID] – 16 bit hex, shown if the message is sent to a group</td>
</tr>
<tr>
<td></td>
<td>&lt;PanID&gt; - 16 bit hex, source PAN ID</td>
</tr>
<tr>
<td></td>
<td>&lt;SrcAddr&gt; - 16 bit hex source node ID or EUI</td>
</tr>
<tr>
<td></td>
<td>&lt;MsgLength&gt; - 8 bit hex, message length</td>
</tr>
<tr>
<td></td>
<td>&lt;Msg&gt; - received message in hex format</td>
</tr>
<tr>
<td>RAW:snn,&lt;data&gt;</td>
<td>A raw message has been received with strength snn dBm</td>
</tr>
<tr>
<td>SDATA:[&lt;EUI64&gt;,],&lt;ioread&gt;,&lt;ADC0&gt;,&lt;ADC1&gt;,&lt;sequenceNo&gt;,&lt;Vcc&gt;</td>
<td>A data message has been received at the sink. ADC data is ADC0 &amp; ADC1</td>
</tr>
<tr>
<td>Prompt</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>FN130: [&lt;EUI64&gt;, &lt;NWK addr&gt;, &lt;ioread&gt;, &lt;sequence no&gt;, &lt;S46&gt;, [&lt;ADC0&gt;], [&lt;ADC1&gt;], [&lt;ADC3&gt;]]</td>
<td>A data message has been received at the sink. The number of ADC data fields depends on how many ADCs are activated at the sender</td>
</tr>
<tr>
<td>FFD: [&lt;EUI64&gt;, &lt;NWK addr&gt;]</td>
<td>A router announcing itself</td>
</tr>
<tr>
<td>SED: [&lt;EUI64&gt;, &lt;NWK addr&gt;]</td>
<td>A sleepy end device announcing itself</td>
</tr>
<tr>
<td>MED: [&lt;EUI64&gt;, &lt;NWK addr&gt;]</td>
<td>A mobile sleepy end device announcing itself</td>
</tr>
<tr>
<td>ZED: [&lt;EUI64&gt;, &lt;NWK addr&gt;]</td>
<td>An end device announcing itself</td>
</tr>
<tr>
<td>NEWNODE: [&lt;NWK addr&gt;, &lt;EUI64&gt;, &lt;Parent NWK addr&gt;]</td>
<td>Shown on Coordinator: New node has been given permission to join the PAN. NB joining is not complete until an “FFD” prompt or similar is received</td>
</tr>
<tr>
<td>LeftPAN</td>
<td>Local Node has left the PAN</td>
</tr>
<tr>
<td>LostPAN</td>
<td>End Device has lost contact with Parent</td>
</tr>
<tr>
<td>JPAN: [&lt;channel&gt;, &lt;PID&gt;, &lt;EPID&gt;]</td>
<td>Local Node has joined PAN with given parameters</td>
</tr>
<tr>
<td>NODELEFT: [&lt;NWK addr&gt;, &lt;EUI64&gt;]</td>
<td>A device has left the PAN (shown on COO only)</td>
</tr>
<tr>
<td>ADSK: [&lt;EUI64&gt;, &lt;NWK addr&gt;]</td>
<td>Received Sink Advertisement</td>
</tr>
<tr>
<td>SREAD: [&lt;NWK addr&gt;, &lt;EUI64&gt;, &lt;Register&gt;, &lt;errorcode&gt;][=&lt;Data&gt;]</td>
<td>Reply to a remote S Register Read operation</td>
</tr>
<tr>
<td>SWRITE: [&lt;NWK addr&gt;, &lt;EUI64&gt;, &lt;errorcode&gt;]</td>
<td>Reply to a remote S Register Write operation</td>
</tr>
<tr>
<td>Bind: [&lt;NWK addr&gt;, &lt;status&gt;]</td>
<td>Create Binding Status</td>
</tr>
<tr>
<td>Unbind: [&lt;NWK addr&gt;, &lt;status&gt;]</td>
<td>Delete Binding Status</td>
</tr>
<tr>
<td>End Device Bind: [&lt;NWK addr&gt;, &lt;status&gt;]</td>
<td>End device binding status</td>
</tr>
<tr>
<td>DataMODE: [&lt;NWK addr&gt;, &lt;EUI64&gt;], &lt;errorcode&gt;</td>
<td>Datamode has been opened remotely</td>
</tr>
<tr>
<td>DataMODE: [&lt;NWK addr&gt;, &lt;EUI64&gt;, &lt;errorcode&gt;]</td>
<td>Response to an attempt to open data mode</td>
</tr>
<tr>
<td>OPEN</td>
<td>Data mode is open</td>
</tr>
<tr>
<td>CLOSED</td>
<td>Data mode is closed</td>
</tr>
<tr>
<td>TRACK: [&lt;EUI64 R&gt;, &lt;EUI64 S&gt;, &lt;RSSI&gt;, &lt;i/o read&gt;, &lt;ADC0&gt;, &lt;ADC1&gt;, &lt;Vcc&gt;, &lt;S46&gt;]</td>
<td>Tracking message: EUIs of receiver and sender, RSSI, input data, Vcc &amp; S46 counter of sender (only the low-order 16 bits of S46 are shown)</td>
</tr>
<tr>
<td>TRACK2: [&lt;EUI64 R&gt;, &lt;EUI64 S&gt;, &lt;RSSI&gt;, &lt;i/o read&gt;, &lt;S46&gt;]</td>
<td>Tracking message: EUIs of receiver and sender, RSSI, I/O data, and S46 counter of sender</td>
</tr>
<tr>
<td>PWRCHANGE: XXXX</td>
<td>Local node has changed Power Mode to XXXX</td>
</tr>
<tr>
<td>AddrResp: [errorcode] [, [&lt;NWK addr&gt;, &lt;EUI64&gt;]]</td>
<td>Response to an address request (also triggered when finding source routes)</td>
</tr>
<tr>
<td>RX: [&lt;EUI64&gt;, &lt;NWK addr&gt;, &lt;profileID&gt;, &lt;destinationEndpoint&gt;, &lt;SourceEndpoint&gt;, &lt;clusterID&gt;, &lt;length&gt;, &lt;payload&gt; [, &lt;RSSI&gt;, &lt;LQI&gt;]]</td>
<td>An incoming message not addressed to the AT command endpoint. EUI64 is only shown if included in network frame header</td>
</tr>
<tr>
<td>NM:ES REPORT WARNING</td>
<td>More than 16 energy scan reports have been recently received by the network manager indicating high packet loss</td>
</tr>
<tr>
<td>ENTERING BLOAD</td>
<td>Passthrough bootloading has been initiated from another node</td>
</tr>
</tbody>
</table>

Table 3: Prompt Overview
2.3 Device Overview

2.3.1 ZigBee types

Table 4 gives an overview of the ZigBee device types mentioned in this document.

<table>
<thead>
<tr>
<th>Device Types</th>
<th>ZigBee Naming Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>COO</td>
<td>Coordinator</td>
</tr>
<tr>
<td>FFD</td>
<td>Router</td>
</tr>
<tr>
<td>ZED</td>
<td>End Device (non sleepy)</td>
</tr>
<tr>
<td>SED</td>
<td>Sleepy End Device</td>
</tr>
<tr>
<td>MED</td>
<td>Mobile Sleepy end Device</td>
</tr>
</tbody>
</table>

Table 4: Device Overview

The terms Full Function device (FFD) and Reduced Function Device (RFD) are obsolete, but the abbreviations are retained in the R309X firmware to avoid problems with users’ legacy application software.

Each ETRX357 coordinator or router can support up to 30 End Devices, in any combination of Sleepy End Devices and Mobile End Devices.

Only end devices should be put into a low-power state because routers and the coordinator must always be powered up to maintain the network connectivity. ZigBee End Devices do not poll for data, instead their incoming messages are relayed immediately by their parent without being buffered. This means that ZEDs must not be put into a sleep mode.

2.3.2 Non-ZigBee types

Sink. The sink is a Telegesis feature. When a node is defined as a sink by setting S10 bit 4, it can broadcast its address to the rest of the network. Other nodes can then send messages to the sink node using AT+SCAST or various built-in functions. This simplifies the application software since it is not necessary to know the EUI64 of the sink in advance.

Routers discover the sink when (1) they receive a regular advertisement broadcast from the sink (2) they are commanded to send a message without knowing the sink address and bit 8 of S10 is set (the first sink-cast message is therefore lost) (3) the AT+SSINK command is used.

To reduce traffic to end devices they do not receive the advertisement broadcasts and are not informed of the sink address when they join the PAN. Instead they automatically search for the sink the first time they send a message to it, even if bit 8 of S10 is not set. The first message returns an error, though, as the sink address is unknown at that stage.

2.4 Addressing modes

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

EUI64. 16 hexadecimal characters. This is flashed on to the chip at manufacture and cannot be changed by the user. This can be compared to the permanent MAC address of an IP-based device.

Network address. 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. Otherwise known as the Node ID.
Address table entry. Range 00-06. Entry 05 is a sink address, entry 06 is the source address of the last received UCAST, SCAST or MCAST that arrived at endpoint 1 with profile C091 and cluster 0002 (ie the default Telegesis parameters).

Binding table entry. Range 10-24 (hexadecimal). Entry FE causes a search of the table for the first entry whose source endpoint and cluster ID matches registers S40 and S42.

FF. In many commands address FF represents the local device.
## 2.5 AT Command Overview

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

<table>
<thead>
<tr>
<th>Command Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module control and configuration</td>
</tr>
<tr>
<td><strong>ATI</strong></td>
</tr>
<tr>
<td><strong>ATZ</strong></td>
</tr>
<tr>
<td><strong>AT+REMZ</strong></td>
</tr>
<tr>
<td><strong>AT&amp;F</strong></td>
</tr>
<tr>
<td><strong>AT+BLOAD</strong></td>
</tr>
<tr>
<td><strong>AT+PASSTHROUGH</strong></td>
</tr>
<tr>
<td><strong>AT+RECOVER</strong></td>
</tr>
<tr>
<td><strong>ATS</strong></td>
</tr>
<tr>
<td><strong>ATREMS</strong></td>
</tr>
<tr>
<td><strong>ATSALL</strong></td>
</tr>
<tr>
<td><strong>AT+TOKDUMP</strong></td>
</tr>
<tr>
<td>Network control and configuration</td>
</tr>
<tr>
<td><strong>AT+ESCAN</strong></td>
</tr>
<tr>
<td><strong>AT+EN</strong></td>
</tr>
<tr>
<td><strong>AT+PANSCAN</strong></td>
</tr>
<tr>
<td><strong>AT+JN</strong></td>
</tr>
<tr>
<td><strong>AT+JPAN</strong></td>
</tr>
<tr>
<td><strong>AT+SN</strong></td>
</tr>
<tr>
<td><strong>AT+N</strong></td>
</tr>
<tr>
<td><strong>AT+NTABLE</strong></td>
</tr>
<tr>
<td><strong>AT+RTABLE</strong></td>
</tr>
<tr>
<td><strong>AT+IDREQ</strong></td>
</tr>
<tr>
<td><strong>AT+EUIREQ</strong></td>
</tr>
<tr>
<td><strong>AT+NODEDESC</strong></td>
</tr>
<tr>
<td><strong>AT+POWERDESC</strong></td>
</tr>
<tr>
<td><strong>AT+ACTEPDESC</strong></td>
</tr>
<tr>
<td><strong>AT+SIMPLETEDESC</strong></td>
</tr>
<tr>
<td><strong>AT+MATCHREQ</strong></td>
</tr>
<tr>
<td><strong>AT+ANNCE</strong></td>
</tr>
<tr>
<td><strong>AT+SR</strong></td>
</tr>
<tr>
<td><strong>AT+FNDNSR</strong></td>
</tr>
<tr>
<td><strong>AT+POLL</strong></td>
</tr>
<tr>
<td><strong>AT+REJOIN</strong></td>
</tr>
<tr>
<td><strong>AT+SN</strong></td>
</tr>
<tr>
<td><strong>AT+KEYUPD</strong></td>
</tr>
<tr>
<td><strong>AT+BECOMETC</strong></td>
</tr>
<tr>
<td><strong>AT+BECOMENM</strong></td>
</tr>
<tr>
<td><strong>AT+CCHANGE</strong></td>
</tr>
</tbody>
</table>
### Command Overview (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Messaging</strong></td>
<td></td>
</tr>
<tr>
<td>AT+ATABLE</td>
<td>Display Address Table</td>
</tr>
<tr>
<td>AT+ASET</td>
<td>Set Address Table Entry</td>
</tr>
<tr>
<td>AT+MTABLE</td>
<td>Display Multicast Table</td>
</tr>
<tr>
<td>AT+MSET</td>
<td>Set Multicast Table Entry</td>
</tr>
<tr>
<td>AT+BCAST</td>
<td>Transmit A Broadcast</td>
</tr>
<tr>
<td>AT+BCASTB</td>
<td>Transmit A Broadcast Of Binary Data</td>
</tr>
<tr>
<td>AT+UCAST</td>
<td>Transmit A Unicast</td>
</tr>
<tr>
<td>AT+UCASTB</td>
<td>Transmit A Unicast Of Binary Data</td>
</tr>
<tr>
<td>AT+SCAST</td>
<td>Transmit Data To The Sink</td>
</tr>
<tr>
<td>AT+SCASTB</td>
<td>Transmit Binary Data To The Sink</td>
</tr>
<tr>
<td>AT+SSINK</td>
<td>Search For A Sink</td>
</tr>
<tr>
<td>AT+MCAST</td>
<td>Transmit A Multicast</td>
</tr>
<tr>
<td>AT+MCASTB</td>
<td>Transmit A Multicast Of Binary Data</td>
</tr>
<tr>
<td>AT+DMODE</td>
<td>Enter Data Mode (Serial Link Mode)</td>
</tr>
<tr>
<td>+++</td>
<td>Leave Data Mode</td>
</tr>
<tr>
<td>AT+IDENT</td>
<td>Play A Tune On Remote Devboard</td>
</tr>
<tr>
<td>AT+SENDUCAST</td>
<td>Send A Raw ZCL/ZDO Unicast</td>
</tr>
<tr>
<td>AT+SENDUCASTB</td>
<td>Send A Raw Binary ZCL/ZDO Unicast</td>
</tr>
<tr>
<td>AT+SENDMCAST</td>
<td>Send A Raw ZCL/ZDO Multicast or Broadcast</td>
</tr>
<tr>
<td>AT+SENDMCASTB</td>
<td>Send A Raw Binary ZCL/ZDO Multicast or Broadcast</td>
</tr>
<tr>
<td>AT+INTERPAN</td>
<td>Send an Interpan Command</td>
</tr>
<tr>
<td>AT+RDATAB</td>
<td>Send Binary Raw Data</td>
</tr>
<tr>
<td><strong>Binding Management</strong></td>
<td></td>
</tr>
<tr>
<td>AT+LBTABLE</td>
<td>Display Local Binding Table</td>
</tr>
<tr>
<td>AT+BSET</td>
<td>Set Local Binding Table Entry</td>
</tr>
<tr>
<td>AT+BCLR</td>
<td>Clear Local Binding Table Entry</td>
</tr>
<tr>
<td>AT+BTABLE</td>
<td>Display Binding Table (ZDO)</td>
</tr>
<tr>
<td>AT+BIND</td>
<td>Create Binding on Remote Device</td>
</tr>
<tr>
<td>AT+UNBIND</td>
<td>Delete Binding on Remote Device</td>
</tr>
<tr>
<td>AT+EDBIND</td>
<td>Request End Device Binding (ZDO)</td>
</tr>
<tr>
<td><strong>Time-related commands</strong></td>
<td></td>
</tr>
<tr>
<td>AT+SETTIME</td>
<td>Set the Local Time</td>
</tr>
<tr>
<td>AT+GETTIME</td>
<td>Get the Local Time</td>
</tr>
<tr>
<td>AT+SYNCTIME</td>
<td>Synchronize the Local Time with Time Server</td>
</tr>
</tbody>
</table>

Table 5: Command Overview
### 2.6 Module Control & Configuration Commands

#### I – Display Product Identification Information

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATI</td>
<td>Telegesis &lt;DeviceName&gt; R&lt;Firmware Revision&gt; &lt;EUI64&gt; OK</td>
</tr>
</tbody>
</table>

Where <DeviceName> is the order code of the device, <Firmware Revision> is the firmware revision and <EUI64> is the Device’s IEEE 802.15.4 identifier

<table>
<thead>
<tr>
<th>SW release</th>
<th>R300</th>
</tr>
</thead>
</table>

#### Z – Software Reset

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATZ</td>
<td>JPAN:&lt;channel&gt;,&lt;PID&gt;,&lt;EPID&gt; OK or OK</td>
</tr>
</tbody>
</table>

Module Performs a 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.

<table>
<thead>
<tr>
<th>SW release</th>
<th>R300</th>
</tr>
</thead>
</table>

### +REMZ – Reset Remote Node (ETRX3 only)

**Execute Command**

```
AT+REMZ:<address>
```

**Response**

- `SEQ:XX`
- `OK`
- `ERROR<errorcode>`

**Prompt**

- `ACK:XX`
- `NACK:XX`

*Where* `<address>` can be the remote node’s EUI64, Network address or address table index.

Use on All Devices

Performs a soft reset on a remote node.

**SW release**

R309

### &F – Restore Factory Defaults

**Execute Command**

```
AT&F
```

**Response**

Module Performs a factory reset

All non-volatile S Registers are updated with their factory defaults and the node leaves the network it is currently joined to.

**SW release**

R300

### +BLOAD – Enter The Bootloader Menu

**Execute Command**

```
AT+BLOAD
```

**Response**

- `<entering bootloader>`

The device leaves the AT command line and enters the bootloader menu for downloading new firmware.

A description of the bootloading process can be found in the Development Kit Product Manual.

Please note that the bootloader will run at a baudrate of 115k2, no parity, 8 data bits regardless of the current serial port settings.

**SW release**

R300
**+PASSTHROUGH – Pass new Firmware Image To Remote Node**

**Execute Command**

AT+PASSTHROUGH:<EUI64>,<password>

**Use on:**
Source: FFD, COO  
Destination: FFD, COO, ZED

**Notes**
Passthrough is not possible to SEDs or MEDs or over multiple hops.  
The default password for R3xx nodes is "password".  
A description of the passthrough process can be found in the Development Kit Product Manual; it is the same procedure as cloning.  
The ETRX357(HR)-LRS module cannot be reliably upgraded by the passthrough process.

**Response**
PASSTHROUGH BLOAD... 
Please start .ebl upload image...
Remote Response
ENTERING BLOAD
or
ERROR<errorcode>

Where <errorcode> represents the error code explained in section 3.  
<password> represents the remote node’s 8-character password. After completion a soft reset is caused on the remote end.

**SW release**
R304

---

**+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 the Passthrough Bootloading operation was interrupted and the target device therefore remains in the bootloader.  
In case the target device has been reset channel 13 must be used for recovering.  
For more information on over-the-air firmware upgrading please refer to the Development Kit Manual.

**Response**
Recovering...

or

ERROR<errorcode>

Where <errorcode> represents the error code explained in section 3.  
Enters Passthrough mode to a remote node which is already in the bootloader.

**SW release**
R300
### S – S-Register Access

<table>
<thead>
<tr>
<th>Read Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSXX[x[x]]?</td>
<td>&lt;data&gt; OK</td>
</tr>
</tbody>
</table>

#### Examples

- `ATS00?`  
- `ATS0AE?`  
- `ATS1812?`  

XX is the S-Register which is to be read. As an option for all 16 bit registers it is also possible to address an individual bit only by specifying the bit number [x]. For all 32 bit registers it is possible to address an individual bit by specifying the bit number in hexadecimal [xx].

<table>
<thead>
<tr>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSXX[x[x]]=&lt;data&gt;[,&lt;password&gt;]</td>
<td>OK or ERROR:&lt;errorcode&gt;</td>
</tr>
</tbody>
</table>

#### Examples

- `ATS00=3FFC`  
- `ATS0AE=1:password`  

### Notes

- Some S-Registers require a password for write access. See S-Register description for details. **The default password for R3xx is “password”**.
- Some S-Registers are read-only and will return an error if you are trying to write to them. When writing an individual bit by specifying [x[x]], <data> can only be either 0 or 1.

### Response

- The module displays the contents of S-register xx or an error message, where <errorcode> represents the error code explained in section 3. All 16- and 32-bit registers can also be accessed bit by bit. In order to do this [x[x]] may specify the bit which is to be read. The result when reading a single bit will always be 0 or 1.
- 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> represents the error code explained in section 3. For all 16- and 32-bit registers individual bits can also be set or cleared by specifying the bit using hexadecimal [x[x]] and setting it to either 0 or 1.
## REMS – Remote S-Register Access

### Read Command

**ATREMS:<address>,XX[X[x]]?**

**Examples**

- `ATREMS:000D6F00000AAC93,00?`
- `ATREMS:000D6F00000AAC93,0AE?`
- `ATREMS:000D6F00000AAC93,1812?`

Where `<address>` can be the remote node’s EUI64, Network address or address table index and XX is the S-Register which is to be read. As an option for all 16 bit registers it is also possible to address an individual bit only by specifying the bit number [X]. For all 32 bit registers it is possible to address an individual bit by specifying the bit number in hexadecimal [xx]. The result when reading a single bit will always be 0 or 1.

**Note**

Also the local node can be the target of this command (e.g. use address table entry FF as the address)

### Write Command

**ATREMS:<address>,XX[X[x]]=<data> [,<password>]**

**Examples**

- `ATREMS:000D6F0000012345,00=3FFC`
- `ATREMS:000D6F0000012345,0AE=1:password`

Where `<address>` can be the remote node’s EUI64, Network address or address table index and XX is the S-Register which is to be written. As an option for all 16- and 32-bit registers it is also possible to address an individual bit only by specifying the bit number [x[x]].

**Notes**

Some S-Registers require a password for write access. See S-Register description for details. **The default password for R3xx is “password”**.

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

When writing an individual bit by specifying [x[x]], `<data>` can only be either 0 or 1.

### Response

**SEQ:XX OK**

or **ERROR:<errorcode>**

The module asks for the contents of the remote S-register using a unicast. The sequence number of the unicast is displayed (an ACK or NACK prompt will follow). `<errorcode>` represents the error code explained in section 3.

**Prompt**

**SREAD:<Network address>,<EUI64>,<Register>,<errorcode>[=<Data>]**

Where Network address is the remote Network address, EUI64 is the remote EUI64, Register is the S-Register which was read and `<errorcode>` is indicating the success (00) or failure of the read operation. The contents of the remote S-Register are following in case of a successful read only.

**Response**

**SEQ:XX OK**

or **ERROR:<errorcode>**

The data is written to the remote 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. The sequence number of the unicast is displayed (an ACK or NACK prompt will follow). `<errorcode>` represents the error code explained in section 3.

**Prompt**

**SWRITE:<Network address>,<EUI64>,<errorcode>**

Where `<Network address>` is the remote Network address, `<EUI64>` is the remote EUI64. Only in case the errorcode is 00 the write operation has been completed successfully.
### SALL – Remote S-Register Access

**Write Command**

```
ATSALL:<group ID>,XX[x][x]=<data>[,<password>]
```

**Examples**

```
ATSALL:FFFF,00=3FFC
ATSALL:FFFC,0AE=1:password
```

Where group IDs are remote node’s multicast IDs or
FFFF - Broadcast to all devices
FFFFD - Broadcast to all non-sleepy devices
FFFC – Broadcast to all Routers

**Notes**

Some S-Registers require a password for write access. See S-Register description for details. **The default password for R3xx is “password”**. Some S-Registers are read-only and cannot be written to.

<table>
<thead>
<tr>
<th>SW release</th>
<th>R300</th>
</tr>
</thead>
</table>

**Response**

OK or ERROR:<errorcode>

The data is written to the remote S-register number XX on all nodes addressed by the multicast group ID. The data format for each individual S-register is given in the S-register description. <errorcode> represents the error code explained in section 3. For all 16- and 32-bit registers individual bits can also be set or cleared by specifying the bit using hexadecimal [x[x]] and setting it to either 0 or 1.

### +TOKDUMP – Display All S-Registers

**Execute Command**

```
AT+TOKDUMP
```

**Notes**

Only used on the local node. You cannot display all the registers of a remote device.

<table>
<thead>
<tr>
<th>SW release</th>
<th>R300</th>
</tr>
</thead>
</table>

**Response**

<data>

OK

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 4.
## 2.7 Network Control & Configuration Commands

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

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>AT+ESCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use on:</td>
<td>All nodes</td>
</tr>
<tr>
<td>Notes</td>
<td>Scanning all channels can take up to 4 seconds. The results are the background radio power in each channel, not the RSSI of incoming ZigBee packets.</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>+ESCAN:</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:XX</td>
</tr>
<tr>
<td>12:XX</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>26:XX</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

or **ERROR:<errorcode>**

<errorcode> represents the error code explained in section 3. XX represents the average energy on the respective channel (see description in Section 7). Channels masked out in S00 are not scanned.

**SW release**

R300

### +EN – Establish Personal Area Network

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>AT+EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use on:</td>
<td>All nodes which are not part of a PAN</td>
</tr>
<tr>
<td>Note</td>
<td>When issuing this command the local device becomes a Coordinator (and Trust Centre). 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.</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>JPAN:&lt;channel&gt;,&lt;PID&gt;,&lt;EPID&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

or **ERROR:<errorcode>**

<errorcode> represents the error code explained in section 3.

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.

**SW release**

R300
### +PANSCAN – Scan For Active PANs

**Execute Command**

`AT+PANSCAN[[:b][,XXXX][,dd]]`

When specifying a value of 0 or 1 for b only responses from nodes with the joining status set accordingly will be shown. Specifying a channel mask using XXXX will override the setting of S00 for this specific command. Using dd it is possible to modify the scan time per channel (default = 3).

(00 = 31ms, 01 = 46ms, 02 = 77ms, 03 = 138ms, 04 = 261ms, 05 = 507ms, 06 = 998ms)

**Examples**

- `AT+PANSCAN`
- `AT+PANSCAN:1`
- `AT+PANSCAN:1,03`
- `AT+PANSCAN:03`
- `AT+PANSCAN:FF0F`
- `AT+PANSCAN:FF0F,03`
- `AT+PANSCAN:1,FF0F`
- `AT+PANSCAN:1,FF0F,03`

**Use on:**

- All nodes

**Note**

Scanning for active PANs can take up to 4 seconds when dd=3

**SW release**

- R300

---

**Response**

`+PANSCAN:<channel>,<PID>,<EPID>,XX,b`

**OK**

or

`+PANSCAN:<channel>,<PID>,<EPID>,XX,b, <rssi>,<LQI>`

**OK**

or **ERROR:**<errorcode>

<errorcode> represents the error code explained in section 2.10. The node gives a list of all PANs found. <channel> represents the channel, <PID> the PAN ID, <EPID> the extended PAN ID, XX the ZigBee stack profile (00 = Custom, 01 = ZigBee, 02 = ZigBee PRO) and b indicates whether the network is allowing additional nodes to join (1 = joining permitted). The node does not join any of the PANs found.

If bit E of S0F is set the response includes RSSI and LQI.
### +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**

<table>
<thead>
<tr>
<th>JPAN: &lt;channel&gt;, &lt;PID&gt;, &lt;EPID&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

or **ERROR: <errorcode>**

<errorcode> represents the error code explained in section 3.

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.

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

#### Remote Action

**On the Trust Centre / Coordinator**

<table>
<thead>
<tr>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWNODE: &lt;node EUI64&gt;,</td>
</tr>
<tr>
<td>&lt;Network address&gt;, &lt;parent EUI64&gt;</td>
</tr>
</tbody>
</table>

**SW release**

R300

---

### +JPAN – Join Specific PAN

**Execute Command**

AT+JPAN: <channel>, <PID or EPID>

**Examples**

AT+JPAN: 20, 1234
AT+JPAN: 24, 0793E14FFB220A38

**Use on**
All nodes which are not part of a PAN

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

The JPAN command ignores the channel mask in register S00 and the PID and EPID settings in S02 and S03.

**Response**

<table>
<thead>
<tr>
<th>JPAN: &lt;channel&gt;, &lt;PID&gt;, &lt;EPID&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

or **ERROR: <errorcode>**

<errorcode> represents the error code explained in section 3.

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.

#### Remote Action

**On the Trust Centre / Coordinator**

<table>
<thead>
<tr>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWNODE: &lt;node EUI64&gt;,</td>
</tr>
<tr>
<td>&lt;Network address&gt;, &lt;parent EUI64&gt;</td>
</tr>
</tbody>
</table>

**SW release**

R300
### +SJN – Silent Join

**Execute Command**

```
AT+SJN:<channel>,<TC EUI64>,
<NM Network address>,<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 the supplied data is correct. The node can only join as a router, not an end device.

- `<channel>` is a decimal number
- Other parameters are hexadecimal

**Example**

```
AT+SJN:11,000D6F00000AAAD0,AFFE,00
```

**Use on**

All joining Devices

**Response**

```
JPAN:<channel>,<PID>,<EPID>
```

OK or ERROR:<errorcode>

> <errorcode> represents the error code explained in section 3. The local node will become part of the network with the channel specified in `<channel>`, the trust centre EUI64 specified in `<TC EUI64>`, the Network address of the network manager specified in `<NM Network address>`, 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. Joining is still possible if the network update ID is incorrect. It is assumed that the key-sequence-number of the network key is 0 when issuing this command.

**SW release**

R305

---

### +DASSL – Disassociate Local Device From PAN

**Execute Command**

```
AT+DASSL
```

**Note**

Use with care on a Coordinator. It will not be able to rejoin the PAN.

**Use on**

All Devices

**Response**

```
OK or ERROR<errorcode>
```

**Prompt**

LeftPAN

<errorcode> represents the error code explained in section 3. Instruct local device to leave the PAN.

**SW release**

R300
### +DASSR – Disassociate Remote Node from PAN (ZDO)

**Execute Command**

```
AT+DASSR:<address>
```

Where `<address>` can be a node’s EUI64, Network address or address table index.

**Note**
Use with care when targeting a Coordinator. It will not be able to rejoin the PAN.

**Use on**
All Devices

**Remote Action**
Node leaves PAN

**Response**
```
SEQ:XX
OK
```

or
```
ERROR:<errorcode>
```

`<errorcode>` represents the error code explained in section 3.

Instruct device to leave the PAN.

**Prompt**
`LeftPAN`

**SW release**
R300

---

### +N – Display Network Information

**Read Command**

```
AT+N?
```

**Use on**
All Devices

**Response**
```
+N=<devicetype>,<channel>,<power>,<PID>,<EPID>
```

or
```
+N=NoPAN
```

followed by
```
OK
```

`<devicetype>` represents the node’s functionality in the PAN (FFD, COO, ZED, SED, MED), `<power>` the node’s output power in dBm, `<channel>` the IEEE 802.15.4 radio channel (11-26), `<PID>` the node’s PAN ID and `<EPID>` the node’s extended PAN ID.

**SW release**
R302
### +NTABLE – Display Neighbour Table

**Read Command**

AT+NTABLE:XX,<address>

Where XX is the start index of the remote LQI table and <address> can be the remote node’s EUI64, Network address or address table entry.

**Note**

Also the local node can be the target of this command (e.g. use address table entry FF as the address)

**Use on**

FFD, COO as the target device

**Response**

SEQ:XX
OK or ERROR<errorcode>

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.

**Prompt (example)**

NTable:<Network address>,<errorcode>
Length:03

<table>
<thead>
<tr>
<th>No.</th>
<th>Type</th>
<th>EUI</th>
<th>ID</th>
<th>LQI</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.</td>
<td>FFD</td>
<td>000D6F000015896B</td>
<td>BC04</td>
<td>FF</td>
</tr>
<tr>
<td>01.</td>
<td>FFD</td>
<td>000D6F00000B3E77</td>
<td>739D</td>
<td>FF</td>
</tr>
<tr>
<td>02.</td>
<td>FFD</td>
<td>000D6F0000AAD11</td>
<td>75E3</td>
<td>FF</td>
</tr>
</tbody>
</table>

In this example the neighbour table of the remote node with the short ID shown in <Network address> 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.

**SW release**

R302
### AT+RTABLE – Display Routing Table

<table>
<thead>
<tr>
<th>Read Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+RTABLE:XX,&lt;address&gt;</td>
<td>SEQ:XX OK or ERROR&lt;errorcode&gt;</td>
</tr>
</tbody>
</table>

Where XX is the start index of the remote Routing table and <address> can be the remote node’s EUI64, Network address or address table entry.

**Note**
Also the local node can be the target of this command (e.g. use address table entry FF as the address)

**Use on**
FFD, COO as the target device

**Response**

This command requests the target node to respond by listing its routing table starting from the requested index.

**Prompt (example)**

RTable:<Network address>,<errorcode>
Length:40
No.| Dest | Next | Status
---|------|------|------
00.| 1234 | ABCD | 00
01.| 4321 | 739D | 00
02.| 0000 | 0000 | 03

In this example the routing table of the remote node with the short ID shown in <Network address> 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 Specification.

In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.

**SW release**
R303
### +IDREQ – Request Node’s Network address

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+IDREQ:&lt;Address&gt;[,XX]</td>
<td>OK or ERROR:&lt;errorcode&gt;</td>
<td>AddressReq:&lt;errorcode&gt; [&lt;Network address&gt;,&lt;EUI64&gt;] [nn. &lt;Network address&gt;]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;EUI64&gt; is the Remote node’s EUI64 and &lt;Network address&gt; is its Network address. In case an extended response has been requested the requested Network addresses from the associated devices list are listed as well.</td>
</tr>
</tbody>
</table>

**Note**
- Providing FF as an address table entry addresses the local node
- Use on All Devices

**Response**
- OK or ERROR:<errorcode>
- <errorcode> represents the error code explained in section 3.

**Prompt**
- AddressReq:<errorcode> [<Network address>,<EUI64>] [nn. <Network address>]
# AT Command Manual (Rev 3.09)

## +EUIREQ – Request Node’s EUI64

**Execute Command**

```
AT+EUIREQ:< Address>,
<Network address>[,XX]
```

Where `<Address>` is the EUI64, Network address or address table entry of the node which is to be interrogated about the node with the Network address specified in `<Network address>`. 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 (ie children).

Sends a unicast to obtain the specified device’s EUI64 and optionally also elements of its associated devices list (extended response).

**Note**

Providing FF as an address table entry addresses the local node.

To find the EUI64 of an end device use its parent’s address as the `<Address>` parameter.

**Use on**

All Devices

**Response**

SEQ:XX

OK

or ERROR:<errorcode>

<errorcode> represents the error code explained in section 3.

**Prompt**

```
AddrResp:<errorcode>
[.<Network address>,<EUI64>]
[dd. <Network address>]
```

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 <Network address> is its Network address. In case an extended response has been requested the requested Network addresses from the associated devices list are listed.

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.

---

**SW release**

R302
### +NODEDESC – Request Node’s Descriptor

**Execute Command**

\[ \text{AT} + \text{NODEDESC:<Address>, <Network address>} \]

Where `<Address>` is the EUI64, Network address or Address table entry of the node which is to be interrogated about the node with the Network address specified in `<Network address>`.

Sends a unicast to obtain the specified device’s node descriptor.

**Note**

Providing FF as an address table entry addresses the local node

**Use on**

All Devices

**Response**

<table>
<thead>
<tr>
<th>Prompt (example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NodeDesc:&lt;Network address&gt;,&lt;errorcode&gt;</td>
</tr>
<tr>
<td>Type:FFD</td>
</tr>
<tr>
<td>ComplexDesc:No</td>
</tr>
<tr>
<td>UserDesc:No</td>
</tr>
<tr>
<td>APSFlags:00</td>
</tr>
<tr>
<td>FreqBand:40</td>
</tr>
<tr>
<td>MacCap:8E</td>
</tr>
<tr>
<td>ManufCode:1010</td>
</tr>
<tr>
<td>MaxBufSize:52</td>
</tr>
<tr>
<td>MaxInSize:0080</td>
</tr>
<tr>
<td>SrvMask:0000</td>
</tr>
<tr>
<td>MaxOutSize:0080</td>
</tr>
<tr>
<td>DescCap:00</td>
</tr>
</tbody>
</table>

In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.

<Network address> is the Remote node’s Network address. 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 specification.

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.

**SW release**

R302
### +POWERDESC – Request Node’s Power Descriptor

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+POWERDESC:&lt;Address&gt;,&lt;Network address&gt;</td>
<td>SEQ:XX OK</td>
</tr>
</tbody>
</table>

Where `<Address>` is the EUI64, Network address or Address table entry of the node which is to be interrogated about the node with the Network address specified in `<Network address>`.

Sends a unicast to obtain the specified device’s power descriptor.

<table>
<thead>
<tr>
<th>Use on All Devices</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PowerDesc: <code>&lt;Network address&gt;,&lt;errorcode&gt; [,&lt;PowerDescriptor&gt;]</code></td>
</tr>
</tbody>
</table>

In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.

- `<Network address>` is the Remote node’s Network address. In addition the power descriptor is displayed as a 16 bit hexadecimal number as described in section 2.3.2.4. of the ZigBee specification.

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.

<table>
<thead>
<tr>
<th>SW release</th>
<th>R302</th>
</tr>
</thead>
</table>

©2015 Telegesis (UK) Ltd

- 31 -

AT Command Manual (Rev 3.09)
**+ACTEPDESC – Request Node’s Active Endpoint List**

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+ACTEPDESC:&lt;Address&gt;, &lt;Network address&gt;</td>
<td>SEQ:XX OK</td>
</tr>
</tbody>
</table>

Where `<Address>` is the EUI64, Network address or Address table entry of the node which is to be interrogated about the node with the Network address specified in `<Network address>`.

Sends a unicast to obtain the specified device’s active endpoint list.

<table>
<thead>
<tr>
<th>SW release</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>R302</td>
<td>ActEpDesc: &lt;Network address&gt;, &lt;errorcode&gt;[,XX,…]</td>
</tr>
</tbody>
</table>

**Use on All Devices**

In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.

<Network address> is the Remote node’s Network address. In addition all active endpoints are listed as 8-bit hexadecimal numbers separated by commas.

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.
<table>
<thead>
<tr>
<th><strong>+SIMPLEDESC</strong> – Request Endpoint’s Simple Descriptor (ZDO)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execute Command</strong></td>
</tr>
<tr>
<td>AT+SIMPLEDESC:&lt;Address&gt;,&lt;Network address&gt;,&lt;XX&gt;</td>
</tr>
<tr>
<td>Where &lt;Address&gt; is the EUI64, Network address or Address table entry of the node which is to be interrogated about the node with the Network address specified in &lt;Network address&gt; and XX is the number of the endpoint, which simple descriptor is to be read.</td>
</tr>
<tr>
<td>Sends a unicast to obtain the specified device’s active endpoint list.</td>
</tr>
<tr>
<td><strong>Response</strong></td>
</tr>
<tr>
<td>SEQ:XX OK</td>
</tr>
<tr>
<td>or ERROR:&lt;errorcode&gt;</td>
</tr>
<tr>
<td>&lt;errorcode&gt; represents the error code explained in section 3.</td>
</tr>
<tr>
<td><strong>Prompt</strong></td>
</tr>
<tr>
<td>SimpleDesc:&lt;Network address&gt;,&lt;errorcode&gt;EP:XX</td>
</tr>
<tr>
<td>ProfileID:XXXX</td>
</tr>
<tr>
<td>DeviceID:XXXXvXX</td>
</tr>
<tr>
<td>InCluster:&lt;Cluster List&gt;</td>
</tr>
<tr>
<td>OutCluster:&lt;Cluster List&gt;</td>
</tr>
<tr>
<td>In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode</td>
</tr>
<tr>
<td>&lt;Network address&gt; is the Remote node’s Network address. In addition all active endpoints are listed as 8 bit hexadecimal numbers separated by commas.</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td><strong>Use on All Devices</strong></td>
</tr>
<tr>
<td><strong>SW release</strong></td>
</tr>
<tr>
<td>R302</td>
</tr>
</tbody>
</table>
**+MATCHREQ – Find Nodes which Match a Specific Descriptor**

**(ZDO)**

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+MATCHREQ: &lt;ProfileID&gt;, &lt;NumInClusters&gt; [,InClusterList], &lt;NumOutClusters&gt; [,OutClusterList]</td>
<td>OK</td>
<td>MatchDesc:&lt;Network address&gt;, &lt;errorcode&gt;,XX,…</td>
</tr>
<tr>
<td></td>
<td>or ERROR:&lt;errorcode&gt;</td>
<td>In case of an error an errorcode other than 00 will be displayed and the prompt will end after the errorcode.</td>
</tr>
</tbody>
</table>

**Where**

- `<ProfileID>` Required profile ID of the device being searched for followed by a specification of required input and output clusters.
- `<InClusterList>` and `<OutClusterList>` must be 2 hexadecimal digits

**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).**

**Examples**

AT+MATCHREQ:C091,01,0002,02,0004,000B
AT+MATCHREQ:C091,00,01,0004
AT+MATCHREQ:C091,01,0002,00

**Use on**

- All Devices

**SW release**

- R302
<table>
<thead>
<tr>
<th>+ANNCE – Announce Local Device In The Network (ZDO)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execute Command</strong></td>
</tr>
<tr>
<td><strong>AT+ANNCE</strong></td>
</tr>
<tr>
<td>Send a ZigBee device announce Broadcast announcing the local node on the network.</td>
</tr>
<tr>
<td><strong>Use on All Devices</strong></td>
</tr>
<tr>
<td><strong>Response</strong></td>
</tr>
<tr>
<td><strong>OK or ERROR&lt;errorcode&gt;</strong></td>
</tr>
<tr>
<td>&lt;errorcode&gt; represents the error code explained in section 3.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remote Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prompt</strong></td>
</tr>
<tr>
<td>FFD:&lt;EUI64&gt;,&lt;Network address&gt;[,syy,zz]</td>
</tr>
<tr>
<td>MED:&lt;EUI64&gt;,&lt;Network address&gt;[,syy,zz]</td>
</tr>
<tr>
<td>SED:&lt;EUI64&gt;,&lt;Network address&gt;[,syy,zz]</td>
</tr>
<tr>
<td>ZED:&lt;EUI64&gt;,&lt;Network address&gt;[,syy,zz]</td>
</tr>
<tr>
<td>The prompt above will be displayed on all nodes which can hear the announcement. In case bit C of register S10 is set the RSSI level (syy dBm) and LQI (zz in hexadecimal) of the last hop are displayed. For a description of the LQI reading please see section 7.</td>
</tr>
<tr>
<td>&lt;EUI64&gt; is the identifier and &lt;Network address&gt; the Network address of the sending device</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R302</td>
</tr>
</tbody>
</table>
### AT+SR – Set Source Route to Remote Device

**Execute Command**

```
AT+SR:<Network address>, <Network address>, ...
```

Set the source route of a message sent to a remote device, starting with the Network address of the remote device, followed by all Network addresses on the route from the remote node to the local node starting at the remote end.

**Note**

Setting up invalid routes may lead to listed devices becoming unavailable. To confirm a route use AT+FNDSR.

**Use on**

All Devices

**Response**

- **OK**
- **ERROR<errorcode>**

<errorcode> represents the error code explained in section 3.

Stores route information for up to 30 hops which will be used when sending any message to a remote node, which is part of the listed devices.

**SW release**

R300

---

### AT+FNDSR – Find the Source Route to a remote device

**Execute Command**

```
AT+FNDSR:<address>
```

Where `<address>` can be the remote node's EUI64 or address table index.

Tries to find source route information to the specified device by sending a ZDO request to the remote device and thus triggering a reply.

**Use on**

COO, Sink

**Response**

- **OK**
- **ERROR<errorcode>**

Prompt

```
SR:XX,<EUI64>,<Network address>,<Network address>...
```

Where XX represents the number of hops to the remote node, EUI64 its EUI64 number followed by a list of Network addresses starting with the remote node listing all nodes along the path to the local node.

<errorcode> represents the error code explained in section 3.

**SW release**

R302
### +POLL – Poll The Parent Device

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>AT+POLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poll the parent device for new data.</td>
<td></td>
</tr>
</tbody>
</table>

**Note**  
Action 0010/8010 is recommended for periodic polling using the built-in timers.

**Use on**  
SED, MED

**Response**  
OK  
or  
ERROR<errorcode>

<errorcode> represents the error code explained in section 3.

**SW release**  
R300

### +REJOIN – Rejoin the network

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>AT+REJOIN:b</th>
</tr>
</thead>
<tbody>
<tr>
<td>If b is set to 0 join without the known network key (unencrypted) and if b is set to 1 join encrypted.</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**  
Polling a parent on an end device that has lost its parent will automatically call AT+REJOIN:1. Furthermore functionality 0012 and 0013 make use of this command.

**Use on**  
All devices except COO

**Response**  
OK  
or  
ERROR<errorcode>

If the contact with the network has been lost because an end device has lost its parent, the network has changed channel, or updated its encryption key the command AT+REJOIN can be used to rejoin the network.

<errorcode> represents the error code explained in section 3.

**SW release**  
R300
### AT+SN – Scan Network

**Execute Command**

\[
\text{AT+SN[:nn]} \\
\]

All Telegesis devices which are up to \( nn \) hops away are listed. If \( nn = 01 \) only direct neighbours will reply and \( nn = 00 \) will search the entire network.

**Notes**

- When no parameter is specified for \( nn \), 30 is used by default.
- If used on nodes other than the COO and a sink the command may be unreliable.

**Use on**

COO, Sink

**Response**

OK or ERROR<errorcode>

**Prompts**

- FFD:<EUI64>,<Network address>[,syy,zz]
- MED:<EUI64>,<Network address>[,syy,zz]
- SED:<EUI64>,<Network address>[,syy,zz]
- ZED:<EUI64>,<Network address>[,syy,zz]

**Parameters**

- \( nn \) ranging from 00 to 30

- \(<\text{errorcode}>\) represents the error code explained in section 3. In case bit C of register S10 is set the RSSI level (syy in dBm) and LQI (zz in hexadecimal) of the last hop are displayed. For a description of the LQI reading please see section 7. Source route messages may also be displayed.

**SW release**

R302

### AT+KEYUPD – Update the Network Key

**Execute Command**

\[
\text{AT+KEYUPD} \\
\]

Updates the Network Key with a new key. If the value in S08 is non-zero and is not the current key, it will be used for the updated key. If S08 is zero or the current key, a random value will be generated.

**Note**

Can only be used on the Trust Centre

**Use on**

Trust Centre

**Response**

OK or ERROR<errorcode>

- \(<\text{errorcode}>\) represents the error code explained in section 3.

**SW release**

R302
### +BECOMETC – Make Local Device the Trust Centre

#### Execute Command

**AT+BECOMETC**

Local Device takes over the Trust Centre. Can only be used if no other device in the network is Trust Centre (i.e. the network has been started in distributed Trust Centre mode).

**Notes**

Can only be used if Network has been started in distributed Trust Centre mode (bit 9 of S0A set).

AT+BECOMETC causes the network key to be updated.

**Use on**

Router that established the PAN in distributed TC Mode

<table>
<thead>
<tr>
<th>SW release</th>
<th>R302</th>
</tr>
</thead>
</table>

#### Response

**OK**

or

**ERROR<errorcode>**

<errorcode> represents the error code explained in section 3.

---

### +BECOMENM – Make the local device Network Manager

#### Execute Command

**AT+BECOMENM**

Local Device takes over role of Network Manager. By default the COO is the Network Manager, but any other router in the network can take over this responsibility. The Network Manager can change the radio channel and the PAN ID.

**Use on**

Router

<table>
<thead>
<tr>
<th>SW release</th>
<th>R304</th>
</tr>
</thead>
</table>

#### Response

**OK**

or

**ERROR<errorcode>**

<errorcode> represents the error code explained in section 3.
### +CCHANGE – Change the network’s channel

**Execute Command**

\[ \text{AT+CCHANGE[}[:XX] \]

Ask all nodes in the network to change their 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)

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

**Use on**

Network Manager

**Parameters**

Optional XX ranging from 0B to 1A

**Response**

OK or **ERROR<errorcode>**

<errorcode> represents the error code explained in section 3.

**Use on**

Network Manager

**SW release**

R304

---

### 2.8 Messaging

### +ATABLE – Display Address Table

**Read Command**

\[ \text{AT+ATABLE} \]

**Notes**

Entry 05 contains the address of the node’s sink. The user can overwrite it to manually select a different sink.

Entry 06 contains the address of the node sending the most recently received UCAST, SCAST or MCAST.

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

**Use on**

All Devices

**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.

**Use on**

All Devices

**SW release**

R300
### ASET – Set Address Table Entry

**Write Command**

```
AT+ASET:XX,<Network address>,<EUI64>
```

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

**Use on**

All Devices

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
</tr>
</tbody>
</table>

**Response**

- **OK**
- or **ERROR:<errorcode>**

<errorcode> represents the error code explained in section 3.

---

### MTABLE – Display Multicast Table

**Read Command**

```
AT+MTABLE
```

**Response**

<table>
<thead>
<tr>
<th>No.</th>
<th>ID</th>
<th>EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1234</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>02</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>03</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>04</td>
<td>0000</td>
<td>00</td>
</tr>
</tbody>
</table>

**OK**

The multicast table contains all multicast IDs which will be received by the local node.

**Use on**

All Devices

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
</tr>
</tbody>
</table>

---

### MSET – Set Multicast Table Entry

**Write Command**

```
AT+MSET:XX,<ID>,<endpoint>
```

Where XX is the index number of the multicast-table entry which is to be written. For the AT-Command interface operation the endpoint should always be set to 01.

Note

SEDs and MEDs cannot receive multicast messages

**Use on**

All Devices

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
</tr>
</tbody>
</table>
**AT+BCAST – Transmit A Broadcast**

**Execute Command**

\[ \text{AT+BCAST:nn,<data>} \]

**Example**

\[ \text{AT+BCAST:00,Hello world} \]

**Response**

OK or ERROR<errorcode>

Where <errorcode> represents the error code explained in section 3.

**Parameters**

- **nn** ranging from 00 to 30

A maximum of 82 bytes are sent (with attached EUI only 74 bytes). The response OK shows successful transmission. Successful transmission does not guarantee successful reception. To make sure data has been received by a specific node use a unicast message. Only neighbours which are up to \( nn \) hops away will receive the broadcast. If \( nn = 01 \) only direct neighbours will receive the broadcast and if \( n = 00 \) the entire network will (max. 30 hops).

**Use on:**

All devices

**Remote action**

**Prompt**

\[ \text{BCAST:}\{\text{EUI64}\},\text{length}==\text{data}\]

or

\[ \text{BCAST:}\{\text{EUI64}\},\text{length}==\text{data}\],\text{RSSI},<\text{LQI}>\]

Every node in the PAN which has received the broadcast message will prompt the above message where <EUI64> is the address of the sender, <length> is the length of the payload and <data> is the data which was attached to the broadcast. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

**SW release**

R300
### AT+BCASTB – Transmit A Broadcast Of Binary Data

**Execute Command**

`AT+BCASTB:XX,nn`

Where `nn` is the number of hops the message will travel and `XX` is the number (in hexadecimal) of data bytes to be sent.

**Note**

This command is particularly useful if the data may contain `<CR>` and `<Backspace>` characters.

**Use on**

All Devices

**Response**

> `<data being entered>`

OK

or **ERROR:** `<errorcode>`

After the ‘>’ prompt a number of `XX` characters are expected to be entered. `<errorcode>` represents the error code explained in section 3. (In case bit 9 of S10 is set a timeout error is generated if no character is received for 1 second.)

**Parameters**

- `XX` ranging from 00 to 52 (hexadecimal)
- `nn` ranging from 00 to 30 (decimal)

A maximum of 82 bytes are sent (with attached EUI only 74 bytes). The response OK shows successful transmission. Successful transmission does not guarantee successful reception. To make sure data has been received by a specific node use a unicast message. Only neighbours which are up to `nn` hops away will receive the broadcast. If `nn`=01 only direct neighbours will receive the broadcast and if `n`=00 the entire network will (max 30 hops).

**Remote action**

Prompt

`BCAST:<EUI64>,<length>=<data>`

or

`BCAST:[<EUI64>,]<length>=<data>,<RSSI>,<LQI>`

Every node in the PAN which has received the broadcast message will prompt the above message where `<EUI64>` is the address of the sender and `<length>` is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

**SW release**

R300
### +UCAST – Transmit A Unicast

**Execute Command**

AT+UCAST:<address>=<data>

**Example**

AT+UCAST:000D6F0000012345=Hello

Where `<address>` can be the remote node’s EUI64, Network address or address table index.

**Note**

Unicasts can be addressed either by referencing the recipient’s EUI64, Network address or an entry in the address table. The maximum payload is 82 bytes. It is reduced by 8 bytes when appending the EUI to the network header (default) and also it is reduced by 2 bytes per hop in case a source route is known. The latter event can neither be suppressed nor foreseen.

Up to 10 unicasts may be in flight at one time.

Unicasts can travel up to 30 hops.

**Use on**

All Devices

**Remote action**

<table>
<thead>
<tr>
<th>Response</th>
<th>SEQ:XX</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>or</td>
<td>ERROR:&lt;errorcode&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Where `<errorcode>` represents the error code explained in section 3.

**Prompt**

ACK:XX

or NACK:XX

Up to 82 bytes are sent to the node up to 30 hops away. On successful transmission the user is given the transmission’s sequence number followed by “OK”. The user is then prompted “ACK” on receipt of an acknowledgement or “NACK” in case the message was not acknowledged. A NACK does not guarantee that the message has not reached its destination.

If bit B of S10 is set, “SEQ”, “ACK” and “NACK” are not reported. “OK” means that the message has been acknowledged by the destination.

**Prompt**

UCAST:[<EUI64>,]<length>=<data>

or

UCAST:[<EUI64>,]<length>=<data>, <RSSI>,<LQI>

Where `<EUI64>` is the address of the sender and `<length>` is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.
<table>
<thead>
<tr>
<th>Command: +UCASTB — Transmit A Unicast Of Binary Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execute Command</strong></td>
</tr>
<tr>
<td>AT+UCASTB:XX,&lt;address&gt;</td>
</tr>
<tr>
<td>Where &lt;address&gt; can be the remote node’s EUI64, Network address or address table index and XX is the number (in hexadecimal) of data bytes to be sent.</td>
</tr>
<tr>
<td><strong>Notes</strong></td>
</tr>
<tr>
<td>This command is particularly useful if the data may contain &lt;CR&gt; and &lt;Backspace&gt; characters.</td>
</tr>
<tr>
<td>The ACK and/or NACK prompt can be disabled in S0E.</td>
</tr>
<tr>
<td>Unicasts can be addressed either by referencing the recipient’s EUI64, Network address or an entry in the address table.</td>
</tr>
<tr>
<td>The maximum payload is 82 bytes. It is reduced by 8 bytes when appending the EUI to the network header (default) and also it is reduced by 2 bytes per hop in case a source route is known. The latter event can neither be suppressed nor foreseen.</td>
</tr>
<tr>
<td>Up to 10 unicasts may be in flight at one time.</td>
</tr>
<tr>
<td>Unicasts can travel up to 30 hops.</td>
</tr>
<tr>
<td>Use on All Devices</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remote action</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCAST: [&lt;EUI64&gt;,]&lt;length&gt;=&lt;data&gt;</td>
<td>UCAST: [&lt;EUI64&gt;,]&lt;length&gt;=&lt;data&gt;,&lt;RSSI&gt;,&lt;LQI&gt;</td>
</tr>
<tr>
<td>or</td>
<td>Where &lt;EUI64&gt; is the address of the sender and &lt;length&gt; is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.</td>
</tr>
<tr>
<td>SW release</td>
<td>R300</td>
</tr>
</tbody>
</table>

©2015 Telegesis (UK) Ltd - 45 - AT Command Manual (Rev 3.09)
### SYNCR - Transmit Data To The Sink

**Execute Command**

AT\+SCAST:<data>

**Example**

AT\+SCAST:Hello world

**Notes**

- When bit 8 of S10 is set, if a sink cannot be reached for three consecutive transmissions the sink is assumed unavailable and a new one is sought.
- The ACK and/or NACK prompt can be disabled in S0E.
- When attaching the node’s EUI64 to the network frame the maximum payload reduces to 74 bytes.
- The maximum payload is 82 bytes. It is reduced by 8 bytes when appending the EUI to the network header (default) and also it is reduced by 2 bytes per hop in case a source route is known. The latter event can neither be suppressed nor foreseen.
- S-casts can travel up to 30 hops.

**Use on All Devices**

**Response**

SEQ:XX
OK

or **ERROR<errorcode>**

Where <errorcode> represents the error code explained in section 3.

**Prompt**

ACK:XX

or **NACK:XX**

**Parameters**

Up to 82 bytes are sent to the node’s sink. On successful transmission the user is given the sequence number followed by “OK”. After that the user is prompted “ACK” on receipt of an acknowledgement or “NACK” in case the message was not acknowledged. A NACK does not guarantee that the message has not reached its destination.

If bit B of S10 is set, “SEQ”, “ACK” and “NACK” are not reported. “OK” means that the message has been acknowledged by the destination.

**Remote action**

Prompt

UCAST:[<EUI64>,]<length>=<data>

or

UCAST:[<EUI64>,]<length>=<data>,<RSSI>,<LQI>

Where <EUI64> is the address of the sender and <length> is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

**SW release**

R300
### +SCASTB – Transmit Binary Data To A Sink

**Execute Command**  
**AT+SCASTB:XX**

Where XX is the number (in hexadecimal) of data bytes to be sent.

- **Notes**
  - When bit 8 of S10 is set, if a sink cannot be reached for three consecutive transmissions the sink is assumed unavailable and a new one is sought.
  - The ACK and/or NACK prompt can be disabled in S0E.
  - When attaching the node’s EUI64 to the network frame the maximum payload reduces to 74 bytes.
  - The maximum payload is 82 bytes. It is reduced by 8 bytes when appending the EUI to the network header (default) and also it is reduced by 2 bytes per hop in case a source route is known. The latter event can neither be suppressed nor foreseen.
  - S-casts can travel up to 30 hops.

- **Use on All Devices**

- **Response**
  - > `<data being entered>`
  - SEQ:XX
  - OK
  - or ERROR<errorcode>

- **Parameters**
  - **XX ranging from 00 to 52 (hex)**

After the ‘>’ prompt a number of characters are expected to be entered as defined by XX. A maximum of 82 bytes are sent to the network’s sink.

When bit 9 of S10 is set a timeout error is generated if no character is received for 1 second. On successful transmission the user is given a transmission number followed by “OK”. After that the user is prompted “ACK” on receipt of an acknowledgement or “NACK” in case the message was not acknowledged. A NACK does not guarantee that the message has not reached its destination.

If bit B of S10 is set, “SEQ”, “ACK” and “NACK” are not reported. “OK” means that the message has been acknowledged by the destination.

### Remote action

- **Prompt**
  - **UCAST: [<EUI64>,.]XX=<data>**
  - or
  - **UCAST: [<EUI64>,.]<length>=<data>,<RSSI>,<LQI>**

Where <EUI64> is the address of the sender and <length> is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

### SW release

- **R300**
### +SSINK – Search For A Sink

**Execute Command**

```
AT+SSINK
```

Search for a sink on the network by sending a broadcast causing all sinks to reply. By default, if a sink is already known and no better sink is found, no prompt will be displayed. A sink which is already known can be found at index 05 of the address table.

**Response**

OK or ERROR<errorcode>

**Prompt**

SINK:<EUI64>,<Network address> or ADSK:<EUI64>,<Network address>

<errorcode> represents the error code explained in section 3.

**Use on:**

All Devices

**SW release**

R300

### +MCAST – Transmit A Multicast

**Execute Command**

```
AT+MCAST:nn,<ID>,<data>
```

**Notes**

When attaching the node’s EUI64 to the network frame the maximum payload reduces to 74 bytes.

Entries in the multicast table must be set to endpoint 01 to trigger the desired prompt.

Use multicasts sparingly! They are a form of broadcast so any node may only repeat or originate up to 8 multicasts in every 8 second interval.

SEDs and MEDs cannot receive multicast messages.

**Use on:**

All devices

**Response**

OK or ERROR<errorcode>

Where <errorcode> represents the error code explained in section 3.

**Parameters**

**nn ranging from 00 to 30**

Up to 82 bytes are sent to the multicast group <ID>. Instead of a 16-bit multicast ID an 8 bit binding table entry can be specified. The response OK shows successful transmission. Successful transmission does not guarantee successful reception. To make sure data has been received by a specific node use a unicast message. Only neighbours which are up to nn hops away will receive the broadcast. If nn = 01 only direct neighbours will receive the broadcast and if nn = 00 the entire network will (max. 30 hops).

**Remote action**

Prompt

MCAST:[<EUI64>,]<Length>=<data> or MCAST:[<EUI64>,]<length>=<data>,<RSSI>,<LQI>

Where <EUI64> is the address of the sender and <length> is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

**SW release**

R300
### +MCASTB – Transmit A Multicast Of Binary Data

**Execute Command**

```
AT+MCASTB:XX,nn,<ID>
```

Where XX is the number (in hexadecimal) of data bytes to be sent and nn is the number of hops the message will travel.

**Notes**

- When attaching the node’s EUI64 to the network frame the maximum payload reduces to 74 bytes.
- This command is particularly useful if the data may contain `<CR>` and `<Backspace>` characters.
- Use multicasts sparingly! They are a form of broadcast so any node may only repeat or originate up to 8 multicasts in every 8 second interval.
- SEDs and MEDs cannot receive multicast messages.
- Use on All Devices

**Response**

> `<data being entered>`

OK

or `ERROR<errorcode>`

After the ‘>’ prompt a number of characters are expected to be entered as defined by XX. `<errorcode>` represents the error code explained in section 3.

When bit 9 of S10 is set a timeout error is generated if no character is received for 1 second.

**Parameters**

- **XX** ranging from 00 to 52 (hex)
- **nn** ranging from 00 to 30

Up to 82 bytes are sent to devices up to nn hops away. The response OK shows successful transmission. Successful transmission does not guarantee successful reception. To make sure data has been received by a specific node use a unicast message. Only neighbours which are up to nn hops away will receive the broadcast. If nn=01 only direct neighbours will receive the broadcast and if n = 00 the entire network will.

**Remote action**

**Prompt**

```
MCAST: [<EUI64>,]<length>=<data>
or
MCAST: [<EUI64>,]<length>=<data>, <RSSI>,<LQI>
```

Where `<EUI64>` is the address of the sender and `<length>` is the length of the message in hexadecimal. The EUI64 is only displayed if it is part of the network header (set bit 0 of S10 to disable attaching the EUI64 to outgoing messages). RSSI and LQI are shown if bit 6 of S0F is set.

**SW release**

R300
## +DMODE – Enter Data Mode (Serial Link Mode)

**Execute Command**

```
AT+DMODE:<address>
```

Where `<address>` can be the remote node’s EUI64, Network address or address table index.

**Note**

Opening a serial link to end devices will result in a limited data rate which depends on the polling interval of the child. In Data mode all prompts are disabled.

**Use on**

All Devices

**Response**

```
SEQ:XX
OK
```

or

```
ERROR<errorcode>
```

**Prompt**

```
ACK:XX
```

or

```
NACK:XX
```

<errorcode> represents the error code explained in section 3 and XX is the sequence number of the unicast.

**Remote Prompt**

```
DataMODE:<Network address>,<EUI64>
```

**OPEN**

Where `<Network address>` is the Network address of the remote node and `<EUI64>` is its EUI64.

**Prompt**

```
DataMODE:<Network address>,<EUI64>,<errorcode>
```

[OPEN]

Where `<Network address>` is the Network address of the remote node and `<EUI64>` is its EUI64. Only if the errorcode equals 0 the data mode will open.

**SW release**

R302

## +++ – Leave Data Mode

**Execute Command**

```
+++ 
```

To leave data mode `+++` must be entered at a minimum of 500ms after the last character which is to be transmitted to the remote node. In case the data payload contains `+++` it can be transmitted safely as long as it is made sure no more than 250ms pass between sending `+++` and the previous character.

**Use on**

All Devices

**Response**

```
CLOSED
```

**SW release**

R302
### +IDENT — Play A Tune On Remote Devboard

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+IDENT:&lt;address&gt;</td>
<td>SEQ:XX OK</td>
</tr>
<tr>
<td></td>
<td>or ERROR&lt;errorcode&gt;</td>
</tr>
</tbody>
</table>

Use on All Devices

#### Prompt

- ACK:XX
- NACK:XX

<errorcode> represents the error code explained in section 3.
Plays a tune on a remote devboard if the Beeper is connected. Useful to identify remote nodes. See devkit manual for details about connecting a buzzer to the ETRXn.

#### SW release

- R300
**+SENDUCAST – Send A Raw ZCL/ZDO Unicast**

**Execute Command**

`AT+SENDUCAST:<Address>,<SourceEP>,<DestEP>,<ProfileID>,<ClusterID>,<data>`

- **<Address>** - can be the remote node’s EUI64, Network address, address table index or binding table index.

- **<SourceEP>** - 8-bit hexadecimal number, specifying the source endpoint. For unicasts to binding table entries, the source endpoint is taken from register S40.

- **<DestEP>** - 8-bit hexadecimal number, specifying the destination endpoint. For unicasts to binding table entries dummy 8-bit numbers need to be specified for both endpoints, which will be overwritten with the information from the binding table.

**Example**

`AT+SENDUCAST:0000,01,01,C091,0002,Test`  
Sends a Unicast to Coordinator

**Notes**

As a unicast command, AT+SENDUCAST is subject to the same limits of payload length and simultaneous messages as AT+UCAST.

SEQ and ACK are only reported when cluster 0002 is used.

**Use on All Devices**

**Response**

SEQ:XX  
OK  

or  
ERROR:<errorcode>

Where <errorcode> represents the error code explained in section 3.

**Prompt**

ACK:XX  

or NACK:XX

If bit B of S10 is set, “SEQ”, “ACK” and “NACK” are not reported. “OK” means that the message has been acknowledged by the destination.

Please check ZigBee Cluster Library and HA Profile for more information about constructing a raw command.
**AT+SENDUCASTB – Send A Raw Binary ZCL/ZDO Unicast**

**Execute Command**

```
AT+SENDUCASTB:<Length>,<Address>,<SourceEP>,<DestEP>,<ProfileID>,<ClusterID>
```

- **<Length>** - 8-bit hexadecimal number indicating the length of the message
- **<Address>** - can be the remote node’s EUI64, Network address, address table index or binding table index.
- **<SourceEP>** - 8-bit hexadecimal number, specifying the source endpoint. For unicasts to binding table entries, the source endpoint is taken from register S40.
- **<DestEP>** - 8-bit hexadecimal number, specifying the destination endpoint. For unicasts to binding table entries dummy 8-bit numbers need to be specified for both endpoints, which will be overwritten with the information from the binding table.

**Example**

```
AT+SENDUCASTB:05,0000,01,01,C091,0002
```

> HELLO

Sends a Unicast to Coordinator

**Note**

As a unicast command, AT+SENDUCASTB is subject to the same limits of payload length and simultaneous messages as AT+UCASTB.

SEQ and ACK are only reported when cluster 0002 is used.

**Use on**

All Devices

<table>
<thead>
<tr>
<th>SW release</th>
<th>R309</th>
</tr>
</thead>
</table>

**Response**

```
> <data being entered>
SEQ:XX
```

**OK**

or **ERROR<errorcode>**

After the ‘>’ prompt a number of characters are expected to be entered as defined by <Length>. <errorcode> represents the error code explained in section 3.

When bit 9 of S10 is set a timeout error is generated if no character is received for 1 second

**Prompt**

**ACK:XX**

or **NACK:XX**

If bit B of S10 is set, “SEQ”, “ACK” and “NACK” are not reported. “OK” means that the message has been acknowledged by the destination.

Please check ZigBee Cluster Library and HA Profile for more information about constructing a raw command.
### +SENDMCAST – Send A Raw ZCL/ZDO Multicast or Broadcast

**Execute Command**

\[ \text{AT+SENDMCAST:} \langle \text{Radius}\rangle, \langle \text{Address}\rangle, \langle \text{SourceEP}\rangle, [\langle \text{DestEP}\rangle], \langle \text{ProfileID}\rangle, \langle \text{ClusterID}\rangle, \langle \text{data}\rangle \]

- **<Radius>** - 2-digit decimal number specifying the maximum number of hops over which the message can pass. Range 00 to 30.
- **<Address>** - can be the remote node’s group address or a broadcast address.
- **<SourceEP>** - 8-bit hexadecimal number, specifying the source endpoint.
- **<DestEP>** - 8-bit hexadecimal number, specifying the destination endpoint.

If DestEP is left empty, the <Address> field set from 0x0000 to 0xFFF7 will be recognized as a group ID for a multicast.

If DestEP is used 0xFFFF will be recognized as a broadcast to all routers, 0xFFFD as a broadcast to all non-sleepy devices and 0xFFFF as a broadcast to all devices including sleepy end devices.

Note SEDs and MEDs cannot receive multicast messages

**Examples**

\[ \text{AT+SENDMCAST:01,1234,01,,C091,0002,Test} \]

Sends a Multicast to group 1234 over one hop

\[ \text{AT+SENDMCAST:00,FFFF,01,01,C091,0002,Test} \]

Sends a Broadcast to all Devices over 30 hops

Note SEDs and MEDs cannot receive multicast messages

As a broadcast command, AT+SENDMCAST is subject to the same limit of message rate as AT+MCAST.

**Use on**

All Devices

**Response**

OK

or ERROR:<errorcode>

Where <errorcode> represents the error code explained in section 3.

Please check ZigBee Cluster Library and HA Profile for more information about constructing a raw command

| SW release | R309 |
### AT+SENDMCASTB – Send A Raw Binary ZCL/ZDO Multicast or Broadcast

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+SENDMCASTB:&lt;Length&gt;,&lt;Radius&gt;,&lt;Address&gt;,&lt;SourceEP&gt;,[&lt;DestEP&gt;],&lt;ProfileID&gt;,&lt;ClusterID&gt;</td>
<td>OK</td>
</tr>
</tbody>
</table>

**<Radius>** - 2-digit decimal number specifying the maximum number of hops over which the message can pass. Range 00 to 30.

**<Length>** - 8-bit hexadecimal number indicating the length of the message

**<Address>** - can be the remote node’s group address or a broadcast address.

**<SourceEP>** - 8-bit hexadecimal number, specifying the source endpoint.

**<DestEP>** - 8-bit hexadecimal number, specifying the destination endpoint.

If DestEP is left empty, the <Address> field set from 0x0000 to 0xFFF7 will be recognized as a group ID for a multicast.

If DestEP is used 0xFFFC will be recognized as a broadcast to all routers, 0xFFFFD as a broadcast to all non-sleepy devices and 0xFFFF as a broadcast to all devices including sleepy end devices.

**Examples**

**AT+SENDMCASTB:05,01,1234,01,,C091,0002**

> HELLO
Sends a Multicast to group 1234 over one hop

**AT+SENDMCASTB:05,00,FFFF,01,01,C091,0002**

> HELLO
Sends a Broadcast to all Devices over 30 hops

**Notes**

SEDs and MEDs cannot receive multicast messages.

As a broadcast command, AT+SENDMCAST is subject to the same limit of message rate as AT+MCAST.

**Use on**

All Devices

**Response**

> <data being entered>
OK

or **ERROR<errorcode>**

After the ‘>’ prompt a number of characters are expected to be entered as defined by <Length>. <errorcode> represents the error code explained in section 3.

In case bit 9 of S10 is set a timeout error is generated if no character is received for 1 second.

Please check ZigBee Cluster Library and HA Profile for more information about constructing a raw command.
### +INTERPAN – Send an Interpan Command

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+INTERPAN:&lt;&lt;AddressMode&gt;,&lt;DstAddress&gt;,&lt;DstPAN&gt;,&lt;ProfileID&gt;,&lt;ClusterID&gt;,&lt;Payload&gt;</td>
<td>OK or ERROR:&lt;errorcode&gt; (see section 3).</td>
</tr>
</tbody>
</table>

**<AddressMode>** - 8 bit hexadecimal number. The user shall use this parameter to specify which type of destination address is used:
- 00 - Node ID
- 01 - Group ID
- 02 - EUI address

**<DstAddress>** - 16 bit hexadecimal number if AddressMode is Node ID or Group ID or EUI address, if AddressMode is long destination address.

**<DstPAN>** - 16 bit hexadecimal number representing destination PAN ID.

**<ProfileID>** - 16 bit hexadecimal number representing profile ID. e.g. 0x0104 for Home automation, 0xC091 for Telegesis profile.

**<ClusterID>** - 16 bit hex number representing Cluster ID.

**<Payload>** - Command payload, formatted as ASCII hex data.

**Notes**
Interpan messages cannot be sent to SEDs or MEDs.
They can only travel one hop.
They are not encrypted or acknowledged.
Source and destination devices must use the same radio channel.

**Example**
AT+INTERPAN:00,0000,1234,C091,0002,4142434445

Use on All Devices

**SW release**
R309
**AT+RDATAB – Send Binary Raw Data**

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AT+RDATAB:XX</strong></td>
<td>&gt; &lt;data being entered&gt; OK or ERROR:&lt;errorcode&gt;</td>
</tr>
</tbody>
</table>

**Use on All Devices**

**Notes**

Can be useful to quickly exchange bulk data with neighbouring node. The application needs to handle addressing, error checking, retries and acknowledgements.

End Devices do not receive raw data. Raw data will only travel one hop.

Use with great care. Raw data messages are not ZigBee-compliant and may even leak into other PANs.

**Parameters**

**XX ranging from 00 to 67 (hex)**

After the ‘>’ prompt a number of XX characters are expected to be entered. Up to 103 bytes of data can be send to all nodes within reach (direct neighbours).

The data is neither encrypted nor error checked. No retries are made and no acknowledgement is received.

<errorcode> represents the error code explained in section 3.

**Remote action**

**Prompt**

RAW:snn,<data>

where snn is the RSSI, or

<data>

in case bit 9 of S0E is set. Displaying the data can also be disabled by setting bit D of S0E.

**SW release**

R300
2.9 Binding Management (ETRX3 Series only)

**+LBTABLE – Display Local Binding Table**

**Read Command**

AT+LBTABLE

**Use on**

All Devices

Entries in the local Binding Table.

**Response**

<table>
<thead>
<tr>
<th>No.</th>
<th>Type</th>
<th>Active</th>
<th>LocalEP</th>
<th>ClusterID</th>
<th>Addr</th>
<th>RemEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Ucast</td>
<td>No</td>
<td>01</td>
<td>DEAD</td>
<td>1234567887654321</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>MTO</td>
<td>No</td>
<td>01</td>
<td>DEAD</td>
<td>E012345678876543</td>
<td>88</td>
</tr>
<tr>
<td>12</td>
<td>Mcast</td>
<td>No</td>
<td>01</td>
<td>DEAD</td>
<td>CDAB</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SW release

R307

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

**Write Command**

AT+BSET:<type>,<LocalEP>,<ClusterID>,<DstAddress>[,<DstEP>]

**Where**

- `<Type>` is the type of binding as shown below.
- `<LocalEP>` is the local endpoint.
- `<ClusterID>` is the cluster ID.
- `<DstAddress>` is either the EUI64 of the target device, or a multicast ID.
- `<DstEP>` is the destination endpoint which is not specified in case of a multicast binding.

The new binding is created in the next available free binding table entry.

**Types:**

1. **Unicast Binding with EUI64 and destination EP specified**
2. **Many to one Binding with EUI64 and destination EP Specified**
3. **Multicast Binding with Multicast ID Specified**

**Example**

AT+BSET:1,01,0002,000d6f000059474e,01

**Note**

All parameters must have exactly the correct number of characters.
Use mode 2 when the source or destination is a coordinator or sink.

**Use on**

All Devices

SW release

R307
## +BCLR – Clear local Binding Table Entry

**Write Command**

`AT+BCLR:XX`

Where `XX` is the entry number of the binding table entry which is to be cleared. To keep the numbering of the local binding table in-line with the numbering of the remote binding table all remaining entries are moved to the beginning of the table.

`AT+BCLR:FF` clears the whole table.

**Use on**

All Devices

**Response**

OK or `ERROR:<errorcode>`

<errorcode> represents the error code explained in section 3.

---

## +BTABLE – Display Binding Table

**Read Command**

`AT+BTABLE:XX,<address>`

Where `XX` is the start index of the remote Binding table and `<address>` can be the remote node's EUI64, Network address or address/binding table entry.

**Note**

Also the local node can be the target of this command (e.g. use address table entry FF as the address)

**Example**

`AT+BTABLE:00,0000`

SEQ:01

OK

**Use on**

All devices

**Response**

SEQ:XX

OK or `ERROR:<errorcode>`

This command requests the target node to respond by listing its binding table starting from the requested index. 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.

**Example**

`BTABLE:0000,00`

Length:03

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

ACK:01

In this example the neighbour table of the remote node with the short ID shown in `<Network address>` 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.
### +BIND – Create Binding on Remote Device

**Write Command**

```
AT+BIND:<address>,<type>,<SrcAddress>,<SrcEP>,<ClusterID>,<DstAddress>[,<DstEP>]
```

Create Binding on a remote device with
- `<address>` the target Node’s EUI64, Network address, or Address/Binding Table entry
- `<type>` the Addressing mode as shown below
- `<SrcAddress>` The EUI64 of the Source
- `<SrcEP>` The source Endpoint
- `<ClusterID>` The Cluster ID on the source Device
- `<DstAddress>` The EUI64 or 16-bit multicast ID, depending on `<type>`
- `<DstEP>` Only in Mode 3: The destination endpoint

**Types:**
1. Multicast Binding with Multicast ID Specified in `<DstAddress>`
2. Unicast Binding with destination EUI64 in `<DstAddress>` and destination EP in `<DstEP>`

**Example**
```
AT+BIND:0000,3,000d6f000059474e,01,abcd,000D6F0000123456,01
```

**Notes**

“Source” and “destination” are defined from the viewpoint of the remote device

The local node can also be the target of this command (e.g. use address table entry FF as the address)

All parameters must have exactly the correct number of characters

**Use on**

All devices

---

**Response**

<table>
<thead>
<tr>
<th>SEQ:XX</th>
<th>OK or ERROR:&lt;errorcode&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

**Prompt**

```
Bind:<network address>,<status>
```

In case of an error an status other than 00 will be displayed
- `<Network address>` is the Remote node’s Network address.
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.

**Example**
```
SEQ:01
OK
```

```
Bind:0000,00
ACK:01
```

**SW release**

R307
+UNBIND – Delete Binding on Remote Device

Write Command

AT+UNBIND:<address>,<type>,
<SrcAddress>,<SrcEP>,<ClusterID>,
<DstAddress>[,<DstEP>]

Delete Binding on a remote device with
<address> the target Node’s EUI64, Network address, or Address/Binding Table entry
$type> the Addressing mode as shown below
<SrcAddress> The EUI64 of the Source
<SrcEP> The source Endpoint
<ClusterID> The Cluster ID on the source Device
<DstAddress> The EUI64 or 16-bit multicast ID, depending on $type$
<DstEP> Only in Mode 3: The destination endpoint

Types:
1= Multicast Binding with Multicast ID Specified in <DstAddress>
3= Unicast Binding with destination EUI64 in <DstAddress> and destination EP in <DstEP>

Note
Also the local node can be the target of this command (e.g. use address table entry FF as the address)

Example

AT+UNBIND:0000,3,000d6f000059474e,01,
abcd,000D6F0000123456,01

Note
All parameters must have exactly the correct number of characters

Use on
All devices

Response
SEQ:XX
OK or ERROR:<errorcode>
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.

Prompt
Unbind:<network address>,<status>

In case of an error an status other than 00 will be displayed
<network address> is the Remote node’s Network address.
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.

Example

SEQ:01
OK
Unbind:0000,00
ACK:01
### +EDBIND – Request End Device Binding

<table>
<thead>
<tr>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+EDBIND:[&lt;target&gt;,]&lt;SrcEP&gt;,&lt;ProfileID&gt;,&lt;NumInClusters&gt;,&lt;InClusterList&gt;,&lt;NumOutClusters&gt;,&lt;OutClusterList&gt;</td>
<td>SEQ:XX OK or ERROR:&lt;errorcode&gt;</td>
</tr>
</tbody>
</table>

Request and end device binding

- `<target>`: the Network address of the local device, or the network’s primary binding cache device. If omitted the Network address of the local device is used.
- `<SrcEP>`: The source Endpoint
- `<ProfileID>`: The Profile ID which is to be matched
- `<NumInClusters>`: The number of clusters provided in the following list
- `<InClusterList>`: List of 16-bit cluster IDs all separated by a comma
- `<NumOutClusters>`: The number of clusters provided in the following list
- `<OutClusterList>`: List of 16-bit cluster IDs all separated by a comma

#### Examples

- **AT+EDBIND:123A,01,C091,01,abcd,02,1234,5678**
  - No input clusters
- **AT+EDBIND:123A,01,C091,00,,02,1234,5678**
  - No input clusters
- **AT+EDBIND:123A,01,C091,01,abcd,00,**
  - No output clusters (note final comma)

All parameters must have exactly the correct number of characters.

#### Use on
All devices

#### SW release
R309

---

*©2015 Telegesis (UK) Ltd

AT Command Manual (Rev 3.09)*
## 2.10 Time-related commands

### +SETTIME - Set The Local Time

**Execute Command**

AT+SETTIME: <year>, <month>, <day>, <hour>, <min>, <sec>

or

AT+SETTIME: <time>

- `<year>` - 4 digits decimal number
- `<month>` - 2 digits decimal number
- `<day>` - 2 digits decimal number
- `<hour>` - 2 digits decimal number
- `<min>` - 2 digits decimal number
- `<sec>` - 2 digits decimal number

- `<time>` - 32 bit hexadecimal number representing time in UTC format (number of seconds since 01.01.2000 00:00)

**Note**

The earliest time that can be set using the first format is 2008,01,01,00,00,00

**Example**

AT+SETTIME: 2009,03,05,08,15,00
(2009, March 5th 8:15 AM)

**Response**

OK
or ERROR: <errorcode>

<errorcode> represents the error code explained in section 3.

---

### +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

<errorcode> represents the error code explained in section 3.
### +SYNCTIME - Synchronize the Local Time with Time Server

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+SYNCTIME:&lt;Node ID&gt;, &lt;End Point&gt;[,Profile ID]</td>
<td>OK or ERROR:&lt;errorcode&gt; followed by Prompt: SYNCINGTIME:&lt;time&gt;</td>
</tr>
</tbody>
</table>

**<Node ID>:** Target node address  
**<End Point>:** Target node’s end point  
**<Profile ID>:** Profile ID used for the request, if unspecified 0x0104 (HA) will be used

**Note:**  
The target shall support the time server cluster. To exchange this message without interfering with any of the other message types, the local endpoint 0x63 (99) is used for this exchange.

**<errorcode> represents the error code explained in section 3**
## 3 List of Error codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Everything OK - Success</td>
</tr>
<tr>
<td>01</td>
<td>Couldn't poll Parent because of Timeout</td>
</tr>
<tr>
<td>02</td>
<td>Unknown command</td>
</tr>
<tr>
<td>04</td>
<td>Invalid S-Register</td>
</tr>
<tr>
<td>05</td>
<td>Invalid parameter</td>
</tr>
<tr>
<td>06</td>
<td>Recipient could not be reached</td>
</tr>
<tr>
<td>07</td>
<td>Message was not acknowledged</td>
</tr>
<tr>
<td>08</td>
<td>No sink known</td>
</tr>
<tr>
<td>09</td>
<td>Address Table entry is in use and cannot be modified</td>
</tr>
<tr>
<td>0A</td>
<td>Message could not be sent</td>
</tr>
<tr>
<td>0B</td>
<td>Local node is not sink</td>
</tr>
<tr>
<td>0C</td>
<td>Too many characters</td>
</tr>
<tr>
<td>0E</td>
<td>Background Scan in Progress (Please wait and try again)</td>
</tr>
<tr>
<td>0F</td>
<td>Fatal error initialising the network</td>
</tr>
<tr>
<td>10</td>
<td>Error bootloading</td>
</tr>
<tr>
<td>12</td>
<td>Fatal error initialising the stack</td>
</tr>
<tr>
<td>18</td>
<td>Node has run out of Buffers</td>
</tr>
<tr>
<td>19</td>
<td>Trying to write read-only register</td>
</tr>
<tr>
<td>1A</td>
<td>Data Mode Refused by Remote Node</td>
</tr>
<tr>
<td>1B</td>
<td>Connection Lost in Data Mode</td>
</tr>
<tr>
<td>1C</td>
<td>Remote node is already in Data Mode</td>
</tr>
<tr>
<td>20</td>
<td>Invalid password</td>
</tr>
<tr>
<td>25</td>
<td>Cannot form network</td>
</tr>
<tr>
<td>27</td>
<td>No network found</td>
</tr>
<tr>
<td>28</td>
<td>Operation cannot be completed if node is part of a PAN</td>
</tr>
<tr>
<td>2C</td>
<td>Error leaving the PAN</td>
</tr>
<tr>
<td>2D</td>
<td>Error scanning for PANs</td>
</tr>
<tr>
<td>33</td>
<td>No response from the remote bootloader</td>
</tr>
<tr>
<td>34</td>
<td>Target did not respond during cloning</td>
</tr>
<tr>
<td>35</td>
<td>Timeout occurred during xCASTB</td>
</tr>
<tr>
<td>39</td>
<td>MAC Transmit Queue is Full</td>
</tr>
<tr>
<td>6C</td>
<td>Invalid Binding Index</td>
</tr>
<tr>
<td>70</td>
<td>Invalid Operation</td>
</tr>
<tr>
<td>72</td>
<td>More than 10 unicast messages were in flight at the same time</td>
</tr>
<tr>
<td>74</td>
<td>Message too long</td>
</tr>
<tr>
<td>80</td>
<td>ZDP Invalid Request Type</td>
</tr>
<tr>
<td>81</td>
<td>ZDP Device not Found</td>
</tr>
<tr>
<td>82</td>
<td>ZDP Invalid Endpoint</td>
</tr>
<tr>
<td>83</td>
<td>ZDP Not Active</td>
</tr>
<tr>
<td>84</td>
<td>ZDP Not Supported</td>
</tr>
<tr>
<td>85</td>
<td>ZDP Timeout</td>
</tr>
<tr>
<td>86</td>
<td>ZDP No Match</td>
</tr>
<tr>
<td>87</td>
<td>ZDP Table Full</td>
</tr>
<tr>
<td>88</td>
<td>ZDP No Entry</td>
</tr>
<tr>
<td>89</td>
<td>ZDP No Descriptor</td>
</tr>
</tbody>
</table>
91  Operation only possible if connected 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
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
C5  NWK Already Present
C7  NWK Table Full
C8  NWK Unknown Device

Bootloader error codes

18  Transfer aborted prematurely
1B  Start of data transfer timed out
1C  Data transfer timed out
44  Unknown tag detected in .EBL image (wrong file format?)
45  Invalid .EBL header signature (wrong file type for chip?)
4E  An invalid length was detected in the .EBL image (corrupt file?)
4 S-Registers

Most S-Registers of the ETRX357 can be read and written locally as well as remotely. The S-Registers are summarised in the table below.

<table>
<thead>
<tr>
<th>S-Register Overview</th>
<th>Local R/W</th>
<th>Remote R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>S00 Channel Mask</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S01 Transmit Power Level</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S02 Preferred PAN ID</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S03 Preferred Extended PAN ID</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S04 Local EUI</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S05 Local Network address</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S06 Parent’s EUI</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S07 Parent’s Network address</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S08 Network Key¹</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S09 Link Key¹</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0A Main Function¹</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0B User Readable Name¹</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0C Password¹</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0D Device Information</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0E Prompt Enable 1</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S0F Prompt Enable 2</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S10 Extended Function</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S11 Device Specific</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S12 UART Setup</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S13 Pull-up enable</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S14 Pull-down enable</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S15 I/O Configuration</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S16 Data Direction of I/O Port (volatile)</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S17 Initial Value of S16</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S18 Output Buffer of I/O Port (volatile)</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S19 Initial Value of S18</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1A Input Buffer of I/O Port (volatile)</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1B Special Function pin 1 (volatile)</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1C Initial Value of S1B</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1D Special Function Pin 2 (volatile)</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1E Initial Value of S1D</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S1F ADC0</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S20 ADC1</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S21 ADC2</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S22 ADC3</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S23 Immediate functionality at IRQ0</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S24 Immediate functionality at IRQ1</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S25 Immediate functionality at IRQ2</td>
<td>(••)</td>
<td>(••)</td>
</tr>
<tr>
<td>S26 Immediate functionality at IRQ3</td>
<td>(••)</td>
<td>(••)</td>
</tr>
</tbody>
</table>

¹ Password Protected Registers
<table>
<thead>
<tr>
<th>S-Register Overview (continued)</th>
<th>Local R/W</th>
<th>Remote R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>S27 Functionality 1 at Boot-up</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S28 Functionality at Network Join</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S29 Timer/Counter 0</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2A Functionality for Timer/Counter 0</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2B Timer/Counter 1</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2C Functionality for Timer/Counter 1</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2D Timer/Counter 2</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2E Functionality for Timer/Counter 2</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S2F Timer/Counter 3</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S30 Functionality for Timer/Counter 3</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S31 Timer/Counter 4</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S32 Functionality for Timer/Counter 4</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S33 Timer/Counter 5</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S34 Functionality for Timer/Counter 5</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S35 Timer/Counter 6</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S36 Functionality for Timer/Counter 6</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S37 Timer/Counter 7</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S38 Functionality for Timer/Counter 7</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S39 Power mode (volatile)</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S3A Initial Power Mode</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S3B Start-up Functionality Plaintext A</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S3C Start-up Functionality Plaintext B</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S3D Supply Voltage</td>
<td>(**)</td>
<td>(**)</td>
</tr>
<tr>
<td>S3E Multicast Table Entry 00</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S3F Multicast Table Entry 01</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S40 Source and Destination Endpoints for xCASTs (volatile)</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S41 Initial Value of S40</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S42 Cluster ID for xCASTs (volatile)</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S43 Initial Value of S42</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S44 Profile ID for xCASTs (volatile)</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S45 Initial Value of S44</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S46 Start-up Functionality 32 bit number (volatile)</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S47 Power Descriptor</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S48 Endpoint 2 Profile ID</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S49 Endpoint 2 Device ID</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4A Endpoint 2 Device Version</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4B Endpoint 2 Input Cluster List</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4C Endpoint 2 Output Cluster List</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4D Mobile End Device Poll Timeout</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4E End Device Poll Timeout</td>
<td>(*)</td>
<td>(*)</td>
</tr>
<tr>
<td>S4F MAC Timeout</td>
<td>(*)</td>
<td>(*)</td>
</tr>
</tbody>
</table>

Table 6: S-Register Overview
With a few exceptions the 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. S16, S18, S1A, S1B, S1D, S39, S40 and S42 are directly accessing volatile I/O registers to prevent memory corruption due to constant I/O access. Registers S17, S19, S1C, S1E, S3A, S41 and S43 represent the non-volatile registers which define the contents of S16, S18, S1B, S1D, S39, S40 and S42 respectively after booting up or reset.

4.1 Recovery of the Factory Default Settings

If the unit seems to be unresponsive to commands on the serial port this is most often due to the unit having been set into a power-down mode or the set-up for the serial connection having been altered. To overcome this a feature has been added which performs a factory reset on any module which seems unresponsive. To factory reset a module, connect it to the PC’s serial port and execute the Factory Reset Tool (downloadable from www.telegesis.com). When pressing the Reset button on the Reset Tool you are prompted to cause a hardware reset to the module by pulling the module’s reset line low for more than 100ms (done by pressing the reset button on the Development Board). Once completed, the factory default settings of the ETRX357 module are restored.
### 4.2 S-Registers for Network Setup

#### S00 – Channel Mask

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The 802.15.4 channel mask.</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Becomes effective</th>
<th>Note</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
<td>When Joining, Scanning or establishing a PAN</td>
<td>The channel mask does not affect the AT+JPAN command</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>Factory Default</th>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001 - FFFF</td>
<td>ETRX3 LRS-Variants: 7FF</td>
<td>R302</td>
</tr>
<tr>
<td>Others: FFFF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### S01 – Transmit Power Level

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The device’s transmit power level in dBm.</td>
<td>snn</td>
</tr>
</tbody>
</table>

Where snn represents a signed 8-bit decimal number.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Notes</th>
<th>Becomes effective</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
<td>The output power of the “-LRS” variant is higher than the value in S01. Please refer to the respective hardware manuals.</td>
<td>When Joining or establishing a PAN</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td>The ETRX357-LRS power is reduced for EC regulatory compliance. See the hardware manual.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>Factory Default</th>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETRX3: 8 to -43</td>
<td>ETRX3 LRS Variants: -7 to -43</td>
<td>R305</td>
</tr>
<tr>
<td>Others: 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Entering a value higher than 3 will automatically enable boost mode regardless of the setting of bit E of S11.

**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.**

©2015 Telegesis (UK) Ltd
## S02 – Preferred PAN ID

**Description**
The 802.15.4 PAN ID.

**Operations**
- R/W LOCAL
- R/W REMOTE

**Becomes effective**
**When Joining or establishing a PAN**

**Notes**
Two networks operating on the same channel with the same PAN ID, but a different EPID 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. The preferred PID does not affect the AT+JPAN command.

**Storage**
**Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>&lt;PID&gt;</th>
</tr>
</thead>
</table>

Where <PID> represents a 16-bit hexadecimal number

<table>
<thead>
<tr>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 – FFFF</td>
</tr>
</tbody>
</table>

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.

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.

**Factory Default**
0000

---

## S03 – Preferred Extended PAN ID

**Description**
The extended PAN ID.

**Operations**
- R/W LOCAL
- R/W REMOTE

**Becomes effective**
**When Joining or establishing a PAN**

**Note**
The EPID is used for PAN ID conflict detection. It is therefore recommended to use a random EPID at all times. The preferred EPIP does not affect the AT+JPAN command.

**Storage**
**Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>&lt;EPID&gt;</th>
</tr>
</thead>
</table>

Where <EPID> represents a 64-bit hexadecimal number

<table>
<thead>
<tr>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000000000000 – FFFFFFFFFFFFFFFF</td>
</tr>
</tbody>
</table>

When establishing a PAN the coordinator will pick a random EPID if S03 is set to all 0’s. If set to any other value this number will be used as EPID instead.

When joining only a PAN with the EPID stored in S03 will be joined unless S03 is set to all 0’s. In this case the next best PAN which allows joining is joined.

**Factory Default**
0000000000000000
### S04 – Local EUI64

**Description**
The local node's unique EUI64 identifier.

**Parameters**
- `<EUI64>`

**Operations**
- R LOCAL
- R REMOTE

**Storage**
- Non-Volatile

**Range**
0000000000000000 – FFFFFFFFDFF

**Factory Default**
- `<unique number>`

**SW release**
- R300

**Note**
Reading this register while not associated with a network will result in an undefined return value.

---

### S05 – Local 16-Bit Network address

**Description**
The local node's 16-bit Network address.

**Parameters**
- `<Network address>`

**Operations**
- R LOCAL
- R REMOTE

**Storage**
- Non-Volatile

**Range**
0000-FFF

**Factory Default**
- n/a

**SW release**
- R300

**Note**
Reading this register while not associated with a network will result in an undefined return value.

---

### S06 – Parent's EUI64

**Description**
The parent node's unique EUI64 identifier.

**Parameters**
- `<EUI64>`

**Operations**
- R LOCAL
- R REMOTE

**Storage**
- Non-Volatile

**Range**
0000000000000000 – FFFFFFFFDFF

**Factory Default**
- n/a

**SW release**
- R300

**Note**
The return value is undefined for nodes without parents (coordinators and nodes that are not joined to a network). For an FFD, S06 is the ID of the node via which the local node joined the PAN.
### S07 – Parent’s 16-Bit Network address

**Description**  
The parent node’s 16-bit Network address.

**Parameters**  
\(<\text{Network address}>\)

**Range**  
0000-FFFF

**Operations**  
R LOCAL  
R REMOTE

**Note**  
The return value is undefined for nodes without parents (coordinators and nodes that are not joined to a network). For an FFD, S07 is the ID of the node via which the local node joined the PAN.

**Storage**  
Non-Volatile

**SW release**  
R300

---

### S08 – Network Key

**Description**  
The network key which can be written using the password. The default password for R3xx is "password".

**Operations**  
W LOCAL  
W REMOTE

**Write operation**  
ATS08=<key>:<password>  
ATREMS:<address>,08=<key>:<password>

**Becomes effective**  
Only when establishing a PAN

**Storage**  
Non Volatile

**Range**  
From 0 to $2^{128}-1$

The 128-bit AES network key in hexadecimal representation (32 characters).

When set to all 0's (default) a random network key is generated when establishing a PAN.

This key is transmitted to all joining nodes and can be encrypted using the link key.

**Factory Default**  
00000000000000000000000000000000
## S09 – Trust Centre Link Key

**Description**
The link key which can be written using the password. The default password for R3xx is "password".

**Operations**
- **W LOCAL**
- **W REMOTE**

**Write operation**
- `ATS09=<key>:<password>`
- `ATREMS:<address>,09=<key>:<password>`

**When Joining or establishing a PAN**

**Storage**
- **Non Volatile**

<table>
<thead>
<tr>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>The link key which can be written using the password. The default password for R3xx is &quot;password&quot;.</td>
<td><strong>From 0 to (2^{128}-1)</strong></td>
</tr>
</tbody>
</table>

The 128-bit trust centre link key in hexadecimal representation (32 characters).

When set to all 0s (default) a random trust centre link key is generated when establishing a PAN.

**Factory Default**

```
00000000000000000000000000000000
```

**SW release**
- **R300**

### 4.3 S-Registers for Module Setup

#### S0A – Main Function

**Description**
Defines the behaviour of the Device.

**Operations**

- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**

- **When joining or establishing a PAN (bits F-D)**
- **When PWM is next enabled (bit C)**
- **Instantly (bits B-0)**

**Write operation**

- **ATS0A=XXXX:<Password>**
- **ATREMS:<address>,0A=XXXX:<Password>**

**Notes**
For security reasons this register is password protected. The default password for R3xx is “password”.

See section 12 regarding secure networks

To block joining, set either bit 5 on the trust centre or bit 0 on every node. Built-in function 0017 only overrides bit 0

An End Device (not Sleepy or Mobile) is primarily to let a ZigBee PRO device join a ZigBee 2006 PAN. An SED or MED is the more usual choice

**Storage**
Non-Volatile

<table>
<thead>
<tr>
<th>Bit F</th>
<th>Bit E</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Router (FFD)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>End Device</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Sleepy End Device</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Mobile End Device</td>
</tr>
</tbody>
</table>

**Bit E-F: Device Selection**

**Bit D:** Set: If a router, do not route any messages

**Bit C:** Prescale PWM clock to reduce frequency by 256

**Bit B:** Set: Allows Endpoint 2 to reply to ZDO endpoint queries

**Bit A:** Set: When joining don’t ask for Trust Centre link key

**Bit 9:** Set: Don’t use central Trust Centre (distributed TC Mode)

**Bit 8:** Set: Use Pre-Configured Trust Centre Link Key when joining

**Bit 7:** Set: Trust centre uses hashed link key

**Bit 6:** Set: Append RSSI and LQI to all RX: prompts

**Bit 5:** Set: Don’t allow nodes to join (TC setting)

**Bit 4:** Set: Send Network key encrypted with the link key to nodes joining

**Bit 3:** Set: Do not allow nodes to re-join unsecured

**Bit 2:** Set: Send Network key encrypted with the link key to nodes re-joining unsecured

**Bit 1:** Set: Disable received interpan messages

**Bit 0:** Set: Do not allow other nodes to join the network via this node

**Factory Default**

- **0000**

**SW release**

- **R301**
### S0B – User Readable Name

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password protected user defined name which can be used to identify the node</td>
<td>cccccccccccccccc</td>
</tr>
</tbody>
</table>

**Operations**

- **R/W LOCAL**
- **R/W REMOTE**

**Write operation**

- `ATS0B=<name>:<password>`
- `ATREMS:<address>,0B=<name>:<password>`

Becomes effective **Instantly**

**Storage**

- **Non-Volatile**

**Parameters**

- Name with up to 16 characters.

**SW release**

- **R302**

### S0C – Password

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The local node’s password.</td>
<td>ccccccc</td>
</tr>
</tbody>
</table>

**Operations**

- **W LOCAL**
- **W REMOTE**

**Write operation**

- `ATS0C=<NEW>:<OLD>`
- `ATREMS:<address>,0C=<NEW>:<OLD>`

Becomes effective **Instantly**

**Storage**

- **Non-Volatile**

**Parameters**

- 8 case sensitive characters (8 bytes).
- Note that the password must have exactly 8 characters.

**Factory Default password**

- **Telegesis**

**SW release**

- **R300**
### S0D – Device Information

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>String containing the module's order code and firmware revision.</td>
<td><code>ccc…ccc</code></td>
</tr>
</tbody>
</table>

**Operations**
- **R LOCAL**
- **R REMOTE**

**Storage**
- **Non-Volatile**
- **SW release**
  - Factory Default
  - N/A
  - R300

---

### S0E – Prompt Enable 1

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the behaviour of the Device.</td>
<td><code>XXXX</code></td>
</tr>
</tbody>
</table>

Where `XXXX` represents a 16-bit hexadecimal number.

**Range**
- **0000 to FFFF**

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **Instantly**

**Storage**
- **Non-Volatile**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Set: Disable ‘&gt;’ prompt when entering binary data</td>
</tr>
<tr>
<td>E</td>
<td>Set: Disable UCAST, MCAST, BCAST, SCAST data</td>
</tr>
<tr>
<td>D</td>
<td>Set: Disable RAW data</td>
</tr>
<tr>
<td>C</td>
<td>Set: Disable SEQ prompt</td>
</tr>
<tr>
<td>B</td>
<td>Set: Disable SINK prompt</td>
</tr>
<tr>
<td>A</td>
<td>Set: Disable SR: prompt</td>
</tr>
<tr>
<td>9</td>
<td>Set: Disable RAW wrapper</td>
</tr>
<tr>
<td>8</td>
<td>Set: Disable NEWNODE prompt</td>
</tr>
<tr>
<td>7</td>
<td>Set: Disable NACK:XX prompt</td>
</tr>
<tr>
<td>6</td>
<td>Set: Disable ACK:XX</td>
</tr>
<tr>
<td>5</td>
<td>Set: Disable UCAST, MCAST, BCAST, SCAST wrapper</td>
</tr>
<tr>
<td>4</td>
<td>Set: Disable LeftPAN prompt</td>
</tr>
<tr>
<td>3</td>
<td>Set: Disable JPAN prompt</td>
</tr>
<tr>
<td>2</td>
<td>Set: Disable PWRCHANGE:nn prompt</td>
</tr>
<tr>
<td>1</td>
<td>Set: Disable OK prompt</td>
</tr>
<tr>
<td>0</td>
<td>Set: Disable ERROR:XX prompt</td>
</tr>
</tbody>
</table>

**Factory Default**
- 0000

**SW release**
- R300
## SOF – Prompt Enable 2

**Description**
Defines the behaviour of the Device.

**Operations**
- R/W LOCAL
- R/W REMOTE

**Becomes effective**
Instantly

**Notes**
Use of bit 8 or bit D requires bit 1 to be unset.

NODELEFT indicates that an end device has left from anywhere in the network, but only routers within one hop from the COO are indicated.

**Storage**
Non-Volatile

<table>
<thead>
<tr>
<th>Parameters</th>
<th>XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where XXXX represents a 16-bit hexadecimal number.</td>
<td></td>
</tr>
</tbody>
</table>

**Range**
0000 to FFFF

- **Bit F**: Add prefix to local S-register reads
- **Bit E**: Show RSSI and LQI for all received unicasts, broadcasts and AT+PANSCAN reports
- **Bit D**: Set: Display incoming ZDO messages by RX prompt instead of normal text prompt
- **Bit C**: Set: Message payload of RX prompt is displayed as hexadecimal instead of ASCII text
- **Bit B**: Set: Show NODELEFT prompt on COO when a device leaves the PAN
- **Bit A**: Set: Add remote endpoint and Network address to ACK and NACK prompts for profile IDs other than the Telegesis one
- **Bit 9**: Set: Disable SWRITE prompt
- **Bit 8**: Set: Show unhandled messages received by Endpoints 1 and above
- **Bit 7**: Set: Hide “AddrResp” prompt
- **Bit 6**: Set: Hide Network Manager Warning
- **Bit 5**: Set: Hide “DataMODE” prompt
- **Bit 4**: Set: Hide “CLOSED” prompt
- **Bit 3**: Set: Hide “OPEN” prompt
- **Bit 2**: Set: Hide all Sink Advertisements
  Unset: Show all Sink Advertisements, except advertisements by the current sink
- **Bit 1**: Set: Disable showing unhandled messages received by all Endpoints
- **Bit 0**: Set: Disable COO, FFD, SED and MED prompts

**Factory Default**
0006

**SW release**
R301
## S10 – Extended Function

### Description
Defines the behaviour of the Device.

### Operations
R/W LOCAL  
R/W REMOTE

Becomes effective  
**Instantly**

### Notes
Bit C: the ETRX357-LRS and ETRX357HR-LRS have an RF preamplifier, so the reported RSSI is 12dB higher than the power at the antenna.

Do not set bit 8 if the sink is likely to be missing and transmissions to the sink are frequent (about once a second)!

### Storage
Non-Volatile

<table>
<thead>
<tr>
<th>Parameters</th>
<th>XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where XXXX represents a 16-bit hexadecimal number.</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>0000 to FFFF</td>
</tr>
</tbody>
</table>

- **Bit F:** Set: Don’t exit data mode in case of data loss
- **Bit E:** Set: Don’t accept Data Mode
- **Bit D:** Set: High RAM concentrator instead of Low RAM concentrator
- **Bit C:** Set: Display RSSI and LQI of the last hop when devices report to AT+SN or AT+ANNCE

- **Bit B:** Set: UCASTs and SCASTs wait for ACK
- **Bit A:** Set: Disable playing Tune when receiving AT+IDENT
- **Bit 9:** Set: Enable one second character timeout when entering data for xCASTB
- **Bit 8:** Set: Actively search for a sink if none is known

- **Bit 7:** Set: Node doesn’t replace existing sink with better one (lower cost)
- **Bit 6:** Set: Node doesn’t lose sink if it couldn’t be reached for three times
- **Bit 5:** Set: Sink won’t reply to nodes searching for a sink
- **Bit 4:** Set: Node is Sink

- **Bit 3:** Set: Changes to S01 take effect instantly
- **Bit 2:** Set: Send BCAST[B] messages to routers only
- **Bit 1:** Set: Send unicast messages unacknowledged
- **Bit 0:** Set: Don’t attach EUI64 to NWK frame when sending a message.

**Factory Default**  
0000

| SW release | R302 |
## S11 – Device Specific

**Description**  
Defines the behaviour of the Device.

**Operations**  
R/W LOCAL  
R/W REMOTE

**Parameters**  

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| F (MSB) | Set: PB7 is PWM as defined by S1B/S1D.  
Unset: Standard I/O pin. |
| E      | Set: Enable Boost Mode                            |
| D      | Set: Show POLLED: prompt                          |
| C      | Reserved                                          |
| B      | Reserved                                          |
| A      | Reserved                                          |
| 9      | Set: Enable wakeup on UART activity (1st input character discarded) |
| 8      | Set: Enable 100ms debouncing for all IRQs          |
| 7      | Set: IRQ3 on rising edge                          |
| 6      | Set: IRQ3 on falling edge                         |
| 5      | Set: IRQ2 on rising edge                          |
| 4      | Set: IRQ2 on falling edge                         |
| 3      | Set: IRQ1 on rising edge                          |
| 2      | Set: IRQ1 on falling edge                         |
| 1      | Set: IRQ0 on rising edge                          |
| 0      | Set: IRQ0 on falling edge                         |

**Storage**  
Non-Volatile

**Note**  
On the ETRX3 series IRQ0,1,2,3 are generated by logic transitions on inputs PA0, PA1, PB0 and PB6 respectively.

If bit 8 is not set, the shortest interrupt pulse is 450ns.

**Factory Default**  
0005

**SW release**  
R301
### 4.4 I/O related S-Registers

#### S12 – UART Setup

<table>
<thead>
<tr>
<th>Description</th>
<th>The device’s RS232 Baudrate and mode. The default setting of 0500 results in: 19200bps, no parity, 1 stop bit, 8 data bits.</th>
</tr>
</thead>
</table>
| Operations  | **R/W LOCAL**  
**R/W REMOTE**  
Becomes effective **Instantly** |
| Note        | If bit 5 is set, bi-directional Hardware Flow Control is used instead of XON/XOFF flow control. If using Hardware flow control PB4 becomes the RTS output and the CTS input is assigned to PB3. Access to these I/Os via S16, S18 is blocked whilst Hardware Flow control is active. Note that in case the 128-byte output buffer of the ETRX357 is full data will be dropped. The parity settings do not affect the bytes transmitted over the air. Storage **Non-Volatile** |

#### Parameters

| XXXX | Where XXXX represents a 16-bit hexadecimal number. |

#### Range of the most significant byte

| 00 to 0C |  
| 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  |

#### Range of the least significant byte

| 00 to FF |  
| bit 7 set: Enable STX ETX wrapper  
| bit 6 Reserved  
| bit 5 set: H/W flow control enable  
| 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

| 0500 | R300 |

---

©2015 Telegesis (UK) Ltd - 81 - AT Command Manual (Rev 3.09)
## S13 – I/O Configuration

**Description**
Configures the I/O pins.

Setting a bit on the ETRX3 will have the following effect:

<table>
<thead>
<tr>
<th>S13</th>
<th>S16</th>
<th>S18</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Floating Input</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>floating input</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Output driving 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Output driving 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Input with pull-down</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Input with pull-up</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Open Drain Output (0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Open Drain Output (open)</td>
</tr>
</tbody>
</table>

### Parameters

Where XXXX represents a 16-bit hexadecimal number and XXXXXXXX represents a 32-bit hexadecimal number.

ETRX3: representing the I/O pins

$\text{xxxxxxx} <\text{PC7...PC0}> <\text{PB7...PB0}> <\text{PA7...PA0}>$

e.g. setting bit 7 to 1 will configure PA7 to either be an input with pull-up or pull-down, or an open-drain output.

**Factory Default**

**ETRX3: 00000000**

### Operations

- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**

**After Reset**

**Note**
The ETRX357’s current consumption may benefit from the use of pull-ups or pull-downs where inputs are otherwise unconnected.

### Storage

**Non-Volatile**

**SW release**

R300

## S14 – Reserved
### S15 – ETRX3: I/O Configuration

**Description**
This Register is used to enable alternate functionalities for each I/O pin. When set to zero the corresponding I/O pin is a standard I/O pin, when set to 1 any other setting for this I/O are overwritten by the peripheral functionality.

representing the I/O pins

\[ \text{<PC7...PC0><PB7...PB0><PA7...PA0>} \]

**Operations**

<table>
<thead>
<tr>
<th>R/W LOCAL</th>
<th>R/W REMOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Becomes effective After Reset</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

PA7 indicates that the UART has data to send.

PB0 is used internally on the ETRX357-LRS and ETRX357HR-LRS and is not available to the user.

**Storage**

Non-Volatile

**Parameters**

\[ \text{XXXXXXX} \]

Where XXXXXX represents a 32-bit hexadecimal number.

- bits 31-24 reserved
- bit 23 Set: PC7 indicates status of DMODE. Set High = Active, set low = Inactive. PC7 needs to be defined as output in S16 and can be overridden using S18
- bit 22: Set: Enable nTX_Active (reserved on -ERS Variants)
- bit 21 Set: Enable TX_Active (reserved on – LRS and –ERS Variants)
- bit 20 reserved (PC4)
- bit 19 reserved (PC3)
- bit 18 reserved (PC2)
- bit 17 Set: Enable ADC3 (PC1)
- bit 16 reserved (PC0)
- bit 15 Set: Enable ADC2, can be used as PWM out when enabled in S11 (PB7)
- bit 14 Set: Enable ADC1 (PB6)
- bit 13 Set: Enable ADC0, not available on – ERS variants (PB5)
- bit 12 Set: reserved, RTS when enabled in S12 (PB4)
- bit 11 Set: reserved, CTS when enabled in S12 (PB3)
- bit 10 Set: Enable RXD input (PB2)
- bit 9 Set: Enable TXD output (PB1)
- bit 8 Set: Enable 1.2V Vref Output during ADC conversions (PB0), reserved on -LRS and -ERS variants
- bit 7 Set: UART TX_ACTIVE (PA7)
- bit 6 reserved (PA6)
- bit 5 reserved (PA5)
- bit 4 reserved (PA4)
- bit 3 reserved (PA3)
- bit 2 reserved (PA2)
- bit 1 reserved (PA1)
- bit 0 reserved (PA0)

**Factory Default**

\[ \text{00000600} \]

**SW release**

R300
### S16 – Data Direction of I/O Port

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data direction of the module’s I/O port</td>
<td>XXXXXXXXX</td>
</tr>
</tbody>
</table>

Where XXXXXXXX represents a 32-bit hexadecimal number.

<table>
<thead>
<tr>
<th>Operations</th>
<th>ETRX3: representing the I/O pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
<td>xxxxxxxxx&lt;PC7…PC0&gt;&lt;PB7…PB0&gt;&lt;PA7…PA0&gt;</td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
</tr>
</tbody>
</table>

Becomes effective **Instantly**

Note: On the “-LRS” variants of the ETRX3 PC5 and PB0 are reserved and cannot be controlled using this register. On the “-ERS” variant PC6 and PB5 are also not freely configurable.

<table>
<thead>
<tr>
<th>Storage</th>
<th>XXXXXXXX&lt;PC7…PC0&gt;&lt;PB7…PB0&gt;&lt;PA7…PA0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Volatile</td>
<td></td>
</tr>
</tbody>
</table>

**Factory Default**

*Defined in S17*

---

### S17 – Initial Setting of S16

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial setting of S16 stored in non volatile memory</td>
<td>XXXXXXXXX</td>
</tr>
</tbody>
</table>

Where XXXXXXXX represents the initial value of S16 which is loaded after boot-up, soft or hard reset.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Factory Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
<td>ETRX3: 000142CC</td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
</tr>
</tbody>
</table>

Becomes effective **After Soft or Hard Reset**

<table>
<thead>
<tr>
<th>Storage</th>
<th>ETRX3: 000142CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Volatile</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
<th>R300</th>
</tr>
</thead>
</table>
### S18 – Output Buffer Of I/O Port

**Description**
The output buffer of the module's I/O port

**Operations**
- R/W LOCAL
- R/W REMOTE

**Becomes effective**
- Instantly

**Note:** On the “-LRS” variants of the ETRX3 PC5 and PB0 is reserved and cannot be controlled using this register.

**Storage**
- **Volatile**

**Parameters**
- XXXXXXXX

Where XXXXXXXX represents a 32-bit hexadecimal number.

ETRX3: representing the I/O pins
- xxxxxxxx <PC7…PC0> <PB7…PB0> <PA7…PA0>

E.g. setting bit 7 to 1 will cause PA7 to drive high (depending on settings in S16 and S15)

**Factory Default**
- Defined in S19

**SW release**
- R300

---

### S19 – Initial Setting of S18

**Description**
The initial setting of S18 stored in non volatile memory

**Operations**
- R/W LOCAL
- R/W REMOTE

**Becomes effective**
- After Soft or Hard Reset

**Storage**
- **Non-Volatile**

**Parameters**
- XXXXXXXX

Where XXXXXXXX represents the initial value of S18 which is loaded after boot-up, soft or hard reset.

**Factory Default**
- ETRX3: 00000000

**SW release**
- R300
## S1A – Input Buffer of I/O Port

<table>
<thead>
<tr>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Logical Levels at the I/O Pins</td>
<td>00000000 to FFFFFFFF (ETRX3)</td>
</tr>
</tbody>
</table>

**ETRX3**: representing the I/O pins

```
xxxxxxxxxx  <PC7...PC0>  <PB7...PB0>
<PA7...PA0>
```

S1A represents the logic level at each pin of the I/O port.

**Storage**

- **Instant Reading of Port Status**

**Operations**

- R LOCAL
- R REMOTE

Becomes effective

- **Instantly**

**SW release**

- R300

---

## S1B – PWM Pin Top Value

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The mode of operation for the special function pin. S1B controls the PWM frequency.</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

Frequency = \( \frac{12\text{MHz}}{(\{S1B\}+1)} \)

**Operations**

- R/W LOCAL
- R/W REMOTE

**Operations**

- **Instantly**

**Storage**

- **Volatile**

**SW release**

- R300

---

This register represents the top value of the 16-bit counter counting from 0 to top repeatedly incrementing at 12MHz. When reaching top PB7 is set, given that the PWM is enabled in S11.

**Factory Default**

- **Defined in S1C**
### S1C – Initial value of S1B

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial setting of S1B stored in non volatile memory</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

Where XXXX represents the initial value of S1B which is loaded after boot-up, soft or hard reset.

<table>
<thead>
<tr>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
</tr>
<tr>
<td>R/W REMOTE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Becomes effective</th>
<th>After Soft or Hard Reset</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>Non-Volatile</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R309</td>
</tr>
</tbody>
</table>

### S1D – PWM Pin Compare Value

| Description | Parameters | Range |
|-------------|------------|
| The mode of operation for the special function pin. S1D controls the PWM duty cycle | XXXX | 0000 to FFFF |

Duty cycle = \( \frac{(S1D)}{(S1B+1)} \)

<table>
<thead>
<tr>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
</tr>
<tr>
<td>R/W REMOTE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Becomes effective</th>
<th>Instantly</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>Volatile</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
</tr>
</tbody>
</table>

If the special function pin is enabled by setting bit F of S11, this register represents the compare value of the 16-bit counter counting from 0 to top repeatedly incrementing at 12MHz. When reaching compare PB7 is cleared.

Factory Default

**Defined in S1E**

### S1E – Initial Value S1D

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial setting of S1D stored in non volatile memory</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

Where XXXX represents the initial value of S1D which is loaded after boot-up, soft or hard reset.

<table>
<thead>
<tr>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W LOCAL</td>
</tr>
<tr>
<td>R/W REMOTE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Becomes effective</th>
<th>After Soft or Hard Reset</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>Non-Volatile</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
</tr>
</tbody>
</table>

Factory Default

**1D4C (800Hz 50% m/s ratio)**
## S1F – ADC0 Reading

**Description**
The analogue reading of ADC0

Valid only when bit 13 (0x0D) of S15 is set, invalid otherwise

**Operations**
- **R LOCAL**
- **R REMOTE**

Becomes effective **Instantly**

**Parameters**
**XXXX**

**Representation**
The hexadecimal reading of the analogue input in mV * 10 with respect to ground. The return value will be undefined in case the corresponding A/D converter has not been enabled.

**Range**
ETRX3: 0000 – 2EE0 (0 – 12000)

**Storage**
- **Instant Reading of analogue input**

---

## S20 – ADC1 Reading

**Description**
The analogue reading of ADC1

Valid only when bit 14 (0x0E) of S15 is set, invalid otherwise

**Operations**
- **R LOCAL**
- **R REMOTE**

Becomes effective **Instantly**

**Parameters**
**XXXX**

**Representation**
The hexadecimal reading of the analogue input in mV * 10 with respect to ground. The return value will be undefined in case the corresponding A/D converter has not been enabled.

**Range**
ETRX3: 0000 – 2EE0 (0 – 12000)
### S21 – ADC2 Reading

**Description**
The analogue reading of ADC2

Valid only when bit 15 (0x0F) of S15 is set, invalid otherwise

**Operations**
- R LOCAL
- R REMOTE

**Becomes effective**
Instantly

**Storage**
**Instant Reading of analogue input**

**Parameters**
- **XXXX**

**Representation**
The hexadecimal reading of the analogue input in mV * 10 with respect to ground. The return value will be undefined in case the corresponding A/D converter has not been enabled.

**Range**
**ETRX3: 0000 – 2EE0 (0 – 12000)**

---

### S22 – ADC3 Reading

**Description**
The analogue reading of ADC3

Valid only when bit 17 (0x11) of S15 is set, invalid otherwise

**Operations**
- R LOCAL
- R REMOTE

**Becomes effective**
Instantly

**Storage**
**Instant Reading of analogue input**

**Parameters**
- **XXXX**

**Representation**
The hexadecimal reading of the analogue input in mV * 10 with respect to ground. The return value will be undefined in case the corresponding A/D converter has not been enabled.

**Range**
**ETRX3: 0000 – 2EE0 (0 – 12000)**
4.5 S-Registers Defining the Functionality of the Module

There are 14 events which can trigger a user-selectable action to prevent the need for a host microcontroller for simple applications. Four out of those 14 events are the external interrupts which can be enabled in register S11. The actions to be performed on those four interrupt events are defined in S23 to S26. The user can pick any of the actions from the list in section 5 of this document and assign them to any event.

Two further events occur when the unit is reset or power cycled, or joins a network.

The remaining 8 events are timed events. Registers S29 to S38 control those 8 timers and their corresponding events. Please note that the first 4 timers are used by default for network management tasks, which can be modified by the user when changing the corresponding registers. A timer will increment every 250ms (4 times a second) and when the timer reaches the value stored in the timer/counter register the corresponding action will be executed.

For examples, see the descriptions of register S23 and register pair S29/S2A.

<table>
<thead>
<tr>
<th>S23 – Immediate Functionality At IRQ0 (PA0)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>XXXX</td>
</tr>
<tr>
<td>Describes the immediate action taken on IRQ0.</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
</tr>
<tr>
<td>R/W LOCAL</td>
<td></td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
</tr>
<tr>
<td>Becomes effective</td>
<td></td>
</tr>
<tr>
<td>Instantly</td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td></td>
</tr>
<tr>
<td>Non-Volatile</td>
<td></td>
</tr>
<tr>
<td>SW release</td>
<td></td>
</tr>
<tr>
<td>Factory Default</td>
<td></td>
</tr>
<tr>
<td>0001 (Wakeup to power mode 0)</td>
<td>R300</td>
</tr>
</tbody>
</table>

If set to 0 the functionality is disabled. Please see section 5 for a list of available functionalities.
### S24 – Immediate Functionality At IRQ1 (PA1)

**Description**
Describes the immediate action taken on IRQ1.

**Operations**

<table>
<thead>
<tr>
<th>R/W LOCAL</th>
<th>R/W REMOTE</th>
</tr>
</thead>
</table>

Becomes effective **Instantly**

**Storage**
**Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
</tr>
</tbody>
</table>

If set to 0 the functionality is disabled. Please see section 5 for a list of available functionalities.

**Factory Default**

| 0000 (none) |

**SW release**

| R300 |

---

### S25 – Immediate Functionality At IRQ2 (PB0)

**Description**
Describes the immediate action taken on IRQ2.

**Operations**

<table>
<thead>
<tr>
<th>R/W LOCAL</th>
<th>R/W REMOTE</th>
</tr>
</thead>
</table>

Becomes effective **Instantly**

**Note**

PB0 is used internally on the ETRX357(HR)-LRS and is not available to the user

**Storage**
**Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
</tr>
</tbody>
</table>

If set to 0 the functionality is disabled. Please see section 5 for a list of available functionalities.

**Factory Default**

| 0000 (none) |

**SW release**

| R300 |

---
### S26 – Immediate Functionality At IRQ3 (PB6)

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describes the immediate action taken on IRQ3.</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

- **Operations**
  - R/W LOCAL
  - R/W REMOTE

- **Becomes effective**
  - **Instantly**

- **Storage**
  - **Non-Volatile**

- **Factory Default**
  - 0000 (none)

- **SW release**
  - R300
### S27 – Functionality at Bootup

**Description**
Describes the immediate action taken after boot-up (and stack initialization).

**Operations**
R/W LOCAL  
R/W REMOTE

**Parameters**

<table>
<thead>
<tr>
<th>XXXX</th>
</tr>
</thead>
</table>

If set to 0 the functionality is disabled. Please see section 5 for a list of available functionalities.

**Note**
On versions before R305 this was executed before the protocol stack was running so it could not be used for network or message functions.

**Storage**
Non-Volatile

**Factory Default**
0000 (none)

**SW release**
R305

### S28 – Functionality at Network Join

**Description**
Describes the immediate action taken after joining a network.

**Operations**
R/W LOCAL  
R/W REMOTE

**Parameters**

<table>
<thead>
<tr>
<th>XXXX</th>
</tr>
</thead>
</table>

If set to 0 the functionality is disabled. Please see section 5 for a list of available functionalities.

**Note**

**Storage**
Non-Volatile

**Factory Default**
0000 (none)

**SW release**
R302
### S29 – Timer/Counter 0

**Description**
A multipurpose Timer/Counter whose functionality is defined by S2A

**Parameters**
- **XXXX**

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **Instantly**

**Storage**
- **Non-Volatile**

**Factory Default**
- **0004 (1s interval)**

**SW release**
- **R300**

### S2A – Functionality For Timer/Counter 0

**Description**
Defines the functionality for Timer/Counter 0 events.

**Parameters**
- **XXXX**

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **Instantly**

**Storage**
- **Non-Volatile**

**Factory Default**
- **8010 (end devices poll parent)**

**SW release**
- **R300**
### S2B – Timer/Counter 1

**Description**  
A multipurpose Timer/Counter whose functionality is defined by S2C

**Operations**  
- R/W LOCAL
- R/W REMOTE

Becomes effective  
**Instantly**

**Storage**  
**Non-Volatile**

**Parameters**  
**XXXX**

A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.

**Factory Default**  
**00F0 (1 min interval)**

**SW release**  
R300

### S2C – Functionality For Timer/Counter 1

**Description**  
Defines the functionality for Timer/Counter 1 events.

**Operations**  
- R/W LOCAL
- R/W REMOTE

Becomes effective  
**Instantly**

**Storage**  
**Non-Volatile**

**Parameters**  
**XXXX**

If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.

**Factory Default**  
**821E (advertise sink for 30 hops and create aggregation routes to COO and sinks)**

**SW release**  
R300
## S2D – Timer/Counter 2

**Description**
A multipurpose Timer/Counter whose functionality is defined by S2E

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

  Becomes effective
  - **Instantly**

**Storage**
- **Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
<td>A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.</td>
</tr>
</tbody>
</table>

**Factory Default**
- **00F4 (1 min 1s interval)**

**SW release**
- **R300**

## S2E – Functionality For Timer/Counter 2

**Description**
Defines the functionality for Timer/Counter 2 events.

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

  Becomes effective
  - **Instantly**

**Storage**
- **Non-Volatile**

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
<td>If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.</td>
</tr>
</tbody>
</table>

**Factory Default**
- **8014 (leave network if I am alone)**

**SW release**
- **R300**
### S2F – Timer/Counter 3

**Description**
A multipurpose Timer/Counter whose functionality is defined by S30

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
**Instantly**

**Storage**
Non-Volatile

**Parameters**
- **XXXX**

A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.

**Factory Default**
- **00F2 (1min interval)**

**SW release**
R300

---

### S30 – Functionality For Timer/Counter 3

**Description**
Defines the functionality for Timer/Counter 3 events.

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
**Instantly**

**Storage**
Non-Volatile

**Parameters**
- **XXXX**

If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.

**Factory Default**
- **8015 (if not part of a network do AT+JN)**

**SW release**
R300
### S31 – Timer/Counter 4

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>A multipurpose Timer/Counter whose functionality is defined by S32</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

**Operations**
- R/W LOCAL
- R/W REMOTE

Becomes effective **Instantly**

**Storage**
- Non-Volatile

**Factory Default**
- 0000

**SW release**
- R302

---

### S32 – Functionality For Timer/Counter 4

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the functionality for Timer/Counter 4 events.</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

**Operations**
- R/W LOCAL
- R/W REMOTE

Becomes effective **Instantly**

**Storage**
- Non-Volatile

**Factory Default**
- 0000

**SW release**
- R302
### S33 – Timer/Counter 5

**Description**
A multipurpose Timer/Counter whose functionality is defined by S34.

**Parameters**
- \textit{XXXX}

**Operations**
- \textbf{R/W LOCAL}
- \textbf{R/W REMOTE}

**Becomes effective**
- \textit{Instantly}

**Storage**
- \textbf{Non-Volatile}

**SW release**
- R300

A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.

**Factory Default**
- 0000

---

### S34 – Functionality For Timer/Counter 5

**Description**
Defines the functionality for Timer/Counter 5 events.

**Parameters**
- \textit{XXXX}

**Operations**
- \textbf{R/W LOCAL}
- \textbf{R/W REMOTE}

**Becomes effective**
- \textit{Instantly}

**Storage**
- \textbf{Non-Volatile}

**Factory Default**
- 0000

If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.
### S35 – Timer/Counter 6

**Description**
A multipurpose Timer/Counter whose functionality is defined by S36

**Operations**
- R/W LOCAL
- R/W REMOTE

**Parameters**
- `XXXX`

A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.

**Storage**
- Non-Volatile

**Factory Default**
- 0000

**SW release**
- R300

---

### S36 – Functionality For Timer/Counter 6

**Description**
Defines the functionality for Timer/Counter 6 events.

**Operations**
- R/W LOCAL
- R/W REMOTE

**Parameters**
- `XXXX`

If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.

**Storage**
- Non-Volatile

**Factory Default**
- 0000

**SW release**
- R300
### S37 – Timer/Counter 7

**Description**
A multipurpose Timer/Counter whose functionality is defined by S38

<table>
<thead>
<tr>
<th>Parameters</th>
<th>XXXX</th>
</tr>
</thead>
</table>

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

Becomes effective
*Instantly*

**Storage**
*Non-Volatile*

**SW release**
R300

A 16-bit hexadecimal number representing a threshold for either a timer or counter event to be triggered. When reading this register the threshold rather than the actual timer/counter value is displayed. If set to 0 the corresponding functionality is disabled.

### S38 – Functionality For Timer/Counter 7

**Description**
Defines the functionality for Timer/Counter 7 events.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>XXXX</th>
</tr>
</thead>
</table>

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

Becomes effective
*Instantly*

**Storage**
*Non-Volatile*

**Factory Default**
0000

If set to 0 the functionality is disabled. Please see section 5 for a list of the functionalities.
### S39 – Power Mode

**Description**
The current power mode of the module

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **Instantly**

**Storage**
- **Volatile**

**Parameters**
- **XXXX**

**Range**
- **0000 – 0004**

The module’s power mode as described in section 6.

**Factory Default**
- Defined in S3A

**SW release**
- **R300**

### S3A – Initial Power Mode

**Description**
The module’s power mode after start-up and reset.

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **After hard or soft reset**

**Storage**
- **Non-Volatile**

**Parameters**
- **XXXX**

**Range**
- **0000 – 0004**

The module’s power mode as described in section 6.

**Factory Default**
- **0000**

**SW release**
- **R300**

### S3B – Start-up Functionality Plaintext A

**Description**
Contains Text which is used by some of the actions described in section 5.

**Operations**
- **R/W LOCAL**
- **R/W REMOTE**

**Becomes effective**
- **Instantly**

**Storage**
- **Non-Volatile**

**Parameters**
- **Up to 50 characters**

**Factory Default**
- **BUTTON3**

**SW release**
- **R300**
### S3C – Start-up Functionality Plaintext B

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains Text which is used by some of the actions described in section 5.</td>
<td><strong>Up to 50 characters</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R/W LOCAL</strong></td>
<td></td>
</tr>
<tr>
<td><strong>R/W REMOTE</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>Factory Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-Volatile</strong></td>
<td><strong>BUTTON4</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R309</td>
<td>●</td>
</tr>
</tbody>
</table>

### S3D – Supply Voltage

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Supply voltage of the device in mV.</td>
<td><strong>nnnn</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R LOCAL</strong></td>
<td></td>
</tr>
<tr>
<td><strong>R REMOTE</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>Factory Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volatile</strong></td>
<td><strong>N/A</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SW release</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R300</td>
<td>●</td>
</tr>
</tbody>
</table>
## Advanced Settings

### S3E – Multicast Table Entry 00

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ID portion of Multicast Table Entry 00</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

### Operations

- **R/W LOCAL**
- **R/W REMOTE**

### Becomes effective

**Instantly**

### Note

- Same effect as AT+MSET, but can be set at boot-up by built-in functionality

### Storage

**Non-Volatile**

### SW release

R300

### Factory Default

0000

---

### S3F – Multicast Table Entry 01

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ID portion of Multicast Table Entry 01</td>
<td>XXXX</td>
</tr>
</tbody>
</table>

### Operations

- **R/W LOCAL**
- **R/W REMOTE**

### Becomes effective

**Instantly**

### Note

- As for S3E

### Storage

**Non-Volatile**

### SW release

R300
## S40 – Source and Destination Endpoints for xCASTs (volatile)

<table>
<thead>
<tr>
<th>Description</th>
<th>The source and destination endpoints for all messages.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td>R/W LOCAL  [\text{R/W REMOTE}]</td>
</tr>
<tr>
<td>Becomes effective</td>
<td>\text{Instantly}</td>
</tr>
<tr>
<td>Storage</td>
<td>Volatile</td>
</tr>
<tr>
<td>Parameters</td>
<td>ssdd</td>
</tr>
<tr>
<td>Factory Default</td>
<td>Defined in S41</td>
</tr>
<tr>
<td>SW release</td>
<td>R300</td>
</tr>
</tbody>
</table>

Where ss is the hexadecimal source endpoint and dd is the hexadecimal destination endpoint.

## S41 – Initial Setting of S40

<table>
<thead>
<tr>
<th>Description</th>
<th>The initial setting of S40 stored in non volatile memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td>R/W LOCAL  [\text{R/W REMOTE}]</td>
</tr>
<tr>
<td>Becomes effective</td>
<td>After Soft or Hard Reset</td>
</tr>
<tr>
<td>Storage</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td>Parameters</td>
<td>ssdd</td>
</tr>
<tr>
<td>Factory Default</td>
<td>\text{0101}</td>
</tr>
<tr>
<td>SW release</td>
<td>R300</td>
</tr>
</tbody>
</table>

Where ss is the hexadecimal source endpoint and dd is the hexadecimal destination endpoint.
### S42 – Cluster ID for xCASTs (volatile)

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cluster ID attached to any xCAST message.</td>
<td>XXXX</td>
</tr>
<tr>
<td>Operations</td>
<td></td>
</tr>
<tr>
<td>R/W LOCAL</td>
<td></td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
</tr>
<tr>
<td>Becomes effective</td>
<td>Instantly</td>
</tr>
<tr>
<td>Storage</td>
<td>Volatile</td>
</tr>
<tr>
<td>Factory Default</td>
<td>Defined in S43</td>
</tr>
<tr>
<td>SW release</td>
<td>R300</td>
</tr>
</tbody>
</table>

### S43 – Initial Setting of S42

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial setting of S42 stored in non volatile memory</td>
<td>XXXX</td>
</tr>
<tr>
<td>Operations</td>
<td></td>
</tr>
<tr>
<td>R/W LOCAL</td>
<td></td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td></td>
</tr>
<tr>
<td>Becomes effective</td>
<td>After Soft or Hard Reset</td>
</tr>
<tr>
<td>Storage</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td>Factory Default</td>
<td>0002</td>
</tr>
<tr>
<td>SW release</td>
<td>R300</td>
</tr>
</tbody>
</table>
### S44 – Profile ID for xCASTs (volatile)

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
<th>Operations</th>
<th>Becomes effective</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>The profile ID for all messages.</td>
<td>XXXX</td>
<td>R/W LOCAL, R/W REMOTE</td>
<td>Instantly</td>
<td>Volatile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SW release</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Factory Default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### S45 – Initial Setting of S44

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
<th>Operations</th>
<th>Becomes effective</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial setting of S44 stored in non volatile memory</td>
<td>XXXX</td>
<td>R/W LOCAL, R/W REMOTE</td>
<td>After Soft or Hard Reset</td>
<td>Non-Volatile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SW release</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Factory Default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### S46 – Start-up Functionality 32-bit number

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
<th>Operations</th>
<th>Becomes effective</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volatile 32-bit register which can be used by some of the build in functionality</td>
<td>XXXXXXXXXX</td>
<td>R/W LOCAL, R/W REMOTE</td>
<td>Instantly</td>
<td>Volatile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SW release</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Factory Default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S47 – Power Descriptor</td>
<td>S48 – Endpoint 2 Profile ID</td>
<td>S49 – Endpoint 2 Device ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------</td>
<td>---------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td><strong>Description</strong></td>
<td><strong>Description</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register defining the Node’s power descriptor, which can be read by remote nodes using ZigBee compliant ZDO messaging.</td>
<td>Register defining the Profile ID of the Node’s endpoint 2.</td>
<td>Register defining the Device ID of the Node’s endpoint 2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td><strong>Operations</strong></td>
<td><strong>Operations</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W LOCAL</td>
<td>R/W LOCAL</td>
<td>R/W LOCAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/W REMOTE</td>
<td>R/W REMOTE</td>
<td>R/W REMOTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Becomes effective</td>
<td>Becomes effective</td>
<td>Becomes effective</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Instantly</strong></td>
<td><strong>Instantly</strong></td>
<td><strong>Instantly</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td><strong>Storage</strong></td>
<td><strong>Storage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Volatile</td>
<td>Non-Volatile</td>
<td>Non-Volatile</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td><strong>Parameters</strong></td>
<td><strong>Parameters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXX</td>
<td>XXXX</td>
<td>XXXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Power Descriptor as described in section 2.3.2.4 of the ZigBee specification.</td>
<td>The 16 bit Profile ID.</td>
<td>The 16 bit device ID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Factory Default</strong></td>
<td><strong>Factory Default</strong></td>
<td><strong>Factory Default</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C110</td>
<td>C091</td>
<td>0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SW release</strong></td>
<td><strong>SW release</strong></td>
<td><strong>SW release</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R302</td>
<td>R302</td>
<td>R302</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>S4A – Endpoint 2 Device Version</strong></td>
<td><strong>Description</strong></td>
<td>Register defining the version of the device attached to the Node’s endpoint 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Operations</strong></td>
<td><strong>R/W LOCAL</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>R/W REMOTE</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Becomes effective</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Instantly</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Non-Volatile</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td><strong>00XX</strong></td>
<td>The 8 bit version preceded by two leading 0’s.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Factory Default</strong></td>
<td><strong>0000</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SW release</strong></td>
<td><strong>R302</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>S4B – Endpoint 2 Input Cluster List</strong></th>
<th><strong>Description</strong></th>
<th>Register defining the input cluster list of the device attached to the Node’s endpoint 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operations</strong></td>
<td><strong>R/W LOCAL</strong></td>
<td></td>
</tr>
<tr>
<td><em>R/W REMOTE</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Becomes effective</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Instantly</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Non-Volatile</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td><em>[XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX]</em></td>
<td>A list of a maximum of 12 x 16 bit cluster IDs separated by commas. The list may have any length from 0 to 12 clusters</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td></td>
<td><strong>ATS4B=1101,000D</strong></td>
</tr>
<tr>
<td><strong>Factory Default</strong></td>
<td><strong>0000</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SW release</strong></td>
<td><strong>R302</strong></td>
<td></td>
</tr>
</tbody>
</table>
### S4C – Endpoint 2 Output Cluster List

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register defining the output cluster list of the device attached to the Node’s endpoint 2</td>
<td><code>[XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX,XXXX]</code></td>
</tr>
</tbody>
</table>

- **Operations**
  - R/W LOCAL
  - R/W REMOTE

- **Becomes effective**
  - Instantly

- **Storage**
  - Non-Volatile

- **Factory Default**
  - 0000

- **SW release**
  - R302

A list of a maximum of 12 x 16 bit cluster IDs separated by commas. The list may have any length from 0 to 12 clusters.

### S4D – Mobile End Device Poll Timeout

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>In an FFD, the register defining the amount of time after which a mobile end device times out of its parent’s child table if it has not polled.</td>
<td><code>00XX</code></td>
</tr>
<tr>
<td>In an MED, defines the delay between an MED losing contact with its parent and reporting “LostPAN”.</td>
<td>8-bit number led by “00”, which represents the timeout in quarter seconds. The default of 0x0014 results in a timeout of 5s.</td>
</tr>
</tbody>
</table>

- **Operations**
  - R/W LOCAL
  - R/W REMOTE

- **Becomes effective**
  - After Soft or Hard Reset

- **Note**
  - The timer tested according to S4D continues to run in Power Mode 3

- **Storage**
  - Non-Volatile

- **Factory Default**
  - 0014

- **SW release**
  - R302
### S4E – End Device Poll Timeout

**Description**
Register in a parent defining the amount of time after which an SED or ZED times out of its parent’s child table if it has not polled

Register in an SED or ZED defining the amount of time after which it reports LostPAN when it loses the parent

**Parameters**

<table>
<thead>
<tr>
<th>XXYY</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

**Operations**

- R/W LOCAL
- R/W REMOTE

**Becomes effective**
After Soft or Hard Reset

**Note**
The timer tested according to S4E continues to run in Power Mode 3

**Storage**
Non-Volatile

**SW release**

<table>
<thead>
<tr>
<th>R303</th>
</tr>
</thead>
</table>

### S4F – MAC Timeout

**Description**
Register defining the MAC timeout

**Parameters**

<table>
<thead>
<tr>
<th>XXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>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. The default value is 7680ms. The maximum is 30s</td>
</tr>
</tbody>
</table>

**Operations**

- R/W LOCAL
- R/W REMOTE

**Becomes effective**
After Soft or Hard Reset

**Storage**
Non-Volatile

**SW release**

| R302 |

<table>
<thead>
<tr>
<th>0E00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory Default</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1E00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory Default</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0605</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory Default</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0EFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7530</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>

©2015 Telegesis (UK) Ltd

- 111 -

AT Command Manual (Rev 3.09)
## Build in Functionality

The following table gives an overview of the built-in functionality which can be triggered either by the four external interrupts, boot-up, joining a network, or by 8 individually programmable timers/counters. If the node is in a low power mode and the action requires the node to wake up, the node will do so and go back to its original power mode after completion of the action. When a function is used in conjunction with a timer, the timer will restart only when the most significant bit of the function code is set to 1 (e.g. instead of 0001 set 8001). The timers count 250ms intervals.

The timers run in power modes 0, 1 and 2, but in power modes 3 and 4 their count values are frozen and their associated functions are therefore not executed.

The timers are firmware functions, not hardware circuits, so negligible power is saved by deleting unused timers.

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>No operation of the corresponding interrupt/timer/counter</td>
</tr>
<tr>
<td>0001</td>
<td>Change to power mode 0.</td>
</tr>
<tr>
<td>0002</td>
<td>Change to power mode 1.</td>
</tr>
<tr>
<td>0003</td>
<td>Change to power mode 2.</td>
</tr>
<tr>
<td>0004</td>
<td>Change to power mode 3.</td>
</tr>
<tr>
<td>0005</td>
<td>Change to power mode 4.</td>
</tr>
<tr>
<td>…</td>
<td>Reserved</td>
</tr>
<tr>
<td>0010</td>
<td>If I am an end device Poll Parent for data.</td>
</tr>
<tr>
<td>0011</td>
<td>Update the Network key with new random key.</td>
</tr>
<tr>
<td>0012</td>
<td>Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the current network key each time this functionality is triggered. Note: No functionality on COOs.</td>
</tr>
<tr>
<td>0013</td>
<td>Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the known network key. If this is unsuccessful try an unsecured rejoin each time this functionality is triggered from there on. Note: No functionality on COOs.</td>
</tr>
<tr>
<td>0014</td>
<td>Check for other devices on the network. If no other devices could be found for three consecutive tries, attempt a rejoin using the known network key. If this is unsuccessful try a rejoin using the current link key the next time this functionality is triggered. If this is unsuccessful leave the current network. Notes: No functionality on COOs. This function may be inhibited if other devices try to join the PAN</td>
</tr>
<tr>
<td>0015</td>
<td>In case I am not joined to a network scan for and join the next best</td>
</tr>
<tr>
<td>0016</td>
<td>Reserved</td>
</tr>
<tr>
<td>0017</td>
<td>Allow joining via the local node for 60 Seconds (when it is disabled using bit 0 of S0A)</td>
</tr>
<tr>
<td>0018</td>
<td>Copy local Inputs to Remote outputs: Read local S1A and if changed since the previous time, write the reading into the remote S18 whose address is given in S3B. S3B can also contain the index of a binding table entry that points to a multicast group</td>
</tr>
<tr>
<td>001D</td>
<td>End Data Mode (if active)</td>
</tr>
<tr>
<td>001E</td>
<td>Disassociate from the PAN if no coordinator or sink has been heard from since the function was last executed. Note 1: this has no effect on an End Device. Note 2: only execute this function every few minutes or devices may leave the PAN unexpectedly. Note 3: The actual code in the register needs to be 801E</td>
</tr>
</tbody>
</table>
### Overview of Actions

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0020</td>
<td>Check for other devices on the network. If no other devices could be found for three consecutive tries, set S11F to 1 to enable PWM, then attempt a rejoin using the current network key each time this functionality is triggered. Note: No functionality on COOs</td>
</tr>
<tr>
<td>0021</td>
<td>Check for other devices on the network. If no other devices could be found for three consecutive tries, set S11F to 0 to disable PWM, then attempt a rejoin using the current network key each time this functionality is triggered. Note: No functionality on COOs</td>
</tr>
<tr>
<td>003x</td>
<td>Toggle PA0-7 (x=0-7) or PB0-7 (x=8-F)</td>
</tr>
<tr>
<td>004x</td>
<td>Flash I/Ox (pull low) for 250ms. Note: can only be triggered by a timer, not an IRQ</td>
</tr>
<tr>
<td>005x</td>
<td>Set PA0-7 (x=0-7) or PB0-7 (x=8-F) to 0</td>
</tr>
<tr>
<td>006x</td>
<td>Set PA0-7 (x=0-7) or PB0-7 (x=8-F) to 1</td>
</tr>
<tr>
<td>0108</td>
<td>The unit sends the contents of S3B to the networks sink.</td>
</tr>
<tr>
<td>0109</td>
<td>The unit sends the contents of S3C to the networks sink.</td>
</tr>
<tr>
<td>0110</td>
<td>Sends the reading of the lower 16 I/O pins, A/D1 &amp; A/D2 or ADC0 &amp; ADC1 and Vcc as well as an 8-bit transmission counter which increments with every transmission to the network’s sink, and if no sink is known the unit will search for a sink immediately when bit 8 of S10 is set.</td>
</tr>
<tr>
<td>0111</td>
<td>Same as 0110, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>0112</td>
<td>Send a Tracking Message to all nearby routers which will forward this message and the RSSI reading to their nearest sink.</td>
</tr>
<tr>
<td>0113</td>
<td>Same as 0112, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>0114</td>
<td>Same as 0112, but tracking message doesn't contain ADC readings to save power on tracked device (TRACK2 Prompt)</td>
</tr>
<tr>
<td>0115</td>
<td>Same as 0114, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>0116</td>
<td>Send a Tracking Message to all nearby routers which will forward this message and the RSSI reading to their nearest sink if the local RSSI reading is higher than the first three digits of S3B (S3B is interpreted as snnsnn). S3B must be set in all routers</td>
</tr>
<tr>
<td>0117</td>
<td>Same as 0116, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>0118</td>
<td>Send a Tracking Message to all nearby routers which will forward this message and the RSSI reading to their nearest sink if the local RSSI reading is higher than the second three digits of S3B (S3B is interpreted as snnsnn). Tracking message does not contain ADC readings to save power on tracked device (TRACK2 Prompt)</td>
</tr>
<tr>
<td>0119</td>
<td>Same as 0118, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>0120</td>
<td>Sends the contents of S3B as a RAW transmission.</td>
</tr>
<tr>
<td>0121</td>
<td>Sends the contents of S3C as a RAW transmission.</td>
</tr>
<tr>
<td>0130</td>
<td>Sends to the network’s sink the reading of the I/O, an 8-bit transmission counter which increments with every transmission, the contents of S46 and the reading of any A/D [1..4] enabled in S15. If no sink is known the unit will search for a sink immediately when bit 8 of S10 is set</td>
</tr>
<tr>
<td>0131</td>
<td>Same as 0130, but to charge an external RC timer PA3 is pulled high whilst sending the data and left high impedance the rest of the time.</td>
</tr>
<tr>
<td>02XX</td>
<td>If I am a Sink advertise me for x hops (max. no. of hops: 30). If I am a COO create aggregation routes needed for Trust Centre. NB message does not reach end devices when parent already knows a sink address</td>
</tr>
<tr>
<td>0300</td>
<td>Increment S46</td>
</tr>
<tr>
<td>0301</td>
<td>Decrement S46</td>
</tr>
</tbody>
</table>
### Overview of Actions

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0302</td>
<td>Clear S46</td>
</tr>
<tr>
<td>0400</td>
<td>Show status on PA7. LED on (pin driven low) = no connection. Blinking fast = Auto-searching for PAN. Blinking slow = connected to PAN. The accompanying counter register defines the update interval. Note: PA7 must be defined to be an output.</td>
</tr>
<tr>
<td>0401</td>
<td>Show status on PB7. LED on (pin driven low) = no connection. Blinking fast = Auto-searching for PAN. Blinking slow = connected to PAN. The accompanying counter register defines the update interval. Note: PB7 must be defined to be an output.</td>
</tr>
<tr>
<td>0402</td>
<td>Show status on PA6. LED on (pin driven low) = no connection. Blinking fast = Auto-searching for PAN. Blinking slow = connected to PAN. The accompanying counter register defines the update interval. Note: PA6 must be defined to be an output.</td>
</tr>
<tr>
<td>2000</td>
<td>When triggered the number of times listed in the accompanying counter a message is sent to the sink containing a transmission counter and the reading of the analogue and digital inputs. Note: Can only be triggered by setting S23, S24, S25 or S26 to 24XX.</td>
</tr>
<tr>
<td>2001</td>
<td>When enabling this action the command line is disabled and as soon as a number of bytes in excess of the number N specified in the accompanying timer/counter register is received on the serial port, a SCAST containing these characters is sent to the network’s sink. Notes: This event is triggered by receiving a character on the serial port. N ≤ 64. N must be written as a 4-character hexadecimal number. This function can only be stopped by overwriting the register that contains code 2001 from another node.</td>
</tr>
<tr>
<td>2002</td>
<td>Sends input from the serial port as a unicast to the node whose address (EUI, network address or table entry) is stored in S3B. The message will be sent when either no more data has come for 20ms or the packet is full. The accompanying timer register is not used.</td>
</tr>
<tr>
<td>2100</td>
<td>The contents of S3B is sent to the local command line followed by carriage return. Note: No AT-Prefix required!</td>
</tr>
<tr>
<td>2101</td>
<td>The contents of S3C is sent to the local command line followed by carriage return. Note: No AT-Prefix required!</td>
</tr>
<tr>
<td>24XX</td>
<td>Start timers masked in XX.</td>
</tr>
<tr>
<td>25XX</td>
<td>Toggle timers masked in XX.</td>
</tr>
<tr>
<td>26XX</td>
<td>Stop timers masked in XX.</td>
</tr>
<tr>
<td>3XXX</td>
<td>Change I/O port to the LSBs.</td>
</tr>
<tr>
<td>4XXX</td>
<td>Change data direction of the I/O port to the LSBs.</td>
</tr>
<tr>
<td>53xx</td>
<td>Toggle I/Oxx</td>
</tr>
<tr>
<td>54xx</td>
<td>Flash I/Oxx (pull low) for 250ms</td>
</tr>
<tr>
<td>55xx</td>
<td>Set I/Oxx to 0</td>
</tr>
<tr>
<td>56xx</td>
<td>Set I/Oxx to 1</td>
</tr>
</tbody>
</table>

**Table 7. Built-in functions**
6 ETRX357 Power Consumption (Provisional data)

Typical values at 3.3V 25°C.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Current</td>
<td>30</td>
<td>mA</td>
<td></td>
<td></td>
<td>TX 8dBm</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>mA</td>
<td></td>
<td></td>
<td>TX 3dBm</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>mA</td>
<td></td>
<td></td>
<td>TX –1dBm</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>mA</td>
<td></td>
<td></td>
<td>TX –43dBm</td>
</tr>
</tbody>
</table>

Table 8: Power Consumption

<table>
<thead>
<tr>
<th>Mode</th>
<th>Router, COO</th>
<th>MED, SED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCU</td>
<td>Radio</td>
</tr>
<tr>
<td>0</td>
<td>Awake</td>
<td>Awake</td>
</tr>
<tr>
<td>1</td>
<td>Idle</td>
<td>Awake</td>
</tr>
<tr>
<td>2</td>
<td>Asleep</td>
<td>Asleep</td>
</tr>
<tr>
<td>3,4</td>
<td>Asleep</td>
<td>Asleep</td>
</tr>
</tbody>
</table>

Table 9: Averaged power consumption during operation

Notes:
- Sleep modes 1-4 should not be used on a router or coordinator, however it was found that mode 1 may work on a router with light to medium network traffic. Successful operation of a router in mode 1 cannot be guaranteed and needs to be evaluated carefully for each target application in case the additional energy saving is vital.
- Wakeup from modes 3 and 4 is only possible by external interrupt or reset. Make sure never to set the initial power mode (S3A) to mode 3 or 4 unless you want the device to always wake up into this mode.
- Modules in power mode 2-4 will not respond to commands at the serial port, so always make sure you have defined means to wake it up from these modes.
- Mode 4 is the same as mode 3 except that the device does not wait for any pending acknowledgements before sleeping.
- If no means of waking up from any of the power down modes has been defined and the module appears unresponsive the Telegesis factory default resetter can be used to reset the modules factory defaults via the serial port.
- In order to achieve ultra low power consumption of sub 1µA it is required to either define all I/Os to be outputs, or to pull all inputs to a defined level as floating input pins will increase the current consumption.

1 Approximate value, assuming the unit polls every second. If no polling and other timed actions are performed the power consumption can be as little as 1.5µA in this mode.
7 Notes on Energy Levels and LQI

7.1 Interpreting LQI

On the EM250 as well as the EM35x, the LQI is closely related to the SNR (signal noise ratio) but is calculated from the chip error rate. The graph below shows the relation between the SNR and the LQI reading on the EM250, which is the basis of the ETRX2 series of modules.

![EM250 SNR vs LQI](source: Ember)

From the LQI the stack calculates the cost for a particular link based on the following table. The cost is used by the stack to compare routes between nodes.

<table>
<thead>
<tr>
<th>Cost</th>
<th>LQI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>254 - 255</td>
</tr>
<tr>
<td>3</td>
<td>247 - 253</td>
</tr>
<tr>
<td>5</td>
<td>200 - 246</td>
</tr>
<tr>
<td>7</td>
<td>0 - 199</td>
</tr>
</tbody>
</table>

Table 10: LQI/Cost relationship
7.2 Interpreting RSSI Energy Levels

On the ETRX3 series modules the readings from AT+ESCAN represent the hexadecimal readings from the RSSI register of the EM250 or EM35x, offset by +256 to make it a positive number; alternatively read it as a signed 8-bit integer. AT+ESCAN returns the background radio power so that the quietest channel can be selected for a new network; if you want to measure the RSSI of incoming messages you must set bit C of register S10 (see page 79).

The EM35x SoCs calculate the RSSI over an 8-symbol period as well as at the end of a received packet. They utilize the RX gain settings and the output level of the ADC within its algorithm. 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).
8 Trademarks

All trademarks, registered trademarks and products names are the sole property of their respective owners.

9 Disclaimer

Product and Company names and logos referenced may either be trademarks or registered trademarks of their respective companies. We reserve the right to make modifications and/or improvements without prior notification. All information is correct at time of issue. Telegesis (UK) Ltd. does not convey any license under its patent rights or assume any responsibility for the use of the described product.

10 Contact Information

Website: www.telegesis.com
E-mail sales@telegesis.com

Telegesis (UK) Limited
Abbey Barn Business Centre
Abbey Barn Lane
High Wycombe
Bucks
HP10 9QQ
UK

Tel: +44 (0)1494 510199
Fax: +44 (0)5603 436999

11 References

Telegesis – www.telegesis.com

Silicon Labs - www.silabs.com/products/wireless/zigbee/Pages/default.aspx
# Appendix A. Forming a secure network

The default S-register settings lead to a network that is easy to set up and to which extra devices can easily be added, but it is not very secure. Not only can malicious devices intrude themselves, but there is also no safeguard against your devices joining a nearby network established by another user with similar Telegesis devices and register settings, and vice-versa. The use of a pre-defined Trust Centre Link Key can avoid all these problems. ZigBee PRO normally uses a Network Key common to all devices for general network traffic, and a Trust Centre Link Key common to all devices for key distribution. More specific applications such as Smart Energy systems can use a higher level of security such as a different link key for each point-to-point link.

The default register settings produce the following behaviour when a device establishes a PAN and another node joins:

1. ZC selects a random Link Key and Network Key
2. ZC sends Network Key to new node, unencrypted
3. ZC sends Link Key to new node, encrypted with Network Key

To create a secure network, use the following settings:

- Write your own Link Key into S09 on every device. If you do this off-line it can never be hacked
- Set bit 8 of register S0A on all devices that will join the PAN (Use Pre-Configured Trust Centre Link Key when joining)
- Set bits 4 and 2 of register S0A on the coordinator (Send Network key encrypted with the link key to nodes joining; Send Network key encrypted with the link key to nodes re-joining unsecured)
- (For simplicity, you can set bits 8, 4 and 2 of S0A on every device)

The joining procedure now becomes:

1. ZC selects a random Network Key
2. ZC sends the Network Key to new node, encrypted with Link Key

A sniffer can now no longer read the Network Key and use it to decrypt your messages, because the Link Key is never sent over the air.

You can choose a Network Key and write it into S08 in the coordinator, but there is not much point in doing this. The other devices ignore S08 as they receive the key from the coordinator, and if the key is ever updated over the air S08 no longer contains the current value. A pre-defined Network Key will be needed, however, if your device has to join a secure ZigBee 2006 network.

Bit A of S0A (When joining don’t ask for Trust Centre link key) is intended for use when your ZigBee PRO device needs to join a ZigBee 2006 network, since ZigBee 2006 does not use Link Keys. In normal use bit A is not set.
13 Appendix B. Bootloading new firmware

13.1 Bootloading through the serial port

The process of bootloading an application file through the serial port or over the air is described more fully in the ETRX357 Development Kit Product Manual. For convenience the main points are described here also, as they relate to Telegesis Terminal. Other terminal applications can be used provided they support the XMODEM protocol.

In order to upgrade the firmware of the ETRX35x 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.

Alternatively the button labelled “Bload” can be pressed on the development board whilst the reset button is pressed and released. The same effect is achieved by holding pin PA5 low while resetting or turning on the device.

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).

After pressing ‘Enter’ (sometimes twice), the bootloader menu will be shown in the terminal window as shown in Figure 2.

![Figure 2. Bootloader Menu](image)

Pressing ‘1’ initiates the upload of the new firmware and a number of ‘C’ characters will indicate that the ETRX35x is ready to receive data. Within 60 seconds, select Tools / Transfer File… and browse for the new firmware file.

Firmware files for the ETRX35x will be in the format ETRX35x_R3xxC.ebl. 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 3.
When the transfer has been completed successfully, press Enter again in order to return to the bootloader menu and type ‘2’ to run the downloaded application software. If the application software has a baudrate other than 115200bps, this will need to be changed to the application baudrate as described above – 19200 baud in the case of R3xx firmware.

Selecting option 2 instead of option 1 just runs the existing firmware without any changes. It is therefore safe to start the bootloader at any time, to check its version for example.

### 13.2 Bootloading over the air

The AT+PASSTHROUGH command enables you to send a bootload file to a remote router, provided it is only one hop away. If the bootload process fails, the remote device will have no valid firmware so it will listen for a new file on channel 13. Move the local device to channel 13 with the AT+CCHANGE command or by starting a new network with the S00 channel mask set to 0004, then use the AT+RECOVER command.

It is possible to bootload to an end device that is a child of the local device by starting the bootloader on the end device then using the recover process. The bootloader can be started by power-cycling or resetting the module with pin PA5 held low. However, after resetting the module it is necessary to send a <cr> character to the serial port which implies that the latter is available for use, so it may be easier to just use the AT+BLOAD command.