BitCloud™ SerialNet™

User Guide

8269A–MCU Wireless–11/09

Table of Contents

Section 1

Introduction

SerialNet is a manufacturer-specific profile developed on top of BitCloud C API. It offers control of embedded BitCloud stack through a serial interface using standardized AT-command set and requires no embedded API programming. Node's parameters can be easily accessed over-the-air without specifically dedicated protocol thus opening a way to network management and remote node control

The document presents the description of the SerialNet AT-command language

Section 2

References

2.1 Related Documents and References

1. RZUSBSTICK. Chapter 4, AVR2016: RZRAVEN Hardware User's Guide. www.atmel.com
2. ZigBit™ 2.4 GHz wireless modules. ATZB-24-A2/B0 datasheet. www.atmel.com/zigbit
3. ZigBit™ 2.4 GHz wireless modules. ATZB-A24-UFL/U0 datasheet. www.atmel.com/zigbit
4. ZigBit™ 700/800/900 MHz wireless modules. ATZB-900-B0 datasheet. www.atmel.com/zigbit
5. ZigBee™ Specification, Document 053474r17, October 2007.
6. Serial asynchronous automatic dialing and control. ITU-T Recommendation V.250, 05/99
7. International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5). Information Technology – 7-Bit Coded Character Set for Information Interchange, CCIT Recommendation T.50, 09/92.
8. General Structure of Signals of International Alphabet No. 5. Code for Character Oriented Data Transmission over Public Telephone Networks. ITU-T Recommendation V.4
9. IEEE Std. 802.15.4-2006 IEEE Standard for Information technology – Part 15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)
10. BitCloud ™ IEEE 802.15.4/ZigBee Software. www.atmel.com/bitcloud
11. AVR2051: BitCloud Stack Documentation. Part of BitCloud SDK.
12. 8-bit AVR Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash ATmega 640/V, ATmega 1280/V, ATmega 1281/V, ATmega 2560/V, ATmega 2561/V. www.atmel.com

2.2 Abbreviations and Acronyms

Table 2-1. Abbreviations and Acronyms

ARQ	Automatic Repeat-reQuest
ASCII	American Standard Code for Information Interchange
BS	Backspace character
CCITT	Consultative Committee on International Telephony and Telegraphy.
CR	Carriage Return
CRE	Coordinator / Router / End device (meaning any of those)
CTS	Clear To Send
DCE	Data Communication Equipment,
DTR	Data Terminal Ready
EEPROM	Electrically Erasable Programmable Read Only Memory

Table 2-1. Abbreviations and Acronyms

GPIO	General Purpose Input/Output
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunications Union
LED	Light Emitting Diode
LF	Line Feed character
LQI	Link Quality Indicator
LSB	Least Significant Bit
MAC	Medium Access Control (Sublayer)
MCU	MultiController Unit/Multi-Chip Unit
NWK	Network layer
OEM	Original Equipment Manufacturer
PAN	Personal Area Network
PHY	PHYsical Layer
PWM	Pulse Width Modulation
R	Read-only parameter
RSSI	Received Signal Strength Indicator
RTS	Request To Send
RW	Read-write parameter
RX	Receiver
TBD	To Be Defined
TX	Transmitter
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
ZDO	ZigBee Device Object

Section 3

Overview

SerilalNet is based on the AT-command protocol which is widely used in embedded networking systems due to its simplicity, textual parameter representation and inherent flexibility. This Chapter gives a brief introduction into the concept of SerialNet protocol, lists HW platforms SerialNet is available for and describes conventions used throughout the document.

3.1 Supported Platforms

The following hardware platforms are supported by SerialNet:

Table 3-1. Supported hardware platforms

Name in This Document	Platform (MCU + RF)	ZigBit Modules	Appropriate SDK
RZUSBSTICK	AT90USB1287 + AT86RF230 See [1.] on page 2-1	N/A	BitCloud for ATAVRRZRAVEN
ZigBit™	ATmega1281 + AT86RF230	ATZB-24-B0 (ZigBit B0); ATZB-24-A2 (ZigBit A2). See [2.] on page 2-1	BitCloud for ZDK
ZigBit™ Amp	ATmega1281 + AT86RF230	ATZB-A24-UFL (ZigBit Amp) [3.] on page 2-1	BitCloud for ZDK Amp
ZigBit™ 900	ATmega1281 + AT86RF212	ATZB-900-B0 (ZigBit 900) [4.] on page page 2-1	BitCloud for ZDK 900

Most of the SerialNet commands are HW-independent and can be executed on all supported platforms. However,a few commands either exhibit platform-specific behavior or are supported on particular HW platforms only. For such cases, command descriptions given in Chapter provide corresponding differences in the command functionality for various platforms. If no reference to platform is given in command description, then platform-independence is implied.

3.2 Conventions

The term *module* will be used throughout the document implying a supported platform (MCU + RF chip) controlled by a *host* equipment using AT-commands.

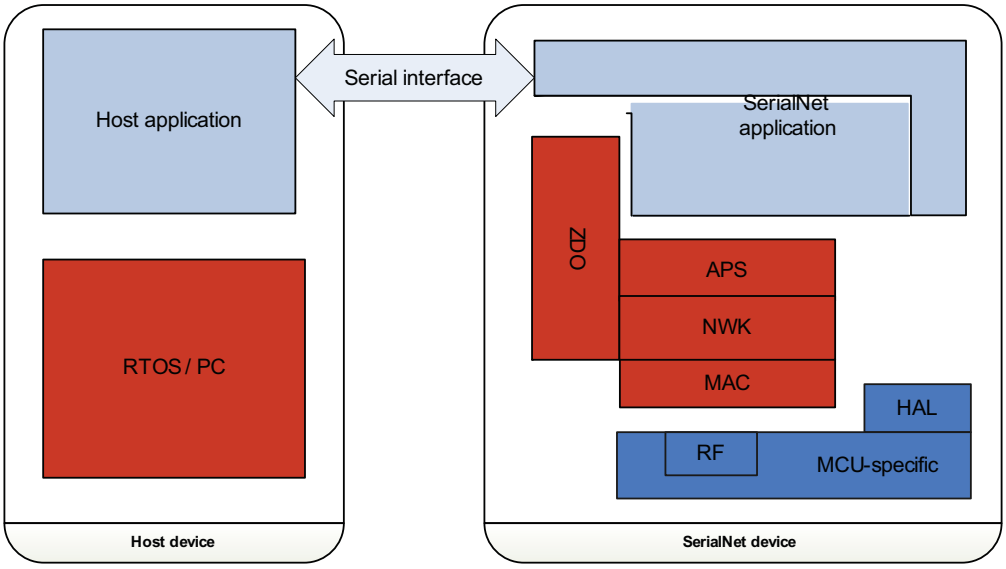
The term *node* will be used in reference to the device's role in the network (End device, Router or Coordinator).

To be distinguished from the rest, the definitions of commands directed to the module are denoted in Courier while the module responses are given in **Bold Courier** font. Angle brackets enclose mandatory parameters. Square brackets contain optional parameters.

3.3 Architecture Overview

SerialNet application is developed on top of Atmel's BitCloud ZigBee PRO-certified stack, see - Step 10.on page 2-1. It provides an easy-to-use control over ZigBee PRO networking functionality that is accessible for the host device through serial connection using an extensive set of AT-commands in ASCII format. SerialNet device executes received requests and responds to the host. Table 3-1 illustrates the basic architecture.

Figure 3-1. SerialNet usage scheme

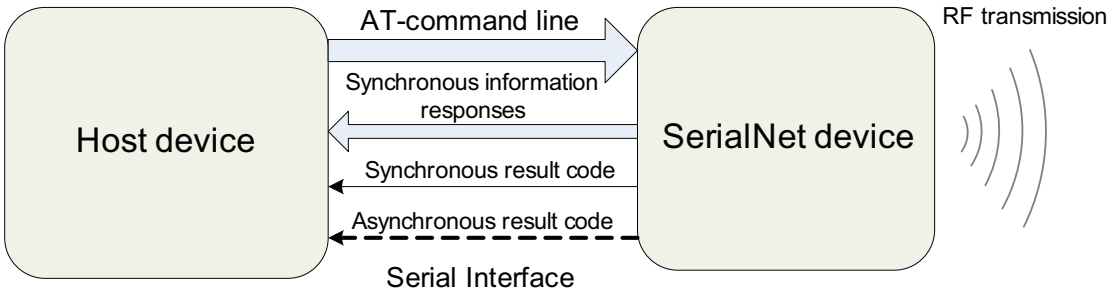


An important feature of SerialNet is the capability to request execution of particular function over the air via ATR command (see Table 5-8 on page 5-8). It allows transferring the AT-command to the remote node in the network, executing it there and redirecting the execution output to the originator. Thus, the remote node can be monitored, commissioned and the corresponding parameters can be set.

3.3.1 Protocol Principles

SerialNet supports an extensive set of AT-commands that provide full control over different functionality of the module. Read/write commands to S-registers can be used to access device and network parameters. In many cases AT-command functionality can be duplicated by certain S-register to reduce overhead of the serial protocol. The basic principle of SerialNet protocol is illustrated in Table 3-2.

Figure 3-2. SerialNet command executions



The host device shall transmit a command line prefixed by the "AT" string followed by the chained SerialNet commands to be executed consecutively. Upon successful execution of each command in the sequence corresponding information response is returned to the host device in an easily recognizable string format. The final result of the command line execution is indicated by the result code. In case of any command executed incorrectly, the command sequence is interrupted and the `ERROR` result code is returned. Result code is OK if all commands in the sequence were executed successfully.

Each command in a sequence may have different syntax, depending on whether it is used to execute an action, to read or to write parameter(s) or to test valid parameter range. An example illustrating different command and response types is provided in Table 3-2.

Table 3-2. At command string execution

	Command/Response	Comment
Command to device	ATE1V1+WTXPWR=-4+WLQI2+WRSSI2S22?	Turn echo on (E1), enable verbose response, set Tx power level to -4 dBm, request for LQI and RSSI for link with node 2, request for active channel
Information responses	**+WLQI:254**	LQI value is 254
	+WRSSI:-80	RSSI is -80 dBm
	B	Node is operating on channel 0x0B
Result code	**OK**	Execution is completed successfully

More complex examples are provided in the section “Examples” on page 5-6.

In addition to synchronous result codes indicating command execution status, SerialNet device upon specific event s can send to the host device asynchronous result codes. The full list of both verbose and numeric forms of the result codes can be found in “Parameter Persistence” on page 5-4.

“AT Commands” on page 5-1 summarizes the basic specifications of AT-commands grouped into functional categories while detailed definition for each command is given in Chapter 6.

“S-registers” on page 5-6 is a functional representation of S-registers with the corresponding AT-commands.

Section 4

Getting Started

4.1 Connection with Board

The supported platform (see"Supported Platforms" on page 3-1) shall be first programmed (via JTAG, USB or RS-232) with the SerialNet firmware version for the corresponding platform. After that it shall be connected to a host device (a PC, MCU, etc.) using USB or RS-232 interface. To start communication the host device shall configure its serial port with default SerialNet parameters:

Table 4-1. Default Serial Net Parameters

Baud rate	38400
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None

Note that these parameters can be modified for SerialNet device and saved in persistent memory using corresponding commands described in .

If a PC is the host, then HyperTerminal software from the standard Windows package can be used to communicate with SerialNet device. To check the connection, AT should be entered on the terminal window followed by <Enter>. If the board responds with OK, then communication between host and SerialNet devices is established successfully.

The section "Examples" on page 5-6 includes examples showing how a SerialNet device can be configured for networking operations, data exchange and remote control.

Section 5

Command Summary

5.1 AT Commands

The AT-commands implemented in SerialNet fall into the following categories:

- Network configuration and management
- Data transmission
- Power management
- Generic control
- Host interface control
- Hardware control
- Remote management.

Table 5 1 provides a full list of SerialNet commands with information about supporting node roles, syntaxes, corresponding S-registers (if any), persistence and references to the detailed command description in Chapter 5.4.5.

Table 5-1. Command Summary

Function	Node type (C/R/E)	S-register	Action syntax	Parameter set syntax	Parameter read syntax	Parameter test syntax	Command	Persistence	Reference
Networking parameters									
Extended PAN ID	CRE	20, 21		x	x	x	+WPANID	x	6.2.1
Active channel	CRE	22			x		+WCHAN		6.2.2
Channel mask	CRE	23		x	x	x	+WCHMASK	x	6.2.3
Channel page	CRE	25		x	x	x	+WCHPAGE	x	6.2.4
Automatic networking	CRE	24		x	x	x	+WAUTONET	x	6.2.5
Node role	CRE	33		x	x	x	+WROLE	x	6.2.6
Device extended address	CRE			x	x		**+GSN or I4**		6.2.7
Node short address	CRE	55		x	x	x	**+WSRC**	**x**	6.2.8
Network management									
Start/Join to network	CRE		x				**+WJOIN**		6.3.1

Table 5-1. Command Summary

Leave the network	CRE		x				**+WLEAVE**		6.3.2
Request for networking status	CRE		x				+WNWK		6.3.3
Request for parent address	E				x		+WPARENT		6.3.4
Request for children addresses	CR				x		+WCHILDREN		6.3.5
Request for a number of neighbor nodes	CRE				x		+WNBSIZE		6.3.6
Request for neighbors' information	CRE				x		+WNB		6.3.7
Network addressing mode	CRE	30		x	x		S30		6.3.8
Request for LQI	CRE		x				+WLQI		6.3.9
Request for RSSI	CRE		x				+WRSSI		6.3.10
Power management									
End device sleep parameters	CRE	31, 32		x	x	x	+WPWR	x	6.5.1
Force to sleep	E		x				+WSLEEP		6.5.2
Tx power level	CRE	34		x	x	x	+WTXPWR	x	6.5.3
Data transmission									
Send data to specific node	CRE		x				D		6.4.2
Send broadcast data	CRE		x				DU		6.4.3
Send S-register value to specific node	CRE		x				DS		6.4.4
Ping the node	CRE		x				+WPING		6.4.5
Indirect poll rate	CRE	37		x	x	x	+WSYNCPRD		6.4.6
Data delivery time-out	CRE	51			x		+WTIMEOUT		6.4.7
Repetition count	CRE	52			x		+WRETRY		6.4.8
Data transmission waiting time-out	CRE	53		x	x	x	+WWAIT	x	6.4.9
Generic control									
Warm reset	CRE		x				Z		6.6.1
Help	CRE		x				&H		6.6.2
Display parameters and S-register values	CRE		x				%H		6.6.3
Display product identification information	CRE		x				I, I0		6.6.4
Request for Manufacturer Identification	CRE		x				`+GMI or I1`		6.6.5
Request for Model Identification	CRE		x				`+GMM or I2`		6.6.6
Request for hardware/software revision Identification	CRE		x				`+GMR or I3`		6.6.7
Set to factory-defined configuration	CRE		x				&F		6.6.8
Host interface commands									
Termination character	CRE	3		x	x		S3	x	6.7.1

Table 5-1. Command Summary

Response formatting character	CRE	4		x	x		S4	x	6.7.2
Command editing character	CRE	5		x	x		S5	x	6.7.3
Command echo	CRE		x				E	x	6.7.4
Result code suppression	CRE		x				Q	x	6.7.5
Response format	CRE		x				V	x	6.7.6
Result code selection	CRE		x				X	x	6.7.7
Serial port communication rate	CRE			x	x	x	+IPR	x	6.7.8
Serial port flow control	CRE			x	x	x	+IFC	x	6.7.9
DTR behavior	CRE	50	x				&D	x	6.7.10
Request for the latest result code	CRE	0			x		S0		6.7.11
Hardware control									
GPIO configuration	CRE	120 … 128		x	x		S120…S128	x	6.8.1
GPIO	CRE	130 … 138		x	x		S130…S138		6.8.2
A/D configuration	CRE	100		x	x		S100	x	6.8.3
A/D	CRE	101 … 104			x		S101…S104		6.8.4
PWM configuration	CRE	140, 141, 142		x	x		S140, S141, S142		6.8.5
PWM frequency control	CRE	143, 144, 145		x	x		S143, S144, S145		6.8.6
PWM duty cycle control	CRE	146, 147 148		x	x		S146, S147, S148		6.8.7
Remote management									
Set a password	CRE		x				+WPASSWORD	x	6.9.1
Remote execution of AT command	CRE		x				R		6.9.2

Note: ***1.*** The second column contains roles of nodes to which a given command is applicable. C stands for Coordinator, R for Router, and E for End device.

5.1.1 Parameter Persistence

In Table 5-1 many parameters associated with AT-commands are indicated as persistent. This means that their values are stored in persistent memory of MCU and in contrast to non-persistent parameters they will not be set to default configuration upon device reset.

However, value assigned to a persistent parameter by corresponding AT command is not written to the persistent memory right away. Instead it is applied to SerialNet operation but is kept in RAM. SerialNet periodically (with 5 minutes interval) verifies whether values of persistent parameters in EEPROM match their actual values in RAM. If differences are detected, then corresponding values in EEPROM are updated. For platforms with warm reset command support (see Table 6-41 on page 6-26) persistent

parameters in EEPROM are updated to actual values (if necessary) automatically upon ATZ command execution.

Upon device reset SerialNet assigns persistent parameters to their values stored in EEPROM. If a parameter value has not been transferred from RAM to EEPROM then the old EEPROM value will be used.

5.2 Result Codes

Result codes appear either synchronously in response to a command or, asynchronously, due to the specific events in the network or on a SerialNet device. See detailed description of result codes in "Device Responses" on page 6-7. Table 5-2 provides both verbose and numeric forms for available result codes.

Table 5-2. Result codes

Verbose Code	Numeric Code	Parameters	Description
OK	**0**	None	Command is executed successfully
ERROR	**4**	None	Error occurred during command execution
DATA	**8**	**<addr>,<bcast>,<length>:<data>**	Indicates data reception from a remote node. **addr** is a short (network) address of a source node data is originating from **bcast** is set to 1 if data is sent by broadcast transmission, otherwise it is set to 0 **length** is a length of the **<data>** field **data** is byte sequence of received data Note: +WPING command (see Table 6-33 on page 6-21) results in the following code on the destination node: **DATA <addr>,0,0:**
EVENT	7	**:<text>**	text is a text specifying an event.
		:JOINED	Indicates that the node has joined to the network Note: Event is returned in auto network mode only and not after +WJOIN command.
		:LOST	Indicates that the node has lost connection to the network (i.e. to its current parent) Note: Event can occur on end device nodes only and is not returned after +WLEAVE.

Table 5-2. Result codes

		:CHILD_JOINED <addr>	Indicates to the node that device with extended address **<addr>** has just joined to it as a child
		:CHILD_LOST <addr>	Indicates to the node that its child end device with extended address **<addr>** has disconnected from the node. Note: Event occurs when child end device switches to a new parent, when it leaves the network using +WLEAVE command or when it is not accessible (powered off, no link, etc.) for *3*(sleep_interval + sync_period)* as configured on parent device by +WPWR and +WSYNCPRD commands.
		:CALIBR	Indicates that the device has successfully calibrated its internal clock after encountering errors on serial interface.

5.3 S-registers

An extensive set of S-registers available in SerialNet provides easy read/write access to device and networking parameters. In many cases AT-command functionality can be duplicated by certain S-register to reduce overhead of the serial ASCII protocol.

Table 5-3. S-Registers

Parameter	Acceptable Operations (R/RW)	S-register	Command Reference
The latest result code	R	S0	6.7.11
Termination character	RW	S3	6.7.1
Response formatting character	RW	S4	6.7.2
Command editing character	RW	S5	6.7.3
PAN ID	RW	S21, S20	6.2.1
Active channel	R	S22	6.2.2
Channel mask	RW	S23	6.2.3
Automatic networking	RW	S24	6.2.5
Channel page	RW	S25	6.2.4
Network addressing mode	RW	S30	6.3.8
Power management	RW	S31, S32	6.5.1
Node role	RW	S33	6.2.6
Tx power level	RW	S34	6.5.3
Indirect poll rate	RW	S37	6.4.6
DTR behavior	RW	S50	6.7.10
Data delivery time-out	R	S51	6.4.7
Repetition count	R	S52	6.4.8

Table 5-3. S-Registers

Data transmission waiting time-out	RW	S53	6.4.9
Own network address	RW	S55	6.2.8
A/D configuration	RW	S100	6.8.3
A/D	R	S101…S104	6.8.4
GPIO configuration	RW	S120…S128	6.8.1
GPIO	RW	S130…S138	6.8.2
PWM configuration	RW	S140, S141, S142	6.8.5
PWM frequency control	RW	S143, S144, S145	6.8.6
PWM duty cycle control	RW	S146, S147, S148	6.8.7

5.4 Examples

The examples given below show usage of AT-commands to control the SerialNet devices and are valid for all supported platforms listed in "Supported Platforms" on page 3-1.

5.4.1 Prepare nodes for networking

The following examples require at least 2 nodes. The first step is configuring network parameters. One of the nodes should function as a coordinator and others could be routers or end devices. It is important that all nodes have different extended (MAC) and short (NWK) addresses. Coordinator node shall have short address 0, and all other nodes shall have non-zero addresses.

Note: Selection of particular addresses is application dependent. It should be done only the first time during the manufacturing process of initial installation.

Table 5-4. Network coordinator

Command/Response	Comment
`ATX`	set a node to transmit **EVENT** and **DATA** to a host
OK	
`AT+GSN=1`	set extended address for the node
OK	
`AT+WPANID=1620`	set node's extended PAN ID
OK	
`AT+WCHMASK=100000`	set node's channel mask (this one enables channel 0x14 only)
OK	
`AT+WROLE=0 +WSRC=0`	set coordinator role and short address to 0x0000
OK	
`AT+WJOIN`	perform network start
`OK`	result code for successful network start

If the node indicates ERROR, that means the embedded software does not support coordinator function and cannot be configured in such a way. In this case, try checking the coordinator support on other nodes using AT+WROLE? command, as described in Table 6-17 on page 6-11.

Then set configure another device to be a router node:

Table 5-5. Network router

Command/Response	Comment
ATX	set a node to transmit **EVENT** and **DATA** to a host
OK	
AT+GSN=2	set extended address for the node
OK	
AT+WPANID=1620	set node's extended PAN ID
OK	
AT+WCHMASK=100000	Set node's channel mask (this one enables channel 0x14 only)
OK	
AT+WROLE=1 +WSRC=55	set router role, short address equal to 0x0055
OK	
AT+WJOIN	perform network join
OK	indication for router having joined the network

5.4.2 Checking network status and basic data transmission

Now we can easily verify networking status on both devices by AT+WNWK command and perform data exchange between them. For example on coordinator:

Table 5-6. Verify networking status on coordinator

Command/Response	Comment
AT+WNWK	request networking status
OK	means that the node is in the network
AT+WWAIT=3000 **OK** ATD55 HELLO **OK**	set 3 sec time-out to wait for input and send HELLO word to the node with short address 55

Simultaneously, HELLO word will appear on the terminal connected to the router in form of DATA event:

Table 5-7. Verify networking status on router terminal

Command/Response	Comment
DATA 0000,0,5:HELLO	data (5 bytes) came from device with address 0 by unicast request

5.4.3 Remote Extension

ATR command provides mechanism for AT-command execution on a remote node with command response redirection to the originator. Thus it allows remote monitoring and configuration over the air.

The example below demonstrates how to execute AT-commands on the router device remotely using ATR command on the coordinator:

Table 5-8. Remote execution of AT-commands on the router

Command/Response	Comment
ATR55,0,+WROLE?+GSN? **+WROLE:1** **+GSN:0000000000000055** **OK**	get node role and extended address from the router
ATR55,0,+GMI? **+GMI:ATMEL** **OK**	get model number from the router
ATR55,0,+WAUTONET=1S30=1 **OK**	set autonet mode and command addressing mode

5.4.4 End Device Power Control

This example demonstrates how to configure an end device node with certain duty cycle, perform network join and deliver data to an end device:

Table 5-9. Configure end device node with duty cycle

Command/Response	Comment
ATX **OK** AT+GSN=3 **OK** AT+WROLE=2 +WSRC=56 **OK** AT+WPANID=1620+WCHMASK=100000 **OK** AT+IFC=2,2 **OK**	set a node to transmit **EVENT** and **DATA** to a host set extended (MAC) address for the node set the board as end device with short address 0x0056 set extended PAN ID and channel mask (channel 0x14) for this node configure RTS and CTS line modes for end device flow control. Reconfigure flow control on the host accordingly. (E.g. select Hardware mode for Flow Control in Hyper Terminal)
AT+WPWR=100,100 **OK** AT+WPWR? **+WPWR:100,100** **OK**	set duty cycle 10 sec sleep / 1 sec active verify that the duty cycle is accepted successfully
AT+WJOIN **OK**	perform network join result code indicating successful network join for the end device

Now, the data intended for the end device can be sent from the coordinator:

Table 5-10. Test data from the coordinator

Command/Response	Comment

`ATD56,0,4`
`test`
`OK`
	send test data from coordinator for the end device staying in a sleep mode

In active state end device periodically polls its parent for buffered data with interval configured by `+WSYNCPRD` parameter. In the given example it retrieves the test frame:

Table 5-11. Polling of buffered data from parent

Command/Response	Comment
`DATA 0000,0,4:test`	the `test` word is received by end device after wake up

5.4.5 Control of LED and DIP switches

The example below is valid only for MeshBean2 development boards. Mapping of I/O pins of the ZigBit module and their functions on the MeshBean2 boards is summarized in the table below

Table 5-12. GPIO Pins Summary

Component	I/O pin	Description
LED1	GPIO0	output, 1 means LED on
LED2	GPIO1	output, 1 means LED on
LED3	GPIO2	output, 1 means LED on
SW4:1	GPIO3	input (no pull-up on the board), ON – logical zero
SW4:2	GPIO4	input (no pull-up on the board), ON – logical zero
SW4:3	GPIO5	input (no pull-up on the board), ON – logical zero
	GPIO6	reserved for MeshBean2 sensor interfaces
	GPIO7	reserved for MeshBean2 sensor interfaces
	GPIO8	reserved for MeshBean2 sensor interfaces

Initially, set DIP-switches physically as SW4:1 to OFF, SW4:2 and SW4:3 to ON, and, next, configure I/O pins via command:

Table 5-13. Configure I/O pins

Command/Response	Comment
`ATS120=3 S121=3 S122=3`	configure GPIO0, GPIO1, GPIO2 for output
`OK`	
`ATS123=1 S124=1 S125=1`	configure GPIO3, GPIO4, GPIO5 for input and turn on internal pull-up
`OK`	

Afterwards, it is possible to control LEDs and to obtain status of DIP-switches using corresponding S-registers:

Table 5-14. Control LEDs and check DIP-switches

Command/Response	Comment
`ATS130=1 S131=0 S132=1`	turn on LED1 and LED3
OK	
`ATS133? S134? S135?`	
1	SW4:1 is in the OFF state
0	SW4:2 is in the ON state
0	SW4:3 is in the ON state
OK	

Section 6

Command Description

6.1 Protocol General Description

6.1.1 Character Formatting and Data Rates

Data transmitted between the host and the module over serial interface conforms to the requirements for start-stop data transmission specified in the ITU-T Recommendation V.4 Step 8. in page 2-1. Parity is even, odd or not used. Each character has at least one complete stop bit. The module accepts commands using any combination of parity and stop bits supported. These include, at least, the following combinations, each of which consists of up to ten bits (including the start bit):

- 7 data bits, even parity, 1 stop bit
- 7 data bits, odd parity, 1 stop bit
- 8 data bits, no parity, 1 stop bit.

Both the host and the module are able to accept commands at 1200 bit/s at least. Particular character formatting and the data rate can be changed using appropriate AT-commands - see Table 6-59 on page 6-37), Table 6-60 on page 6-38 and Table 6.7.6 on page 6-35. The host has the means to select explicitly data rate and character formatting according to the specifications above.

6.1.2 Alphabet

For any information exchange between the module and the host the T.50 International Alphabet 5 (IA5) is used - see Step 7.in "Related Documents and References" on page 2-1. Only the seven low-order bits of each character are significant, any of eighth or higher-order bit(s), if present, are ignored for the purpose of identifying commands and parameters. Lower-case characters (hex codes `0x61` through `0x7A`) are considered identical to their upper-case equivalents (hex codes `0x41` through `0x5A`) when received by the module from the host. Result codes from the module, which are particularly defined, are specified in upper case.

6.1.3 Basic Command-Line Operations

Command line editing, echoing and repeating are done in accordance with the Clauses 5.2.2, 5.2.3 and 5.2.4 of the Recommendation V.250. The description below follows the statements introduced in Step 6. in "Related Documents and References" on page 2-1.

The module may echo the characters received from the host back to the host, depending on the setting of the E command (see Table 6.7.4 on page 6-34). If so enabled, the characters received from the host are echoed at the same rate, parity, and format as those received.

The module checks on the characters coming from the host first, to see if they match the termination character S3 (see Table 6-51 on page 6-32). Next, it checks the editing character (S5, see Table 6-53 on page 6-33), before considering any other character. That insures the characters will be properly recognized even though they were set to values which the module uses for other purposes. If S3 and S5 are

set to the same value, the character checked will be treated as a character matching S3 (as S3 is checked before S5).

The character defined by S5 parameter (by default, it is backspace character - BS [hex code 0x08], see Table 6-53 on page 6-33) is intended to be interpreted as a request from the host to the module to delete the previous character. Any control characters (hex codes 0x00 through 0x1F, inclusive) that remain in command line after receiving the termination character will be ignored by the module.

Once the module finds the termination character, it starts processing the command line. Command line starts with AT (characters 0x41, 0x54) and should contain a sequence of commands in the following syntax formats:

Table 6-1. Command Syntax Formats

Command	Syntax
Action command	<command>[<value>]
Parameter set command	<command>=<value>
Parameter read command	<command>?
Testing a range of valid values	<command>=?

Where <command> is one of the following:

- a single character
- '&' character (0x26) followed by a single character
- '%' character (0x25) followed by a single character
- '+' character followed by a string of characters.

The characters allowed to be used in <command> should be taken from the T.50 International Alphabet 5. The first three of the command cases above are referred to as basic commands; they may be of the action command syntax only. Commands beginning with the plus sign are known as the extended syntax commands and can fit all the syntax rules depending on their type. Typically, a command that supports the parameter set syntax also supports the testing syntax.

A command (with associated parameters, if any) may be followed by additional commands in the same command line without using any delimiting character. Some commands may cause the remainder of the command line being ignored (the D command, see Table 6-30 on page 6-20, for instance).

If command line is started with the 'A/' or 'a/' prefix (hex codes 0x41, 0x2F or 0x61,0x2F), the module repeats immediately the execution of the preceding command line. No editing is possible, and no termination character is required. With this mechanism, a command line may be repeated as much as desired.

6.1.4 Parameter Values

Parameters may take either a single value, or multiple (compound) values. A compound value consists of any combination of numeric values (as defined in the description of the action or parameter command). The comma character (hex code 0x2C) is included as a separator, before the second and all subsequent values in the compound value. If a value is not specified as missed (i.e. defaults assumed), the required comma separator should be specified; however, trailing comma characters may be omitted if all the associated values are also omitted.

Note: When any of optional parameters is misused in a command, the command would be performed as if the parameter was be omitted. That parameter would be further treated as if the

other subsequent command were input, probably causing an ERROR message. To avoid confusions follow the command syntax.

Actions may have more than one of associated sub-parameters, and parameters may have more than one value. These are known as "compound values", and their treatment is the same in both the action command syntax and the parameter command syntax.

Each value may be either decimal or hexadecimal number. The choice depends on a particular command and hexadecimal numbers if they are not preceded with '0x'. Hexadecimal numbers can represent 16-bit, 32-bit, 64-bit and 128-bit values.

Decimal numeric constants consist of a sequence of one or more of the characters '0' (hex code 0x30) through '9' (hex code 0x39), inclusive, and can be preceded by minus "-". The most significant digit is specified first. The leading '0' characters will be ignored.

Hexadecimal numbers consist of characters "0" through "9" and "A" through "F", inclusive. Minus sign is not allowed. The leading '0' characters will be ignored. To prevent misinterpretation of hexadecimal numbers in cases when the command containing them is not the last in the AT string, it is strongly recommended to add the leading zeroes. So, if a parameter is 32-bit long, it would be 8 characters long, if it is a 64-bit number, it would contain 16 characters and so on.

As a special case, string constant appears in R command (see Table 6-71 on page 6-47) only. Then, it is just a sequence of displayable IA5 characters, each in the range of 0x20 to 0x7F, inclusive.

6.1.5 Command Types

A command type may be one of the following:

- An action command
- A parameter command
- An S-registers command.

Parameters may be defined as "Read-only" (R) or "Read/Write" (RW). "Read-only" parameters are used to provide the host with the status or identifying information, but are not set by the host. Attempting to set such a parameter will result in an error. In some cases (depending on the particular parameter), the module may ignore any attempt to set the value for such parameter rather than respond with the ERROR result code. "Read-only" parameters may be read and tested.

"Read/Write" parameters may be set by the host in order to store a value or values for later use. "Read/Write" parameters may be set, read, and tested.

If <command> is not recognized, the module generates the ERROR result code and stops processing of the command line. The ERROR result code is also generated if: a sub-parameter is specified for an action that does not imply using sub-parameters; too many sub-parameters are specified; a mandatory sub-parameter is not specified; a value is specified of the wrong type; or if a value is specified that is not within the supported range.

Some commands allow omitting a value. If a command does omit one, then it should be immediately followed by another command (or the termination character) in the command line. The '0' value is assumed unless otherwise specified in the <command> description. If the <command> does not expect a value but the value is present, the ERROR code is generated.

6.1.6 Action Command Syntax

The format of the action commands, except for the D, DU and S commands, is as follows:

Table 6-2. Action command syntax

Command	AT Syntax
Action command with no parameters used	<command>
Action command with one or more sub-parameters used	<command>[<value>]

The value may be either a single value parameter or a compound value parameter as described in 6.1.4. Some commands may have no parameters at all. Expected value is noted in the description of a particular command.

Table 6-3. Example of action command

Command/Response	Comment
AT+WLEAVE	Leave the network
OK	Result code
ATX2	2 - Disables events and data indications
OK	Result code

6.1.7 Parameter Set Command Syntax

The following syntax is used for a parameter set command:

Table 6-4. Parameter set command syntax

Command	AT Syntax
Parameter set command	<command>=[<value>]

If the named parameter is implemented in the module, all the mandatory values are specified, and all values are valid according to the definition of the parameter, the specified values should be stored. If <command> is not recognized, one or more of mandatory values are omitted, or one or more values are of wrong type or beyond the valid range, the module generates the ERROR result code and terminates processing of the command line. ERROR is also generated if too many values are specified. In case of error, the previous values of the parameter are unaffected:

Table 6-5. Example of parameter set command

Command/Response	Comment
AT+WWAIT=4000	Set parameter +WWAIT
OK	Result code

6.1.8 Parameter Read Command Syntax

The host may determine current value or values stored in a parameter by using the following syntax:

The following syntaxes are used

Table 6-6. Parameter read command syntax

Command	AT Syntax
Parameter read command	<command>?

If the named parameter is implemented, its current values are sent to the host in an information text response. The format of this response is described in definition of the parameter. Generally, the response string is beginning with <command> followed by ':' character and the values represented in the same form, in which they would be generated by the host in a parameter set command. If multiple values are supported, they will generally be separated by commas, as in a parameter set command. For example:

Table 6-7. Example of parameter read command syntax

Command/Response	Comment
AT+WRETRY?	Request for parameter +WRETRY
+WRETRY:3	Returned value
OK	Result code

6.1.9 Parameter Test Command Syntax

Table 6-8. Parameter test command syntax

Command	AT Syntax
Parameter test command	<command>=?

If the module does not recognize the indicated <command>, it returns the ERROR result code and terminates processing of the command line. If the module does recognize the parameter name, it returns an information text response to the host, followed by the OK result code. The information text response will indicate the values supported by the module for each of sub-parameters, and, possibly, additional information. The format of this information text response is defined for each parameter. See"Information Text Formats" on page 6-7 for the general formats for specification of sets and ranges of numeric values. Generally, an information text response is started with a <command> followed by ':'.

When an action/parameter accepts a single numeric sub-parameter, or the parameter accepts only one numeric value, the set of supported values may be presented in an information text as an ordered list of values. The list should be preceded by left parenthesis '(', (hex code 0x28), and closed by right parenthesis ')', (hex code 0x29). If that very single value is supported, it should appear in parentheses. If more than one value is supported, then the values may be listed individually, separated by comma characters (hex code 0x2C). When a continuous range of values is supported, the values appear in form of the first value in the range, and the last value in the range, both separated by a hyphen character (hex code 0x2D). The specification of single values and value ranges may be alternated within a single information text. Nevertheless, the supported values should be indicated in an ascending order. For example, the following are some examples of value range indications:

Table 6-9. Value range indications

Value	Comment
(0)	Only the 0 value is supported.
(1,2,3)	The values 1, 2, and 3 are supported.
(1-3)	The values 1 through 3 are supported.
(0,4,5,6,9,11,12)	The several listed values are supported.
(0,4-6,9,11-12)	Alternative expression of the previous list.

The value may be either a single value parameter or a compound value parameter as described in 6.1.4. Some commands may have no parameters at all. Expected value is noted in the description of a particular command.

Table 6-10. Example of parameter test command syntax

Command/Response	Comment
`AT+WSRC=?`	Request for valid range of the short address
`+WSRC:(0000-FFF7)`	Returned value
`OK`	Result code

When an action/parameter accepts more than one sub-parameter, or the parameter accepts more than one value, the set of supported values may be presented as a list of the parenthetically-enclosed value range strings, separated by commas. For example, the information text in response to testing an action that accepts three sub-parameters, and supports various ranges for each of them, could appear as follows:

```
(0),(1-3),(0,4-6,9,11-12)
```

This indicates that the first sub-parameter accepts only the 0 value, the second accepts any value from `1` through `3`, inclusively, and the third sub-parameter accepts any of the values `0, 4, 5, 6, 9, 11` or `12`.

6.1.10 S-registers

S-registers represent a group of numerical parameters that can be addressed in a special syntax. Each S-register has its own address and value. Some S-registers are standardized by the V.250 recommendations and are used in the module. Some of the S-registers are non-standard defined specifically by the SerialNet software.

AT-commands that begin with the 'S' character are allowed for S-register access. These differ from other AT-commands in some respects. The number following the 'S' character indicates the referenced "register number". If the number is not recognized as a valid register number (register is omitted), the `ERROR` result code is generated.

Immediately following that number, either a `'?'` or `'='` character (hex codes `0x3F` or `0x3D`, respectively) should appear. `'?'` is used to read the current value of the indicated S-parameter. `'='` is used to set the S-parameter to a new value.

Table 6-11. S-Registers

Command	AT Syntax
Reading the S-register	`S<parameter_number>?`
Setting the S-register	`S<parameter_number>=[<value>]`

If the '=' character is used, the new value to be stored in the S-parameter is specified in decimal form following the '=' character. If no value is given (i.e. the end of the command line occurs or the next command follows immediately), the corresponding S-parameter will be set to 0. The ranges of acceptable values are given in description of each S-register.

"S-registers" on page 6-6 gives functional representation of S-registers associated to the commands.

6.1.11 Device Responses

There are two types of responses that may be generated by the module:

- information responses
- result codes.

Basically, any information response consists of three parts: header, text, and trailer. The characters generated in header are determined by user's setting (see V command, Table 6-56 on page 6-35). Trailer consists of two characters, namely the ordinal value of parameter S3 followed by the ordinal value of parameter S4. Information text may contain multiple lines, and the text may include any formatting characters to improve readability.

A result code consists of three parts: header, the result text, and trailer. The characters to be generated in header and trailer are determined by user's setting (see the V command, Table 6-56 on page 6-35). The result text may be generated as a number or a string, depending on the user-selected setting (see the V command, Table 6-56 on page 6-35).

There are two general types of result codes: final and unsolicited.

Final result codes (**OK/ERROR**) indicate completion of the module action and readiness to accept new commands from the host. Unsolicited result codes (such as **DATA**) may not be directly associated with the issuance of a command from the host. They indicate the occurrence of another **EVENT** causing them.

Command X (see Table 6-58 on page 6-36) controls the generation of result codes, while command Q (see Table 6-55 on page 6-35) results in their total suppression.

"Parameter Persistence" on page 5-3 summarizes representations the result codes are in both verbose and numeric forms with the corresponding parameter(s), if any, and their brief description. Each command description itself refers to the specific result codes that may be generated in relation to the command and the circumstances, under which they may be issued.

6.1.12 Information Text Formats

In general, the particular format of information text returned by extended syntax commands will be specified in the command definition.

Note that the module may insert intermediate <CR> characters in very long information text responses, in order to avoid overflow in the host receive buffers. If intermediate <CR> characters are included, the module does not include the character sequences "**0** <CR>" (0x30, 0x0D) or "**OK**<CR>" (0x4F, 0x4B, 0x0D), so that the host can avoid false detection of the end of these information text responses.

6.2 Networking Parameters

This section describes SerialNet commands associated with networking parameters. Most of the parameters shall be set on each device according to desired network characteristics prior to executing network start/join procedure. Not that if default setting or persistent value from the EEPROM (see Section 5.1.1 on page 5-3) already has desired value for a network parameter, there is no need to assign it explicitly again prior to network start/join.

There is also a number of hard-coded parameters that cannot be changed by AT-commands but which have direct impact on possible network topology and performance:

```
CS_NEIB_TABLE_SIZE = 10
CS_MAX_CHILDREN_AMOUNT = 8
```

```
CS_MAX_CHILDREN_ROUTER_AMOUNT = 3
CS_MAX_NETWORK_DEPTH = 5
CS_ROUTE_TABLE_SIZE = 10
CS_ROUTE_DISCOVERY_TABLE_SIZE = 7
CS_ADDRESS_MAP_TABLE_SIZE = 10
CS_BTT_SIZE = 16
```

Their values shall be taken into account during network establishment and operation. Details about each parameter can be found in BitCloud Stack Documentation, see Step 11. in "Related Documents and References" on page 2-1.

6.2.1 "+WPANID" - Set/Get extended PAN ID

Table 6-12. "+WPANID" - Set/Get extended PAN ID

Syntax/Descriptor	Explanation
`+WPANID=<value>`	The command sets extended PAN ID for the device. `value` is extended PAN ID in form of a hexadecimal 64-bit number that uniquely identifies target network. If PAN ID is set to `0`, coordinator will form a network with extended PAN ID equal to its extended (MAC) address. Router and end device nodes in such case will join the first available network irrespectively to its extended PAN ID. Notes: 1. Setting the extended PAN ID is possible only when the device is not in the network. 2. Several networks with different PANIDs can be operated in parallel on the same frequency channel.
`+WPANID?`	The command returns extended PAN ID that is specified on the device for network operation.
`+WPANID=?`	The command requests valid range for extended PAN ID value.
`S-register`	`S21` (RW). This register is just keeping a copy of the parameter accessible through `+WPANID` command. `S20` (R). This register contains actual extended PAN ID that is used for networking. If `S21` register is set to `0`, and device is in the network, this register will keep extended PAN ID of the selected network. If device has not been connected, this register contains `0`.
`Result codes`	The set command is executed if device is not in the network and extended PAN ID is in the valid range. In such case device returns **`OK`** upon completion. Otherwise extended PAN ID is ignored and device responds with **`ERROR`**.

Example	`AT+WPANID=10`
	`OK`
	`AT+WPANID?`
	`+WPANID:0000000000000010`
	`OK`
	`AT+WPANID=?`
	`+WPANID:(0000000000000000-FFFFFFFFFFFFFFFE)`
	`OK`

Default value	`0000000000000000`
Persistence	`value` is stored in EEPROM
Node types	Coordinator / Router / End device

6.2.2 "+WCHAN"- Get active channel

Table 6-13. "+WCHAN"- Get active channel

Syntax/Descriptor	Explanation
+WCHAN?	The command requests the channel number (in hexadecimal form) the device is currently operating on. If the node is not in the network, **FF** is returned.
S-register	S22 (R)
Result codes	**OK**
Example	AT+WCHAN? **+WCHAN:0B** **OK**
Node types	Coordinator / Router / End device

6.2.3 "+WCHMASK" - Set/Get channel mask

Table 6-14. "+WCHMASK" - Set/Get channel mask

Syntax	Explanation
+WCHMASK=<value>-	The command sets channel mask to be used for network operation. value is a 32-bit field which specifies the channel numbers supported by the node. The 5 most significant bits of channel mask (b31,...,b27) shall be set to 0. The rest 27 bits (b26, b25,...b0) indicate availability status for each of the 27 valid channels (1 = supported, 0 = unsupported). Channels are distributed across frequency bands as follows: - 780 MHz: channel numbers 0 – 3 - 868 MHz: channel number 0 - 915 MHz: channel numbers 1 – 10 - 2.4 GHz: channel numbers 11 – 26 For sub-GHz bands corresponding channel page shall be configured by +WCHPAGE command (see Table 6-15). Detailed description of channel mask parameter can be found in the clause 6.1.2 of the 802.15.4-2006 standard. Notes: 1. Only channels from frequency bands supported by the platform's RF chip can be selected in the channel mask. 2. The command is not accessible when the node is joined to a network.
+WCHMASK?	The command returns actual channel mask. Returned channel mask can be different from the channel mask set by +WCHMASK=<value> command and depends on the hardware capabilities. The cleared bits mark unsupported channels.
+WCHMASK=?	The command returns channel capability mask in form of two 32-bit unsigned hexadecimal numbers. It returns 00000800-07FFF800 for 2.4 GHz chipset and 00000001-000007FF for Sub-GHz. Note: Strictly speaking, these two numbers do not represent "range" in its direct sense, but rather are the maximum and minimum values achievable by the composition of corresponding bits.
S-register	S23 (RW).
Result codes	The set command is executed if the node is not in the network and channel mask is set according to hardware capabilities really available. In such case device returns **OK**. Otherwise, channel mask is ignored and device responds with **ERROR**.

Table 6-14. "+WCHMASK" - Set/Get channel mask

Example	AT+WCHMASK=40000 **OK** AT+WCHMASK? **+WCHMASK:00040000** **OK** AT+WCHMASK=? **+WCHMASK(00000800-07FFF800)** **OK**
Default value	00000800 for 2.4 GHz chipset or 00000001 for Sub-GHz one.
Persistence	value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.2.4 +WCHPAGE" - Set/Get channel page

The command is available only for platforms with AT86RF212 radio part.

Table 6-15. "+WCHPAGE" - Set/Get channel page

Syntax	Explanation
+WCHPAGE=<value>	The command sets channel page that will be used for networking. Values 0 and 2 correspond respectively to BPSK and O-QPSK modulations on 868/915 MHz channels. Value 5 means that 780 MHz frequency band with O-QPSK modulation shall be used. Detailed description of channel page parameter can be found in the clause 6.1.2 of the 802.15.4-2006 standard Note: The command is not accessible when the node is joined to a network.
+WCHPAGE?	The command returns actual channel page.
+WCHPAGE=?	The command returns possible channel pages: **0,2,5**.
S-register	S25 (RW).
Result codes	**OK** if the device contains RF 212 radio chip and is not in the network; otherwise **ERROR** is returned.
Example	AT+WCHPAGE=0 **OK** AT+WCHPAGE? **+WCHPAGE:0** **OK** AT+WCHPAGE=? **+WCHPAGE:(0,2,5)** **OK**
Default value	0
Persistence	value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.2.5 +WAUTONET" - Enable/Disable automatic networking

Table 6-16. "+WAUTONET" - Enable/Disable automatic networking

Syntax	Explanation
+WAUTONET=<value>	The command controls the node activity behavior at power-up, reset or when a connection loss is detected. value has a Boolean type. 1 implies automatic joining to the network, 0 means that automatic joining is disabled and +WJOIN command shall be used for network start procedure.
+WAUTONET?	The command requests current automatic networking configuration.
+WAUTONET=?	The command requests the range of supported values.
S-register	S24 (RW).
Result codes	OK
Example	AT+WAUTONET=1 **OK** AT+WAUTONET? **+WAUTONET:1** **OK** AT+WAUTONET=? **+WAUTONET:(0,1)** **OK**
Default value	0 - automatic networking is disabled.
Persistence	value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.2.6 "+WROLE" - Set/Get node role (Coordinator / Router / End device)

Table 6-17. "+WROLE" - Set/Get node role (Coordinator / Router / End device

Syntax	Explanation
+WROLE=<value>	The command sets the node role to value as follows: 0 – Coordinator 1 – Router 2 – End device. Note: The command is not accessible when the node is joined to a network.
+WROLE?	The command requests the actual node role.
+WROLE=?	The command requests the node roles available for the device. Actual capabilities depend on the particular firmware version loaded on the device.
S-register	S33 (RW).
Result codes	**OK** is returned if value is in the valid range, otherwise **ERROR.**

Table 6-17. "+WROLE" - Set/Get node role (Coordinator / Router / End device

Example	AT+WLEAVE **OK** AT+WROLE=? **+WROLE:(0,1,2)** **OK** AT+WROLE=2 **OK** AT+WROLE? **+WROLE:2** **OK**	Leave the network Switch to the End device role
Default value	Depends on the firmware version. Typically 1 – Router.	
Persistence	value is stored in the EEPROM.	
Node types	Coordinator / Router / End device	

6.2.7 "+GSN" – Set/Get extended (MAC) address

Table 6-18. "+GSN" – Set/Get extended (MAC) address

Syntax	Explanation	
+GSN=<value>	The command assigns device extended (MAC) address. value is a 64-bit hexadecimal number that uniquely identifies the device. Note: The command is not accessible when the node is joined to a network.	
+GSN? I4	The command returns device extended (MAC) address in form of a 64-bit hexadecimal number.	
Result codes	**OK** is always returned	
Example	AT+GSN=FEDCBA0987654321 **OK** AT+GSN? **+GSN:FEDCBA0987654321** **OK** ATI4 **FEDCBA0987654321** **OK**	Just an alias to I4
Default value	0000000000000000 Notes: 1. If extended address is equal to zero then upon power up or reset SerialNet searches for the MAC address on 1-wire interface and applies it if detected. 2. User-defined MAC address shall be a non-zero values less than 0xFFFFFFFFFFFFFFFF (these values are reserved).	
Persistence	value is stored in EEPROM	
Node types	Coordinator / Router / End device	

6.2.8 "+WSRC" - Set/Get short (NWK) address

Table 6-19. "+WSRC" - Set/Get short (NWK) address

Syntax	Explanation
`+WSRC=<value>`	The command assigns device short (network) address. `value` is a 16-bit hexadecimal number which will be used by the device for communication in the network. It shall be unique within the network. Notes: 1. The command is not accessible when the node is joined to a network. 2. Coordinator node shall always have short address set as 0000. Nodes of other roles shall have non-zero short addresses.
`+WSRC?`	The command returns device short address in form of 16-bit hexadecimal number.
`+WSRC=?`	The command requests the range of valid addresses.
S-register	S55 (RW).
Result codes	**OK** is returned if `value` is in range, otherwise **ERROR** is returned.

Example	`AT+WSRC=2ABC`
	OK
	`AT+WSRC?`
	+WSRC:2ABC
	OK
	`AT+WSRC=?`
	+WSRC:(0000-FFF7)
	OK

Default value	FFFF Note: The default value is outside the allowed range, which means that the device will not join the network unless provided with the user-defined short (network) address.
Persistence	`addr` value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.3 Network Management Functions

SerialNet commands described in this section execute various network management functionality including network join and leave operations, obtaining network topology-related information, getting link quality data, etc.

When exploring network topology it is important to take into account the fact that due to mesh networking only an end device node can be a child and has a dedicated parent node (coordinator or router) during its lifetime in the network. Router nodes use coordinator or other routers only as network entry points and are not associated as direct children after network join. However, if there is enough space in node's neighbor table it will contain information about neighbor coordinator/router nodes.

6.3.1 "+WJOIN" - Start/Join to the network

Table 6-20. "+WJOIN" - Start/Join to the network

Syntax	Explanation
`+WJOIN`	The command forces device to form a network (for Coordinator node) or to join an existing network (for Router or End device nodes). Desired network and device characteristics shall be set prior to `+WJOIN` request using if necessary SerialNet commands from "Networking Parameters" on page 6-7.
Result codes	**OK** is returned if network formation/join is completed successfully; **ERROR** is returned if failed. If the node is in the network already, it returns **OK** immediately.
Example	`AT+WJOIN` **OK**
Node types	Coordinator / Router / End device

6.3.2 "+WLEAVE" - Leave the network

Table 6-21. "+WLEAVE" - Leave the network

Syntax	Explanation
`+WLEAVE`	The command forces the node to leave the network. If node has any children it will automatically force them to leave the network as well. Note: Parameters stored in EEPROM persist even after the node leaves the network.
Result codes	**OK** is returned on the process completion. If the device was not connected before starting the process, it returns **ERROR** immediately.
Example	`AT+WLEAVE` **OK**
Node types	Coordinator / Router / End device

6.3.3 "+WNWK" – Get networking status

Table 6-22. "+WNWK" – Get networking status

Syntax	Explanation	
`+WNWK`	The command requests current networking status of the device.	
Result codes	**OK** is returned if the device is joined to a network, otherwise it returns **ERROR**.	

Example	`AT+WLEAVE`
	OK
	`AT+WNWK`
	ERROR
		Leave the network first
		
		Device is not in a network now

Node types	Coordinator / Router / End device	

6.3.4 +WPARENT" - Get parent address

Table 6-23. "+WPARENT" - Get parent address

Syntax	Explanation
+WPARENT?	The command requests parent node address the device is associated to. Extended (MAC) address of the parent node is returned as a 64-bit hexadecimal number if S30 register is set to 0. Short (NWK) parent address is returned if S30 register is set to 1. See Table 6-27) for details. Note: This command does not cause network operations and just returns a copy of the parent address assigned during the joining process.
Result codes	**OK** is returned if the module is in the network and has a parent. **ERROR** will be returned if the device is not in the connected state or has node role Coordinator or Router.
Example	AT+WPARENT? **+WPARENT:0123456789ABCDEF** **OK**
Node types	End devices

6.3.5 “+WCHILDREN” – Get children addresses

Table 6-24. “+WCHILDREN” – Get children addresses

Syntax	Explanation
+WCHILDREN?	The command requests addresses of children end devices associated to the node. Extended (MAC) addresses of children nodes are returned as 64-bit hexadecimal numbers if S30 register is set to 0. Short (NWK) addresses of children nodes are returned if S30 register is set to 1. See Table 6-27for details. Children addresses are returned delimited by commas. Notes: 1. An end device is removed from the children list if the parent node receives no poll requests from the child during *3*(sleep_interval + sync_period)* time interval as configured on parent device by +WPWR and +WSYNCPRD commands. 2. This command does not cause network operations and just returns copies of the children addresses stored in the parent memory.
Result codes	**OK** is returned if the module is in the network even though there is no child connected yet. **ERROR** will be returned if the device is not in the connected state or has End device node role.
Example	AT+WCHILDREN? **+WCHILDREN:0123456789ABCDEF,123456789ABCDEF0** **OK**
Node types	Coordinator and Routers

6.3.6 "+WNBSIZE" - Get number of neighbors

Table 6-25. "+WNBSIZE" - Get number of neighbors

Syntax	Explanation
+WNBSIZE?	The command requests a number of entries in node’s neighbor table. Returned result consists of two values: the first one is the current number of occupied entries in node’s neighbor table; the second value is the maximum possible number of entries (size of the neighbor table).

Table 6-25. "+WNBSIZE" - Get number of neighbors

Result codes	**OK** is returned if the node is in the network. If device is not in the connected state **ERROR** will be returned.
Example	AT+WNBSIZE? **+WNBSIZE:2,5** **OK**
Node types	Coordinator / Router / End device

6.3.7 "+WNB" - Get neighbor information

Table 6-26. "+WNB" - Get neighbor information

Syntax	Explanation
+WNB <node_role> [,<device_addr>]	The command requests content of node's neighbor table. node_role parameter specifies node role of neighboring nodes to be extracted from the neighbor table. Following values are accepted: 0 – coordinator 1 – router 2 – end device 3 – all device types Optional parameter device_addr specifies the address of the neighboring node to be extracted. If S30 register is set to 0. device_addr is accepted as short (NWK) address if S30 register is set to 1, device_addr is expected to be an extended (MAC) address. See Table 6-27 for details. The command's information response has the following format: seqNr ¦ nodeRole ¦ extAddr ¦ nwkAddr ¦ relationship ¦ depth where seqNr – is the sequence number in the neighbor table nodeRole – is the node role of the neighbor extAddr – is neighbor's extended address nwkAddr – is neighbor's network address relationship – is neighbor's relationship to current node (0-parent, 1 – child, 3 – no relationship) depth – is neighbor's network depth Notes: 1. A neighbor entry is removed from the table if the node during certain interval doesn't receive any periodic management frames expected from the neighbor. If neighbor is a router/coordinator this interval is 45 seconds (management frames are sent once per 15 sec). If neighbor is an end device then interval equals *3*(sleep_interval +sync._period)* as configured on the node by +WPWR and +WSYNCPRD commands. 2. Although right after network join an end device node can have information about several nodes in its neighbor table, only actual parent node persists in the table while information about other nodes is removed shortly after end device join. Same is valid for information about an end device neighbor – in long term period it is present only in the neighbor table of its parent and is not directly "visible" for other routers in its neighborhood. 3. This command does not cause network operations and just returns information from node's current neighbor table.

Table 6-26. "+WNB" - Get neighbor information

Result codes	**OK** is returned if the node is in the network. If the node is not in the connected state **ERROR** will be returned.
Example	AT+WNB 3 **1 | 0 | 0000000000000001 | 0000 |3 | 2** **2 | 1 | 0000000000000002 | 0002 |0 | 1** **OK** AT+WNB 1,2 **1 | 1 | 0000000000000002 | 0002 |0 | 1** **OK**
Node types	Coordinator / Router / End device

6.3.8 "S30" - Set node addressing mode

Table 6-27. "S30" - Set node addressing mode

Syntax	Explanation
S30=<value>	The command sets the node addressing scheme to be used by some SerialNet commands. value: specifies addressing mode 0 -extended (64-bit) addressing 1- short (16-bit) addressing
S30?	The command requests current addressing mode.
Result codes	The command returns **OK** if value is in range, otherwise **ERROR**.
S-register	S30 (RW)
Example	ATS30=0 **OK** AT+WPARENT? **+WPARENT:000100000A3B98CC** **OK** ATS30=1 **OK** AT+WPARENT? **+WPARENT:0000** **OK**
Node types	Coordinator / Router / End device
Default value	0
Persistence	value is NOT stored in EEPROM

Note: Setting the addressing mode, the S30 command affects the performance of the following commands: +WPARENT? (see Section Table 6-23), +WCHILDREN? (see Table 6-24), and +WNB (see Table 6-26). These commands use extended (MAC) address if S30 is set to 0, but will switch to using short (NWK) addressing if S30 is set to 1.

6.3.9 "+WLQI" - Get LQI value

Table 6-28. "+WLQI" - Get LQI value

Syntax	Explanation	
`+WLQI <addr>`	The command requests LQI for the link to the node with short (NWK) address equal to `addr` specified in 16-bit hexadecimal format. The command returns the actual LQI value in the range of `0…255`. Notes: 1. LQI information can be retrieved for links within one-hop radius only. 2. An end device can obtain LQI only to its current parent node and vice versa: LQI to an end device can be obtained only from its current parent node. 3. LQI value is measured during data transmission initiated by `ATD` command. If `ATD` has not been performed yet, `+WLQI` may return irrelevant value.	
Result codes	The node returns **`OK`** if device is in the network and LQI value for this particular link exists, otherwise **`ERROR`** will be returned.	
Example	`AT+WLQI 1` **`+WLQI:254`** **`OK`**	request LQI for the link to the node with short address 0x0001
Node types	Coordinator / Router / End device	

6.3.10 "+WRSSI" - Get RSSI value

Table 6-29. "+WRSSI" - Get RSSI value

Syntax	Explanation	
`+WRSSI <addr>`	The command requests RSSI value for the link to the node with short (NWK) address equal to `addr` specified in 16-bit hexadecimal format. The command returns the actual RSSI value expressed in dBm. If RSSI is not available, then **-91** value is returned. Notes: 1. RSSI information can be retrieved for links within one-hop radius only. 2. An end device can obtain RSSI only to its current parent and vice versa: RSSI to an end device can be obtained only from its current parent node. 3. RSSI value is measured during data transmission initiated by `ATD` command. If `ATD` has not been performed yet, `+WRSSI` may return irrelevant value.	
Result codes	The node returns **`OK`** if device is in the network and RSSI value for this particular link exists, otherwise **`ERROR`** will be returned.	
Example	`AT+WRSSI 0001` **`+WRSSI:-80`** **`OK`**	request RSSI for the link to the node with short address 0x0001 -80 dBm
Node types	Coordinator / Router / End device	

6.4 Data Transmission

In SerialNet data can be transmitted in two ways:

- "Unicast transmission to a particular node using the D, DS or +WPING commands;
- "Broadcast transmission to all nodes using the DU command or D command with broadcast address.

It is important that extended (MAC) addresses are not used for data transmission directly; instead, they are substituted by short (network) addresses which are convenient for node replacement in network installation and maintenance.

Route establishment procedure to the target node is implemented inside the stack. It is executed automatically upon data transmission request and then if a route exists, data delivery (one-hop or multi-hop) is performed to the destination node.

Following application identifiers are used in SerialNet for all data exchange operations:

- "Profile ID: 0xC31A
- "Endpoint ID: 0x01
- "Cluster ID: 0x00

Note: **To ensure data transmission safely over serial interface between a host and an MCU, it is strongly recommended setting hardware flow control (see** Table 6-60 on page 6-38**for details). When running terminal software to control the node, the chosen COM port should be set with the Hardware flow control option selected.**

6.4.1 Parent polling mechanism

Data delivery to an end device over the last hop (i.e. from the parent node to the child) is performed using polling mechanism described below.

Upon frame reception destined for its child node or broadcast frame with non-exhausted transmission radius and destination address equal 0xFFFF, parent node buffers the frame and waits for poll request from the child. The maximum waiting time is (sleep_interval+3*sync._period) as configured on the parent by +WPWR and +WSYNCPRD commands.

In awake state an end device polls its parent node periodically every +WSYNCPRD ms (as configured on end device). Parent node can transmit a data frame to a child only after receiving corresponding data poll from it. After data frame reception is completed the end device issues another data poll request to verify whether there are any frames buffered at the parent.

6.4.2 "D" - Send data to a specific node

Table 6-30. "D" - Send data to a specific node

Syntax	Explanation	
`D <addr>[,[<arq>] [,<length>]] <data>`	The command sends data to a specific node. `addr` is a 16-bit hexadecimal short (network) address of the destination node. Optional `arq` parameter (equals to 1 or 0) controls ARQ/nonARQ data delivery mode, meaning 1 (i.e. ARQ) as default if omitted. `length` means the length in bytes of the data portion to be sent. The data portion length shall not exceed the maximum allowable number (84 characters). If length parameter is omitted, the maximum allowable number is implied by default. Data transmission starts either when the specified number of data bytes is received over serial interface or when the time interval between two consecutive data symbols exceeds the time-out preset (`+WWAIT` command - seeTable 6-37 on page 6-23). Notes: 1. `data` should be preceded by `<CR>` (S3 character, see Table 6-51 on page 6-32). This symbol is not transmitted over the air and it is not counted in `length`. 2. If the destination address is a broadcast address (FFFF for all nodes or FFFE for router/coordinator nodes), the broadcast transmission is performed.	
Result codes	If acknowledgement is requested (`arq` is set to 1), the node responds with **OK** upon receiving an acknowledgement in several attempts (see parameter `+WRETRY` in Table 6-36 on page 6-23)), otherwise it returns **ERROR**. If the destination node or the sending node itself is not in the network **ERROR** is returned.	

Example	`ATD 12,1,5`
	`HELLO`
	OK
	`ATD 12`
	`HELLO`
	OK
		Send `HELLO` to the node with address 12 using ARQ.
		
		The same as above, but the node will be waiting for the time-out expiration before going to the air.

Node types	Coordinator / Router / End device	

6.4.3 "DU" - Send broadcast data

Table 6-31. "DU" - Send broadcast data

Syntax	Explanation
`DU [<length>] <data>`	The command sends `data` using broadcast transmission. `length` means the length in bytes of the `data` portion to be sent. The `data` portion may not exceed the maximum allowable number (84 characters). If `length` parameter is omitted, the maximum allowable number is implied by default. Data transmission starts either when the specified number of data bytes is received over serial interface or when the time interval between two consecutive data symbols exceeds the time-out preset (`+WWAIT` command,Table 6-37 on page 6-23). Notes: 1. `ATDU` is, in fact, shorthand for `ATD` command with broadcast address (FFFF) as destination. 2. Data should be preceded by `<CR>` (S3 character, see Table 6-51 on page 6-32). This symbol is not transmitted over the air and it is not counted in length. 3. Data is broadcasted to the whole network (radius 0).

Table 6-31. "DU" - Send broadcast data

Result codes	The node responds with **OK** immediately after the transmission if the node itself is in the network. Otherwise, **ERROR** is returned.	
Example	ATDU HELLO **OK**	Send **HELLO** to all nodes in the network
Node types	Coordinator / Router / End device	

6.4.4 "DS" - Send S-register value to a specific node

Table 6-32. "DS" - Send S-register value to a specific node

Syntax	Explanation	
DS <S-reg>,<addr> [,[<arq>]	The command sends S-register value to a specific node. Default arq parameter (is set to 1 or 0) specifies whether the ARQ or non-ARQ data delivery mode is used. 1 is implied if arq is omitted. Destination node address addr should be a 16-bit hexadecimal short (network) address. S-register data is sent in the form readable by ATS command without the line termination characters. Note: S-registers defined by user extensions are also accessible by this command.	
Result codes	If acknowledgement is requested (arq is set to 1), the node responds with **OK** upon receiving acknowledgement in several attempts (see parameter +WRETRY, Table 6-36 on page 6-23), otherwise it returns **ERROR**. If the destination node or the sending node itself is not in the network **ERROR** is returned. Also, if the specified S-register can not be read, the command returns **ERROR** and the node does not send anything to the air.	
Example	ATDS130,2,0 **OK**	Send GPIO0 value to the node with address 2 without using ARQ.
Node types	Coordinator / Router / End device	

6.4.5 "+WPING" - Ping the node

Table 6-33. "+WPING" - Ping the node

Syntax	Explanation
+WPING <addr>	The command pings the targeted node. addr specifies destination address as 16-bit hexadecimal short (network) address. This command is equivalent to D command with ARQ and zero data length: ATD <addr>,1,0.
Result codes	The node responds with **OK** upon receiving acknowledgement in several attempts (see parameter +WRETRY, Table 6-36 on page 6-23) , otherwise it returns **ERROR**. If the destination node or the sending node itself is not in the network **ERROR** is returned.
Example	AT+WPING 1 **OK**
Node types	Coordinator / Router / End device

6.4.6 "+WSYNCPRD" - Poll rate for requesting indirect transactions from the parent

Table 6-34. "+WSYNCPRD" - Poll rate for requesting indirect transactions from the parent

Syntax	Explanation	
+WSYNCPRD=<rate	The command sets poll interval to the <rate> value measured in milliseconds. This value is used by the End device as the poll rate for requesting indirect transmission messages from the parent. Coordinator and router use this rate to verify children presence. Notes: 1. On End devices, the <rate> value must not be increased by this command. Otherwise, BitCloud behavior is unpredictable. 2. On routers and coordinators this parameter must be set to the largest <rate> value among all children. Otherwise, child presence status may be detected incorrectly. 3. This value should be at least 2 times smaller than the value of +WTIMEOUT (see Table 6-35 on page 6-22). 4. The command is not accessible when the node is joined to a network.	
+WSYNCPRD?	The command requests the actual poll rate.	
+WSYNCPRD=?	The command requests allowable range of poll rate values.	
S-registers	S37 (RW).	
Result codes	**OK** is always returned.	
Example	AT+WSYNCPRD=1000 **OK** ATS37? **300** **OK** AT+WSYNCPRD=? **+WSYNCPRD:(10-30000)** **OK**	Set poll rate to 1 sec
Default values	1400	
Node types	Coordinator / Router / End device	
Persistence	rate is NOT stored in EEPROM	

6.4.7 "+WTIMEOUT" - Data delivery time-out

Table 6-35. "+WTIMEOUT" - Data delivery time-out

Syntax	Explanation
+WTIMEOUT?	The command returns the time-out value in milliseconds. The returned value corresponds to the apscAckWaitDuration variable introduced by ZigBee recommendation Step 2. in "Related Documents and References" on page 2-1.
S-register	S51 (R).
Result codes	**OK** is always returned
Example	AT+WTIMEOUT? **+WTIMEOUT:2800** **OK**
Node types	Coordinator / Router / End device

6.4.8 "+WRETRY" - Repetition count

Table 6-36. "+WRETRY" - Repetition count

Syntax	Explanation
+WRETRY?	The command returns actual number of retransmission. The returned value corresponds to the apscMaxFrameRetries variable introduced by ZigBee recommendation .
S-register	S52 (R).
Result codes	**OK** is always returned
Example	AT+WRETRY? **+WRETRY:3** **OK**
Node types	Coordinator / Router / End device

6.4.9 "+WWAIT" - Data transmission waiting time-out

Table 6-37. "+WWAIT" - Data transmission waiting time-out

Syntax	Explanation
+WWAIT=<value>	The value parameter sets the time-out (in milliseconds) for the module to wait for entering the D (see Table 6-30 on page 6-20) or the DU (see Table 6-31 on page 6-20) command. If a pause between two consecutive characters coming from serial interface exceeds the specified time-out, the node will start data transmission even though the data length encountered has not yet reached the number specified by the length argument of the D/DU command. In such case, the length is replaced with its actual value according to the data transmitted.
+WWAIT?	The command returns actual time-out value.
+WWAIT=?	The command requests for the range of valid time-outs.
S-register	S53 (RW).
Result codes	**OK** is returned if the value is in range, otherwise **ERROR** is returned.
Example	AT+WWAIT=500 **OK** AT+WWAIT? **+WWAIT:500** **OK** AT+WWAIT=? **+WWAIT:(100-500)** **OK**
Default value	5000
Persistence	value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.5 Power Management

Because power consumption is a major concern in applications with battery-powered devices, SerialNet provides AT-commands that allow switching between awake and sleep modes as well as setting transmit power level.

Note that sleep mode is supported on end device nodes designed on ZigBit modules only and is not available for nodes using RZUSBSTICK platform. To avoid issues in network stability coordinator and router nodes are always kept in active mode and hence require continuous power supply.

In addition to power management of ZigBit module SerialNet simplifies power management of external peripherals or the host device via CTS line. If hardware flow control is enabled by +IFC command (see Table 6-60 on page 6-38), the line becomes high when the ZigBit module is in the sleep state.

6.5.1 "+WPWR" - Configuration of sleep/active intervals

Table 6-38. "+WPWR" - Configuration of sleep/active intervals

Syntax	Explanation	
+WPWR=<sleep>, <active>	The command sets duration of sleep and active intervals for end device node. sleep duration is specified in 100 msec units but active duration – in 10 msec units. Zero active period means that the node can be put asleep only explicitly by +WSLEEP command (in which case it will stay asleep for given sleep duration). On a coordinator/router node sleep interval is used for children tracking and should be not less than on its children nodes. It is also used as maximum time interval the data destined for the child can be buffered for. See "Parent polling mechanism" on page 6-19 for more details. Note: 1. Actual sleep/active periods will be slightly different and their values depend on multiple circumstances such as the network activity, external interfaces to the sensors, and so on. They can not be used for absolute timing. 2. The command is not accessible when the node is joined to a network.	
+WPWR?	The command requests current sleep/active intervals.	
+WPWR=?	The command requests valid ranges of sleep/active intervals.	
S-registers	S31, S32 (RW).	
Result codes	**OK** is returned if parameters are within their valid ranges. Otherwise **ERROR** is returned.	
Example	AT+WPWR=600,10 **OK** AT+WPWR? **+WPWR:600,10** **OK** ATS31? **600** **OK** AT+WPWR=? **+WPWR:(2-30000),(0-30000)** **OK**	Set duty cycle 1 min. sleep / 100 msec active Verify setting is applied Get sleep interval via S-register Get valid ranges for sleep/active intervals
Default values	100,0 (the node sleeps for 10 seconds if put asleep by +WSLEEP command)	
Persistence	The sleep, active values are stored in the EEPROM.	
Node types	Coordinator / Router / End device	

6.5.2 "+WSLEEP" - Force node to sleep

Table 6-39. "+WSLEEP" - Force node to sleep

Syntax	Explanation
+WSLEEP	The command forces the node to fall into the sleep mode. The command is supported on ZigBit modules only and is not available on RZUSBSTICK. Important: Take in mind that the node in sleep mode can respond to the subsequent commands with a delay, depending on the sleeping interval specified (see Table 6-38 on page 6-24), the node version and DTR configuration (see Table 6-61 on page 6-39). The command is accessible only when the node is joined to a network.
Result codes	**OK** is returned for End devices, otherwise **ERROR**. Note: The command is executed as follows: the node returns the result code first, and then it disables any of subsequent commands, completes pending operations and finally falls into the sleep mode. Wake-up occurs as scheduled by +WPWR command or DTR interrupt if enabled.
Example	AT+WSLEEP **OK**
Node types	End devices

6.5.3 "+WTXPWR" - TX power level

Table 6-40. "+WTXPWR" - TX power level

Syntax	Explanation	
+WTXPWR=<value>	The command sets transmit power level for the device. The value represents TX power level measured in dBm. Note: This setting will be applied to the radio circuitry during the warm reset procedure only. Thus, the accurate setting of TX power requires warm reboot of the node in using Z command, see Table 6-41 on page 6-26.	
+WTXPWR?	The command requests actual Tx power level. Notes: 1. Power level resolution is hardware dependent and may be coarser than 1 dB, so that some power values (say, -4, -6, -8…) may be forbidden, even despite being within the allowed range. On input, such values are rounded to the nearest allowed value. 2. This command just returns the number set by the +WTXPWR= command, but does not indicate real power level, which can vary due to the temperature, supply voltage and another factors.	
+WTXPWR=?	The command requests the allowable range of TX power level.	
S-register	S34 (RW).	
Result codes	**OK** is returned if value is in the valid range, otherwise **ERROR**.	
Example	AT+WTXPWR=-5 **OK** AT+WTXPWR? **+WTXPWR:-5** **OK** AT+WTXPWR=? **+WTXPWR:(-17-3)** **OK**	set -5dBm Tx power level

Table 6-40. "+WTXPWR" - TX power level

Default value	Hardware dependent, typically 3
Persistence	`value` is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.6 Generic Control

6.6.1 "Z" - Warm reset

Table 6-41. "Z" - Warm reset

Syntax	Explanation
`Z`	The command instructs the device to execute warm (software) reset. This command resets the hardware, restores all persistent variables from EEPROM and restarts the firmware. The command is supported on ZigBit modules only and is not available on RZUSBSTICK. Important: The command should be used with precautions since it does not send 'leaving the network' signals to other nodes and hence can affect PAN's integrity. Therefore, the node should better be put out of the network by the `+WLEAVE` command prior to reset. If automatic networking is disabled then the node will not join the network automatically after reset. Note that the parameters stored in EEPROM persist after software reset; to erase them, use the `AT&F` command (see Table 6-48 on page 6-30). If `Z` is put in a line together with some other commands, the processing of those placed after `Z` is disabled. Result code is sent upon the reset process is completed. During the reset process some transients can be observed on the module pins (including GPIO) because of the nature of the MCU used. It is strongly recommended to wait until **`OK`** result code is received (or an equivalent numerical code 0 if verbose result codes are disabled by `V0` command, see Table 6-56 on page 6-35) before sending any new command to the module.
Result codes	**`OK`** is returned if command is supported for the device's platform. **`ERROR`** is returned otherwise.
Example	`ATZ` **`OK`**
Node types	Coordinator / Router / End device

6.6.2 "&H" - Command Help

Table 6-42. "&H" - Command Help

Syntax	Explanation
`&H`	The command outputs a list of valid AT-commands. The listing order may change. It depends on firmware version.
Result codes	**OK** is always returned

Example	`AT&H`
	E
	V
	Q
	Z
	&F
	+IPR
	+IFC
	&D
	&H
	%H
	I
	+GMI
	+GMM
	+GMR
	+GSN
	`(skipped…)`
	S146
	S147
	S148
	OK

Node types	Coordinator / Router / End device

6.6.3 "%H" - Display parameters and S-register values

Table 6-43. "%H" - Display parameters and S-register value

Syntax	Explanation
%H	The command outputs the values of parameters and S-registers. The listing order may change. It depends on firmware version.
Result codes	**OK** is always returned
Example	AT%H **+WPANID: 0000000000000000** **+WCHAN: FF** **+WCHMASK: 00000800** **+WAUTONET: 0** **+WPWR: 100,1000** **+WROLE: 2** **+WSRC: 0001** **+WSYNCPRD: 1400** **+WTXPWR: 0** **+WTIMEOUT: 2800** **+WRETRY: 3** **+WWAIT: 5000** **E: 1** **Q: 0** **V: 1** **X: 0** **+IPR: 38400** **+IFC: 0,0** **+GMI: ATMEL** **+GMM: ZIGBIT** **+GMR: BitCloud v.1.5.0; SerialNet v.2.2.0** **+GSN: 0001000011672CFC** *(skipped…)* **S146:0** **S147:0** **S148:0** **OK**
Node types	Coordinator / Router / End device

6.6.4 "I" - Display product identification information

Table 6-44. "I" - Display product identification information

Syntax	Explanation		
`I[<value>]`	The command instructs the node to return information text identifying the device. Information text depends on the `value` as follows:		
	value	Information text	Reference
	0 1 2 3 4	All the identifiers below Manufacturer identifier Model identifier Hardware/software revision identifier Product serial number identifier	 Section 6.6.5 Section 6.6.6 Section 6.6.7 Section 6.2.7
	If `value` is omitted, 0 is implied by default.		
Result codes	**OK** for any of the aforementioned values, **ERROR** otherwise.		
Example	`ATI0` **ATMEL** **ZIGBIT** **BitCloud v.1.5.0; SerialNet v.2.2.0** **000100001090C3F9** **OK**		
Node types	Coordinator / Router / End device		

6.6.5 "+GMI" - Get manufacturer identifier

Table 6-45. "+GMI" - Get manufacturer identifier

Syntax	Explanation	
`+GMI?` `I1`	The command instructs the node to output information text i identifying the manufacturer.	
Result codes	**OK** is always returned	

Example	`AT+GMI?`
	+GMI:ATMEL
	OK
	`ATI1`
	ATMEL
	OK
		Just an alias to `+GMI`

Node types	Coordinator / Router / End device	

6.6.6 "+GMM" - Request for the model identifier

Table 6-46. "+GMM" - Request for the model identifier

Syntax	Explanation	
+GMM? I2	The command instructs the node to transmit information text identifying the particular model of the device.	
Result codes	**OK** is always returned	
Example	AT+GMM? **+GMM:ZIGBIT** **OK** **ATI2** **ZIGBIT** **OK**	Just an alias to +GMM
Node types	Coordinator / Router / End device	

6.6.7 "+GMR" - Request for the hardware/software revision identifier

Table 6-47. "+GMR" - Request for the hardware/software revision identifier

Syntax	Explanation	
+GMR? I3	This command instructs the node to transmit an information text intended to identify the actual revision of hardware or software product burned into the device.	
Result codes	**OK** is always returned	
Example	AT+GMR? **+GMR: BitCloud v. 1.5.0; SerialNet v.2.2.0** **OK** ATI3 **+GMR: BitCloud v. 1.2 5.0; SerialNet v.2.2.0** **OK**	Just an alias to +GMR
Node types	Coordinator / Router / End device	

6.6.8 "&F" – Set to factory-default configuration

Table 6-48. "&F" – Set to factory-default configuration

Syntax	Explanation
&F	The command instructs the module to set all the parameters (including the persistent variables from EEPROM) to the factory defaults. This command forces hardware reset like the Z command does, so all the precautions in should be considered. Result code will be issued according to actual result code suppression setting (see Table 6-55 on page 6-35), response formatting (see Table 6-56 on page 6-35) and the transmission rate (see Table 6-59 on page 6-37) set before execution of this command. Note that &F command does not reset the password, once it has been set by the +WPASSWORD command (see Table 6-70 on page 6-46).

Table 6-48. "&F" – Set to factory-default configuration

Result codes	**OK** is always returned
Example	AT&F **OK**
Node types	Coordinator / Router / End device

6.6.9 "+WACALIBRATE" - Configure periodic internal clock calibration

Table 6-49. "+WACALIBRATE" - Configure periodic internal clock calibration

Syntax	Explanation
+WACALIBRATE=<value>	The command requests the device to automatically calibrate the internal clock. value is an unsigned integer between 0 and 65535 which determines the period of calibration in minutes (i.e. how many minutes elapse between consecutive calibrations). The command is supported on ZigBit modules only and is a no-op on RZUSBSTICK. It can be used to prevent frequency drift of MCU's internal RC-oscillator that can impact or even block serial communication with the host.
+WACALIBRATE?	The command returns the period of calibration (in minutes).
+WACALIBRATE=?	The command returns permitted range of values for the period of calibration.
Result codes	**OK** is returned on successful command completion Otherwise, value is ignored and device responds with **ERROR**.
Example	AT+WACALIBRATE=60 **OK** AT+WACALIBRATE? **+WACALIBRATE:60** **OK** AT+WACALIBRATE=? **+WACALIBRATE(0-65535)** **OK**
Default value	0
Persistence	The value is stored in the EEPROM.
Node types	Coordinator / Router / End device

6.6.10 "+WCALIBRATE" - Calibrate internal clock

Table 6-50. "+WCALIBRATE" - Calibrate internal clock

Syntax	Explanation
+WCALIBRATE	The command requests the device to calibrate the internal clock. The command is supported on ZigBit modules only and is a no-op on RZUSBSTICK. It can be used to prevent frequency drift of MCU's internal RC-oscillator that can impact or even block serial communication with the host.

Table 6-50. "+WCALIBRATE" - Calibrate internal clock

Result codes	**OK** is returned on successful calibration. Otherwise, device responds with **ERROR**.
Example	AT+WCALIBRATE **OK**
Node types	Coordinator / Router / End device

6.7 Hot Interface Commands

6.7.1 "S3" - Termination character

Table 6-51. "S3" - Termination character

Syntax	Explanation
S3=<value>	The command sets ASCII code to be used as termination character in command line, response and result code formatting. value may be specified in the range of 0…127. Note: It is strongly recommended to avoid changing of this parameter during the network operation.
S3?	The command requests for actual ASCII code currently used as the termination character.
Result codes	The module returns **OK** if value is in range, otherwise **ERROR**. Important: It is the previous value of S3 which is used in entering the command line containing the S3 setting command to specify the next command line termination character. However, the result code when issued will use the value of S3 as that one set during the processing of the command line. For example, if S3 was previously set to 13 and the 'ATS3=30' command line is issued, the command line will be terminated with a CR character, but the result code when issued will use the character with the decimal value 30 instead of **<CR>**.
Example	ATS3=13 **OK** ATS3? **13** **OK**
Node types	Coordinator / Router / End device
Default value	13 - **<CR>** (carriage return character)
Persistence	value is stored in the EEPROM.

6.7.2 "S4" - Response formatting character

Table 6-52. "S4" - Response formatting character

Syntax	Explanation
S4=<value>	The command sets ASCII code of character to be used in responses and result code formatting along with the S3 parameter (see Table 6-51 on page 6-32). The description of V command shows the parameter usage, see Table 6-56 on page 6-35 for details. value may be specified in the range of 0…127. Note: It is strongly recommended to avoid changing of this parameter during the network operation.
S4?	The command requests for actual ASCII code currently used as the response formatting character.
Result codes	The module returns **OK** if value is in the allowed range, and **ERROR** otherwise. Note: The changed value of S4 will be used in formatting of the result code and information responses immediately after processing the 'S4=<value>' command. If the value of S4 is changed in a command line, the result code issued in response to that command line will use the new value of S4.
Example	ATS4=10 **OK** ATS4? **10** **OK**
Node types	Coordinator / Router / End device
Default value	10 - <LF> (Line Feed character)
Persistence	value is stored in the EEPROM.

6.7.3 "S5" - Command editing character

Table 6-53. "S5" - Command editing character

Syntax	Explanation
S5=<value>	The command sets ASCII code to be used as the control character pointing to delete the just preceding character in the command line, see "Basic Command-Line Operations" on page 6-1. value may be specified in the range of 0…127. Note: It is strongly recommended not to set this parameter to any letter or other symbol that can be a part of a command. For example, setting it to letter A, either upper- or lowercase (ASCII code 65 or 97) would effectively prevent entering of any subsequent AT command.
S5?	The command requests for actual ASCII code of the command editing character.

Table 6-53. "S5" - Command editing character

Result codes	The module returns **OK** if `value` is in range, otherwise **ERROR**. Note: The changed value of S5 will be used in editing of subsequent command lines and will be applied after processing the line containing S5 register change.

Example	`ATS5=8`
	OK
	`ATS5?`
	8
	OK

Node types	Coordinator / Router / End device
Default value	`8 - <BS> (Backspace Character)`
Persistence	`value` is stored in the EEPROM.

6.7.4 "E" - Command echo

Table 6-54. "E" - Command echo

Syntax	Explanation	
`E[<value>]`	Setting this parameter instructs if the module should echo the characters received from UART. `value` may be specified as 0 or 1 to disable or enable echoing, correspondingly. If `value` is omitted 0 is implied by default.	
Result codes	The module returns **OK** if `value` is 0 or 1, otherwise **ERROR**.	
Example	`ATE` **OK**	Disable echo
	`ATE1` **OK**	Enable echo
Node types	Coordinator / Router / End device	
Default value	`1` - echoing is enabled	
Persistence	`value` is stored in the EEPROM.	

6.7.5 "Q" - Result code suppression

Table 6-55. "Q" - Result code suppression

Syntax	Explanation	
`Q[<value>]`	Setting this parameter instructs if the module should transmit the result codes to UART. When result codes are being suppressed, no portion of any intermediate, final, or unsolicited result code – header, result text, line terminator, or trailer (see “Parameter Persistence” on page 5-3, and) – is transmitted. Information text transmitted in response to a command is not affected by setting of this parameter. There are two possibilities for `value`: `0` The module transmits result codes. `1` Result codes are suppressed and so not transmitted. If `value` is omitted, 0 is implied.	
Result codes	Nothing will be received for `ATQ1` command, **OK** if `value` is 0, otherwise the module returns **ERROR**.	
Example	`ATQ0` **OK**	Enable the result codes
	`ATQ1`	Suppress the result codes. No **OK** will be sent because it is suppressed
Node types	Coordinator / Router / End device	
Default value	0 – enables result codes	
Persistence	`value` is stored in the EEPROM.	

6.7.6 "V" - Response format

Table 6-56. "V" - Response format

Syntax	Explanation	
`V[<value>]`	Setting this parameter defines the contents of header and trailer transmitted with result codes and information responses. It also determines whether result codes are transmitted in numeric, alphabetic, or "verbose", form. The text portion of information responses is not affected by this setting. shows the effect of the setting of this parameter on the format of information text and result codes. If `value` is omitted, `0` is implied.	
Result codes	**0**	If `value` is `0` (because numeric response text is being used)
	OK	If `value` is `1`.
	4	For unsupported values (if previous `value` was `0`).
	ERROR	For unsupported values (if previous `value` was `1`).

Example	`ATV1`
	OK
	`ATV0`
	0
		`0` will be output on the same line because <LF> is not used for formatting of result code!

Table 6-56. "V" - Response format

Node types	Coordinator / Router / End device
Default value	1 – verbose format
Persistence	value is stored in the EEPROM

Table Table 6-52 below summarizes the usage of response formats. All references to <CR> mean "the character ASCII coded as specified in parameter S3 (see Table 6-51 on page 6-32)"; all references to <LF> likewise mean "the character ASCII coded as specified in parameter S4 (see Table 6-52 on page 6-33)". Numeric and verbose codes are discussed in "Parameter Persistence" on page 5-3.

Table 6-57. Response formatting

Value	0	1
Information responses	**<text><CR><LF>**	**<CR><LF><text><CR><LF>**
Result codes	**<numeric code><CR>**	**<CR><LF><verbose code><CR><LF>**

6.7.7 "X" - Result code selection

Table 6-58. "X" - Result code selection

Syntax	Explanation	
X[<value>]	Setting this parameter defines whether the module transmits particular result codes (see "Parameter Persistence" on page 5-3) to the host, or it does not	
	value	Description
	0 1 2	all result codes are sent to the host **EVENT** result codes are not sent **EVENT** and **DATA** result codes are not sent
	If value is omitted, 0 is implied	
Result codes	**OK** if value is from valid range. Otherwise, **ERROR** is returned.	
Example	ATX2 **OK** — Disable events and data indications	
Node types	Coordinator / Router / End device	
Default value	1 – all result codes will be sent, excluding **EVENT**.	
Persistence	value is stored in the EEPROM.	

6.7.8 "+IPR" - Serial port communication rate

Table 6-59. "+IPR" - Serial port communication rate

Syntax	Explanation
`+IPR=<value>`	The command specifies the data rate at which the DCE will accept commands and will respond. At least, 1200 bit/s and 9600 bit/s are supported, but particular hardware version can support extended set of rates. Note: The rate specified takes effect following the issuance of any result code associated with the current command line even subsequent commands in a command line will return **ERROR**.
`+IPR?`	The command requests for actual communication rate.
`+IPR=?`	The command requests for the list of supported rates. This depends on the hardware capabilities of the particular model.
Result codes	The module returns **OK** if the requested rate is present in the supported list, otherwise **ERROR**.

Example	`AT+IPR=38400`
	OK
	`AT+IPR?`
	+IPR:38400
	OK
	`AT+IPR=?`
	+IPR:(1200,9600,38400)
	OK

Node types	Coordinator / Router / End device
Default value	38400
Persistence	`value` is stored in the EEPROM

6.7.9 "+IFC" - Serial port flow control

Table 6-60. "+IFC" - Serial port flow control

Syntax	Explanation	
`+IFC=<rx_flow>, <tx_flow`	The command is used to specify the methods for local flow control over the UART interface between the host and the module. It accepts two numeric sub-parameters: ■ `rx_flow`, which specifies the method for the host to control the flow of data received from the module ■ `tx_flow`, which specifies the method for the module to control the flow of data transmitted from the host	
	rx_flow	
	`0` `2`	None use RTS (Request to Send) line
	`tx_flow`	
	`0` `2`	None use CTS (Clear to Send) line
	Note: It is strongly recommended to use the CTS method because, if no flow control method is selected, there would be no means to use power-down modes when the module would not accept any data coming to UART.	
`+IFC?`	The command requests for actual flow control settings.	
`+IFC=?`	The command requests to list the flow control settings supported.	
Result codes	**`OK`** is returned if specified flow control combinations are supported, otherwise **ERROR**.	

Example	`AT+IFC=2,2`
	`OK`
	`AT+IFC?`
	`+IFC:2,2`
	`OK`
	`AT+IFC=?`
	`+IFC:(0,2),(0,2)`
	`OK`	

Node types	Coordinator / Router / End device	
Default value	Depends on the hardware version. For MeshBean2 boards it is `0,0`	
Persistence	`value` is stored in the EEPROM	

6.7.10 "&D" - DTR behavior

Table 6-61. "&D" - DTR behavior

Syntax	Explanation	
`&D<value>`	The command specifies the method how the module manages DTR line.	
	value	Description
	0 1	module ignores DTR line module wakes up if it is sleeping, thus it can process the data coming from UART with a shortest delay
S-register	S50 (RW).	
Result codes	**OK** is returned if the requested mode is supported, otherwise **ERROR**.	
Example	AT&D1 **OK**	
Node types	Coordinator / Router / End device	
Default value	0	
Persistence	`value` is stored in the EEPROM.	

6.7.11 S0 - Request for the latest result code

Table 6-62. S0 - Request for the latest result code

Syntax	Explanation	
`S0?`	Request for result code from the latest executed command. If the latest executed command was completed with **ERROR** result code, register `S0` will contain nonzero value. Returned values:	
	0 1 2 3 4 5 6 7 8	no error syntax error improper number of parameters parameter value(s) is out of range (example: `AT+IFC=12,34`) unspecified error requested value cannot be read (example: `+WLQI` command for non-existent link) operation is not permitted in current state (example: setting PAN ID in the connected state or `+WSLEEP` for router) operation cannot be completed due to networking problems, e.g. due to connection loss data transmission error

Table 6-62. S0 - Request for the latest result code

Result codes	Always **OK**	

Example	`AT+WROLE=0+WPWR=30,30`
	ERROR
	`ATS0?`
	6
	OK
		6 is returned as setting +WPWR is not permitted for coordinator

	`AT+ABCD`
	ERROR
	`ATS0?`
	1
	OK
		syntax error

	`AT+IFC=12,34`
	ERROR
	`ATS0?`
	3
	OK
		parameter is out of range

Node types	Coordinator / Router / End device	

6.8 Hardware Control

AT-commands described in this section are supported for ZigBit modules only and provide control over such hardware functionality as GPIO, ADC and PWM.

6.8.1 GPIO configuration

Table 6-63. GPIO configuration

Syntax	Explanation	
`S<reg>=<value`	Command selects configuration of particular GPIO pins. `reg` corresponds to GPIO pins, GPIO0…GPIO8, on the module and it is in the range of 120…128.	

	`value`
	
	`0`
	`3`
	`2`
	`1`
		Description
		
		input pin, no internal pull-up
		output
		tri-state
		input pin, internal pull-up is turned on

	Note: Using of internal pull-up improves noise immunity but take in mind that it results in power consumption increased. On the MeshBean2 board, tri-stated pins are configured as input with no pull-up.	
`S<reg>?`	The command requests for actual GPIO pin configuration.	
Result codes	**OK** is returned if the `value` is in valid range, otherwise **ERROR** is returned.	
Example	`ATS120=1 S121=3` **OK**	Set GPIO0 as input with internal pull-up and GPIO 1 as output

Table 6-63. GPIO configuration

Default value	2, tri-state
Persistence	Values are stored in the EEPROM.
Node types	Coordinator / Router / End device

6.8.2 GPIO

Table 6-64. GPIO

Syntax	Explanation	
`S<reg>=<value>`	he command assigns value to a particular GPIO pin. Each of pins GPIO0…GPIO8 of the module is numbered by `reg` which is in the range of 130…138, correspondingly.	

	`<value>`
	
	`0`
	`1`
		Description
		
		Logical 0
		Logical 1

	Note: Command does not affect any pin configured as input or tri-state.	
`S<reg>?`	The command reads a particular GPIO pin numbered and coded as above, so it returns 0 or 1. If pin is configured for output or as tri-state, returned value is not defined	
Result codes	**`OK`** is returned if `value` is 0 or 1, otherwise **ERROR** is returned.	

Example	`ATS120=1 S121=3`
	`ATS130?`
	`1`
	`OK`
	`ATS131=0`
	`OK`
		Set GPIO0 as input and GPIO1 as output, both with internal pull-up
		GPIO0 is 1
		
		Clear GPIO1

Default value	`0`	
Persistence	Values are not stored in the EEPROM because GPIO pins are configured as tri-state at the startup.	
Node types	Coordinator / Router / End device	

6.8.3 A/D configuration

Table 6-65. A/D configuration

Syntax	Explanation	
S100=<value>	The command selects configuration of particular A/D pins. value is a hexadecimal number containing a bit-field. 4 least significant bits (b0... b3) can be used to enable or disable each of 4 A/D channels. Bits b4... b7 are ignored in value field. If bit is cleared then A/D conversion of a corresponding channel is disabled and A/D pin goes to the high impedance without internal pull-up. Note: Take in mind that enabling A/D conversion increases power consumption. Note: Conversion is executed in single conversion mode (see ATmega datasheet with 125 kHz clock rate and external reference), thus enabling the maximum conversion rate of approximately 5 kbps. Note: Proper conversion results are achieved for ZigBit if the external reference signal of 1.25V is applied to the A_VREF pin. If conversion is disabled on all A/D pins, the A_VREF pin is moved to tri-state. Note: Pins AD4…AD7 can be also used as JTAG port and ADC function for this inputs are disabled. Note: When using the ZigBit module installed on the MeshBean2 board, the following restriction is imposed due to the board schematics. Before configuring or reading of the particular A/D pins, you must configure GPIO6, GPIO7 and GPIO8 for output, next set GPIO6 to 0 while setting GPIO7 and GPIO8 to 1. For example, you must send the following commands: ATS126=3 S127=3 S128=3 ATS136=0 S137=1 S138=1 before performing ATS100=0F See additionally Table 6-66 on page 6-43	
S<reg>?	The command requests for actual A/D configuration.	
Result codes	**OK** is always returned.	
Example	ATS100=08 **OK**	Enable conversion on pin AD3
Default value	00 – disable A/D conversion for all 4 A/D pins	
Persistence	Value is stored in the EEPROM.	
Node types	Coordinator / Router / End device	

6.8.4 A/D

Table 6-66. A/D

Syntax	Explanation	

`S<reg>?`	The command reads particular A/D pin and returns its value in decimal format. `reg` corresponds to pins AD0…AD3 on the module and it is in the range of 101…104. If A/D conversion for particular channel is disabled by the `S100` register, no value is returned.
	Note: When using the ZigBit module installed on the MeshBean2 board, the following restriction is imposed due to the board schematics. Configure GPIO 6, GPIO 7 and GPIO 8 for output. Set GPIO6 to 0 while setting GPIO7 and GPIO8 to 1. Then you can configure or read the particular A/D pins. For example, you must send the following commands:
	`ATS126=3 S127=3 S128=3`
	`ATS136=0 S137=1 S138=1`
	before performing these commands:
	`ATS100=0F`
	`ATS101? S102? S103? S104?`	

Result codes	**OK** is always returned.	

Example	`ATS100=08`
	OK
	`ATS104?`
	125
	OK
		Enable conversion on pin AD3
		
		Read AD3 pin

Node types	Coordinator / Router / End device	

6.8.5 PWM configuration

Table 6-67. PWM configuration

<table>
<tr><th>Syntax</th><th colspan="3">Explanation</th></tr>
<tr><td></td><td colspan="3">The command configures particular PWM channel:</td></tr>
<tr><td>S<reg>=<value></td><td>PWM channel
0
1
2</td><td>Output pin
GPIO0
GPIO1
GPIO2</td><td>reg
140
141
142</td></tr>
<tr><td></td><td><value>
0, 2
1

3</td><td colspan="2">Description
Disable PWM channel
Enable channel, setting non-inverted output polarity (output is low when duty cycle = 0% and it is high when duty cycle = 100%)
Enable channel, setting inverted output polarity (output is high when duty cycle = 0% and it is low when duty cycle = 100%)</td></tr>
<tr><td></td><td colspan="3">Notes: 1. When PWM channel is enabled, the corresponding output pin is configured as output to be controlled by that PWM channel. Duty cycle is set to 0 for the channel. PWM frequency is set for the channel to default value (5kHz) if there was no channel opened, otherwise that very frequency is valid for the channel which has been set the last for any other channel.
2. When PWM channel is disabled by setting reg to 0 or 2, the corresponding output pin is configured as tri-state, so it is fully controlled as GPIO.
3. On MeshBean2 board, GPIO0...GPIO2 pins are connected to LEDs</td></tr>
<tr><td>Result codes</td><td colspan="3">OK is returned if the value is in valid range, otherwise ERROR is returned.</td></tr>
<tr><td>S<reg>?</td><td colspan="3">The command requests for current PWM configuration.</td></tr>
<tr><td>Result codes</td><td colspan="3">OK is always returned.</td></tr>
<tr><td>Example</td><td>ATS140=1
S142=3
OK</td><td colspan="2">Enable PWM channel 0, setting non-inverted polarity output, and enable PWM channel 2, setting inverted polarity output.</td></tr>
<tr><td>Default value</td><td colspan="3">0, disabled</td></tr>
<tr><td>Persistence</td><td colspan="3">value is not stored in the EEPROM.</td></tr>
<tr><td>Node types</td><td colspan="3">Coordinator / Router / End device</td></tr>
</table>

6.8.6 PWM frequency control

Table 6-68. PWM frequency control

Syntax	Explanation		
S<reg>=<value>	The command selects PWM operating frequency for particular PWM channel.		
	PWM channel 0 1 2.	**Output pin** GPIO0 GPIO1 GPIO2	**Frequency reg** 143 144 145
	<value> 0 1 2 3 4	**PWM frequency** 5 kHz 10 kHz 20 kHz 50 kHz 100 kHz	
	In fact, PWM frequency selection for any channel affects all channels (frequency is common for all channels). Changing frequency for any PWM channel results in the reset of duty cycle to 0 for all channels		
Result codes	**OK** is returned if value is in valid range, otherwise **ERR0R** is returned.		
S<reg>?	The command reads PWM operating frequency for particular PWM channel coded as above, so it returns 0 to 4.		
Result codes	**OK** is always returned.		
Example	ATS143=2 **OK** ATS144=4 **OK** ATS143? **4** **OK**	Set PWM frequency to 20kHz for PWM channel 0. Set PWM frequency to 100kHz for PWM channel 1. Request for PWM frequency on channel 0. The latest set frequency is returned which has been set recently for channel 1.	
Default value	0 (meaning 5kHz)		
Persistence	value is not stored in the EEPROM.		
Node types	Coordinator / Router / End device		

6.8.7 PWM duty cycle control

Table 6-69. PWM duty cycle control

Syntax	Explanation		
	The command selects duty cycle value for particular PWM channel.		
S<reg>=<value>	**PWM channel** 0 1 2	**Output pin** GPIO0 GPIO1 GPIO2	**Duty cycle reg** 146 147 148

Table 6-69. PWM duty cycle control

	<value> is an integer number in the range of 0 to 100 representing PWM duty cycle in percents. Note: Currently stated duty cycle on the output pin will be changed as soon as current period of PWM frequency.is ended. Note: Resolution of duty cycle setting depends on the PWM frequency, as below:	
	PWM frequency 5 kHz 10 kHz 20 kHz 50 kHz 100 kHz	**Duty cycle resolution** 1% 1% 1% 2,5% 5%
Result codes	**OK** is returned if value is in valid range, otherwise **ERROR** is returned.	
S<reg>?	The command reads duty cycle given for particular PWM channel in percents.	
Result codes	**OK** is always returned.	
Example	ATS146=45 **OK**	Set duty cycle to 45% for PWM channel 0.
Default value	0 (%)	
Persistence	value is not stored in the EEPROM.	
Node types	Coordinator / Router / End device	

6.9 Remote Management

Remote management functions include the password-protected AT-commands that come from originating node to a target node. The received AT-command sequences are executed on the destination node, as if they would come from a serial port. Information response and result code of the command execution are sent back to the originating node in the form as if they are normally returned over serial interface.

Remote execution service is protected by 32-bit password that can be set during the node installation or manufacturing.

Remote management function is an important tool that allows organization of commissioning procedures on PC, using commercial off-the-shelf terminal software.

6.9.1 "+WPASSWORD" - Set a password

Table 6-70. "+WPASSWORD" - Set a password

Syntax	Explanation
+WPASSWORD <psw>	The command sets a new password for remote management command. Password is in form of 32-bit hexadecimal number. Note: This command is not to be confused with the parameter set commands. Unlike those, it does not include the “=” symbol.
Result codes	**OK** is always returned.
Example	AT+WPASSWORD 65432178 **OK**

Table 6-70. "+WPASSWORD" - Set a password

Default value	0
Persistence	psw value is stored in the EEPROM. Note: The password cannot be reloaded with default value through *&F* command (see Table 6-48 on page 6-30) but it can be rewritten over the air using remote AT-command (see Table 6-71 on page 6-47).
Node types	Coordinator / Router / End device

6.9.2 "R"-Remote execution of AT command

Table 6-71. "R"-Remote execution of AT command

Syntax	Explanation	
R<addr>,<psw>, <cmd>	The command lets the execution of AT-commands on a remote node, with output redirected. Password (psw) is a 32-bit hexadecimal number, which is set for this specific node. addr specifies short (network) address of the destination node. cmd is a sequence of AT-commands without AT prefix. Note: It is strongly recommended not to use the &H and %H commands for cmd, as they produce extremely lengthy output.	
Result codes	All the responses and result codes are received from the remote node in text form and thus can be normally processed. If a connection loss will be detected, the **ERROR** result code will be returned after time-out since last response packet is received (approx 3 sec). In particular, remote execution of +WLEAVE command will result in **ERROR** code, despite being executed successfully. If remote command is send to End device with sleeping period longer than time-out, **ERROR** will be returned. If the controlled node is not in the PAN, **ERROR** will be returned. Remote execution is not allowed for commands that cause the receiving node to send data over the network: D, DU, DS, +WPING, R. Attempting will result in **ERROR** code with the command processing aborted.	
Example	ATR1,65432178,+GMM?+WRSSI 2 **+GMM:ZIGBIT** **+WRSSI:-80** **OK** ATR1,65432178,+WLEAVE **ERROR**	Get model number and RSSI Remove node from network – **ERROR** will be returned but delayed.
Node types	Coordinator / Router / End device	

Section 7

User Guide Revision History

7.1 Rev.8021A – 11/09

1. First version of the BitCloud SerialNet User Guide

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature