Atmel AVR2051:
Atmel BitCloud SerialNet

User Guide

8389B–MCU Wireless–05/12

Table of contents

Section 7

Section 1

Introduction

SerialNet is a manufacturer-specific profile developed on top of the Atmel® BitCloud® C API. It offers control of an embedded BitCloud stack through a serial interface using a standardized AT command set, and requires no embedded API programming. Node parameters can be easily accessed over the air without a 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: http://www.atmel.com/dyn/resources/prod_documents/doc8117.pdf
2. ZigBit® 2.4GHz wireless modules. ATZB-24-A2/B0 datasheet. www.atmel.com/zigbit
3. ZigBit 2.4GHz wireless modules. ATZB-A24-UFL/U0 datasheet. www.atmel.com/zigbit
4. ZigBit 700/800/900MHz 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. AVR® 8-bit microcontrollers with 64K/128K/256K bytes in-system programmable flash: Atmel 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 character
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
GPIO	General purpose input/output

Table 2-1. Abbreviations and acronyms. (Continued)

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	Microcontroller unit/multichip 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

SerialNet 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 the SerialNet protocol, lists hardware 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 Step 1. in Section 2-1	N/A	BitCloud for ATAVRRZRAVEN
ZigBit	ATmega1281 + AT86RF230	ATZB-24-B0 (ZigBit B0); ATZB-24-A2 (ZigBit A2). See Step 2. in Section 2-1	BitCloud for ZigBit modules
ZigBit Amp	ATmega1281 + AT86RF230	ATZB-A24-UFL (ZigBit Amp) See Step 3. in Section 2-1	BitCloud for ZigBit modules
ZigBit 900	ATmega1281 + AT86RF212	ATZB-900-B0 (ZigBit 900) See Step 4. in Section 2-1	BitCloud for ZigBit modules
megaRF	ATmega128RFA1	N/A	BitCloud for megaRF

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

3.2 Conventions

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

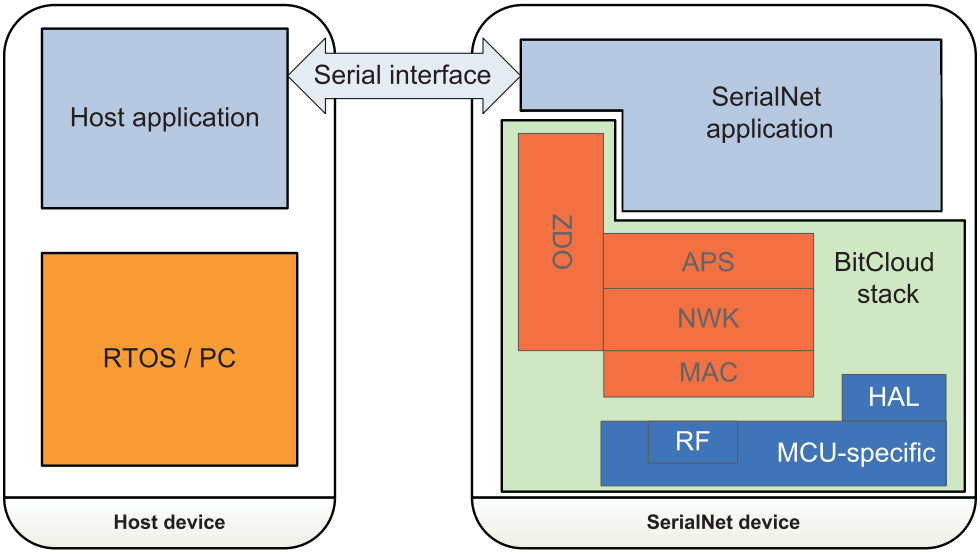
The term *node* will be used in reference to a device's role in the network (end device, router or coordinator).

To be distinguished from the rest, the definitions of commands directed to a module are denoted in the `Courier` font, while module responses are shown in **`Courier Bold`** font. Angle brackets, < >, enclose mandatory parameters. Square brackets, [ ], contain optional parameters.

3.3 Architecture overview

The SerialNet application is developed on top of the Atmel BitCloud ZigBee PRO-certified stack (see - [Step 10.] in Section 2-1). It provides an easy-to-use control over ZigBee PRO networking functionality that is accessible to the host device through a serial connection using an extensive set of AT commands in ASCII format. A SerialNet device executes received requests and responds to the host. Figure 3-1 illustrates the basic architecture.

Figure 3-1. SerialNet usage scheme.

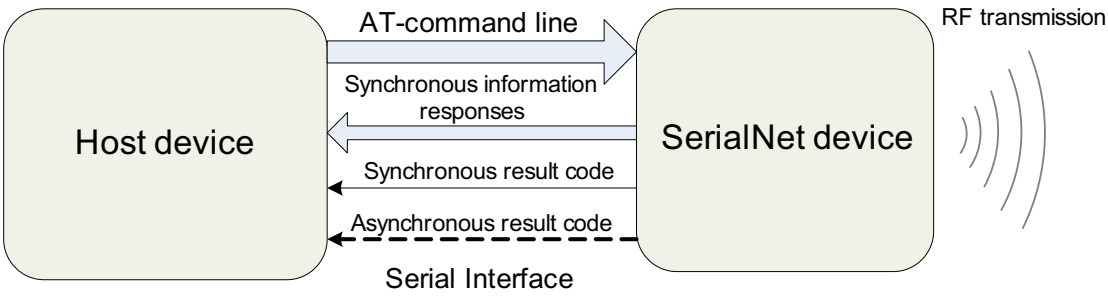


An important feature of SerialNet is the capability to request execution of a particular function over the air via the ATR command (see Table 5-8 on page 5-8). It allows transferring the AT command to the remote node in the network, executing it there and redirecting the execution output to the originator. Thus, the remote node can be monitored and 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-registers to reduce overhead of the serial protocol. The basic principle of the SerialNet protocol is illustrated in Figure 3-2.

Figure 3-2. SerialNet command executions.



The host device shall transmit a command line prefixed by the "AT" string and followed by the chained SerialNet commands to be executed consecutively. Upon successful execution of each command in the sequence, corresponding information responses are 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 any command is executed incorrectly, the command sequence

is interrupted and the **ERROR** result code is returned. The **OK** result code is returned if all commands in the sequence were executed successfully.

Each command in a sequence may have a different syntax, depending on whether it is used to execute an action, read or write parameter(s), or test a 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 -4dBm, request LQI and RSSI for link with node 2, request active channel
Information responses	**+WLQI:254**	LQI value is 254
	+WRSSI:-80	RSSI is -80dBm
	B	Node is operating on channel 0x0B
Result code	**OK**	Execution is completed successfully

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

In addition to synchronous result codes indicating command execution status, a SerialNet device can send asynchronous result codes to the host device upon specific events. 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 Section 6 on page 6-1.

“S-registers” on page 5-5 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) is first programmed (via JTAG, USB or RS-232) with the SerialNet firmware version for the corresponding platform. After that it is connected to a host device (a PC, MCU, etc.) using a USB or RS-232 interface. To start communication, the host device configures its serial port with default SerialNet parameters.

Table 4-1. Default SerialNet parameters.

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

Note that these parameters can be modified for a SerialNet device and saved in persistent memory using the corresponding commands described in Section 6.8.8 on page 6-44 and Section 6.8.9 on page 6-45.

If a PC is the host, the HyperTerminal application included with Microsoft® Windows® can be used to communicate with the SerialNet device. To check the connection, “AT” should be entered in the terminal window, followed by a CR. If the board responds with an `OK` result code, the communication between the host and SerialNet devices has been established successfully.

“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 controlRemote management

Table 5-1 provides a full list of SerialNet commands, along with information about supporting node roles, syntaxes, corresponding S-registers (if any), persistence, and references to the detailed command description in Section 5.4.5 on page 5-9.

Table 5-1. Command summary.

Command	Function	Node type (C/R/E)	S-register	Action syntax	Parameter set syntax	Parameter read syntax	Parameter test syntax	Persistence	Reference
Networking parameters									
+WPANID	Extended PAN ID	CRE	20, 21		x	x	x	x	6.2.1
+WCHAN	Active channel	CRE	22			x			6.2.2
+WCHMASK	Channel mask	CRE	23		x	x	x	x	6.2.3
+WCHPAGE	Channel page	CRE	25		x	x	x	x	6.2.4
+WAUTONET	Automatic networking	CRE	24		x	x	x	x	6.2.5
+WROLE	Node role	CRE	33		x	x	x	x	6.2.6
+GSN or I4	Device extended address	CRE			x	x			6.2.7
+WSRC	Node short address	CRE	55		x	x	x	x	6.2.8
+WNWKPANID	Short (network) PAN ID	CRE	55		x	x	x	x	6.2.9
Network management Function									
+WJOIN	Start/join the network	CRE		x					6.3.1

Table 5-1. Command summary. (Continued)

+WLEAVE	Leave the network	CRE		x					6.3.2
+WNWK	Request for networking status	CRE		x					6.3.3
+WPARENT	Request for parent address	E				x			6.3.4
+WCHILDREN	Request for children addresses	CR				x			6.3.5
+WNBSIZE	Request for a number of neighbor nodes	CRE				x			6.3.6
+WNB	Request for neighbors' information	CRE				x			6.3.7
S30	Network addressing mode	CRE	30		x	x			6.3.8
+WLQI	Request for LQI	CRE		x					6.3.9
+WRSSI	Request for RSSI	CRE		x					6.3.10
Security									
+WSECON	Enable/disable security	CRE	x	x					6.4.1
+WSECSTATUS	Set/get security status	CRE		x					6.4.2
+WNETKEY	Set/get network encryption key	CRE		x					6.4.3
+WTCADDR	Set/get trust center address	E				x			6.4.4
Data transmission									
D	Send data to a specific node	CRE		x					6.5.2
DB	Send binary data to a specific node	CRE		x					6.5.3
DU	Send broadcast data	CRE		x					6.5.4
DS	Send S-register value to a specific node	CRE		x					6.5.5
+WPING	Ping the node	CRE		x					6.5.6
+WSYNCPRD	Indirect poll rate	CRE	37		x	x	x		6.5.7
+WTIMEOUT	Data delivery timeout	CRE	51			x			6.5.8
+WRETRY	Repetition count	CRE	52			x			6.5.9
+WWAIT	Data transmission waiting timeout	CRE	53		x	x	x	x	6.5.10
Power management									
+WPWR	End device sleep parameters	CRE	31, 32		x	x	x	x	6.6.1
+WSLEEP	Force to sleep	E		x					6.6.2
+WTXPWR	TX power level	CRE	34		x	x	x	x	6.6.3
Generic control									
Z	Warm reset	CRE		x					6.7.1
&H	Help	CRE		x					6.7.2
%H	Display parameters and S-register values	CRE		x					6.7.3
I, I0	Display product identification information	CRE		x					6.7.4
+GMI or I1	Request manufacturer identification	CRE		x					6.7.5
+GMM or I2	Request model identification	CRE		x					6.7.6

Table 5-1. Command summary. (Continued)

+GMR or I3	Request hardware/software revision identification	CRE		x					6.7.7
&F	Set to factory-defined configuration	CRE		x					6.7.8
+WACALIBRATE	Configure periodic internal clock calibration	CRE		x					6.7.9
+WCALIBRATE	Calibrate internal clock	CRE		x					6.7.10
Host interface commands									
S3	Termination character	CRE	3		x	x		x	6.8.1
S4	Response formatting character	CRE	4		x	x		x	6.8.2
S5	Command editing character	CRE	5		x	x		x	6.8.3
E	Command echo	CRE		x				x	6.8.4
Q	Result code suppression	CRE		x				x	6.8.5
V	Response format	CRE		x				x	6.8.6
X	Result code selection	CRE		x				x	6.8.7
+IPR	Serial port communication rate	CRE			x	x	x	x	6.8.8
+IFC	Serial port flow control	CRE			x	x	x	x	6.8.9
&D	DTR behavior	CRE	50	x				x	6.8.10
S0	Request the latest result code	CRE	0			x			6.8.11
Hardware control									
S120…S128	GPIO configuration	CRE	120 128		x	x		x	6.9.1
S130…S138	GPIO	CRE	130 138		x	x			6.9.2
S100	A/D configuration	CRE	100		x	x		x	6.9.3
S101…S104	A/D	CRE	101 104			x			6.9.4
S140, S141, S142	PWM configuration	CRE	140 141 142		x	x			6.9.5
S143, S144, S145	PWM frequency control	CRE	143 144 145		x	x			6.9.6
S146, S147, S148	PWM duty cycle control	CRE	146 147 148		x	x			6.9.7
POKE	Writing to hardware registers	CRE		x					6.9.8
PEEK	Reading from hardware registers	CRE		x					6.9.8
Remote management									
+WPASSWORD	Set a password	CRE		x				x	6.10.1
R	Remote execution of AT command	CRE		x					6.10.2

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 the MCU's persistent memory, and in contrast to nonpersistent parameters, they will not be set to default configuration upon device reset.

However, the value assigned to a persistent parameter by the corresponding AT command is not written to the persistent memory right away. Instead, it is applied to the SerialNet operation, but is kept in RAM. SerialNet periodically (with a five-minute interval) verifies whether or not the values of persistent parameters in EEPROM match their actual values in RAM. If differences are detected, the corresponding values in EEPROM are updated. For platforms with warm reset command support (see Table 6-48 on page 6-34), 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 specific events in the network or on a SerialNet device. See a detailed description of result codes in "Device responses" on page 6-6. Table 5-2, which 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 the source node the data is originating from **bcast** is set to 1 if data are sent by broadcast transmission; otherwise, it is set to 0 **length** is a length of the **<data>** field **data** is a byte sequence of received data Note: The +WPING command (see Table 6-40 on page 6-29) results in the following code on the destination node: **DATA <addr>,0,0:**
EVENT	**7**	**:<text>**	text is text specifying an event
		:JOINED	Indicates that the node has joined the network Note: Event is returned in auto-network mode only, and not after the +WJOIN command
		:LOST	Indicates that the node has lost connection to the network (that is, to its current parent) Note: This event can occur on end-device nodes only, and is not returned after +WLEAVE

Table 5-2. Result codes. (Continued)

		:CHILD_JOINED <addr>	Indicates to the node that a 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: This event occurs when the child end device switches to a new parent, when it leaves the network using the +WLEAVE command, or when it is not accessible (powered off, no link, etc.) for *3*(sleep_interval + sync_period),* as configured on the parent device by the +WPWR and +WSYNCPRD commands
		:CALIBR	Indicates that the device has successfully calibrated its internal clock after encountering errors on the 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-registers 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	6.8.11
Termination character	RW	S3	6.8.1
Response formatting character	RW	S4	6.8.2
Command editing character	RW	S5	6.8.3
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	6.6.1
Node role	RW	S33	6.2.6
TX power level	RW	S34	6.6.3
Indirect poll rate	RW	S37	6.5.7
DTR behavior	RW	S50	6.8.10
Data delivery timeout	R	S51	6.5.8
Repetition count	R	S52	6.5.9
Data transmission waiting timeout	RW	S53	6.5.10

Table 5-3. S-registers. (Continued)

Own network address	RW	S55	6.2.8
A/D configuration	RW	S100	6.9.3
A/D	R	S101…S104	6.9.4
GPIO configuration	RW	S120…S128	6.9.1
GPIO	RW	S130…S138	6.9.2
PWM configuration	RW	S140, S141, S142	6.9.5
PWM frequency control	RW	S143, S144, S145	6.9.6
PWM duty cycle control	RW	S146, S147, S148	6.9.7

5.4 Examples

The examples given below show the usage of AT commands to control 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 two nodes. The first step is to configure network parameters. One of the nodes should function as a coordinator, and the others can be routers or end devices. It is important that all nodes have different extended (MAC) and short (NWK) addresses. The coordinator node shall have short address 0, and all other nodes shall have nonzero 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**, it means the embedded software does not support the coordinator function and cannot be configured in such a way. In this case, try checking the coordinator support on other nodes using the AT+WROLE? command, as described in Table 6-18 on page 6-11.

Then 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 sending the AT+WNWK command and performing data exchange between them. For example, on the coordinator:

Table 5-6. Verify networking status on the coordinator.

Command/response	Comment
AT+WNWK	Request networking status
OK	Means the node is in the network
AT+WWAIT=3000 **OK** ATD55 HELLO **OK**	Set three-second timeout to wait for input, and send HELLO to the node with short address 55

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

Table 5-7. Verify networking status on router terminal.

Command/response	Comment
DATA 0000,0,5:HELLO	Data (five bytes) came from the device with address 0 by unicast request

5.4.3 Remote extension

The ATR command provides a 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 the 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:0000000000000002** **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 auto-network mode and command addressing mode

5.4.4 End device power control

This example demonstrates how to configure an end-device node with a certain duty cycle, perform a 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 an 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 (for example, select hardware mode for flow control in HyperTerminal)
AT+WPWR=100,100 **OK** AT+WPWR? **+WPWR:100,100** **OK**	Set duty cycle to 10s sleep / 1s active Verify that the duty cycle is accepted successfully
AT+WJOIN **OK**	Perform a 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 the coordinator to the end device staying in sleep mode

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

Table 5-11. Polling of buffered data from parent.

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

5.4.5 Control of LEDs and DIP switches

The example below is valid only for MeshBean2 development boards. Mapping of ZigBit module I/O pins and their functions on the MeshBean2 board is summarized in Table 5-12.

Table 5-12. GPIO pin 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, physically set DIP switches as SW4:1 to OFF, SW4:2 and SW4:3 to ON, and, next, configure the I/O pins via commands:

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 the LEDs and obtain DIP switch status using the 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 descriptions

6.1 Protocol general description

6.1.1 Character formatting and data rates

Data transmitted between the host and the module over a serial interface conform to the requirements for start-stop data transmission specified in the ITU-T Recommendation V.4 [Step 8.] on 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 10 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 bits per second, at least. Particular character formatting and the data rate can be changed using appropriate AT commands (see Table 6-66 on page 6-44, Table 6-67 on page 6-45 and Section 6.8.6 on page 6-43). The host has the means to explicitly select the 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, and any eighth or higher-order bit(s), if present, are ignored for the purpose of identifying commands and parameters. Lowercase characters (hex codes `0x61` through `0x7A`) are considered identical to their upper-case equivalents (hex codes `0x41` through `0x5A`) when received from the host by the module. Result codes from the module, which are particularly defined, are specified in uppercase.

6.1.3 Basic command-line operations

Command-line editing, echoing, and repeating are done in accordance with 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 7.] in "Related documents and references" on page 2-1.

The module may echo back to the host the characters received from the host, depending on the setting of the E command (see Section 6.8.4 on page 6-42). If 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 6-58 on page 6-40). Next, it checks the editing character, S5 (see Table 6-60 on page 6-41), before considering any other character. This insures the characters will be properly recognized even though they were set to values 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 the S5 parameter (by default, it is the backspace character, BS [hex code 0x08], see Table 6-60 on page 6-41) 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 the 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. The command line starts with AT (characters 0x41 and 0x54) and should contain a sequence of commands in the following syntax formats (Table 6-1):

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 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 to be ignored (the D command, for instance, see Table 6-36 on page 6-26).

If the command line is started with the 'A/' or 'a/' prefix (hex codes 0x41, 0x2F or 0x61, 0x2F), the module immediately repeats 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 (that is, 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 omitted. That parameter would be further treated as if the other subsequent command were input, probably causing an **ERROR** message. To avoid confusion, follow the command syntax.

Actions may have more than one associated sub-parameter, 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 a decimal or hexadecimal number. The choice depends on whether or not a particular command and hexadecimal number is 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. Leading '0' characters will be ignored.

Hexadecimal numbers consist of characters "0" through "9" and "A" through "F," inclusive. The minus sign is not allowed. 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. Thus, if a parameter is 32 bits long, it would be eight characters long, and if it is a 64-bit number, it would contain 16 characters, and so on.

As a special case, a string constant appears in the R command (see Table 6-80 on page 6-55) only. Then it is just a sequence of display-able 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-register command

Parameters may be defined as "read-only" (R) or "read/write" (RW). Read-only parameters are used to provide the host with 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 a 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 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 of the wrong type is specified; or if a value that is not within the supported range is specified.

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 Section 6.1.4 on page 6-2. 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 where 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 the wrong type or beyond the valid range, then 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 an 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 the current value or values stored in a parameter by using the following syntax:

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 the definition of the parameter. Generally, the response string begins with <command> followed by the ':' 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 the sub-parameters, and, possibly, additional information. The format of this information text response is defined for each parameter. See "Information text formats" on page 6-6 for the general formats for the specification of sets and ranges of numeric values. Generally, an information text response starts with a <command> followed by the ':' character.

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 the 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. 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 Section 6.1.4 on page 6-2. Some commands may have no parameters at all. Expected values are 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 (that is, 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 the 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 the header are determined by the user's setting (see V command, Table 6-63 on page 6-43). The 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, result text, and trailer. The characters to be generated in the header and trailer are determined by the user's setting (see the V command, Table 6-63 on page 6-43). The result text may be generated as a number or a string, depending on the user-selected setting (see the V command, Table 6-63 on page 6-43).

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 6-65 on page 6-44) controls the generation of result codes, while command Q (see Table 6-62 on page 6-42) results in their total suppression.

“Parameter persistence” on page 5-4 summarizes the result codes in both verbose and numeric forms with their corresponding parameter(s), if any, and a brief description. Each command description 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 information text format 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 the desired network characteristics prior to executing the network start/join procedure. Note that if the default setting or persistent value from the EEPROM (see "Parameter persistence" on page 5-4) already has the desired value for a network parameter, there is no need to assign it explicitly again prior to the network start/join.

There are 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. Table 6-12 lists such parameters, and provides their values in SerialNet firmware with and without security support.

Table 6-12. Values of ConfigServer parameters applied in SerialNet firmware.

BitCloud ConfigServer parameter	No security support	With security support
CS_NEIB_TABLE_SIZE	25	10
CS_MAX_CHILDREN_AMOUNT	25	10
CS_MAX_CHILDREN_ROUTER_AMOUNT	10	5
CS_ROUTE_TABLE_SIZE	40	25
CS_MAX_NETWORK_DEPTH	6	5
CS_NWK_BTT_SIZE	50	30
CS_ADDRESS_MAP_TABLE_SIZE	10	10
CS_ROUTE_DISCOVERY_TABLE_SIZE	4	3
CS_APS_DATA_REQ_BUFFERS_AMOUNT	4	4
CS_APS_ACK_FRAME_BUFFERS_AMOUNT	3	3
CS_DUPLICATE_REJECTION_TABLE_SIZE	5	5
CS_NWK_BUFFERS_AMOUNT	4	4

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 6-13. "+WPANID" - Set/Get extended PAN ID.

Syntax/descriptor	Explanation
+WPANID=<value>	The command sets the extended PAN ID for the device. value is the extended PAN ID in the form of a 64-bit hexadecimal number that uniquely identifies the target network. If PAN ID is set to 0, the coordinator will form a network with the extended PAN ID equal to its extended (MAC) address. Router and end device nodes in such case will join the first available network, irrespective of 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 the extended PAN ID that is specified on the device for network operation.
+WPANID=?	The command requests a valid range for the extended PAN ID value.
S-register	S21 (RW). This register is just keeping a copy of the parameter accessible through the +WPANID command. S20 (R). This register contains the actual extended PAN ID that is used for networking. If the S21 register is set to 0 and the device is in the network, this register will keep the extended PAN ID of the selected network. If the device has not been connected, this register contains 0.
Result codes	The set command is executed if the device is not in the network and the extended PAN ID is in the valid range. In such case, the device returns **OK** upon completion. Otherwise, the extended PAN ID is ignored, and the 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 6-14. "+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 6-15. "+WCHMASK" - Set/Get channel mask.

Syntax	Explanation
+WCHMASK=<value>-	The command sets the channel mask to be used for network operation. value is a 32-bit field that specifies the channel numbers supported by the node. The five most-significant bits of the channel mask (b31,...,b27) shall be set to 0. The remaining 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: 780MHz: channel numbers 0 – 3 868MHz: channel number 0 915MHz: channel numbers 1 – 10 2.4GHz: channel numbers 11 – 26 For sub-GHz bands, the corresponding channel page shall be configured by the +WCHPAGE command (see Table 6-16 on page 6-10). A detailed description of the channel mask parameter can be found in 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 the actual channel mask. The returned channel mask can be different from the channel mask set by the +WCHMASK=<value> command, and depends on the hardware capabilities. The cleared bits mark unsupported channels.
+WCHMASK=?	The command returns the channel capability mask in the form of two 32-bit unsigned hexadecimal numbers. It returns 00000800-07FFF800 for 2.4GHz chipsets and 00000001-000007FF for sub-GHz. Note: Strictly speaking, these two numbers do not represent "range" in its direct sense, but are rather 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 the channel mask is set according to hardware capabilities actually available. In such case, the device returns **OK**. Otherwise, the channel mask is ignored, and the device responds with **ERROR**.
Example	AT+WCHMASK=40000 **OK** AT+WCHMASK? **+WCHMASK:00040000** **OK** AT+WCHMASK=? **+WCHMASK(00000800-07FFF800)** **OK**
Default value	00000800 for 2.4GHz chipsets or 00000001 for sub-GHz
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 the Atmel AT86RF212 radio part.

Table 6-16. "+WCHPAGE" - Set/Get channel page.

Syntax	Explanation
+WCHPAGE=<value>	The command sets the channel page that will be used for networking. Values 0 and 2 correspond, respectively, to BPSK and O-QPSK modulations on 868/915MHz channels. Value 5 means that the 780MHz frequency band with O-QPSK modulation shall be used. A detailed description of the channel page parameter can be found in 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 the RF212 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 6-17. "+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 the +WJOIN command shall be used for the 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**

Table 6-17. "+WAUTONET" - Enable/Disable automatic networking. (Continued)

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 6-18. "+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** is returned.	
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 6-19. "+GSN" – Set/Get extended (MAC) address.

Syntax	Explanation	
`+GSN=<value>`	The command assigns the 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 the device extended (MAC) address in the 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 the 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 6-20. "+WSRC" - Set/Get short (NWK) address.

Syntax	Explanation
+WSRC=<value>	The command assigns the device short (network) address. value is a 16-bit hexadecimal number that will be used by the device for communication in the network. It shall be unique within the network. There are two approaches for the short address assignment: ■ **Stochastic addressing** is applied if value is set to FFFF when the device is not in the network. In such case, short address is assigned randomly by the stack when the device joins the network. The stack also automatically resolves all possible address conflicts. After network join, this parameter contains the current short address of the node, but after leaving the network, the +WSRC value is reset to FFFF. Hence the node can get a new short address assigned during the next network join. ■ **Static addressing** is applied if value is set to non-FFFF prior to network join. After joining the network, the device will use the assigned value as its short address. Moreover, after leaving the network, the +WSRC value will be kept and reused during the next network join attempt, unless explicitly overwritten with another value. If static addressing is used, the user is responsible for ensuring that the node's short address is unique within the network. All nodes across the network shall use the same addressing mode. Notes: 1. The command is not accessible when the node is joined to a network. 2. The coordinator node shall always have its short address set as 0000. Nodes of other roles shall have nonzero short addresses.
+WSRC?	The command returns the device short address in the form of a 16-bit hexadecimal number. If the node is in the network, the command returns its actual short address independently of the addressing scheme used. If the node is not joined to a network, FFFF is returned if stochastic addressing is used on the device and the user pre-configured address in the case of static addressing.
+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 FFFF default value implies that stochastic addressing will be used on the device upon network join.
Persistence	value is stored in the EEPROM
Node types	Coordinator / router / end device

6.2.9 "+WNWKPANID" - Set/Get short (NWK) PANID

The short PAN ID value is used in ZigBee frame headers during data exchange to identify which network frames belong to. Its 16-bit value is set by the coordinator during network formation.

By default, the short PAN ID is chosen randomly and is intentionally different from the short PAN IDs of other networks present in the same location on channels specified in the channel mask.

Such behavior may lead to the following issue. If the network coordinator leaves the current network while routers continue their operation and then initiates a new network formation, it will establish a network with a short PAN ID different from the short PAN ID of its previous network. Hence, communication between the coordinator and its former network will not be possible.

To avoid the problem described above, the `+WNWKPANID` command can be used to configure a predefined short PAN ID on the network coordinator.

Table 6-21. "+WNWKPANID" - Set/Get short (NWK) PAN ID.

Syntax	Explanation
`+WNWKPANID=<value>`	The command assigns the device short (network) PAN ID of the target network for the device. `value` is a 16-bit hexadecimal number. If the node is configured as the coordinator, then upon the `+WJOIN` command, it will form the network with its short PANID equal to `value`. If `+WNWKPANID` is set to `FFFF`, the stack running on a coordinator automatically assigns the short PAN ID randomly and intentionally different from all the other networks detected while forming its own network. If `+WNWKPANID` is set to `FFFF` on an end device or a router, it will join to any network matching its `+WPANID [ref]` setting. Otherwise it will be able to join only to a network with a target short PAN ID equal to `value`. Notes: 1. The command is not accessible when the node is joined to a network. 2. Using `+WNWKPANID` does not cancel configuration of the `+WPANID` parameter, and both parameters need to be used together to achieve the desired behavior.
`+WNWKPANID?`	The command returns device short address in the form of 16-bit hexadecimal number.
`+WNWKPANID=?`	The command requests the range of valid addresses.
S-register	Not available
Result codes	**`OK`** is returned if `value` is in range; otherwise, **`ERROR`** is returned.

Example	`AT+WNWKPANID=3A2F`
	`OK`
	`AT+WNWKPANID?`
	`+WNWKPANID:3A2F`
	`OK`
	`AT+WNWKPANID=?`
	`+WNWKPANID:(0000-FFFF)`
	`OK`

Default value	`FFFF` Note: The default `FFFF` value implies that no predefined short PAN ID is used by the device.
Persistence	`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 have a dedicated parent node (coordinator or router) during its lifetime in the network. Router nodes use the 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 the node's neighbor table, it will contain information about neighbor coordinator/router nodes.

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

Table 6-22. "+WJOIN" - Start/Join to the network.

Syntax	Explanation
`+WJOIN`	The command forces the device to form a network (for coordinator node) or join an existing network (for router or end device nodes). The desired network and device characteristics shall be set prior to the `+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 6-23. "+WLEAVE" - Leave the network.

Syntax	Explanation
`+WLEAVE`	The command forces the node to leave the network. If the node has any children, it will automatically force them to leave the network, too. Note: Parameters stored in EEPROM persist even after the node leaves the network.
Result codes	**`OK`** is returned on 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 6-24. "+WNWK" – Get networking status.

Syntax	Explanation	
+WNWK	The command requests the 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 6-25. "+WPARENT" - Get parent address.

Syntax	Explanation
+WPARENT?	The command requests the parent node address the device is associated with. The extended (MAC) address of the parent node is returned as a 64-bit hexadecimal number if the S30 register is set to 0. The short (NWK) parent address is returned if the S30 register is set to 1. See Table 6-29 on page 6-19 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 a node role of coordinator or router.
Example	AT+WPARENT? **+WPARENT:0123456789ABCDEF** **OK**
Node types	End devices

6.3.5 "+WCHILDREN" – Get children addresses

Table 6-26. "+WCHILDREN" – Get children addresses.

Syntax	Explanation
+WCHILDREN?	The command requests the addresses of children end devices associated with the node. The extended (MAC) addresses of children nodes are returned as 64-bit hexadecimal numbers if the S30 register is set to 0. The short (NWK) addresses of children nodes are returned if the S30 register is set to 1. See Table 6-29 on page 6-19 for details. The children addresses returned are 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 the 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.

Table 6-26. "+WCHILDREN" – Get children addresses. (Continued)

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 an 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 6-27. "+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 is the current number of occupied entries in node's neighbor table; the second is the maximum possible number of entries (size of the neighbor table).
Result codes	**OK** is returned if the node is in the network. If the 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 6-28. "+WNB" - Get neighbor information.

Syntax	Explanation
`+WNB <node_role> [,<device_addr>]`	The command requests the contents of node's neighbor table. The `node_role` parameter specifies the node role of neighboring nodes to be extracted from the neighbor table. The 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 the `S30` register is set to 0, `device_addr` is accepted as the short (NWK) address. If the `S30` register is set to 1, `device_addr` is expected to be an extended (MAC) address. See Table 6-29 on page 6-19 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 the neighbor's extended address `nwkAddr` – is the neighbor's network address `relationship` – is the neighbor's relationship to the current node (0 – parent, 1 – child, 3 – no relationship) `depth` – is the neighbor's network depth Notes: 1. A neighbor entry is removed from the table if the node, during a 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 seconds). If the neighbor is an end device, then the 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 the actual parent node persists in the table while information about other nodes is removed shortly after an end device join. The same is valid for information about an end device neighbor – in a 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 the node's current neighbor table.
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 6-29. "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 the addressing mode: 0 – extended (64-bit) addressing 1 – short (16-bit) addressing
S30?	The command requests the current addressing mode
Result codes	The command returns **OK** if value is in range; otherwise, it returns **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 6-25 on page 6-16), +WCHILDREN? (see Table 6-26 on page 6-16), and +WNB (see Table 6-28 on page 6-18). These commands use an 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 6-30. "+WLQI" - Get LQI value.

Syntax	Explanation	
+WLQI <addr>	The command requests the LQI 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 LQI value in the range of 0…255. Notes: 1. LQI information can be retrieved for links within a 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 the ATD command. If ATD has not been performed yet, +WLQI may return an irrelevant value.	
Result codes	The node returns **OK** if the device is in the network and the 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 6-31. "+WRSSI" - Get RSSI value.

Syntax	Explanation	
`+WRSSI <addr>`	The command requests the 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 the value **-91** is returned. Notes: 1. RSSI information can be retrieved for links within a 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 the `ATD` command. If `ATD` has not been performed yet, `+WRSSI` may return an irrelevant value.	
Result codes	The node returns **OK** if the device is in the network and the 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 -80dBm
Node types	Coordinator / router / end device	

6.4 Security

SerialNet firmware images with names having _Security at the end support standard security mechanism defined in the ZigBee PRO specification, see [Step 5.] in section "References" on page 2-1. In this scheme, a special128-bit key (called the network key in ZigBee PRO), the same for all devices, is used in the network to encrypt/decrypt messages with the AES-128 algorithm. The security mechanism can be enabled/disabled using the `+WSECON` command and configured as described below. If security is enabled, then the payload of all data frames sent by the device will be encrypted with the network key.

To employ the security mechanism, the network coordinator shall be configured as a trust center (TC) by assigning a `+WTCADDR` value to its extended address. The TC shall also be programmed with a network key (`+WNETKEY` command) prior to network start.

Joining a secured network can be organized in an unsecured or secured way using the `+WSECSTATUS` command. To avoid possible confusion during the network join, all nodes, including the TC, shall have the same `+WSECSTATUS` configuration prior to network join/start.

For an unsecured join, `+WSECSTATUS` shall be set to 3. In this case, only the TC shall know the network key in advance, while other devices usually only have information about the TC's extended address (`+WTCADDR`). In such a configuration, when a new device tries to enter the network, a special message is propagated to the TC informing it about the joining device. Then, the TC replies with the transport key command that contains the current network key. However, to allow the joining device to understand this command, it is delivered over the last hop in an un-encrypted frame. The joining device accepts the key if the sender address of the received transport key command matches its pre-configured TC's extended address, and rejects the key if the addresses don't match. After the key is accepted, the device is authenticated and starts exchanging data using the received network key.

A secured join completely eliminates the un-encrypted key transmission performed in an unsecured join. For a secured join, nodes shall have `+WSECSTATUS` set to 0 and be pre-configured with the network key (`+WNETKEY` command) prior to a network join attempt. The `+WTCADDR` setting is required on the joining device only if it has `+WPANID` set to 0 and `+WSRC` set to `FFFF` (stochastic addressing). In such a configuration, the device expects a transport key command from the TC as in an unsecured join. But in this case, the command payload contains zeros, and, hence, doesn't reveal the secret network key.

The sections below provide detailed descriptions of the commands that shall be used to configure and operate a secured network.

6.4.1 "+WSECON" -Enable/Disable security

Table 6-32. "+WSECON" - Enable/Disable security.

Syntax	Explanation
`+WSECON=<value>`	The command enables/disables security for a network join and data exchange on the device. `value` is a Boolean type. 1 means that the ZigBee PRO standard security mechanism is enabled on the node. 0 implies that encryption is not used by the node. Notes: 1. The command is not accessible when the node is joined to a network. 2. The command configures the `CS_SECURITY_ON` parameter in the BitCloud stack.
`+WSECON?`	The command returns value indicating whether security on the node is enabled or not.
`+WSECON=?`	The command requests the range of supported values.
S-register	Not available.
Result codes	**`OK`** is returned if `value` is in range; otherwise, **`ERROR`** is returned.

Table 6-32. "+WSECON" - Enable/Disable security. (Continued)

Example	AT+WSECON=1 **OK** AT+WSECON? **+WSECON:1** **OK** AT+WSECON=? **+WSECON:(0-3)** **OK**
Default value	0
Persistence	value is stored in the EEPROM
Node types	Coordinator / router / end device

6.4.2 "+WSECSTATUS" - Set/Get security status

Table 6-33. "+WSECSTATUS" - Set/Get security status.

Syntax	Explanation
+WSECSTATUS=<value>	The command configures the security status on the devices as follows: 0 – for secured key transmission; the device should have a pre-configured, nonzero network key prior to network join. 3 – the device obtains the network key from the trust center via un-encrypted transmission over the air. 1, 2 – reserved for high-security mode, not supported in SerialNet. Notes: 1. The command is not accessible when the node is joined to a network. 2. The command configures the CS__ZDO_SECURITY_STATUS parameter in the BitCloud stack. 3. On the trust center, independently of +WSECSTATUS configuration, +WNETKEY shall be set to the non-zero network key to be used for encryption.
+WSECSTATUS?	The command returns device security status.
+WSECSTATUS=?	The command requests the range of valid statuses.
S-register	Not available.
Result codes	**OK** is returned if value is in range; otherwise, **ERROR** is returned.
Example	AT+WSECSTATUS=3 **OK** AT+WSECSTATUS? **+WSECON:3** **OK** AT+WSECSTATUS=? **+WSECSTATUS:(0-3)** **OK**
Default value	0
Persistence	value is stored in the EEPROM
Node types	Coordinator / router / end device

6.4.3 "+WNETKEY" - Set/Get network encryption key

Table 6-34. "+WNETKEY" - Set/Get network encryption key.

Syntax	Explanation
+WNETKEY=<val64bit0,val64bit1>	The command assigns the network key on the device. val64bit0 and val64bit1 are 64-bit hexadecimal numbers that compose the 128-bit secret encryption key to be used for data encryption/decryption in the network. Notes: 1. The command is not accessible when the node is joined to a network. 2. Zero value implies no encryption to be used. 3. The command configures the CS_NETWORK_KEY parameter in the BitCloud stack.
+WNETKEY?	The command returns the network key configuration on the node.
+WNETKEY=?	The command requests the range of valid values for the network key.
S-register	Not available.
Result codes	**OK** is returned if val64bit0 and val64bit1 are in range; otherwise, **ERROR** is returned.
Example	AT+WNETKEY=CCCCCCCCCCCCCCCC, BBBBBBBBBBBBBBBB **OK** AT+WNETKEY? **+WNETKEY:CCCCCCCCCCCCCCCC, BBBBBBBBBBBBBBBB** **OK** AT+WNETKEY=? **+WNETKEY:(0000000000000000-FFFFFFFFFFFFFFFF),(0000000000000000-FFFFFFFFFFFFFFFF)** **OK**
Default value	0000000000000000,0000000000000000
Persistence	val64bit0 and val64bit1 are stored in the EEPROM
Node types	Coordinator / router / end device

6.4.4 "+WTCADDR" - Set/Get Trust Center address

Table 6-35. "+WTCADDR" - Set/Get Trust Center address.

Syntax	Explanation
+WTCADDR=<value>	The command assigns the trust center extended address. value is a 64-bit hexadecimal number that defines the extended address of the trust center node in the network. The trust center itself shall set it to its own extended address. The value shall be configured on the device prior to network join. If the device cannot know the TC address in advance, it can use a FFFFFFFFFFFFA value instead. In this case, the device considers the first node replying with the transport key command during network join to be the network's TC. Notes: 1. The command is not accessible when the node is joined to a network. 2. The command configures the CS_APS_TRUST_CENTER_ADDRESS parameter in the BitCloud stack.
+WTCADDR?	The command returns the trust center address configured on the node.
+WTCADDR=?	The command requests the range of valid values.
S-register	Not available.
Result codes	**OK** is returned if value is in range; otherwise, **ERROR** is returned.

Table 6-35. "+WTCADDR" - Set/Get Trust Center address. (Continued)

Example	AT+WTCADDR=000100001090C96D **OK** AT+WTCADDR? **+WTCADDR:000100001090C96D** **OK** AT+WTCADDR=? **+WTCADDR:(0000000000000001-FFFFFFFFFFFFFFFE)** **OK**
Default value	AAAAAAAAAAAAAAAA
Persistence	`value` is stored in the EEPROM
Node types	Coordinator / router / end device

6.5 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 or D commands 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.

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

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

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

Note: To ensure safe data transmission over a serial interface between a host and an MCU, it is strongly recommended to set hardware flow control (see Table 6-67 on page 6-45 for details). When running terminal software to control the node, the chosen COM port should be set with the hardware flow control option selected.

6.5.1 Parent polling mechanism

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

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

In awake state, an end device polls its parent node periodically every +WSYNCPRD milliseconds (as configured on the end device). The parent node can transmit a data frame to a child only after receiving a 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.

6.5.2 "D" - Send data to a specific node

Table 6-36. "D" - Send data to a specific node.

Syntax	Explanation	
`D <addr>[,[<arq>][,<length>]]` `<data>`	The command sends data to a specific node. `addr` is the 16-bit hexadecimal short (network) address of the destination node. An optional `arq` parameter (equal to 1 or 0) controls the ARQ/nonARQ data delivery mode, with 1 (that is, ARQ) as the default, if omitted. The `length` parameter specifies the length in bytes of the data portion to be sent. It shall not exceed the maximum allowable number (95 bytes for an un-encrypted frame, and 77 bytes for an encrypted frame). If the `length` parameter is omitted, the maximum possible value is implied by default. Data transmission starts either when the specified number of data bytes is received over the serial interface, or when the time interval between two consecutive data symbols exceeds the timeout preset (`+WWAIT` command - see Table 6-44 on page 6-30). Notes: 1. `data` should be preceded by `<CR>` (`S3` character, see Table 6-58 on page 6-40). 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 data transmission requires acknowledgement from the destination node (ARQ is set to 1), then `OK` is returned only after such acknowledgement frame is received. If no acknowledgement is received after +WRETRY transmission attempts, then `ERROR` is returned as a result code for the DB command. If an acknowledgement is not requested (arq is set to 0), then `OK` is returned just upon successful data transmission over the first hop. There is no guarantee on frame delivery to the final destination. If first-hop transmission cannot be performed (channel busy, route unknown, etc.), or if the sending node is not in the network, then `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 timeout expiration before going to the air

Node types	Coordinator / router / end device	

6.5.3 "DB" - Send binary data to a specific node

Table 6-37. "DB" - Send binary data to a specific node.

Syntax	Explanation	
`DB <addr>[,<arq>[,<length>]]` `<data>`	The command sends binary data (not encoded in ASCII symbols) to a specific node. `addr` is the 16-bit hexadecimal short (network) address of the destination node. An optional `arq` parameter (equal to 1 or 0) controls the ARQ/nonARQ data delivery mode. If omitted, then ARQ is used. The `length` parameter specifies the length in bytes of the data portion to be sent. It shall not exceed the maximum allowable number (95 bytes for an un-encrypted frame, and 77 bytes for an encrypted frame). If the `length` parameter is omitted, the maximum possible value is implied by default. `data` should be preceded by `<CR>` (`S3` character. See ""S3" - Termination character" on page 6-40.) This symbol is not included in frame payload to be transmitted over the air, and it is not counted in `length`. Data transmission starts either when the specified number of data bytes is received over the serial interface, or when the time interval between two consecutive data symbols exceeds the timeout preset (`+WWAIT` command, see Table 6-44 on page 6-30). In contrast to the D command, data transmission doesn't start if a `<CR>` code (`S3` character, see See ""S3" - Termination character" on page 6-40.) occurs inside the `data`. Note: 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 data transmission requires an acknowledgement from the destination node (arq is set to 1), then an `OK` result code is returned only after such acknowledgement frame is received. If no acknowledgement is received after +WRETRY transmission attempts, then ERROR is returned as a result code for the DB command. If an acknowledgement is not requested (arq is set to 0), then OK is returned just upon successful data transmission over the first hop. There is no guarantee on frame delivery to the final destination. If first-hop transmission cannot be performed (channel busy, route unknown, etc.), or if the sending node is not in the network, ERROR is returned.	

Example	`ATDB 12,1,5`
	`97CA2`
	`OK`
	`ATDB 12,0`
	`97CA2`
	`HELLO`
	`OK`
		Send `97CA2` to the node with address 12 using ARQ
		
		The same as above, but without using ARQ, and with the node waiting for timeout expiration before sending data

Node types	Coordinator / router / end device	

6.5.4 "DU" - Send broadcast data

Table 6-38. "DU" - Send broadcast data.

Syntax	Explanation	
DU [<length>] <data>	The command sends data using broadcast transmission. The length parameter specifies the length in bytes of the data portion to be sent. It shall not exceed the maximum allowable number (95 bytes for an un-encrypted frame, and 77 bytes for an encrypted frame). If the length parameter is omitted, the maximum possible value is implied by default. Data transmission starts either when the specified number of data bytes is received over the serial interface, or when the time interval between two consecutive data symbols exceeds the timeout preset (+WWAIT command, Table 6-44 on page 6-30). Notes: 1. ATDU is, in fact, shorthand for the ATD command with broadcast address (FFFF) as the destination. 2. Data should be preceded by <CR> (S3 character, see Table 6-58 on page 6-40). This symbol is not transmitted over the air, and it is not counted in length. 3. Data are broadcasted to the whole network (radius 0).	
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	

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

Table 6-39. "DS" - Send S-register value to a specific node.

Syntax	Explanation	
DS <S-reg>,<addr> [,[<arq>]	The command sends the S-register value to a specific node. The default arq parameter (set to 1 or 0) specifies whether the ARQ or non-ARQ data delivery mode is used. 1 is implied if arq is omitted. The addr destination node address should be a 16-bit hexadecimal short (network) address. S-register data are 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 an acknowledgement is requested (arq is set to 1), the node responds with **OK** upon receiving an acknowledgement in several attempts (see parameter +WRETRY, Table 6-43 on page 6-30); 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 cannot 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	

6.5.6 "+WPING" - Ping the node

Table 6-40. "+WPING" - Ping the node.

Syntax	Explanation
`+WPING <addr>`	The command pings the targeted node. `addr` specifies the destination address as a 16-bit hexadecimal short (network) address. This command is equivalent to the `D` command with ARQ and zero data length: `ATD <addr>,1,0`.
Result codes	The node responds with **`OK`** upon receiving an acknowledgement in several attempts (see parameter `+WRETRY`, Table 6-43 on page 6-30); 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

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

Table 6-41. "+WSYNCPRD" - Poll rate for requesting indirect transactions from the parent.

Syntax	Explanation	
`+WSYNCPRD=<rate>`	The command sets the 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 two times smaller than the value of `+WTIMEOUT` (see Table 6-42 on page 6-30). 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 the 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 one second

Default values	`1400`	
Node types	Coordinator / router / end device	
Persistence	`rate` is NOT stored in EEPROM	

6.5.8 "+WTIMEOUT" - Data delivery timeout

Table 6-42. "+WTIMEOUT" - Data delivery timeout.

Syntax	Explanation
+WTIMEOUT?	The command returns the timeout 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

6.5.9 "+WRETRY" - Repetition count

Table 6-43. "+WRETRY" - Repetition count.

Syntax	Explanation
+WRETRY?	The command returns the 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

6.5.10 "+WWAIT" - Data transmission waiting timeout

Table 6-44. "+WWAIT" - Data transmission waiting timeout.

Syntax	Explanation
+WWAIT=<value>	The value parameter sets the timeout (in milliseconds) for the module to wait for entering the D (see Table 6-36 on page 6-26) or the DU (see Table 6-38 on page 6-28) command. If a pause between two consecutive characters coming from the serial interface exceeds the specified timeout, 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 this case, the length is replaced with its actual value according to the data transmitted.
+WWAIT?	The command returns actual timeout value.
+WWAIT=?	The command requests the range of valid timeouts.
S-register	S53 (RW)
Result codes	**OK** is returned if the value is in the range; otherwise, **ERROR** is returned.

Table 6-44. "+WWAIT" - Data transmission waiting timeout. (Continued)

Example	AT+WWAIT=500 **OK** AT+WWAIT? **+WWAIT:500** **OK** AT+WWAIT=? **+WWAIT:(100-5000)** **OK**
Default value	5000
Persistence	value is stored in the EEPROM
Node types	Coordinator / router / end device

6.6 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 the 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 the 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 modules, SerialNet simplifies power management of external peripherals or the host device via the CTS line. If hardware flow control is enabled by the +IFC command (see Table 6-67 on page 6-45), the line becomes high when the ZigBit module is in the sleep state.

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

Table 6-45. "+WPWR" - Configuration of sleep/active intervals.

Syntax	Explanation	
`+WPWR=<sleep>, <active>`	The command sets the duration of sleep and active intervals for end device nodes. The `sleep` duration is specified in 100ms units, but `active` duration isspecified in 10ms units. Zero active period means that the node can be put asleep explicitly only by the `+WSLEEP` command (in which case it will stay asleep for the 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 the maximum time interval the data destined for the child can be buffered. See "Parent polling mechanism" on page 6-25 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 cannot 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 for one minute sleep / 100ms 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 ten seconds if put asleep by the `+WSLEEP` command)	
Persistence	The `sleep,active` values are stored in the EEPROM	
Node types	Coordinator / router / end device	

6.6.2 "+WSLEEP" - Force node to sleep

Table 6-46. "+WSLEEP" - Force node to sleep.

Syntax	Explanation
`+WSLEEP`	The command forces the node into the sleep mode. The command is supported on ZigBit modules only, and is not available on the RZUSBSTICK platform. Important: The node in sleep mode can respond to the subsequent commands with a delay, depending on the sleeping interval specified (see Table 6-45 on page 6-32), the node version, and DTR configuration (see Table 6-68 on page 6-46). The command is accessible only when the node is joined to a network.
Result codes	**OK** is returned for end devices; otherwise, **ERROR** is returned. Note: The command is executed as follows: the node returns the result code first, and then it disables any subsequent commands, completes pending operations, and finally goes into the sleep mode. Wake up occurs as scheduled by the `+WPWR` command or DTR interrupt, if enabled.
Example	`AT+WSLEEP` **OK**
Node types	End devices

6.6.3 "+WTXPWR" - TX power level

Table 6-47. "+WTXPWR" - TX power level.

Syntax	Explanation	
`+WTXPWR=<value>`	The command sets the transmit power level for the device. The `value` represents the 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 the `Z` command (see Table 6-48 on page 6-34).	
`+WTXPWR?`	The command requests the actual TX power level. Notes: 1. Power level resolution is hardware dependent, and may be coarser than 1dB, and so some power values (-4, -6, -8…) may be forbidden despite being within the allowed range. On input, such values are rounded to the nearest allowed value. 2. This command returns only the number set by the `+WTXPWR=` command, and does not indicate the real power level, which can vary due to temperature, supply voltage, and other factors.	
`+WTXPWR=?`	The command requests the allowable range of TX power levels.	
S-register	`S34` (RW)	
Result codes	**OK** is returned if `value` is in the valid range; otherwise, **ERROR** is returned.	

Example	`AT+WTXPWR=-5`
	OK
	`AT+WTXPWR?`
	+WTXPWR:-5
	OK
	`AT+WTXPWR=?`
	+WTXPWR:(-17-3)
	OK
		set -5dBm TX power level

Table 6-47. "+WTXPWR" - TX power level. (Continued)

Default value	Hardware dependent, typically `3`
Persistence	`value` is stored in the EEPROM
Node types	Coordinator / router / end device

6.7 Generic control

6.7.1 "Z" - Warm reset

Table 6-48. "Z" - Warm reset.

Syntax	Explanation
`Z`	The command instructs the device to execute a 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 the RZUSBSTICK platform. Important: The command should be used with caution since it does not send "leaving the network" signals to other nodes, and, hence, can affect PAN integrity. Therefore, it is better to put the node out of the network prior to reset using the `+WLEAVE` command. 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 6-55 on page 6-38). If `Z` is put in a line together with some other commands, the processing of commands placed after `Z` is disabled. The result code is sent after the reset process is completed. During the reset process, some transients may be observed on the module pins (including GPIO) because of the nature of the MCU used. It is strongly recommended to wait until the **OK** result code is received (or an equivalent numerical code) before sending any new command to the module.
Result codes	**OK** is returned if the command is supported by the device's platform. Otherwise, **ERROR** is returned.
Example	`ATZ` **OK**
Node types	Coordinator / router / end device

6.7.2 "&H" - Command Help

Table 6-49. "&H" - Command Help.

Syntax	Explanation
&H	The command outputs a list of valid AT- commands. The listing order may change, depending on the 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

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

Table 6-50. "%H" - Display parameters and S-register values.

Syntax	Explanation
%H	The command outputs the values of parameters and S-registers. The listing order may change, depending on the 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

6.7.4 "I" - Display product identification information

Table 6-51. "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 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 6.7.5 Section 6.7.6 Section 6.7.7 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		

6.7.5 "+GMI" - Get manufacturer identifier

Table 6-52. "+GMI" - Get manufacturer identifier.

Syntax	Explanation	
+GMI? I1	The command instructs the node to output information text 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	

6.7.6 "+GMM" - Request for the model identifier

Table 6-53. "+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	

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

Table 6-54. "+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	

6.7.8 "&F" – Set to factory default configuration

Table 6-55. "&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 a hardware reset just like the Z command, and so all the same precautions should be considered. The result code will be issued according to the actual result code suppression setting (see Table 6-62 on page 6-42), response formatting (see Table 6-63 on page 6-43), and the transmission rate (see Table 6-66 on page 6-44) set before execution of this command. Note that the &F command does not reset the password once it has been set by the +WPASSWORD command (see Table 6-79 on page 6-54).
Result codes	**OK** is always returned.
Example	AT&F **OK**
Node types	Coordinator / router / end device

6.7.9 "+WACALIBRATE" - Configure periodic internal clock calibration

Table 6-56. "+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 that determines the period of calibration, in minutes (that is, how many minutes will elapse between consecutive calibrations). The command is supported on ZigBit modules only, and is a no-op on the RZUSBSTICK platform. It can be used to prevent frequency drift of the MCU's internal RC oscillator, which can impact or even block serial communication with the host.
+WACALIBRATE?	The command returns the period of calibration (in minutes).
+WACALIBRATE=?	The command returns the permitted range of values for the period of calibration.
Result codes	**OK** is returned on successful command completion. Otherwise, value is ignored, and the 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

6.7.10 "+WCALIBRATE" - Calibrate internal clock

Table 6-57. "+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 the RZUSBSTICK platform. It can be used to prevent frequency drift of the MCU's internal RC oscillator, which can impact or even block serial communication with the host.
Result codes	**OK** is returned on successful calibration. Otherwise, the device responds with **ERROR.**
Example	AT+WCALIBRATE **OK**
Node types	Coordinator / router / end device

6.8 Host interface commands

6.8.1 "S3" - Termination character

Table 6-58. "S3" - Termination character.

Syntax	Explanation
`S3=<value>`	The command sets the ASCII code to be used as the 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 this parameter during network operation.
`S3?`	The command requests the actual ASCII code currently used as the termination character.
Result codes	The module returns **`OK`** if `value` is in range; otherwise, **`ERROR`** is returned. Important: The current value of `S3` is used to terminate the command line when entering the `S3` setting command to specify a new command line termination character. However, the result code will use the new value of `S3`-specified in the command line. For example, if `S3` is currently set to 13 when the '`ATS3=30`' command line is issued, the command line must be terminated with a **<CR>** character, but the result code will use the character with a decimal value of 30.

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

6.8.2 "S4" - Response formatting character

Table 6-59. "S4" - Response formatting character.

Syntax	Explanation
`S4=<value>`	The command sets the ASCII code of the character used in response and result code formatting along with the `S3` parameter (see Table 6-58 on page 6-40). The description of the `V` command shows the parameter usage (see Table 6-63 on page 6-43 for details). `value` may be specified in the range of 0…127. Note: It is strongly recommended to avoid changing this parameter during network operation.
`S4?`	The command requests the actual ASCII code currently used as the response formatting character.

Table 6-59. "S4" - Response formatting character. (Continued)

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 to format 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 be formatted using 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

6.8.3 "S5" - Command editing character

Table 6-60. "S5" - Command editing character.

Syntax	Explanation
S5=<value>	The command sets the ASCII code used as the control character to delete the immediately 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 the entering of any subsequent AT- command.
S5?	The command requests the actual ASCII code of the command editing character.
Result codes	The module returns **OK** if value is in range; otherwise, **ERROR** is returned. Note: The new value of S5 will be used when editing of subsequent command lines, and will be applied after processing the line containing the 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

6.8.4 “E” - Command echo

Table 6-61. “E” - Command echo.

Syntax	Explanation	
`E[<value>]`	This parameter instructs the module to echo characters received by the 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`** is returned.	
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	

6.8.5 “Q” - Result code suppression

Table 6-62. “Q” - Result code suppression.

Syntax	Explanation	
`Q[<value>]`	This parameter instructs the module to transmit result codes to the 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 “Result codes” on page 5-4) – is transmitted. Information text transmitted in response to a command is not affected by the setting of this parameter. There are two possibilities for `value`: `0` – The module transmits result codes `1` – Result codes are suppressed and not transmitted If `value` is omitted, 0 is implied.	
Result codes	Nothing will be received for the `ATQ1` command. **`OK`** if `value` is 0; otherwise, the module returns **`ERROR`**.	
Example	`ATQ0` **`OK`**	Enable result codes
	`ATQ1`	Suppress 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	

6.8.6 "V" - Response format

Table 6-63. "V" - Response format.

Syntax	Explanation	
V[<value>]	This parameter defines the contents of the 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. A result code 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 the result code
Node types	Coordinator / router / end device	
Default value	1 – verbose format	
Persistence	value is stored in the EEPROM	

Table 6-59 on page 6-40, below, summarizes the usage of response formats. All references to <CR> mean the character ASCII coded specified in parameter S3 (see Table 6-58 on page 6-40); likewise, all references to <LF> mean the character ASCII coded specified in parameter S4 (see Table 6-59 on page 6-40). Numeric and verbose codes are discussed in "Parameter persistence" on page 5-4.

Table 6-64. 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>**

6.8.7 "X" - Result code selection

Table 6-65. "X" - Result code selection.

Syntax	Explanation	
X[<value>]	This parameter defines whether the module transmits particular result codes (see "Result codes" on page 5-4) to the host or not.	
	value	Description
	0	All result codes are sent to the host
	1	**EVENT** result codes are not sent
	2	**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	

6.8.8 "+IPR" - Serial port communication rate

Table 6-66. "+IPR" - Serial port communication rate.

Syntax	Explanation
+IPR=<value>	The command specifies the data rate at which the DCE will accept commands and respond. At minimum, 1200b/s and 9600b/s are supported, but specific hardware may support an extended set of rates. Note: The rate specified takes effect following the issuance of any result code associated with the current command line, even if subsequent commands in a command line return **ERROR**.
+IPR?	The command requests the actual communication rate.
+IPR=?	The command requests 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, it returns **ERROR**.
Example	AT+IPR=38400 **OK** AT+IPR? **+IPR:38400** **OK** AT+IPR=? **+IPR:(1200,9600,38400)** **OK**

Table 6-66. "+IPR" - Serial port communication rate. (Continued)

Node types	Coordinator / router / end device
Default value	`38400`
Persistence	`value` is stored in the EEPROM

6.8.9 "+IFC" - Serial port flow control

Table 6-67. "+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`	Description
	`0` `2`	None use RTS (request to send) line
	`tx_flow`	Description
	`0` `2`	None use CTS (clear to send) line
	Note: It is strongly recommended to use the CTS method because there would be no means to use power-down modes if no flow control method is selected as the module is not accepting any data from the UART.	
`+IFC?`	The command requests the actual flow control settings.	
`+IFC=?`	The command requests a list of the flow control settings supported.	
Result codes	**`OK`** is returned if the specified flow control combinations are supported; otherwise, it returns **`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	

6.8.10 "&D" - DTR behavior

Table 6-68. "&D" - DTR behavior.

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

6.8.11 "S0" - Request for the latest result code

Table 6-69. "S0" - Request for the latest result code.

Syntax	Explanation	
`S0?`	Request for the result code from the most recently executed command. If that command was completed with an **`ERROR`** result code, register `S0` will contain a nonzero value. Returned values:	
	`0`	No error
	`1`	Syntax error
	`2`	Improper number of parameters
	`3`	Parameter value(s) is out of range (example: `AT+IFC=12,34`)
	`4`	Unspecified error
	`5`	Requested value cannot be read (example: `+WLQI` command for nonexistent link)
	`6`	Operation is not permitted in current state (example: setting PAN ID in the connected state or `+WSLEEP` for router)
	`7`	Operation cannot be completed due to networking problems (example: connection loss)
	`8`	Data transmission error

Table 6-69. "S0" - Request for the latest result code. (Continued)

Result codes	Always **OK**	
Example	AT+WROLE=0+WPWR=30 ,30 **ERROR** ATS0? **6** **OK** AT+ABCD **ERROR** ATS0? **1** **OK** AT+IFC=12,34 **ERROR** ATS0? **3** **OK**	Setting +WPWR is not permitted for a coordinator Syntax error Parameter is out of range
Node types	Coordinator / router / end device	

6.9 Hardware control

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

6.9.1 GPIO configuration

Table 6-70. GPIO configuration.

Syntax	Explanation	
S<reg>=<value>	Command selects the configuration of particular GPIO pins. reg corresponds to the GPIO pins, GPIO0…GPIO8, on the module, and it is in the range of 120…128.	
	value	Description
	0 3 2 1	Input pin, no internal pull-up Output Tri-state Input pin, internal pull-up is turned on
	Note: Use of internal pull-up improves noise immunity, but also results in increased power consumption. On the MeshBean2 board, tri-stated pins are configured as input with no pull-up.	
S<reg>?	The command requests the actual GPIO pin configuration.	
Result codes	**OK** is returned if value is in the valid range; otherwise, **ERROR** is returned.	
Example	ATS120=1 S121=3 **OK**	Set GPIO0 as input with internal pull-up and GPIO1 as output
Default value	2, tri-state	
Persistence	Values are stored in the EEPROM	
Node types	Coordinator / router / end device	

6.9.2 GPIO

Table 6-71. GPIO.

Syntax	Explanation	
S<reg>=<value>	The command assigns value to a particular GPIO pin. Each pin, GPIO0…GPIO8, of the module is numbered by reg, which is in the range of 130…138, correspondingly.	
	<value>	Description
	0 1	Logical 0 Logical 1
	Note: This 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, and returns 0 or 1. If the pin is configured for output or as tri-state, the 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 startup	
Node types	Coordinator / router / end device	

6.9.3 A/D configuration

Table 6-72. A/D configuration.

Syntax	Explanation	

`S100=<value>`	The command selects the configuration of particular A/D pins. `value` is a hexadecimal number containing a bit field. The four least-significant bits (b0... b3) can be used to enable or disable each of four A/D channels. Bits b4... b7 are ignored in the `value` field.
	If a bit is cleared, then A/D conversion of a corresponding channel is disabled and the A/D pin goes to the high-impedance state without internal pull-up.
	Notes: 1. Enabling A/D conversion increases power consumption.
	2. Conversion is executed in single conversion mode (see the ATmega datasheet with 125kHz clock rate and external reference), thus enabling the maximum conversion rate of approximately 5kb/s.
	3. 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.
	4. Pins AD4…AD7 can be also used as a JTAG port, and then A/D conversion functionality for these inputs is disabled.
	5. When using the ZigBit module installed on the MeshBean2 board, the following restriction is imposed due to the board schematics. Before configuring or reading the particular A/D pins, you must configure GPIO6, GPIO7, and GPIO8 for output, and then 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 6-73 on page 6-50.	

`S100?`	The command requests the 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 four A/D pins	
Persistence	`value` is stored in the EEPROM	
Node types	Coordinator / router / end device	

6.9.4 A/D

Table 6-73. A/D.

Syntax	Explanation	

`S<reg>?`	The command reads a 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 a 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 GPIO6, GPIO7, and GPIO8 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	

6.9.5 PWM configuration

Table 6-74. PWM configuration.

<table>
<tr><th>Syntax</th><th colspan="3">Explanation</th></tr>
<tr><td></td><td colspan="3">The command configures a particular PWM channel:</td></tr>
<tr><td></td><td>PWM channel</td><td>Output pin</td><td>reg</td></tr>
<tr><td rowspan="4">S<reg>=<value></td><td>0
1
2</td><td>GPIO0
GPIO1
GPIO2</td><td>140
141
142</td></tr>
<tr><td><value></td><td colspan="2">Description</td></tr>
<tr><td>0, 2
1

3</td><td colspan="2">Disable PWM channel
Enable channel, setting non-inverted output polarity (output is low when duty cycle = 0% and high when duty cycle = 100%)
Enable channel, setting inverted output polarity (output is high when duty cycle = 0% and low when duty cycle = 100%)</td></tr>
<tr><td colspan="3">Notes: 1. When a PWM channel is enabled, the corresponding output pin is configured as output to be controlled by that PWM channel. Duty cycle for the channel is set to 0. PWM channel frequency is set to the default value (5kHz) if no channel has yet been opened. Otherwise, the frequency last set for any other channel is used.
2. When a PWM channel is disabled by setting reg to 0 or 2, the corresponding output pin is configured as tri-state and is fully controlled as GPIO.
3. On the MeshBean2 board, the 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 the valid range; otherwise, ERROR is returned.</td></tr>
<tr><td>S<reg>?</td><td colspan="3">The command requests the 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 noninverted 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>

6.9.6 PWM frequency control

Table 6-75. PWM frequency control.

Syntax	Explanation		
S<reg>=<value>	The command selects the PWM operating frequency for a particular PWM channel		
	PWM channel	**Output pin**	**Frequency `reg`**
	0 1 2.	GPIO0 GPIO1 GPIO2	143 144 145
	<value>	PWM frequency	
	0 1 2 3 4	5kHz 10kHz 20kHz 50kHz 100kHz	
	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 the duty cycle to 0 for all channels.		
Result codes	**OK** is returned if value is in the valid range; otherwise, **ERROR** is returned.		
S<reg>?	The command reads the PWM operating frequency for a particular PWM channel, coded as above, and returns 0 to 4.		
Result codes	**OK** is always returned.		
Example	ATS143=2 **OK** ATS144=4 **OK** ATS143? **4** **OK**	Set the PWM frequency to 20kHz for PWM channel 0. Set the PWM frequency to 100kHz for PWM channel 1. Request the PWM frequency on channel 0. The most recent frequency set is returned.	
Default value	0 (meaning 5kHz)		
Persistence	value is NOT stored in the EEPROM		
Node types	Coordinator / router / end device		

6.9.7 PWM duty cycle control

Table 6-76. PWM duty cycle control.

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

Table 6-76. PWM duty cycle control. (Continued)

	<value> is an integer number in the range of 0 to 100 representing the PWM duty cycle, in percent. Notes: 1. The duty cycle currently set on the output pin will be changed as soon as the current period of PWM frequency is ended. 2. Resolution of the duty cycle setting depends on the PWM frequency, as below:	
	PWM frequency	**Duty cycle resolution**
	5kHz 10kHz 20kHz 50kHz 100kHz	1% 1% 1% 2.5% 5%
Result codes	**OK** is returned if value is in the valid range; otherwise, **ERROR** is returned.	
S<reg>?	The command reads the duty cycle for a particular PWM channel, in percent.	
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	

6.9.8 Reading and writing registers

Table 6-77. Writing to hardware registers.

Syntax	Explanation	
POKE <type>, <addr>, <value>	Command writes <value> at the address specified by <addr> in registers. <type> specifies the type of registers. For more information about registers available for a certain platform, and ranges of addresses and values, refer to the hardware datasheets.	
	<type>	Description
	0 1 3	Radio transceiver registers MCU internal RAM MCU internal EEPROM
	Note: By using this command, it is not possible to write to MCU internal flash memory.	
Result codes	OK is returned if the address is within the allowed range. ERROR is returned if type or address is out of range.	
Example	Write value 0x06 (RX_ON command) to the TRX_STATE register (of Atmel AT86RF231 transceiver): ATPOKE 0, 002, 00000006 OK	
Node types	Coordinator / router / end device	

Table 6-78. Reading from hardware registers.

Syntax	Explanation	
`PEEK <type>, <addr>`	Command reads the value from registers at the address specified by `<addr>`. `<type>` specifies the type of registers. For more information about registers available for a certain platform, and ranges of addresses and values, refer to the hardware datasheets.	
	`<type>`	Description
	0 1 2 3	Radio transceiver registers MCU internal RAM MCU internal flash MCU internal EEPROM
Result codes	`OK` is returned if the address is within the allowed range. `ERROR` is returned if type or address is out of range of available addresses.	

Example	Write and read GPIOR1 (address 0x004A) register of the Atmel ATmega1281 MCU:
	`ATPOKE 1, 004A, 000000AA`
	`OK`
	`ATPEEK 1, 004A`
	`000000AA`
	`OK`	

Node types	Coordinator / router / end device	

6.10 Remote management

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

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

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

6.10.1 "+WPASSWORD" - Set a password

Table 6-79. "+WPASSWORD" - Set a password.

Syntax	Explanation
`+WPASSWORD <psw>`	The command sets a new password for the remote management command. The password is in form of a 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 6-79. "+WPASSWORD" - Set a password. (Continued)

Default value	0
Persistence	The `psw` value is stored in the EEPROM. Note: The password cannot be reloaded with a default value through the `&F` command (see Table 6-55 on page 6-38), but it can be rewritten over the air using the remote execution AT command (see Table 6-80 on page 6-55).
Node types	Coordinator / router / end device

6.10.2 "R" - Remote execution of AT command

Table 6-80. "R" - Remote execution of AT command.

Syntax	Explanation	
`R<addr>,<psw>,<cmd>`	The command enables 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 the short (network) address of the destination node. `cmd` is a sequence of AT commands without the `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 that can be processed normally. If a connection loss is detected, the **`ERROR`** result code will be returned after a timeout from when the last response packet was received (approx. three seconds). In particular, remote execution of the `+WLEAVE` command will result in an **`ERROR`** code, despite being executed successfully. If the remote execution command is sent to an end device with a sleeping period longer than the timeout, **`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 this will result in an **`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. 8369B – 05/12

B. Section 6.9.8, and “POKE” and “PEEK” in Table 5-1 is added.

7.2 Rev. 8369A – 08/11

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section is referring to the document revision.

A. Initial version

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

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

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
16F, Shin Osaki Kangyo Bldg.
1-6-4 Osaki Shinagawa-ku
Tokyo 104-0032
JAPAN
Tel: (+81) 3-6417-0300
Fax: (+81) 3-6417-0370